Research on a semiautonomous mobile robot for loosely structured environments focused on transporting mail trolleys

by

Víctor Javier GONZÁLEZ VILLELA

A Doctoral Thesis Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy (PhD) of Loughborough University

April 2006

To my family

Jaqueline

Brenda and Andrés

To my parents

Loló and José†

Abstract

In this thesis is presented a novel approach to model, control, and planning the motion of a nonholonomic wheeled mobile robot that applies stable pushes and pulls to a nonholonomic cart (York mail trolley) in a loosely structured environment. The method is based on grasping and ungrasping the nonholonomic cart, as a result, the robot changes its kinematics properties. In consequence, two robot configurations are produced by the task of grasping and ungrasping the load, they are: the single-robot configuration and the robot-trolley configuration. Furthermore, in order to comply with the general planar motion law of rigid bodies and the kinematic constraints imposed by the robot wheels for each configuration, the robot has been provided with two motorized steerable wheels in order to have a flexible platform able to adapt to these restrictions.

The approach guarantees the collision avoidance while the robot is reaching a point or tracking a desired trajectory. To ensure the application of stable pushes and pulls, the cart is grasped by the robot at two points to create a fixed line of contact between them. As a result, the robot loses mobility and changes its kinematics, dynamics and size. In consequence, the approach studies two configurations: the single-robot configuration and the robot-trolley configuration.

The approach is based on proposing a method for obtaining a set of kinematic constraints that are associated with each wheel of the robot. Then, a configuration kinematic state space representation of the robot is obtained by finding a smooth codistribution from the kinematic constraints, where for first time a set of holonomic parametric functions that are useful for coordinating the steering of multiple steerable wheels are presented. These holonomic parametric functions ensure the existence of an annihilator, which is a distribution that spans the null space of the kinematic constraints of the robot. On the other hand, the Lagrange formalism is applied to obtain the dynamic equation of motion at the configuration space. After that, some transformations are done on the dynamic equation of motion at the configuration space to find the dynamic equation of motion at the steering space, which includes the torque components related to the motorized steerable wheels that are useful to determine the correct motor configuration of the robot.

To eliminate the nonlinearity of the kinematic and dynamic models of the robot, the nonlinear methods of the input and input-output static state feedback linearization are applied. As a result, a state space representation of the system in new coordinates is obtained, constituting both a controllable linear subsystem and a nonlinear subsystem that has stable zero dynamics, but no asymptotically stable, with internal unstable behaviour when going backwards. The linear part of the system, expressed in new coordinates, is stabilized by a linear feedback control law. On the other hand, the robot-trolley configuration has been added with a virtual steerable wheel, located somewhere on the robot platform, to control a specific point located on the virtual steerable wheel. In this way, it is possible to control indirectly the movements of a specific point located somewhere on the robot platform and produce the commands to coordinate the motorized steerable wheels.

On the other hand, a novel and generic reactive motion planning approach based on virtual goals and tracking defined in a goal domain space is proposed. As a result of working on a goal domain space, the modelling of both the attractive and the repulsive goal do not have to belong to a potential field function. The method is based on tracking a given collision free trajectory, when the robot detects an obstacle a reactive reference trajectory is computed to be tracked by the robot to avoid collision. The reactive reference trajectory is a deformation of the original collision free trajectory, which is found by adding both the effects produced for the attractive and the repulsive goal models. The collision free trajectories are first defined in local coordinates in a relative time framework and transformed in absolute coordinates.

The simulation results for the selected single-robot and robot-trolley configuration are presented to illustrate the efficacy of the method and to compute the required velocity and torque commands needed for the motorized joints. First, the single-robot and the robot-trolley configuration are tested on simple tasks, such as: point stabilization, line tracking, and circle tracking. Then, the single-robot and the robot-trolley configuration are tested in a loosely structured environment carrying out the complete task of moving a trolley from one position to another. Finally, the robot is tested whilst avoiding fixed obstacles in corridors and moving obstacles in open areas while tracking a reference trajectory.

Key words

Mobile manipulation, nonholonomic mobile robots, push and pull, kinematics, dynamics, control, coordinating multiple steerable wheels, motion planning, trajectory deformation, obstacle avoidance, virtual goals, virtual wheels.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. R.M. Parkin, for his invaluable and continue support, supervision, and availability over these years that made possible to define, develop and finish this research.

I like to thank to my family who have shared this journey with me, my wife Jaqueline, my daughter Brenda and my son Andrés. Also, to all of them who have shared with me this long journey from the distance, specially my parents Loló and José†, and all my close relatives.

I also would like to thank to all of the people at the Mechatronics Research Centre for sharing their information, knowledge, and priceless friendship. I am grateful to Dr. M.R. Jackson for his appreciable advices, to Dr. F. Zorriassatine and Dr. A. Al-Habaibeh for all his appreciable friendship and suggestions, and to Phill, Neil, Ian, Lian, Casey, Bilal, Pin, Sultan, and Nashtara for their enjoyable friendship.

At Loughborough, I am grateful to Celia and Geoff for sharing their time with my family in such lovely way, Erika and William, Claudia and David, and Lilia and Martin for their friendship.

Heartfelt thanks to all of them who have supported me at the National Autonomous University of Mexico (UNAM), especially to Dr. Marcelo López-Parra for his support, advice, feedback and friendship over these years, Dr. Jesús M. Dorador González for his support and friendship, and all my friends at the "Departamento de Mecatrónica" and the "Centro de Diseño y Manufacura" of the "Universidad Nacional Autonoma de México."

Also, I would like to acknowledge to the "Facultad de Ingeniería" and the "Dirección General de Asuntos del Personal Académico", of the "Universidad Nacional Autónoma de México" for sponsoring this research.

Víctor J. González Villela, Loughborough, UK, April 2006

Glossary of terms

AG: Artificial goal.

AGV: Automatic guided vehicle.

AGVS: Automatic guided vehicles system.

Cancelling mail: The action of putting a sale on the mail stamp.

Facing mail: The action of putting the mail in the right position and orientation to be processed.

GDS: Goal domain space.

Generalized coordinates. The *generalized coordinates* of a mechanical system are all those displacement variables, whether rotational or translational, which determine uniquely a configuration of the system.

Holonomic constraints: Constraints that when they are integrated their generalized coordinates are interconnected by geometric constraints.

ICR: Instantaneous centre of rotation.

Involutive distribution: Distribution that is closed under the Lie bracket operation.

Lie bracket: The Lie bracket is the vector $[f,g]=\frac{\partial g}{\partial x}f-\frac{\partial f}{\partial x}g$ and the recursive operation is established by $ad_f^i=\left[f,ad_f^{i-1}g\right]$.

Lie derivative. The Lie derivative of h with repect to f is the scalar $L_f h=\frac{\partial h}{\partial x}f$ and derivatives satisfy the recursion $L_f^i h=L_f\left(L_f^{i-1}h\right)$ with $L_f^0 h=h$.

MHIA: Material Handling Industry of America.

MIMO: Multiple-input-multiple-output.

Nonholonomic constraints: Constraints that their generalized velocities are interconnected by kinematic constraints. They cannot be integrated.

PCV: Powered caster vehicle.

RM: Royal Mail.

SGV: Self-guided vehicle.

SISO: Single-input-single-output.

Sorting mail: The action of classifying mail by delivering address.

Tray: A box for carrying and storing mail.

Tote: Large bag for carrying and storing mail.

Vector field. A vector field on C is a map $x: q \in \rightarrow x(q) \in T_q(C)$.

Virtual wheel: A wheel that dynamic parameters are zero (mass and moments of inertia) and can be added to the wheeled mobile robot altering or not altering the robot type characteristic (i.e., the degree of mobility and degree of steerability).

WMR: Wheeled mobile robot.

Nomenclature

Subscripts:

c : Denotes centred steerable wheels.

f : Denotes fixed wheels.

ϕ : Denotes rotational coordinates.

oc : Denotes off-centred wheels.

l : Denotes left side.

p : Denotes posture coordinates or parametric functions.

q : Denotes configuration coordinates.

r : Denotes remaining coordinates or right side.

s : Denotes parameterized centred angular coordinates

T : Denotes total.

v : Denotes virtual.

z : Denotes new coordinates.

ξ : Denotes posture of the robot.

A : A point defined on the robot frame.

a : Acceleration of the straight line trajectory.

$A(q_p)$: Decoupling matrix.

$A_T(q)$: Matrix associated to the external kinematic constraints.

$A_{T_1}(q)$: Matrix associated to the internal kinematic constraints.

$A_z(z)$: Decoupling matrix expressed in new coordinates.

α : Steering angle of the wheel.

$\dot{\alpha}$: Steering velocity of the wheel.

N.B.: For chapters IV to VIII.

$\alpha_c = [\alpha_{c1} \quad ... \quad \alpha_{cN_c}]^T$: Centred angular coordinates of the fixed and centred wheels.

α_{cr} : Steering angle of the right centred steerable wheel of the robot-trolley.

α_{cl} : Steering angle of the left centred steerable wheel of the robot-trolley.

α_{cv} : Steering angle of the virtual centred steerable wheel of the robot-trolley.

$\alpha_{oc} = [\alpha_{oc1} \quad ... \quad \alpha_{ocN_{oc}}]^T$: Off-centred angular coordinates of the off-centred wheels.

$\alpha_p(\alpha_s) = [\alpha_{p1} \quad ... \quad \alpha_{pN_c - r_s}]^T$: Parametric functions.

$\alpha_s = [\alpha_{s1} \quad ... \quad \alpha_{sr_s}]^T$: Parameterizer centred angular coordinates.

$\dot{\alpha}_c = [\dot{\alpha}_{c1} \quad ... \quad \dot{\alpha}_{cN_c}]^T$: Generalized angular velocities of the centred wheels.

$\dot{\alpha}_{oc} = [\dot{\alpha}_{oc1} \quad ... \quad \dot{\alpha}_{ocN_{oc}}]^T$: Generalized angular velocities of the off-centred wheels.

$\dot{\alpha}_{p_i}$: Generalized parametric centred angular velocity of the $i-th$ parametric function.

b : Wheel parameter.

B : A point defined on the rod of an off-centred wheel.

$B(x_p)$: Matrix associated to the system in new coordinates.

$B_z(z)$: Matrix associated to the system expressed in new coordinates.

(c_{x_1}, c_{y_1}) : Position of centre of mass with respect to the robot frame of reference.

C : A point in the world frame.

$C \rightarrow R^2$: Configuration space.

$C(q,\dot{q})\dot{q}$: Vector of centripetal and Coriolis forces.

$\overline{C}(q,\dot{q})u$: Vector of centripetal and Coriolis forces expressed at the steering space.

$C_T(q,\dot{q}) = 0$: Kinematic constraints.

$C_{T_i}(q) = 0$: Holonomic constraint or geometric constraint.

d : Wheel parameter.

d_c : Wheel parameter of the robot-trolley configuration.

d_f : Wheel parameter of the robot-trolley configuration.

$d_{q_1} = \dim[A_{T_1}(q)]$: Dimension of $A_{T_1}(q)$.

d_v : Wheel parameter of the robot-trolley configuration.

Δ_p : Distribution spanned by the vectors fields of $S_{q_p}(q)$.

$\overline{\Delta}_p$ Involutive closure of Δ_p.

Δ_T : Distribution spanned by the vectors fields of $S_T(q)$.

$\overline{\Delta}_T$ Involutive closure of Δ_T.

e : Rod length of the off-centred steerable wheels.

e_{xr} : Robot reference point tracking error along x axis.

e_{yr} : Robot reference point tracking error along y axis.

$f_p(x)$: Drift vector of the posture dynamical state space model.

$f_T(x)$: Drift vector of the configuration dynamical state space model.

ϕ : Rotation angle of the wheel around its horizontal axle.

$\dot{\phi}$: Rotational velocity of the wheel around its horizontal axle.

ϕ_{cl} : Rotation angle of the centred-left wheel of the robot-trolley.

ϕ_{cr} : Rotation angle of the centred-right wheel of the robot-trolley.

ϕ_{cv} : Rotation angle of the virtual centred wheel of the robot-trolley.

ϕ_f : Rotation angle of the left wheel of the single-robot.

ϕ_{fl} : Rotation angle of the fixed-left wheel of the robot-trolley.

ϕ_{fr} : Rotation angle of the fixed-right wheel of the robot-trolley.

ϕ_r : Rotation angle of the right wheel of the single-robot.

φ: Denote vector of rotational coordinates of the wheels.

$\dot{\phi}$: Denote vector of rotational velocities of the wheels.

$\varphi = [\varphi_f \quad \varphi_c \quad \varphi_{oc}]^T$: Rotation coordinates of the wheels.

$\varphi_c = [\phi_{c1} \quad \ldots \quad \phi_{cN_c}]^T$: Rotation coordinates of the centred wheels.

$\varphi_f = [\phi_{f1} \quad \ldots \quad \phi_{fN_f}]^T$: Rotation coordinates of the fixed wheels.

$\varphi_{oc} = [\phi_{oc1} \quad \ldots \quad \phi_{ocN_{oc}}]^T$: Rotation coordinates of the off-centred wheels.

$\dot{\varphi} = [\dot{\varphi}_f \quad \dot{\varphi}_c \quad \dot{\varphi}_{oc}]^T$: Generalized rotational velocities.

$g_p(x)$: Input matrix of the posture dynamical state space model.

$g_T(x)$: Input matrix of the configuration dynamical state space model.

$G_a(z_1, x_d)$: Scalar attractive function.

$G_{ri}(z_1, x_{oi})$: Scalar repulsive function .

$\overline{G}_a$: Attractive goal vector expressed in local coordinates.

$\overline{G}_r$: Repulsive goal vector expressed in local coordinates.

$\overline{G}_{ref}$: Reference goal expressed in local coordinates.

$\sigma_{m1}(q),...,\sigma_{mr_m}(q)$: Smooth vector fields of $\Sigma(q)$ associated to the degree of mobility.

$\sigma_{s1}(q),...,\sigma_{sr_s}(q)$: Smooth vector fields of $\Sigma(q)$ associated to the degree of steerability.

$h(q_p)$: Output equations.

$\eta(t) = [\eta_1 \quad \ldots \quad \eta_{r_m}]^T$: Mobility input velocity vector.

η_i: $i-th$ constant gain for obstacle modelling.

$I_A \in R^{A \times A}$: Identity matrix of dimension $A \times A$.

I_{yyw} : Moment of inertia of the wheel about the y axis of the wheel reference frame.

I_{zzc} : Moment of inertia of the cart about z axis.

I_{zzw} : Moment of inertia of the wheel about z axis.

$\zeta = [\zeta_1 \quad ... \quad \zeta_{r_s}]^T$: Steerability input velocity vector.

λ : Vector of the langrage multipliers.

Λ_1 and Λ_2 : Arbitrary positive diagonal matrices of $r_M \times r_M$.

m_c : Mass of the cart.

m_T : Total mass of the robot.

m_{Tot} : Total number of bilateral kinematic constraints.

m_w : Mass of the wheel.

$M(q)$: Inertial matrix of the system.

M_h : Degree of holonomy.

M_{nonh} : Degree of nonholonomy

$\overline{M}(q)$: Inertial matrix of the system expressed at the steering space.

n_T : Total number of configurations coordinates.

N : Number of wheels.

$N(A_{T1})$: Null space of $A_{T1}(q)$.

O : Origin of the world frame of reference.

$P = (x, y) = (x_p, y_p)$: Origin of the robot frame of reference.

$P \in R^{n_T \times n_T}$: Input transformation matrix.

$P_{\alpha cr}$: Power required for steering the right wheel of the robot-trolley.

$P_{\alpha cl}$: Power required for steering the left wheel of the robot-trolley.

P_{ϕ_r} : Power required for rotating the right wheel of the single-robot.

P_{ϕ_l} : Power required for rotating the left wheel of the single-robot.

$P_{\phi_{cr}}$: Power required for rotating the right wheel of the robot-trolley.

$P_{\phi_{cl}}$: Power required for rotating the left wheel of the robot-trolley.

P_η : Power required for the variable η .

P_ζ : Power required for the variable ζ .

$\overline{P}$: Input transformation matrix expressed at the steering space.

$q = [q_1,...,q_{n_T}]^T$: Configuration coordinates.

$q = [\xi \quad \alpha_c \quad \alpha_{oc} \quad \varphi]^T$: Generalized coordinates of conventional wheeled mobile robots.

$\dot{q} = S_T(q)u$: State space representation of the configuration kinematic model.

$\dot{q} = [\dot{\xi} \quad \dot{\alpha}_c \quad \dot{\alpha}_{oc} \quad \dot{\varphi}]^T$: Generalized velocities of conventional wheeled mobile robots.

$\dot{q}_1 = [\dot{\xi}_1 \quad \dot{\alpha}_c \quad \dot{\alpha}_{oc} \quad \dot{\varphi}]^T$: Vector of the generalized velocities defined at the robot frame.

q_i : $i-th$ generalized coordinates of the system.

$\dot{q}_p = S_{q_p}(q)u$: Posture kinematic model.

$\dot{q}_r = S_{q_r}(q)u$: Remaining generalized velocities.

$Q(q_p)$: Matrix associated to the nonlinear part of the system in new coordinates.

$Q_z(z)$: Matrix associated to the nonlinear part of the system expressed in new coordinates.

r : Radius of the wheel.

r_c : Radius of the circular trajectory.

r_f : Number of common axles of conventional fixed wheels.

r_m : Degree of mobility .

$r_M = r_m + r_s$: Degree of manoeuvrability.

$r_{q_1} = rank[A_{T_1}(q)]$: Rank of $A_{T_1}(q)$.

r_s : Degree of steerability.

$R(\theta)$: Posture rotation matrix.

R_q : Configuration rotation matrix.

ρ_{0i} : Limit distance of the repulsive goal function influence.

$\rho_{ci} = \|z_{oi} - z_1\|$: Distance to the centre of the obstacle.

ρ_{ri} : Radius of the circular obstacle.

$S_p(q)$: Matrix associated to the state space representation of the posture kinematic model of the robot.

$S_T(\dot{q})$: Matrix associated to the state space representation of the configuration kinematic model of the robot.

$\Sigma(q)$: Matrix consisting of the smooth vectors $[\sigma_1(q) \quad ... \quad \sigma_{r_M}(q)]$.

$\Sigma_c(\alpha_s)$: Submatrix of $\Sigma(q)$ associated to the centred angular coordinates.

$\Sigma_\varphi(\alpha_s, \alpha_{oc})$: Submatrix of $\Sigma(q)$ associated to the rotation coordinates.

$\Sigma_{oc}(\alpha_s, \alpha_{oc})$: Submatrix of $\Sigma(q)$ associated to the off-centred angular coordinates.

$\Sigma_p(\alpha_s)$: Submatrix of $\Sigma(q)$ associated to the parametric centred angular coordinates.

$\Sigma_\xi(\alpha_s)$: Submatrix of $\Sigma(q)$ associated to the posture coordinates.

T : Kinetic energy of the robot.

t : Current time defined in absolute time.

t_L : Duration time of the trajectory.

t_{rel} : Relative time of the straight line trajectory.

t_o : Time when the trajectory started defined in absolute time.

$t(q_p)$: Function that is chosen such that $\Phi(x)$ is no singular.

$type(r_m, r_s)$: Type of mobile robot.

τ : Input torque vector.

$\tau_{\alpha_{cr}} = T_{\alpha_{cr}}$: Steering torque of the centred right wheel robot-trolley.

$\tau_{\alpha_{cl}} = T_{\alpha_{cl}}$: Steering torque of the centred left wheel robot-trolley.

$\tau_{\theta} = T_{\theta}$: Torque associated to the rotation of the single-robot platform.

$\tau_{\phi_{cr}} = T_{\phi_{cr}}$: Rotational torque of the centred right wheel robot-trolley.

$\tau_{\phi_{cl}} = T_{\phi_{cl}}$: Rotational torque of the centred left wheel robot-trolley.

$\tau_{\eta} = T_{\eta}$ Torque needed to compute the rotational torques $\tau_{\phi_{cr}}$ and $\tau_{\phi_{cl}}$.

$\tau_{\phi_l} = T_{\phi_l}$: Rotational torque required at the left wheel single-robot.

$\tau_{\phi_r} = T_{\phi_r}$: Rotational torque required at the right wheel single-robot.

τ_i : Generalized force at joint i.

$\tau_{\zeta} = T_{\zeta}$: Torque needed to compute the steering torques $\tau_{\alpha_{cr}}$ and $\tau_{\alpha_{cl}}$.

τ_u : Driving torque vector.

$\tau_{x_1} = T_{x_1}$: Force associated to the linear displacement of the single-robot along the x_1 axis.

$\bar{\tau}$: Input torque vector expressed at the steering space.

θ : Orientation of the robot frame.

$\dot{\theta} = \omega$: Angular velocity of the robot frame.

θ_c : Angular initial condition of the circular trajectory.

θ_T : Orientation with respect to the global reference frame of the circular trajectory.

u : Input vector of the configuration and posture kinematic state space representations.

$\dot{u}$: Derivate of u.

$v = \dot{x}_1$: Linear velocity of the robot frame along the x_1 axis.

v_0 : Initial velocity of the straight line trajectory.

v_t : Tangential velocity of the circular trajectory.

v : Input vector of the configuration and posture dynamical state space representations.

w : Input vector of the linearized system in new coordinates.

$\omega = \dot{\theta}$: Angular velocity of the robot frame.

x: Position of the point P along the x axis.

$\{x, y\}$: Inertial fixed world frame of reference.

$\|x\| = \sqrt{x^T x}$: Norm.

$x_0 \in R^2$: Initial position.

(x_0, y_0): Origin of the local coordinates of a trajectory.

$\{x_1, y_1\}$: Moving robot frame of reference.

$\dot{x}_1 = v$: Linear velocity of the robot frame along the x_1 axis.

$\{x_2, y_2\}$: Wheel frame of reference for fixed and centred steerable wheels.

$\{x_3, y_3\}$: Wheel frame of reference for off-centred wheels.

$x_f \in R^2$ Final position.

$(x_d, y_d) = z_{1ref}$: A reference trajectory.

$x_{oi} \in R^2$: Reference position of the $i-th$ obstacle.

$(x_p, y_p) = (x, y)$: Origin of the robot frame of reference (point P).

$x_p = [q_p \quad u]^T$: Generalized coordinates for the posture dynamical state space model.

$x_p = \Phi^{-1}(z)$: Nonlinear inverse change of coordinates.

$x'_p = \dot{x}$: Velocity of the point P along x axis.

$y'_p = \dot{y}$: Velocity of the point P along y axis.

(x_r, y_r): Robot reference point for the simulation results.

(x_{r_1}, y_{r_1}): Coordinates of the single-robot reference point with respect to the robot frame of reference.

(x_{r_2}, y_{r_2}): Coordinates of the robot-trolley reference point with respect to the virtual wheel reference frame.

x_{rel} : Smooth reference trajectory defined in local coordinates.

$x_T = [q \quad u]^T$: Generalized coordinates for the configuration dynamical state space model.

$\dot{x}$: Linear velocity of the robot reference point P along the x axis.

y : Position of the point P along the y axis.

$y = h(q_p)$: Position of the robot reference point or output equations.

$\dot{y}$: Linear velocity of the robot reference point P along the y axis.

$\dot{y}_1$: Linear velocity of the robot along the y_1 axis.

$z = \Phi(x_p)$: Nonlinear change of coordinates or new coordinates.

$z_e = \tilde{z}_1$: Feedback error.

$z_1 = y = h(q_p)$: Position of the robot reference point or output equations.

z_{1ref} : Reference point or trajectory.

$\tilde{z}_1 = z_1 - z_{1ref}$: Feedback error.

z_i : $i-th$ new coordinates.

$\dot{z}_i$: Derivate of the $i-th$ new coordinate.

$\bar{z}_d$: Normalized unit vector.

$\bar{z}_{oi}$: Normalized unit vector.

$\xi = [x \quad y \quad \theta]^T$: Posture coordinates.

$\dot{\xi} = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$: Posture velocity vector.

$\dot{\xi}_1 = [\dot{x}_1 \quad \dot{y}_1 \quad \dot{\theta}]^T$: Posture velocity vector expressed at the robot frame.

TABLE OF CONTENTS

APENDICES

LIST OF TABLES

LIST OF FIGURES

I Introduction

I.1 Overview

In most mail distribution centres, mail trolley transport systems are handled by hand. The need for the development of a semiautonomous wheeled mobile robot for moving mail trolleys by pulling and pushing them, to automate the process without making significant restructuring of the current plan layout, is presented in this chapter. The general definitions, background information and motivation for the research are also presented. The contents of the chapter are arranged as follows:

I.2 Background for the research

An autonomous robot may be useful for many application areas such as hostile environments, workshop services, cleaning tasks, forest, agriculture and lumbering, medical services, client support, construction and demolishing, space, military, safety, civil transportation, elderly and handicapped, entertainment, and so on. Different examples of mobile robots such as walking, biped, autonomous welding, plant maintenance, façade-cleaning can be found in Rooks (2002). Examples in service areas such as aircraft cleaning, underwater robots, vacuum cleaning, surgical robots, harvest robots, etc., can be found in Carlisle (2000). More about modern applications of the automatic guided vehicles (AGVs) in the area of engine assembly, heavy loads, wood board transportation, automated pallet stacker, and so on in Rooks (2001). A brief history of mobile robots with examples such as commercial platforms, LEGO Mindstorms, non-land robots, RoboCup, and real applications can be found in Lima & Ribeiro (2002).

According to the Robotics Industries Association (RIA):

> "A robot is a reprogrammable, multi-functional, manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks" [cited in Groover (1987) and Arkin (1998)].

other definitions that include intelligent behaviour:

> "Robot is a word derived from the Czech word for work and denotes a machine built to resemble a human in behaviour, intelligence and sometimes in appearance" [Krippendorff (1986)].

> "An intelligent robot is a machine able to extract information from its environment and use knowledge about its world to move safety in a meaningful and purposive manner" [Arkin (1998)].

> "An autonomous robot is an active artificial agent, which environment is the physical world, where its own decisions are taken based on the feedback provided by its respective physical sensors" [Russell & Norvig (1995)].

The above-mentioned definitions are related to a machine with specific characteristics. The first one is a factory oriented definition that puts its attention in moving objects to perform a pre-programmed variety of tasks, it is an engineering design point of view definition (in the sense that the system should satisfy a necessity to be useful [Gonzalez-Villela & Dorador-Gonzalez (2003b)]). The second one is a cybernetic oriented definition focused on reproducing the behaviour and intelligence of the human being (" ... the study of the mathematics of machines ... in terms of all the possible behaviours that an individual machine can produce" according to the definition of cybernetics given by Brooks (1991)). The third one is a definition oriented to the behaviour-based robotics ("the new style of the artificial intelligence discipline" [Brooks (1991)]), and the final one is an artificial intelligence oriented definition ("An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators" [Russell & Norvig (1995)]).

In the spirit of the mechatronics design discipline – that is a diplomatic discipline able to take a broad view of negotiations between disciplines, which chooses the synergistic combination of different areas of knowledge to design intelligent machines and processes [Gonzalez-Villela et al. (2003a)] (se publication 1 in appendix XXVII) – , all four above-mentioned definitions can be taken to fulfil a complete approach. That is to say, the system should satisfy the necessity of being useful. The modelling, control and motion planning of the robot is full of a multitude of rigorous mathematical formalisms. In addition, the robot should be able to produce a variety of behaviours, and should be endowed with some intelligent connection of perception to action.

Experience shows that successful robots are focused on performing a particular function (e.g. machines for replacing human operators). But, the applications of the results of the research activity seem to be far from practical applications. For instance, for many years the artificial intelligence community has built control systems that present intelligent behaviour, but that systems are not usually implemented in the real world, only in some simulated or specially designed environments [see Orebäck & Christensen (2003)]. By contrast, traditional real-time control research has designed robust control systems acting in the real world but usually without truly autonomous or intelligent behaviour [see Laumond (1998)].

Research on motion planning and control of mobile robots has received much attention in the past two decades. Autonomous robot navigation requires the solving of several problems [see Salichs & Moreno (2000)]. Generally speaking, these problems are related to the development of both the mobile robot architecture and the robot platform. The architecture of the mobile robot endows the robot with the ability to perform tasks described in a high level language by decomposing them to simple tasks or control actions [Brooks (1990a)]. The platform endows the robot with the power capacity, external and internal sensing, and mobility to get around in the environment. But, the results of the research in mobile robots are still far from realised in industrial applications, see, e.g., Jacoff et al. (2002).

Research on autonomous mobile robots can be divided into two major categories: the first one is oriented towards outdoor automobiles and terrain vehicles [e.g., Bentivegna et al. (1998), Rollins et al. (1998), Huntsberger (2001)], the other is oriented towards indoor wheeled mobile robots [e.g., Arkin (1988), Buhmann et al. (1995), Simmons et al. (2000), Xu et al. (2003)]. The consequence of this division is observed in the design of the vehicles, and the operational and environmental conditions. For instance, in the mechanical steering system, suspension system, the inclination of the wheels respect to the ground, the operational speed, the coupling of the motors to the wheels, the architecture for known or unknown environments, etc.

However, the industrial use of autonomous mobile robots has been difficult to apply because of the research activity in the area of autonomous mobile robot behaviour is still beginning:

> "Because research on quantitative description of mobile robot behaviour is still in its infancy, mobile robotics to date is still an empirical discipline that uses existence proofs extensively. ... The first step towards a science of mobile robotics, therefore, would be the development of quantitative, rather than qualitative descriptions of mobile robots behaviour" [Nehmzow (2001)].

The problem of developing an autonomous mobile robot can be described as follows [Salichs & Moreno (2000)]:

> An autonomous mobile robot must be capable of moving fast enough while avoiding fixed and moving obstacles (motion control problem), be aware of its

location (location problem), and collect knowledge to the environment (world modelling problem), optimizing its movements (planning problem), all should be integrated in one functional system (architecture problem), and it has to pass the following test: "The robot is placed in an environment that is unknown, large, complex, and dynamic. After a time needed by the robot to explore the environment, the robot must be able to go to any selected place, trying to minimize a cost function (e.g. time, energy, etc)."

A semiautonomous mobile robot is a robot that is capable of interacting and performing complicated tasks in both known and structured environments, like factories. More precisely, in this work:

> A semiautonomous mobile robot is a robot system that is semidependent on the planner, in the sense of given a task connected to a master plan (described in a high level language), the planner optimizes the robot's movements (as a consequence of having prior partial knowledge of the global and local environments) by decomposing the task into simple tasks (control actions) to be performed in autonomous way by the robot in known structured environments with dynamic changes.
>
> When a known environment experiences only minor dynamic changes due to the presence of static or mobile obstacles because of human, robot, and/or vehicle activity, this environment will be called "**Loosely Structured Environment.**"
>
> In addition, when the mobile robot is built based on wheels, it is called "Wheeled Mobile Robot" (WMR).
>
> The words "robot", "mobile robot", and "WMR" will be used in the same sense to express "wheeled mobile robot", other meanings of "robot" and "mobile robot" will be made explicit, e.g., legged mobile robot.

Semiautonomous mobile robots have been used as semiautonomous guided vehicles (AGVs) or self-guided vehicles (SGVs) in manufacturing systems with high requirements of manoeuvrability [Groover (1987)]. In this type of indoor environments the use of forks, tows, and unit loads have been the usual way of moving loads between stations [see AGV-Products (2004)].

I.2.1 Automation

Automation plays an increasingly important role in the economy and in the quality of life for all citizens; it is directed toward the creation of techniques to improve productivity and reliability in our society [Goldberg et al. (2002)]. Automation involves, not only automated devices, but also the mathematical, organizational and systems integration tools that allow a complex system to function efficiently, and involves the creation, design, and organization of entire systems that include the environment. Automation is fully explained in Groover (1987) and has been divided into the following classifications:

Automation and control technologies, such as: numerical control, industrial robotics, and programmable logic controllers.

Materials handling technology, such as: equipment for transporting, storing, and automatically identifying materials for material control purposes.

Integration of automation and material handling technologies into manufacturing systems, such as: production lines, assembly systems, group technology, and flexible manufacturing systems.

Quality control, such as: statistical process control, inspection issues, machine vision and coordinate measuring machines.

Manufacturing support functions, such as: design, process planning, concurrent engineering and design for manufacturing, production planning and control, material requirements planning, manufacturing resource planning, and just-in-time production systems.

Research in Automation also includes topics at the enterprise/operational level such as system modelling, analysis, performance evaluation, production planning, scheduling, coordination, risk management, and supply chain management. Automation raises fundamental and scientific problems that require expertise in areas such as algorithms, computational geometry, control theory, operations research, and stochastic modelling.

The fields of robotics and automation, although related, are distinct. The term "**Robot**", from the Czech "robota" (worker) was coined in 1923, but the term "automata" dates back to the 17th century (it was used by Descartes), and the term "**Automation**" was first

used by Ford Motor Co. in 1948. Both terms are used to characterize systems that can operate with minimal human supervision, but "Robotics" often connotes anthropomorphic machines, while "Automation" often connotes manufacturing and system operations [Goldberg et al. (2002)].

I.2.1.a Materials handling technology

A brief classification of the material handling equipment can be found in Groover (1987). Material handling is defined by the Material Handling Industry of America [MHIA (2005)] as "the movement, storage, protection and control of materials throughout manufacturing and distribution processes including their consumption and disposal." Some characteristics that should fulfil the material handling are: safety, efficiently, low cost, in a timely manner, accurately (the right materials in the right quantities to the right locations), and without damage to the materials. The material handling equipment can be divided into four categories:

- Transport equipment.
- Storage systems.
- Initialising equipment.
- Identification and tracking systems.

The transport equipment is used to move materials inside the factory, warehouse, or other facility, it is divided in five categories: industrial trucks, Automated guided vehicles, Monorails, Conveyors, Cranes and hoists (see Figure I.1) [Groover (1987)].

The automatic guided vehicles systems (AGVS) have been classified as: towing vehicles, unit load vehicles, pallet trucks, fork trucks, light loads AGVS, and AGVS assembly line vehicles [Koff & Demag (1985)], Figure I.2 shows some examples [Groover (1987)].

Some industrial application examples of forks (e.g., Figure I.12) and a tows (e.g., Figure I.13) for transporting mail can be found in AGV-Products (2004), with the characteristics showed in the Table I.1. Also, Figure I.14 shows a puller and pusher assistant for wheeled loads [Dane Industries (2003)].

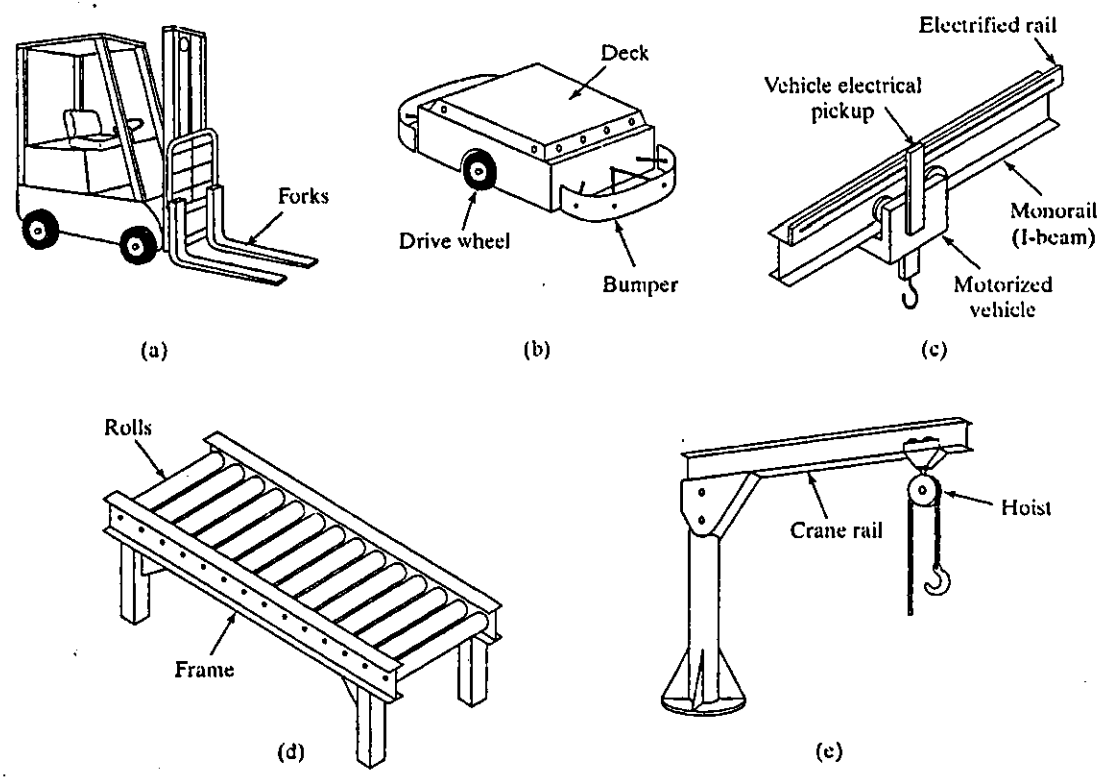


Figure I.1: Basic types of material handling equipment: (a) industrial truck, (b) AGV, (c) monorail, (d) roller conveyor, and (e) jib crane with hoist.

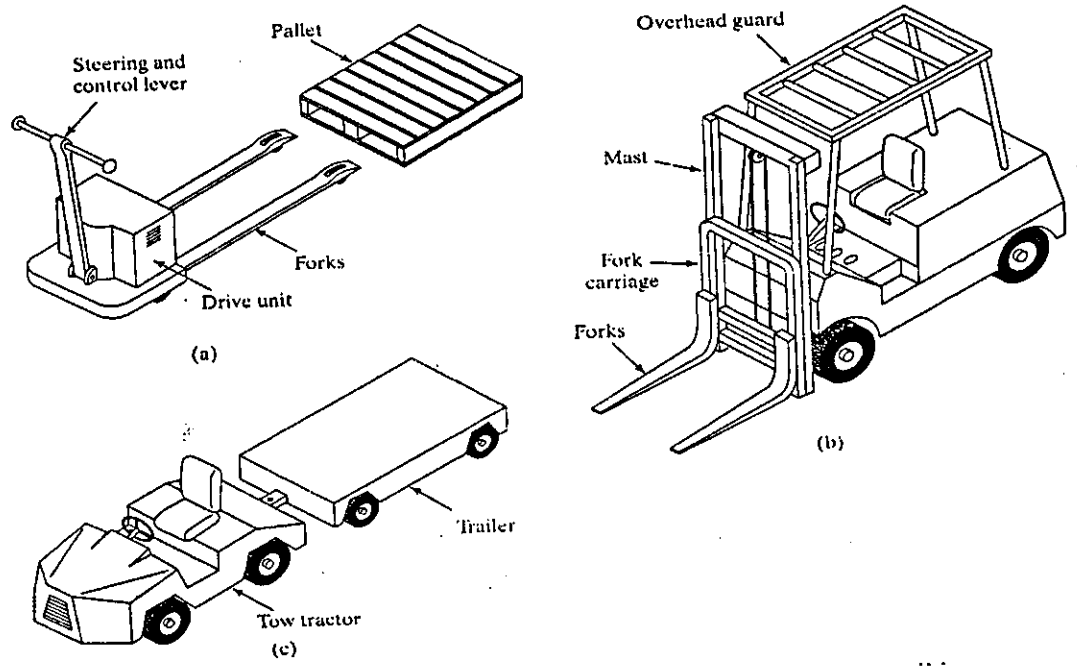


Figure I.2: Powered industrial trucks: (a) Pallet truck, (b) fork truck, and (c) towing vehicle.

Table I.1: Fork and tow characteristics for transport mail

Project Type	Automated fork Figure I.12	Automated tow Figure I.13
Number of Vehicles	3	3
Load Capacity	1,500 lbs	1,500 lbs
Vehicle Type	Laser Guided Fork Lift	Laser Guided Tugger

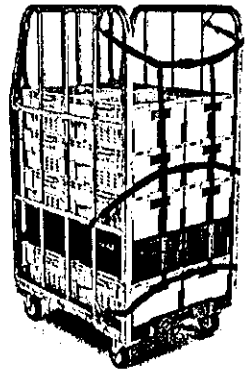


Figure I.3: York Roll Container [Envosort (2004)

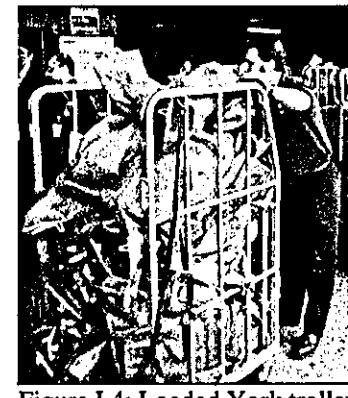


Figure I.4: Loaded York trolley

Figure I.5: York trolleys destined for major regional destinations as labelled on the green cards

Figure I.6: Outward mail awaiting shipment into Royal Mail vehicles

Figure I.7: Platforms to load and unload trains at PRDC

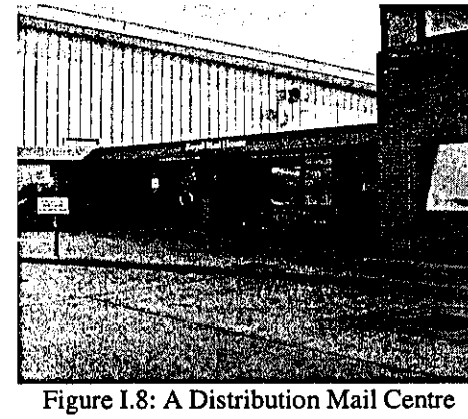


Figure I.8: A Distribution Mail Centre

Figure I.9: Manual handling for sorting mail

Figure I.10: Programmable automation technology for sorting mail

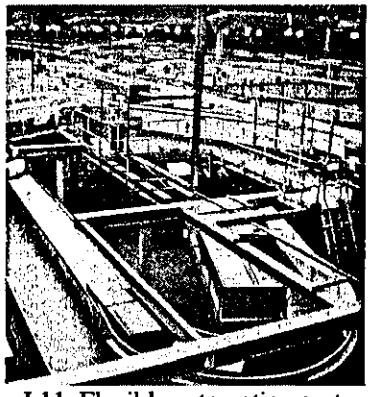


Figure I.11: Flexible automation systems for sorting parcels

Figure I.12: Fork

Figure I.13: Tow

Figure I.14: Puller and pusher assitant.

I.2.2 The need

Nowadays Royal Mail (RM) is more than ever before looking for improvements in the quality of its services in order to fulfil the national and international postal standards. Also, it needs to be a strong competitor when the new European rivals get access to the UK postal arena. Another requirement is that Royal Mail must be prepared to introduce new businesses. For this reason, the Royal Mail University Technology Centre (UTC), at the Mechatronics Research Centre, Loughborough University has been conducting research in order to solve real problems for them.

The identification of the need was made while touring various sites of Royal Mail referred to as 'the pipelines' [Gonzalez-Villela (2002), Zorriassatine et al. (2002a) and Zorriassatine et al. (2002b)]. So far, the transport of sorted mail in process is done using York trolleys (see Figure I.3) that are powered by hand (see Figure I.4). They are pushed and pulled to a specific position in order to wait their turn to process the mail they carry. The control of such trolleys is made by a label, which has the information of the final destination (see Figure I.5). The extensive use of this type of trolley to transport mail trays and mailbags between Mail Distribution Centres and mail sorting processes makes it difficult to redesign the process (see Figure I.6, Figure I.7, and Figure I.5).

I.2.2.a The Royal Mail Distribution Centres

Royal Mail Distribution Centres (e.g., Figure I.8), despite having their particular problems and way of working, have common characteristics and similar processes. The Royal Mail is a Service business that processes mail to deliver it to the correct addressed place or recipient. In order to do so, they use manual handling (e.g., Figure I.9), programmable automation technology (e.g., Figure I.10), and flexible automation systems (e.g., Figure I.11). The process of sorting mail can be classified into automated production systems and it goes from totally manual to fully automated. They handle mail using different processes, such as: moving, segregating, culling, cancellation and classing, addressing resolution, sorting, storing, delivering, and so on, with intermittent or continuous processes of variable mixtures of products, with low, medium or high processing rates, and with the flexibility to deal with mail variations. The technology that they are using now includes automatic machines for letter sorting, flat sorting, reading/coding, industrial robots for handling, and automatic material handling and storage systems for transport and storage mail [Zorriassatine et al. (2002b)].

The plant layout associated with the Mail Distribution Centres is of the type of Process Layout, according to the classification given by Groover (1987), when the departments or processing machine are arranged into groups according to a general type of process using in job shops (e.g. Figure I.9: discrete processing), batch production (e.g. Figure I.10: medium-sized lots of the same item of mail), and in flow type mass processing (e.g. Figure I.11: the mail is processed by flowing through the sequence of operations by material handling devices, e.g., conveyors, moving belts, transfer devices, etc.).

I.2.2.b The need of new materials handling equipment

Automated guided vehicles (AGVs) or self guided vehicles (SGVs) are battery-powered, automatically steered, and follow defined paths. They are used to move loads between load and unload stations in the facility and there can be some of them that interface with other systems to achieve the full benefits of integrated automation [Groover (1987)]. Thus far, the architecture and platforms of these types of equipments have been limited mainly to tows, forklifts, and unit loads [AGV-Products (2004)].

One strategy found to improve the efficiency of the Royal Mail Distribution Centres is to improve the material handling and storage systems by including automatic guided vehicles (AGVs) as a suitable means of automating the material handling system [Gonzalez-Villela (2002) and Zorriassatine et al. (2002a)], in order to unify the process of mail transfer between batch process and job shops. However, the introduction of the automatic guided vehicles (AGVs) technology generally requires significant restructuring of the workplace in order to be useful, so the concept of flexibility somewhat compromises this strategy [Arkin (1988)], and the considerable cost of restructuring the workplace is also a limitation.

In this work, in order to unify the process of the Mail Distribution Centres, the feasibility of developing a new semiautonomous wheeled mobile robot is studied, whose implementation into the process does not demand big changes in the plant layout and in the process itself, taking into account that the York trolley (see Figure I.3) plays a central role in the transportation of mail.

I.2.3 The task: push and pull

Postal automation started with mechanizing the facing, cancelling and sorting of each piece of mail about 40 years ago. Handling of trays and totes is very heavy, tiring work

that must be very accurate so mail is displaced to the correct location. Today, ninety percent of the hand-addressed mail can be "read" and stored automatically [Bloss (2002)].

Manual handling activities involve the combination of pulling, lifting, carrying, lowering and pushing. The risk of performing these combinations of tasks by a person have been studied in Straker et al. (1997). Pushing and pulling tasks have replaced lifting, carrying, and lowering activities in order to prevent low back disorders [Kingma et al. (2003)]. For instance, two-wheeled containers for refuse collecting [e.g., Laursen & Schibye (2002) and Kuijer et al. (2002)], and four-wheeled containers for mail transferring [e.g., van der Beek et al. (2000)].

One representation of a freebody diagram of a pushing subject for biomechanics analysis can be found in Hoozemans et al. (1998). The factors that affect the push or pull forces are associated with the intensity, frequency, duration, distance, handle height, cart weight, foot distance, hand force of the activity, and posture and anthropometry of the person [Hoozemans et al. (1998)]. The magnitude of the push and pull forces are also affected by the type of surface during initial and sustained phases [Laursen & Schibye (2002)]. Experimental results associated with the push and pull activities for four-wheeled loads in different ways can be found in the literature, where the direction of the highest and lowest exerted forces have been studied by Looze et al. (2000), and the three-dimensional space exerted forces: forward-backward, right-left, and upward-downward by van der Beek et al. (1999). The direction of exerted forces due to the gender differences in the pushing and pulling activity in handling four-wheeled cages under eight conditions, corresponding to the cage weight (130, 250, 400, 550 kg), in a laboratory test simulating the daily working practice of postal workers, can be found in van der Beek et al. (2000).

All the above mentioned research has been done only to determine the different disorders produced by pushing and pulling wheeled loads, which have been classified in [Hoozemans et al. (1998)]: epidemiology, psychophysics, physiology, and biomechanics. But, they all together present a complete research work of the forces required to push and pull mail trolleys.

On the other hand, among the different types of automatic guided vehicles AGVs used in industry for load transfer (e.g., manual load transfer, automatic couple and uncouple, power roller, belt, chain, power lift/lower, push and pull) the push and pull type is less popular. They are equipped with an automatic push or pull mechanism to pull or push the

load off the vehicles or trailers. This permits the vehicle to be extremely simple. This method is quite useful if the automatic load transfer is centralized. A good example of this approach would be a central storage area supporting manufacturing [Koff & Demag (1985)]. Some examples of assistant systems for handling the movement of wheeled loads in commercial and industrial settings using wheeled pusher or puller assistant can be found in Dane Industries (2003) (e.g., Figure I.14).

I.3 Motivation of the thesis

Most of the Royal Mail Distribution Centres cannot fully exploit the advantage of the flexible automation systems. Their programmable automation technology is decoupled because of the internal trolley transport system, which is the base of the mail transport system, is handled by hand. So, some parts of the process are totally dependent on human actions. Some of them already have a unified process of material handling. They use powered systems like: monorails, roller conveyors, pallet trucks, fork trucks, towing vehicles, and automatic guided vehicles (AGVs) inclusively (see section I.2), but those Royal Mail Distribution Centres were designed like this from the beginning.

Section I.2.2.b introduced the idea to unify the process of those Royal Mail Distribution Centres that have decoupled automation system. The possibility of implementing a transport system based on semiautonomous wheeled mobile robots for moving York trolleys (see Figure I.3), whose implementation into the process does not demand big changes in the plant layout of the current Royal Mail Distribution Centres and in the process itself, seems to be the solution to the problem.

The introduction of the existing automated guide vehicles (AGVs) technology to unify the process of transporting mail trolleys would require a big restructuring of the workspace [Arkin (1988)]. Moreover, the whole process would become less flexible than it now is, due to the size of the new equipment, the requirements of space to navigate and manoeuvre, and the complete change of plant layout makes this solution unfeasible due to the high cost of restructuring.

To avoid major restructuring in the workspace, other ways of transporting loads in an automatic way can be explored, for instance, moving trolleys with wheeled mobile robots by ***grasping without lifting*** (pushing and pulling while the load is held firmly).

Moreover, as it has been shown in the section I.2.1.a, no mobile robot that automatically pushes and pulls wheeled loads for this kind of work has been found. Only, a wheeled system for working as pusher or puller assistant was found in Dane Industries (2003).

By automating the internal trolley transport system, by having a new type of transport equipment, the Royal Mail Distribution Centres will be able to unify the whole process and be able to exploit all the advantages of automatization.

Also, this type of new material handling equipment can be useful in helping to move medium loads between workstations without interfering with the human activity in the manufacturing and distribution processes. Consequently smaller and flexible "wheeled mobile robots" can be developed for moving "wheeled loads" between work stations without making big changes in the configuration of the plants, but it is necessary to develop robot skills of recognizing, negotiating and mapping obstacles in their environment, and show robustness and reliability at the same time. Moreover, intelligence and reconfigurability capacities of such robots should be developed.

I.4 Summary

The convenience of developing a semiautonomous mobile robot for moving wheeled loads (mail trolleys) while the load is held firmly by the robot has been introduced. The main purpose is to couple the programmable automation technology of mail sorting at Royal Mail Distribution Centres by introducing this new technology, in order to exploit the advantage of the full automation. The implementation of the existing automatic guided vehicle (AGV) technology or other type of technology like monorails, conveyors, cranes and hoists has been discarded, because it would require significant restructuring of the workspace and, in consequence, huge expense. Some useful information about pushing and pulling wheeled loads by hand has been reviewed, in order to have a reference for forces exerted and health disorders produced in these tasks. Finally, the general definitions and the motivation of the thesis have also been presented.

II Literature review and outline of the research

II.1 Overview

Wheeled mobile robots for pushing and pulling wheeled loads when the load is held firmly by the robot to apply both stable pushes and pulls to the wheeled load have not been studied so far. Different aspects of developing that kind of semiautonomous wheeled mobile robot are presented in this chapter by reviewing the most relevant previous work. First, the problems of motion control and motion planning for this kind of robots were reviewed. Then, the robot models associated to the kinematics, dynamics, state space representation, static state feedback linearization, and architecture were examined, as well. Also, the aspects associated with the tasks of pushing and pulling, and the different types of mobile robot manipulators and theory related to them were studied too. In addition, the identification of the research approaches, how the current approaches are useful, what research approach is convenient, what is the aim, questions, scope and outline of the research, and the research work is also presented. The literature reviewed shows that the theory on wheeled mobile robots is interdisciplinary and that no wheeled mobile robot for this kind of activity was found. Also, it shows that a flexible platform and a motion planning approach is required to solve the problem. The convenience of using virtual wheels to control the robot movements when moving forwards and backwards, and coordinate the steering of the multiple steerable wheels is suggested. Also, grasping the wheeled load in two points in order to have a line contact seems to be the solution to have a motion planning approach without deep look ahead and without trapping configurations. The contents of the chapter are arranged as follows:

II.2 The problem of developing a semiautonomous mobile robot

It is well known that robot systems are interdisciplinary in nature. They can be analyzed from different viewpoints. Each discipline endows to the robot its particular characteristics. A choice of characteristics supported by each discipline produces a unique solution. Therefore, the literature review is presented from a thematic perspective. So, before starting with the review, the problem of developing a semiautonomous mobile robot by restating the definition seen in section I.2 is defined as follows:

> A semiautonomous mobile robot is a robot system that is semidependent on the motion planner, in the sense of given a task connected to a master plan (*described in a high level language*). The planner, as a consequence of having prior partial knowledge of the global and local environments, optimizes the robot's movements based on a cost function, e.g., time, energy, etc. (*motion planning problem*) by decomposing the task in simple tasks (*control actions*), to be performed in autonomous way by the robot, in partially known structured environments with dynamic changes. The robot must be capable of moving fast enough while avoiding fixed and moving obstacles (*motion control problem*), collecting knowledge from the local environment (*local world modelling problem*), and must be aware of its location (*location problem*). All should be integrated in a functional system (*physical platform and architecture problem*). Finally, it has to pass the following test:
>
> > The robot is placed in an environment that is partially known, large, complex, and dynamic. After given the location of the robot, the robot must be able to follow a task, given by the planner, in an autonomous way.

II.2.1 Motion control problems of wheeled mobile robots

In motion control of wheeled mobile robots four basic problems have been defined, classified into: path planning, point-to-point stabilization, path following, and trajectory tracking [e.g., Latombe (1991), Canudas de Wit et al. (1993), d'Andréa-Novel et al.

(1995) and Fierro & Lewis (1997), Laumond (1998), and Iannitti (2002)]. Starting with the following control problem definitions:

P0. Path planning: Find a path that joints the initial configuration q^0 with the final desired configuration q^d.

P1. Point-to-point stabilization: The robot must reach a desired goal configuration q^d starting from a given initial configuration q^0.

P2. Path following: The robot must reach and follow a geometric path in the Cartesian space starting from a given initial configuration q^0 (on or off the path).

P3. Trajectory tracking: The robot must reach and follow a trajectory in the Cartesian space (i.e. a geometric path with an associated timing law) starting from a given initial configuration q^0 (on or off the trajectory).

If it is not possible to solve the path-planning ***P0*** problem, the corresponding point-to-point ***P1***, path following ***P2***, and trajectory tracking ***P3*** problems clearly become meaningless. However, if the problem ***P0*** is solvable but it has not been computed, in this case, it may still be possible to solve ***P1***. The study of *controllability* (path planning ***P0*** problem) and *stabilizability* (feedback control ***P1***, ***P2***, and ***P3*** problem) for a given system must clarify this situation [Iannitti (2002)].

II.2.1.a Controllability of wheeled mobile robots

Controllability refers to the existence of the open loop controls (input signals) able to steer the state from any initial state q^0 to any final one q^d over a given finite time interval [Kolmanovsky & McClamroch (1995)]. Since wheeled mobile robots are subject to nonholonomic constraints this analysis should be done using nonholonomic theory.

The nonholonomic constraints of wheeled mobile robots are expressed in the tangent space of the configuration space, which is obtained from the kinematic model of the robot [e.g Yamamoto & Yun (1993) and Campion et al. (1996)], in contrast with underactuated manipulators where the representation of nonholonomic constraints are expressed in the dynamic model [e.g., Reyhanoglu et al. (1999) and Karkoub & Zribi (2002)]. The tangent space of wheeled mobile robots not only contains nonholonomic constraints but also

contains holonomic constraints [e.g., Yun & Sarkar (1998)]. In order to know if these constraints are nonholonomic is needed to answer the question given by Frobenius theorem: the equations are integrable if, and only if, the distribution Δ, associated to the tangent space, is closed under the Lie bracket operation, i.e., Δ is an involutive distribution [Isidori (1995)]. Then, if Δ is a no involutive distribution (Δ is not closed under the Lie bracket operation), then one or more constraints are not integrable, therefore they are called nonholonomic constraints.

In the presence of nonholonomic constraints it is possible to a find a distribution called the control Lie algebra, denoted $CLA(\Delta)$ [Laumond & Risler (1996)] (also called involutive closure $\overline{\Delta}$ [Nijmeijer & van der Schaft (1990) and Isidori (1995)]), which is the smallest distribution that contains Δ and is closed under the Lie bracket operation. The computation of the $CLA(\Delta)$ may be organized as an iterative procedure, e.g., see De Luca & Oriolo (1995), more efficient strategies exist, e.g., see Laumond & Risler (1996). If the rank of the control Lie algebra is full at a given configuration q, then there exists a neighbourhood U of q whose points represent configurations reachable by the system moving from q along an admissible path. Moreover, when the system is symmetric without drift, this path stays in U [Laumond & Risler (1996)]. This condition is known as Chow's theorem or the "***accessibility rank condition***" [De Luca & Oriolo (1995) and Altafini (2001)], which is a local condition. If the rank condition holds everywhere in the configuration space, the robot is termed as controllable. Proving the controllability of an n-dimensional system using the controllability rank condition involves showing that, for any point q in the manifold, there exists a family of n vectors fields in the Control Lie Algebra whose values at q span R^n [Laumond & Risler (1996)]. Other controllability definitions are given by Sussmann (1983), Sussmann (1987), Nijmeijer & van der Schaft (1990), and Lynch & Mason (1996).

On the other hand, the controllability of conventional wheeled mobile robots has several cases. The posture kinematic model of omnidirectional wheeled mobile robots does not have any nonholonomic constraints [d'Andréa-Novel et al. (1995)], it has full rank and satisfies the *Kalman controllability rank condition* for linear systems, so, it is completely controllable. For restricted mobility robots, which have nonholonomic constraints, the control Lie algebra satisfies the *accessibility rank condition* [Campion et al. (1996)], which implies controllability for symmetric driftless systems [Isidori (1995)]. This

property implies that a mobile robot can be always driven from any initial posture q^0 to any final one q^d, in a finite time, by manipulating the velocity control inputs. If a driftless system is controllable its corresponding extended dynamic model is controllable as well, a constructive proof can be found in Sussmann (1991). For further explanation on nonholonomic motion planning methodologies the reader is referred to Latombe (1991), Li & Canny (1993), Zheng (1993), Murray et al. (1994), Kolmanovsky & McClamroch (1995), and Laumond (1998).

Once proved that a wheeled mobile robot is controllable it is possible to plan the motion of the robot through the environment. The process of finding feasible paths or feasible trajectories for the robot in the environment is called *the motion planning problem*, which will be reviewed in section II.2.2.

II.2.1.b Stabilizability of wheeled mobile robots

Stabilizability refers to the existence of closed loop controls (feedback laws), which guarantee that an equilibrium of the closed loop system is asymptotically stable [Kolmanovsky & McClamroch (1995)]. The meaning of *stability* has been divided in three [e.g., Fantoni & Lozano (2001) and Khalil (2002)]: stability, asymptotic stability, and exponential stability. Stability means that the system behaves *uniformly stable* staying near to the equilibrium point. Asymptotic stability means that the system behaves *uniformly asymptotically stable* (globally or locally) tending to the equilibrium point. Exponential stability means that the system behaves *exponentially stable* (globally or locally) tending exponentially to the equilibrium point [Becerra (2004)].

Nonholonomic systems cannot be stabilized by a smooth (or even continuous) time-invariant static state feedback due to the necessary condition for feedback stabilization [Brockett (1983)] that makes a specified equilibrium of the closed loop locally asymptotically stable [Bloch et al. (1992)]. Therefore, different techniques have been used to (locally) asymptotically stabilize the robot about an equilibrium point. Some techniques have been cited by Kolmanovsky & McClamroch (1995):

a) Discontinuous time-invariant stabilization: piecewise continuous controllers [e.g., Canudas de Wit & Sordalen (1992)] and sliding mode controllers [e.g., Bloch & Drakunov (1994)].

b) Time-varying stabilization: time-periodic feedback laws [e.g., Coron (1992), Pomet (1992), Samson (1993) and M'Closkey (1998)].

c) Hybrid feedback laws: combining continuous-time features with either discrete-event features or discrete-time features [e.g Bloch et al. (1992), Young et al. (2000), Ji et al. (2003) and Shim & Sung (2003)], where the operation of hybrid controllers is based on switching at discrete-event features or discrete-time features between various low-level continuous time controllers.

Nonholonomic wheeled mobile robots are controllable (see section II.2.1.a), their equilibrium point at the origin can be made stable but cannot be asymptotically stable by a smooth state feedback. They are not input-state linearizable by state feedback because they do not satisfy the involutive condition, but they can be input-output feedback linearizable if a proper set of output equations is chosen [Yamamoto & Yun (1993)]. As a result, they can be partially stabilizable by smooth state feedback if the decoupling matrix has full rank [Isidori (1995)].

The input-output feedback linearizable property of the wheeled mobile robots, differentially flat system [Fliess et al. (1995)], has been used to control the position of a point defined somewhere on the robot's platform in order to solve the path following problem [e.g., Sarkar et al. (1994)], and trajectory tracking problem [e.g., Yamamoto & Yun (1993) and d'Andréa-Novel et al. (1995)]. However, one drawback of this technique is that despite the robot's zero dynamics being stable, it is not asymptotically stable. In addition, the internal dynamics of the system exhibits unstable behaviour when moving backwards [Yun & Yamamoto (1997) and Bolzern et al. (2001)], so, this behaviour should be taken into account for the motion planning problem.

Restricted mobility robots can be fully linearized by dynamic state feedback [Campion et al. (1996)], as long as some part of the input vector remains bounded away from zero [e.g., d'Andréa-Novel et al. (1995)]. Similar work can be found in Oriolo et al. (2002a) where trajectory tracking and posture stabilization problem are simultaneously solved using dynamic state feedback, as well.

On the other hand, the posture tracking control for kinematic models using a reference robot has been proposed in Kanayama et al. (1990), which presupposes the perfect velocity tracking of the system. Fierro & Lewis (1997) solved the problem for dynamic

control using the backstepping approach, by converting the input velocity commands, which were proposed by Kanayama et al. (1990), into motor torque commands.

II.2.2 The motion planning problem

Steering a control system from an initial state to a final state over a given finite time interval by applying open loop controls is called the motion-planning problem (see section II.2.1.a). That is to say, the study of open loop controls that transfer the system from a specified initial state space to a specified final state [Kolmanovsky & McClamroch (1995)]. So, the motion planning problem is also called the controllability problem [Agrachev (2002)]. Motion planning is in charge of the study of the methods for computing motion strategies (e.g., geometric paths, time-parameterized trajectories, sequence of sensor-based motion commands) to achieve high-level goals (e.g., go to A without colliding with obstacles, assemble product P, build map of environment E, find object O) [Latombe (2003)]. The term motion planning is used for problems involving time, dynamic constraints, object coordination, sensory interaction, and so on [Latombe (1999)], e.g., the dynamical path planning where the path planning and the dynamics of the robot is considered together [Choi & Lee (1996)].

As a consequence of the introduction of the robot's configuration space [Lozano-Perez (1983)], which represents the robot's configuration (position and orientation) as a point in the configuration space, the path planning was reduced to the problem of planning a path for a point [Latombe (1999)]. As a result, multiresolution path planning techniques and searching methods were introduced [e.g., Kambhampati & Davis (1986)]. For a summary of searching and exploring methods see e.g., Russell & Norvig (1995).

The most studied methods in motion planning are the methods to avoid collision, due to the direct practical application in solving problems of finding the free collision path in robot motion and avoid collision with unexpected obstacles, they can be classified in two types [Latombe (1991) and Cameron (1994)]: 1) Path planning techniques, which are global deliberative techniques, and 2) obstacle avoidance techniques, which are local reactive techniques.

In mobile robotics these techniques use *global models* to model the workspace in two dimensions for the ***path planning problem*** [e.g, Kambhampati & Davis (1986), Esquivel & Chiang (2002)] and *local models* for ***goal reaching and obstacle avoidances*** [e.g.,

Khatib (1986), Khosla & Volpe (1988) and Wang & Chirikjian (2000)], or both global and local methods [e.g., Buhmann et al. (1995), Chatila (1995), Arkin & Balch (1997), Maravall et al. (2000) and Low et al. (2002)].

II.2.2.a The path planning problem

The term path planning refers to the purely geometric problem of computing a collision-free path (geometrical path) for a robot among static obstacles [e.g., Laumond (1998), Latombe (1999) and Salichs & Moreno (2000)]. This collision-free path is built for rigid or articulated objects among static obstacles, taking into account the constraints (geometric, kinematic, physical, and temporal constraints) by decomposing the path in continuous sequence of collision-free robot configurations to connect the robot from an initial configuration to a final goal configuration [Latombe (2003)]. Thus, for global methods the available standard techniques are [Latombe (1991)]: the Roadmap techniques (Visibility Graphs, Varonoi Diagrams, Freeway, and Silhouette), and Cell Decomposition techniques (Exact Cell Decomposition, and Approximate Cell Decomposition). When this free path (geometrical path) and the configuration of the robot are time-parameterized they produce trajectories and control actions to be followed by the robot (physical navigation control actions). In addition to the aspects of path planning and trajectory generation, in certain situations, the motion planning techniques take into the account more advanced planning aspects: robot cooperation, manoeuvres, push and pull, and uncertainty of available information [Latombe (1991)]. For more explanation on motion planning methodologies the reader is referred to see Latombe (1991), Hwang & Ahuja (1992), Cameron (1994) and Laumond (1998), and an up-to-date course can be found in Latombe (2003).

In path planning for nonholonomic robots (e.g., carlike robots, multibody tractor-trailer robots, etc.), the presence of nonholonomic kinematic constraints (which are not integrable) leads to the dependence of robot's degrees freedom (e.g., to rotate around its axis must change its position), where the traditional path planning algorithms for robotic manipulator are not applicable [Campion et al. (1996)]. Thus, a path in the free configuration space does not necessarily correspond to a feasible one, so the concepts from the areas of Differential Geometry Control Theory are used to study the controllability of the robot (see section II.2.1.a) to produce effective planning techniques [Laumond (1998)]. So, the planning problem for locally controllable nonholonomic

robots is reduced to find a feasible free path, between two given configurations, to be decomposed into a finite number of nonholonomic subpaths [Halperin et al. (2004)]. In addition, when a number of movable objects are required to be displaced in order to reach a goal configuration for a given process, "manipulation planning" is required, where one must plan the motions of the movable objects, the robots, and the robot grasping the movable object [Latombe (1999)]. The problem of rearrange movable objects has been presented in Ben-Shahar & Rivlin (1995), Rus et al. (1995), Lynch & Mason (1996), Ben-Shahar & Rivlin (1998a) and Ben-Shahar & Rivlin (1998b). A brief summary and useful definitions in the area of nonholonomic planning and manipulation planning can be found in Halperin et al. (2004).

II.2.2.b Goal reaching and obstacle avoidance: Local navigation

The capability of deciding or providing robot motion actions in automatic way in order to achieve goals in environments with spatial arrangements of physical objects is a motion planning task as well [Halperin et al. (2004)].

Navigating in known indoor environments is simplified to navigating in designed predefined routes. The robot follows the path suggested by the path planner and reacts to avoid unexpected obstacles while reaching a goal. In unknown environments, the path planning stage has no sense, so the navigation strategy is purely reactive [e.g. Chatila (1995), Maaref & Barret (2002), Victorino et al. (2003a) and Victorino et al. (2003b)]. If there are no obstacles between the robot and its goal, the reference path is just a straight line between them. If an unpredicted obstacle is detected, some avoidance strategy is used [Gomez Ortega & Camacho (1996)]. In structured environments, where the uncertainty level is low, this problem is reduced to control the trajectory of the robot on a desired precalculated reference path. For locally controllable nonholonomic robots, the planning problem is reduced to find a feasible free path, between two given configurations, to be decomposed into a finite number of nonholonomic subpaths [Halperin et al. (2004)]. If there is the possibility of finding unexpected obstacles in the trajectory of the robot, the problem is much more complex, involving sensor data processing and obstacle avoidance strategies [Gomez Ortega & Camacho (1996)]. Some different ways of dealing with the problem have been suggested by using local path planning, potential field approach, or behaviour based techniques [e.g., Khatib (1986), Khosla & Volpe (1988), Arkin (1988), Borenstein et al. (1995), Simmons et al. (1997), Choi & Lee (1996), Arkin (1998),

Berman & Edan (2002), Ulrich & Borenstein (2000), Maaref & Barret (2002), Felner et al. (2002), Althaus & Christensen (2003), Victorino et al. (2003a) and Victorino et al. (2003b)].

II.2.2.b.1 *Potential field approach*

For locally controllable nonholonomic robots the automatic search to find the nonholonomic subpaths that adjust to the existing feasible free path can be guided by a potential field function, a function over the free space that has a global minimum at the goal configuration [Halperin et al. (2004)]. This function may be constructed as the sum of an attractive and a repulsive potential field [e.g., Khatib (1986) and Zhao & BeMent (1992)].

Obstacle Avoidance techniques compute the reaction actions to the unforeseen changes in the local environment. For its simplicity the most used have been the potential field techniques. The attractive potential has a single minimum at the goal and grows to infinity as the distance to the goal increases. The repulsive field is null at all configurations where the distance between the robot and the obstacles is greater than some predefined value, and grows to infinity as the robot gets closer to an obstacle. Evaluating the repulsive potential requires an efficient distance computation algorithm and is necessary to deal with local minima problem [Halperin et al. (2004)]. These potential field methods couple environment sensing feedback with the lowest level control for real-time operation, they were first introduced by Khatib (1986), Krogh (1984), and Krogh & Thorpe (1986). These methods have been distinguished for the form in which they solve the problem of local minima by defining the potential function or by designing the search algorithm. The first one is related to *specify a potential function without local minima points*, improving the "local dynamical behaviour" [e.g., Rimon & Koditschek (1992)] or by reducing the number of local minimums or the size of the attractive points [e.g. using harmonic functions: Kim & Khosla (1992), Rosell & Iniguez (2002)], or using physical analogies [e.g. electrostatic: Valavanis et al. (2000), heat transfer: Wang & Chirikjian (2000)]. The second one, is related to *include the suitable potential functions* in the search algorithm to escape of the local minimums, e.g.: Virtual Force Field method [Borenstein & Koren (1989)] and its modifications [Borenstein & Koren (1991), Ulrich & Borenstein (1998) and Ulrich & Borenstein (2000)], using virtual obstacle concept [Chengqing et al. (2000)], fictitious charges [Maravall et al. (2000)].

They can be either pure Artificial Potential Field (APF) methods or Artificial Intelligent (AI) methods [Vadakkepat et al. (2000)]. Thus, the goal generates an attractive potential, the obstacle generates a repulsive potential, and the gradient of the resultant potential is treated as an artificial force applied to the robot frame. So, the navigation problem of directing the robot into a given goal location, while avoiding unforeseen static or moving obstacles can become a stabilization problem. Artificial potential field techniques are used for Local Navigation to move the robot through the path or between two cells, endowing to the robot with ability of avoiding obstacles. Some drawbacks or limitations of these methods have been reported by Koren & Borenstein (1991): cyclic behaviour due to the local minima, no passage between closely spaced obstacles, oscillations in the presence of obstacles, and oscillations in narrow passages.

II.2.3 Kinematics, dynamics, state space representation, and input-output state feedback linearization

The analysis of stability and controllability of wheeled mobile robots requires the robot modelling, as seen in section II.2.1. Two basic models are required: kinematic model and dynamic model [e.g., Campion et al. (1991)]. But, in order to find simpler design controllers or feedback stabilizers the original models are manipulated using codistributions, input feedbacks, change of coordinates, input-output linearization [e.g., Yamamoto & Yun (1993)], and/or canonical representations [Kolmanovsky & McClamroch (1995)]: chained form, power form, and Caplygin form.

The state feedback linearization and control of mobile robots can be found in d'Andréa-Novel et al. (1995) and Canudas de Wit et al. (1996), and the different aspects associated to the control and stabilization of wheeled mobile robots in Kolmanovsky & McClamroch (1995) and reference therein (see section II.2.1, as well).

The kinematics and dynamics of the wheeled mobile robots are typified by the state space representation of its configuration coordinates. The most important representations due to the different properties that can be obtained and for being the basis of the dynamic behaviour analysis of the system, are two: the kinematic state space representation (kinematics constraints), and the dynamic equations of motion [e.g., Campion et al. (1996)].

So, after finding a codistribution to the kinematic constraints, applying a series of transformations in the dynamic equation of motion, applying a partial input feedback linearization, and obtaining a partial input-output feedback linearization of the system given a set of output functions, it is possible to control an specific point related to the robot platform, which maps the inputs and outputs linearly, with a remaining nonlinear part having stable zero dynamics (not asymptotically stable), and unstable internal dynamics when moving backwards [e.g., Yamamoto & Yun (1993), Sarkar et al. (1994), d'Andréa-Novel et al. (1995), Fierro & Lewis (1997), Yun & Yamamoto (1997), and Bolzern et al. (2001)].

II.2.3.a Kinematics

Kinematics is the branch of physics and a subdivision of classical mechanics concerned with the geometrical possible motion of a body or system of bodies without consideration of the forces involved (i.e. causes and effects of the motions) [Britannica-Corporate-Site (2004)].

Kinematics modelling is one of the most important tools on robotics, it is used for modelling mechanisms, actuators, and sensors, is used for on-line control, and off-line programming, and simulation. In mobile robots specifically, you need kinematics models for steering (control and simulation), perception (image formation), sensor head and communication antenna pointing, world modelling (maps, object models), terrain following (control feed forward), and gain control of legged vehicles [Kelly (1996)].

Different kinematic models for wheeled mobile robots can be found in the literature [e.g., Zhao & BeMent (1992), Yamamoto & Yun (1993), Sarkar et al. (1994), De Luca & Oriolo (1995), Campion et al. (1996), Thuilot et al. (1996), Fierro & Lewis (1997), Laumond (1998)]. These are associated, not only with the dimensions of robot and type of wheels configurations, but also with the position of the robot frame reference system, the definition of the wheel frame reference system, and the selection of the steering system. One drawback on the kinematic models of wheeled mobile robots with restricted mobility is that they are subject to the nonholonomic kinematic constraints, which means that they cannot be integrated. As a consequence, coordinates cannot be eliminated by integrating the kinematic equations of constraints [Gonzalez-Villela et al. (2004)] (se publication 2 in appendix XXVII).

A complete work in kinematics modelling for wheeled mobile robots can be found in Muir & Neuman (1987) and Alexander & Maddocks (1989). The analysis of the kinematic properties of wheeled mobile robots built with conventional wheels (fixed wheels and castor wheels) and Swedish wheels can be found in Campion et al. (1996) and Canudas de Wit et al. (1996), where the wheeled mobile robots have been classified in five nondegenerated types, using the characteristics of mobility and steerability of the robot.

The kinematic equations of motion of wheeled mobile robots from a general point of view were firstly formulated by Muir & Neuman (1987). They decompose kinematic into two sections: internal and external kinematics. The internal kinematics describes the relation between the linkages within the robot. The External kinematic describes the relationship between the robot and the rest of the world [McKerrow (1991)]. Some differences in modelling wheeled mobile robots, manipulators, and legged robots are shown in Table II.1

Table II.1: Modelling differences between wheeled mobile robots, manipulators, and legged robots

	Wheeled mobile robot	Manipulator	Legged robot
Contact with the work space	Fixed multiple contacts.	Open and close contact.	Variable multiple contact.
Model	Fixed multiple closed-link chain, since it has more than one wheel in contact with the surface and it is travelling over.	Single open-link chain when in free space. A closed link chain when in contact with the workspace.	A chain for each leg, which opens and closes as the foot is lifted off the ground and placed back on the ground
Degrees of freedom of the end effector.	A wheel can both turn and translate. It is described as a higher order pair constrained as a point or line contact.	Constrained by the task.	Constrained by the task.
Actuating joints	Some wheels are not actuated at all and some degrees of freedom are not actuated on some wheels.	All joints are actuated and are used to control the motion of the end effector.	All joints are actuated and are used to control the motion of the end effector.
To controlling the trajectory of the end-effector	There is no need to measure the position, velocity and acceleration of each degree of freedom of each wheel.	The position, velocity and acceleration of each joint most be measure.	The position, velocity and acceleration of each joint most be measure.

The general framework proposed by Muir & Neuman (1987) to develop their kinematic models started with the definition of wheeled mobile robot as:

> "A robot capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel

assembly is a devise, which provides or allows relative motion between its mount and a surface on which it is intended to have a *single* point of *rolling* contact" [Muir & Neuman (1987)].

Muir & Neuman (1987) introduced six practical assumptions classified in design assumptions and operational assumptions:

Design assumptions:

1. The wheeled mobile robot does not contain flexible parts.
2. There is zero or one steering link per wheel.
3. All steering axes are perpendicular to the surface.

Operational assumptions:

1. The wheeled mobile robot moves on a planar surface.
2. The translational friction at the point of contact between the wheel and the surface is large enough so that no transactional slips may occur.
3. The rotational friction at the point of contact between the wheel and the surface is small enough so that rotational slip may occur.

These constraints limit the design of such robots and restrict the range of the practical applications to environments that satisfy these assumptions.

These assumptions were taken by Campion et al. (1996) in order to formulate the structural properties of the kinematic and dynamic models, considering the restrictions of the robot mobility introduced by the constraints. Also, he introduced the concepts of ***degrees of mobility***, ***degrees of steerability,*** and ***degrees of manoeuvrability***. He proved that independent of the variety of possible constructions and wheel configurations (with an arbitrary number of wheels of various types and various motorizations) the total non degenerated classes of mobile robots can be partitioned in five. Besides, he introduced two different kinds of models in the state space representation for the kinematic modelling: The ***posture kinematic model*** and the ***configuration kinematic model***. The ***posture kinematic model*** gives a general global description of the robot. Moreover, its properties are useful to understand the manoeuvrability properties and to verify the

irreducibility, controllability and stabilizability of the robot. The ***configuration kinematic model*** is useful for analysing the complete kinematic robot behaviour within the framework of the theory of nonholonomic systems.

Thus, the kinematics of wheeled mobile robots consists of a set of kinematic constraints imposed by the robot's wheels. It is the most important modelling of the wheeled mobile robots, because of the inherent properties of their kinematic constraints that define the robot's degree of mobility and steerability [e.g., Campion et al. (1996)].

The *kinematic model* of wheeled mobile robots has two types of kinematic constraints, holonomic and nonholonomic constraints [Yun & Sarkar (1998)]. The nonholonomic constraints cannot be integrated; therefore some generalized coordinates cannot be eliminated from the system description [Zhao & BeMent (1992)]. The presence of holonomic constraints is associated with the existence of the geometric constraints (*geometric relations*) between the generalized coordinates of the system. The presence of nonholonomic constraints is associated with the existence of the kinematic constraints (*kinematic relations*) between the generalized velocities of the system [Campion et al. (1991)]. The no integrability of the nonholonomic constrains can be interpreted as the no existence of a set of scalar functions that could be the independent solutions of the nonholonomic differential equations. On the other hand, one way to find the state space representation of the wheeled mobile robots can be obtained by finding a codistribution (without drift) of the kinematic constraints, as suggested by Isidori (1995). Although, the state space representation of nonholonomic mobile robots is a kind of underactuated system, which has fewer inputs than degrees of freedom, in some sense (from the physical point of view) they are fully actuated [Fantoni & Lozano (2001)], so the techniques used in underactuated manipulators are not necessarily applicable [Iannitti (2002)]. These fewer inputs can be or not be associated to the joint space variables, that is why these inputs are best called the steering system [Fierro & Lewis (1997)], auxiliary inputs [Campion et al. (1996)], mobility control vector [Bayle et al. (2003b)], or quasi-velocities [Neimark & Fufaev (1972)], which are used as control inputs to the kinematics state space model of the robot.

II.2.3.b Dynamics

Dynamics is the branch of physical science and subdivision of mechanics that is concerned with the effect of forces and torques on the motion of bodies having mass [Britannica-Corporate-Site (2004)].

The dynamics of the wheeled mobile robots is described by its equation of motion. The most used method to obtain it has been the Lagrange formalism [Zhao & BeMent (1992)]. Since, these systems are in contact with a surface, the dynamic equation of motion (expressed at the configuration space) include the generalized reaction forces of the actuators. A complete work of the dynamics of nonholonomic systems can be found in Neimark & Fufaev (1972). Examples of modelling the dynamics of the wheeled mobile robots can be found in Campion et al. (1991), Zhao & BeMent (1992), Yamamoto & Yun (1993), Sarkar et al. (1994), Campion et al. (1996), Fierro & Lewis (1997), Laumond (1998), and Tan et al. (2003).

The first dynamic equations of motion of the robot are generally obtained at the configuration space applying the Lagrange formalism, including the generalized reaction forces taken from the kinematic constraints, which is multiplied by the Lagrange multipliers. Later on, some transformations are done on them in order to obtain a second set of motion equations expressed at the steering space that are free of Lagrange coefficients, useful for computing the torques needed to produce accelerations in the steering system.

II.2.3.c State space representation

The complete state space representation of the wheeled mobile robot takes into account the *kinematic model* and the *dynamic equation of motion* at the steering space. As a consequence, a nonlinear system with drift is obtained, where the input variables represented by the steering system are included as new state space variables. Then, applying a smooth static time-invariant state feedback, from the dynamic equation of motion at the steering space, one obtains a partially input feedback linearized system. Thus, the complete nonlinear state space representation is reduced to a simple form, whose structure only depends on the kinematic constraints [Campion et al. (1991)]. The inputs to the new nonlinear system (with drift) are the derivatives of the steering variables, as a result, the dynamic control problem is reduced to a kinematic control

problem of the system [Fierro & Lewis (1997)]. This model is called *dynamical model* and can be used to obtain a partially linearized system in a new coordinates by applying an input-output static state feedback linearization.

II.2.3.d Input-output static state feedback linearization

The input-output static state feedback linearization can be described as the procedure to find the static state feedback control law and the changes of coordinates that transform the nonlinear system into an equivalent linear system, which map the inputs and outputs linearly [Khalil (2002)]. The system is input-output static state feedback linearizable and transformable (expressed in a new coordinates) into both a controllable linear subsystem and a remaining nonlinear subsystem that do not affect the behaviour of the system [Campion et al. (1991)], if a proper set of output equations is chosen [Sarkar et al. (1994)]. The necessary and sufficient condition for input-output static state feedback linearization is that the decoupling matrix has full rank [Isidori (1995)].

Given a proper output function, the input-output static state space representation for the posture dynamic model (with drift) of wheeled mobile robots have a globally defined relative degree of two for input-output static state feedback linearization proposes [d'Andréa-Novel et al. (1995)]. In this case, the first derivative of the output function does not depend on the input vector, but the second does. Then, it is possible to define a change of coordinates and an input state feedback that partially linearize the system. A set of linearizing output equations, remaining nonlinear coordinates, and regularity conditions for each type of wheeled mobile robot have been proposed by d'Andréa-Novel et al. (1995). As a result, the input and output variables are mapped linearly. The zero dynamics of the system are guaranteed but not asymptotically. Besides, an unstable behaviour is presented in the internal dynamics when moving backwards [Yun & Yamamoto (1997) and Bolzern et al. (2001)]. Therefore, a smooth linear feedback control law can be applied to the partially linearized variables to make the system partially asymptotically stabilizable o partially exponentially stabilizable, but in this case, the posture stabilization using smooth linear feedback remains unsolvable, in accordance with Brockett (1983).

II.2.4 Architecture point of view

Robotic architecture is the discipline devoted to the design of highly specific and individual robots from a collection of common software building blocks. Where the topology, interconnection, specific functionality of each component (perception, reasoning, and action), and the interface between the set of structural components, provides a way of organizing the control systems [Arkin (1998)]. An architecture is a sort of computing device with physical sensors and actuators, which is capable of supporting a program able to map perceptions to actions [Russell & Norvig (1995)]. A brief summary of some architectures can be found in the literature [e.g., Pettersson (1997), Kortenkamp (1998), Arkin (1998), and Low et al. (2002)].

At the beginning architectures were developed on plan-based actions, Sense-Model-Plan-Act, with deliberative models and using a top-down approach on a hierarchical way for static environments. Sensed information of the environment was used firstly to build maps, secondly to plan the trajectories and finally to generate the motor actions to allow the robot follows the trajectories, where the decisions were based only on the information of the maps. Robots using this architecture moved very slowly to avoid collision, but later on were proposed architectures that used the information from sensors directly instead of world models. The new vision was "The world is its own best model" [Brooks (1991)]. With this new vision were developed reactive models using a bottom-up approach with no hierarchical subordination based on sensor-based actions (Subsumption Architectures), they were first introduced by [Brooks (1985)]. Robots using this architecture moved fast to avoid collision. As a result of using both deliberative and reactive approaches, the hybrid models appeared. Planners were in charge of long term decisions and reactive behaviours were in charge of motion execution, creating the problem of balancing authority against freedom. Thus the three layers architectures appear with deliberative layer, sequencer layer (it is usually a controller of discrete events), and reactive layer [Gat (1998)]. Although hybrid architectures have developed different layers, the three-layered architectures offer the way to organize different methodologies (deliberative and reactive) in a systematic way. Some varieties of layered architecture are listed bellow.

Low et al. (2002) presented a two-layered architecture for commercial research robots (K-Team-Corporation (2002-2004)) that was evaluated in a commercial simulator software [Cyberbotics-Ltd (2003)], which combined the deliberative navigation module and the

reactive control behaviour module with a command fusion. Volpe et al. (2001) presented the CLARAty architecture, which is a two-layered architecture based on Object-oriented-functional-layer and Decision-layer, which performs the planning and execution by sharing a common database. Nicolescu & Mataric (2002) presented an architecture, which allows for the representation and execution of complex, sequential, hierarchical structured tasks, within a behaviour-based framework using abstract behaviours to encode complex hierarchical plan-like strategies. Orebäck & Christensen (2003) evaluated three commercial architectures and a proposed generic architecture based on object models. Nilsson & Johansson (1999) showed a typical architecture for an industrial robot manipulator that was built on three layers and could be subdivided in eight layers, where each layer had to be programmed by a different skilled programmer, showing that each layer was implemented with different knowledge. In general, in three layers architectures the upper layers (deliberative layers) are based on planners for long-term decisions, classic planners normally produce rigid orders (for a bird's eye of the deliberative problem in automated guided vehicles see e.g., Mantel & Landeweerd (1995), Hoff & Sarker (1998) and Berman & Edan (2002)). The lower layers (reactive layers) are based on reactive behaviours for motor execution [e.g., Arkin (1988) and Ulrich & Borenstein (2000)]. One example of the complexity of the design a control architecture can be found in Chatila (1995).

Autonomous mobile robots architectures have been defined as behaviour compilers [Brooks (1990a)] that should be programmed. The language for programming behaviours has been call "The Behaviour Language", which was first introduced by Brooks (1990b). In fact, reactive behaviour is a standard for motion control of mobile robots [Salichs & Moreno (2000)], for a complete study of behaviour-based robotics see Arkin (1998). A basic classification of programs (behaviours) useful for intelligent systems can be found in Russell & Norvig (1995): Simple reflex, Model-based reflex, Goal-based, and Utility-based. Where, a learning program consists of: a learning element for making improvements, a performance element for selecting external actions, a critic element for modifying the performance element, and a problem generator for suggesting actions. Some examples of behaviour based architecture can be found in Chung et al. (1998) and Nicolescu & Mataric (2002).

On the other hand, the Qualitative Physic Compilers (QPC) can be used for programming behaviours as well [Keppens & Shen (2001)]: the main task is to extrapolate the

behaviour of the system described in a scenario, from a set of initial conditions using a domain theory of model fragments or knowledge (each fragment contains the model of a system's fragment), each model with different level of abstraction. This type of methodology can be useful for the application of the theory of artificial intelligence in the development of intelligent systems.

In addition, an architectures may be ***portable*** to be flexible and able to adapt to others robot platforms [Orebäck & Christensen (2003) and Virk (2003)]. In some case, the architecture must be ***reconfigurable*** to be flexible and able to adapt to changes of the robot kinematic, dynamic, and control configurations [e.g., Fujita et al. (1999) and Ji et al. (2003)]. In other cases, the architecture must be ***self-reconfigurable*** to be flexible and able to adapt to the changes of the modular self-reconfigurable robots [e.g., Murata et al. (2001), Rus et al. (2002), Stoy et al. (2002) and Yim et al. (2003)].

II.3 The task: Push and pull

In robotics, push and pull are manipulation tasks that have been used for positioning and orienting objects on the plane, in contrast with grasping that can exploit the third dimension. Push and pull are two tasks that require different analyse. Pushing has been studied for rearranging objects [e.g., Rus et al. (1995), Lynch & Mason (1996), Ben-Shahar & Rivlin (1998a) and Ben-Shahar & Rivlin (1998b)]. Pulling has been studied in the context of multibody wheeled mobile robots, also called trailers [e.g., Laumond & Risler (1996), Laumond (1998), Wang & Cartmell (1998), Jiang & Nijmeijer (1999), Huo & Ge (2001) and Astolfi et al. (2004)].

II.3.1 Push

Moving an object from one position to another is one of the most basic tasks for a robotic manipulator. The pick-and-place is the common task for manipulators provided with a gripper. This way, the grasp planning and path planning can be decoupled if the grasp can resist all forces that could reasonably act on the object during the motion. Using the task of grasping for moving too heavy or too large objects it is not the best approach because underutilizes the resources available of the robot. Pushing without grasping can be used for positioning and orientating objects instead, but the manipulator path planning cannot be decoupled [Lynch & Mason (1996)].

Pushing is a type of graspless or nonprehensile manipulation, among the others, e.g., tumbling, pivoting, objects on support surface, tray-tilting, striking, juggling, throwing, rolling, etc. [Lynch & Mason (1996)]. Pushing allows groups of objects to be simultaneously manipulated, moreover, larger and heavier objects can be easily manipulated, and most importantly it requires a cheaper robot structure than does grasping. On the other hand, pushing is constrained to a support surface, that is, the object manipulation is restricted to two dimensions. Control problems come up when pushing because it is mechanically unstable and sometimes is an irreversible process. Grasping can easily change its path or reverse its moves upon the realization of a wrong decision or unexpected changes in the environment [Ben-Shahar & Rivlin (1998b)].

The motion of a pushed object is generally unpredictable due to the unknown support friction forces. If the pushing contact resists all expected forces during the motion on the manipulator path, then the object can be effectively rigidly attached to the pusher and the push is called *stable push*. Stable pushing (finding a pushing direction – point contact or stable line contact – that causes the object to remain fixed to the manipulator) can be achieved by having two or more simultaneous pushing contacts, line contact [Lynch & Mason (1996)].

The ability of positioning and orientating objects in the plane by pushing with wheeled mobile manipulators, instead of grasping and lifting, was studied by Ben-Shahar & Rivlin (1998a) and Ben-Shahar & Rivlin (1998b). The problem of pushing a box by applying stable pushes using a nonholonomic wheeled mobile robot was studied by Lynch & Mason (1996). Tan et al. (2003) worked with a nonholonomic cart that was pushed by a mobile manipulator, where a pushing task required both force and position control along the same task direction, in this case the high degree of kinematic redundancy produced by the fusion of the platform and manipulator make possible to perform secondary tasks.

II.3.2 Pull

Pulling has been oriented to the study of pulling carts with wheeled mobile robots [e.g., Laumond (1998) and Lizárraga et al. (2001)], where the problem of controllability and stabilizability is reduced to the analysis of chains of carts (trailers) joined by pivots for planning their movements to reach certain configuration [e.g., Laumond & Risler (1996), Laumond (1998), Wang & Cartmell (1998), Jiang & Nijmeijer (1999), Lamiraux et al.

(1999), Huo & Ge (2001) and Astolfi et al. (2004)]. They used the same analysis theory of wheeled mobile robots, see section II.2.

II.4 Mobile manipulators

A mobile manipulator is a robotic arm mounted on a mobile platform [e.g., Yamamoto & Yun (1993) and Bayle et al. (2003a)]. A more general definition is: *"Mobile manipulator is a term to refer to a robot that combine capabilities of locomotion and manipulation"* [Bayle et al. (2003b)].

Due to the additional degrees of freedom introduced by the platform, a mobile arm manipulator is endowed with a high degree of kinematic redundancy with an increased workspace as well, enabling it to accomplish secondary tasks, e.g., joint torques can be optimized, singular configurations of the manipulator can be avoided, and the decoupled force/position control along the task direction can be achieved [Tan et al. (2003)].

The problems on mobile arm manipulators have been roughly classified in [Bayle et al. (2003a)]: control, path optimization problem, dynamic modelling, and multi-robot cooperation. Where a systematic modelling for this kind of robots and the extension of the manipulability for robotics arms to mobile arm manipulators can be found there, as well.

The control strategies for mobile arm manipulators can be classified into two: 1) separated control of the mobile platform and the manipulator arm, and 2) integrated control of both the mobile platform and the manipulator arm [Tan et al. (2003)].

For the purposes of this work, the study of the mobile manipulators has been classified into single mobile manipulators, cooperative mobile manipulators, and modular and reconfigurable robots.

II.4.1 Single mobile manipulators

Yamamoto & Yun (1993) presented the problem of having given an arbitrary end-effector trajectory; the position of the manipulator in certain preferred configurations is calculated while the mobile platform is moving. The control algorithm is developed using nonholonomic system theory because of the nonholonomic characteristic of the mobile platform. Using the nonlinear state feedback technique to linearize the system, the input-

output linearization and input-output decoupling is achieved by taking the reference point as the coordinates of the base of the two degrees manipulator, in order to move the platform in such a way that the manipulability preferred configuration is maximized while the manipulator end-effector is tracking the predefined trajectory.

Lynch & Mason (1996) analyzed the problem of positioning and orientating objects in a plane by pushing when the manipulator lacks the size, strength, or dexterity to grasp and lift them, where the "grasp" (pushing contact) and manipulator path cannot be decoupled. Then, applying a stable pushing (using two or more pushing points to the object) and modelling the support friction forces, it is possible to simultaneously design the pushing contact and manipulator path such that the contact resists all the expected forces during the motion, maintaining its configuration relative to the pusher when the pusher is moving. So, a pushing path is formed by a series of stable pushes.

Yamamoto & Yun (1996) studied the effect of the dynamic interaction between the manipulator and the mobile platform, where the role of the mobile platform is to position a manipulator in a preferred configuration when the end point of the manipulator tracks a desired trajectory. Using an input-output feedback linearization the linear and decoupled input-output transformation, which was stabilized by a linear feedback, was found. Four different control schemes were analyzed.

Chen & Zalzala (1997) studied the problem of given the trajectory of the manipulator end-effector, near-optimal trajectories for the mobile platform and manipulator joints were obtained using a robust genetic algorithm, which solved the problem of torque minimization, manipulability maximization, and obstacle avoidance for a mobile manipulator. The Newton-Euler equations were used to obtain the complete dynamics of the system.

Ben-Shahar & Rivlin (1998a) and Ben-Shahar & Rivlin (1998b) presented the problem of planning the movements of a nonprehensile mobile pusher manipulator that tried to rearrange several movable objects in its workspace, as a result of using nonprehensile manipulation, the pushing manipulation required a planner with a deep look ahead to avoid trap configurations due to existence of the irreversible configurations produced for the nonprehensile manipulation.

Lamiraux et al. (1999) presented two experiments for a mobile robot pulling a trailer in known environments and its practical navigation system. The first one with the trailer hooked up above the wheel axis of the robot, and the second one with the trailer hooked up behind the wheel axis. The mobile robot moved sufficiently slowly to make all the dynamical effects negligible. The nonholonomic path planning was defined as: since both systems are small-time controllable, any collision free path can be approximated by a sequence of collision-free feasible paths, being addressed by dealing separately with the physical constraints due to the obstacles and the kinematic constraints due to the wheels. Firstly it built a collision free path ignoring the nonholonomic constraints of the system (geometric planner: random path planner), and secondly, this path was approximated by a sequence of collision free feasible sub-paths computed by a steering method (steering exactly any nonholonomic system to a goal is an open problem). A method for transforming a feasible path into a feasible trajectory for the articulated system subjected to bounds of velocities and accelerations was presented as well. The motion control was based on a classical trajectory tracking control law for mobile robots. When the robot went forward the trailer was not taken into account and the robot was stabilized according to a simple control law. When the robot went backwards was defined a virtual robot, symmetrically to the real robot with respect to the wheel axle of the trailer, where the steering variables and state space of the virtual robot were defined with respect to the reference robot variables. So, the virtual robot went forward and virtually pulled the trailer, then was possible to apply the same control law. A movie of the experiment can be seen at http://www.laas.fr/~jpl/movies/trailer.mov, as is cited in the paper.

Dong et al. (2000) proposed a global tracking controller based on the dynamics of the defined tracking error and the extended Barbalat's lemma for mobile manipulators in the generalized coordinates space, which ensures full state global asymptotic convergence of the system given desired trajectory.

Tanner & Kyriakopoulos (2000) analyzed a mobile manipulator that was kinematically steered between two arbitrary configurations amongst obstacles, while avoiding both obstacles and singular configurations, using a discontinuous feedback law derived from a navigation function under the influence of a special potential field for nonholonomic motion planning for mobile manipulators moving in cluttered environments.

Holmberg & Khatib (2000) presented the design of an omnidirectional mobile robot called powered caster vehicle (PCV) for mobile manipulators. The control of the system is based on force control of the linearized unit mass system.

Papdopoulos et al. (2002) developed a planning methodology for nonholonomic mobile manipulator in presence of obstacles that employs smooth and continuous functions such as polynomials.

Bayle et al. (2003a) presented a systematic modeling of nonholonomic mobile manipulators, where the standard definition of manipulability is extended and generalized for mobile manipulators.

Bayle et al. (2003b) presented a kinematic modeling and velocity redundancies for a mobile manipulator, where the manipulability of the system is defined and is used in a pseudo-inversion control scheme.

Tan et al. (2003) presented an approach to decoupled/force position control of the end-effector along the same direction for a holonomic redundant mobile manipulator (a robot arm mounted on a holonomic mobile platform) and an approach to nonholonomic cart pushing with a holonomic mobile manipulators, where its kinematic redundancy enabled independent control of force and position along the same task directions. Firstly, using a kinematic redundancy resolution method the robot was kept away from singular configurations. Secondly, using a decoupled force/position control scheme, the end effector was provided with both force control and motion control long the same task direction. Finally, using an integrated task planning and control approach allowed the mobile manipulator to accomplish the regulation of its output force and position independently, and to coordinate the motion of a pushed cart and mobile manipulator by a common motion reference point. The problems associated with mobile manipulators were classified in: 1) kinematic redundancy: additional tasks, and optimization of certain criteria, 2) modelling and control: considering the mobile robot and the manipulator as two systems, and considering the mobile robot and the manipulator as integrated system, 3) force control schemes: external force control, hybrid position/force control, and impedance control, and 4) motion planning: in tracker-trailer systems is based on the kinematic model, but in mobile manipulators the nonholonomic cart can be manipulated by a regulated output force given by the mobile manipulator. A virtual moving coordinate frame parallel to the inertial world frame and attached to the same origin of the robot

moving frame is given to easily integrate the robot platform model and the robot manipulation model. The dynamic model is decoupled and linearized using the nonlinear feedback technique in a unified frame. The coordination of the platform and the robot arm (to resolve the kinematic redundancy) is based on additional task or based on the optimization of certain criteria, some examples are cited. Using a unifying approach, the kinematic and dynamic models are derived. The tracking controller is found applying a nonlinear feedback control, derived by the dynamic model, in order to find the linearized and decoupled dynamic model, afterwards, the linear system feedback is designed being asymptotically stable. An unified path planning and control architecture was proposed where an optimal trajectory is computed (by minimizing the control input energy), and the force planning of the cart is computed by decomposing the applied force in two and defining the control input as the lateral force using backstepping approach based on the design of the angular velocity of the cart. A force control input is designed after making a coordinate transformation.

II.4.2 Cooperative mobile manipulators

Ben-Shahar & Rivlin (1995) presented the problem of planning a pushing manipulation using a team of wheeled and legged mobile robots which try to rearrange several movable objects in their work space.

Mataric et al. (1995) used two autonomous six-legged robots equipped with object and goal sensing, and a repertoire of contact and light-following behaviors for pushing a large box towards a goal region. The main issue was the coordination of the task-sharing between the two autonomous six-legged robots by cooperating at three levels: sensing, action, and control, and takes the advantage of a simple communication protocol to compensate for the robots' noisy and uncertain sensing.

Rus et al. (1995) presented a manipulation task in which a coach was pushed out of a room by a team of two wheeled mobile robots, where the problem of the reorientation task is investigated by defining the following basic behaviors as: pushing-tracking (compliantly apply a sliding force on the face of the object), broken contact (when the robot has lost contact with the object because of it had gone past the end of the face), and impediment (despite of applying the maximum force, the robot does not change its position).

Khatib (1999) presented a cooperative mobile manipulation with dynamic decoupled end-effector task and platform self-posture control for two holonomic mobile platforms. The method is based on the modeling of the system's closed-chain dynamics, and the characterization of the internal forces.

Pimentel et al. (2002) studied the problem of transporting a relatively large box in an unknown, unstructured, cluttered environment with static and dynamic obstacles using two nonholonomic robots endowed with a behaviour-based architecture, where the problems of coordination, communication and interaction between multiple agents is the main issue. The basic behaviours are defined as follows: leading, following, world observation, goal, avoid, and grasp.

Sugar & Kumar (2002) studied the problem of grasping, manipulating, and transporting large and flexible objects (a box and a flexible board) in a two dimensional framework by mobile manipulators coordinated by a leader. The control algorithm is based on decoupling the locomotion and manipulation problems by controlling the robots independently and coordinately through sensing and communication. The arm is an in-parallel planar three degree-of-freedom mechanism.

Tanner et al. (2003) presented the problem of handling deformable material by multiple nonholonomic mobile manipulators in an obstacle environment using a new potential field technique for motion planning applicable to articulated, nonpoint nonholonomic robots that guarantees the collision avoidance and convergence, based on diffeomorphic transformations that exploits the resulting point-world topology. The approach is applied to yield a centralized coordinating control law.

II.4.3 Modular and reconfigurable robots

Burke & Durrant-Whyte (1993) presented the design and implementation of a decentralized drive system for modular wheeled mobile robots, where the kinematics of that kind of robots was obtained using the basic theory of planar motion. So, the computation of the controls actions to steer and/or drive each module is obtained by computing the inverse kinematics of each module when a proposed driven and steered imaginary wheel, which complies with the rolling rigid motion conditions (virtual wheel located at the centre of the vehicle), follows a trajectory of a given geometric path. Besides, the inverse kinematics is modular, where a least-squares estimator is required to

modularize the forward kinematics. An implementation is described and results are shown.

Fujita et al. (1998) and Fujita et al. (1999) described the implementation of a reconfigurable robot platform for robot entertainment systems based on OPEN-R. The architecture allows the replacement of components that can have various styles of robots: quadruped robot, wheeled-based, and car type robot. The OPEN-R architecture is divided into three levels of reconfigurability: 1) electrical/mechanical connection, 2) function / shape / topology / dynamics configuration, and 3) semantic configuration. The software architecture is built it with both a mechanical dependent layer and a mechanical independent layer (the portable layer).

Unsal & Khosla (2000) presented a modular serf-reconfigurable robotic system (a system that combines reconfigurability with autonomy) based on a collection of independent controlled moving links and passive connection elements called I-Cubes, where the design and hardware implementation is the main issue.

Mutambara & Durrant-Whyte (2000) studied a modular wheeled mobile robot with fully decentralized data fusion and control algorithms, where each module was endowed with its own hardware, software, driven and steered units, sensors, communication links, power units, kinematics, path planning, obstacle avoidance, sensor fusion, and control system. The control method was based on a modular decentralized kinematic model obtained from the kinematic model of each module by using plane motion kinematics to derive forward and inverse kinematics. A decentralized estimation based on a Kalman filter, which used estimation techniques in information space rather than state space, and control algorithms based on optimality principle, separation principle, and certainty equivalence principle was used to provide decentralized control to the wheeled mobile robot. Where the concept of the virtual wheel form Burke & Durrant-Whyte (1993) was used to formulate the modular kinematics of the wheeled mobile robot.

Jantapremjit & Austin (2001) presented a heterogeneous self-reconfigurable robot with genderless, fail-safe connecting mechanisms with three basic module types (joint, power and special units). A survey of existing modular robots is given as well where some limitations of homogeneous designs and connection mechanisms are discussed.

Yang et al. (2001) presented the kinematic design for a non-redundant 6-DOF modular reconfigurable in-parallel robot with fixed-dimension joints (*actuator modules*: revolute and prismatic; *passive joints*: rotary joint, pivot joint, and spherical joint; and end-effector modules) and variable dimension links (rigid links, module connectors, and mobile platform). The kinematic analysis, of a class of three-legged modular parallel robot, is based on a local frame representation of the product-of-exponentials formula.

Lee & Sanderson (2001) presented a real-time distributed computation scheme of the kinematics, dynamics, redundancy resolution, and control inputs for the control of Tetrobot modular parallel reconfigurable robotic system. Using a modular approach, Newton's equations of motion, and a recursive propagation algorithm, two different dynamic resolutions of redundancy schemes, based on the centralized Jacobian method and the distributed virtual force method, are proposed to optimize the actuating forces. Distributed dynamic control algorithms provide an efficient modular implementation of the control architecture for a large family of configurations.

II.5 Literature review analysis.

The automated production systems are built of existing automation technologies. These automation technologies consist of programmable automation technologies and materials handling technologies. Among the materials handling technologies are the transport systems, which are used to move materials inside the factory, warehouse, or other facilities. One category of the transport systems that has gained the attention in the past few decades is automated guided vehicles (AGVs). They have been applied to different scenarios and one example is mail processing, where automatic tows and forks move the mail between mail processing stations (see section I.2.1).

Unfortunately, in the United Kingdom, most of the Royal Mail Distribution Centres do not have this kind of automated guided vehicles integrated in their mail sorting process. They still use the traditional trolley transport system, which is handled by hand (see section I.2.2). As a result, their programmable automation technology (see section I.2.2.a) is decoupled because some parts of the process are totally dependent on human actions (see section I.2.2.b), therefore they cannot fully exploit the advantage of the flexible automation systems. In addition, the task of pushing and pulling wheeled loads by hand

requires the application of forces on the trolley cage that can lead to health disorders (see section I.2.3).

One proposed strategy to improve the efficiency of the Royal Mail Distribution Centres was to introduce automated guided vehicles (AGVs) as a suitable means of automating the material handling system (see section I.2.2.b). However, the current plant layout of the Royal Mail Distribution Centres, designed with this internal transport technology (mail trolleys), requires a significant restructuring of the work space and the process itself to introduce automatic guided tows or forks. So, in order to do not have to restructure, it was proposed to explore the possibility to develop a new type of wheeled mobile robot for moving trolleys by grasping without lifting (pushing and pulling while the load is held firmly, see section I.3). This way, the robot could be introduced to replace the operator without restructuring and replacing the current transport system. Moreover, it promises to be extremely simple (see sections I.2.3 and II.3.1). Consequently, some market research was done in order to find out more about the existing systems and only was found a "puller and pusher assistant" from Dane Industries (2003) (see section I.2.1.a). This system can be a partial solution to the problem of pushing and pulling mail trolleys, by eliminating the health disorders produced by the task, but does not solve the technical problem because the programmable automation technology is still decoupled due to the presence of the human actions in the process.

Thus, in the absence of availability of this type of robot in the market, a survey on mobile robot manipulators for pushing and pulling wheeled loads was done. It was found that in order to develop this new type of mobile robot it is required to specify the architecture and the platform of the robot. In addition, the development of the mobile robot behaviours should be oriented to quantitative rather than qualitative descriptions (see section I.2). Also, the complete solution of the robot must to include solving several problems like: motion planning, motion control, world modelling, location, physical platform and architecture (see section II.2).

The general theories and methodologies for the analysis of wheeled mobile robots are extensive (see section II.2). Only a few specific applications associated to the task of pushing and pulling were found (see section II.3). Also, it was noticed that the area of mobile manipulators has been growing fast in the past few years (see section II.4). So, the analysis of the literature review can be discomposed in the following topics:

- Existent wheeled mobile robots for pushing and pulling carts.
- The tasks of pushing and pulling with wheeled mobile robots.
- Analysis of the general methodologies useful for solving the problem.

II.5.1 Existing wheeled mobile robots for pushing and pulling carts

Despite the theory and general methodology for the analysis of wheeled mobile robots being extensive, only a few systems loosely relating to the problem of pushing and pulling wheeled loads with a wheeled mobile robot were found (see section II.4): a mobile robot pushing a trailer [Lamiraux et al. (1999)], a mobile robot arm manipulators pushing a wheeled chart [Tan et al. (2003)], and modular reconfigurable wheeled mobile robots [Burke & Durrant-Whyte (1993) and Mutambara & Durrant-Whyte (2000)].

Lamiraux et al. (1999) introduced a virtual robot in order to deal with the problem of pushing a wheeled chart (trailer) backwards with a wheeled mobile robot. The virtual robot works as puller when goes backwards and virtually pulls the trailer. When the robot goes forwards the trailer is not taken into account and the robot is stabilized according to a simple control law. This way he was able to apply the same classical trajectory tracking control law when going forwards and when going backwards (see section II.4.1).

Tan et al. (2003) presented an approach to push a nonholonomic cart with a holonomic redundant mobile manipulator (a robot arm mounted on a holonomic mobile platform). The main problem is to use the kinematic redundancy to achieve secondary tasks like: avoid singular configurations, decouple force/position control scheme along the same task direction, and coordinate the motion of a pushed cart and a mobile manipulator by sharing a common motion reference point. The motion planning in tracker-trailer systems is based on the kinematic model, but in mobile manipulators the nonholonomic cart can be manipulated by a regulated output force given by the mobile manipulator (see section II.4.1).

Burke & Durrant-Whyte (1993) and Mutambara & Durrant-Whyte (2000) presented the design and implementation of a decentralized drive system for modular wheeled mobile robots, where a kinematics is based on the basic theory of planar motion. The computation of the controls actions to steer and/or drive each module comes from

computation of the inverse kinematics of each module when a proposed driven and steered imaginary wheel, which complies with the rolling rigid motion conditions (virtual wheel located at the centre of the vehicle), follows a trajectory of a given geometric path to coordinate the whole wheels. The difference between those works is the use of the decentralized estimator. In the first case, a least-squares estimator was used to modularize the forward kinematics. In the second one, a Kalman filter and control algorithms were used to provide decentralized control to the wheeled mobile robot (see section II.4.3).

In summary, the method used for Lamiraux et al. (1999) is inadequate to solve the problem of pushing and pulling the trolley with a wheeled mobile robot, because the main problem is about the stabilization of a trailer (cart) about a trajectory when moving backwards. The method of Tan et al. (2003) is inadequate too, because the main problem is to exploit the kinematic redundancy of the holonomic mobile manipulator by applying the appropriate forces to the cart when the robot and the cart share the same reference contact point, and because we are not looking to design a mobile arm manipulator, which would be an expensive solution. The methods of Burke & Durrant-Whyte (1993) and Mutambara & Durrant-Whyte (2000) are more closely related and could be applied to solve the problem. But, the methods are based on the basic theory of planar motion and estimation techniques. So, the standard methods for wheeled mobile robots cannot be applied. However, some preliminary conclusions can be listed as a consequence of these works:

- From Lamiraux et al. (1999): To control the robot when going backwards use virtual wheeled elements.

- From Tan et al. (2003): the motion planning and tracking problem is based on the kinematic model.

- From Burke & Durrant-Whyte (1993) and Mutambara & Durrant-Whyte (2000): In order to coordinate and control multiple wheels of modular reconfigurable wheeled mobile robots use virtual wheeled elements.

II.5.2 The tasks of pushing and pulling with wheeled mobile robots

Push and pull are manipulation tasks for positioning and orienting objects on the plane, in contrast with grasping that can exploit the third dimension. Pushing has been used for

arranging objects while pulling has been used in the context of multibody wheeled mobile robots (see section II.3).

One problem of pushing is that the motion of a pushed object is generally unpredictable due to the unknown support friction forces. Lynch & Mason (1996) solve that problem by applying stable pushes to the object by having two or more simultaneous pushing contacts (line contact) to find a pushing direction that causes the object to remain fixed to the manipulator. Another problem with pushing objects with wheeled mobile robots is that pushing manipulation requires a planner with a deep look ahead to avoid trap configurations, due to existence of the irreversible configurations produced for the nonprehensile manipulation [Ben-Shahar & Rivlin (1998a) and Ben-Shahar & Rivlin (1998b)] (see sections II.3.1and II.4.1).

Pulling has been oriented to the study of pulling carts with wheeled mobile robots [e.g., Laumond (1998) and Lizárraga et al. (2001)], where the problem of controllability and stabilizability is reduced to the analysis of chains of carts (trailers) joined by pivots for planning their movements to reach certain configuration. They use the same analysis theory of wheeled mobile robots, where the main problem is to find the solution of the robot controllability or stabilizability, which arise from the well-known obstruction of Brockett (1983) (see sections II.2.1 and II.2.3.d).

Some preliminary conclusions can be obtained from this analysis:

- From Lynch & Mason (1996): a line contact between the robot and the object is needed to have a stable pushing. In addition, to grasp the object to maintain it fixed to the manipulator.

- From Ben-Shahar & Rivlin (1998a) and Ben-Shahar & Rivlin (1998b): grasp the object to avoid trap configurations. In consequence, to avoid a planner with a deep look ahead.

- From the area of multibody wheeled mobile robots: avoid trailer configurations.

II.5.3 Preliminary conclusion summary

Table II.2 shows a summary of the preliminary literature review conclusions that can be classified in motion control and motion planning:

- Motion control preliminary conclusions:
    - Use virtual wheeled elements for moving backwards and coordinating multiple wheels.
    - Use kinematic models for tracking.
    - Grasp the object to have a line contact (for applying stable pushes and pulls) and to avoid trailer configuration.
- Motion planning preliminary conclusions:
    - Use kinematic models for solving the motion planning problem.
    - Grasp the object to avoid both trap configurations and a planner with deep look ahead.

Table II.2: Preliminary conclusions of the literature review

	Use virtual wheeled elements	Base on Kinematic models	Line contact	Grasping the object
MOTION CONTROL				
Moving backwards	XXX			
Tracking		XXX		
Coordination of multiple wheels	XXX			
Stable pushing and pulling			XXX	XXX
Avoid trailer configuration			XXX	XXX
MOTION PLANNING				
Motion planning		XXX		
Avoid trap configurations				XXX
Avoid planner with deep look ahead				XXX

From the previous paragraph it can be concluded that a wheeled mobile robot with the following characteristics is needed (see Table II.2):

- A wheeled mobile robot able to grasp a wheeled load at least in two points, in order to have a line contact for applying stable pushes and pulls to the wheeled load.

- A wheeled mobile robot with tracking and motion planning approaches based on extended robot kinematic models.

- A wheeled mobile robot with virtual wheeled elements for coordinating the steering of multiple wheels, and for controlling the robot motion when going backwards.

With this type of new wheeled mobile robot more novel characteristics arise, because of the advantage and disadvantage of grasping, pushing, and pulling can be balanced (see sections I.2.3 and II.3). So, grasping wheeled loads with wheeled mobile robots without lifting has some very important advantages:

- It is useful for simultaneous manipulation of large or heavy groups of objects on the plane.

- It is mechanically stable.

- It is a reversible process (i.e. it is easy to change the path, reverse the movements upon wrong decisions or unexpected changes in the environment).

- It is predictable due to the knowledge of the kinematic and dynamic models of the system.

- It is quite useful for automatic load transfer systems that are centralized.

- Moreover, it requires a cheaper robot structure and size because there is no need to lift the object, as a result no lifting mechanisms and no heavy structures are needed, but a special gripper should be designed (not a real problem).

Some additional considerations should be taken into account when a wheeled mobile robot grasps a wheeled load without lifting it, because the robot embodies the kinematics, dynamics, and size characteristics of the load.

- Firstly, the wheeled mobile robot changes its kinematics properties when it holds a nonholonomic wheeled load losing mobility.
- Secondly, the size and dynamics changes are important to be taken into account.
- Finally, the configuration space (local model and global model) changes according with the actual load when a nonpoint robot approach is considered.

So, in order to find the solution to the problem, the specific general methodologies for the analysis of wheeled mobile robots are reviewed in the next section.

II.5.4 Analysis of the general methodologies useful for solving the problem

The general theories and methodologies for the analysis of wheeled mobile robots are extensive. Apart from already having been presented in the section II.2, some additional observations can be done about restricted mobility robots:

- The kinematic and dynamic models of restricted mobility robots have two main representations: At the configuration space and at the steering space. Several representations can be obtained depending on the steering systems selected, the position and orientation of the robot reference frame, the coordinate system used to specify the position and orientation of the wheels, and the coordinate system used to specify the robot reference point. But, in some senses all of them are equivalent.
- The techniques of input state feedback linearization, change of coordinates, and the differential flat property of restricted mobility robots can give new family of partially linearized representations: Caplygin, canonical, normal, chained form, etc., which can be controlled by using the existent controllability methods or stabilizability methods.

- The models of the robot can change along the time when they are operated on a specific singularity producing a new model with new behaviours. Also, the original model can be recovered by moving away from the singularity.

- One problem is the lack of the existence of complete kinematic models useful for design analysis. Most of the models are reduced to the input or input-output static state feedback linearized controllable version, which are irreducible models of the posture kinematic model of the robot. As a consequence, the dynamic models are reduced models, as well. Thus, the kinematic and dynamic models are incomplete models for mechatronics design viewpoint.

- The exact control of the posture stabilization of that kind of robots, which does not satisfy the well known Brockett necessary condition [Brockett (1983)], is still an open question. The control problem has been divided in two: controllability (motion planning problem) and stabilizability (motion control problem).

- The remaining nonlinear part of the input-output static state feedback linearized map has a stable zero dynamics with unstable behaviour when moving backwards.

- The use of potential fields and tracking to produce exact navigation techniques are good for automatically search nonholonomic trajectories to be followed by the robot.

- On the path planning problem, the definition of the configuration space that considers the configuration of the robot as a point, the multiresolution path planning techniques that divide the work space into reachable configuration space areas, the searching methods that find the free collision path, and the methods to avoid collision have proven to be good for planning and control the motion of the robot, where the robot trajectory should fulfil the nonholonomic characteristic of the robot.

- On wheeled mobile robots architecture the three layered architecture has been proven to be good [Gat (1998)], but the design of a well evolved architecture, able to accept specialized designed algorithms for each architecture layer, without losing the unity of the system, are needed.

II.6 The research work

So far, the literature review has revealed that no such system for moving a wheeled load with a wheeled mobile robot by ***grasping without lifting*** (pushing and pulling while the wheeled load is held firmly by the wheeled mobile robot) has been developed or investigated. It seems to be similar to the work done on modular reconfigurable wheeled mobile robots, however, only a few studies have been reported (see section II.5). Also, the literature review not only showed that many available techniques are promising to be used for developing the proposed system in its different aspects (see section II.5.2), but also showed that some approaches are convenient for the solution of the problem (see section II.5.3). On the other hand, the problem faces two important aspects, they are:

- The first one is associated to the behaviour of the robot itself due to the configuration of the robot's platform. It has to do with the kinematics, dynamics, state space representation and control of the robot, and their changes due to embodying and disembodying the wheeled load.

- The second one is associated with the behaviour of the robot due to the configuration of the environment. It has to do with the configuration space definition and motion planning problem, and their changes due to embodying and disembodying the wheeled load.

The particular configuration of the mail distribution centres (loosely structured environments), the task (to move mail trolleys), and the semiautonomy of the robot (dependent of the motion planner module) makes this problem unique, which arises in a few extensions of the theories that evolve in new research questions.

II.6.1 The aim of the research

"The aim of the present research is to determine whether or not it is possible to develop a semiautonomous wheeled mobile robot, for material handling applications, which is able to apply stable pushes and pulls when holding a wheeled load firmly (mail trolleys) and while it moves them about a loosely structured environment and between the load and unload stations."

II.6.2 The research questions

Based on literature review analysis and the above aim of the research, many questions arise about the technical and practical issues of the system. In particular, questions associated with the robot behaviour linked to the robot platform, which determines the robot platform configuration, and questions associated to the robot behaviour linked to the work space, which determine the robot architecture structure, they are:

- What is the desired configuration of the robot platform?
    - Which are the kinematics, dynamics, and state space models of the robot?
    - Which is the desired control approach?
    - How the kinematic, dynamic, and dimension changes of the robot, due to the embodying and disembodying the wheeled load, affects the overall behaviour of the system?
    - How should the system behave in the presence of static and mobile obstacles?
- What is the desired architecture of the robot?
    - What are the desired components of the architecture?
    - What are the desired reactive, task sequencer, and deliberative approaches for the motion planning problem?
    - How does the kinematic, dynamic, and dimension changes of the robot, due to the embodying and disembodying the wheeled load, affect the overall motion planning approach?
    - How should the motion planner react in the presence of static and mobile obstacles?

II.6.3 The scope of the research

The complete solution to the problem of moving wheeled loads with a semiautonomous wheeled mobile robot seemed to arrive at the synthesis of an useful *flexible platform*, and the development of a *reconfigurable architecture* based on *behaviours*, with its respective

path planning and *trajectory generation* modules, and a *switching coordinator* of behaviours.

The scope of this research will span the theoretical bases to solve the problem by using input-output static state feedback linearization techniques and tracking artificial goals (a technique similar to the artificial potential field approach but based on a goal domain approach instead on a potential fields), as follows:

- Propose a unifying theory for modelling conventional WMRs, with the following:
    - Kinematic and dynamic models of the robots, taking into account the most relevant configuration space variables and their respective irreducible controllable versions.
    - Robot modelling using nonlinear static state feedback linearization techniques.
- The selection and testing of the useful particular cases:
    - Selection of the particular cases.
    - Testing by simulating particular cases.
- Propose a motion planning approach based on artificial goals and tracking:
    - Purpose a motion planning approach.
    - Testing by simulating the motion planning approach.

II.7 The outline of the research

The outline of the research is organized as follows: The necessity of developing a new wheeled mobile robot useful for moving wheeled loads was introduced in **Chapter I.** The state of the art about the problem of developing a semiautonomous mobile robot by reviewing the most relevant prior research connected to the problem of moving wheeled loads with wheeled mobile robots, the analysis of the literature reviewed, and the outline of the research work (research aim, research questions, research scope, and research outline) is presented **Chapter II.** A brief review on nonlinear input-output static state feedback linearization for control multiple-input-multiple-output (MIMO) systems is

presented in **Chapter III.** An original unifying theory on conventional wheeled mobile robots is presented in **Chapter IV.** Based on a series of criteria, the particular cases are selected and modelled in **Chapter V.** The general architecture of the simulation and the simulation results for the particular cases are presented in **Chapter VI.** A motion planning approach based on artificial goals and tracking is proposed in **Chapter VII.** The general architecture of the simulation and the simulation results of the motion planning approach are presented in **Chapter VIII.** The conclusions, contributions, and further work will be found in **Chapter IX. Finally,** the references and appendices of the thesis are presented.

II.8 Summary

The literature on wheeled mobile robots is interdisciplinary and abundant. The main aspects on the kinematics, dynamics, state space representation, static state feedback linearization, motion control, motion planning, architecture of wheeled mobile robots and wheeled manipulators (for pushing and pulling manipulation tasks) have been reviewed in this chapter. These theories are useful for analyzing the behaviour and proposing the control techniques for wheeled robot manipulators. But, despite the fact that general methodologies for the analysis of wheeled mobile robots have been developed, they have not been applied to this type of problem, yet. During the literature review, any prior publication about pushing and pulling a wheeled load when the wheeled load is held firmly by a wheeled mobile robot or anything similar was found. Only a few works loosely related to the topic were found on a mobile robot pushing a trailer [Lamiraux et al. (1999)], in a mobile robot arm manipulators pushing a wheeled chart [Tan et al. (2003)], and in modular reconfigurable wheeled mobile robots [Burke & Durrant-Whyte (1993) and Mutambara & Durrant-Whyte (2000)].

Developing this new type of semiautonomous wheeled mobile robot (semidependent of the motion planner) for moving wheeled loads (mail trolleys) in a loosely structured environment, requires the development of a flexible platform and a motion planning approach to deal with the problem of changing the kinematics, dynamics and size of the robot, as a result of embodying and disembodying the wheeled load characteristics. Furthermore, the kinematic, dynamic, and state space representation modelling, the motion control, and the motion planning techniques are required to solve the problem with a high degree of performance and to take advantage of the natural response of the

system provided by the application of the nonlinear control feedback techniques. Moreover, the specification of the motion planning approach must come from the analysis of the behaviour of the robot models, and the specification of the robot platform must come from the analysis of the kinematic constraints. The nonlinear static state feedback linearization techniques, the selection of the linear feedback control law (that partially stabilizes the posture of the wheeled mobile robot), the design of feasible motion planning techniques based on the controllability of the robot models, and tracking a desired trajectory using an artificial goal method with ability to avoid obstacles must arrive to a theoretical validation of the feasibility of developing that kind of robot.

Therefore, an original unifying theory useful for modelling conventional wheeled mobile robots, and a new motion planning approach for planning the robot movements in loosely structured environments are required, as well. In this work will be proved that it is possible to develop that kind of mobile robot able to hold firmly wheeled loads (mail trolleys) while it moves them about the environment, and between the load and unload stations. The validation of the models will be done by simulating the problem in SIMULINK of MATLAB.

III A brief review on nonlinear input-output static state feedback linearization for multiple-input-multiple-output (MIMO) systems

III.1 Overview

This chapter summarises the most relevant concepts of nonlinear control system theory that are required for applying input-output linearization techniques, based on static state feedback for multiple-input-multiple-output (MIMO) dynamic systems. They will be applied in subsequent chapters to analyse the behaviour of wheeled mobile robots. Also, the reader is referred to the following literature for more extended explanations of the topic of this chapter: Nijmeijer & van der Schaft (1990), Murray et al. (1994), Isidori (1995), De Luca & Oriolo (1995), Henson & Seborg (1997), and Khalil (2002). The contents of the chapter are arranged as follows:

III.2 Multivariable nonlinear control systems

The input-output behaviour of a nonlinear MIMO system, which is linear with respect to the manipulated input, can be described by differential equations of the form:

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i$$
$$y_j = h_j(x) \quad 1 \le j \le p$$

(III.1)

where $x = x_1,...,x_n$ is the state variables vector, $u_i = u_1,...,u_m$ are the manipulated inputs, and $y_j = h_j(x_1,...,x_n)$ are the controlled outputs $y_1,...,y_p$. For simplicity, the nonlinear functions $f_1,...,f_n$, $g_1,...,g_m$, and $h_1,...,h_p$ are assumed C^∞ functions (smooth functions), so the vector fields $f(x_1,...,x_n)$, $g(x_1,...,x_n)$, and $h(x_1,...,x_n)$ are smooth in a domain $U \in R^n$.

III.2.1 Nonlinear change of coordinates

A nonlinear change of coordinates:

$$z(t) = \Phi(x)$$

(III.2)

is a *global diffeomorphism* on R^n, i.e. has the following properties:

i) $\Phi(x)$ is invertible, i.e. there exist a function $\Phi^{-1}(z)$ such that:

$$\Phi^{-1}(\Phi(x)) = x$$

(III.3)

for all $x \in R^n$.

ii) $\Phi(x)$ and $\Phi^{-1}(z)$ are both smooth mappings, i.e. have continuous partial derivatives of any order.

Where $\Phi(x)$ represents a R^n-valued function of n variables, i.e.:

$$\Phi(x)=\begin{pmatrix}\phi_1(x)\\ \phi_2(x)\\ \cdots\\ \phi_n(x)\end{pmatrix}=\begin{pmatrix}\phi_1(x_1,\ldots,x_n)\\ \phi_2(x_1,\ldots,x_n)\\ \cdots\\ \phi_n(x_1,\ldots,x_n)\end{pmatrix}$$

(III.4)

If the Jacobean matrix of $\Phi(x)$ is not singular at a point $x=x_0$, and $\Phi(x)$ is a smooth function defined on some subset U of R^n. Then, on a suitable open subset U^0 of U, containing x^0, $\Phi(x)$ defines a *local diffeomorphism.*

The application of a change of coordinates can be described by differentiating the both sides of (III.2) with respect to time. This yields:

$$\dot{z}=\frac{dz}{dt}=\frac{\partial\Phi}{\partial x}\frac{dx}{dt}=\frac{\partial\Phi}{\partial x}\dot{x}=\frac{\partial\Phi}{\partial x}[f(x)+g(x)u]$$

(III.5)

Then, expressing x as $x=\Phi^{-1}(z)$, one obtains:

$$\begin{aligned}\dot{z}&=\bar{f}(z)+\bar{g}(z)u\\ y&=\bar{h}(z)\end{aligned}$$

(III.6)

where the new description of the nonlinear system is expressed as:

$$\bar{f}(z)=\left[\frac{\partial\Phi}{\partial x}f(x)\right]_{x=\Phi^{-1}(z)};\quad \bar{g}(z)=\left[\frac{\partial\Phi}{\partial x}g(x)\right]_{x=\Phi^{-1}(z)};\quad \bar{h}(z)=[h(x)]_{x=\Phi^{-1}(z)}$$

(III.7)

III.2.2 Distributions

A *smooth distribution* Δ is defined by a set of smooth vector fields $\{f_1(x),\ldots,f_d(x)\}$ that are defined on an open set U of $x\in R^n$ such that:

$$\Delta(x) = span\{f_1(x),...,f_d(x)\}$$

(III.8)

the vectors $f_1(x),...,f_d(x)$ span a vector space (a subspace of the vector space), where $\Delta(x)$ denotes the value of Δ at a point x. Then Δ is called *non-singular distribution* if it is defined on an open set of U where there exist an integer d such that:

$$\dim(\Delta(x)) = d$$

(III.9)

for all $x \in U$. A *singular distribution*, distribution of variable dimension, is a distribution that no satisfies the above condition. A *regular point* of Δ is a point $x^0 \in U$ where the distribution Δ is non-singular on the neighbourhood U^0 of x^0. A *no regular point* of U is called *point of singularity*.

A *distribution is involutive* if the Lie bracket[1] $[\tau_1, \tau_2]$ of any pair of vector fields τ_1 and τ_2 belonging to a Δ is a vector field which belong to Δ, i.e.:

$$\tau_1 \in \Delta,\ \tau_2 \in \Delta \Rightarrow [\tau_1, \tau_2] \in \Delta$$

(III.10)

As a matter of fact, *any 1-dimensional distribution is involutive.*

The family of all involutive distributions containing Δ has a unique minimal element (the intersection of all members of the family) that is called involutive closure of Δ and denoted by $inv(\Delta)$ or $\overline{\Delta}$, which is the smallest involutive distribution containing Δ.

III.2.3 Codistributions

A *smooth codistribution* Ω is defined by a set of smooth vector fields $\{\omega_1(x),...,\omega_d(x)\}$ that are defined on an open set U of $x \in (R^n)^*$ such that:

---

[1] ***Lie bracket.*** The Lie bracket is the vector field resulted by applying the operation $ad_f = [f,g] = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g$ between two vectors fields f and g, both defined on an open subset U of R^n. The recursive operation is defined by $ad_f^i = [f, ad_f^{i-1} g]$ [Isidori (1995)].

$$\Omega(x) = span\{\omega_1(x),...,\omega_d(x)\}$$

(III.11)

$(R^n)^*$ is a dual space of R^n where the vectors $\omega_1(x),...,\omega_d(x)$ span a vector space (a subspace of the vector space), where $\Omega(x)$ denotes the value of Ω at a point $x \in U$. Since a codistribution is a vector space the concepts of addition, intersection, inclusion, dimension, regular points, and singular points can be easily extended to it.

Occasionally, it is this possible to construct a *codistribution* Ω starting from a given distribution Δ, and conversely, by finding the *annihilator* of $\Delta(x)$ for each x in U, which is the set of all covectors that annihilates all vectors of $\Delta(x)$:

$$\Delta^{\perp}(x) = \{\omega^* \in (R^n)^* : \langle \omega^*, v \rangle = 0 \text{ for all } v \in \Delta(x)\}$$

(III.12)

Since $\Delta^{\perp}(x)$ is a subspace of $(R^n)^*$, this construction identifies exactly a codistribution, in that assigns to each x of U a subspace $(R^n)^*$. The codistribution $\Delta^{\perp}$ is called the *annihilator* of Δ. Some care should be taken in constructing a distribution/codistribution annihilator because they may fail to be smooth.

If a distribution Δ is spanned by the columns of matrix F, whose entries are smooth functions of x, its annihilator is identified, at each $x \in U$, by the set of row vectors w^* satisfying the condition:

$$w^* F(x) = 0$$

(III.13)

Conversely, if a codistribution Ω is spanned by the rows of the matrix W, whose entries are smooth functions of x, its annihilator is identified, at each point x, by the set of vectors v satisfying:

$$W(x)v = 0$$

(III.14)

Thus, in this case $\Omega^{\perp}(x)$ is the kernel of the matrix W at the point x:

$$\Omega^{\perp}(x) = \ker(W(x))$$

(III.15)

If x^0 is a regular point of a smooth codistribution Ω, and $\dim(\Omega(x^0)) = d$, it is possible to find an open neighbourhood U^0 of x^0 and a set of smooth vectors fields $\{\omega_1,...,\omega_d\}$ that are linearly independent at each $x \in U^0$ as:

$$\Omega(x) = span\{\omega_1(x),...,\omega_d(x)\}$$

(III.16)

Moreover, every smooth vector field ω belonging to Ω can be expressed, on U^0, as:

$$\omega(x) = \sum_{i=1}^{d} c_i(x)\omega_i(x)$$

(III.17)

where $c_1,...,c_d$ are smooth real-valued functions of x, defined on U^0.

III.2.4 Frobenius theorem

Consider a non-singular distribution Δ, defined on an open set U of R^n, and let d denote its dimension. Thus a neighbourhood U^0 of each point x^0 of U, there exist d smooth vector fields $f_1,...,f_d$, all defined in U^0, which span Δ:

$$\Delta(x) = span\{f_1(x),...,f_d(x)\}$$

(III.18)

at each x in U^0. Besides, the codistribution $\Omega = \Delta^{\perp}$ is again smooth and non-singular, has dimension $n-d$ and locally around each x^0, is spanned by $n-d$ covectors fields $\omega_1(x),...,\omega_{n-d}(x)$. By construction, the covector field ω_j is such that:

$$\langle \omega_j(x), f_i(x) \rangle = 0 \text{ for all } 1 \le i \le d,\ 1 \le j \le n-d$$

(III.19)

for all $x \in U^0$, i.e. solves the equation:

$$\omega_j(x)F(x)=0$$

(III.20)

where $F(x)$ is the $n\times d$ matrix:

$$F(x)=(f_1(x),...,f_d(x))$$

(III.21)

At any fixed $x\in U$, the equation (III.20) can be simply regarded as a linear homogeneous equation in the unknown $\omega_j(x)$. The rank of the coefficient matrix $F(x)$ is d by assumption and the space solution is spanned by $n-d$ linearly independent row vectors. In fact, the row vectors $\omega_1(x),...,\omega_{n-d}(x)$ are exactly a basis of this space. Then, the *complete integrability* of the distribution $\Delta(x)$ spanned by the columns of the matrix $F(x)$ is a synonymous for "existence of $n-d$ independent solutions of the differential equation:

$$\frac{\partial\lambda_j}{\partial x}(f_1(x)...f_d(x))=\frac{\partial\lambda_j}{\partial x}F(x)=\omega_jF(x)=0$$

(III.22)

where the real valued smooth functions $\lambda_1,...,\lambda_{n-d}$ on U^0 span the annihilator $\Delta^\perp$ of dimension $n-d$ of a non-singular distribution Δ of dimension d, both defined on an open set U of R^n, such that:

$$span\{d\lambda_1,...,d\lambda_{n-d}\}=\Delta^\perp$$

(III.23)

Then the necessary and sufficient conditions for *complete integrability* can be restated as: "A non-singular distribution is completely integrable if an only if it is involutive (Frobenius)."

III.2.5 Local decomposition of control systems

A distribution Δ is said to be *invariant distribution* under the vector field f if the Lie bracket $[f,\tau]$ of f with every vector field τ of Δ is again a vector field of Δ, i.e. if:

$$\tau \in \Delta \Rightarrow [f,\tau] \in \Delta$$

(III.24)

Thus, if a non-singular involutive distribution Δ of dimension d is invariant under the vector fields $f, g_1,...,g_m$ and the distribution $span\{g_1,...,g_m\}$ is contained in Δ, then, it is possible to find a local coordinates transformation $z=\Phi(x)$ defined on a neighbourhood U^0 of x^0 for each point $x^0 \in U^0$ such that, in the new coordinates, the control system (III.1) can be represented by the equation of the form:

$$\begin{aligned}
\dot{\zeta}_1 &= f_1(\zeta_1,\zeta_2)+\sum_{i=1}^{m} g_{1i}(\zeta_1,\zeta_2)u_i \\
\dot{\zeta}_2 &= f_2(\zeta_2) \\
y_i &= h_i(\zeta_1,\zeta_2)
\end{aligned}$$

(III.25)

where $\zeta_1=(z_1,...,z_d)$ and $\zeta_2=(z_{d+1},...,z_n)$.

On the other hand, if a non-singular involutive distribution Δ of dimension d is invariant under the vector fields $f, g_1,...,g_m$, and the codistribution $span\{dh_1,...,dh_p\}$ is contained in the codistribution $\Delta^{\perp}$, then, it is possible to find a local coordinates transformation $z=\Phi(x)$ defined on a neighbourhood U^0 of x^0 for each point $x^0 \in U^0$ such that, in the new coordinates, the control system (III.1) can be represented by the equation of the form:

$$\begin{aligned}
\dot{\zeta}_1 &= f_1(\zeta_1,\zeta_2)+\sum_{i=1}^{m} g_{1i}(\zeta_1,\zeta_2)u_i \\
\dot{\zeta}_2 &= f_2(\zeta_2)+\sum_{i=1}^{m} g_{2i}(\zeta_2)u_i \\
y_i &= h_i(\zeta_1,\zeta_2)
\end{aligned}$$

(III.26)

where $\zeta_1=(z_1,...,z_d)$, $\zeta_2=(z_{d+1},...,z_n)$, and $\dfrac{\partial h_i}{\partial z_j}=0$.

III.2.6 Local reachability

If there is a d-dimensional non-singular distribution Δ with the properties that Δ is involutive, Δ contains the distribution $span\{g_1,...,g_m\}$, and Δ is invariant under the vector fields $f, g_1,..., g_m$, then at each point $x^0 \in U$ it is possible to find a coordinates transformation defined on a neighbourhood U^0 of x^0 and a partition of U^0 into slices of dimension d, such that the points reachable at some time T along the trajectories, starting from some initial state $x^0 \in U^0$, stay in U^0 for all $t \in [0,T]$, and lie inside a slice of U^0.

III.2.7 Local observability

If there is a d-dimensional non-singular distribution Δ with the properties that Δ is involutive, Δ is contained in the distribution $(span\{dh_1,...,dh_p\})^{\perp}$, and Δ is invariant under the vector fields $f, g_1,..., g_m$, then at each point $x^0 \in U$ it is possible to find a coordinates transformation defined on a neighbourhood U^0 of x^0 and a partition of U^0 into slices of dimension d, such that points on each slice produce the same output under any input u which keeps the state trajectory evolving on U^0.

III.2.8 Controllability

Consider a nonlinear MIMO system that is linear with respect to the manipulated input, and can be described by differential equations of the form:

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i \qquad \text{(III.27)}$$

where $x \in R^n$ is the state variables vector, $u \in R^m$ are the manipulated inputs. For simplicity, the nonlinear functions $f \in R^n$ and $g \in R^m$ are assumed C^∞ functions (smooth functions). So, the vector fields $f(x)$ and $g(x)$ are smooth in a domain $U \in R^n$. Let that each component of the control input vector $u \in R^m$ be a piecewise-constant input function over time. Denote by $x(t,0,x_0,u)$ the unique solution of equation (III.27) at time $t \geq 0$ given an input function $u(\cdot)$ and the initial condition $x(0) = x_0$.

The control system (III.27) is controllable if, for any choice of $x_1, x_2 \in R^n$, there exists a finite time T and an input $u:[0,T] \rightarrow U$ such that $x(T,0,x_1,u) = x_2$. So, let $R^V(x_0,\tau)$ be the reachable set from x_0 at time $\tau > 0$, with trajectories remaining in the neighbourhood V of x_0 for every $t \leq \tau$. Also, define:

$$R_T^V(x_0) = \bigcup_{\tau \leq T} R^V(x_0,\tau)$$

(III.28)

as the set of states reachable within time T from x_0 with trajectories remaining in the neighbourhood V of x_0.

The system (III.27) is called:

1. Locally accessible from x_0 if, for all neighbourhoods V of x_0 and all T, $R_T^V(x_0)$ contains a non-empty open set Ω.
2. Small-time locally controllable from x_0 if, for all neighbourhoods V of x_0 and all T, $R_T^V(x_0)$ contains a non-empty neighbourhood of x_0.

The previous definitions are local in nature. If the system (III.27) is locally accessible or small-time locally controllable for every x_0 in R^n, then the definitions becomes global definitions.

Small-time local controllability implies local accessibility as well as controllability, while local accessibility does not imply controllability in general. However, if no drift vector is present, then local accessibility implies controllability for symmetric systems.

The local accessibility and small-time local controllability of the system (III.27) can be tested by finding the accessibility algebra C (Control Lie Algebra or the involutive closure), which is the smallest subalgebra of $V(R^n)$ that contains $f, g_1, ..., g_m$ and is closed under the Lie bracket operation. The accessibility distribution $\overline{\Delta}$ of the system (III.27) is defined as:

$$\overline{\Delta} = span\{v \mid v \in C\}$$

(III.29)

i.e., the involutive closure of the distribution associated with $f, g_1, ..., g_m$. However, the computation of $\overline{\Delta}$ may be organized as an iterative procedure where:

$$\begin{aligned}
&\overline{\Delta} = span\{v \mid v \in \Delta_i; \quad \forall i \geq 1\} \\
&\Delta_1 = span\{f, g_1, ..., g_m\} \\
&\Delta_i = \Delta_{i-1} + span\{[g, v] \mid g \in \Delta_1, v \in \Delta_{i-1}\}, \quad i \geq 2
\end{aligned}$$

(III.30)

The above procedure stops after k steps, where k is the smallest integer such that $\Delta_{k+1} = \Delta_k = \overline{\Delta}$. Since $\dim \Delta_k \leq n$ necessarily, it follows that one stops after at most $n-m$ steps. The accessibility distribution may be used for verifying local accessibility.

Chow's theorem: If the accessibility rank condition:

$$\dim \overline{\Delta}(x_0) = n$$

(III.31)

holds, then the system (III.27) is locally accessible from x_0. If the accessibility rank condition holds for all $x \in R^n$, the system is locally accessible. Conversely, if the system (III.27) is locally accessible, then $\dim \overline{\Delta}(x_0) = n$ holds in an open and dense subset of R^n.

In particular, if the vector fields of the system are analytic, the accessibility rank condition is necessary and sufficient for local accessibility. If Chow's theorem is applied to a driftless control system:

$$\dot{x} = \sum_{i=1}^{m} g_i(x) u_i$$

(III.32)

it provides a sufficient condition for controllability. The same is true for systems with drift, if $f(x)$ is such that:

$$f(x) \in span\{g_1(x),...,g_m(x)\}, \quad \forall x \in R^n$$

(III.33)

Moreover, if system (III.32) is controllable, then its dynamic extension:

$$\dot{x} = \sum_{i=1}^{m} g_i(x) u_i$$
$$\dot{u} = v_i \quad i = 1,...,m$$

(III.34)

is also controllable. The converse is trivial true.

III.2.9 Stabilizability

The stabilization problem for the control system (III.27) consists in finding a feedback control law of the form:

$$u = \alpha(x) + \beta(x)v$$

(III.35)

so as to make a closed-loop equilibrium point x_e or an admissible trajectory $x_e(t)$ asymptotically stable. For point stabilization, x_e is typically an equilibrium point for the open loop system, i.e., $f(x_e) = 0$. Indeed, for the driftless control system (III.32), any point is an open loop equilibrium point. As for the tracking case, it is necessary to ensure that the trajectories to be stabilized are admissible for the system. This is of particular importance in the case of nonholonomic systems, for which the kinematic constraints preclude the possibility of following a generic trajectory.

III.2.10 Input-state feedback linearization

The ***input-state feedback linearization.*** A nonlinear system:

$$\dot{x} = f(x) + g(x)u$$

(III.36)

where $f : D \to R^n$ and $g : D \to R^{n \times p}$ are sufficiently smooth on a domain $D \subset R^n$, is said to be feedback linearizable (input-state linearizable) if there exist a diffeomorphism

$T: D \to R^n$ such that $D_z = T(D)$ contains the origin and the changes of coordinates $z = T(x)$ that transform the system into the form:

$$\dot{z} = Az + B\gamma(x)[u - \alpha(x)]$$

(III.37)

with (A,B) controllable and $\gamma(x)$ non-singular for all $x \in D$. This type of linearization of the state equation do not necessary linearizes the output equation.

III.3 Input-output static state feedback linearization.

The ***input-output state feedback linearization*** can be described as the procedure to find the *state feedback control law* and the *changes of coordinates* that transform the nonlinear system into an equivalent linear system, which map the inputs and outputs linearly.

III.3.1 The Input-Output state feedback linearization: SISO systems

Considering a SISO system (single-input-single-output system):

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

(III.38)

the derivative $\dot{y}$ is given by[2]:

$$\dot{y} = \frac{\partial h}{\partial x}\dot{x} = \frac{\partial h}{\partial x}[f(x) + g(x)u] = L_f h(x) + L_g h(x)u$$

(III.39)

If $L_g h(x) = 0$ then $\dot{y} = L_f h(x)$, thus it is independent to u.

The second derivative $y^{(2)}$ is denoted by:

---

[2] The first term $L_f h(x) = \frac{\partial h}{\partial x} f(x)$ is called the *Lie Derivative* of h with respect to f along f, i.e. the derivative of h along the trajectories of the system $\dot{x} = f(x)$, where the higher Lie derivates satisfy, in general, the recursion $L_f^k h(x) = L_f L_f^{k-1} h(x)$ where $L_f^0 h(x) = h(x)$.

$$y^{(2)} = \frac{\partial (L_f h)}{\partial x}[f(x) + g(x)u] = L_f^2 h(x) + L_g L_f h(x)u$$

(III.40)

if $L_g L_f h(x)u = 0$ then $y^{(2)} = L_f^2 h(x)$, thus it is independent to u as well. If we apply recursively this process, $h(x)$ may satisfy:

$$L_g L_f^{i-1} h(x) = 0, \quad i = 1,2,...,r-1; \quad L_g L_f^{r-1} h(x) \neq 0$$

(III.41)

then u does not appear in the equations of $\dot{y}, \ddot{y}, ..., y^{(r-1)}$ and appears in the equation of $y^{(r)}$ with a nonzero coefficient as:

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x)u$$

(III.42)

The foregoing equation shows clearly that the system is *input-output linearizable*, since the state feedback control:

$$u = \frac{1}{L_g L_f^{r-1} h(x)}\left[-L_f^r h(x) + v\right]$$

(III.43)

reduces the input-output map to:

$$y^{(r)} = v$$

(III.44)

which is chain of r integrators. In this case, the integer r is called the relative degree of the system. Using this relative degree of the system, one can find the following *nonlinear coordinate transformation:*

$$\phi(x)=\begin{pmatrix}\phi_1(x)\\ \phi_2(x)\\ \dots\\ \phi_r(x)\\ \phi_{r+1}(x)\\ \dots\\ \phi_n(x)\end{pmatrix}=\begin{pmatrix}h(x)\\ L_f h(x)\\ \dots\\ L_f^{r-1}h(x)\\ t_1(x)=L_f\phi_{r+1}(x)\\ \dots\\ t_{n-r}(x)=L_f\phi_n(x)\end{pmatrix}$$

(III.45)

where the $t_i(x)$[3] functions are chosen such that $\phi(x)$ is a *diffeomorphism*[4] on x^0:

$$L_g t_i(x)=0$$

(III.46)

and

$$z=\Phi(x)\rightarrow\begin{pmatrix}z_1\\ \dots\\ z_n\end{pmatrix}=\begin{pmatrix}\phi_1(x)\\ \dots\\ \phi_n(x)\end{pmatrix}$$

(III.47)

with invertible Jacobian matrix $\left|\frac{\partial\Phi(x)}{\partial x}\right|\neq 0$ that ensure the inverse operation:

$$x=\Phi^{-1}(z)\rightarrow\begin{pmatrix}x_1\\ \dots\\ x_n\end{pmatrix}=\begin{pmatrix}\phi_1(z)\\ \dots\\ \phi_n(z)\end{pmatrix}$$

(III.48)

Then, the description of the system in a *new coordinates* $z_i=\Phi_i(x)$, $1\le i\le n$, is found by changing of coordinates using the Jacobian matrix $\frac{\partial\Phi(x)}{\partial x}$, as follows:

---

[3] The value of these $t_i(x)$ additional functions can be fixed arbitrary, this is the simplest way to obtain them, with the drawback that the last state variables of the model will be affected by the input u in the form $\dot{z}_i=q_i(z)+p_i(z)\cdot u$ where $r+1\le i\le n$.

[4] Since r is $1\le r\le n$, it is always possible to find $n-r$ more functions $\phi_{r+1}(x),\dots,\phi_n(x)$ that have a no singular Jacobian matrix $\partial\phi/\partial x$ at x^0.

$$\dot{z}(x) = \frac{\partial \Phi}{\partial x}[f(x) + g(x)u]$$

(III.49)

and substituting (III.48) in (III.49) in order to find the new state-space description of the system in the new coordinates, as follows:

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 \\
&\cdots \\
\dot{z}_{r-1} &= z_r \\
\dot{z}_r &= b(z) + a(z)u \\
\dot{z}_{r+1} &= q_{r+1}(z) \\
&\cdots \\
\dot{z} &= q_n(z)
\end{aligned}$$

(III.50)

were

$$a(z) = \left[L_g L_f^{r-1} h(x)\right]_{x=\phi^{-1}(z)},\ b(z) = \left[L_f^r h(x)\right]_{x=\phi^{-1}(z)},\ \text{and}\ q_i(z) = \left[t_i(x)\right]_{x=\phi^{-1}(z)}$$

(III.51)

and

$$y = z_1$$

(III.52)

the static state feedback control law:

$$u = \frac{1}{a(z)}[-b(z) + v]$$

(III.53)

transforms the $r-th$ equation in (III.50) into $z_r = v$, as a result, the map between the transformed input v and the output y is exactly linear. Thus a linear feedback controller can be synthesizes to stabilize the z subsystem, e.g., using pole placement:

$$v = -\alpha_r z_r - \alpha_{r-1} z_{r-1} - \ldots - \alpha_1 z_1$$

(III.54)

with the characteristic polynomial for the linear subsystem:

$$s^r + \alpha_r s^{r-1} + ... + \alpha_2 s + \alpha_1 = 0$$

(III.55)

III.3.2 The Input-Output decoupling linearization: MIMO systems

The *input output decupling linearization* is an extension of the input-output linearization approach for single-input-single-output (SISO) processes to the multiple-input-multiple-output (MIMO) processes. This extension is also called *input-output decoupling* because the input-output response is both linearized and decoupled. The input-decoupling problem is to find a *diffeomorphism* and a *state feedback control law* such that:

- The map between the transformed inputs v and controlled outputs y is linear;
- The $i-th$ output y_i is decoupled from all inputs v_j for $i \neq j$.

In this chapter only the *input-output decoupling* based on the *static state feedback* control technique is considered.

III.3.2.a Input output MIMO system representation

The input-output behaviour of a nonlinear MIMO system that is linear with respect to the manipulated input can be described by differential equations (III.1) of the form:

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x) u_i$$
$$y_j = h_j(x) \quad 1 \leq j \leq p$$

(III.56)

where x is the state variables vector with state variables $x_1,...,x_n$, u_i are the manipulated inputs $u_1,...,u_m$, and $y_j = h_j(x_1,...,x_n)$ are the controlled outputs $y_1,...,y_p$. For simplicity, the nonlinear functions $f_1,...,f_n$, $g_1,...,g_m$, and $h_1,...,h_p$ are assumed C^∞ functions (smooth functions), so the vector fields $f(x_1,...,x_n)$, $g(x_1,...,x_n)$, and $h(x_1,...,x_n)$ are smooth in a domain $U \in R^n$.

III.3.2.b The decoupling matrix

The nonlinear system (III.56) is said to have vector relative degree $\{r_1, r_2, ..., r_m\}$ at the point x^0 if:

1. $L_{gj} L_f^k h_i(x) = 0$ for all $1 \le i \le m$, $1 \le j \le m$, for all $k < r_i - 1$, and for all x in a neighbourhood of x_0.

2. The $m \times m$ decoupling matrix $A(x)$ is nonsingular at the point $x = x_0$, where

$$A(x) = \begin{bmatrix} L_{g_1} L_f^{r_1 - 1} h_1(x) & \dots & L_{g_m} L_f^{r_1 - 1} h_1(x) \\ \dots & \dots & \dots \\ L_{g_1} L_f^{r_m - 1} h_m(x) & \dots & L_{g_m} L_f^{r_m - 1} h_m(x) \end{bmatrix}$$

(III.57)

The integer r_i represents the smallest relative degree of the i-th output with respect to any of the m inputs. Nonsingulatrity of the decoupling matrix $A(x)$ may be viewed as a MIMO generalization of the condition that $L_g L_f^{r-1} h(x_0) \neq 0$. It is a sufficient condition for the solution of the *input-output decoupling* control problem with *static state feedback.*

III.3.2.c The normal form

Because $A(x)$ is no singular, the diffeomorphism $[\xi^T, \eta^T]^T = \phi(x)$ can be constructed by choosing the first r coordinates, where $r = r_1 + ... + r_m$, as:

$$\xi_k^i = \phi_k^i(x) = L_f^{k-1} h_i(x)$$

(III.58)

where $1 \le k \le r_i$ and $1 \le i \le m$. It is always possible to find $n - r$ additional coordinates $\eta^T = [\phi_{r+1}(x), ..., \phi_n(x)]$ such that:

$$\phi(x) = [\phi_1^1(x), ..., \phi_{r_1}^1(x), ..., \phi_1^m(x), ..., \phi_{r_m}^m(x), \phi_{r+1}(x), ..., \phi_n(x)]^T$$

(III.59)

has Jacobian matrix, which is nonsingular (invertible) at the point $x = x_0$. Moreover, if the distribution:

$$G = span\{g_1, ..., g_m\}$$

(III.60)

is involutive near x^0, it is always possible to choose the additional coordinates $L_{g_j}\phi_{r+i}(x) = 0$ for all $1 \le i \le n-r$ and for all $1 \le j \le m$. In general, the involutivity condition is not satisfied and the normal form is:

$$\begin{aligned}
\dot{\xi}_1^i &= \xi_2^i \\
\dot{\xi}_2^i &= \xi_3^i \\
&\cdots \\
\dot{\xi}_{r_i}^i &= b_i(\xi,\eta) + \sum_{j=1}^{m} a_{ij}(\xi,\eta)u_j \\
\dot{\eta} &= q(\xi,\eta) + \sum_{j=1}^{m} p_j(\xi,\eta)u_j \\
y_i &= \xi_1^i
\end{aligned}$$

(III.61)

where $1 \le i \le m$. The functions $a_{ij}(\xi,\eta)$ are elements of the decoupling matrix A expressed in terms of the transformed coordinates; the functions b_i are defined as:

$$b_i(\xi,\eta) = L_f^{r_i} h_i\left[\phi^{-1}(\xi,\eta)\right]$$

(III.62)

The vector q is defined as in the SISO case; and the vector $p_j(\xi,\eta)$ are defined as:

$$p_j(\xi,\eta) = \begin{bmatrix} L_{g_j}\phi_{r+1}\left[\phi^{-1}(\xi,\eta)\right] \\ \cdots \\ L_{g_j}\phi_n\left[\phi^{-1}(\xi,\eta)\right] \end{bmatrix}$$

(III.63)

The r_i-th equation in the normal form can be collected and written as:

$$\begin{bmatrix} \xi_{r_1}^{1} \\ \dots \\ \xi_{r_m}^{m} \end{bmatrix} = b(\xi,\eta) + A(\xi,\eta)u$$

(III.64)

III.3.2.d Controller design

Since the decoupling matrix is nonsingular, the *static state feedback* control law which-achieves *input-output decoupling* can be derived from (III.61) as:

$$u = A^{-1}(\xi,\eta)[v - b(\xi,\eta)]$$

(III.65)

where v is an m-dimensional vector of the transformed input variables. This control law – which can be interpreted as the MIMO generalization of the input-output linearizing control law – changes (III.64) as follows:

$$\begin{bmatrix} \xi_{r_1}^{1} \\ \dots \\ \xi_{r_m}^{m} \end{bmatrix} = \begin{bmatrix} v_1 \\ \dots \\ v_m \end{bmatrix}$$

(III.66)

As a result, the *input-output response* is both linear and decoupled:

$$\begin{aligned} \dot{\xi}_1^i &= \xi_2^i \\ \dot{\xi}_2^i &= \xi_3^i \\ &\dots \\ \dot{\xi}_{r_i}^i &= v_i \\ y_i &= \xi_1^i \end{aligned}$$

(III.67)

where $1 \le i \le m$. Each input v_i can be chosen as in the SISO case:

$$v_i = -\alpha_{r_i}^i \xi_{r_i}^i - \alpha_{r_i-1}^i \xi_{r_i-1}^i - \dots - \alpha_1^i \xi_1^i$$

(III.68)

When expressed in the original coordinates, the decoupling control law has (III.65) the form:

$$u = -A^{-1}(x)b(x) + A^{-1}(x)v$$

(III.69)

where $A(x)$ is the decoupling matrix (III.57) and the m-dimensional vector $b(x)$ has elements $L_f^{r_i}h_i(x)$.

III.3.2.e Nominal stability

The input-output decoupling control law generally provides only partial linearization of the closed-loop system. An expression for the remaining $(n-r)$-dimensional nonlinear subsystem can be derived from (III.61) and (III.65):

$$\dot{\eta} = q(\xi,\eta) - P(\xi,\eta)A^{-1}(\xi,\eta)b(\xi,\eta) + P(\xi,\eta)A^{-1}(\xi,\eta)v$$

(III.70)

where the definition of the matrix P follows directly from (III.61). If each input v_i is designed as (III.68), the nonlinear subsystem can be written as $\dot{\eta} = \tilde{q}(\xi,\eta)$. The zero dynamics are obtained by setting $\xi(t) = 0$ for all $t \geq 0$:

$$\dot{\eta} = \tilde{q}(0,\eta)$$

(III.71)

This is a sufficient condition for local asymptotic stability of the closed-loop system, but not for global asymptotic stability.

III.4 Summary

A brief review of the main concepts on the standard techniques of nonlinear control for MIMO systems has been presented in this chapter, which are useful for the analysis of the input-output linearization techniques based on static state feedback. They have been used by many authors to analyze the dynamic behaviour of control systems. These techniques will be used to model, analyze, and control conventional wheeled mobile robots in the next chapters.

IV A unifying theory on conventional wheeled mobile robots

IV.1 Overview

An original general unifying theory useful to model, linearize, and stabilize conventional wheeled mobile robots based on the kinematic, dynamic, input static state feedback and input-output static state feedback models is developed in this chapter. This is an original general unifying theory for conventional wheeled mobile robots, which is an extension of the existing theory, in the sense that it uses all the configuration coordinates to model the kinematics and dynamics of the robot, according to a set of established restrictions. Moreover, a special contribution can be found in the state space representation of the kinematic model of robots with multiple centred steerable wheels, where the parametric functions for modelling the centred steerable wheels (able to coordinate the orientation of multiple centred steerable wheels) are introduced for the first time. The contents of the chapter are arranged as follows:

IV.2 Introduction

As has been mentioned in section II.2.3.a, different robot kinematic models can be found depending on the robot dimensions, type of wheels configuration, position of the robot frame reference system, definition of the wheel frame reference system, and the selection of the steering system. Also, the robot dynamic equations of motion have several representations, the most useful are two: the dynamic equations of motion expressed at the configuration space (e.g., obtained by applying the Lagrange formalism) and the dynamic equations of motion expressed at the steering space (e.g., obtained by applying a series of transformations on the dynamic equation of motion that is expressed at the configuration space, see section II.2.3.b as well). Besides, the input feedbacks techniques (see section II.2.3.c) and the selection of a set of output functions produces different input-output state space representation models (see section II.2.3.c). Thus, the method followed in this chapter is a mix of previous works (see sections II.2.3 and II.5.4), but original in the sense that both the position of the robot frame reference system and the definition of the wheel frame reference system have been done in order to reduce the number of variables needed to model the robot. Also, this is a general unifying theory, which is an extension of the existing theory in the sense that it uses all the configuration coordinates to model the kinematics and dynamics of the robot. Moreover, a special contribution can be found in the state space representation of the kinematic model of robots with multiple centred steerable wheels, where the parametric functions for modelling the centred steerable wheels (able to coordinate the orientation of multiple centred steerable wheels) are introduced for the first time. Besides, here is fully explained the following aspects about the models of the conventional wheeled mobile robots:

- External and internal kinematics.
- Kinematic motion properties and robot types.
- Configuration kinematic state space representation and properties.
- Posture kinematic state space representation and properties.
- Computation of the dynamics at the configuration space and at the steering space.
- Motor configuration.

- Configuration and posture dynamical state space representations and properties.
- Input-output static state feedback linearization.
- Stabilization of the linearized input-output map of the robot.

IV.3 Robot posture and wheels description

A wheeled mobile robot is equipped with motors that are driven by on board computer. It is assumed that the wheeled mobile robot is made of a rigid frame, non-deformable wheels that do not slip, and that it moves on a horizontal plane. Each wheel remains vertical and the wheel rotates around its horizontal axle. With these assumptions we can introduce the following definitions.

IV.3.1 Posture of the robot

A wheeled mobile robot has two frames of reference that are useful for the analysis and design (see Figure IV.1). The first one is the inertial fixed ***world frame*** of reference, represented by the coordinate system $\{x, y\}$ attached to the fixed point 0, which contains everything that is in the environment of the robot. The second one is the moving ***robot frame*** of reference, represented by the coordinate system $\{x_1, y_1\}$ attached to point P that is firmly fixed to the robot platform, which contains everything linked to the robot body. Thus, the ***robot posture*** is fully described by the vector:

$$\xi = [x \quad y \quad \theta]^T \tag{IV.1}$$

where ξ describes the position of the point $P = (x, y)$ and orientation θ of the robot frame $\{x_1, y_1\}$ relative to world frame $\{x, y\}$. The orientation θ is measured from the x axis to the x_1 axis. In addition, the ***posture rotation matrix***, an orthonormal rotation matrix, is defined to map the world frame into the robot frame $R(\theta) \in R^{3\times3}$, and vice versa $R(\theta)^T \in R^{3\times3}$:

$$R(\theta) \equiv \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{IV.2}$$

$$R(\theta) \cdot R^T(\theta) = I \tag{IV.3}$$

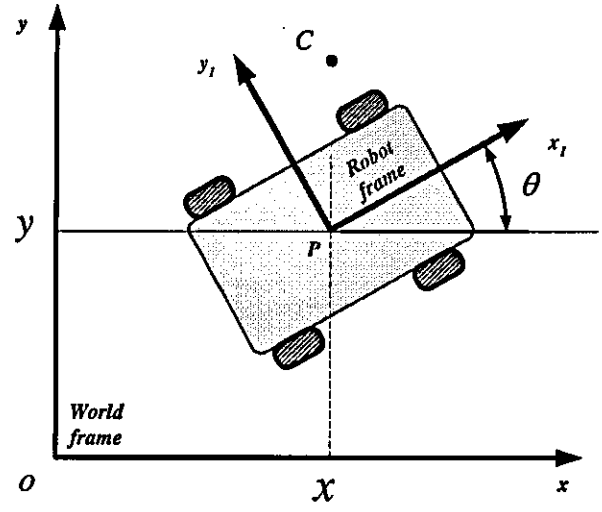


Figure IV.1: Posture definition

IV.3.2 Mapping position vectors between world and robot frames

Let us define the vector $\vec{OC}$ relative to the world frame as ${}^0C$, the vector $\vec{PC}$ relative to the robot frame as ${}^1C$, and the vector $\vec{OP}$ relative to the world frame as ${}^0P$ (see Figure IV.1). Then, the vector ${}^0C$ can be expressed as [Craig (1989)]:

$${}^0C = R^T(\theta){}^1C + {}^0P \tag{IV.4}$$

on the other hand the vector ${}^1C$ as:

$${}^1C = R(\theta)({}^0C - {}^0P) \tag{IV.5}$$

IV.3.3 Mapping velocity vectors of a point between world and robot frames

The velocity of a point is a free vector. A free vector may be positioned anywhere in space without loss or change of meaning provided that the magnitude and direction are preserved [Craig (1989)]. If the velocity of the above-defined point C with respect to the robot frame is expressed by the vector ${}^1\dot{C}$, and the velocity of point C with respect to the world frame is expressed by the vector ${}^0\dot{C}$ then:

$$ {}^0\dot{C} = R^T(\theta)\,{}^1\dot{C} \tag{IV.6} $$

on the other hand, the vector ${}^1\dot{C}$ as:

$$ {}^1\dot{C} = R(\theta)\,{}^0\dot{C} \tag{IV.7} $$

IV.3.4 Wheels description

Conventional wheels are only described in this work, they are: fixed wheels, centred steerable wheels, and off-centred wheels (castor wheels), which are useful for our purposes. For more wheels description the reader is referred to the work of Muir & Neuman (1987) and Campion et al. (1996).

The kinematic description of each wheel for ***pure rolling and nonslipping*** is calculated by computing both the longitudinal and orthogonal velocity components of the contact point of each wheel relative to the wheel frame. As a result, a system of Pfaffian[5] equations is found from the kinematic constraints contributed for each wheel. A general way to compute these equations, for a selected point P, is proposed in this chapter using the velocity propagation technique from link i to link $i+1$.

IV.3.4.a Velocity propagation technique from link i to link $i+1$

The kinematic constraints are computed using the velocity propagation technique from link i to link $i+1$ [e.g., Craig (1989)] (see Figure IV.2).

---

[5] Pfaffian form: A differential 1-from $w = \sum_{i=1}^{n} a_i(x)dx_i$, such that $w = 0$.

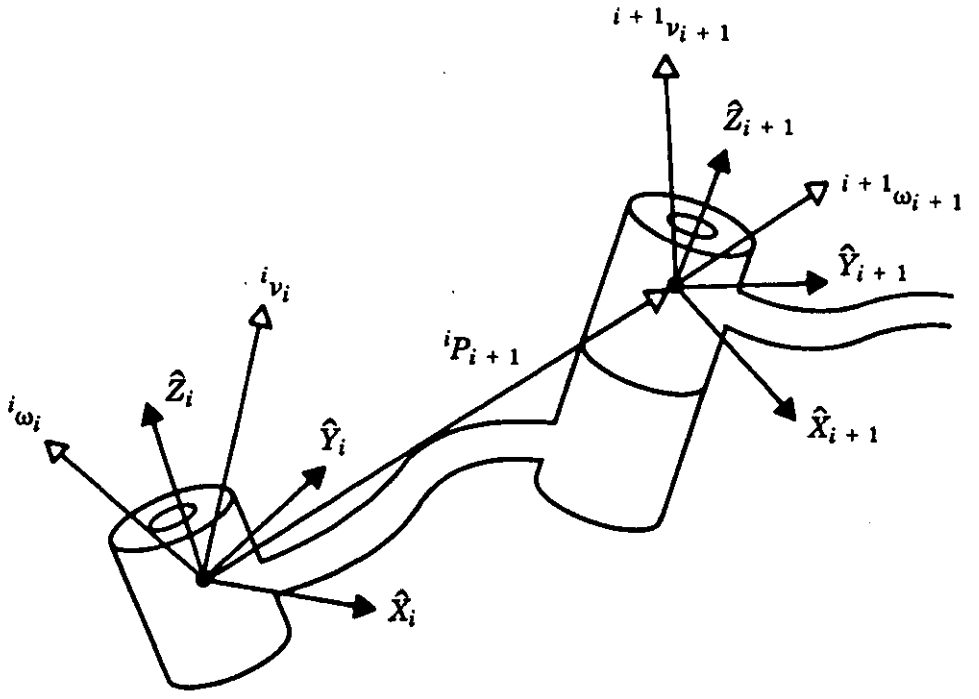


Figure IV.2: Velocity vectors of neighbouring links [Craig (1989)].

The angular velocity of the link $i+1$ relative to the frame $i+1$ ($^{i+1}\omega_{i+1}$) is the same as that of angular velocity of the link i relative to the frame $i+1$ ($^{i+1}_{i}R\,^{i}\omega_{i}$) plus a new component caused by the rotational velocity at the link $i+1$ relative to the frame $i+1$ ($^{i+1}\dot{\hat{\theta}}_{i+1}$):

$$^{i+1}\omega_{i+1}=^{i+1}_{i}R\,^{i}\omega_{i}+^{i+1}\dot{\hat{\theta}}_{i+1}$$

(IV.8)

where $^{i+1}\omega_{i+1}\in R^3$, $^{i}\omega_{i}\in R^3$, $^{i+1}\dot{\hat{\theta}}_{i+1}\in R^3$, and $^{i+1}_{i}R\in R^{3\times3}$ is the rotation matrix used to transform velocities expressed at the frame i into velocities expressed at the frame $i+1$.

On the other hand (see Figure IV.2), the linear velocity of the origin of frame $i+1$ relative to the frame $i+1$ ($^{i+1}v_{i+1}$) is the same as that of linear velocity of the origin of the frame i relative to the frame i ($^{i}v_{i}$), plus a new component caused at the distance $\left\|^{i}\mathrm{P}_{i+1}\right\|$ from the link i to the link $i+1$ expressed at the frame i due to the rotation of the vector $^{i}\mathrm{P}_{i+1}$ about the link i ($^{i}\omega_{i}\times^{i}\mathrm{P}_{i+1}$), and transformed to be expressed at the frame $i+1$ by applying the linear transformation $^{i+1}_{i}R$:

$$^{i+1}v_{i+1} = {}^{i+1}_{i}R(^{i}v_{i} + {}^{i}\omega_{i} \times {}^{i}\mathrm{P}_{i+1})$$

(IV.9)

where $^{i+1}v_{i+1} \in R^3$, $^{i}v_{i} \in R^3$, $^{i}\mathrm{P}_{i+1} \in R^3$, $^{i}\omega_{i} \in R^3$, and $^{i+1}_{i}R \in R^{3\times3}$ is the rotation matrix used to transform velocities expressed at the frame i into velocities expressed at the frame $i+1$.

Thus, the kinematic constraints are computed by propagating the velocities $\dot{\xi}_1 = [\dot{x}_1 \quad \dot{y}_1 \quad \dot{\theta}_1]^T$ at the robot frame $\{x_1, y_1\}$ to the wheel frame. On the other hand, from (IV.2) and (IV.7) it yields that $\theta_1 = \theta$, so $\dot{\xi}_1$ can be expressed as:

$$\dot{\xi}_1 = [\dot{x}_1 \quad \dot{y}_1 \quad \dot{\theta}]^T$$

(IV.10)

which is the posture velocity vector at the robot frame with respect to the robot frame and describes the linear velocity of the robot at the point P, represented by $\dot{x}_1$ and $\dot{y}_1$, and the angular velocity the robot frame, represented by $\dot{\theta}$, such that:

$$\dot{\xi}_1 = R(\theta)\dot{\xi}$$

(IV.11)

and

$$\dot{\xi} = R^T(\theta)\dot{\xi}_1$$

(IV.12)

Thus, in order to start the process of finding the kinematic constraints using (IV.8) and (IV.9) to compute $^{i+1}\omega_{i+1}$ and $^{i+1}v_{i+1}$ set $i=1$, $^{1}\omega_1 = [0 \quad 0 \quad \dot{\theta}]^T$, $^{1}v_1 = [\dot{x}_1 \quad \dot{y}_1 \quad 0]^T$, $^{1}\mathrm{P}_2 = [d \quad b \quad 0]^T$, $^{2}\dot{\theta}_2 = [0 \quad 0 \quad \dot{\alpha}]^T$ and the rotation matrix $^{2}_{1}R = R(\alpha)$ by substituting α in (IV.2), where the variables d, b, and α are defined in the following two sections. Use the figure Figure IV.3 and Figure IV.4 respectively to propagate the velocities.

IV.3.4.b Fixed and centred steerable wheels

Fixed and centred steerable wheels can be modelled as shown in Figure IV.3, where A is a fixed point on the robot frame, and represents the vertical axle for centre steerable

wheels, which is positioned at the centre of the wheel at a distance d along the x_1 axis, and a distance b along the y_1 axis. The ***wheel frame*** of reference $\{x_2, y_2\}$ is attached to the point A where x_2 axis is positioned along the wheel plane and heading to the linear velocity of the wheel contact point. The angle α represents the orientation angle of the wheel plane with respect to the x_1 axis, ϕ denotes the rotation angle of the wheel around its horizontal axle and $\dot{\phi}$ denotes the rotational velocity of the wheel around its horizontal axle. The radius of the wheel is denoted by r.

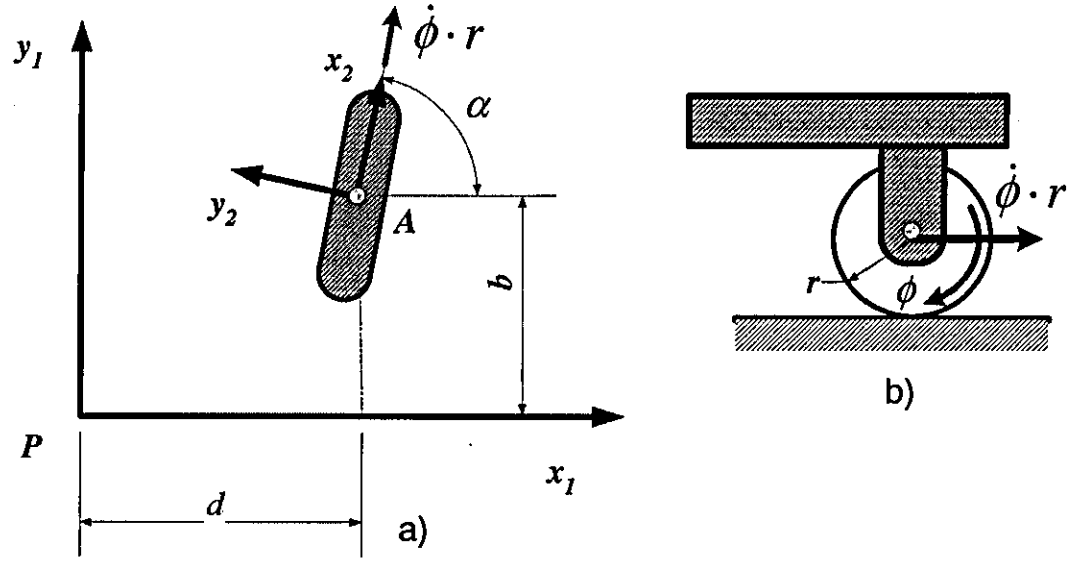


Figure IV.3: Fixed and centred steerable wheel

Then, by applying the velocity propagation technique from link i to link $i+1$ the constraints along and orthogonal to the wheel plane can be computed. That is to say, the kinematic constraints are computed propagating the velocities $\dot{\xi}_1 = [\dot{x}_1 \quad \dot{y}_1 \quad \dot{\theta}]^T$ at the robot frame $\{x_1, y_1\}$ to the wheel frame of reference $\{x_2, y_2\}$, these are:

Along the wheel plane:

$$[\cos(\alpha) \quad \sin(\alpha) \quad d\sin(\alpha) - b\cos(\alpha)]\dot{\xi}_1 - r\dot{\phi} = 0 \qquad \text{(IV.13)}$$

Orthogonal to the wheel plane:

$$[-\sin(\alpha) \quad \cos(\alpha) \quad d\cos(\alpha)+b\sin(\alpha)]\dot{\xi}_I = 0$$

(IV.14)

where d and b are constants, $\phi(t)$ is a time-varying angle, $\alpha(t)$ is a constant angle for fixed wheels and time-varying angle for centred steerable wheels.

IV.3.4.c Off-centred wheels

Off-centred wheels can be modelled as shown in Figure IV.4, where A is a fixed point on the robot frame placed at a distance d along the x_I axis and a distance b along the y_I axis, where the frame $\{x_2, y_2\}$ is attached. The x_2 axis is pointing to the centre of wheel. Note that the rotation of the wheel plane is around the vertical axle attached to the point A, which does not pass through the centre of the wheel. The orientation of the wheel plane is described by α. In addition, a point B is positioned at the centre of the wheel and connected to the frame by a rigid rod of constant length e. At this point B is placed the ***wheel frame*** of reference $\{x_3, y_3\}$, where x_3 is pointing to the same direction as the linear velocity of the wheel contact point. The angle ϕ represents the rotation of the wheel around its horizontal axle and $\dot{\phi}$ represents the rotational velocity of the wheel around its horizontal axle. The radius of the wheel is denoted by r.

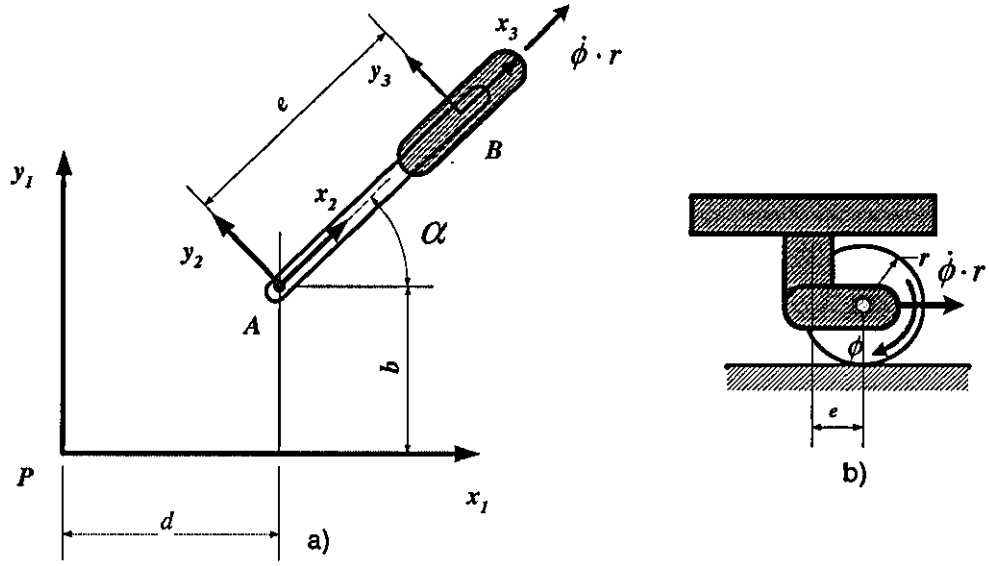


Figure IV.4: Conventional off-centred wheel

Same as the fixed and centred steerable wheels, the kinematic constraints using the velocity propagation technique from link i to link $i+1$ can be computed. That is to say,

the kinematic constraints are computed propagating the velocities $\dot{\xi}_1 = [\dot{x}_1 \quad \dot{y}_1 \quad \dot{\theta}]^T$ at the robot frame $\{x_1, y_1\}$ to the wheel frame of reference $\{x_3, y_3\}$, these are:

Along the wheel plane:

$$[\cos(\alpha) \quad \sin(\alpha) \quad d\sin(\alpha) - b\cos(\alpha)]\dot{\xi}_1 - r\dot{\phi} = 0 \tag{IV.15}$$

Orthogonal to the wheel plane:

$$[-\sin(\alpha) \quad \cos(\alpha) \quad e + d\cos(\alpha) + b\sin(\alpha)]\dot{\xi}_1 + e\dot{\alpha} = 0 \tag{IV.16}$$

where d, b, and e are constants, $\phi(t)$ and $\alpha(t)$ are time-varying angles.

IV.4 Kinematics of conventional wheeled mobile robots

The most important model of a wheeled mobile robot is the kinematic model, because it defines the type of robot and its kinematics properties: mobility, steerability, manoeuvrability, controllability and stabilizability [Campion et al. (1996)]. It is the base for developing dynamic models. Moreover, dynamic models are an extension of kinematic models for wheeled mobile robots.

The kinematic description of wheeled mobile robots starts with arranging the kinematics constraints of each wheel (see section IV.3.4) in matrix form. Using the subscripts f for fixed wheels, c for centred steerable wheels, and oc for off-centred wheels to describe the class of each wheel, and N to define the number of wheels, the following concepts can introduced.

IV.4.1 The configuration coordinates

A conventional wheeled mobile robot, having a total number of $N_T = N_f + N_c + N_{oc}$ conventional wheels has a total number of $n_T = 3 + N_T + N_c + N_{oc}$ configurations coordinates $q = [q_1, ..., q_{n_T}]^T$, is constituted of the whole set of the following configuration coordinates:

- $\xi = [x \quad y \quad \theta]^T$ posture coordinates, where x and y denote the position of the point P, and θ the orientation of the robot reference frame with respect to the fixed world inertial reference frame $\{x, y\}$.

- $\alpha_c = [\alpha_{c1} \quad \ldots \quad \alpha_{cN_c}]^T$ centred angular coordinates of the centred steerable wheels, where α_{ci} denotes the orientation angle of the i-th wheel's vertical plane, with respect to the axis x_1 of the moving robot reference frame $\{x_1, y_1\}$.

- $\alpha_{oc} = [\alpha_{oc1} \quad \ldots \quad \alpha_{ocN_{oc}}]^T$ off-centred angular coordinates of the off-centred wheels, where α_{oci} denotes the orientation angle of the i-th wheel's vertical plane, with respect to the axis x_1 of the moving robot reference frame $\{x_1, y_1\}$.

- $\varphi = [\varphi_f \quad \varphi_c \quad \varphi_{oc}]^T$ rotation coordinates of the wheels, with $\varphi_f = [\phi_{f1} \quad \ldots \quad \phi_{fN_f}]^T$, $\varphi_c = [\phi_{c1} \quad \ldots \quad \phi_{cN_c}]^T$ and $\varphi_{oc} = [\phi_{oc1} \quad \ldots \quad \phi_{ocN_{oc}}]^T$, where φ denotes vectors of rotational coordenaties constituted of ϕ rotation angles, which define the rotation angle of the wheel around its horizontal axle for each wheel.

This way, the generalized coordinates $q = [q_1, \ldots, q_{n_T}]^T$ can be expressed of the following form:

$$q = [\xi \quad \alpha_c \quad \alpha_{oc} \quad \varphi]^T$$

(IV.17)

with n_T generalized velocities:

$$\dot{q} = [\dot{\xi} \quad \dot{\alpha}_c \quad \dot{\alpha}_{oc} \quad \dot{\varphi}]^T$$

(IV.18)

where $\dot{\xi} = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$ is the robot posture velocity vector, $\dot{x}$ and $\dot{y}$ denote the linear velocities of the robot reference point along the corresponding axis, and $\dot{\theta}$ denotes the angular velocity of the robot's body. In addition, $\dot{\alpha}_c = [\dot{\alpha}_{c1} \quad \ldots \quad \dot{\alpha}_{cN_c}]^T$,

$\dot{\alpha}_{oc} = [\dot{\alpha}_{oc1} \quad \ldots \quad \dot{\alpha}_{ocN_{oc}}]^T$ and $\dot{\varphi} = [\dot{\varphi}_f \quad \dot{\varphi}_c \quad \dot{\varphi}_{oc}]^T$ are vectors related to the corresponding generalized angular and rotational velocities.

IV.4.2 The kinematics

The ***external kinematics*** (the robot kinematics model associated to the fixed world inertial reference frame $\{x, y\}$) for conventional wheeled mobile robots having n_T generalized coordinates q and n_T generalized velocities $\dot{q}$, and subject to a total of $m_{Tot} = 2N_T$ bilateral kinematic constraints can be expressed of the following form:

$$C_T(q, \dot{q}) = 0$$

(IV.19)

If a constraint equation is in the form $C_{Ti}(q) = 0$, or can be integrated into this form, it is called a holonomic constraint (geometric constraint). Otherwise, it is a nonholonomic constraint (kinematic constraint). How many holonomic and nonholonomic constraints the system has will be shown in section IV.5.1. Thus, the system (IV.19) can be written in the form (see section IV.3.4):

$$A_T(q)\dot{q} = 0$$

(IV.20)

where $A_T(q) \in R^{m_T \times n_T}$ is the matrix associated to the external kinematic constraints and is parameterized by the generalized coordinates θ , α_c and α_{oc} (i.e., $A_T(q) = A_T(\theta, \alpha_c, \alpha_{oc})$). The vector fields in $A_T(q)$ associated to the centred angular generalized velocities $\dot{\alpha}_c$ are null vector fields, they are related to the absence of the $\dot{\alpha}_c$ generalized angular velocities in the kinematic constraints (IV.19). It has a simple explanation: the generalized velocities $\dot{\alpha}_c$ do not produce generalised reaction forces. On the other hand, the presence of the generalized coordinates θ, α_c and α_{oc} in the matrix $A_T(q)$, and in consequence in the kinematic constraints (IV.19), has the following explanation: the value of the generalized coordinates θ, α_c and α_{oc} define the direction of the reaction forces produced by the generalized velocities $\dot{\xi}$. As a result, the generalized velocities $\dot{\alpha}_c$ cannot produce accelerations on the robot's body, but they can produce instantaneous changes on the direction of the accelerations produced by the robot

generalized velocities $\dot{\xi}$. Thus, when the centred angular generalized velocities $\dot{\alpha}_c$ are nonzero, and the other generalized velocities $\dot{\xi}$, $\dot{\alpha}_{oc}$ and $\dot{\phi}$ are zero the posture of the robot remains stable.

On the other hand, by setting $\theta = 0$, it is possible to find the ***internal kinematics*** of the robot (the kinematics associated to the moving robot reference frame $\{x_1, y_1\}$), which is independent of the robot orientation θ, and can be expressed as follows:

$$A_{T1}(q)\dot{q}_1 = 0$$

(IV.21)

where $A_{T1}(q) \in R^{m_T \times n_T}$ is the matrix associated to the internal kinematic constraints, which is parameterized by the generalized coordinates α_c and α_{oc} (i.e., $A_{T1}(q) = A_{T1}(\alpha_c, \alpha_{oc})$). The vector $\dot{q}_1 = [\dot{\xi}_1 \quad \dot{\alpha}_c \quad \dot{\alpha}_{oc} \quad \dot{\phi}]^T$, the vector of the generalized velocities at the robot frame $\{x_1, y_1\}$, is obtained by substituting (IV.12) in (IV.18) when $\theta = 0$. The vector $\dot{\xi}_1 = [\dot{x}_1 \quad \dot{y}_1 \quad \dot{\theta}]^T$ is the robot posture velocity vector at the robot frame $\{x_1, y_1\}$, see equation (IV.10).

IV.4.2.a Kinematic properties of wheeled mobile robots

The set of wheels' kinematic descriptions (IV.13), (IV.14), (IV.15), and (IV.16) can be arranged in the matrix form (IV.21). Using the notation O to denote matrices associated to the kinematic constraints orthogonal to the wheel plane, the notation A to denote matrices matrices associated to the kinematic constraints along the wheel plane and the subscripts defined in section IV.4, the matrix can be expressed as follows:

$$\begin{bmatrix} O_f & 0 & 0 & 0 & 0 & 0 \\ O_c(\alpha_c) & 0 & 0 & 0 & 0 & 0 \\ O_{oc}(\alpha_{oc}) & 0 & E_{oc} & 0 & 0 & 0 \\ A_f & 0 & 0 & -R_f & 0 & 0 \\ A_c(\alpha_c) & 0 & 0 & 0 & -R_c & 0 \\ A_{oc}(\alpha_{oc}) & 0 & 0 & 0 & 0 & -R_{oc} \end{bmatrix} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\alpha}_c \\ \dot{\alpha}_{oc} \\ \dot{\varphi}_f \\ \dot{\varphi}_c \\ \dot{\varphi}_{oc} \end{bmatrix} = 0$$

(IV.22)

From the constraints computed orthogonal to the wheel plane one can say that $O_f \in R^{N_f \times 3}$ is constant since the angles $\alpha_c(t) \in R^{N_f}$ are constant for fixed wheels,

$O_c \in R^{N_c \times 3}$ and $O_{oc} \in R^{N_{oc} \times 3}$ are respectively time-varying through the angles $\alpha_c(t) \in R^{N_c}$ and $\alpha_{oc}(t) \in R^{N_{oc}}$. The matrix $E_{oc} \in R^{N_{oc} \times N_{oc}}$ is a constant diagonal matrix where its elements are the dimension e of the off-centred wheels. From the constraints computed along the wheel plane one can say that $A_f \in R^{N_f \times 3}$ is constant since the angles $\alpha_c(t)$ are constant for fixed wheels, $A_c \in R^{N_c \times 3}$ and $A_{oc} \in R^{N_{oc} \times 3}$ are time-varying through $\alpha_c(t)$ and $\alpha_{oc}(t)$ respectively. The matrixes $R_f \in R^{N_f \times N_f}$, $R_c \in R^{N_c \times N_c}$, and $R_{oc} \in R^{N_{oc} \times N_{oc}}$ are constant diagonal matrixes where its elements are the radius r of the wheels. A matrix of dimension $n_T \times N_c$, constituted of a set of null vector fields, associated to the generalized velocities $\dot{\alpha}_c$ is added to complete the kinematic representation. Having defined the above mentioned, let us start with the properties associated to this matrix.

IV.4.2.a.1 *The kinematic motion properties of the robot*

At each instant time the motion of the robot frame can be viewed as an instantaneous rotation around the instantaneous centre of rotation (ICR), whose position with respect to the world and robot frame can be time-varying. For this reason, at each instant the linear velocity vector of any point of the robot frame is perpendicular to the straight line joining the point at the ICR. Moreover, from (IV.22), the robot mobility belongs to the null space of the matrix $O_{fc} = \begin{bmatrix} O_f \\ O_c(\alpha_c) \end{bmatrix}$ since $O_{fc} \cdot \dot{\xi}_1 = 0$, then $\dot{\xi}_1 \in null[O_{fc}]$. Thus, from the analysis of the matrix O_{fc} can be defined the properties of ***degree of mobility*** r_m, ***degree of steerability*** r_s, ***degree of manoeuvrability*** $r_M = r_m + r_s$, ***type of mobile robot*** $type(r_m, r_s)$, and the limitations for ***nondegenerate mobile robots***, as showed in Campion et al. (1996).

IV.4.2.a.1.1 Degree of mobility

The degree of mobility r_m is defined as $r_m = \dim[Null(O_{fc})] = 3 - rank(O_{fc})$[6]. That is to say, the dimension of $\dot{\xi}_1 \in R^3$ less the number of linearly independent rows in O_{fc} define the number of variables that are free to control, and as a consequence, the instant degrees

[6] $\dim[R^n] = \dim[Null(A)] + \dim[Range(A)]$ or $\dim[R^n] = \dim[Null(A)] + rank(A)$

of freedom that the robot can manipulate from the inputs without steering any of its wheels.

IV.4.2.a.1.2 Degree of steerability

The $rank[O_c(\alpha_c)]$ is called the degree of steerability r_s. It defines how many centred steerable wheels are free to control.

IV.4.2.a.1.3 Degree of manoeuvrability

The degree of manoeuvrability r_M is defined as $r_M = r_m + r_s$. It defines the total degrees of freedom that the robot can manipulate.

IV.4.2.a.1.4 A nondegenerate wheeled mobile robot

A wheeled mobile robot is nondegenerate if :

a) $$rank[O_f] \leq 1 \qquad \text{(IV.23)}$$

b) $$rank[O_{fc}] = rank[O_f] + rank[O_c(\alpha_c)] \leq 2 \qquad \text{(IV.24)}$$

which means:

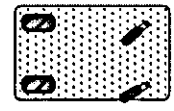
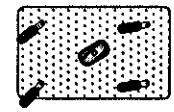
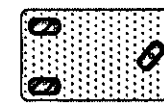
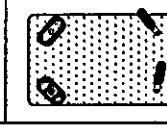
- Multiple conventional fixed wheels most be placed along a single common axle.
- The centres of the centred steerable wheels must not be placed on the common axle of the fixed wheels.
- The $rank[O_c(\alpha_c)]$ is the number of conventional centred steerable wheels that can be oriented independently in order to steer the robot, as follows:
    - 2 if there are not conventional fixed wheels.
    - 1 if there is one common axle of conventional fixed wheels.

  The remaining centred steerable wheels should be coordinated in order to fulfil the general planar motion of the robot rigid body.

IV.4.2.a.2 *Mobile robot type*

As a consequence of the previous section, nondegenerate wheeled mobile robots can be classified in five non-singular configurations, according to both degree of mobility (instantaneous degrees of freedom) and steerability (number of degrees of steerability), and the sum of both gives the total degrees of manoeuvrability (total degrees of freedom). The degree of manoeuvrability by itself does not define the type of robot since two robots with the same r_M value, but different r_m are not equivalent. Then the values of r_m and r_s are needed to define the type of robot as $type(r_m, r_s)$. Since, the degree of steerability, if it exists, is restricted to be $r_s = 2 - r_f$, where $r_f \leq 1$ is the number of common axles of conventional fixed wheels. Therefore, there are only five types of nondegenerate wheeled mobile robots according to the Table IV.1. Robots with $r_m = 3$ can assign freely the position of the instantaneous centre of rotation (ICR) and are called omnidirectional because they can move on the plane in any direction at each time instant. Robots with $r_m \leq 2$ are said to have restricted mobility because their instantaneous movements are restricted on the plane in some directions. Robots with $r_m = 2$ are restricted to positioning the ICR along the straight line defined by the axle of the fixed wheels. In addition, when a mobile robot is equipped with more that r_s centred steerable wheels, the motion of extra wheels must be coordinated to guarantee the existence of the ICR at each time instant.

Table IV.1: Robot type

		Type (r_m, r_s)				
		(3,0)	(2,0)	(2,1)	(1,1)	(1,2)
Degree	r_m	3	2	2	1	1
	r_s	0	0	1	1	2
	r_M	3	2	3	2	3
Robot configuration						
Motorized wheels		 Fixed	 Centred	 Off-centred		
Unmotorized wheels			 Passive off-centred	 Passive centred		

IV.5 The configuration kinematic state space representation

One way to find a controllable state space representation of the kinematic constraints is by finding a smooth codistribution of the matrix $A_{T_1}(q)$, which maps velocity inputs into velocity outputs [Isidori (1995)] (see section III.2.3). By selecting a set of r_m input velocities $\eta(t) = [\eta_1 \quad ... \quad \eta_{r_m}]^T$ and a set of r_s parameterizer centred angular coordinates $\alpha_s = [\alpha_{s1} \quad ... \quad \alpha_{sr_s}]^T$ from the set of centred angular coordinates α_c, and letting $N(A_{T_1})$ be the null space of $A_{T_1}(q)$, $d_{q_1} = \dim[A_{T_1}(q)]$, $r_{q_1} = rank[A_{T_1}(q)]$, $r_m = d_{q_1} - r_{q_1}$, $r_s = 2 - r_f$ and $r_f \leq 1$. Then, by spanning $N(A_{T_1})$, it is possible to find a set of r_m smooth vector fields $\sigma_{m1}(q),...,\sigma_{mr_m}$ parameterized by both r_s centred $\alpha_s = \alpha_{s1},...,\alpha_{sr_s}$ and N_{oc} off-centred $\alpha_{oc} = \alpha_{oc1},...,\alpha_{ocN_{oc}}$ angular configuration coordinates and a set of $N_c - r_s$ parametric functions $\alpha_p(\alpha_s) = [\alpha_{p_1} \quad ... \quad \alpha_{p_{N_c - r_s}}]^T$, associated to the remaining centred angular coordinates $\alpha_c \notin \alpha_s$, which can be expressed as follows:

$$\alpha_p(\alpha_s) = \{\alpha_{p_i}(\alpha_s) = \alpha_{ci}(\alpha_{s1},...,\alpha_{sr_s}) / \quad \alpha_{ci} \notin \alpha_{sj}: \quad i = 1,...,N_c - r_s; \quad j = 1,...,r_s\}$$

(IV.25)

useful to coordinate the remaining α_{ci} centred angular coordinates to guarantee the existence of the ICR at each time instant, where:

$$\dot{\alpha}_{p_i}(\alpha_s) = \frac{\partial(\alpha_{ci})}{\partial\alpha_{s1}}\dot{\alpha}_{s1} + ... + \frac{\partial(\alpha_{ci})}{\partial\alpha_{sr_s}}\dot{\alpha}_{sr_s}$$

(IV.26)

$$\dot{\alpha}_{p_i}(\alpha_s) = \frac{\partial(\alpha_{ci})}{\partial\alpha_{s1}}\zeta_1 + ... + \frac{\partial(\alpha_{ci})}{\partial\alpha_{sr_s}}\zeta_{r_s}$$

(IV.27)

$$\zeta_j = \dot{\alpha}_{sj}; \quad j = 1,...,r_s$$

(IV.28)

$$\zeta = [\zeta_1 \quad ... \quad \zeta_{r_s}]^T$$

(IV.29)

and finaly, a set of r_s vector fields $\sigma_{s1}(q),...,\sigma_{sr_s}(q)$ to complete the representation.

If we let $\Sigma(q) \in R^{n_T \times r_M}$ be the matrix consisting of the vectors $\Sigma(q) = [\sigma_{m1}(q),...,\sigma_{mr_m}(q),\sigma_{s1}(q),...,\sigma_{sr_s}(q)]$ parameterized by the angular coordinates $\alpha_s = \alpha_{s1},...,\alpha_{sr_s}$ and $\alpha_{oc} = \alpha_{oc1},...,\alpha_{ocN_{oc}}$, it is always possible express the state space representation of the kinematic constraints such as for all t:

$$\dot{q}_1 = \Sigma(\alpha_s, \alpha_{oc})[\eta \quad \zeta]^T$$

(IV.30)

where $\eta \in R^{r_m \times 1}$ and $\zeta \in R^{r_s \times 1}$ are the mobility and steerability components of the steering system of the vehicle, respectively. So, let us introduce the ***configuration rotation matrix*** as an extension of the ***posture rotation matrix*** (IV.2), as follows:

$$R_q = \begin{bmatrix} R(\theta) & 0 & 0 & 0 \\ 0 & I_{N_c} & 0 & 0 \\ 0 & 0 & I_{N_{oc}} & 0 \\ 0 & 0 & 0 & I_{N_T} \end{bmatrix}$$

(IV.31)

Where $R_q \in R^{(3+N_c+N_{oc}+N_T)\times(3+N_c+N_{oc}+N_T)}$, $R(\theta) \in R^{3\times 3}$ is the posture rotation matrix, and $I_{N_c} \in R^{N_c \times N_c}$, $I_{N_{oc}} \in R^{N_{oc} \times N_{oc}}$ and $I_{N_T} \in R^{N_T \times N_T}$ are identity matrices. Then, expressing (IV.30) as follows:

$$\dot{q}_1 = \begin{bmatrix} \dot{\xi}_1 \\ \dot{\alpha}_c \\ \dot{\alpha}_{oc} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \Sigma_\xi(\alpha_s) & 0 \\ 0 & \Sigma_c(\alpha_s) \\ \Sigma_{oc}(\alpha_s, \alpha_{oc}) & 0 \\ \Sigma_\varphi(\alpha_s, \alpha_{oc}) & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(IV.32)

where $\Sigma_\xi \in R^{3\times r_m}$ is a matrix associated to the posture coordinates, $\Sigma_{oc} \in R^{N_{oc} \times r_m}$ matrix associated to the off-centred angular coordinates, $\Sigma_\varphi \in R^{N_T \times r_m}$ matrix associated to the rotation coordinates, and $\Sigma_c \in R^{N_c \times r_s}$ is a matrix of associated to the centred angular

coordinates. Premultiplying both sides of (IV.32) by the transpose of the configuration rotation matrix R_q^T, yields:

$$\dot{q} = R_q^T \dot{q}_1 \equiv R_q^T \begin{bmatrix} \dot{\xi}_1 \\ \dot{\alpha}_c \\ \dot{\alpha}_{oc} \\ \dot{\varphi} \end{bmatrix} = R_q^T \begin{bmatrix} \Sigma_\xi(\alpha_s) & 0 \\ 0 & \Sigma_c(\alpha_s) \\ \Sigma_{oc}(\alpha_s, \alpha_{oc}) & 0 \\ \Sigma_\varphi(\alpha_s, \alpha_{oc}) & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(IV.33)

the complete state space representation of the configuration kinematic model of the robot, can be written in the form $\dot{q} = S_T(q)u$ as follows:

$$\dot{q} = S_T(q)u \equiv \begin{bmatrix} \dot{\xi} \\ \dot{\alpha}_c \\ \dot{\alpha}_{oc} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} R^T(\theta)\Sigma_\xi(\alpha_s) & 0 \\ 0 & \Sigma_c(\alpha_s) \\ \Sigma_{oc}(\alpha_s, \alpha_{oc}) & 0 \\ \Sigma_\varphi(\alpha_s, \alpha_{oc}) & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(IV.34)

where $S_T(q) \in R^{n_T \times r_M}$ and $u(t) = [\eta \quad \zeta]^T \in R^{r_M \times 1}$. Since $S_T(q) \in null[A_T(q)]$ then $S_T^T(q)A_T^T(q) = 0$.

Several different representations of the equation (IV.34) can be found if the robot does not have fixed wheels, centred steerable wheels or off-centred wheels, or any combination of those.

IV.5.1 Configuration kinematic state space representation properties

Let Δ_T be the distribution spanned by the vectors fields of $S_T(q)$:

$$\Delta_T = span\{s_{T1}(q), \ldots, s_{T r_M}(q)\}$$

(IV.35)

Let $\overline{\Delta}_T$ be the involutive closure of Δ_T, which is the smallest involutive distribution closed under the Lie bracket operation that contains Δ_T (see section III.2.8), therefore [Campion et al. (1996)]:

- The distribution $\Delta_T \in R^{n_T \times r_M}$ is not involutive.

- The ***degree of holonomy***, the number of constraints that can be eliminated from the generalized coordinates by integration of the kinematic constraints (holonomic constraints) is $M_h = \dim(q) - \dim(\overline{\Delta}_T)$, where $0 < M_h < m_{Tot}$.

- Because $\dim(\overline{\Delta}_T) \leq \dim(q)$, the configuration kinematic model is reducible. Furthermore, $\dim(\Delta_T) = r_m + r_s \leq \dim(\overline{\Delta}_T) \leq \dim(q) = 3 + N_c + N_{oc} + N_T$.

- The ***degree of nonholonomy***, the number of constraints that are nonintegrable and cannot be eliminated whatever the choice of generalized coordinates is $M_{nonh} = \dim(\overline{\Delta}_T) - (r_m + N_c)$.

IV.5.2 The posture kinematic state space representation

The posture kinematics state space representation is a subsystem of the configuration kinematic model (IV.34), which only takes into account the posture $\dot{\xi}$ and the centred angular $\dot{\alpha}_s$ generalized velocities of the system model (IV.34). By dividing the system (IV.34) into two subsystems (posture kinematic model and remaining generalized velocities) and two representations (when $N_c \neq 0$ and $N_c = 0$), we have:

The ***posture kinematic model*** (when $N_c \neq 0$):

$$\dot{q}_p = S_{q_p}(q)u \equiv \begin{bmatrix} \dot{\xi} \\ \dot{\alpha}_s \end{bmatrix} = \begin{bmatrix} R^T(\theta)\Sigma_\xi(\alpha_s) & 0 \\ 0 & I_{r_s} \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \tag{IV.36}$$

where $\dot{q}_p \in R^{3+r_s}$ and $I_{r_s} \in R^{r_s \times r_s}$ is an identity matrix associated to the generalized centred angular velocities $\dot{\alpha}_s$ and $u(t) = [\eta \quad \zeta]^T \in R^{r_u \times 1}$. The remaining generalized velocities (when $N_c \neq 0$):

$$\dot{q}_r = S_{q_r}(q)u \equiv \begin{bmatrix} \dot{\alpha}_p \\ \dot{\alpha}_{oc} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & \Sigma_p(\alpha_s) \\ \Sigma_{oc}(\alpha_s, \alpha_{oc}) & 0 \\ \Sigma_\varphi(\alpha_s, \alpha_{oc}) & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \tag{IV.37}$$

where $\dot{q}_r \in R^{(N_c - r_s)+N_{oc}+N_T}$ and $\Sigma_p \in R^{(N_c - r_s)\times r_s}$ is the matrix associated to the generalized parametric centred angular velocities $\dot{\alpha}_p$ and $u(t) = [\eta \quad \zeta]^T \in R^{r_M \times 1}$.

The ***posture kinematic model*** (when $N_c = 0$):

$$\dot{q}_p = S_{q_p}(q)u \equiv \dot{\xi} = R^T(\theta)\Sigma_\xi \cdot \eta$$

(IV.38)

where $\dot{q}_p \in R^3$ and $u(t) = \eta \in R^{r_m \times 1}$. The remaining generalized velocities (when $N_c = 0$):

$$\dot{q}_r = S_{q_r}(q)u \equiv \begin{bmatrix} \dot{\alpha}_{oc} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \Sigma_{oc}(\alpha_{oc}) \\ \Sigma_{\varphi}(\alpha_{oc}) \end{bmatrix} \eta$$

(IV.39)

where $\dot{q}_r \in R^{N_{oc}+N_T}$ and $u(t) = \eta \in R^{r_m \times 1}$.

IV.5.3 Some robot posture kinematic control properties

First, one can concentrate the analysis on a subsystem called posture kinematic model, (IV.36) and (IV.38) respectively. After that, in the complete subsystem called posture dynamical model by adding the remaining state variables $\dot{u} = \upsilon$. Therefore, the control properties of the posture kinematic model can be classified as follows:

1. Nonholonomy [Campion et al. (1996)]:

   a. The posture kinematic model $\dot{q}_p = S_{q_p} u$ of restricted mobility robots is not involutive. The involutive closure $\overline{\Delta}_p$ of the distribution $\Delta_p = span\{S_{q_p}\}$ has $\dim \overline{\Delta}_p \geq \dim q_p$. As a result, both (IV.36) and (IV.38) are irreducible, as a consequence of the Frobenius Theorem.

   b. The input matrix $S_{q_p}(q)$ has full rank:

$$rank[S_{q_p}(q)] = r_M \qquad \text{for all } q$$

(IV.40)

c. The involutive distribution $\overline{\Delta}_p = span\{inv(S_{q_p})\}$ has constant maximal dimension:

$$\dim \overline{\Delta}_p(q) = 3 + r_s \qquad \text{for all } q \tag{IV.41}$$

2. Controllability [Campion et al. (1996)]:

   a. The Kalman controllability rank condition of the linear approximation of the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots around an equilibrium state is r_M. Therefore, omnidirectional robots are completely controllable since $\dim(\dot{q}_p) = r_m$ and not controllable for restricted mobility robots where $r_M < \dim(\dot{q}_p) = 3 + r_s$.

   b. The control Lie algebra *CLA* (also called the involutive closure $\overline{\Delta}$) of the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots has constant full rank [Campion et al. (1996)], which implies controllability for driftless systems [Nijmeijer & van der Schaft (1990) and Isidori (1995)]. That is, the system can be always driven from any initial posture q_{p_0} to any final posture q_{p_f}, in any finite time, by manipulating the velocity control inputs $u = [\eta \quad \zeta]^T$.

3. Input feedback linearization [Campion et al. (1996)]:

   a. The posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of omnidirectional robots are full input state feedback linearizable.

   b. The restricted mobility robots are not full state linearizable, as a consequence of the clause 1.a. Also, the distribution $span\{S_{q_p}(q)\}$ of restricted mobility robots is not involutive. So, it is not input state feedback linearizable [Isidori (1995)].

   c. The largest linearizable subsystem of the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ by static state feedback is r_M, as a result of applying the

algorithm of Marino (1986). Therefore, the restricted mobility robots are partially linearizable by static state feedback.

d. The posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots is a differentially flat system [Campion et al. (1996)], which implies that it is full state linearizable by dynamic state feedback provided the η part of the input vector u is non zero [Campion et al. (1996) and d'Andréa-Novel et al. (1995)].

4. Stabilizability [Campion et al. (1996)]:

   a. The Brockett necessary condition [Brockett (1983)] for the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of omnidirectional robots is satisfied. Therefore, there is a smooth feedback control law that makes an equilibrium point $q_{p_f} = (\xi_f)$ stable and attractive.

   b. The Brockett necessary condition [Brockett (1983)] for the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots is not satisfied, $r_M < \dim(\dot{q}_p) = 3 + r_s$ [Campion et al. (1996)]. Therefore, they are not stabilizable by continuous static time-invariant feedback $u(q_p)$ [Bloch et al. (1990)].

   c. The posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots is a kind of driftless system, which is stabilizable by continuous time-varying static state feedback $u(q_p, t)$ [Coron (1992) and Pomet (1992)].

5. Input-output state feedback linearization [d'Andréa-Novel et al. (1995)]:

   a. The posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots is partially input-output static state feedback linearizable.

   b. The input-output map of the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots has relative degree 1, when a valid choice of output equations $y = h(q_p)$ is proposed.

6. Input-output state feedback stabilization [Campion et al. (1996)]:

   a. The partially linearized input-output map of the posture kinematic model $\dot{q}_p = S_{q_p}(q)u$ of restricted mobility robots is stabilizable by smooth static state feedback. The Kalman rank condition for linear systems is satisfied.

IV.6 Dynamics of conventional wheeled mobile robots

In mobile robotics, as in robot manipulators, the method of deriving the dynamics equation of motion is not unique (e.g. Newton's law and vector analysis, Newton-Euler equation, and Lagrange's formulation), but the Lagrange's formulation is the most used [Zhao & BeMent (1992)].

The Lagrange equations of motion of nonholonomic systems with Lagrange multipliers are commonly written in the form [Neimark & Fufaev (1972)]:

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} = \tau_i + \sum_{k=1}^{n-m} \lambda_k a_{ki} \qquad i = 1,...,n$$

(IV.42)

where T represents the kinetic energy of the robot, the variables q_i constitute a set of generalized coordinates of the system, τ_i is a generalized force at joint i (i.e., torque for revolute joints and force for prismatic joints) and represents the torque that can potentially be applied for rotation and orientation of the wheels of the robot, λ_k is the associated Lagrange coefficient to the corresponding nonholonomic constraint, a_{ki} is the coefficient associated to the nonholonomic constraints and $\sum_{k=1}^{n-m} \lambda_k a_{ki}$ represents the generalized reaction forces of the kinematic constraints.

The generalized joint torque τ_i at joint i is a function of the dynamic parameters of the link i, where each joint i is driven by an actuator (direct driving or gear driving). Then, the following torque contributions appear:

$$\tau_i = \tau_{ui} - \tau_{mi} - \tau_{fi} - \tau_{di} + \tau_{ei}$$

(IV.43)

where τ_{ui} is the actual driving torque at the joint i, τ_{mi} denotes the inertia torque due to the rotor of motor i, τ_{fi} is the torque due to the joint friction i, τ_{di} denotes vector of bounded unknown disturbances including unstructured unmodelled dynamics, and τ_{ei} is the torque caused by external forces and moments applied to the frame of the robot or to the end effector. Thus, the generalized forces vector can be defined as follows:

$$\tau_T = [\tau_1 \quad \dots \quad \tau_{n_T}]^T$$

(IV.44)

IV.6.1 Computation of the kinetic energy

The kinetic energy of the robot T is a scalar function composed only of terms whose dependence on the $\dot{q}_i$ is quadratic, that is why it is known as a "*quadratic form*" [Craig (1989)] and is given by:

$$T(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$$

(IV.45)

where $M(q) \in R^{n_T \times n_T}$ is the inertial matrix of the system. Besides, the kinetic energy of the robot T is given by the sum of the contributions of each link [Craig (1989)], as follows:

$$T = \sum_{i=1}^{n} T_i$$

(IV.46)

where T_i denotes the kinetic energy of the link i which can be computed by:

$$T_i = \frac{1}{2} m_i v_{C_i}^T v_{C_i} + \frac{1}{2} {}^i\omega_i^T \, {}^{C_i}I_i \, {}^i\omega_i$$

(IV.47)

where m_i is the mass of the link i, ${}^{C_i}I_i$ is the inertia tensor of the link i at the centre of the mass, v_{C_i} is the linear velocity of the centre of mass of the link i, and ${}^i\omega_i$ is the angular velocity of the link i with respect to the frame of the link i. The first term of the

equation (IV.47) is the kinetic energy due to the linear velocity of the link's centre of mass, and the second term is the kinetic energy due to the link's angular velocity. Since, the velocity of the centre of the mass can be expressed as a function of the linear velocity ${}^{i}v_i$ of the link i with respect to the frame of the link i as:

$$v_{C_i} = {}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{C_i} \tag{IV.48}$$

by applying the parallel axis theorem as in Craig (1989):

$${}^{i}I_i = {}^{C_i}I_i + m_i\left[{}^{i}P_{C_i}^{T}\,{}^{i}P_{C_i} I_3 - {}^{i}P_{C_i}\,{}^{i}P_{C_i}^{T}\right] \tag{IV.49}$$

solving (IV.49) for ${}^{C_i}I_i$:

$${}^{C_i}I_i = {}^{i}I_i - m_i\left[{}^{i}P_{C_i}^{T}\,{}^{i}P_{C_i} I_3 - {}^{i}P_{C_i}\,{}^{i}P_{C_i}^{T}\right] \tag{IV.50}$$

where $P_{C_i} = \begin{bmatrix} x_{C_i} & y_{C_i} & z_{C_i} \end{bmatrix}^T$ locates the centre of the mass relative to the frame of the link i, and I_3 is the 3x3 identity matrix. Then, substituting equations (IV.48) and (IV.50) the equation (IV.47) becomes:

$$T_i = \frac{1}{2} m_i \left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{C_i}\right)^T \left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{C_i}\right) + \frac{1}{2} {}^{i}\omega_i^T\, {}^{C_i}I_i\, {}^{i}\omega_i \tag{IV.51}$$

or

$$T_i = \frac{1}{2} m_i \left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{C_i}\right)^T \left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{C_i}\right) + \frac{1}{2} {}^{i}\omega_i^T \left({}^{i}I_i - m_i\left[P_{C_i}^{T} P_{C_i} I_3 - P_{C_i} P_{C_i}^{T}\right]\right) {}^{i}\omega_i \tag{IV.52}$$

IV.6.2 Computation of the inertial, Coriolis and centripetal elements

Substituting the resultant kinetic energy due to each link (IV.52) in (IV.46) the Lagrange equations of the robot can be computed, as follows [Khalil & Dombre (2001)]:

$$\Gamma_i = \frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} \qquad i = 1,...,n$$

(IV.53)

then, the elements $M(q)$ can be computed, as follows:

$$m_{ij}(q) = \frac{\partial \Gamma_i(q)}{\partial \ddot{q}_j} \qquad \text{for} \quad i = 1,...,n_T \quad \text{and} \quad j = 1,...,n_T$$

(IV.54)

where m_{ij} is the (i, j) element of the matrix:

$$M(q) = \begin{bmatrix} m_{11} & \cdots & m_{1n_T} \\ \cdots & \cdots & \cdots \\ m_{n_T 1} & \cdots & m_{n_T n_T} \end{bmatrix}$$

(IV.55)

and the elements of $C(q,\dot{q})$ can be computed, as follows:

$$C(q,\dot{q})\dot{q} = \dot{M}(q)\dot{q} - \frac{\partial T}{\partial q}$$

(IV.56)

IV.6.3 The dynamic equation of motion at the configuration space

The dynamic equation of motion at the configuration space with nonholonomic constraints for a wheeled mobile robot system having n_T -dimensional configuration space with n_T generalized coordinates $q = (q_1,...,q_{n_T})$, subject to m_{Tot} bilateral kinematic constraints, r_m degree of mobility, $r_s = 2 - r_f$ degree of steerability where $r_f \leq 1$ is the number of common axles of conventional fixed wheels, and $r_M = r_m + r_s$ degree of manoeuvrability, is represented by a set of n_T second-order coupled and nonlinear differential equations relating to the positions, velocities, and accelerations of the robot posture and robot joints. They can be written by expressing (IV.42) using (IV.55) and (IV.56) in the following form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = P\tau + A_T^T(q)\lambda$$

(IV.57)

where $M(q) \in R^{n_T \times n_T}$ is the inertia matrix, $C(q,\dot{q})\dot{q} \in R^{n_T \times 1}$ is the vector of centripetal and Coriolis forces, $P \in R^{n_T \times n_T}$ is the input transformation matrix, which selects the torque input components of the motorized joints that are effectively used as control inputs, $\tau \in R^{n_T \times 1}$ is the input vector, $A_T^T(q) \in R^{m_T \times n_T}$ is the transpose of the matrix associated to the external kinematic constraints, $\lambda \in R^{n_T \times 1}$ is the vector of the langrage multipliers. The properties of the dynamic equation can be summarised as:

Property 1. $M(q)$ is symmetric and positive definite.

Property 2. $\dot{M}(q) - 2C(q,\dot{q})$ is skew-symmetric.

Property 3. Because this kind of robots have only revolute joints, there exist positive constants $m_{\min}$, $m_{\max}$, $v_{\max}$, f_1 and f_2 such that $m_{\min} \le \|M(q)\| \le m_{\max}$, and $\|C(q,\dot{q})\| \le v_{\max}\|\dot{q}\|$, where $\|\cdot\|$ represent a suitable norm.

IV.6.4 The dynamic equation of motion at the steering space

With the purpose of finding the dynamic equation of motion at the steering space, let us differentiate (IV.34), it yields:

$$\ddot{q} = S_T(q)\dot{u} + \dot{S}_T(q)u \tag{IV.58}$$

substituting (IV.58) in (IV.57) and premultiplying both sides by $S^T(q)$. The desired model can be written as:

$$\overline{M}(q)\dot{u} + \overline{C}(q,\dot{q})u = \overline{P}\tau \equiv \overline{\tau} \tag{IV.59}$$

where $\overline{M}(q) = S_T^T M S_T$, $\overline{C}(q,\dot{q})u = S_T^T(M\dot{S}_T u + C)$, and $\overline{P} = S_T^T P$. Besides, $\overline{M}(q) \in R^{r_M \times r_M}$ is an inertia matrix, $\overline{C}(q,\dot{q})u \in R^{r_M \times 1}$ is the vector of centripetal and Coriolis forces, and $\overline{\tau} \in R^{r_M \times 1}$ is the input vector. $\overline{P}$ is a constant matrix that depends on the design specification of the mobile robot. Since $S_T(q)$ belongs to the null space of $A_T(q)$ then $S_T^T(q)A^T(q)$ vanishes. There are several properties that hold for this system:

Property 4. $\overline{M}(q)$ is symmetric and positive definite.

Property 5. $\dot{\overline{M}}(q) - 2\overline{C}(q,\dot{q})$ is skew-symmetric.

Property 6. $\overline{M}(q)$ and the norm of $\overline{C}(q)$ are bounded.

The properties 2 and 5 are useful for the stability analysis of certain control schemes based on Lyapunov stability analysis. There are several techniques for position control and trajectory tracking schemes based on the dynamic equation of motion, see e.g., Murray et al. (1994) and Khalil & Dombre (2001).

IV.6.5 The motor configuration

The vector τ_u is the driving torque vector constituted by the joint torques τ_α and τ_ϕ, which represent all the toques that can be potentially applied for the orientation and rotation of the robot's wheels respectively. In practice, only a limited number of motors will be used, which means that many components of τ_α and τ_ϕ will be identically zero. Moreover, if the matrix $\overline{P}$ in (IV.59) has full rank at any time instant [Yamamoto & Yun (1993)] or has a valid Moore-Penrose generalized matrix pseudoinverse [Yun & Sarkar (1998)], the motor configuration that guarantees the full manoeuvrability of the robot can be found by providing the following motors [Campion et al. (1996)]:

- N_c motors for steering each centred steerable wheel.
- $N_m \geq r_m$ additional motors for rotating some wheels around its axle and/or for steering some off-centred wheels that ensure the full robot mobility.

Thus, the torques developed at the steering space, due to the motor configuration, are defined as:

$$\overline{\tau} = S_T^T P \tau_u = \overline{P} \tau_u \qquad \text{(IV.60)}$$

IV.7 The configuration dynamical state space representation

With the purpose of finding the complete state space representation of the systems let us solve (IV.59) for $\dot{u}$:

$$\dot{u} = -\overline{M}^{-1}(q)\overline{C}(q,\dot{q})u + \overline{M}^{-1}(q)\overline{\tau}$$

(IV.61)

Thus, by adding the kinematic model (IV.34) to the dynamic equation (IV.61), the ***complete dynamical state space representation*** of the system can be written as follows:

$$\begin{aligned}\dot{q} &= S_T(q)u \\ \dot{u} &= -\overline{M}^{-1}(q)\overline{C}(q,\dot{q})u + \overline{M}^{-1}(q)\overline{\tau}\end{aligned}$$

(IV.62)

A suitable static nonlinear input state feedback may be applied:

$$\overline{\tau} = \overline{M}(q)\upsilon + \overline{C}(q,\dot{q})u$$

(IV.63)

So, the ***configuration dynamical state space representation*** of the robot is found by substituting (IV.63) in (IV.62). This way, the state equation is simplified to the form:

$$\begin{aligned}\dot{q} &= S_T(q)u \\ \dot{u} &= \upsilon\end{aligned}$$

(IV.64)

where $\upsilon \in R^{r_u \times 1}$ is the input vector of this model. Then, by adding the steering variables u to the generalized coordinates of the system, and defining the new generalized coordinates as $x_T = [q \quad u]^T$, the system (IV.64) can be expressed in the form:

$$\dot{x}_T = f_T(x) + g_T(x)\upsilon$$

(IV.65)

where

$$f_T(x) = \begin{bmatrix} S_T(q)u \\ 0 \end{bmatrix}; \quad g_T(x) = \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix}$$

(IV.66)

thus

$$\dot{x}_T = \begin{bmatrix} S_T(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix} \upsilon$$

(IV.67)

where $f_T(x)$ is the drift vector, $g_T(x)$ is the input matrix and $I_{r_M} \in R^{r_M \times r_M}$ is a identity matrix.

IV.7.1 The posture dynamical state space representation

The posture dynamical state space representation is a subsystem of the configuration dynamical state space representation (IV.64), which only takes into the account the posture ξ and the centred angular $\dot{\alpha}_s$ generalized velocities of the system model. It can be expressed as an extension of the posture kinematic state space representation, models (IV.36) and (IV.38), by adding to them the dynamic extension $\dot{u} = \upsilon$, as follows:

$$\begin{aligned} \dot{q}_p &= S_{q_p}(q)u \\ \dot{u} &= \upsilon \end{aligned}$$

(IV.68)

When $N_c \neq 0$, the input vector is $u = [\eta \quad \zeta]^T$ and the remaining generalized velocities are (IV.37). When $N_c = 0$, the input vector is $u = \eta$ and the remaining generalized velocities are (IV.39). Then, by adding the steering variables u to the generalized coordinates of the system, and defining the new generalized posture coordinates as $x_p = [q_p \quad u]^T$, the system (IV.68) can be expressed in the form:

$$\dot{x}_p = f_p(x_p) + g_p(x_p)\upsilon$$

(IV.69)

where

$$f_p(x_p)=\begin{bmatrix} S_p(q)u \\ 0 \end{bmatrix}; \qquad g_p(x_p)=\begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix}$$

(IV.70)

thus

$$\dot{x}_p=\begin{bmatrix} S_p(q)u \\ 0 \end{bmatrix}+\begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix}\upsilon$$

(IV.71)

where $f_p(x_p)$ is the drift vector, $g_p(x_p)$ is the input matrix and $I_{r_M} \in R^{r_M \times r_M}$ is a identity matrix.

IV.7.2 Some robot posture dynamic control properties

The properties for the posture dynamical state space representation, inherited from the kinematic state space representation, can be listed as follows [Campion et al. (1996)]:

- It is generic and irreducible.
- It is Small-Time-Locally Controllable.
- For restricted mobility robots, it is not stabilizable by continuous static time-invariant state feedback, but is stabilizable by a time-varying static state feedback.
- The dimension of the largest feedback linearizable subsystem by static state feedback is $2r_M$. Omnidirectional robots are therefore state feedback linearizable.
- It is a differentially flat system.
- It is partially input-output state feedback linearizable, with relative degree 2 when a valid choice of output equations $y=h(q_p)$ is proposed [d'Andréa-Novel et al. (1995)].
- The partially linearized input-output map of restricted mobility robots is stabilizable by smooth static state feedback [d'Andréa-Novel et al. (1995)].

IV.8 Input-output static state feedback linearization

The system (IV.68) for restricted mobility robots is a differentially flat system [Campion et al. (1996)]. Therefore, there exist a set of outputs y (flat outputs) such that the state and the input can be expressed algebraically in terms of y and a certain number of its derivates [Fliess et al. (1995)]. The system (IV.69) and the output vector can be expressed as follows:

$$\begin{aligned} \dot{x}_p &= f_p(x_p) + g_p(x_p)v \\ y &= h(q_p) \end{aligned}$$

(IV.72)

where the output equations $y = h(q_p)$ [7] are functions of posture state variables ξ and the angular coordinates α_s only, but not on the states η and ζ. Since, the number of the degrees of freedom of the system that can be controlled is r_M, therefore, one may have at most r_M independent position outputs equations, as a consequence of the Brockett obstruction [Brockett (1983)]. On the other hand, the dimension of the largest feedback linearizable subsystem by static state feedback is $2r_M$ [Campion et al. (1996)]. Thus, the system (IV.72) can be generically transformed by state feedback and diffeomorphism into the largest controllable linear subsystem, of dimension $2r_M$ with a remaining nonlinear subsystem of dimension $3 - r_m$ in the following form [d'Andréa-Novel et al. (1995)]:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= w \\ \dot{z}_3 &= Q_z(z) z_2 \end{aligned}$$

(IV.73)

---

[7] A set of suitable output equations $z_1 = h(\xi, \alpha_s)$ for each type of wheeled mobile robot can be found in d'Andréa-Novel, B., Campion, G. & Bastin, G. (1995).

where $z_1 \in R^{r_M}$, $z_2 \in R^{r_M}$ and $z_3 \in R^{3-r_m}$ are the new coordinates, and $w \in R^{r_M}$ is the acceleration control input at the steering space. The linearizing output vector can be defined as follows:

$$z_1 = y = h(q_p) = h(\xi, \alpha_s)$$

(IV.74)

Since, the relative degree for all the posture dynamical models of conventional wheeled mobile robots of restricted mobility is 2 [d'Andréa-Novel et al. (1995)], the following nonlinear change of coordinates $z = \Phi(x_p)$ can be defined:

$$z = \Phi(x_p) \rightarrow \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} \phi_1(x_p) \\ \phi_2(x_p) \\ \phi_3(x_p) \end{pmatrix} = \begin{pmatrix} h(q_p) \\ A(q_p)u \\ t(q_p) \end{pmatrix}$$

(IV.75)

where $z_2 = \dot{z}_1$, and $z_3 = t(q_p)$ is chosen such that $z = \Phi(x_p)$ is a *diffeomorphism*, i.e. smooth transformation with invertible Jacobian matrix $\left|\partial\Phi(x_p)/\partial x_p\right| \neq 0$, which ensure the inverse operation:

$$x_p = \Phi^{-1}(z) \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \phi_1(z) \\ \phi_2(z) \\ \phi_3(z) \end{pmatrix}$$

(IV.76)

that guarantee the necessary and sufficient condition for input-output linearization (i.e., the decoupling matrix $A(q_p)$ has full rank [Nijmeijer & van der Schaft (1990) and Isidori (1995)]), and:

$$\frac{\partial}{\partial u} t(q_p) = 0$$

(IV.77)

to avoid input dependence of the remaining nonlinear part. The description of the input-output system, expressed in *new coordinates* $z_i = \Phi_i(x_p)$, $1 \le i \le 3 + 2r_M - r_m$, is found by differentiating (IV.75) and substituting (IV.69) as follows:

$$\dot{z} = \frac{\partial \Phi(x_p)}{\partial x_p} \dot{x}_p = \frac{\partial \Phi(x_p)}{\partial x_p} \left[f_p(x_p) + g_p(x_p) \upsilon \right]$$

(IV.78)

it yields:

$$\dot{z}_1 = z_2 = \frac{\partial h}{\partial x_p} \dot{x}_p = A(q_p) u$$

(IV.79)

$$\dot{z}_2 = B(x_p) u + A(q_p) \upsilon$$

(IV.80)

$$\dot{z}_3 = \frac{\partial t(q_p)}{\partial x_p} \dot{x}_p = Q(q_p) u$$

(IV.81)

where $u \in R^{r_M}$, $\upsilon \in R^{r_M}$, $B(x_p) \in R^{r_M \times r_M}$, $Q(q_p)u \in R^{3-r_m}$ and $A(q_p) \in R^{r_M \times r_M}$ is a full rank decoupling matrix defined as:

$$A(q_p) = \left(\frac{\partial h}{\partial \xi} R^T(\theta) \Sigma(\alpha_s) \quad \frac{\partial h}{\partial \alpha_s} \right)$$

(IV.82)

and

$$B(x_p) = \left(\frac{\partial}{\partial \xi} [A(q_p)u] \cdot R^T(\theta) \Sigma_\xi(\alpha_s) \quad \frac{\partial}{\partial \alpha_s} [A(q_p)u] \right)$$

(IV.83)

$$Q(q_p) = \left(\frac{\partial t}{\partial \xi} R^T(\theta) \Sigma_\xi(\alpha_s) \quad \frac{\partial t}{\partial \alpha_s} \right)$$

(IV.84)

Proof: The largest linearizable subsystem is obtained by twice differentiating (IV.74) and substituting (IV.69), as follows:

- Differentiating z_1 yields:

$$\dot{z}_1 = \frac{\partial h}{\partial x_p}\dot{x}_p = \frac{\partial h}{\partial x_p}\left[\begin{bmatrix} S_p(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix}\upsilon\right] \tag{IV.85}$$

substituting (IV.36) and rearranging the elements, it yiels:

$$\dot{z}_1 = \begin{pmatrix} \frac{\partial z_1}{\partial \xi} & \frac{\partial z_1}{\partial \alpha_s} & \frac{\partial z_1}{\partial u} \end{pmatrix} \begin{bmatrix} R^T(\theta)\Sigma_\xi(\alpha_s) & 0 & 0 \\ 0 & I_{r_s} & 0 \\ 0 & 0 & I_{r_M} \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \\ \upsilon \end{bmatrix} \tag{IV.86}$$

$$\dot{z}_1 = \begin{pmatrix} \frac{\partial h}{\partial \xi} R^T(\theta)\Sigma_\xi(\alpha_s) & \frac{\partial h}{\partial \alpha_s} & 0 \end{pmatrix} \begin{bmatrix} \eta \\ \zeta \\ \upsilon \end{bmatrix} \tag{IV.87}$$

$$\dot{z}_1 = \begin{pmatrix} \frac{\partial h}{\partial \xi} R^T(\theta)\Sigma_\xi(\alpha_s) & \frac{\partial h}{\partial \alpha_s} \end{pmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \tag{IV.88}$$

$$\dot{z}_1 = z_2 = A(q_p)u \tag{IV.89}$$

- Differentiating z_2 yields:

$$\dot{z}_2 = \frac{\partial z_2}{\partial x_p}\left[\begin{bmatrix} S_p(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix}\upsilon\right] \tag{IV.90}$$

substituting (IV.36) and rearranging the elements, it yiels:

$$\dot{z}_2 = \begin{pmatrix} \frac{\partial z_2}{\partial \xi} & \frac{\partial z_2}{\partial \alpha_s} & \frac{\partial z_2}{\partial u} \end{pmatrix} \begin{bmatrix} R^T(\theta)\Sigma_\xi(\alpha_s) & 0 & 0 \\ 0 & I_{r_s} & 0 \\ 0 & 0 & I_{r_M} \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \\ \upsilon \end{bmatrix} \tag{IV.91}$$

$$\dot{z}_2 = \begin{pmatrix} \frac{\partial z_2}{\partial \xi} R^T(\theta)\Sigma_\xi(\alpha_s) & \frac{\partial z_2}{\partial \alpha_s} & \frac{\partial z_2}{\partial u} \end{pmatrix} \begin{bmatrix} \eta \\ \zeta \\ \upsilon \end{bmatrix}$$

(IV.92)

$$\dot{z}_2 = \frac{\partial z_2}{\partial \xi} \cdot R^T(\theta)\Sigma_\xi(\alpha_s)\eta + \frac{\partial z_2}{\partial \alpha_s}\zeta + \frac{\partial z_2}{\partial u}\upsilon$$

(IV.93)

$$\dot{z}_2 = \frac{\partial}{\partial \xi}[A(q_p)u] \cdot R^T(\theta)\Sigma_\xi(\alpha_s)\eta + \frac{\partial}{\partial \alpha_s}[A(q_p)u]\zeta + \frac{\partial}{\partial u}[A(q_\xi)u]\upsilon$$

(IV.94)

$$\dot{z}_2 = \frac{\partial}{\partial \xi}[A(q_p)u] \cdot R^T(\theta)\Sigma_\xi(\alpha_s)\eta + \frac{\partial}{\partial \alpha_s}[A(q_p)u]\zeta + A(q_\xi)\upsilon$$

(IV.95)

$$\dot{z}_2 = \left(\frac{\partial}{\partial \xi}[A(q_p)u] \cdot R^T(\theta)\Sigma_\xi(\alpha_s) \quad \frac{\partial}{\partial \alpha_s}[A(q_p)u] \right) u + A(q_\xi)\upsilon$$

(IV.96)

$$\dot{z}_2 = B(x_p)u + A(q_p)\upsilon$$

(IV.97)

- To ensure a diffeomorphism transformation (see equation (IV.77)), z_3 is chosen as $z_3 = t(q_p)$. Differentiating z_3:

$$\dot{z}_3 = \frac{\partial t(q_p)}{\partial x_p}\dot{x}_p = \frac{\partial k}{\partial x_p}\left[\begin{bmatrix} S_p(q_p)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix}\upsilon \right]$$

(IV.98)

substituting (IV.36) and rearranging the elements, it yiels:

$$\dot{z}_3 = \begin{pmatrix} \frac{\partial t}{\partial \xi} & \frac{\partial t}{\partial \alpha_s} & \frac{\partial t}{\partial u} \end{pmatrix} \begin{bmatrix} R^T(\theta)\Sigma_\xi(\alpha_s) & 0 & 0 \\ 0 & I_{r_s} & 0 \\ 0 & 0 & I_{r_M} \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \\ \upsilon \end{bmatrix}$$

(IV.99)

$$\dot{z}_3 = \begin{pmatrix} \frac{\partial t}{\partial \xi} R^T(\theta)\Sigma_\xi(\alpha_s) & \frac{\partial t}{\partial \alpha_s} & 0 \end{pmatrix} \begin{bmatrix} \eta \\ \zeta \\ \upsilon \end{bmatrix}$$

(IV.100)

$$\dot{z}_3 = \begin{pmatrix} \frac{\partial t}{\partial \xi} R^T(\theta)\Sigma_\xi(\alpha_s) & \frac{\partial t}{\partial \alpha_s} \end{pmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(IV.101)

$$\dot{z}_3 = Q(q_p)u$$

(IV.102)

This completes the proof. □

When the system (IV.68) does not have centred steerable wheels the system (IV.81) has the following components:

$$A(q_p) = \left(\frac{\partial h}{\partial \xi} S_p(q) \right)$$

(IV.103)

$$B(q_p) = \frac{\partial}{\partial \xi}[A(q_p)u] \cdot S_p(q)$$

(IV.104)

$$Q(q_p) = \left(\frac{\partial t}{\partial \xi} S_p(q) \right)$$

(IV.105)

On the other hand, by substituting (IV.76) in (IV.78):

$$\dot{z}(z) = \left| \frac{\partial \Phi(x_p)}{\partial x_p} \dot{x}_p \right|_{x_p = \Phi^{-1}(z)} = \left| \frac{\partial \Phi(x_p)}{\partial x_p} [f_p(x_p) + g_p(x_p)\upsilon] \right|_{x_p = \Phi^{-1}(z)}$$

(IV.106)

the system (IV.78) in the new coordinates can be written as :

$$\begin{aligned}\dot{z}_1 &= z_2\\ \dot{z}_2 &= B_z(z) + A_z(z)\upsilon\\ \dot{z}_3 &= Q_z(z)z_2\end{aligned}$$

(IV.107)

where $B_z(z) = \left|B(x_p)u\right|_{x_p=\Phi^{-1}(z)}$, $A_z(z) = \left|A(q_p)\right|_{x_p=\Phi^{-1}(z)}$, and $Q_z(z)z_2 = \left|Q(q_p)u\right|_{x_p=\Phi^{-1}(z)}$ are expressed in the new coordinates. Since, the decoupling matrix $A_z(z)$ is nonsingular, the *input-output decoupling* can be achieved by applying to (IV.107) the following *static state feedback* control law:

$$\upsilon = A_z^{-1}(z)\left[w - B_z(z)\right]$$

(IV.108)

expressed in the original coordinates:

$$\upsilon = A^{-1}(q_p)\left[w - B(x_p)\right]$$

(IV.109)

the expected partially linearized subsystem (IV.73) can be obtained, where the input-output response is both linear and decoupled, where $w \in R^m$ is the input vector of the linear part of the system in new coordinates, with the observable part:

$$y = z_1; \qquad \dot{z}_1 = z_2; \qquad \dot{z}_2 = w$$

(IV.110)

with unobservable zero dynamics (obtained by substituting $z_1 = 0$ and $z_2 = 0$) :

$$\dot{z}_3 = Q_z(z)z_2 = 0$$

(IV.111)

which is stable but not asymptotically stable, with internal dynamics exhibiting unstable behaviour when moving backwards [Yun & Sarkar (1998) and Bolzern et al. (2001)].

IV.9 Stabilization of the input-output linearized map

Since, the map between the transformed input w and the output y is exactly linear, the input w can then be used to freely assign the dynamics of z_1, as the dynamics of a

second-order stable linear system. More precisely, let z_{1ref}, $\dot{z}_{1ref}$, and $\ddot{z}_{1ref}$ be a smooth reference trajectory such that $\|z_{1ref}(t)\|$, $\|\dot{z}_{1ref}(t)\|$, and $\|\ddot{z}_{1ref}(t)\|$ are bounded for every t and $\int_0^{\infty}|\dot{z}_{1ref}(t)|dt < +\infty$, and being the matrix $Q_z(z)$ bounded for every z_1 and z_3 by assumption, then the following auxiliary control law can be applied:

$$w = \ddot{z}_{1ref} - (\Lambda_1 + \Lambda_2)\dot{\tilde{z}}_1 - \Lambda_1\Lambda_2\tilde{z}_1 \tag{IV.112}$$

where $\tilde{z}_1 = z_1 - z_{1ref}$, Λ_1 and Λ_2 are arbitrary positive diagonal $r_M \times r_M$ matrices, generically ensures that: $\tilde{z}_1$ and $\dot{\tilde{z}}_1$ exponentially converge to zero and $z_3(t)$ is bounded for every t.

IV.10 Summary

The complete kinematic and dynamic modelling of conventional wheeled mobile robots have introduced a new set of holonomic parametric functions in the configuration kinematic state space model, able to coordinate the orientation of the centred steerable wheels of robots with multiple centred steerable wheels, and to compute the evolution of the generalized velocities of the robot. The holonomic parametric functions can be eliminated from the robot state space representation for analysing the controllability and stabilizability of the robot on the plane. On the other hand, the dynamic equation of motion can be used to compute the torques needed to produce the corresponding accelerations. The complete kinematics and dynamic modelling allows the determination of the correct motor configuration and the computation of the motor design parameters.

Since, the modelling of robots are based on Cartesian reference frames, with a minimum number of modelling variables, the resulting models can be compared with those of many other authors. This modelling can be used to design modular wheeled mobile robots, since they include the whole representation of the system. The smooth static state feedback stabilization characteristic of the partially input-output feedback linearized model, of the restricted mobility robots and their stable zero dynamics, with asymptotic stable zero dynamics when moving forwards, can be used to design trajectories to stabilize the posture of the robot.

In this chapter an original and general methodology has been presented, which is useful to analyze the controllability and stabilizability of the posture of conventional wheeled mobile robots, based on the complete representation of the kinematic and dynamic models and their differentially flatness property using input-output static state feedback linearization techniques.

V Moving mail trolleys with wheeled mobile robots: Selecting and modelling particular cases

V.1 Overview

Only five types of nondegenerated conventional wheeled mobile robots can be obtained from the configuration of their conventional wheels (see Chapter IV). The analysis carried out to determine the feasible type of single-robot and robot-trolley configuration to be studied and their modelling is presented in this chapter. The selection of the best configurations was based on the analysis of the kinematic properties (high degree of mobility), feasible wheel motorization (no singular configurations), and less space to navigate (symmetric configurations). One single-robot and one robot-trolley configuration were selected for solving the problem. The kinematic, state space representation, controllability, degree of holonomy and nonholonomy, dynamic modelling, input static state feedback linearization, input-output static state feedback linearization, and stabilization of the linearized input-output map of the single-robot and the robot-trolley configuration are presented in this chapter, as well. The contents of the chapter are arranged as follows:

V.2 The York trolley

The Royal Mail York Trolley, York Roll Container [Envosort (2004)], is a passive wheeled U-frame cage that can transport 28 Royal Mail letter trays (see Figure V.1), also it is useful for all sorts of other heavyweight transportation and distribution duties in offices, postrooms, hospitals, warehouses and retail environments. It can be nested for space-saving storage.

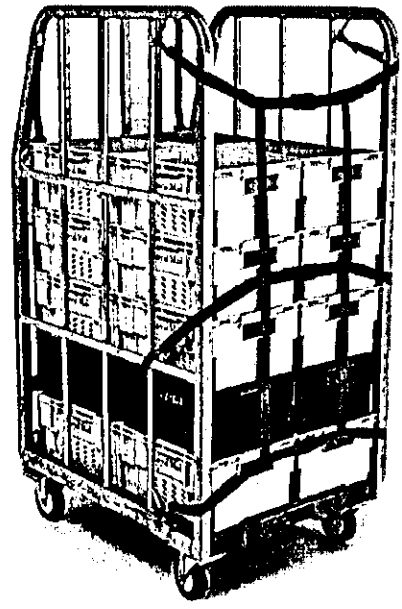


Figure V.1: York trolley

It has 125mm rubber wheels (2 fixed, 2 swivel), galvanised finish protects against bumps, maintenance-free self-adjusting parking brake operated by hand, heavy-duty polymer base, snap-fit nylon webbing to secure high loads, with:

- **Load capacity**: 500kg
- **External dimensions**: 670mm (w) x 850mm (l) x 1500mm (h)
- **Internal dimensions**: 606mm (w) x 800mm (l) x 1286mm (h)
- **Weight**: 50kg

V.3 The mail trolley held by a wheeled mobile robot

Another way of moving wheeled loads, different to that reported so far in the literature (see sections I.3, I.4, II.4.3 and II.5), is incorporating the wheeled load to the wheeled mobile robot without lifting or towing, just by holding it firmly in two points and pushing or pulling it (applying stable pushes, Lynch & Mason (1996)). In this case, the kinematics, dynamics, and size of the robot change depending on the wheeled load kinematics, weight, size, and held side. As a consequence, the robot has to take into account these changes and make the necessary adjustments to support the new configuration. Therefore, in order to select the correct robot configuration to solve the problem, the feasible robot-trolley configurations are analyzed in the next section. Afterwards, in the section V.4 the right robot platform wheel configuration is selected.

V.3.1 Feasible robot-trolley configurations

In this section we are interested in finding the feasible robot-trolley symmetric configurations with a high degree of mobility, associated to the side that the wheeled load is held and the type of robot they produce, which meet the following criteria:

1. Our study is restricted to moving mail York trolleys: Pushing and pulling tasks have replaced lifting, carrying, and lowering activities in order to prevent low back disorders [Kingma et al. (2003)], i.e., four-wheeled containers for mail transferring [e.g., van der Beek et al. (2000)]. Thus, the use of the York trolley is irreplaceable (see section I.2.2), because this type of wheeled load is useful for pushing medium loads by hand in a safer way.

2. For simplicity, the robot will use only one side to hold the load.

3. Configurations with a low degree of manoeuvrability will be discharged, because they require more space to manoeuvre.

4. Asymmetric configurations will be discharged in order to have configurations with low turning radius, more configuration space to navigate, simpler model representations, and better dynamic behaviour.

The Figure V.2 displays the twenty symmetric feasible ways of holding a trolley by the five types of wheeled mobile robots, which hold the ***first and second criteria***. The trolley has been rotated 90^0 each time in order to show the four possible ways of holding it.

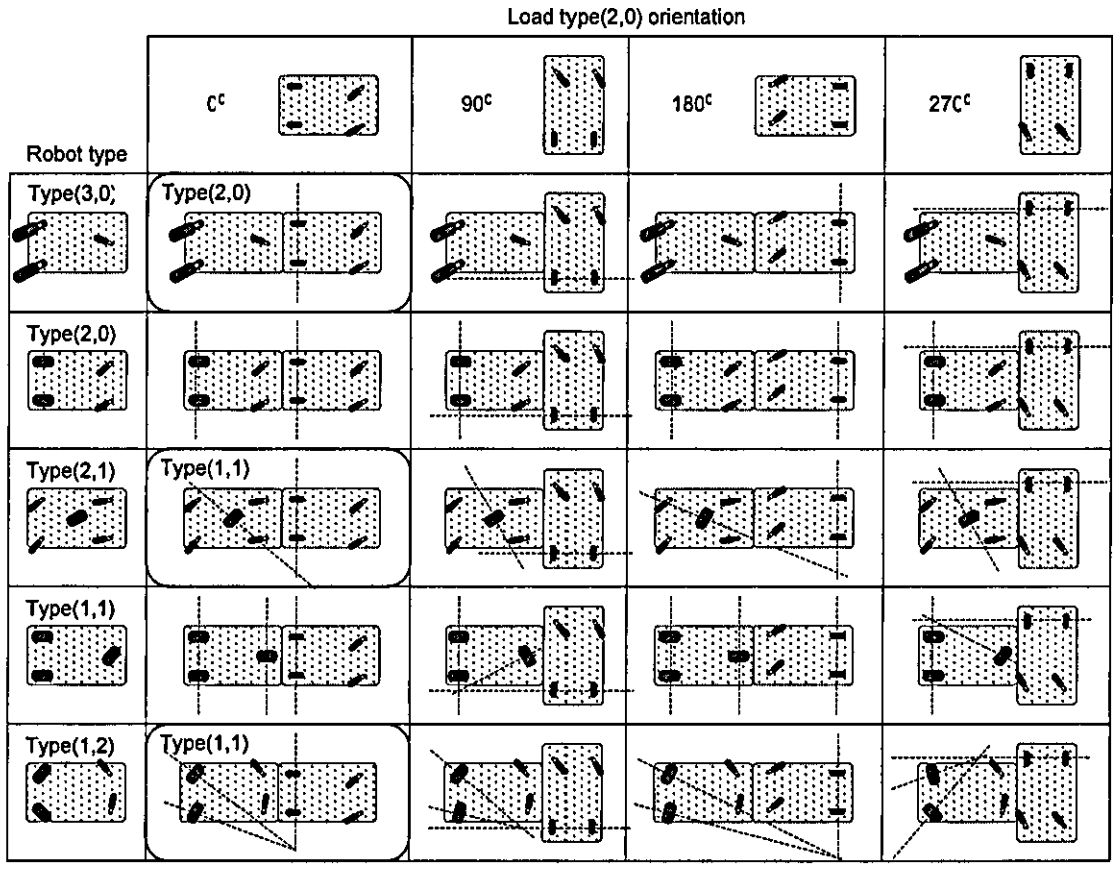


Figure V.2: Robot-trolley feasible configurations

One thing that can be observed from Figure V.2 is that whichever the combination, the robot-trolley configuration is a degraded type of the original robot type, which means a loss of degree of mobility, as a consequence, loss of degrees of freedom that can be controlled. This is a big difference with respect to mobile arm manipulators.

In order to eliminate configurations with a low degree of manoeuvrability, ***third criterion***, we can start applying the definition of nondegenerate mobile robot, see section IV.4.2.a.1.4 [Campion et al. (1996)]:

- Multiple conventional fixed wheels must be placed along a single common axle.
- The centres of the centred steerable wheels must not be placed on the common axle of the fixed wheels.
- If a mobile robot is equipped with more centred steerable wheels than their degree of steerability, the orientation of extra centred steerable wheels must be coordinated to guarantee the existence of the instantaneous centre of rotation (ICR) at each time instant.

The application of the definition to the Figure V.2 can be explained using dashed lines (imaginary infinite long wheel axles), which represent the feasible location of the ICR associated to each wheel, with the following interpretations:

- **Single dashed line**: The ICR can be placed anywhere on the dashed line.
- **Parallel dashed lines**: The configuration can only move perpendicularly to the dashed lines.
- **Convergent dashed lines**: The configuration can only rotate around the ICR.
- The dashed line is fixed if the wheel is fixed.
- The dashed line can rotate along with the wheel and around the wheel vertical axle, if the wheel is centred.
- Multiple dashed lines must converge into the ICR, otherwise should be parallel to be accorded with the general planar motion of rigid bodies.

Therefore, one can conclude that the robot-trolley configurations produced by the robots type (2,0) and (1,1) do not meet the definition of nondegenerate mobile robot. Thus, they can be eliminated from the analysis.

Applying the asymmetric criterion, ***forth criterion***, the configurations produced by the rotation of the trolley by an angle of 90^0, 180^0 and 270^0 can be eliminated for the analysis, because they are asymmetric configurations.

Therefore, the feasible single-robots and robot-trolley configuration can be reduced to three: the configurations produced by the robots *type*(3,0), *type*(2,1), and *type*(1,2) with the 0^0 rotated trolley. In addition, some conclusions from the analysis of the motorization can be done (see section IV.6.5), as follows:

- The configuration robot *type*(3,0) and trolley produce a robot *type*(2,0), which has a very good motorization if the fixed wheels of the trolley are motorized. Nothing else can be concluded if any other combination is chosen when excluding the fixed wheels, because the input transformation matrix has singular inverse or

singular Moore-Penrose matrix inverse. Since, the trolley is unmotorized this configuration can be eliminated.

- The configuration robot *type*(2,1) and trolley produce a robot *type*(1,1), which has a very good motorization if the centred steerable wheel is motorized for both being steered and being rotated, which is exactly the case.

- The configuration robot *type*(1,2) and trolley produce a robot *type*(1,1), which has a very good motorization if the two centred steerable wheels are motorized to be steered and at least one is motorized to be rotated. The orientation of the centred steerable wheels should be coordinated to meet the planar motion of rigid bodies. This configuration meets the motorization criteria.

Thus, in terms of robot-trolley configuration two configurations are useful:

1) Configuration robot *type*(2,1) and trolley that produces a robot *type*(1,1).

2) Configuration robot *type*(1,2) and trolley that produces a robot *type*(1,1).

V.4 Robot platform selection

In this section we are interested in selecting the robot platform with the best characteristics between the two mentioned above. Once again, the motorization configuration analysis can be used to determine the best robot-trolley configuration, as follows:

- Robot *type*(2,1): This kind of robot has the characteristic that, in general terms, the posture of the robot is full stabilizable, as long as the wheel is motorized for both being steered and being rotated, and if an external torque is applied to control the orientation of the robot platform, which is an irregular situation. Given that, the orientation of the robot is determined for applied external torques, it makes this robot configuration unfeasible.

- Robot *type*(1,2): This kind of robot has the characteristic that the posture of the robot is partially stabilizable, as long as both wheels are motorized for being steered and rotated. In general, this robot configuration can be kinematically modelled and its posture can be partially stabilizable if a non-singular

codistribution is obtained form the kinematic constraints [Campion et al. (1996)], and if a specific selection of output equations are proposed [d'Andréa-Novel et al. (1995)].

The Robot *type*(1,2) seams to be good for solving the problem, but, in general terms this is a multi-type robot which can be reconfigured to produce equivalent behaviours to the other robots of restricted mobility. This is a flexible robot platform, whose behaviours should be studied.

V.5 Defining the particular cases

In this section is analysed the single-robot and robot-trolley configuration in order to select the particular cases to be studied. The analysis is restricted to the following cases:

- The single-robot *type*(1,2).
- The robot-trolley configuration, robot *type*(1,1), produced by the union of the robot *type*(1,2) and the trolley.

In the first case, the single-robot *type*(1,2) is studied to understand the multi-type behaviour of this configuration, then one behaviour is selected to be used in the solution of the problem. In the second case, the robot-trolley configuration *type*(1,1) is studied to solve the problem of coordinating multiple wheels by including an additional virtual centred steerable wheel in the analysis.

V.5.1 The single-robot: Robot type(1,2)

The single-robot type selected (see Figure V.3) is a robot *type*(1,2) that has two centred steerable wheels.

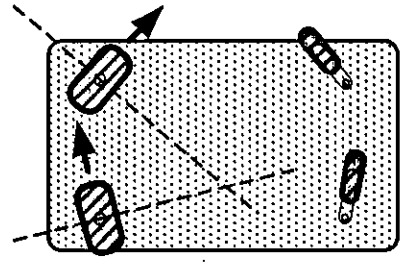


Figure V.3: Robot type(1,2)

This type of mobile robot has multiple behaviours depending on the angular position of their two centred steerable wheels and the way they are coordinated. The different kinds of behaviours can be seen in the Figure V.4. They can be obtained when:

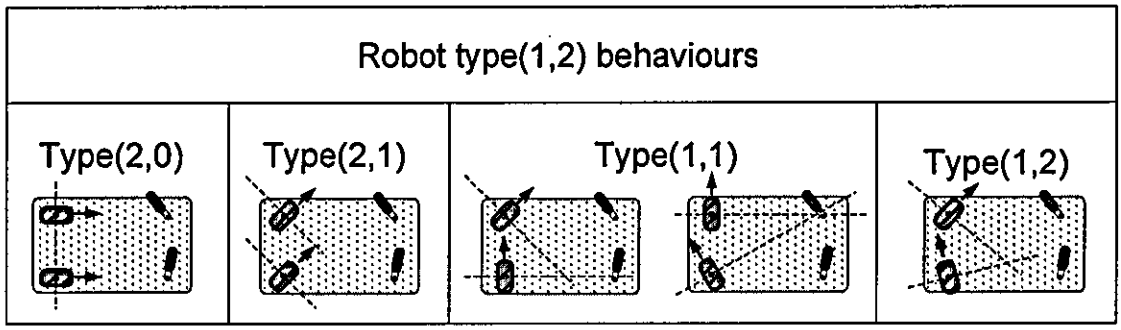


Figure V.4: Robot type(1.2) behaviours

- Robot *type*(2,0) : the centred steerable wheels are positioned such that they seem to share a single common axle and behave as fixed wheels.

- Robot *type*(2,1) : the centred steerable wheels are coordinated such that they always have the same orientation angle and behave as centred steerable wheels.

- Robot *type*(1,1): one centred steerable wheel is positioned such as is heading to the vertical axle of the other centred steerable wheel and behaves as fixed wheel. The other wheel must behave as centre steerable wheel, but it must avoid the singularity produced when the centred steerable wheel imaginary axle is positioned on the vertical axle of the fixed wheel.

- Robot *type*(1,2) : the centred steerable wheels are coordinated such that they can have any orientation, but it must be taken into account that this configuration can meet the singularities that produce the other above mentioned behaviours and degenerate mobile robot configurations.

All these behaviours have their particular kinematic and dynamic properties. But, following a similar criteria used in section V.3.1 to select the best behaviour, it can be concluded that: the robot *type*(1,1) is an asymmetric configuration, robot *type*(1,2) has several singularities, and the robot *type*(2,1) is a kind of robot that never changes the orientation of its platform, thus they can be eliminated. Therefore, for simplicity, the

single-robot *type*(1,2) can be configured to have an e*quivalent behaviour of robot *type*(2,0).

V.5.2 The robot-trolley configuration: Robot type(1,1)

The robot-trolley configuration type selected (see Figure V.5) is a robot *type*(1,1) that has two rear full motorized centred steerable wheels, two passive fixed wheels, four passive off-centred wheels, and one virtual centred steerable wheel for coordinating the motorized centred steerable wheels and the movements when going forwards and backwards.

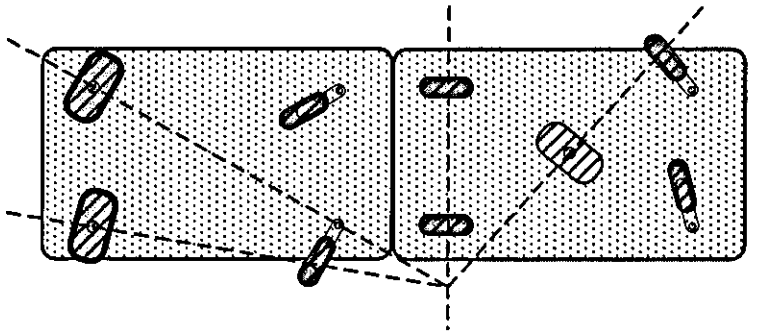


Figure V.5: Robot-trolley configuration type

This type of mobile robot has two behaviours depending on the angular position of their two centred steered wheels, as follows (see Figure V.6):

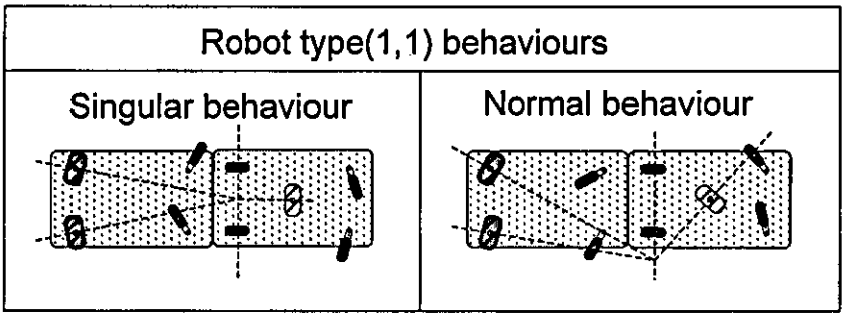


Figure V.6: Robot type(1,1) behaviours

- Singular behaviour: when the imaginary axles of the coordinated centred steerable wheels meet the middle of the imaginary common axle between the two fixed wheels.

- Normal behaviour: Any other configuration when the imaginary axles of the coordinated centred steerable wheels meet any point along the imaginary common axle of the two fixed wheels, excluding the explained for the singular behaviour.

These two behaviours have their particular kinematic and dynamic properties. So, a virtual centred steerable wheel can be introduced in order to coordinate the orientation of the centred steerable wheels and to control the forwards and backwards movements of the robot. In addition, in order to avoid the modelling of singular configurations, one can control the behaviour of the robot by applying an input-output state feedback linearization procedure using a particular output equation.

V.6 Modelling particular cases

The modelling of the kinematics, dynamics, state space representation, and input and input-output static state feedback linearization of the selected single-robot and robot-trolley configurations is presented in this section. The modelling is restricted to the following types of robots:

- The single-robot configuration, robot *type*(1,2), with equivalent behaviour of robot *type*(2,0).

- The robot-trolley configuration, robot *type*(1,1), produced by the union of the robot *type*(1,2) and the trolley, with an additional virtual centred steerable wheel to coordinate the orientation of the centred steerable wheels of the robot.

In the first case, the single-robot configuration, robot *type*(1,2), is modelled and analyzed as robot *type*(2,0). In the second case, the robot-trolley configuration, robot *type*(1,1), is modelled and analyzed in order to solve the problem of coordinating the orientation of multiple centred steerable wheels by including an additional virtual centred steerable wheel. The passive castor wheels, which are used to stabilize the platform, were ignored for the analysis for the following reasons: they are unmotorized, the dynamic effects are negligible, and the kinematic effects are null.

V.6.1 Robot type(2,0)

The proposed platform configuration, robot *type*(2,0), has two rear motorized centred steerable wheels (see Figure V.7), which played the role of fixed wheels (i.e. $\alpha = 0$, see chapter V), and two front passive castor wheels, which were ignored for the analysis.

V.6.1.a Kinematic modelling of the robot type (2,0)

Following the methodology presented in chapter IV, the five n_T configuration coordinates of the robot are described by:

$$q=[x \quad y \quad \theta \quad \phi_r \quad \phi_l]^T \tag{V.1}$$

where x and y denote the position of the point P, and θ the orientation of the robot reference frame $\{x_1, y_1\}$ with respect to the fixed world inertial reference frame (see Figure V.7). Also, the variables ϕ_r and ϕ_l denote the rotation angles of the right and left wheels around its horizontal axle respectively. The radius of the wheel is denoted by r. Assuming perfect rolling (without both lateral and rolling slipping), $\alpha=0$, and substituting (IV.11) in (IV.13) and (IV.14), the total of four m_{Tot} bilateral kinematic constraints $C_T(q,\dot{q})=0$ at the robot reference frame can be written as:

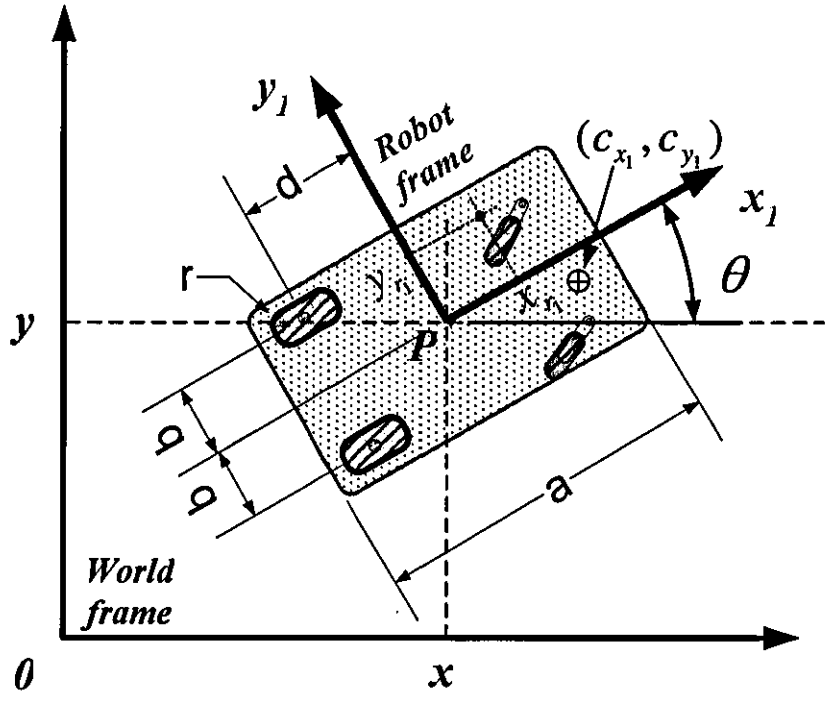


Figure V.7: Posture of the robot type(2,0)

- Kinematic constraints orthogonal to the wheels' plane (no lateral slipping):

$$\begin{aligned} -\sin(\theta)\dot{x}+\cos(\theta)\dot{y}+d\dot{\theta}=0 \\ -\sin(\theta)\dot{x}+\cos(\theta)\dot{y}+d\dot{\theta}=0 \end{aligned}$$

(V.2)

- Kinematic constraints along the wheels' plane (rolling without slipping):

$$\begin{aligned} \cos(\theta)\dot{x}+\sin(\theta)\dot{y}+b\dot{\theta}-r\dot{\phi}_r=0 \\ \cos(\theta)\dot{x}+\sin(\theta)\dot{y}-b\dot{\theta}-r\dot{\phi}_l=0 \end{aligned}$$

(V.3)

expressed in the form $A_T(q)\dot{q}=0$:

$$\begin{bmatrix} -\sin(\theta) & \cos(\theta) & d & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & d & 0 & 0 \\ \cos(\theta) & \sin(\theta) & b & -r & 0 \\ \cos(\theta) & \sin(\theta) & -b & 0 & -r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = 0$$

(V.4)

In order to find the robot internal kinematics $A_{T1}(q)\dot{q}_1=0$, let set $\theta=0$ in (V.4). So, the internal kinematic constraints can be expressed as:

$$\begin{bmatrix} 0 & 1 & d & 0 & 0 \\ 0 & 1 & d & 0 & 0 \\ 1 & 0 & b & -r & 0 \\ 1 & 0 & -b & 0 & -r \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = 0$$

(V.5)

Therefore, the kinematic motion properties of the robot can be summarised as follows:

- $r_m = 3 - rank(O_{fc}) = 3-1=2$
- $r_s = rank(O_c) = 0$
- $r_M = r_m + r_s = 2$

which define a robot *type*(2,0).

A controllable kinematic state space representation of the robot is found by computing a smooth codistribution of the matrix A_{T1}. So, let $N(A_{T1})$ be the null space of A_{T1}, $d_{q1} = \dim(A_{T1}) = 5$, $r_{q1} = rank(A_{T1}) = 3$, $r_m = d_{q1} - r_{q1} = 5 - 3 = 2$, and $r_s = 0$. Then, by spanning $N(A_{T1})$ it is possible to find a set of $r_m = 2$ smooth vector fields $\sigma_{m1}(q)$ and $\sigma_{m2}(q)$. If we let $\Sigma(q) \in R^{n_T \times r_M}$ be the matrix consisted of these vectors as $\Sigma(q) = [\sigma_{m1}(q) \quad \sigma_{m2}(q)]$ and selecting the $r_m = 2$ input velocities as $\eta = [\dot{x}_1 \quad \dot{\theta}]^T = [v \quad \omega]^T$, such as $\dot{q} = R_q^T \dot{q}_1 = R_q^T \Sigma(q)\eta$, where $\dot{x}_1 = v$ is the robot heading linear velocity, $\dot{\theta} = \omega$ is the robot angular velocity, and R_q^T is the transpose of the configuration rotation matrix. Thus, the resultant kinematic state space representation $\dot{q} = S_T(q)u$ of the robot can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = \begin{bmatrix} \cos(\theta) & d\sin(\theta) \\ \sin(\theta) & -d\cos(\theta) \\ 0 & 1 \\ 1/r & b/r \\ 1/r & -b/r \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{V.6}$$

where $S_T^T A_T^T = 0$ since $S_T \in null(A_T)$.

Let Δ_T be the distribution spanned by the vector fields of S_T:

$$\Delta_T = \left\{ \begin{bmatrix} \cos(\theta) & d\sin(\theta) \\ \sin(\theta) & -d\cos(\theta) \\ 0 & 1 \\ 1/r & b/r \\ 1/r & -b/r \end{bmatrix} \right\} \tag{V.7}$$

Let $\overline{\Delta}_T$ be the involutive closure of Δ_T obtained from the computation of the control Lie algebra (see equation (III.30)):

$$\overline{\Delta}_T = \begin{Bmatrix} \cos(\theta) & d\sin(\theta) & \sin(\theta) & \cos(\theta) \\ \sin(\theta) & -d\cos(\theta) & -\cos(\theta) & \sin(\theta) \\ 0 & 1 & 0 & 0 \\ 1/r & b/r & 0 & 0 \\ 1/r & -b/r & 0 & 0 \end{Bmatrix}$$

(V.8)

The following conclusions can be deduced:

- The distribution Δ_T is not involutive.
- The distribution Δ_T is reducible. The degree of holonomy $M_h = \dim(q) - \dim(\overline{\Delta}_T) = 5 - 4 = 1$.
- The degree of nonholonomy $M_{nonh} = \dim(\overline{\Delta}_T) - (r_m + r_s) = 4 - (2 + 0) = 2$.

The holonomic constraint can be found by integrating the equation resulted from the subtraction of the equations (V.3) as follows:

- Subtracting equations (V.3):

$$2b\dot{\theta} = r(\dot{\phi}_r - \dot{\phi}_l)$$

(V.9)

- Integrating equation (V.9):

$$\theta = \frac{r}{2b}(\phi_r - \phi_l) + \theta_0$$

(V.10)

where θ_0 is the initial condition of the robot orientation angle.

The posture kinematics state space representation and remaining generalized velocities can be written as follows:

- Posture kinematic state space representation $\dot{q}_p = S_{q_p} u$ (see equation (IV.38)):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & d\sin(\theta) \\ \sin(\theta) & -d\cos(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

(V.11)

- Remaining generalized velocities $\dot{q}_r = S_{q_r} u$ (see equation (IV.39)):

$$\begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = \begin{bmatrix} 1/r & b/r \\ 1/r & -b/r \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

(V.12)

Let Δ_p be the distribution spanned by the vector fields of S_{q_p}:

$$\Delta_p = \left\{ \begin{bmatrix} \cos(\theta) & d\sin(\theta) \\ \sin(\theta) & -d\cos(\theta) \\ 0 & 1 \end{bmatrix} \right\}$$

(V.13)

Let $\overline{\Delta}_p$ be the involutive closure of Δ_p obtained from the computation of the control Lie algebra (see equation (III.30)):

$$\overline{\Delta}_p = \left\{ \begin{bmatrix} \cos(\theta) & d\sin(\theta) & \sin(\theta) \\ \sin(\theta) & -d\cos(\theta) & -\cos(\theta) \\ 0 & 1 & 0 \end{bmatrix} \right\}$$

(V.14)

The following conclusions can be done:

- The distribution Δ_p is not involutive.

- The distribution Δ_p is irreducible and controllable. The degree of holonomy $M_h = \dim(q_p) - \dim(\overline{\Delta}_p) = 3 - 3 = 0$.

- The degree of nonholonomy $M_{nonh} = \dim(\overline{\Delta}_p) - (r_m + r_s) = 3 - (2 + 0) = 1$.

V.6.1.b Dynamic modelling of the robot type (2,0)

The dynamic model of the robot has been obtained using the Lagrange's formulation, as showed in the chapter IV. Thus, the dynamic equation of motion at the configuration space with nonholonomic constraints for a wheeled mobile robot system having five n_T-dimensional configuration space, with five n_T generalized coordinates $q=[x \quad y \quad \theta \quad \phi_r \quad \phi_l]^T$, subject to four m_{Tot} bilateral kinematic constraints, $r_m=2$ degree of mobility, $r_s=0$ degree of steerability, and $r_M=r_m+r_s=2+0=2$ degree of manoeuvrability, is represented by a set of five n_T second-order coupled and nonlinear differential equations relating to the positions, velocities, and accelerations of the robot posture and robot joints variables. They can be written in the form (see equation (IV.57)):

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}=P\tau+A_T^T(q)\lambda \tag{V.15}$$

The components of the equation (V.15) are defined by:

$$M(q)=\begin{bmatrix} m_T & 0 & A_M & 0 & 0 \\ 0 & m_T & B_M & 0 & 0 \\ A_M & B_M & I_c & 0 & 0 \\ 0 & 0 & 0 & I_{yyw} & 0 \\ 0 & 0 & 0 & 0 & I_{yyw} \end{bmatrix};\ C(q,\dot{q})\dot{q}=\begin{bmatrix} -B_M\dot{\theta}^2 \\ A_M\dot{\theta}^2 \\ 0 \\ 0 \\ 0 \end{bmatrix};\ P=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix};\ \tau=\begin{bmatrix} \tau_{\phi_r} \\ \tau_{\phi_l} \end{bmatrix};$$

$$A_T^T(q)=\begin{bmatrix} -\sin(\theta) & -\sin(\theta) & \cos(\theta) & \cos(\theta) \\ \cos(\theta) & \cos(\theta) & \sin(\theta) & \sin(\theta) \\ d & d & b & -b \\ 0 & 0 & -r & 0 \\ 0 & 0 & 0 & -r \end{bmatrix};\ \lambda=\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} \tag{V.16}$$

where

$$
\begin{aligned}
m_T &= m_c + 2m_w \\
A_M &= -[c_{x_1}\sin(\theta) + c_{y_1}\cos(\theta)]m_c - 2dm_w\sin(\theta) \\
B_M &= [c_{x_1}\cos(\theta) - c_{y_1}\sin(\theta)]m_c + 2dm_w\cos(\theta) \\
I_c &= 2(d^2+b^2)m_w + (c_{x_1}^2 + c_{y_1}^2)m_c + 2I_{zzw} + I_{zzc}
\end{aligned}
$$

(V.17)

where m_T is the total mass of the robot, m_c is the mass of the cart, m_w is the mass of the wheel, (c_{x_1}, c_{y_1}) is the coordinates of centre of mass with respect to the robot frame of reference, I_{zzw} is the moment of inertia of the wheel about z axis, I_{yyw} is the moment of inertia of the wheel about the y axis of the wheel reference frame, and I_{zzc} is the moment of inertia of the cart about z axis.

The properties of the dynamic equation can be summarised as:

Property 1. M is symmetric and positive definite.

Property 2. $\dot{M} - 2C$ is skew-symmetric.

Property 3. M and the norm of C are bounded.

V.6.1.c State space representation and input state feedback linearization

The dynamic equation of motion at the steering space can be written as follows (see equation (IV.59)):

$$
\overline{M}(q)\dot{u} + \overline{C}(q,\dot{q})u = \overline{P}\tau \equiv \overline{\tau}
$$

(V.18)

where

$$
\overline{M}(q) = \begin{bmatrix} m_T + 2I_{yyw}/r^2 & -c_{y_1}m_c \\ -c_{y_1}m_c & [(d-c_{x_1})^2 + c_{y_1}^2]m_c + 2m_w b^2 + 2I_{zzw} + I_{zzc} + 2b^2 I_{yyw}/r^2 \end{bmatrix};
$$

$$
\overline{C}(q,\dot{q})\dot{q} = \begin{bmatrix} (d-c_{x_1})m_c\omega^2 \\ -(d-c_{x_1})m_c v\omega \end{bmatrix};\ \overline{P} = \begin{bmatrix} 1/r & 1/r \\ b/r & -b/r \end{bmatrix};\ \overline{\tau} = \begin{bmatrix} \tau_{x_1} \\ \tau_\theta \end{bmatrix};\ u = \begin{bmatrix} v \\ \omega \end{bmatrix};\ \dot{u} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix}
$$

(V.19)

where τ_{x_1} is the force associated to the linear velocity of the single-robot along the x_1 axis, and τ_θ is the torque associated to the rotation of the single-robot platform.

Applying the following static nonlinear state feedback to the dynamic equation of motion (V.18):

$$\bar{\tau} = \bar{M}(q)\upsilon + \bar{C}(q,\dot{q})u$$

(V.20)

yields $\dot{u} = \upsilon$

and adding the configuration kinematic model $\dot{q} = S_T(q)u(t)$ (see equation (IV.34)) to the configuration dynamical state space representation can be written as:

$$\begin{aligned} \dot{q} &= S_T(q)u \\ \dot{u} &= \upsilon \end{aligned}$$

(V.21)

where $\dot{q} = S_T(q)u$ is represented by equation (V.6) and $\upsilon = \dot{u}$ as in equation (V.19). The irreducible posture dynamical state space representation can be written as follows (see equation (IV.68)):

$$\begin{aligned} \dot{q}_p &= S_{q_p}(q)u \\ \dot{u} &= \upsilon \end{aligned}$$

(V.22)

were $\dot{q}_p = S_{q_p}(q)u$ is represented by the equation (V.11) and and $\upsilon = \dot{u}$ as in equation (V.19), with remaining generalized velocities $\dot{q}_r = S_{q_r}(q)u$ (see equation (V.12)).

Defining the new generalized coordinates as $x_p = [q_p \quad u]^T = [x \quad y \quad \theta \quad v \quad \omega]^T$, one can express (V.22) in the following form (see equation (IV.71)):

$$\dot{x}_p = \begin{bmatrix} S_p(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix} \upsilon$$

(V.23)

V.6.1.d Input-output static state feedback linearization

Since $r_M = 2$, one can only propose two output equations $y = h(q_p)$, which must only be parameterised by the posture generalized coordinates q_p. Thus, let be (x_{r_1}, y_{r_1}) a robot reference point on the robot platform defined at the robot reference frame (see Figure V.7). Expressing this point at the world reference frame (see section IV.3.2) and using them as output equations, it yields:

$$h(q_p) = \begin{bmatrix} x + x_{r_1}\cos(\theta) - y_{r_1}\sin(\theta) \\ y + x_{r_1}\sin(\theta) + y_{r_1}\cos(\theta) \end{bmatrix}$$

(V.24)

In addition, since the relative degree of the posture dynamical mode is 2 and $\dim(x_p) = 5$, the following nonlinear change of coordinates $z = \Phi(x_p)$ can be defined:

$$z = \Phi(x_p) \rightarrow \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} h(q_p) \\ A(q_p)u \\ t(q_p) \end{pmatrix}$$

(V.25)

with decoupling matrix:

$$A(q_p) = \left(\frac{\partial h}{\partial \xi} S_p(q_p)\right) = \begin{pmatrix} \cos(\theta) & (d - x_{r_1})\sin(\theta) - y_{r_1}\cos(\theta) \\ \sin(\theta) & -(d - x_{r_1})\cos(\theta) - y_{r_1}\sin(\theta) \end{pmatrix}$$

(V.26)

where $z_2 = \dot{z}_1$, and $z_3 = t(q_p) = \theta$, such that $z = \Phi(x_p)$ is a *diffeomorphism* with inverse operation $x_p = \Phi^{-1}(z)$, which guarantee the necessary and sufficient condition for input-output static state feedback linearization, with a full rank decoupling matrix $A(q_p)$ (see equation (IV.103)) that is globally defined when $x_{r_1} \neq d$, where $t(q_p) = \theta$ avoids input dependence of the remaining nonlinear part, since $\partial t / \partial u = 0$. The description of the input-output system can be expressed in new coordinates $z_i = \Phi_i(x_p)$ as follows (see equations (IV.79)– (IV.81)):

$$\dot{z}_1 = A(q_p)u$$
$$\dot{z}_2 = B(x_p)u + A(q_p)\upsilon$$
$$\dot{z}_3 = Q(q_p)u$$

(V.27)

with the following remaining component definitions:

$$B(q_p) = \frac{\partial}{\partial \xi}[A(q_p)u] \cdot S_p(q_p) = \begin{pmatrix} 0 & -v\sin(\theta) + [(d - x_{r_1})\cos(\theta) + y_{r_1}\sin(\theta)]\omega \\ 0 & v\cos(\theta) + [(d - x_{r_1})\sin(\theta) - y_{r_1}\cos(\theta)]\omega \end{pmatrix}$$

(V.28)

$$Q(q_p) = (0 \quad 1)$$

(V.29)

By substituting $x_p = \Phi^{-1}(z)$ in (V.27) the system in the new coordinates can be written as (see equation (IV.107)):

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = B_z(z) + A_z(z)\upsilon$$
$$\dot{z}_3 = Q_z(z)z_2$$

(V.30)

Since, the decoupling matrix $A_z(z)$ is nonsingular, the *input-output decoupling* can be achieved by applying to (V.30) the following *static state feedback* control law:

$$\upsilon = A_z^{-1}(z)[w - B_z(z)]$$

(V.31)

a partially linearized subsystem is obtained, where the input-output response is both linear and decoupled, with the observable part of the system as:

$$y = z_1; \qquad \dot{z}_1 = z_2; \qquad \dot{z}_2 = w$$

(V.32)

with unobservable zero dynamics (obtained by substituting $z_1 = 0$ and $z_2 = 0$):

$$\dot{z}_3 = Q_z(z)z_2 = \frac{-1}{(x_r - d)}[\sin(z_3) \quad -\cos(z_3)]z_2 = 0$$

(V.33)

which is stable but not asymptotically stable, with internal dynamics exhibiting unstable behaviour when moving backwards [Yun & Sarkar (1998) and Bolzern et al. (2001)].

V.6.1.e Stabilization of the input-output map

The input w can then be used to freely assign the dynamics of z_1, as the dynamics of a second-order stable linear system. More precisely, let z_{1ref}, $\dot{z}_{1ref}$, and $\ddot{z}_{1ref}$ be a smooth reference trajectory, then the following auxiliary control law can be applied:

$$w = \ddot{z}_{1ref} - (\Lambda_1 + \Lambda_2)\dot{z}_e - \Lambda_1\Lambda_2 z_e$$

(V.34)

where $z_e = z_1 - z_{1ref}$, and Λ_1 and Λ_2 are arbitrary positive diagonal $r_M \times r_M$ matrices, generically ensures that: z_e and $\dot{z}_e$ exponentially converge to zero and $z_3(t)$ is bounded for every t.

V.6.2 Robot type(1,1)

The proposed robot-trolley configuration, robot *type*(1,2) , played the role of robot *type*(1,1) (see section IV.4.2.a.2). It has two rear motorized centred steerable wheels and one virtual centred steerable wheel (see Figure V.8) for coordinating the orientation of the centred steerable wheels of the robot. Also, it has two passive fixed wheels, and four passive castor wheels that were ignored in the analysis for being unmotorized.

V.6.2.a Kinematic modelling of the robot type (1,1)

Following the methodology presented in chapter IV, the eleven n_T configuration coordinates of the robot are described by:

$$q = [x \quad y \quad \theta \quad \alpha_{cr} \quad \alpha_{cl} \quad \alpha_{cv} \quad \phi_{fr} \quad \phi_{fl} \quad \phi_{cr} \quad \phi_{cl} \quad \phi_{cv}]^T$$

(V.35)

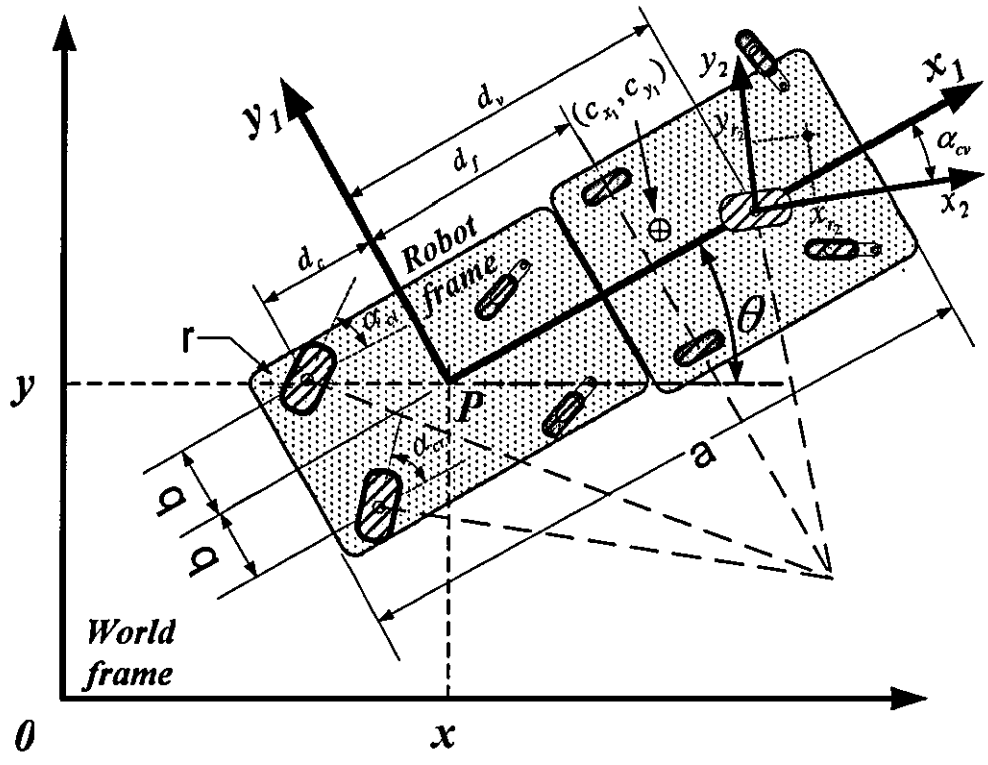


Figure V.8: Posture of the robot type(1,1)

where x and y denote the position of the point P, and θ the orientation of the robot reference frame $\{x_1, y_1\}$ with respect to the fixed world inertial reference frame (see Figure V.8). Also, the variables α_{cr}, α_{cl}, and α_{cv} denote the orientation angle of the right, left, and virtual centred steerable wheels with respect to the x_1 axis respectively. In addition, the variables ϕ_{fr}, ϕ_{fl}, ϕ_{cr}, ϕ_{cl}, and ϕ_{cv} denote the rotation angle of the fixed-right, fixed-left, centred-right, centred-left, and virtual-centred wheels around its horizontal axle. The radius of the wheel is denoted by r. The distances d_f, d_c, and d_v are dimensions relative to the robot reference frame $\{x_1, y_1\}$. Assuming perfect rolling (without both lateral and rolling slipping), $\alpha = 0$ for the fixed wheels, and substituting (IV.11) in (IV.13) and (IV.14), the total of ten m_{Tot} bilateral kinematic constraints $C_T(q, \dot{q}) = 0$ at the robot reference frame can be written as:

- Kinematic constraints orthogonal to the wheels' plane (no lateral slipping):

$$
\begin{aligned}
&-\sin(\theta)\dot{x}+\cos(\theta)\dot{y}-d_f\dot{\theta}=0\\
&-\sin(\theta)\dot{x}+\cos(\theta)\dot{y}-d_f\dot{\theta}=0\\
&-\sin(\theta+\alpha_{cr})\dot{x}+\cos(\theta+\alpha_{cr})\dot{y}+(d_c\cos(\alpha_{cr})-b\sin(\alpha_{cr}))\dot{\theta}=0\\
&-\sin(\theta+\alpha_{cl})\dot{x}+\cos(\theta+\alpha_{cl})\dot{y}+(d_c\cos(\alpha_{cl})+b\sin(\alpha_{cl}))\dot{\theta}=0\\
&-\sin(\theta+\alpha_{cv})\dot{x}+\cos(\theta+\alpha_{cv})\dot{y}+d_v\cos(\alpha_{cv})\dot{\theta}=0
\end{aligned}
$$

(V.36)

- Kinematic constraints along the wheels' plane (rolling without slipping):

$$
\begin{aligned}
&\cos(\theta)\dot{x}+\sin(\theta)\dot{y}+b\dot{\theta}-r\dot{\phi}_{fr}=0\\
&\cos(\theta)\dot{x}+\sin(\theta)\dot{y}-b\dot{\theta}-r\dot{\phi}_{fl}=0\\
&\cos(\theta+\alpha_{cr})\dot{x}+\sin(\theta+\alpha_{cr})\dot{y}+(d_c\sin(\alpha_{cr})+b\cos(\alpha_{cr}))\dot{\theta}-r\dot{\phi}_{fr}=0\\
&\cos(\theta+\alpha_{cl})\dot{x}+\sin(\theta+\alpha_{cl})\dot{y}+(d_c\sin(\alpha_{cl})+b\cos(\alpha_{cl}))\dot{\theta}-r\dot{\phi}_{fl}=0\\
&\cos(\theta+\alpha_{cv})\dot{x}+\sin(\theta+\alpha_{cv})\dot{y}+d_v\sin(\alpha_{cv})\dot{\theta}-r\dot{\phi}_{cv}=0
\end{aligned}
$$

(V.37)

expressed in the form $A_T(q)\dot{q}=0$:

$$
\begin{bmatrix}
-\sin(\theta) & \cos(\theta) & d_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\sin(\theta) & \cos(\theta) & d_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\sin(\theta+\alpha_{cr}) & \cos(\theta+\alpha_{cr}) & d_c\cos(\alpha_{cr})-b\sin(\alpha_{cr}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\sin(\theta+\alpha_{cl}) & \cos(\theta+\alpha_{cl}) & d_c\cos(\alpha_{cl})+b\sin(\alpha_{cl}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\sin(\theta+\alpha_{cv}) & \cos(\theta+\alpha_{cv}) & d_v\cos(\alpha_{cv}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\cos(\theta) & \sin(\theta) & b & 0 & 0 & 0 & -r & 0 & 0 & 0 & 0\\
\cos(\theta) & \sin(\theta) & -b & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0\\
\cos(\theta+\alpha_{cr}) & \sin(\theta+\alpha_{cr}) & d_c\sin(\alpha_{cr})+b\cos(\alpha_{cr}) & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0\\
\cos(\theta+\alpha_{cl}) & \sin(\theta+\alpha_{cl}) & d_c\sin(\alpha_{cl})-b\cos(\alpha_{cl}) & 0 & 0 & 0 & 0 & 0 & 0 & -r & 0\\
\cos(\theta+\alpha_{cv}) & \sin(\theta+\alpha_{cv}) & d_v\sin(\alpha_{cv}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r
\end{bmatrix}
\begin{bmatrix}
\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{\alpha}_{cr}\\ \dot{\alpha}_{cl}\\ \dot{\alpha}_{cv}\\ \dot{\phi}_{fr}\\ \dot{\phi}_{fl}\\ \dot{\phi}_{cr}\\ \dot{\phi}_{cl}\\ \dot{\phi}_{cv}
\end{bmatrix}=0
$$

(V.38)

In order to find the robot internal kinematics, let set $\theta=0$ in (V.4). So, the internal kinematic constraints $A_{T_1}(q)\dot{q}_1=0$ can be expressed as:

$$\begin{bmatrix} 0 & 1 & d_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & d_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin(\alpha_{cr}) & \cos(\alpha_{cr}) & d_c\cos(\alpha_{cr})-b\sin(\alpha_{cr}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin(\alpha_{cl}) & \cos(\alpha_{cl}) & d_c\cos(\alpha_{cl})+b\sin(\alpha_{cl}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin(\alpha_{cv}) & \cos(\alpha_{cv}) & d_v\cos(\alpha_{cv}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & b & 0 & 0 & 0 & -r & 0 & 0 & 0 & 0 \\ 1 & 0 & -b & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0 \\ \cos(\alpha_{cr}) & \sin(\alpha_{cr}) & d_c\sin(\alpha_{cr})+b\cos(\alpha_{cr}) & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0 \\ \cos(\alpha_{cl}) & \sin(\alpha_{cl}) & d_c\sin(\alpha_{cl})-b\cos(\alpha_{cl}) & 0 & 0 & 0 & 0 & 0 & 0 & -r & 0 \\ \cos(\alpha_{cv}) & \sin(\alpha_{cv}) & d_v\sin(\alpha_{cv}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta} \\ \dot{\alpha}_{cr} \\ \dot{\alpha}_{cl} \\ \dot{\alpha}_{cv} \\ \dot{\phi}_{fr} \\ \dot{\phi}_{fl} \\ \dot{\phi}_{cr} \\ \dot{\phi}_{cl} \\ \dot{\phi}_{cv} \end{bmatrix} = 0$$

(V.39)

Therefore, the kinematic motion properties of the robot can be summarised as follows:

- $r_m = 3 - rank(O_{fc}) = 3 - 2 = 1$
- $r_s = rank(O_c) = 1$
- $r_M = r_m + r_s = 2$

which define a robot *type*(1,1).

A controllable kinematic state space representation of the robot at the configuration space is found by computing a smooth codistribution of the matrix A_{T1}. So, let $N(A_{T1})$ be the null space of A_{T1} where $d_{q_1} = \dim(A_{T1}) = 8$, $r_{q_1} = rank(A_{T1}) = 7$, $r_m = d_{q_1} - r_{q_1} = 8 - 7 = 1$, $r_f = 1$, $r_s = 2 - r_f = 1$. Then, by spanning $N(A_{T1})$ it is possible to find $r_m = 1$ smooth vector field $\sigma_m(q)$ and $r_s = 1$ smooth vector field $\sigma_s(q)$ parameterized by one preselected centred angular configuration coordinate α_s (since $r_s = 1$). If we let $\Sigma(q) \in R^{n_T \times r_M}$ be the matrix consisted of the vectors $\Sigma(q) = [\sigma_m(q) \quad \sigma_s(q)]$ parameterized by the angular coordinates $\alpha_s = \alpha_{cv}$, one can select the input vector as $u = [\eta \quad \zeta]^T$, where $\eta = [r/(d_v - d_f)]\dot{\phi}_{cv}$ (since $r_m = 1$, notice that $v_{cv} = r\dot{\phi}_{cv}$ is the virtual wheel heading linear velocity) and $\zeta = \dot{\alpha}_s = \dot{\alpha}_{cv}$ (since $r_s = 1$, where $\dot{\alpha}_{cv}$ is the steering angular velocity of the virtual wheel), such as $\dot{q} = R_q^T \dot{q}_1 = R_q^T \Sigma(q)[\eta \quad \zeta]^T$, where R_q^T is

the transpose of the configuration rotation matrix. Also, one can find $N_c - r_s = 3-1=2$ parametric equations $\alpha_p(\alpha_s)=[\alpha_{p_1}(\alpha_s) \quad \alpha_{p_2}(\alpha_s)]^T=[\alpha_{cr}(\alpha_{cv}) \quad \alpha_{cl}(\alpha_{cv})]^T$, to coordinate the associated remaining centred angular coordinates α_c that guarantee the existence of the ICR at each time instant, which can be expressed as follows:

$$\alpha_{cr}(\alpha_{cv}) = \tan^{-1}\left[-(d_f - d_c)\sin(\alpha_{cv})/(b\sin(\alpha_{cv}) - (d_f - d_v)\cos(\alpha_{cv}))\right] \tag{V.40}$$

$$\alpha_{cl}(\alpha_{cv}) = \tan^{-1}\left[-(d_f - d_c)\sin(\alpha_{cv})/(-b\sin(\alpha_{cv}) - (d_f - d_v)\cos(\alpha_{cv}))\right] \tag{V.41}$$

where

$$\dot{\alpha}_{cr}(\alpha_{cv}) = \frac{\partial \alpha_{cr}}{\partial \alpha_{cv}}\dot{\alpha}_{cv} = \frac{\partial \alpha_{cr}}{\partial \alpha_{cv}}\zeta \tag{V.42}$$

$$\dot{\alpha}_{cl}(\alpha_{cv}) = \frac{\partial \alpha_{cl}}{\partial \alpha_{cv}}\dot{\alpha}_{cv} = \frac{\partial \alpha_{cl}}{\partial \alpha_{cv}}\zeta \tag{V.43}$$

where

$$\frac{\partial \alpha_{cr}}{\partial \alpha_{cv}} = \frac{(d_c - d_f)(d_v - d_f)}{[b\sin(\alpha_{cv}) + (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2\sin(\alpha_{cv})^2} \tag{V.44}$$

$$\frac{\partial \alpha_{cl}}{\partial \alpha_{cv}} = \frac{(d_c - d_f)(d_v - d_f)}{[b\sin(\alpha_{cv}) - (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2\sin(\alpha_{cv})^2} \tag{V.45}$$

Thus, the resultant kinematic state space representation $\dot{q} = S_T(q)u(t)$ of the robot can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\alpha}_{cr} \\ \dot{\alpha}_{cl} \\ \dot{\alpha}_{cv} \\ \dot{\phi}_{fr} \\ \dot{\phi}_{fl} \\ \dot{\phi}_{cr} \\ \dot{\phi}_{cl} \\ \dot{\phi}_{cv} \end{bmatrix} = \begin{bmatrix} d_v\cos(\theta)\cos(\alpha_{cv}) - d_f\cos(\theta+\alpha_{cv}) & 0 \\ d_v\sin(\theta)\cos(\alpha_{cv}) - d_f\sin(\theta+\alpha_{cv}) & 0 \\ \sin(\alpha_{cv}) & 0 \\ 0 & \dfrac{(d_c-d_f)(d_v-d_f)}{[b\sin(\alpha_{cv})+(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2} \\ 0 & \dfrac{(d_c-d_f)(d_v-d_f)}{[b\sin(\alpha_{cv})-(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2} \\ 0 & 1 \\ [(d_v-d_f)\cos(\alpha_{cv})+b\sin(\alpha_{cv})]/r & 0 \\ [(d_v-d_f)\cos(\alpha_{cv})-b\sin(\alpha_{cv})]/r & 0 \\ \left[[b\sin(\alpha_{cv})+(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2\right]^{\frac{1}{2}}/r & 0 \\ \left[[b\sin(\alpha_{cv})-(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2\right]^{\frac{1}{2}}/r & 0 \\ (d_v-d_f)/r & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(V.46)

where $S_T^T A_T^T = 0$ since $S_T \in null(A_T)$. Thus, let Δ_T be the distribution spanned by the vector fields of S_T:

$$\Delta_T = \left\{ \begin{array}{cc} d_v\cos(\theta)\cos(\alpha_{cv}) - d_f\cos(\theta+\alpha_{cv}) & 0 \\ d_v\sin(\theta)\cos(\alpha_{cv}) - d_f\sin(\theta+\alpha_{cv}) & 0 \\ \sin(\alpha_{cv}) & 0 \\ 0 & \dfrac{(d_c-d_f)(d_v-d_f)}{[b\sin(\alpha_{cv})+(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2} \\ 0 & \dfrac{(d_c-d_f)(d_v-d_f)}{[b\sin(\alpha_{cv})-(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2} \\ 0 & 1 \\ [(d_v-d_f)\cos(\alpha_{cv})+b\sin(\alpha_{cv})]/r & 0 \\ [(d_v-d_f)\cos(\alpha_{cv})-b\sin(\alpha_{cv})]/r & 0 \\ \left[[b\sin(\alpha_{cv})+(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2\right]^{\frac{1}{2}}/r & 0 \\ \left[[b\sin(\alpha_{cv})-(d_v-d_f)\cos(\alpha_{cv})]^2+(d_f-d_c)^2\sin(\alpha_{cv})^2\right]^{\frac{1}{2}}/r & 0 \\ (d_v-d_f)/r & 0 \end{array} \right\}$$

(V.47)

A basis computed using the method proposed by P.Hall for the computation of a basis of the Lie algebra of degree 5 with 2 generators can be found in Murray & Sastry (1993):

$$\begin{aligned} &B_1 - B_2 : \sigma_m, \sigma_s \\ &B_3 : [B_1, B_2] \\ &B_4 - B_5 : [B_1, B_3], [B_2, B_3] \\ &B_6 - B_8 : [B_1, B_4], [B_2, B_4], [B_2, B_5] \\ &B_9 - B_{14} : [B_1, B_6], [B_2, B_6], [B_2, B_7], [B_2, B_8], [B_3, B_4], [B_3, B_5] \end{aligned}$$

(V.48)

Now, let $\overline{\Delta}_T$ be the involutive closure of Δ_T obtained from the computation of the control Lie algebra of degree 5 with 2 generators (see equation (V.48)):

$$\overline{\Delta}_T = \{B_1 \quad B_2 \quad B_3 \quad B_4 \quad B_5 \quad B_6 \quad B_8 \quad B_{12}\}$$

(V.49)

with involutive Lie products $B_7, B_9, B_{10}, B_{11}, B_{13}, B_{14}$. The following conclusions can be deduced:

- The distribution Δ_T is not involutive.
- The distribution Δ_T is reducible. The degree of holonomy $M_h = \dim(q) - \dim(\overline{\Delta}_T) = 11 - 8 = 3$.
- The degree of nonholonomy $M_{nonh} = \dim(\overline{\Delta}_T) - (r_m + N_c) = 8 - (1+3) = 4$.

The three holonomic constraints can be found by integrating a choice of equations from (V.46). The first one by subtracting $\dot{\phi}_{fr} - \dot{\phi}_{fl}$:

$$\begin{aligned}
\dot{\phi}_{fr} &= [1/r][(d_v - d_f)\cos(\alpha_{cv}) + b\sin(\alpha_{cv})]\eta \\
\dot{\phi}_{fl} &= [1/r][(d_v - d_f)\cos(\alpha_{cv}) - b\sin(\alpha_{cv})]\eta \\
\dot{\phi}_{fr} - \dot{\phi}_{fl} &= [1/r]2b\sin(\alpha_{cv})\eta
\end{aligned}$$

(V.50)

substituting $\dot{\theta} = \sin(\alpha_{cv})\eta$ in (V.50), and solving for $\dot{\theta}$:

$$\dot{\theta} = \frac{r}{2b}(\dot{\phi}_{fr} - \dot{\phi}_{fl})$$

(V.51)

integrating the equation (V.51) gives the first holonomic constraint:

$$\theta = \frac{r}{2b}(\phi_r - \phi_l) + \theta_0$$

(V.52)

where θ_0 is the initial condition of the robot orientation angle.

The second and third holonomic constraints can be found by integrating $\dot{\alpha}_{cr}$ and $\dot{\alpha}_{cl}$, from (V.46):

$$\dot{\alpha}_{cr} = \frac{(d_c - d_f)(d_v - d_f)}{[b\sin(\alpha_{cv}) + (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2}\zeta$$

(V.53)

$$\dot{\alpha}_{cl} = \frac{(d_c - d_f)(d_v - d_f)}{[b\sin(\alpha_{cv}) - (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2}\zeta$$

(V.54)

where $\zeta = \dot{\alpha}_{cv}$. Since (V.53) were obtained from (V.40), the result of the integration gives:

$$\alpha_{cr}(\alpha_{cv}) = \tan^{-1}[-(d_f - d_c)\sin(\alpha_{cv})/(b\sin(\alpha_{cv}) - (d_f - d_v)\cos(\alpha_{cv}))]$$

(V.55)

$$\alpha_{cl}(\alpha_{cv}) = \tan^{-1}[-(d_f - d_c)\sin(\alpha_{cv})/(-b\sin(\alpha_{cv}) - (d_f - d_v)\cos(\alpha_{cv}))]$$

(V.56)

On the other hand, the posture kinematics state space representation and remaining generalized velocities can be written as follows:

- Posture kinematic state space representation $\dot{q}_p = S_{q_p} u$ (see equation (IV.36)):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\alpha}_{cv} \end{bmatrix} = \begin{bmatrix} d_v\cos(\theta)\cos(\alpha_{cv}) - d_f\cos(\theta + \alpha_{cv}) & 0 \\ d_v\sin(\theta)\cos(\alpha_{cv}) - d_f\sin(\theta + \alpha_{cv}) & 0 \\ \sin(\alpha_{cv}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(V.57)

- Remaining generalized velocities $\dot{q}_r = S_{q_r} u$ (see equation (IV.37)):

$$\begin{bmatrix} \dot{\alpha}_{cr} \\ \dot{\alpha}_{cl} \\ \dot{\phi}_{fr} \\ \dot{\phi}_{fl} \\ \dot{\phi}_{cr} \\ \dot{\phi}_{cl} \\ \dot{\phi}_{cv} \end{bmatrix} = \begin{bmatrix} 0 & \frac{(d_c - d_f)(d_v - d_f)}{[b\sin(\alpha_{cv}) + (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2} \\ 0 & \frac{(d_c - d_f)(d_v - d_f)}{[b\sin(\alpha_{cv}) - (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2} \\ [(d_v - d_f)\cos(\alpha_{cv}) + b\sin(\alpha_{cv})]/r & \\ [(d_v - d_f)\cos(\alpha_{cv}) - b\sin(\alpha_{cv})]/r & \\ [b\sin(\alpha_{cv}) + (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2]^{\frac{1}{2}}/r & \\ [b\sin(\alpha_{cv}) - (d_v - d_f)\cos(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2]^{\frac{1}{2}}/r & \\ (d_v - d_f)/r & \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

(V.58)

Thus, let Δ_p be the distribution spanned by the vector fields of S_{q_p}:

$$\Delta_p = \left\{ \begin{bmatrix} d_v \cos(\theta)\cos(\alpha_{cv}) - d_f \cos(\theta + \alpha_{cv}) & 0 \\ d_v \sin(\theta)\cos(\alpha_{cv}) - d_f \sin(\theta + \alpha_{cv}) & 0 \\ \sin(\alpha_{cv}) & 0 \\ 0 & 1 \end{bmatrix} \right\}$$

(V.59)

Now, let $\overline{\Delta}_p$ be the involutive closure of Δ_p obtained from the computation of the control Lie algebra of degree 3 with 2 generators, [see (V.48)]:

$$\overline{\Delta}_p = \{B_1 \quad B_2 \quad B_3 \quad B_4\}$$

(V.60)

with one involutive Lie product B_5, where:

$$B_1 = \begin{bmatrix} d_v \cos(\theta)\cos(\alpha_{cv}) - d_f \cos(\theta + \alpha_{cv}) \\ d_v \sin(\theta)\cos(\alpha_{cv}) - d_f \sin(\theta + \alpha_{cv}) \\ \sin(\alpha_{cv}) \\ 0 \end{bmatrix}; \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \quad B_3 = \begin{bmatrix} d_v \cos(\theta)\cos(\alpha_{cv}) - d_f \sin(\theta + \alpha_{cv}) \\ d_v \sin(\theta)\sin(\alpha_{cv}) + d_f \cos(\theta + \alpha_{cv}) \\ -\cos(\alpha_{cv}) \\ 0 \end{bmatrix};$$

$$B_4 = \begin{bmatrix} [d_v \sin(\theta)\sin(\alpha_{cv}) + d_f \cos(\theta + \alpha_{cv})]\sin(\alpha_{cv}) \\ -[d_v \cos(\theta)\sin(\alpha_{cv}) - d_f \sin(\theta + \alpha_{cv})]\sin(\alpha_{cv}) \\ 0 \\ 0 \end{bmatrix}$$

(V.61)

The following conclusions can be deduced:

- The distribution Δ_p is not involutive.
- The distribution Δ_p is irreducible and controllable. The degree of holonomy $M_h = \dim(q_p) - \dim(\overline{\Delta}_p) = 4 - 4 = 0$.
- The degree of nonholonomy $M_{nonh} = \dim(\overline{\Delta}_p) - (r_m + r_s) = 4 - (1+1) = 2$.

V.6.2.b Dynamic modelling of the robot type (1,1)

The dynamic model of the robot *type*(1,1) was obtained using the Lagrange's formulation, as showed in the chapter IV. Thus, the dynamic equation of motion at the configuration space with nonholonomic constraints for a wheeled mobile robot system having eleven n_T -dimensional configuration space with eleven n_T generalized coordinates $q=[x \quad y \quad \theta \quad \alpha_{cr} \quad \alpha_{cl} \quad \alpha_{cv} \quad \phi_{fr} \quad \phi_{fl} \quad \phi_{cr} \quad \phi_{cl} \quad \phi_{cv}]^T$, subject to ten m_{Tot} bilateral kinematic constraints, $r_m=1$ degree of mobility, $r_s=1$ degree of steerability, and $r_M=r_m+r_s=1+1=2$ degree of manoeuvrability, is represented by a set of eleven n_T second-order coupled and nonlinear differential equations relating to the positions, velocities, and accelerations of the robot posture and robot joints. They can be written in the form (see equation (IV.57)):

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}=P\tau+A_T^T(q)\lambda$$

(V.62)

The components of the equation (V.62) are defined by:

$$M(q)=\begin{bmatrix} m_T & 0 & A_M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_T & B_M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_M & B_M & I_c & I_{zzw} & I_{zzw} & I_{zzv} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{zzw} & I_{zzw} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{zzw} & 0 & I_{zzw} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{zzv} & 0 & 0 & I_{zzv} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_{yyw} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{yyw} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{yyw} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{yyw} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{yyv} \end{bmatrix};$$

(V.63)

$$C(q,\dot{q})\dot{q}=\begin{bmatrix}-B_M\dot{\theta}^2\\ A_M\dot{\theta}^2\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\end{bmatrix};\ P=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix};\ \tau=\begin{bmatrix}\tau_{\alpha_{cr}}\\ \tau_{\alpha_{cl}}\\ \tau_{\phi_{cr}}\\ \tau_{\phi_{cl}}\end{bmatrix};\ \lambda=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\end{bmatrix};$$

(V.64)

$$A_T(q)=\begin{bmatrix}
-\sin(\theta) & \cos(\theta) & d_f & 0&0&0&0&0&0&0&0\\
-\sin(\theta) & \cos(\theta) & d_f & 0&0&0&0&0&0&0&0\\
-\sin(\theta+\alpha_{cr}) & \cos(\theta+\alpha_{cr}) & d_c\cos(\alpha_{cr})-b\sin(\alpha_{cr}) & 0&0&0&0&0&0&0&0\\
-\sin(\theta+\alpha_{cl}) & \cos(\theta+\alpha_{cl}) & d_c\cos(\alpha_{cl})+b\sin(\alpha_{cl}) & 0&0&0&0&0&0&0&0\\
-\sin(\theta+\alpha_{cv}) & \cos(\theta+\alpha_{cv}) & d_c\cos(\alpha_{cv}) & 0&0&0&0&0&0&0&0\\
\cos(\theta) & \sin(\theta) & b & 0&0&0&-r&0&0&0&0\\
\cos(\theta) & \sin(\theta) & -b & 0&0&0&0&-r&0&0&0\\
\cos(\theta+\alpha_{cr}) & \sin(\theta+\alpha_{cr}) & d_c\sin(\alpha_{cr})+b\cos(\alpha_{cr}) & 0&0&0&0&0&-r&0&0\\
\cos(\theta+\alpha_{cl}) & \sin(\theta+\alpha_{cl}) & d_c\sin(\alpha_{cl})-b\cos(\alpha_{cl}) & 0&0&0&0&0&0&-r&0\\
\cos(\theta+\alpha_{cv}) & \sin(\theta+\alpha_{cv}) & d_c\sin(\alpha_{cv}) & 0&0&0&0&0&0&0&-r
\end{bmatrix}$$

(V.65)

where

$$\begin{aligned}
m_T &= m_c+4m_w+m_v\\
A_M &= -[c_{x_1}\sin(\theta)+c_{y_1}\cos(\theta)]m_c-2(d_f+d_c)\sin(\theta)m_w-d_v\sin(\theta)m_v\\
B_M &= [c_{x_1}\cos(\theta)-c_{y_1}\sin(\theta)]m_c+2(d_f+d_c)\cos(\theta)m_w+d_v\cos(\theta)m_v\\
I_c &= 2({d_f}^2+{d_c}^2+2b^2)m_w+d_v^2m_v+(c_{x_1}^2+c_{y_1}^2)m_c+4I_{zzw}+I_{zzc}+I_{zzv}
\end{aligned}$$

(V.66)

where m_T is the total mass of the robot, m_c is the mass of the cart, m_w is the mass of the wheel, m_v is the mass of the virtual wheel, (c_{x_1},c_{y_1}) is the coordinates of the position of centre of mass with respect to the robot frame of reference, I_{yyw} is the moment of inertia of the wheel about the y axis of the wheel reference frame, I_{zzw} is the moment of inertia

of the wheel about z axis, I_{zv} is the moment of inertia of the virtual wheel about z axis, and I_{zzc} is the moment of inertia of the cart about z axis.

The properties of the dynamic equation can be summarised as:

Property 1. M is symmetric and positive definite.

Property 2. $\dot{M} - 2C$ is skew-symmetric.

Property 3. M and the norm of C are bounded.

V.6.2.c State space representation and input state feedback linearization

The dynamic equation of motion at the steering space can be written as follows (see equation (IV.59)):

$$\overline{M}(q)\dot{u} + \overline{C}(q,\dot{q})u = \overline{P}\tau \equiv \overline{\tau}$$

(V.67)

Applying to the dynamic equation of motion (V.67) the following static nonlinear state feedback

$$\overline{\tau} = \overline{M}(q)\upsilon + \overline{C}(q,\dot{q})u$$

(V.68)

yields $\dot{u} = \upsilon$

and adding the configuration kinematic model $\dot{q} = S_T(q)u(t)$ (see equation (IV.34)) to the configuration dynamical state space representation can be written as:

$$\begin{aligned} \dot{q} &= S_T(q)u \\ \dot{u} &= \upsilon \end{aligned}$$

(V.69)

were $\dot{q} = S_T(q)u$ is represented by equation (V.46) and $\upsilon = \dot{u} = [\dot{\eta} \quad \dot{\zeta}]^T$. The irreducible posture dynamical state space representation can be written as follows (see equation (IV.68)):

$$\dot{q}_p = S_{q_p}(q_p)u$$
$$\dot{u} = \upsilon$$

(V.70)

were $\dot{q}_p = S_{q_p}(q_p)u$ is represented by equation (V.57) and $\upsilon = \dot{u} = [\dot{\eta} \quad \dot{\zeta}]^T$, with remaining generalized velocities $\dot{q}_r = S_{q_r}(q)u$ (see equation (V.58)).

Defining the new generalized coordinates as $x_p = [q_p \quad u]^T = [x \quad y \quad \theta \quad \alpha_{cv} \quad \eta \quad \zeta]^T$, one can express (V.70) in the following form (see equation (IV.71)):

$$\dot{x}_p = \begin{bmatrix} S_p(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_{r_M} \end{bmatrix} \upsilon$$

(V.71)

V.6.2.d Input-output static state feedback linearization

Since $r_M = 2$, one can only propose two output equations $y = h(q_p)$, which must only be parameterised by the posture generalized coordinates q_p. Thus, let (x_{r_2}, y_{r_2}) be a robot reference point defined on the virtual wheel reference frame $\{x_2, y_2\}$ (see Figure V.8). Expressing this point at the world reference frame (see section IV.3.2) and using them as output equations, it yields:

$$h(q_p) = \begin{bmatrix} x + d_v \cos(\theta) + x_{r_2} \cos(\theta + \alpha_{cv}) - y_{r_2} \sin(\theta + \alpha_{cv}) \\ y + d_v \sin(\theta) + x_{r_2} \sin(\theta + \alpha_{cv}) + y_{r_2} \cos(\theta + \alpha_{cv}) \end{bmatrix}$$

(V.72)

In addition, since the relative degree of the posture dynamical mode is 2 and $\dim(x_p) = 6$, the following nonlinear change of coordinates $z = \Phi(x_p)$ can be defined:

$$z = \Phi(x_p) \rightarrow \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} h(q_p) \\ A(q_p)u \\ t(q_p) \end{pmatrix}$$

(V.73)

with decoupling matrix (see equation (IV.82)):

$$A(q_p)=\left(\frac{\partial h}{\partial \xi}R^T(\theta)\Sigma(\alpha_s) \quad \frac{\partial h}{\partial \alpha_s}\right)=\begin{pmatrix}a_1 & a_2\end{pmatrix}$$

(V.74)

where

$$a_1=\begin{pmatrix}(d_v-d_f)\cos(\theta+\alpha_{cv})+[-x_{r2}\sin(\theta+\alpha_{cv})-y_{r2}\cos(\theta+\alpha_{cv})]\sin(\alpha_{cv}) \\ (d_v-d_f)\sin(\theta+\alpha_{cv})+[x_{r2}\cos(\theta+\alpha_{cv})-y_{r2}\sin(\theta+\alpha_{cv})]\sin(\alpha_{cv})\end{pmatrix}$$

$$a_2=\begin{pmatrix}-x_{r2}\sin(\theta+\alpha_{cv})-y_{r2}\cos(\theta+\alpha_{cv}) \\ x_{r2}\cos(\theta+\alpha_{cv})-y_{r2}\sin(\theta+\alpha_{cv})\end{pmatrix}$$

(V.75)

where $z_2=\dot{z}_1$, and $z_3=t(q_p)=[\theta \quad \alpha_{cv}]^T$, such that $z=\Phi(x_p)$ is a *diffeomorphism* with inverse operation $x_p=\Phi^{-1}(z)$, which guarantee the necessary and sufficient condition for input-output static state feedback linearization, with a full rank decoupling matrix $A(q_p)$ that is globally defined when $x_{r_2}\neq 0$ and $d_v\neq d_f$, where $t(q_p)=[\theta \quad \alpha_{cv}]^T$ avoids input dependence of the remaining nonlinear part since $\partial t/\partial u=0$. The description of the input-output system can be expressed in new coordinates $z_i=\Phi_i(x_p)$ as follows (see equations (IV.79) - (IV.81)):

$$\begin{aligned}\dot{z}_1&=A(q_p)u\\ \dot{z}_2&=B(x_p)u+A(q_p)\upsilon\\ \dot{z}_3&=Q(q_p)u\end{aligned}$$

(V.76)

with the following remaining component definitions:

$$B(x_p)=\left(\frac{\partial}{\partial \xi}[A(q_p)u]\cdot R^T(\theta)\Sigma_\xi(\alpha_s) \quad \frac{\partial}{\partial \alpha_s}[A(q_p)u]\right)=\begin{pmatrix}b_1 & b_2\end{pmatrix}$$

(V.77)

where:

$$b_1 = \begin{pmatrix} -(\eta\sin(\alpha_{cv})+\zeta)\sin(\alpha_{cv})\cos(\theta+\alpha_{cv})x_{r2} - [(d_v - d_f - \sin(\alpha_{cv})y_{r2})\eta - y_{r2}\zeta]\sin(\alpha_{cv})\sin(\theta+\alpha_{cv}) \\ -(\eta\sin(\alpha_{cv})+\zeta)x_{r2}\sin(\alpha_{cv})\sin(\theta+\alpha_{cv}) + [(d_v - d_f - y_{r2}\sin(\alpha_{cv}))\eta - y_{r2}\zeta]\sin(\alpha_{cv})\cos(\theta+\alpha_{cv}) \end{pmatrix}$$

$$b_2 = \begin{pmatrix} [(-\sin(\alpha_{cv})x_{r2} - \cos(\alpha_{cv})y_{r2})\eta - x_{r2}\zeta]\cos(\theta+\alpha_{cv}) - [(d_v - d_f - \sin(\alpha_{cv})y_{r2} + \cos(\alpha_{cv})x_{r2})\eta - y_{r2}\zeta]\sin(\theta+\alpha_{cv}) \\ [(-\sin(\alpha_{cv})x_{r2} - \cos(\alpha_{cv})y_{r2})\eta - x_{r2}\zeta]\sin(\theta+\alpha_{cv}) + [(d_v - d_f - \sin(\alpha_{cv})y_{r2} + \cos(\alpha_{cv})x_{r2})\eta - y_{r2}\zeta]\cos(\theta+\alpha_{cv}) \end{pmatrix}$$

(V.78)

and

$$Q(q_p) = \begin{pmatrix} \dfrac{\partial t}{\partial \xi} R^T(\theta)\Sigma_\xi(\alpha_s) & \dfrac{\partial t}{\partial \alpha_s} \end{pmatrix} = \begin{pmatrix} \sin(\alpha_{cv}) & 0 \\ 0 & 1 \end{pmatrix}$$

(V.79)

By substituting $x_p = \Phi^{-1}(z)$ in (V.76) the system in the new coordinates can be written as (see equation (IV.107)):

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= B_z(z) + A_z(z)\upsilon \\ \dot{z}_3 &= Q_z(z)z_2 \end{aligned}$$

(V.80)

Since, the decoupling matrix $A_z(z)$ is nonsingular, the *input-output decoupling* can be achieved by applying to (V.80) the following *static state feedback* control law:

$$\upsilon = A_z^{-1}(z)[w - B_z(z)]$$

(V.81)

a partially linearized subsystem is obtained, where the input-output response is both linear and decoupled, with the observable part of the system as:

$$y = z_1; \qquad \dot{z}_1 = z_2; \qquad \dot{z}_2 = w$$

(V.82)

with unobservable zero dynamics (obtained by substituting $z_1 = 0$ and $z_2 = 0$):

$$\dot{z}_3 = Q_z(z)z_2 = 0$$

(V.83)

which is stable but not asymptotically stable, with internal dynamics exhibiting unstable behaviour when moving backwards [Yun & Sarkar (1998) and Bolzern et al. (2001)].

V.6.2.e Stabilization of the input-output map

The input w can then be used to freely assign the dynamics of z_1, as the dynamics of a second-order stable linear system. More precisely, let z_{1ref}, $\dot{z}_{1ref}$, and $\ddot{z}_{1ref}$ be a smooth reference trajectory, then the following auxiliary control law can be applied:

$$w = \ddot{z}_{1ref} - (\Lambda_1 + \Lambda_2)\dot{z}_e - \Lambda_1\Lambda_2 z_e \tag{V.84}$$

where $z_e = z_1 - z_{1ref}$, and Λ_1 and Λ_2 are arbitrary positive diagonal $r_M \times r_M$ matrices, generically ensures that: z_e and $\dot{z}_e$ exponentially converge to zero and $z_3(t)$ is bounded for every t.

V.7 Summary

The analysis of the feasible robot-trolley configurations for pushing and pulling mail trolleys using stable pushing has arrived at the selection of only one feasible robot-trolley configuration among the twenty, and one single-robot configuration with multiple behaviours, which meet the selection criteria. The best behaviour of the single-robot has been selected in order to have the highest manoeuvrability. The robot-trolley configuration was incorporated with a virtual centred steerable wheel in order to have a common control reference point to coordinate the orientation of the centred steerable wheels of the robot.

The analysis of the internal kinematic constraints showed the kinematic properties of the robot: mobility, steerability, manoeuvrability, and robot type. The configuration and posture kinematic models, obtained by finding the corresponding smooth codistribution, were tested using nonholonomic theory, showing that the configuration kinematical model is involutive and reducible, with holonomic parametric functions $\dot{\alpha}_{cr}$ and $\dot{\alpha}_{cl}$ for wheel steering. The posture kinematical and dynamical models were shown to be not involutive, irreducible, and controllable. The dynamic equation of motion at the configuration space was obtained by taking into the account that the centre of mass can be positioned in any part of the robot platform making these models more general than those

presented in the literature reviewed in this work. The complete model of the dynamic equation of motion at the steering space for the single-robot was obtained (see equations (V.18) and (V.19)). The complete input-output models of the robot (see equations (V.25) - (V.29)) and robot-trolley configuration (see equations (V.73) - (V.79)) were obtained, as well. The input-output static state feedback linearization technique and the stabilization of the input-output map were presented using Cartesian coordinates, instead of polar coordinates, as a consequence of representing the output function (robot reference point) in Cartesian coordinates.

The models can be used to simulate the behaviour of the selected robots when reaching a point or following a predefined trajectory. The remaining generalized velocities can be used as control commands for controlling the motorized joints and for showing the evolution of the other remaining generalized velocities of the unmotorized joints of the robot. Besides, the dynamic equation of motion can be used to compute the torques needed to perform a given task.

VI Simulation results on particular cases

VI.1 Overview

In order to understand the behaviour of the single-robot and the robot-trolley configuration, modelled in Chapter V, a series of simulations were performed using the following tests: reaching a point, tracking straight lines trajectories, and tracking circular trajectories. The architecture of the simulation, definition and generation of trajectories, robot dimensions, robot parameters, stabilization control parameters, and simulation results are presented in this chapter. The tests performed were satisfactory and can be used for design the motion planning approach and for motor selection. The contents of the chapter are arranged as follows:

VI.2 General architecture of the simulation of the particular cases

Figure VI.1 shows the general architecture of the simulation framework useful for simulating both the single-robot and the robot-trolley configuration. It takes into the account all the configuration variables used for modelling the robot configurations and it is based on the unifying theory presented in Chapter IV. They have been arranged in the way that the "mobile robot dynamic system" (equation (IV.62)), a nonlinear model with drift, is expressed in terms of the posture coordinates (equation (IV.36) or (IV.38)) and remaining generalized velocities (equation (IV.37) or (IV.39)). The static nonlinear input state feedback (equation (IV.63)) cancels the nonlinearity of the robot dynamic model (equation (IV.61)). As a result, the mobile robot dynamic system (equation (IV.62)) is reduced to the equation (IV.65), which is nonlinear with drift. The remaining nonlinearity still present in the reduced model (equation (IV.69)), associated to the robot kinematic model, is cancelled by applying a nonlinear input-output static state feedback (equation (IV.109)), which transforms Cartesian accelerations into steering accelerations. Moreover, the equation (IV.109) linearize and decouples the input-output map of the posture dynamic model (equation (IV.69)). The linear feedback (equation (IV.112)) stabilizes the robot in both ways about a point and about a reference trajectory simultaneously. Where the reference point or trajectory is described by z_{1ref} . The equation (IV.74) represents the observable output map and the equation (IV.60) can be used to compute the torques needed for the motorized joints.

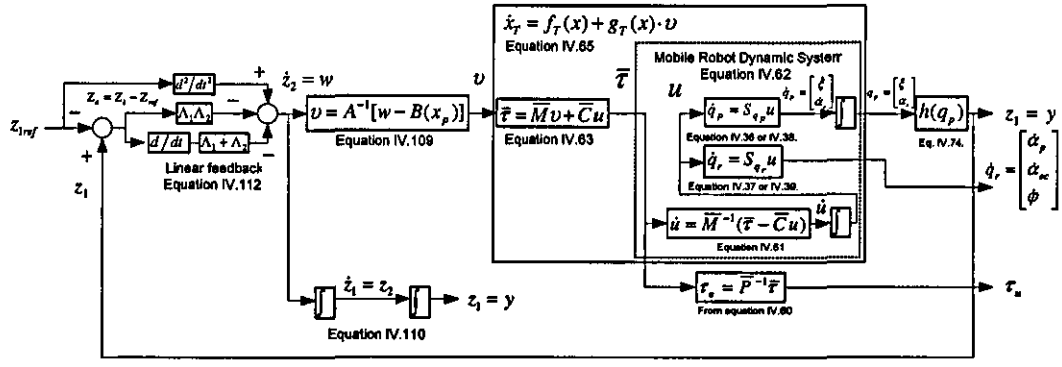


Figure VI.1: Simulation framework general architecture

The general architecture (see Figure VI.1) is useful for computing both the steering system torques (see equation (IV.63)) and the motorized joint torques (see equation (IV.60)), when the Mobile Robot Dynamic System (equation (IV.62)) is replaced for a real robot.

Figure VI.2 shows a reduced model for simulation purposes, where the equation (IV.63) has been substituted in equation (IV.61) cancelling the nonlinearity of the robot dynamic model (see equation (IV.64)). Also, the equation (IV.63) has been substituted in equation (IV.60) to turn the steering accelerations into joint torques.

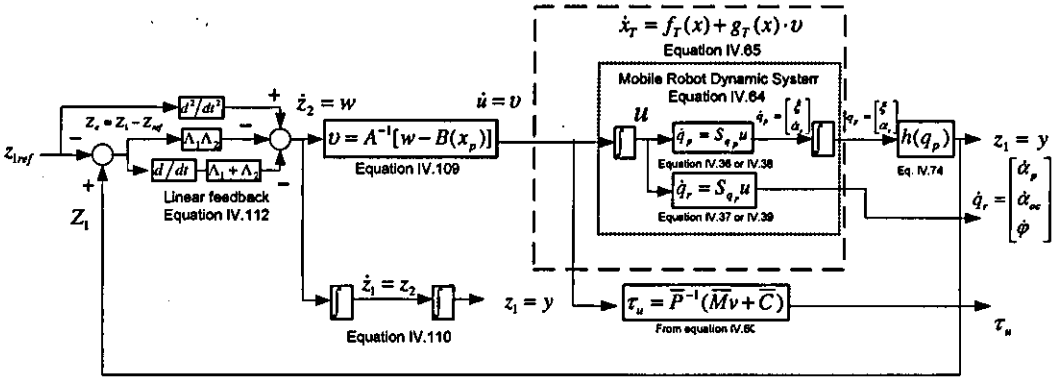


Figure VI.2: Reduced simulation framework architecture

VI.3 Testing particular cases

Point stabilization and trajectory stabilization can be used to control the posture of the robot. Point stabilization is useful when the motion planning method is based on a sequence of checkpoints that should be reached one by one for the mobile robot, where the sequence of checkpoints joints the initial and final robot posture [e.g., Low et al. (2002)]. Trajectory stabilization is useful when the motion planning method is based on following a predefined geometric paths parameterized by time (trajectories) [e.g., Lamiraux et al. (1999)].

Since, any path can be divided into segments of straight lines and circular arcs for the car moving forwards [Dubins (1957)] and for the car moving both forwards and backwards [Reeds & Shepp (1990)]. Then, given any initial and final posture (position and orientation) of a car, there exist a minimal length path constituted of a family of segments of straight lines and circular arcs (with minimal turning radius). Although, these paths are curvature discontinuous at the transitions between segments and arcs, the curvature continuity property of a path is a desirable property only [Fraichard & Scheuer (2004)].

Thus, the single-robot and robot-trolley configuration can be tested just using a family of segments of straight lines and circular arcs, since they are C^2 defined piecewise paths [Lamiraux & Lammond (2001)].

The tests were carried out on the single-robot, unloaded robot-trolley configuration moving backwards, unloaded robot-trolley configuration moving forwards, full-loaded robot-trolley configuration moving backwards, and full-loaded robot-trolley configuration moving forwards, because they were useful for selecting the motion planning approach that will be presented in Chapter VII.

In this work, the values of the kinematic and dynamical parameters, for the single-robot and the robot-trolley configuration, were based on heuristic expected values taken from the commercial electric walkie stackers [e.g., Crown (2002-2005)] and for the commercial York Trolleys [e.g., Envosort (2004)].

VI.3.1 Trajectory definitions

A straight line trajectory of duration t_L, having acceleration a and initial velocity v_0, can be defined in local coordinates in a relative time framework, as follows:

$$x_{rel}(t_{rel}) = \frac{1}{2} a t_{rel}^{\ 2} + v_0 t_{rel}$$

(VI.1)

where x_{rel} is a smooth reference trajectory $x_{rel}(t_{rel}) \in [0, t_L] \rightarrow R$ defined in local coordinates at relative time $t_{rel} = t - t_0$, where t is the current time and t_o is the time when the trajectory started, both defined in absolute time.

Positioning the local coordinates origin at (x_0, y_0) with orientation θ_T with respect to the global reference frame, the straight line trajectory can be defined as follows:

$$\begin{aligned} x_d &= x_0 + \cos(\theta_T) x_{rel} \\ y_d &= y_0 + \sin(\theta_T) x_{rel} \end{aligned}$$

(VI.2)

where $(x_d(t_{rel}), y_d(t_{rel})) \in [0, t_L] \rightarrow R^2$ is a straight line trajectory that has acceleration a and initial velocity v_0, defined on the absolute reference frame in a relative time

framework $t_{rel} = t - t_0$, where (x_0, y_0) is the starting point defined in absolute coordinates and θ_T is the orientation angle of the straight line defined in absolute coordinates as well.

A circular trajectory of duration t_L, having radius r_c, tangential velocity v_t and angular initial condition θ_c, can be defined in local coordinates in a relative time framework as follows:

$$x_{rel}(t_{rel}) = r_c \cos\left(\frac{v_t}{r_c} t_{rel} + \theta_c\right)$$
$$y_{rel}(t_{rel}) = r_c \sin\left(\frac{v_t}{r_c} t_{rel} + \theta_c\right)$$

(VI.3)

where (x_{rel}, y_{rel}) is a smooth reference trajectory $(x_{rel}(t_{rel}), y_{rel}(t_{rel})) \in [0, t_L] \rightarrow R^2$ defined in local coordinates at relative time $t_{rel} = t - t_0$, where t is the current time and t_o is the time when the trajectory started, both defined in absolute time.

Positioning the local coordinates origin at (x_0, y_0) with orientation θ_T with respect to the global reference frame, the circular trajectory can be defined as follows:

$$x_d = x_0 + \cos(\theta_T) x_{rel} - \sin(\theta_T) y_{rel}$$
$$y_d = y_0 + \sin(\theta_T) x_{rel} + \cos(\theta_T) y_{rel}$$

(VI.4)

where $(x_d(t_{rel}), y_d(t_{rel})) \in [0, t_L] \rightarrow R^2$ defines a circular trajectory, having radius r_c and tangential velocity v_t, defined on the absolute reference frame in a relative time framework $t_{rel} = t - t_0$, where (x_0, y_0) is the centre of the circular trajectory defined in absolute coordinates and $\theta_c + \theta_T$ is the angular initial condition defined in absolute coordinates, as well.

VI.3.2 Trajectory generation

The tests carried out to each case, useful for selecting the motion planning approach, were:

1. **Point stabilization**: for the analysis of the robot behaviour when reaching a point.

2. **Constant acceleration line tracking**: for the analysis of the robot behaviour when it starts tracking a new trajectory with different direction.

3. **Constant deceleration line tracking**: for the analysis of the robot behaviour when it is tracking a trajectory while it is reaching the final point of the trajectory.

4. **Constant velocity line tracking**: for the analysis of the robot behaviour when it is tracking a straight line trajectory at constant velocity in steady state.

5. **Constant velocity circle tracking**: for the analysis of the robot behaviour when it is tracking a circular trajectory at constant velocity in steady state.

6. **Constant velocity circle-line tracking**: for the analysis of the robot behaviour when changing from tracking a circular trajectory at constant velocity to tracking a straight line trajectory at constant velocity in steady state and vice versa.

Table VI.1 shows the parameters used to generate each straight line trajectory for testing the robot configurations. Notice that, the definition of a point, for the point stabilization test, was done using a straight line with zero acceleration and zero initial velocity.

Table VI.1: Straight line trajectory test parameters

	a $[m/s^2]$	v_0 $[m/s]$	t_0 $[s]$	t_L $[s]$	x_0 $[m]$	y_0 $[m]$	θ_T $[rad]$
Point stabilization	0	0	0	20	2	2	$\frac{\pi}{4}$
Constant acceleration line tracking	1	0	0	3.5	-2	-2	$\frac{\pi}{4}$
Constant deceleration line tracking	-1	3.5	0	3.5	-2	-2	$\frac{\pi}{4}$
Constant velocity line tracking	0	1	0	6	-2	-2	$\frac{\pi}{4}$
Constant velocity circle-line tracking: Segment I.	0	1	0	2	-0.707	-0.707	$\frac{\pi}{4}$
Constant velocity circle-line tracking: Segment III	0	1	6.717	8.738	0.707	-0.707	$\frac{3}{4}\pi$
Constant velocity circle-line tracking: Segment V	0	1	13.448	15	-0.707	-0.707	$\frac{\pi}{4}$

Table VI.2 shows the test parameters used to generate the circular trajectories for testing the robot configurations. Notice that the circle-line tracking trajectory generation was discomposed in five consecutive segments of straight line and circular trajectories (piecewise continuous trajectories), which were activated consecutively according to the timing (see Table VI.1, as well).

Table VI.2: Circular trajectory test parameters

	r_c [m]	v_t $\left[\frac{m}{s}\right]$	θ_c [rad]	t_0 [s]	t_L [s]	x_0 [m]	y_0 [m]	θ_T [rad]
Constant velocity circle tracking	1	1	0	0	10	0	0	0
Constant velocity circle-line tracking: Segment II	1	-1	$\frac{3}{4}\pi$	2	6.717	1.414	0	0
Constant velocity circle-line tracking: Segment IV	1	1	$\frac{\pi}{4}$	8.738	13.448	-1.414	0	0

The parameters of the table Table VI.1 and Table VI.2 were selected based on the following criteria:

- The linear velocity and acceleration of the robot should be suitable for carrying out the task when interacting with people, e.g., for constant acceleration $a=1$ $[m/s^2]$, and for constant velocity $v_o=1$ $[m/s]$ and $v_t=1$ $[m/s]$. Parameters that were deduced from the analysis of the tests performed in the work of van der Beek et al. (2000) about simulating the daily working practice of postal workers.

- The initial position (x_o, y_o), orientation θ_T, initial time t_0, and duration time t_L of the trajectories were selected in order to produce symmetric trajectories of adequate length for the analysis of the robot stabilisation.

VI.3.3 Stabilization control law parameters

The proposed control law (equation ((IV.112), (V.34) or (V.84))) shows exponential stability behaviour according to section IV.9. Although this is an expected ideal behaviour for some applications, it can demand both high torques and high power to the actuators, since the acceleration components of the reference trajectory is part of the feedback (feed forward term). Also, this approach can be undesirable for reactive behaviour, where the unexpected presence of obstacles can produce high reactive accelerations commands when avoiding obstacles, in consequence, high torques and high power motor demands. So, eliminating the feed forward term $\ddot{z}_{1ref}$, it gives:

$$w=-(\Lambda_1+\Lambda_2)\dot{z}_e-\Lambda_1\Lambda_2 z_e \qquad \text{(VI.5)}$$

where $z_e = z_1 - z_{1ref}$, and Λ_1 and Λ_2 are arbitrary 2×2 positive diagonal matrices. Thus, having a C^1 smooth reference trajectory z_{1ref}, the close loop behaviour of the system can be resumed in the following points:

- The system behaves exponentially stable when reaching a fixed point (point stabilization problem) or tracking a straight line trajectory of constant velocity.
- The system behaves stable when tracking a straight line trajectory of constant acceleration or any trajectory with Cartesian components of constant acceleration.

The components of the arbitrary 2×2 positive diagonal matrices Λ_1 and Λ_2 were selected as follows:

- For point stabilization:

$$\Lambda_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \text{and} \quad \Lambda_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

- For line tracking:

$$\Lambda_1 = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad \text{and} \quad \Lambda_2 = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

which ensure critical damping behaviour (damping ratio $\zeta_d = 1$, and undamped natural frequency $\omega_n = 0.5$ and $\omega_n = 5$ respectively) for both point stabilization and tracking a straight line trajectory of constant velocity.

VI.3.4 Single-robot simulation results: Robot type(2,0)

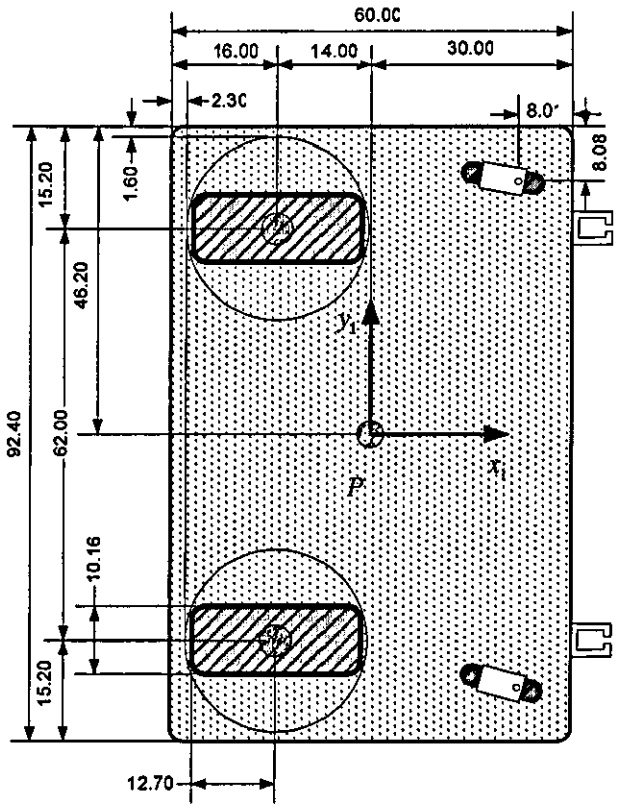


Figure VI.3: Single-robot type(2,0)

Figure VI.3 shows the single-robot *type*(2,0) considered for the simulation that has been endowed with two front gripers to hold the trolley, which consists of two caster wheels (that were ignored in the analysis, see section V.6), and two rear fixed wheels (the orientation angle of the centred steerable wheels was fixed to zero all the time, i.e. $\alpha_{cr} = \alpha_{cl} = 0$, to make that this robot *type*(1,1) behaved itself as robot *type*(2,0)). The single-robot reference frame $\{x_1, y_1\}$ was positioned at the centre of the robot platform coinciding with the robot gravity centre of mass. Also, the single-robot reference point was positioned at the origin of the robot reference frame $(x_{r1}, y_{r1}) = (0,0)$. Besides, Table VI.3 shows the single-robot posture and steering system velocity initial conditions for each test. Table VI.4 shows the heuristic kinematic and dynamic parameters of the robot, as well.

Table VI.3: Initial conditions of the single-robot posture and steering system velocities

	$x(0)$ $[m]$	$y(0)$ $[m]$	$\theta(0)$ $[rad]$	$v(0)$ $[m/s]$	$\omega(0)$ $[rad/s]$
Point stabilization	-2	-2	$\frac{3}{4}\pi$	0	0
Constant acceleration line tracking	-2	-2	$\frac{3}{4}\pi$	0	0
Constant deceleration line tracking	-2	-2	$\frac{\pi}{4}$	3.5	0
Constant velocity line tracking	-2	-2	$\frac{\pi}{4}$	1	0
Constant velocity circle tracking	1	0	$\frac{1}{2}\pi$	1	0
Constant velocity circle-line tracking	-0.707	-0.707	$\frac{1}{4}\pi$	1	0

Table VI.4: Single-robot kinematic and dynamic parameters

Kinematic parameters	**Single-robot** *type*(2,0)	
Wheels dimensions	r_w x w_w [cm]	12.7 x 10.16
Robot dimensions	w_r x l_r x h_r [cm]	92.4 x 60 x 60
b	[cm]	31
d	$-\frac{l_r}{2}+r_w+3.3$ [cm]	-14
a	l_r [cm]	60
Dynamic parameters	**Single-robot** *type*(2,0)	
m_w	[kg]	10
m_c	[kg]	320
I_{zzc}	$\frac{m_c}{12}[w_r^2+l_r^2]$ [kg-m^2]	32.36736
I_{zzw}	$\frac{m_w}{4}\left[r_w^2+\frac{w_w^2}{3}\right]$ [kg-m^2]	48.9246^{-3}
I_{yyw}	$\frac{1}{2}m_w r_w^2$ [kg-m^2]	80.645^{-3}
c_{x_1}	[cm]	0
c_{y_1}	[cm]	0

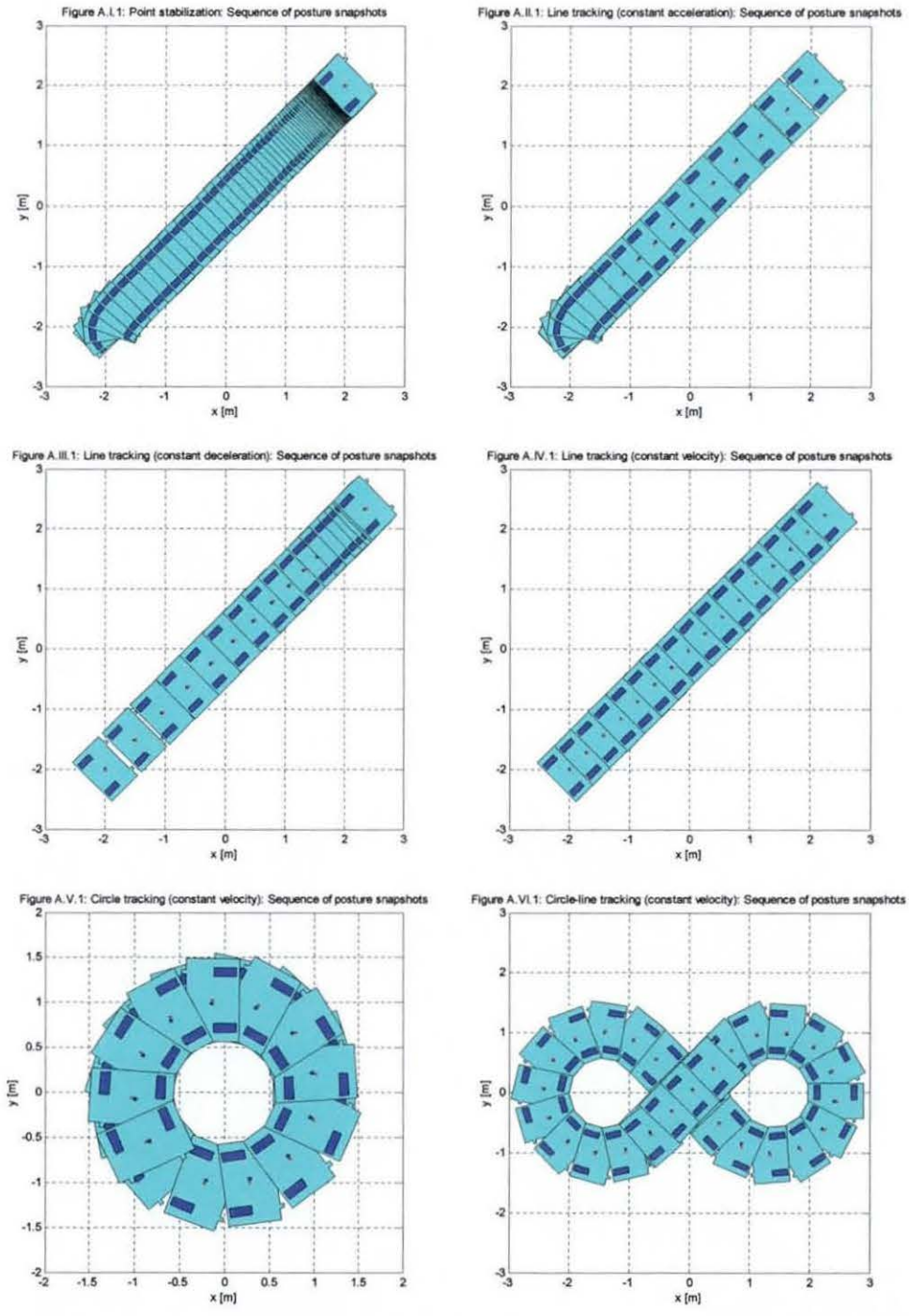


Figure VI.4: Single-robot sequence of posture snapshots.

A.I.1 Point stabilization, A.II.1 Line tracking (constant acceleration), A.III.1 Line tracking (constant deceleration), A.IV.1 Line tracking (constant velocity), A.V.1 Circle tracking (constant velocity), and A.VI.1 Circle-line tracking (constant velocity).

The complete results of the simulation carried out to this robot can be found in the appendices I to VI. On the other hand, Figure VI.4 shows the sequence of posture snapshots of each test extracted from those appendices (the snapshot sample time selected for each test can be found in Table VI.5), where the robot reference point (small red point) and the trajectory reference point (blue point), and the consequences of the

exponential and stable behaviour of the closed loop system can be observed. More precisely, Figure VI.5 shows the robot reference point tracking errors of each test (extracted from the appendices I to VI), where the error behaviour was according to the presented in section VI.3.3, i.e., exponential stability for both point stabilization and constant velocity line tracking, and stable for the others (the error and robot orientation behaviour of each test can be found in Table VI.5, as well).

Also, notice that in **Figure VI.5** all figures have initial zero tracking error, except Figure A.I.4, because the robot posture initial conditions were set such that the robot reference point was on the trajectory from the beginning. Due to the linear steering system velocity initial condition $v(0)$ was set at the same value of the straight line trajectory velocity initial condition v_0, the error exhibited in Figure A.IV.4 is practically always zero (perfect tracking).

Figure VI.6 shows the single-robot reference point posture variables of each test (extracted from the appendices I to VI), where can be observed that except for the Figures A.V.3 and A.VI.3, the orientation of the robot θ behaved asymptotically stable when moving on straight lines (see Figures A.I.3, A.II.3, A.III.3, and A.IV.3), stable when moving on cyclical paths (see Figure A.VI.3) and unstable (in some sense bounded, due to the nature of the variable) when moving on circular paths (see Figure A.V.3).

Additionally to the above-mentioned for this single-robot (see the corresponding appendices when the figures are cited), the computed wheel velocity commands showed that when the robot was rotating (i.e., $\dot{\theta} \neq 0$), the motor rotational velocities had different values (otherwise equal values, e.g., Figure A.II.6). Also, the motor torque commands, needed to be able to follow the velocity commands, showed that both the point stabilization (see Figure A.I.10) and constant acceleration line tracking tasks (see Figure A.II.10) required high starting torque commands. In steady state, the constant velocity line tracking task did not require any sustained torque command (see Figure A.IV.10), but the constant velocity circle tracking did (see Figure A.V.10). The steering system force and torque (e.g., see Figure A.II.9) were expected, since they were according to the force and torque needed to move and rotate a mass of 340 [kg] at an acceleration of 1[m/s]. In addition, the motor power demands needed to accomplish the task showed that they can be achieved having motors of 1 HP (745.7 watts, see Figures 11 of the appendices I to VI,

as well). Finally, negative power must be interpreted as the power used to produce torques against the current direction of the wheel rotational speed.

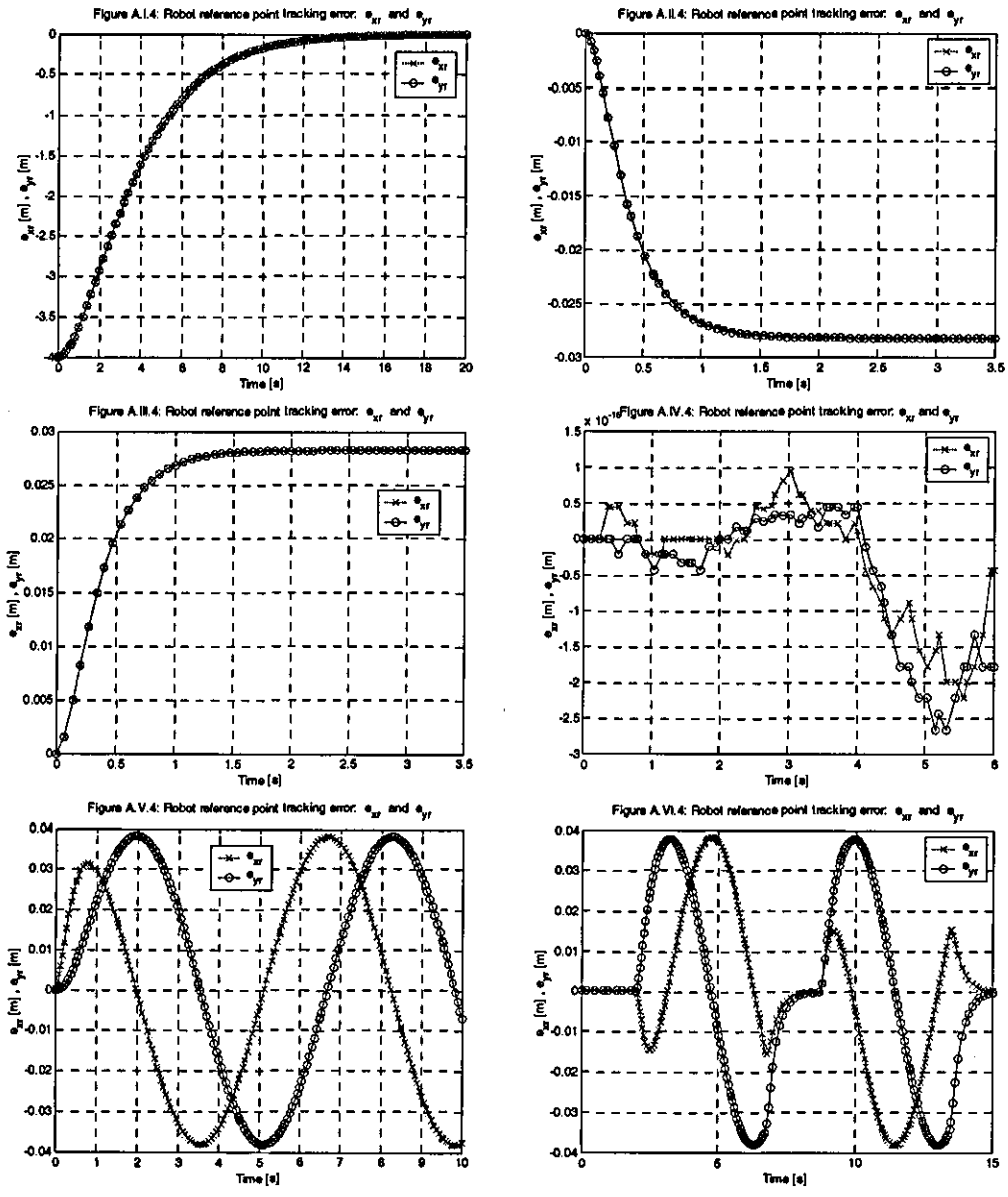


Figure VI.5: Single-robot reference point tracking errors.

A.I.4 Point stabilization, A.II.4 Line tracking (constant acceleration), A.III.4 Line tracking (constant deceleration), A.IV.4 Line tracking (constant velocity), A.V.4 Circle tracking (constant velocity), and A.VI.4 Circle-line tracking (constant velocity).

Table VI.5: Single-robot snapshot sample time, and error and orientation behaviour

	t_{samp} [s]	Error behaviour	θ behaviour
Point stabilization	0.2	Exponential	Asymptotic
Constant acceleration line tracking	0.2	Stable	Asymptotic
Constant deceleration line tracking	0.2	Stable	Asymptotic
Constant velocity line tracking	0.4	Exponential	Asymptotic
Constant velocity circle tracking	0.5	Stable	Bounded
Constant velocity circle-line tracking	0.5	Stable and Exp.	Stable

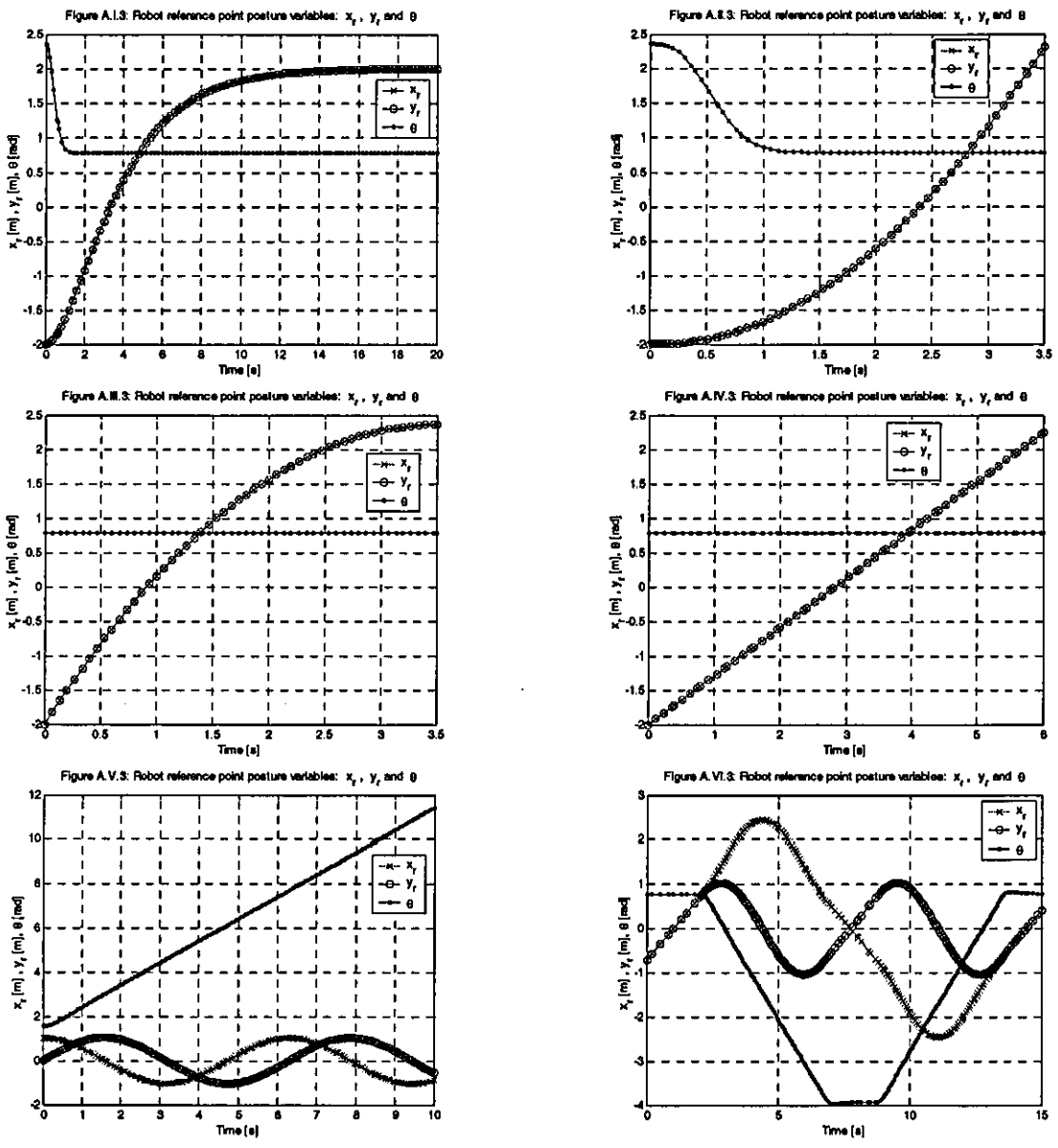


Figure VI.6: Single-robot reference point posture variables.

A.I.3 Point stabilization, A.II.3 Line tracking (constant acceleration), A.III.3 Line tracking (constant deceleration), A.IV.3 Line tracking (constant velocity), A.V.3 Circle tracking (constant velocity), and A.VI.3 Circle-line tracking (constant velocity).

VI.3.5 Robot-trolley configuration simulation results: Robot type(1,1)

Figure VI.7 shows the robot-trolley configuration, robot *type*(1,1), considered for the simulation, which consists of four caster wheels (that were ignored in the analysis, see section V.6), two trolley fixed wheels (which rotational variables were ignored in the analysis for being unmotorized), and two rear centred steerable wheels.

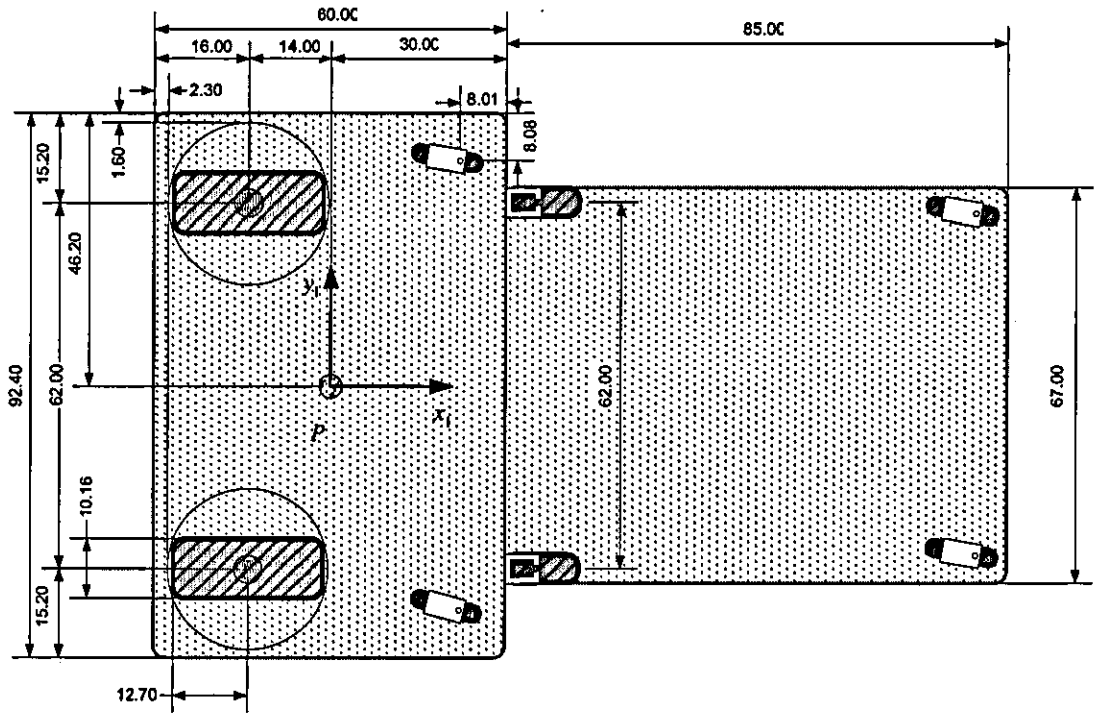


Figure VI.7: Robot-trolley configuration type(1,1)

Table VI.6: Initial conditions of the robot-trolley posture and steering system velocities

Moving backwards	$x(0)$ $[m]$	$y(0)$ $[m]$	$\theta(0)$ $[rad]$	$\alpha_{cv}(0)$ $[rad]$	$\eta(0)$ $[rad/s]$	$\zeta(0)$ $[rad/s]$
Point stabilization	-2.1944	-1.8055	$\frac{\pi}{4}$	π	0	0
Constant acceleration line tracking	-2.1944	-1.8055	$\frac{\pi}{4}$	π	0	0
Constant deceleration line tracking	-1.8055	-1.8055	$-\frac{3}{4}\pi$	π	-56	0
Constant velocity line tracking	-1.8055	-1.8055	$-\frac{3}{4}\pi$	π	-16	0
Constant velocity circle tracking	1	0.275	$-\frac{1}{2}\pi$	π	-16	0
Constant velocity circle-line tracking	-0.5126	-0.5126	$-\frac{3}{4}\pi$	π	-16	0
Moving forwards	$x_p(0)$ $[m]$	$y_p(0)$ $[m]$	$\theta(0)$ $[rad]$	$\alpha_{cv}(0)$ $[rad]$	$\eta(0)$ $[rad/s]$	$\zeta(0)$ $[rad/s]$
Point stabilization	-1.55	-2	π	0	0	0
Constant acceleration line tracking	-1.6818	-2.3181	$\frac{3}{4}\pi$	0	0	0
Constant deceleration line tracking	-2.3181	-2.3181	$\frac{\pi}{4}$	0	56	0
Constant velocity line tracking	-2.3181	-2.3181	$\frac{\pi}{4}$	0	16	0
Constant velocity circle tracking	1	-0.45	$\frac{1}{2}\pi$	0	16	0
Constant velocity circle-line tracking	-1.0253	-1.0253	$\frac{1}{4}\pi$	0	16	0

Table VI.7: Robot-trolley configuration kinematic and dynamic parameters

Kinematic parameters	Robot-trolley *type*(1,1)	
Wheels dimensions	r_w x w_w [cm]	12.7 x 10.16
Robot dimensions	w_r x l_r x h_r [cm]	92.4 x 60 x 60
Load dimensions	w_l x l_l x h_l [cm]	67 x 85 x 150
b	[cm]	31
d_f	$\frac{l_r}{2}+6.25$ [cm]	36.25
d_c	$-\frac{l_r}{2}+r_w+3.3$ [cm]	-14
d_v	Moving Backwards / Forwards [cm]	30 / 42.50
x_{r2}	[cm]	2.5
y_{r2}	[cm]	0
a	l_r+l_l [cm]	145
Dynamic parameters	**Robot-trolley** *type*(1,1)	
m_w	[kg]	10
m_c	[kg]	320
m_l	[kg]	50 - 550
m_c+m_l	[kg]	370 – 870
I_{zzc}	Unloaded / Full-loaded See * [kg-m^2]	59.9779 / 192.3899
I_{zzw}	$\frac{m_w}{4}\left[r_w^2+\frac{w_w^2}{3}\right]$ [kg-m^2]	48.9246^{-3}
I_{yyw}	$\frac{1}{2}m_w r_w^2$ [kg-m^2]	80.645^{-3}
c_{x_1}	Unloaded / Full-loaded $\frac{1}{2}m_l\frac{l_r+l_l}{m_c+m_l}$ [cm]	9.80 / 45.83
c_{y_1}	Unloaded / Full-loaded [cm]	0 / 0
* $I_{zzc}=\frac{1}{12}\frac{(l_r^2+(2b)^2)m_c^2+2(2l_l^2+(2b)^2+3l_l l_r+2l_r^2)m_l m_c+(l_l^2+(2b)^2)m_l^2}{m_c+m_l}$		

The robot-trolley configuration reference frame $\{x_1, y_1\}$ was positioned at the centre of the single-robot platform coinciding with the single-robot gravity centre of mass. Also, a virtual centred steerable wheel[8] was added to create a virtual frame of reference, useful to control the robot-trolley configuration, where the robot reference point (x_{r2}, y_{r2}) was defined, as showed in Figure V.8. On the other hand, Table VI.6 shows the robot-trolley configuration posture and steering system velocity initial conditions of each test. Table VI.7 shows the heuristic kinematic and dynamic parameters of the robot.

[8] Virtual wheel: a wheel that dynamic parameters are zero (mass and moments of inertia) and can be added to the wheeled mobile robot altering or not altering the robot type characteristic (i.e., the degree of mobility and degree of steerability).

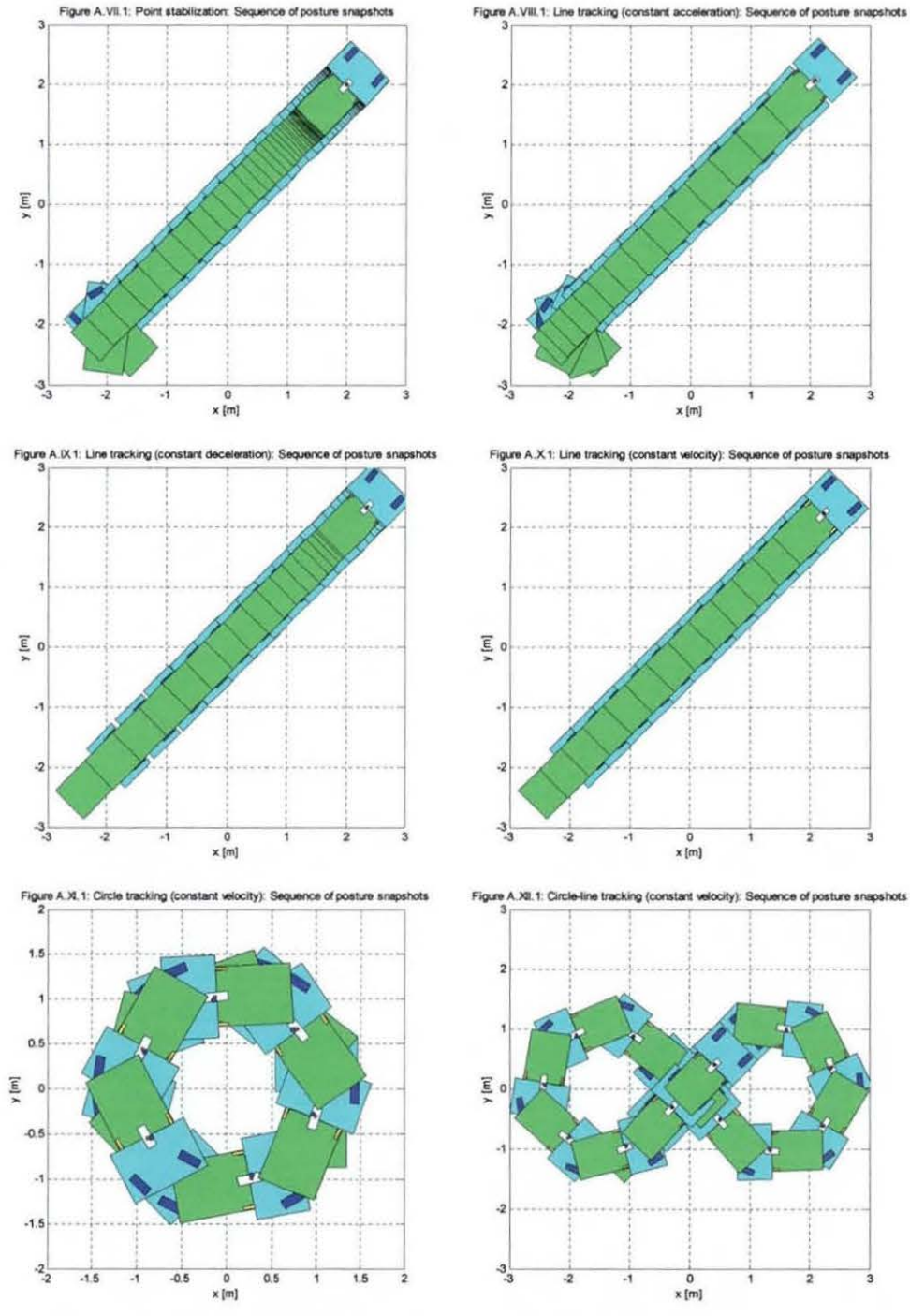


Figure VI.8: Robot-trolley configuration sequence of posture snapshots: Moving backwards.

A.VII.1 Point stabilization, A.VIII.1 Line tracking (constant acceleration), A.IX.1 Line tracking (constant deceleration), A.X.1 Line tracking (constant velocity), A.XI.1 Circle tracking (constant velocity), and A.XII.1 Circle-line tracking (constant velocity).

The complete results of the simulation carried out to this robot can be found in the appendices VII to XVIII. On the other hand, **Figure VI.8** and **Figure VI.9** show the sequence of posture snapshots of each test (extracted from those appendices) when moving backwards and forwards respectively (the snapshot sample time selected for each test can be found in Table VI.8), where the robot reference point (small red point), the

trajectory reference point (blue point), the virtual wheel (white wheel), and the effect of the exponential and stable behaviour of the closed loop system can be observed.

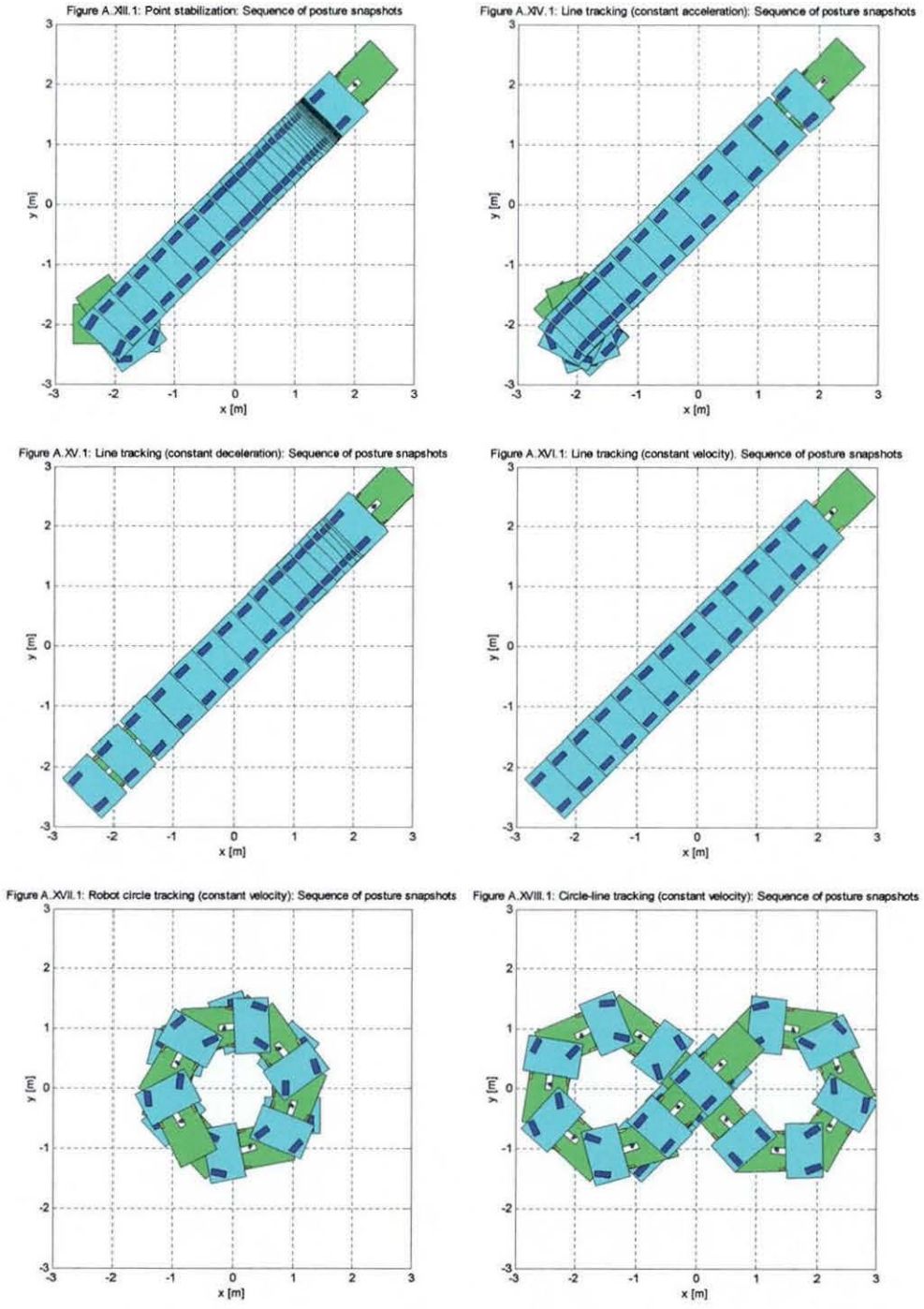


Figure VI.9: Robot-trolley configuration sequence of posture snapshots: Moving forwards.

A.XIII.1 Point stabilization, A.XIV.1 Line tracking (constant acceleration), A.XV.1 Line tracking (constant deceleration), A.XVI.1 Line tracking (constant velocity), A.XVII.1 Circle tracking (constant velocity), and A.XVIII.1 Circle-line tracking (constant velocity).

More precisely, **Figure VI.10** and **Figure VI.11** show the robot reference point tracking errors of each test (extracted from the appendices VII to XVIII), where the error

behaviour was according to the presented in section VI.3.3, i.e., exponential stability for both point stabilization and constant velocity line tracking, and stable for the others (the error, robot orientation, and virtual wheel orientation behaviour of each test can be found in Table VI.8, as well).

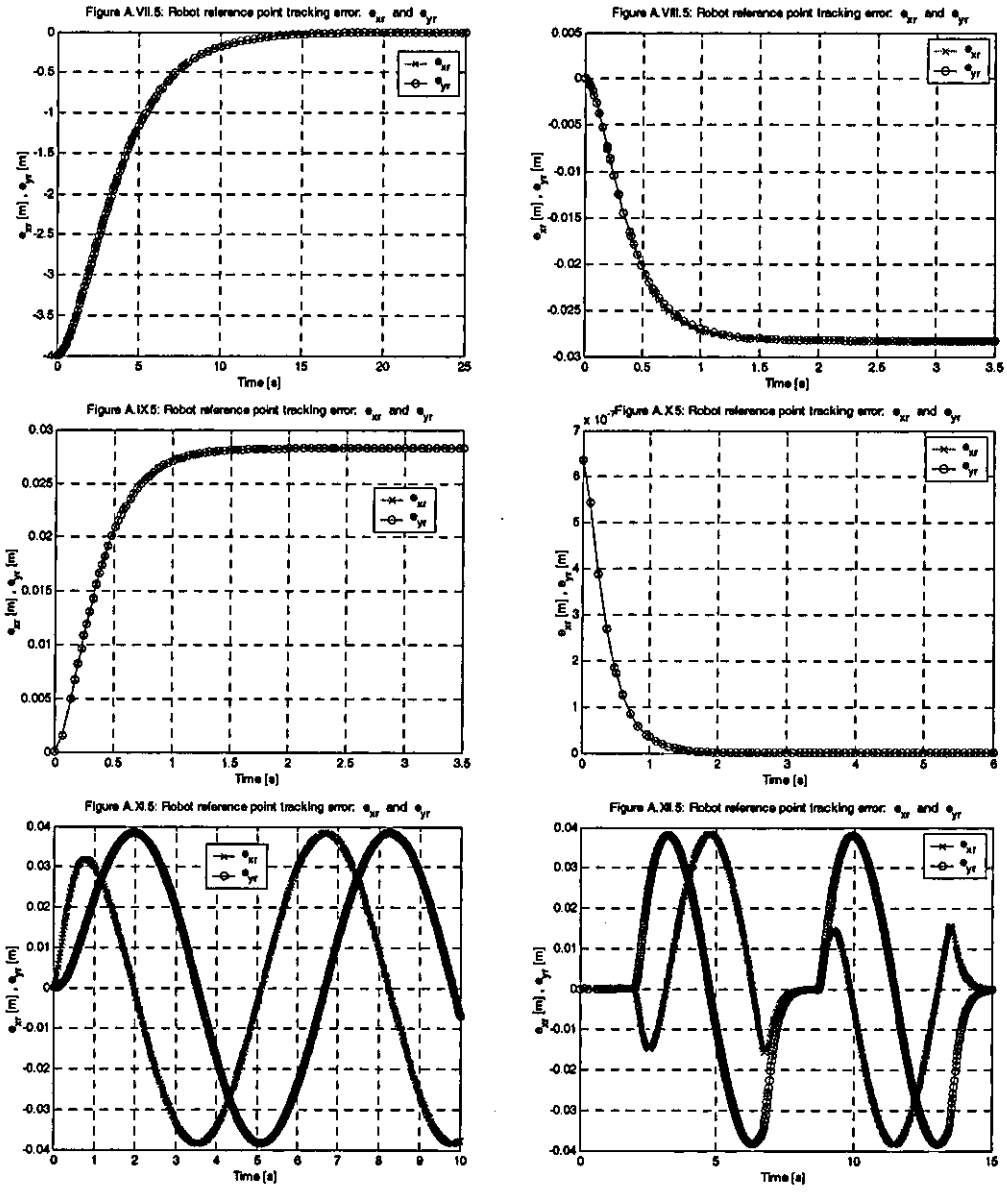


Figure VI.10: Robot-trolley configuration reference point tracking errors: Moving backwards.

A.VII.5 Point stabilization, A.VIII.5 Line tracking (constant acceleration), A.IX.5 Line tracking (constant deceleration), A.X.5 Line tracking (constant velocity), A.XI.5 Circle tracking (constant velocity), and A.XII.5 Circle-line tracking (constant velocity).

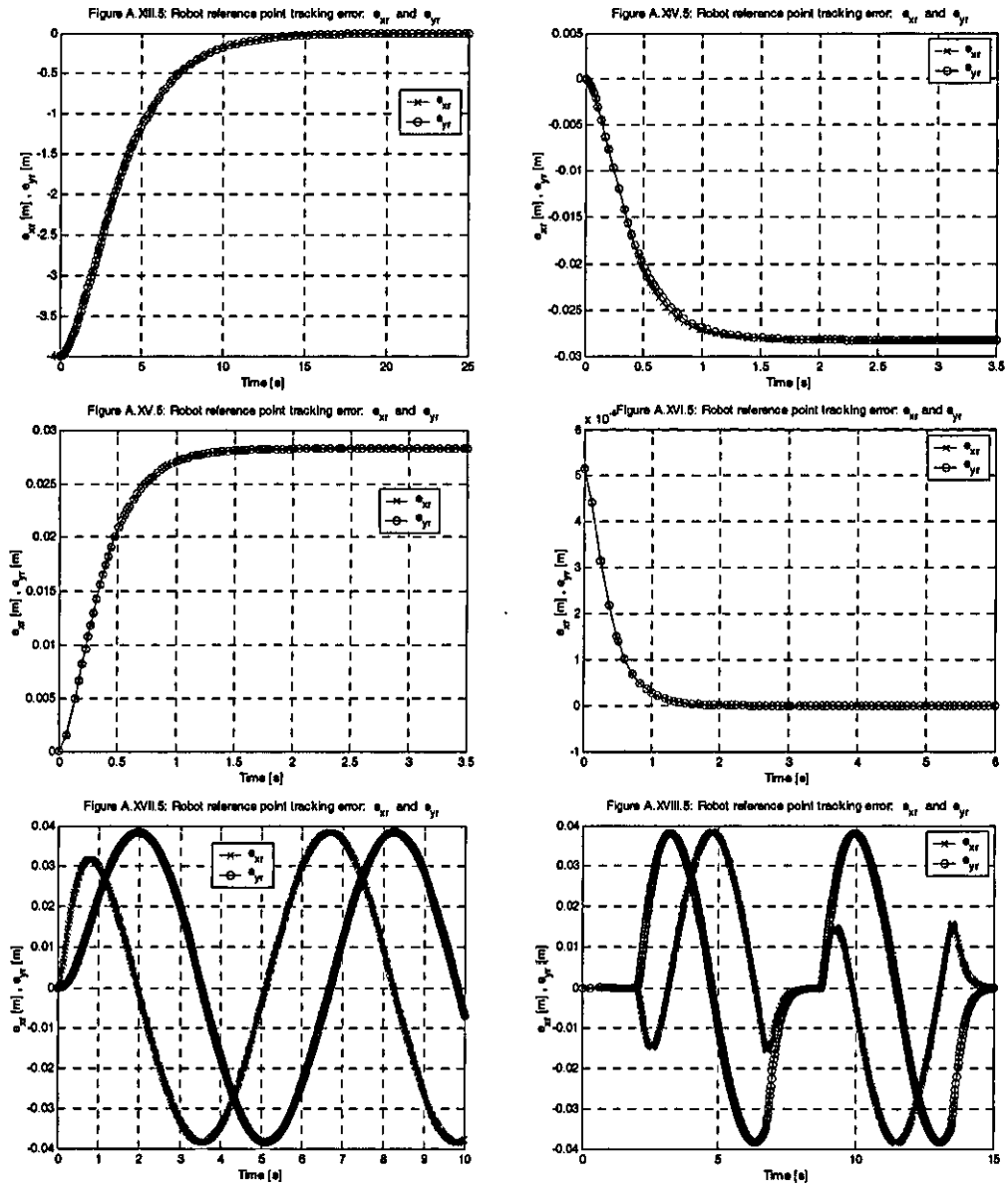


Figure VI.11: Robot-trolley configuration reference point tracking errors: Moving forwards.

A.XIII.5 Point stabilization, A.XIV.5 Line tracking (constant acceleration), A.XV.5 Line tracking (constant deceleration), A.XVI.5 Line tracking (constant velocity), A.XVII.5 Circle tracking (constant velocity), and A.XVIII.5 Circle-line tracking (constant velocity).

Table VI.8: Snapshot sample time t_{samp}, and error, θ and α_{cv} behaviour

	t_{samp} [s]	Error behaviour	θ behaviour	α_{cv} behaviour
Point stabilization	0.4	Exponential	Asymptotic	Stable
Constant acceleration line tracking	0.2	Stable	Asymptotic	Stable
Constant deceleration line tracking	0.2	Stable	Asymptotic	Stable
Constant velocity line tracking	0.5	Exponential	Asymptotic	Stable
Constant velocity circle tracking	1	Stable	Bounded	Stable
Constant velocity circle-line tracking	1	Stable and Exp.	Stable	Stable

Also, notice that in **Figure VI.10** and **Figure VI.11** all the figures have initial zero tracking error, except for the Figure A.VII.5 and Figure A.XIII.5, because the robot posture initial conditions were set such as the robot reference point was on the trajectory from the beginning. Due to the steering systems variable initial condition, $\eta(0)$ was fixed such that the linear velocity of the virtual wheel had the same value of the straight line trajectory velocity initial condition, the error exhibit in Figure A.X.5 and Figure A.XVI.5 is practically always zero (perfect tracking).

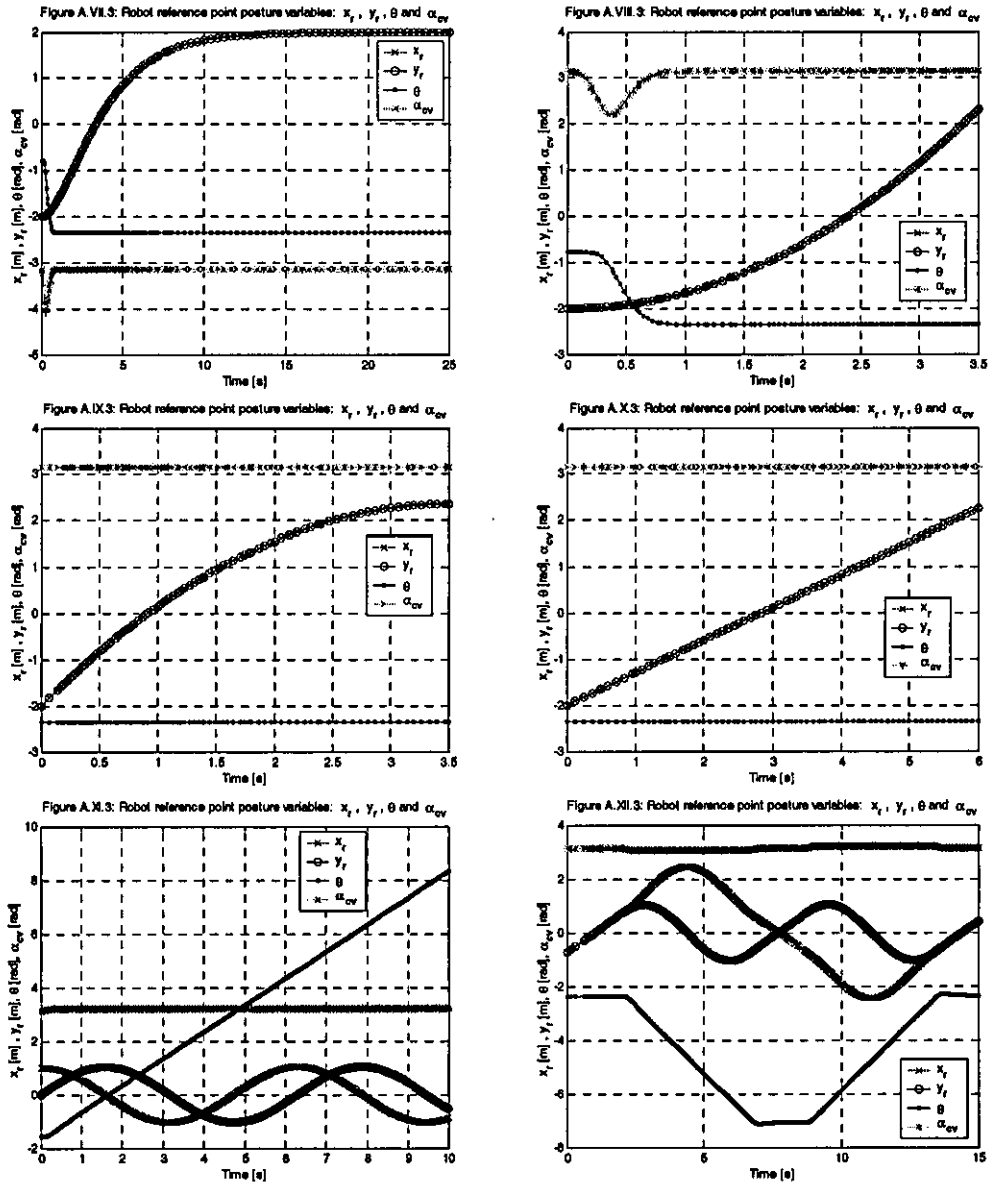


Figure VI.12: Robot-trolley configuration reference point posture variables: Moving backwards.

A.VII.3 Point stabilization, A.VIII.3 Line tracking (constant acceleration), A.IX.3 Line tracking (constant deceleration), A.X.3 Line tracking (constant velocity), A.XI.3 Circle tracking (constant velocity), and A.XII.3 Circle-line tracking (constant velocity).

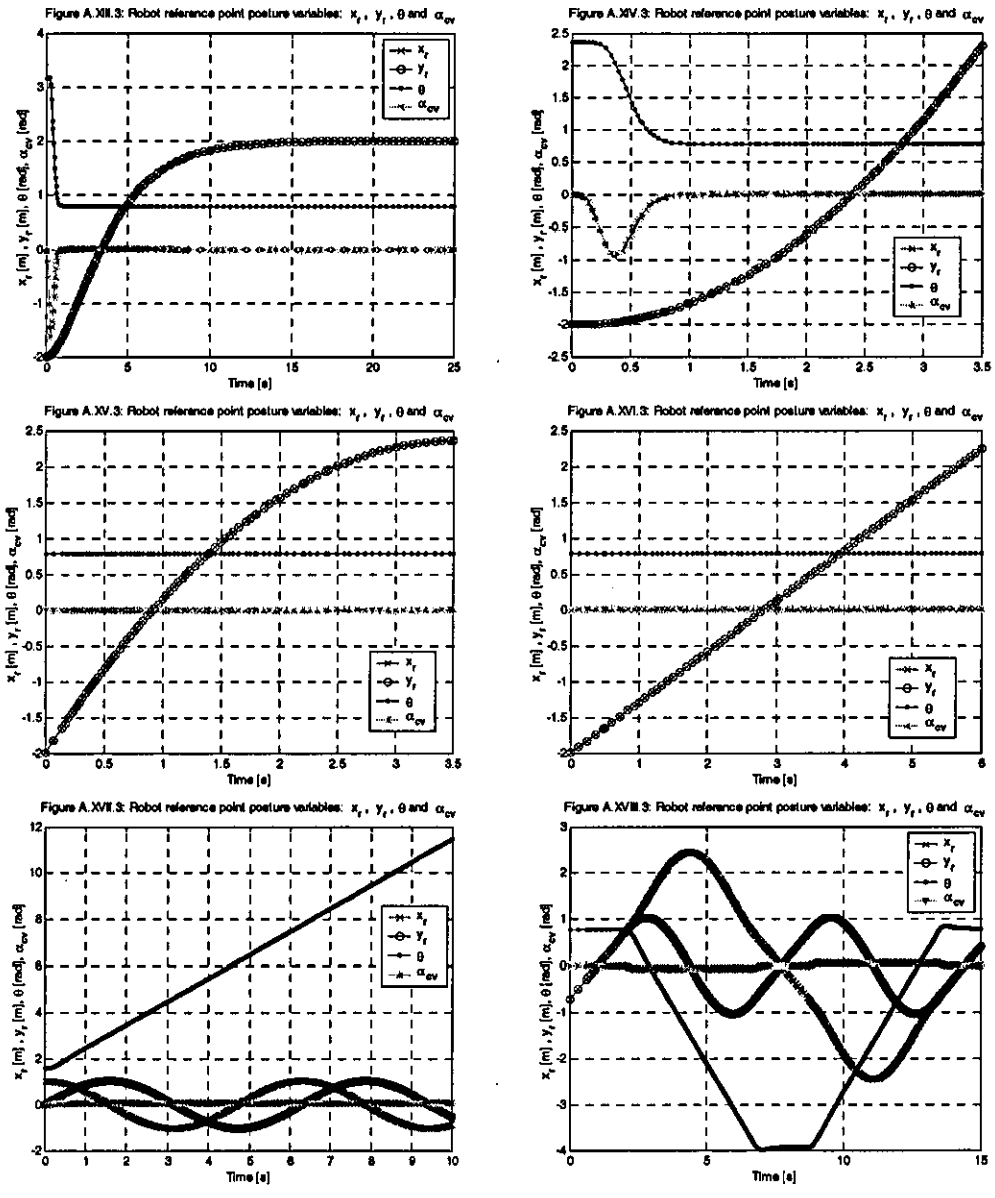


Figure VI.13: Robot-trolley configuration reference point posture variables: Moving backwards.

A.XIII.3 Point stabilization, A.XIV.3 Line tracking (constant acceleration), A.XV.3 Line tracking (constant deceleration), A.XVI.3 Line tracking (constant velocity), A.XVII.3 Circle tracking (constant velocity), and A.XVIII.3 Circle-line tracking (constant velocity).

Figure VI.12 and **Figure VI.13** show the robot reference point posture variables of each test (extracted from the appendices VII to XVIII), where except for Figures A.XI.3 and A.XII.3, and Figures A.XVII.3 and A.XVIII.3, the orientation of the robot θ behaved asymptotically stable when moving on straight lines, stable when moving on cyclical paths (see Figure A.XII and Figure A.XVIII) and unstable (in some sense bounded, due to the nature of the variable) when moving on circular paths (see Figure A.XI.3 and Figure A.XVII). The orientation of the virtual centred steerable wheel α_{cv} behaved stable for every case (see Table VI.8).

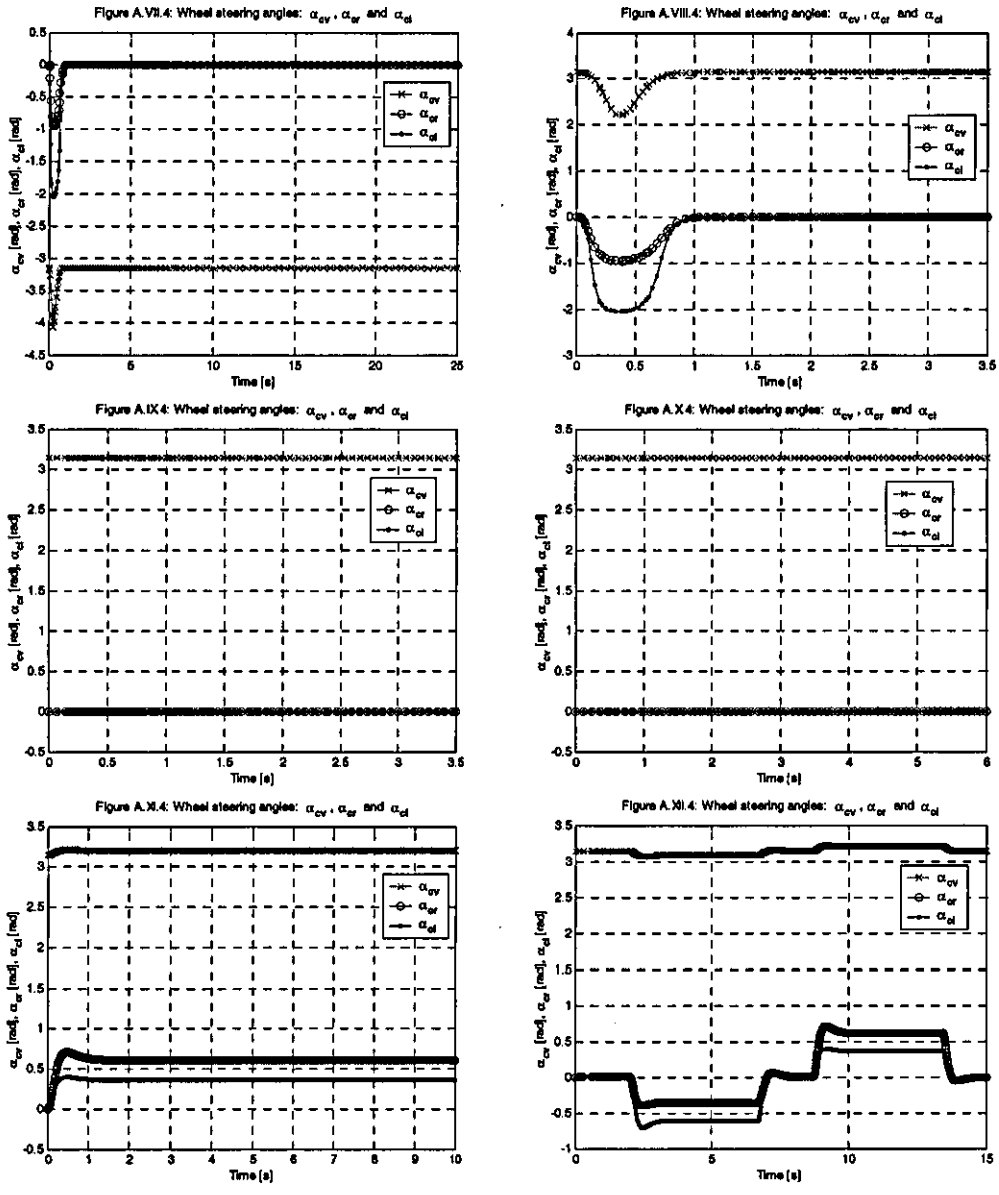


Figure VI.14: Robot-trolley configuration wheel steering angles: Moving backwards.

A.VII.4 Point stabilization, A.VIII.4 Line tracking (constant acceleration), A.XI.4 Line tracking (constant deceleration), A.X.4 Line tracking (constant velocity), A.XI.4 Circle tracking (constant velocity), and A.XII.4 Circle-line tracking (constant velocity).

Figure VI.14 and **Figure VI.15** show the computed values of the robot-trolley configuration "wheel steering angles" to ensure that each imaginary extension of the centred steerable wheel axle meets the instantaneous centre of rotation (ICR) of the planar motion of the robot platform, every time. Also, the results showed that having a steering angle of $-135^0 \leq \alpha_{cr,cl} \leq 135^0$ for each centred steerable wheel is sufficient for accomplishing all the tasks.

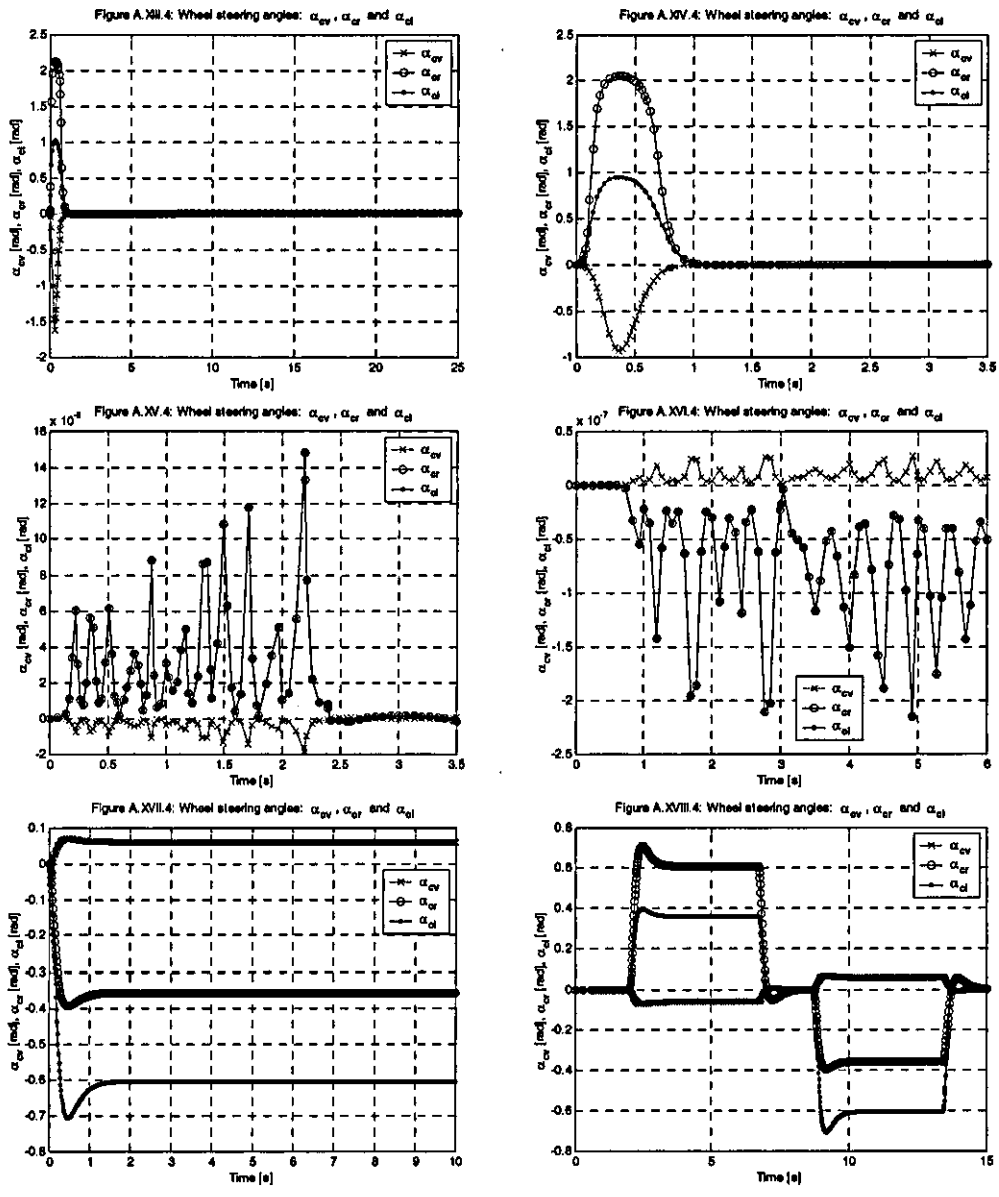


Figure VI.15: Robot-trolley configuration wheel steering angles: Moving forwards.

A.XIII.4 Point stabilization, A.XIV.4 Line tracking (constant acceleration), A.XV.4 Line tracking (constant deceleration), A.XVI.4 Line tracking (constant velocity), A.XVII.4 Circle tracking (constant velocity), and A.XVIII.4 Circle-line tracking (constant velocity).

Additionally, for this robot-trolley configuration (see the corresponding appendix when the figures are cited), the computed wheel velocity commands showed that when the robot was rotating (i.e., $\dot{\theta} \neq 0$), the motor rotational velocities, $\dot{\phi}_{cr}$ and $\dot{\phi}_{cl}$, had different values otherwise equal values (e.g., see Figure A.XII.8). Also, they did not follow the pattern followed by the virtual wheel rotational velocity $\dot{\phi}_{cv}$ when $\dot{\theta} \neq 0$ (e.g., see Figure A.XIV.8). Besides, the motor torque commands showed that both the point stabilization (see Figure A.VII.18) and constant acceleration line tracking tasks (see Figure A.VIII.18)

required high starting torque commands. In steady state, for the single-robot and robot-trolley configuration the constant velocity line tracking test (see Figure A.IV.10 and Figure A.X.18) did not require practically any sustained torque commands. For the robot-trolley configuration the constant velocity circle tracking test (Figure A.XI.18) did not require practically any sustained torque commands, contrary to the single robot test (see Figure A.V.10) that it did.. The steering system variables (Figures 9, 10, 11, and 16 in appendices VII to XVII) are scaled since the virtual rotational velocity can be computed as (see equation (V.46)):

$$\dot{\phi}_{cv} = [(d_v - d_f)/r]\eta$$

(VI.6)

In this way, the virtual rotational acceleration can be computed as:

$$\ddot{\phi}_{cv} = [(d_v - d_f)/r]\dot{\eta}$$

(VI.7)

On the other hand, expressing the input torque vector at the steering space (see equation (V.67)) by its components $\bar{\tau} = [\tau_\eta \quad \tau_\zeta]^T$, where τ_η is the torque needed to compute the torques $\tau_{\phi_{cr}}$ and $\tau_{\phi_{cl}}$, and τ_ζ is the torque needed to compute the torques $\tau_{\alpha_{cr}}$ and $\tau_{\alpha_{cl}}$. Besides, if the motors share the same amount of power, we have:

$$P_{\phi_{cr}} = P_{\phi_{cl}} = \frac{1}{2}P_\eta = \frac{1}{2}\tau_\eta\eta$$
$$P_{\alpha cr} = P_{\alpha cl} = \frac{1}{2}P_\zeta = \frac{1}{2}\tau_\zeta\zeta$$

(VI.8)

where P_η is the power required for the variable η, $P_{\phi_{cr}}$ is the power required to rotate the right wheel, $P_{\phi_{cl}}$ is the power required to rotate the left wheel, P_ζ is the power required for the variable ζ, $P_{\alpha cr}$ is the power required to steer the right wheel, and $P_{\alpha cl}$ is the power required to steer the left wheel. Thus, the torques can be computed as follows:

$$\tau_{\phi_{cr}} = \frac{r}{2\sqrt{[(-d_f + d_v)\cos(\alpha_{cv}) + b\sin(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2]}}\tau_\eta$$

$$\tau_{\phi_{cl}} = \frac{r}{2\sqrt{[(d_f - d_v)\cos(\alpha_{cv}) + b\sin(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2]}}\tau_\eta$$

$$\tau_{\alpha_{cr}} = \frac{[(-d_f + d_v)\cos(\alpha_{cv}) + b\sin(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2]}{2\cdot[(d_f - d_v)(d_f - d_c)]}\tau_\zeta$$

$$\tau_{\alpha_{cl}} = \frac{[(d_f - d_v)\cos(\alpha_{cv}) + b\sin(\alpha_{cv})]^2 + (d_f - d_c)^2 \sin(\alpha_{cv})^2]}{2\cdot[(d_f - d_v)(d_f - d_c)]}\tau_\zeta$$

(VI.9)

where $\tau_{\phi_{cr}}$ is the torque required to rotate the right wheel, $\tau_{\phi_{cl}}$ is the torque required to rotate the left wheel, $\tau_{\alpha_{cr}}$ is the torque required to steer the right wheel, and $\tau_{\alpha_{cl}}$ is the torque required to steer the left wheel.

In addition, the motor power demands needed to accomplish the task showed that they can be of higher value for wheel rotation (e.g., see Figure A.XVIII.20) than for the wheel steering (e.g., see Figure A.XVIII.19). Moreover, the torque (e.g., see Figures A.XIV.12 and A.XIV.17) and power (e.g., see Figures A.XIV.14 and A.XIV.19) demands for wheel steering did not change with the change of the load, according to the theory (the generalized velocity $\dot{\alpha}_c$ do not produce generalized reaction forces, see section IV.4.2) and contrary to the torques (e.g., see Figures A.XIV.13 and A.XIV.18) and power (e.g., see Figures A.XIV.15 and A.XIV.20) demands required for motor rotation, which did change. On the other hand, the power demanded for steering or for rotating the left and right wheel was the same magnitude respectively (e.g., see Figures A.XIV.19 and A.XIV.20), according to the design criterion, which was selected as follows: "Because of the elementary matrix $\overline{P}$ (see equation (IV.59) and section IV.6.5) does not have full rank or valid Moore-Penrose generalized matrix pseudoinverse, the criterion followed to control the wheels was that the motors shared the same amount of power needed to accomplish the corresponding task" (see equation (VI.8)). Finally, negative power must be interpreted as the power used to produce torques against the current direction of the wheel rotational speed.

VI.4 Summary

The single-robot and robot-trolley configuration were tested using a family of segments (piecewise paths C^2 defined) of straight lines and circular arcs. The trajectories were defined in local coordinates in a relative time framework and transformed to absolute coordinates. The stabilization control laws were designed to ensure critical damped behaviour for point stabilization and constant velocity straight line tracking, stable behaviour for trajectories with constant Cartesian acceleration components. The single-robot and robot-trolley configuration tests showed that straight lines and circular arcs trajectories can be used to design motion planning approaches, having relatively low robot reference point tracking errors. The orientation of the robot showed to be asymptotically stable for line tracking and stable or bounded for circle tracking. The orientation of the centred steerable wheels showed to be stable. Finally, the torques and power demanded showed that the robot-trolley configuration was more efficient (compared to the single-robot), since it did not demand power when moving in steady state or on a circular trajectory of constant velocity (e.g., see to compare the Figures A.V.11 and A.XVII.20).

VII Motion planning approach

VII.1 Overview

The single-robot and robot-trolley configuration must be able to move about the environment without colliding with the objects. An original motion planning approach that is defined in a goal domain space for mobile robots is developed in this chapter. The motion planning approach is based on virtual goals and is constituted of both reactive and deliberative modules. The reactive module endows the robot with the capability of avoiding static and moving obstacles while tracking a desired trajectory. Also, it is constituted of both goal reaching module and avoiding obstacle module. The deliberative motion planning approach is based on the generation of straight line trajectories that are defined in a relative time domain for loosely structured environments. With this strategy is possible to make the robot behave in a semiautonomous way (semidependent on the planner, see section II.2). The contents of the chapter are arranged as follows:

VII.2 Motion planning approach

Gonzalez-Villela et al. (2004) (see publication 2 in appendix XXVII) proved that potential fields can be applied to purely kinematic robot models, when the potential fields are taken as distance vectors useful to produce velocity commands to guide the robot through the environment. Also, Gonzalez-Villela et al. (2005) (see publication 4 in appendix XXVII) showed that a dynamic extension for torque control based on backstepping [Fierro & Lewis (1997)] may produce tracking errors when tracking the kinematic reference robot proposed by Kanayama et al. (1990). But, in a decoupled and linearized input-output state feedback framework, where the kinematic and dynamic nonlinearities of the robot have been partially cancelled, the potential field forces can be interpreted as distance vectors. This way, the meaning of the "artificial forces," defined on an "artificial potential field" framework, can be changed to mean "artificial goals (AG)" defined on a "goal domain space (GDS)," eliminating the problem of finding the corresponding potential function, and being only necessary to define the "attractive and repulsive goals" as smooth functions, which can be or not be a gradient of a virtual potential goal function. In consequence, the variety of goal functions that can be used to model objects and goals increase. Also, the concept of the nonholonomic navigation with potential fields and tracking [Kyriakopoulos et al. (1996)] can be redefined as "nonholonomic navigation approach with goals and tracking" defined in a goal domain space.

VII.3 Goal domain space approach

Representing the robot as a particle on a configuration space C in an obstacle avoidance framework, the problem of guiding the robot, in a goal domain space, from an initial position $x_0 \in R^2$ to a final position $x_f \in R^2$ can be discomposed into two:

- Goal reaching
- Obstacle avoidance

The sum of both effects produces a reference goal $\overline{G}_{ref} \in R^2$, expressed in local coordinates, which defines the position of a virtual goal to be reached. The goal reference $\overline{G}_{ref}$ is globally defined on U for all $z_1 \in R^2$ defined in C and can be written as follows:

$$\overline{G}_{ref}(z_1) = \overline{G}_a(z_1, z_{1ref}) + \overline{G}_r(z_1, x_{o1}, ..., x_{on})$$

(VII.1)

where z_{1ref} define a reference trajectory that joints the initial position $x_0 \in R^2$ to the final position $x_f \in R^2$. The attractive goal vector, expressed in local coordinates, is defined as:

$$\overline{G}_a(z_1, z_{1ref}) = G_a(z_1, z_{1ref})\overline{z}_d - z_1$$

(VII.2)

and the resultant repulsive goal vector, expressed in local coordinates, is defined as:

$$\overline{G}_r(z_1, x_{o1}, ..., x_{on}) = -\sum_{i=1}^{n} G_{ri}(z_1, x_{oi})\overline{z}_{oi}$$

(VII.3)

the vectors $\overline{G}_a \in R^2$ and $\overline{G}_r \in R^2$ are globally defined on U for all $z_1 \in C$, where $z_1 \in R^2$ is a point defined on C and somewhere on the robot platform, $z_{1ref} \in R^2$ is a smooth reference trajectory $z_{1ref} \in [0,T] \rightarrow C$, and $x_{oi} \in R^2$ is the corresponding reference position of the $i-th$ obstacle defined on the obstacle space O, useful to compute the position of the corresponding $i-th$ repulsive goal. The scalar attractive function $G_a(z_1, z_{1ref})$ has a unique minimum in $G_a(z_{1ref}, z_{1ref}) = 0$. Also, each scalar smooth function $G_{ri}(z_1, x_{oi})$ is null at all configurations where the distance between the robot reference point z_1 and the corresponding $i-th$ obstacle position x_{oi} is greater than some predefined value, and grows to infinity as the robot gets closer to the corresponding $i-th$ obstacle. Also, the normalized unit vectors $\overline{z}_d \in R^2$ and $\overline{z}_{oi} \in R^2$ are defined as follows:

$$\overline{z}_d = \frac{z_{1ref} - z_1}{\left\| z_{1ref} - z_1 \right\|}$$

$$\overline{z}_{oi} = \frac{x_{oi} - z_1}{\left\| x_{oi} - z_1 \right\|}$$

(VII.4)

where $\|x\| = \sqrt{x^T x}$ represents the suitable norm.

VII.4 Reactive motion planning

In a goal domain space, the reactive motion planning approach can be useful to work as a path-planning algorithm, as a trajectory-planning algorithm, and as a feedback control law, similar to that reported by Rimon & Koditschek (1992), and also as a nonholonomic path planner as reported by Halperin et al. (2004) (see section II.2.1.b).

VII.4.1 Tracking strategy

For semiautonomous mobile robots, these characteristics may be defined locally. If it is necessary for it to have exponential stability behaviour, the reactive motion planning can be defined using the control law as follows (see equation (IV.112)):

$$w = \ddot{z}_{1ref} - (\Lambda_1 + \Lambda_2)\dot{z}_{err} - \Lambda_1\Lambda_2 z_{err}$$

(VII.5)

where $z_{err} = z_1 - z_{1ref}$, and Λ_1 and Λ_2 are arbitrary positive diagonal $r_M \times r_M$ matrices. The virtual reference goal or trajectory (see equation (VII.1)) expressed in the world frame of reference can be written as follows:

$$z_{1ref} = \overline{G}_{ref}(z_1) + z_1$$

(VII.6)

where $\overline{G}_{ref}(z_1)$ must be defined as C^2 function. The error position is defined as:

$$z_{err} = z_1 - z_{1ref}$$

(VII.7)

which is the difference between the current position of the robot reference point z_1 and the current position of the virtual goal z_{1ref}. The control command w (see equation (VII.5)) may be applied to stabilize the decoupled and linearized model of the robot (see equation (IV.110)), about an equilibrium point or a trajectory:

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = w$$

(VII.8)

It may be only necessary to have exponential stability for trajectories that do not have Cartesian acceleration components (see section VI.3.3), since it is only tracking a virtual goal. So, the feed forward term $\ddot{z}_{1ref}$ can be eliminated from the equation (VII.5), as follows:

$$w = -(\Lambda_1 + \Lambda_2)\dot{z}_{err} - \Lambda_1\Lambda_2 z_{err}$$

(VII.9)

being only needed to define $\overline{G}_{ref}(z_1)$ as C^1 function. On this framework it is possible to define the behaviours of goal reaching and obstacle avoidance.

VII.4.2 Goal reaching: attractive goal modelling

Goal reaching is defined as the problem of finding the desirable virtual attractive goal position (see equation (VII.2)) to guide the robot to the goal. The simplest way to define the attractive goal function is to use the real goal as the virtual goal, expressed in local coordinates, as follows:

$$\overline{G}_a = G_a(z_1, z_{1ref})\overline{z}_d - z_1 = z_{1ref} - z_1$$

(VII.10)

where $G_a(z_1, z_{1ref})\overline{z}_d = z_{1ref}$ is the reference trajectory.

VII.4.3 Obstacle avoidance: repulsive goal modelling

Avoiding obstacles is defined as the problem of finding the desirable virtual repulsive goal position expressed in local coordinates (equation (VII.3)), to guide the robot far from the obstacles. One simple way of modelling a repulsive goal is by using a modified version of the model presented by Gonzalez-Villela et al. (2004) (see publication 2 in appendix XXVII), as a repulsive goal vector, as follows:

$$G_{ri}(z, x_{oi})\overline{z}_{oi} = \begin{cases} \eta_i\left(\dfrac{1}{\rho_{ci} - \rho_{ri}} - \dfrac{1}{\rho_{0i}}\right)\overline{z}_{oi} & if\ \rho_{ci} \le \rho_{0i} + \rho_{ri} \\ 0 & if\ \rho_{ci} > \rho_{0i} + \rho_{ri} \end{cases}$$

(VII.11)

where η_i is a constant gain, ρ_{0i} represents the limit distance of the repulsive goal function influence, ρ_{ri} is the radius of the circular obstacle, and $\rho_{ci} = \|z_{oi} - z_1\|$ is the distance to the centre of the obstacle, all of them related to the $i-th$ obstacle. Also, large objects can be divided into circular fragments [Quinlan (1994)] that can be modelled using equation (VII.11). Besides, the complete object space can be modelled as the sum of the repulsive goals vectors as follows:

$$\overline{G}_r(z_1, z_{o1}, \ldots, z_{on}) = -\sum_{i=1}^{n} G_{ri}(z_1, x_{oi})\bar{z}_{oi} = -\sum_{i=1}^{n} \begin{cases} \eta_i \left(\dfrac{1}{\rho_{ci} - \rho_{ri}} - \dfrac{1}{\rho_{0i}} \right) \bar{z}_{oi} & if \ \rho_{ci} \le \rho_{0i} + \rho_{ri} \\ 0 & if \ \rho_{ci} > \rho_{0i} + \rho_{ri} \end{cases}$$

(VII.12)

VII.5 Deliberative motion planning approach

In a goal domain space, the deliberative motion planning approach can be expressed in terms of smooth reference trajectories $z_{1ref} \in [0,T] \rightarrow C$ as follows:

- Trajectories for goal reaching
- Trajectories for posture reaching

The first approach is oriented to reach a desirable point (a starting point for docking) without taking care of the final robot orientation, taking the advantage of the behaviour of the decoupled and linearized model of the robot (see equation (VII.8)). The second approach is oriented to reach the desired posture (docking point), taking advantage of both the behaviour of the decoupled and linearized model of the robot (see equation (VII.8)) and its asymptotic stability when moving forwards, expressed in its unobservable internal nonlinear dynamics (see equation (IV.107)), represented by:

$$\dot{z}_3 = Q_z(z) z_2$$

(VII.13)

that can guide the robot to the final orientation when moving forwards.

VII.5.1 Reference trajectory generation for loosely structured environments

Loosely structured environments are normally defined by geometric objects, straight corridors, open areas, and tracks, where the mobile robots can move freely. For instance, Figure VII.1 shows the collision free tracks (dotted lines), which define track maps. They can be represented in graph structure as shown in Figure VII.2, whose lines have some information related to the track (e.g., the distance between nodes). On the other hand, the nodes represent both track joints and change in directions [Gonzalez-Villela & Parkin (2005)] (see publication 3 in appendix XXVII). After defining the route it is necessary to compute the precise traversal time, deal properly with junctions and turns, schedule a multi-vehicle environment, and deal with the unexpected [Cameron (1994)].

Thus, the robot must follow the paths in scheduled time (predefined trajectories) while avoiding fixed obstacles (in corridors) or mobile obstacles (in open areas). Due to the narrowness of the aisles and the necessity to keep the robot in the predefined routes, the motion planning based on straight line trajectories is the most convenient for saving space, since it does not require much space to navigate.

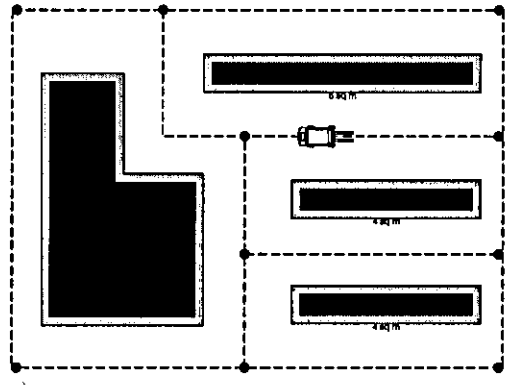


Figure VII.1: Track maps in loosely structured environment

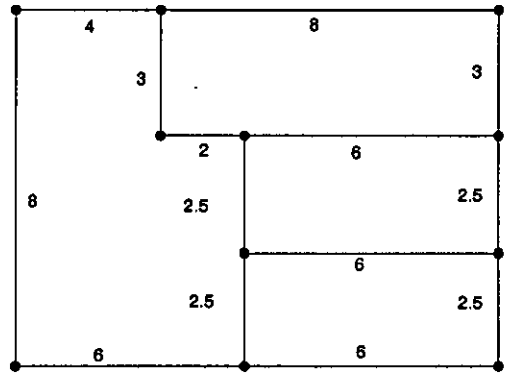
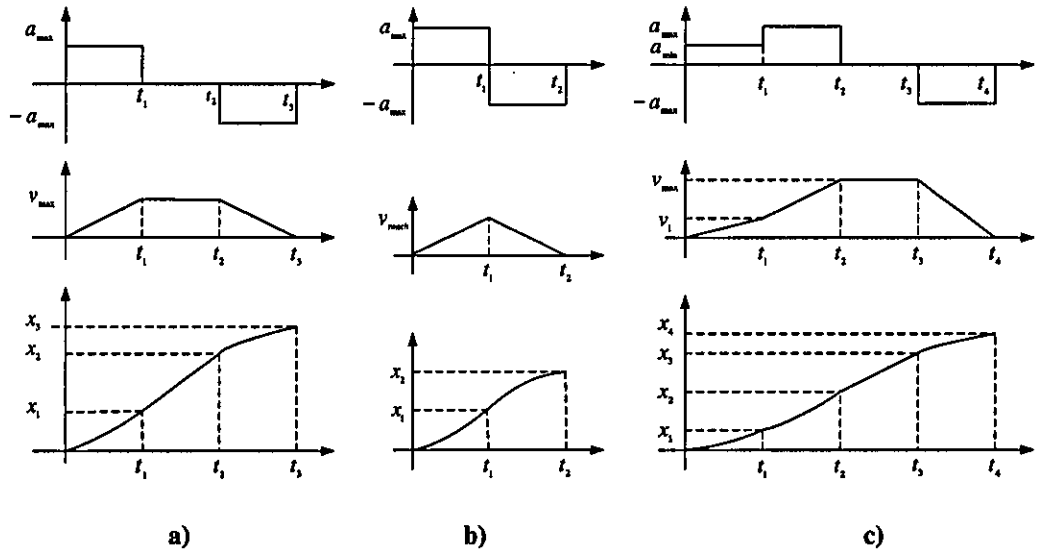


Figure VII.2: Graph structure of the environment

On the other hand, having a linearized and decoupled input-output static state feedback model of the robot that guarantees that a specific point z_1, defined somewhere on the robot platform, follows a desired trajectory with internal dynamics exhibiting asymptotic stability when moving forwards. It is possible to design, based on the distance between nodes, a reference trajectory generator that uses piecewise reference trajectories in a relative time defined framework (see section VI.3.1). If there are a series of nodes that lie on a straight line, the planner must optimize the number of nodes in order to have a starting node and an ending node for each straight track, in which nodes define both track joints and direction changes. Thus, the basic trajectories designed for going from one node to another were the following (they are expressed in local coordinates):

- Starting-ending trajectory for the single-robot (see Figure VII.3.a).
- Defective starting-ending trajectory for the single-robot and the robot-trolley configuration: When the distance is too short that the robot can not reach its maximal velocity when accelerating and decelerating (see Figure VII.3.b).
- Starting-ending trajectory changing starting accelerations for the robot-trolley configuration: To reduce the starting power demands (see Figure VII.3.c).



a) b) c)

Figure VII.3: Starting ending trajectories: a) Starting-ending trajectory, b) Defective starting-ending trajectory, and c) Starting-ending trajectory changing starting accelerations.

VII.5.2 A task for moving wheeled loads from one posture to another

Figure VII.4 shows the proposed task sequence (in flowchart form) needed to perform a complete task. When a new task arrives, the single-robot must undock from the last position and travel to a specific docking point, associated to the corresponding load position. Along the way, the single-robot must accelerate from the trajectory starting node until reaching the maximal velocity (except when the distance between nodes only allows the application of a defective starting-ending trajectory), keeps on this velocity for a while and decelerates while reaching the trajectory ending node. Then, starts again by making the last trajectory ending point being the next trajectory starting point, until reaching the docking point. When docking, the trajectory acceleration and deceleration must be lower than travelling in order to keep the tracking errors lower than travelling. On the other hand, the single-robot, tested on constant acceleration line tracking (see appendix II), constant velocity line tracking (see appendix IV), and constant deceleration (see appendix III) showed that it is suitable for this kind of trajectory tracking tasks (see Figure VII.3.a or Figure VII.3.b, as applicable).

Once the single-robot is docked, the single-robot must grasp the load and undock as robot-trolley configuration to travel along the trajectories by reaching the ending nodes up to the next docking point is reached. Then, the robot-trolley configuration must position the load by docking (when the trajectory acceleration and deceleration must be lower than travelling, as well, in order to keep the tracking errors lower than travelling). On the other hand, the robot-trolley configuration tested when going backwards on constant velocity line tracking (see appendix X) and constant deceleration (see appendix IX) showed that it is suitable for this kind of trajectory tracking tasks, but for constant acceleration line tracking (see appendix VIII) was not, because it demanded both high motor torques and high motor power for wheel rotations when starting a new trajectory (see appendix VIII). So, the suitable trajectory is that that starts with a low acceleration command and speed up after stabilizing the orientation of the robot-trolley configuration (see Figure VII.3.c or Figure VII.3.b, as applicable). Finally, the robot-trolley configuration must ungrasp the load, and receive a new task or go to a rest position.

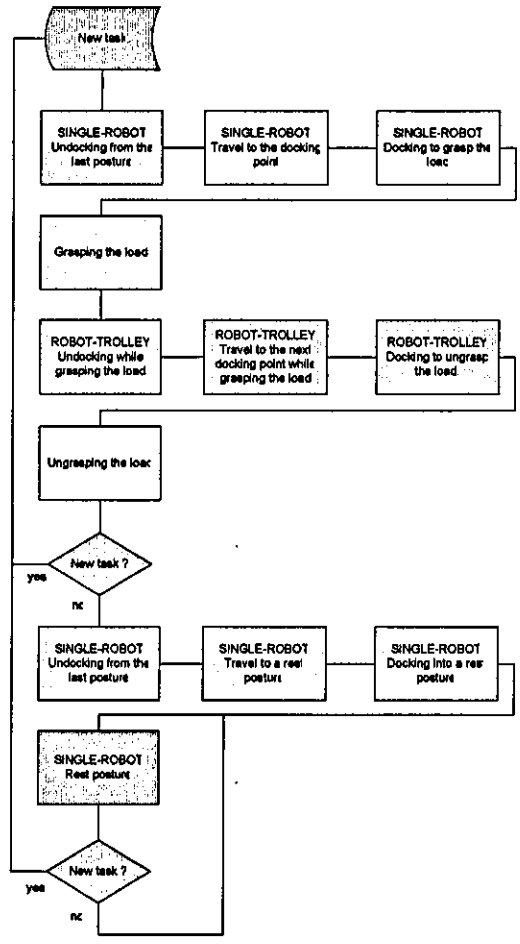


Figure VII.4: Task sequence

VII.6 Summary

The reactive and deliberative motion planning strategy presented in this chapter makes the robot behave in a semiautonomous way (semidependent on the planner). It is useful for tracking a desired trajectory or stabilizing the robot about an equilibrium point while avoiding static and mobile obstacles. Because this strategy is defined in a goal domain space is not required to define a potential function. Also, it is possible to define new functions for the modelling of both the attractive and the repulsive goal. The motion planning approach presented in this chapter is "a nonholonomic navigation approach with goals and tracking".

VIII Motion planning simulation results

VIII.1 Overview

A motion planning approach for reaching a goal while avoiding obstacles for conventional wheeled robots has been proposed in Chapter VII. This approach is simulated in this chapter on a loosely structured environment, which was designed similar to those that can be found in a mail distribution centre. The single-robot and the robot-trolley configuration, modelled in Chapter V, were used to be tested on the following two simulation tasks: corridor deliberative path planning test and reactive motion planning test. The tests performed were oriented to evaluate the performance of the robot in these tasks, and to find the design parameters of the motors. In the deliberative test, the robot followed a predefined deliberative trajectory to accomplish a complete basic task suggested by the trajectory planner. In the reactive test, the robot followed a reactive trajectory suggested by the reactive module in order to avoid fixed objects while moving in corridors, or to avoid moving objects while the objects were moving along with the robot, in the same direction and oscillating about their trajectory. The travelling tracking error due to accelerations and decelerations were up to ten times bigger than docking errors. The deformation of the trajectories to avoid obstacles in the reactive tests was reasonable according to the space to navigate. The results suggest that the motors required for wheel steering are smaller than the motors required for wheel rotation. The contents of the chapter are arranged as follows:

VIII.2 Architecture of the motion planning approach

Figure VI.1 shows the architecture of the motion planning approach. Where, given a sequence of reachable nodes, the trajectory planner optimizes the number of nodes and produces straight line trajectories according to section VII.5.1. The reactive module, with the current position of the robot, the local object space model and the current position of the trajectory endpoint, computes the final reference trajectory as seen in section VII.4 such that a mobile robot (e.g., input-output state feedback linearized and stabilized with a linear feedback) can follow the trajectory generated by the modules.

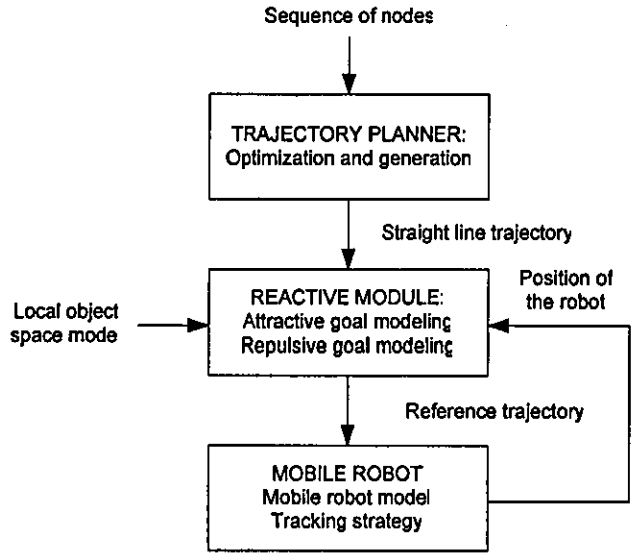


Figure VIII.1: Simulation framework general architecture for the motion planning approach

VIII.3 Motion planning simulation results

Appendices XIX to XVI show the motion planning simulation results. The motion planning approach was divided into two approaches, according to chapter VII, in deliberative and reactive motion planning. The deliberative motion planning produced a sequence of straight line trajectories (deliberative reference trajectories) that joined the initial node to the final one according to the task (see section VII.5), which were followed by the robot. The reactive motion planning approach (see section VII.4) produced robot reactions that kept the robot far from unexpected obstacles. In consequence, the robot followed the deliberative reference trajectory with certain looseness (without precision in

order to avoid obstacles), while avoided fixed obstacles (in corridors) or mobile obstacles (in open areas). The simulation tests performed to the robot were (see appendices from XIX to XXVI):

- Corridor deliberative path planning.
    - **Appendix XIX:** Line tracking and docking. Single-robot.
    - **Appendix XX:** Tracking moving backwards. Robot-trolley configuration.
    - **Appendix XXI:** Docking moving forwards. Robot-trolley configuration.
    - **Appendix XXII:** Undocking and line tracking. Single-robot.
- Corridor reactive path planning
    - **Appendix XXIII:** Single-robot.
    - **Appendix XXIV:** Robot-trolley configuration.
- Open area reactive path planning
    - **Appendix XXV:** Single-robot.
    - **Appendix XXVI:** Robot-trolley configuration.

Thus, apart from those already mentioned in chapter VI, additional comments on the corridor deliberative motion planning, corridor reactive motion planning, and open area reactive motion planning tests, are presented in the next sections.

VIII.3.1 Corridor deliberative motion planning simulation results

Figure VIII.2 shows the purposed loosely structured environment for simulation purposes. The facilities (red objects) were fixed and the mail trolleys (green objects) were arranged in a similar way to a real mail distribution centre facility. The single robot (blue object) was at rest position. In order to run the corridor deliberative path planning test, the corridors were free of obstacles and the trolleys in its respective position, where no reactive behaviour could emerge from the unexpected presence of obstacles when tracking a desired trajectory. A trolley was selected to be transferred to a new position (dark green trolley). The task required (see section VII.5.2) is represented by the arrows

(see Figure VIII.2), which define the tasks that must be done by the corresponding single-robot or robot-trolley configuration, respectively. The task sequence was the following (see appendices XIX to XXII, as well):

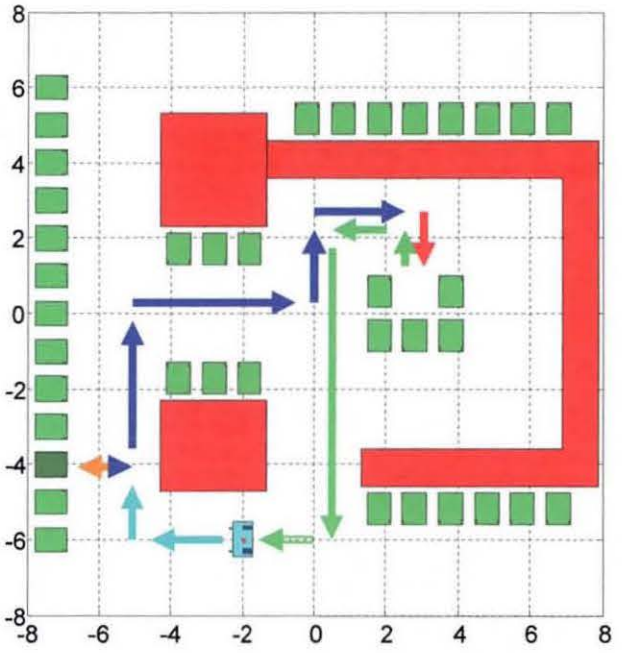


Figure VIII.2: Task sequence of the corridor deliberative path planning test

- **Appendix XIX**: Single-robot line tracking (light blue arrow) and docking (orange arrow).
- **Appendix XX:** Robot-trolley configuration tracking moving backwards (dark blue arrow).
- **Appendix XXI:** Robot-trolley configuration docking moving forwards (red arrow).
- **Appendix XXII:** Single-robot undocking and line tracking (light green arrow).

Also, Figure VIII.3 shows a zoom of the sequence of posture snapshots of each test, extracted from those appendixes. The snapshot sample time selected for each test was: 1 second for figures a), b) and d), and 0.2 second for figure c).

The velocity and acceleration parameters of the trajectories were set for all of them as follows (see section VII.5.1): $a_{max} = 1$ [m/s^2], $a_{min} = 0.1$ [m/s^2], and $v_{max} = 1$ [m/s], respectively. Besides, the distance x_1 (the distance where the "starting-ending trajectory

changing starting accelerations" changes from a_{min} to a_{max} (see Figure VII.3.c)) was set to $x_1 = 1$ [m], which allows enough distance to stabilize the robot (see appendix VIII).

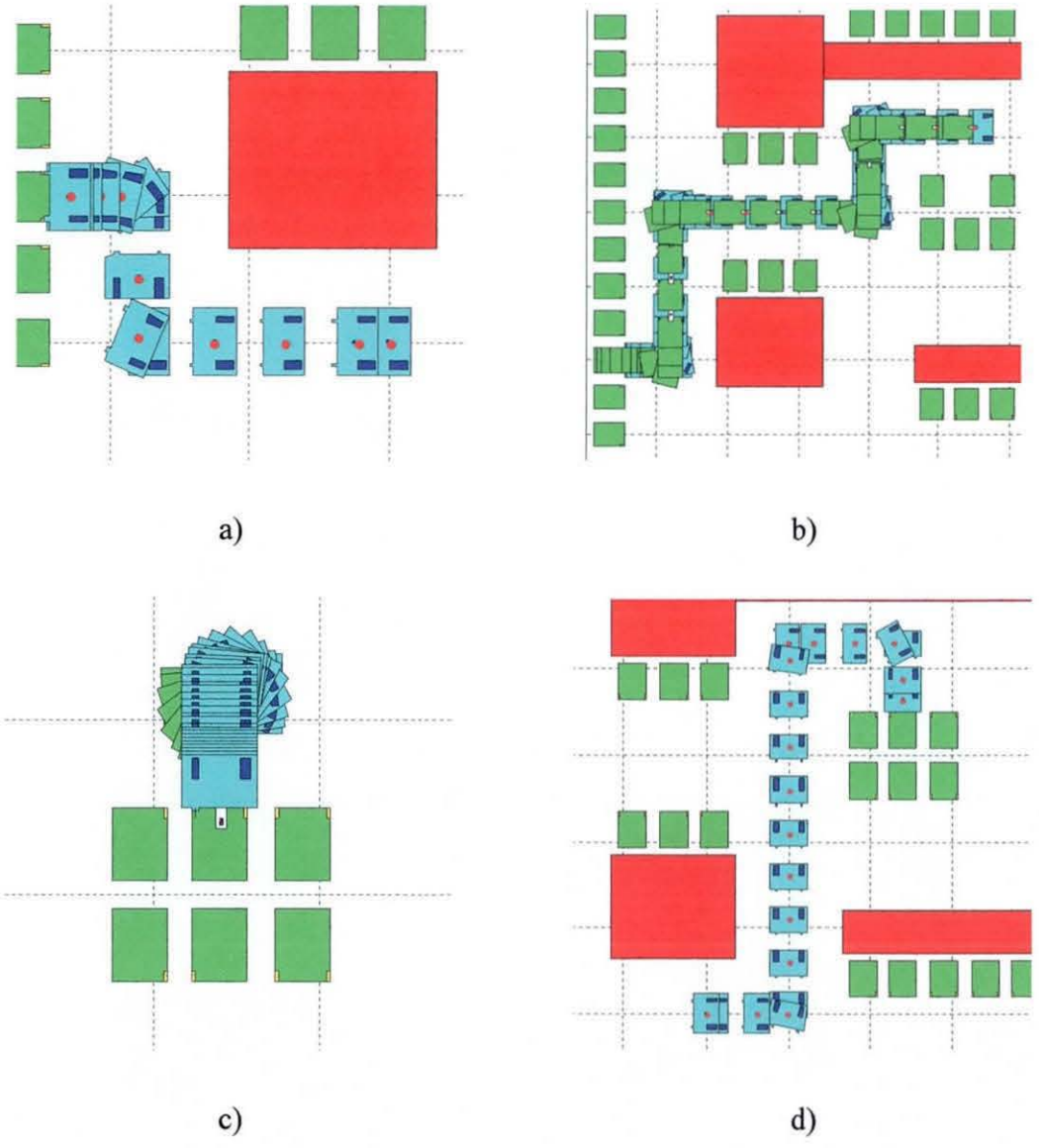


Figure VIII.3: Corridor deliberative path planning test: Sequence of snapshots.

a) Figure A.XIX.1: Line tracking and docking (Robot type(2,0)), b) Figure A.XX.1: Line tracking moving backwards (Robot type(1,1)), c) Figure A.XXI.1: Docking moving forwards (Robot type(1,1)), and d) Figure A.XXII.1: Undocking and line tracking (Robot type(2,0))

Figure VIII.4 shows the robot reference point tracking errors of each test (extracted from the appendices XIX to XXII), which showed exponential stability for constant velocity line tracking and were stable for constant acceleration and deceleration line tracking, with the following error values for each one: maximum **travelling error** of ± 4 [cm] for constant acceleration or deceleration line tracking, maximum **docking error** of ± 4 [mm] for constant acceleration or deceleration line tracking, and zero **cruising speed error** for constant velocity line tracking in steady state.

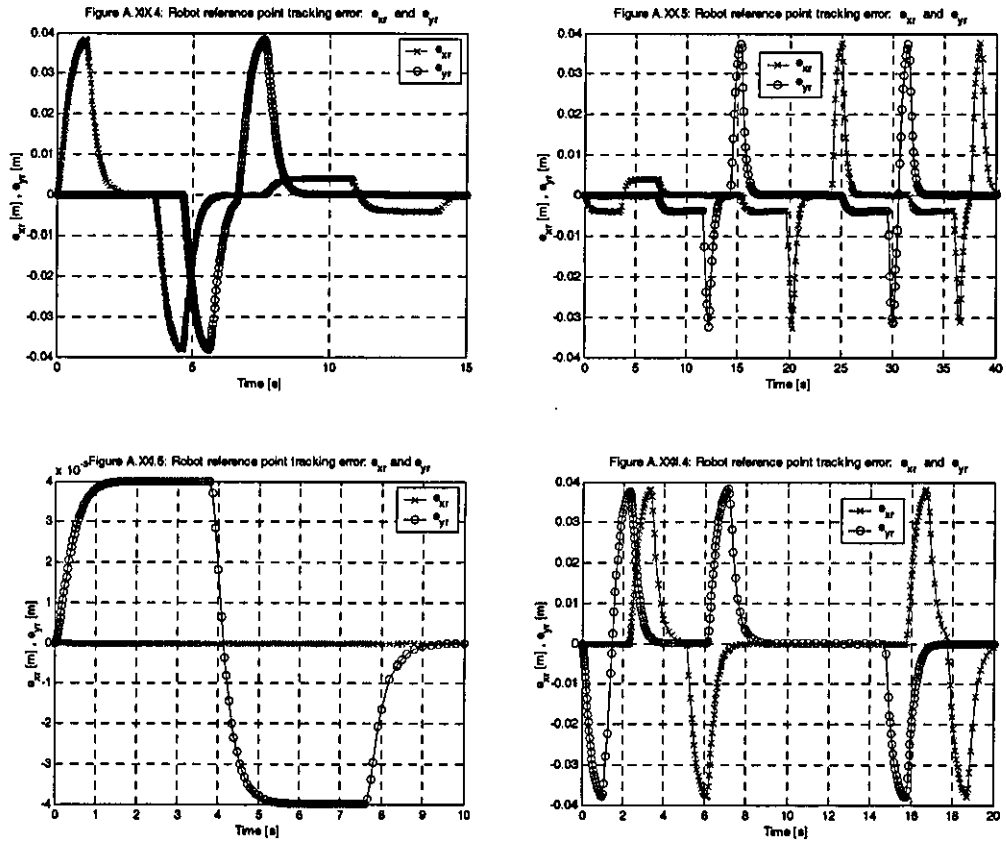


Figure VIII.4: Corridor deliberative path planning test: Robot reference point tracking errors.

Figure A.XIX.4: Line tracking and docking (Robot type(2,0)), Figure A.XX.5: Line tracking moving backwards (Robot type(1,1)), Figure A.XXI.5: Docking moving forwards (Robot type(1,1)), and Figure A.XXII.4: Undocking and line tracking (Robot type(2,0)).

Figure VIII.5 shows the wheel steering angles for the robot-trolley configuration required to accomplish the task (extracted from the appendices XX and XXI). The result showed that having a steering angle of $-135^0 \leq \alpha_{cr,ci} \leq 135^0$ for each centred steerable wheel is sufficient for accomplishing all the tasks (as in Figure VI.14 and VI.15).

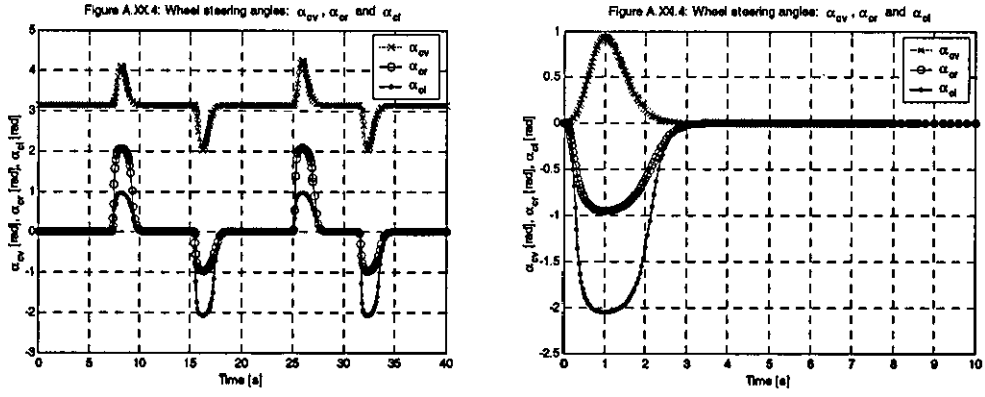


Figure VIII.5: Corridor deliberative path planning test: Wheel steering angles.

Figure A.XX.4: Line tracking moving backwards (Robot type(1,1)) and Figure A.XXI.4: Docking moving forwards (Robot type(1,1))

Figure VIII.6 shows the steering system velocities and accelerations of each test respectively (extracted from the appendices XIX to XXII), where it can be observed that the linear velocity reached in steady state was 1 [m/s] for both the single-robot (see Figure A.XIX.7) and the robot-trolley configuration (see Figure A.XX.9). The linear velocity at the vertical axle of the virtual wheel of the robot-trolley configuration can be computed using the equation (VI.6), as follows:

$$v_{cv} = r\dot{\phi}_{cv} = (d_v - d_f)\eta$$

(VIII.1)

where v_{cv} is the heading linear velocity of the virtual wheel.

On the other hand, the linear acceleration reached in steady state was 1 [m/s^2] for both the single-robot (see Figure A.XIX.8) and the robot-trolley configuration (see Figure A.XX.10), as expected. The linear acceleration at the vertical axle of the virtual wheel of the robot-trolley configuration can be computed using the equation (VI.7), as follows:

$$a_{cv} = r\ddot{\phi}_{cv} = (d_v - d_f)\dot{\eta}$$

(VIII.2)

where a_{cv} is the heading linear acceleration of the virtual wheel.

Figure VIII.7 shows the force and torque commands needed for the single-robot and robot-trolley configuration. Following the notation suggested in section IV.6 second paragraph, the force τ_{x_1} must be interpreted as the force needed to produce the required linear acceleration to the single-robot platform, where the maximum force τ_{x_1} was about ± 450 [N]. The toque τ_θ must be interpreted as the torque needed to produce the required angular acceleration of the single-robot platform, the maximum torque was about ± 350 [Nm]. The torque τ_η must be interpreted as the torque needed to compute the torques $\tau_{\phi_{cr}}$ and $\tau_{\phi_{cl}}$ required to produce rotational accelerations in the right and left centred steerable wheels of the robot-trolley configuration respectively (see equations (VI.9)). The torque τ_ζ must be interpreted as the torque needed to compute the torques $\tau_{\alpha_{cr}}$ and $\tau_{\alpha_{cl}}$ required to produce the steering accelerations in right and left centred steerable wheels of the robot-trolley configuration respectively (see equations (VI.9)).

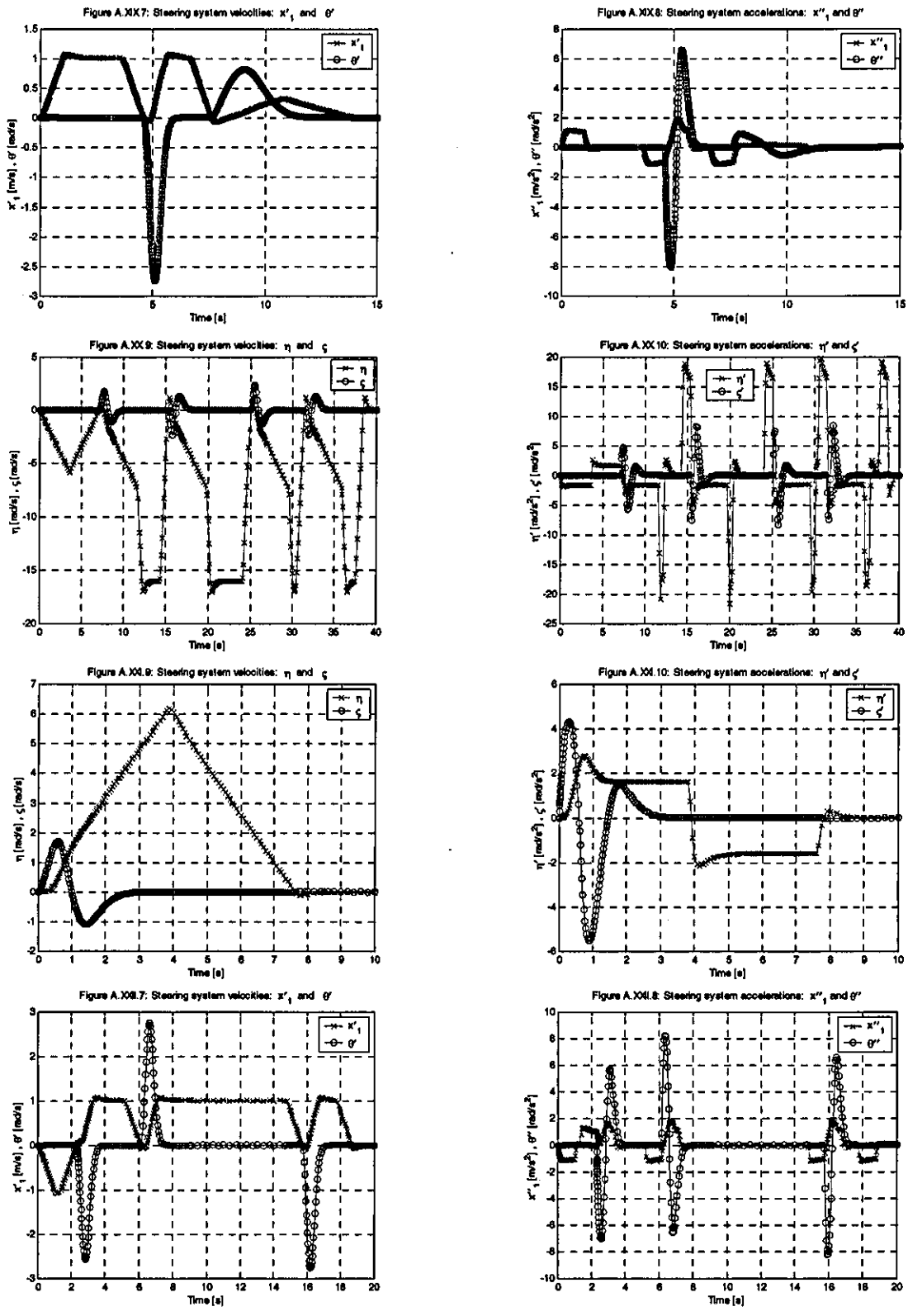


Figure VIII.6: Corridor deliberative path planning test: Steering system velocities and accelerations.

Figures A.XIX.7 and A.XIX.8: Line tracking and docking (Robot type(2,0)), Figures A.XX.9 and A.XX.10: Line tracking moving backwards (Robot type(1,1)), Figures A.XXI.9 and A.XXI.10: Docking moving forwards (Robot type(1,1)), and Figures A.XXII.7 and A.XXII.8: Undocking and line tracking (Robot type(2,0))

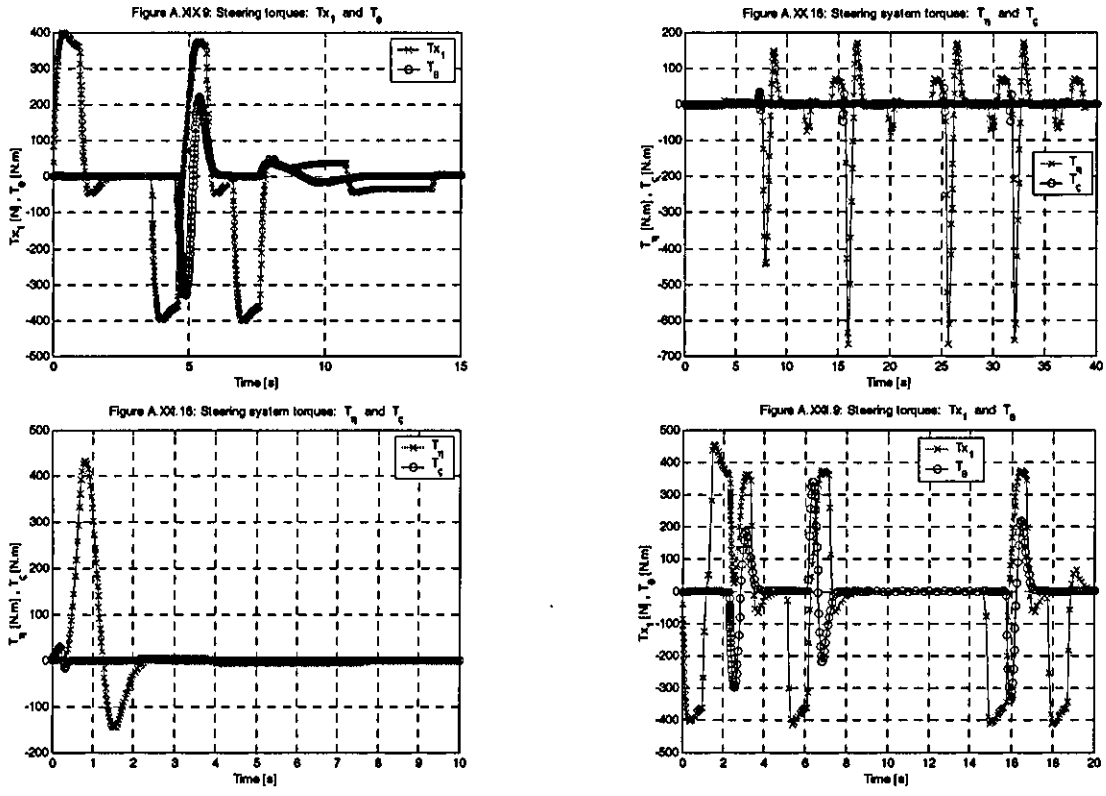


Figure VIII.7: Corridor deliberative path planning test: Steering system torques.

Figure A.XIX.9: Line tracking and docking (Robot type(2,0)), Figure A.XX.16: Line tracking moving backwards (Robot type(1,1)). Load=550kg, Figure A.XXI.16: Docking moving forwards (Robot type(1,1)). Load=550kg, and Figure A.XXII.9: Undocking and line tracking (Robot type(2,0)).

Figure VIII.8 shows the motor torques and motor power for wheel steering demanded by the task, whose maximum amounts were about 7 $[N \cdot m]$ and 20 $[w]$.

Figure VIII.9 shows the motor torques and motor power for wheel rotation demanded by the task, whose maximum torques were about 100 $[N \cdot m]$ and maximum power about 500 $[w]$. Besides, Figure VIII.10 shows the steering and rotational velocities required to carry out the task, whose maximum values, considering both steering and rotational velocities, were no more than 10 $[rad/seg]$.

The characteristics of the motors needed for this kind of task are presented in Table VIII.1. These values must be read with care, because the torques associated to the motors and transmission inertia, joint frictions, disturbances, unmodelled dynamics, and unmodelled external forces and moments need to be incorporated in them. But, they can be considered later on when designing the controllers for the motors, whose design is out of the scope of this thesis.

Table VIII.1: Motor specifications for the deliberative motion planning.

MOTOR ESPECIFICATIONS	Torque [*Nm*]	Power [*W*]	Rotational velocity [*rad / seg*]
Steering motors	7	20	10
Rotational motors	100	500	10

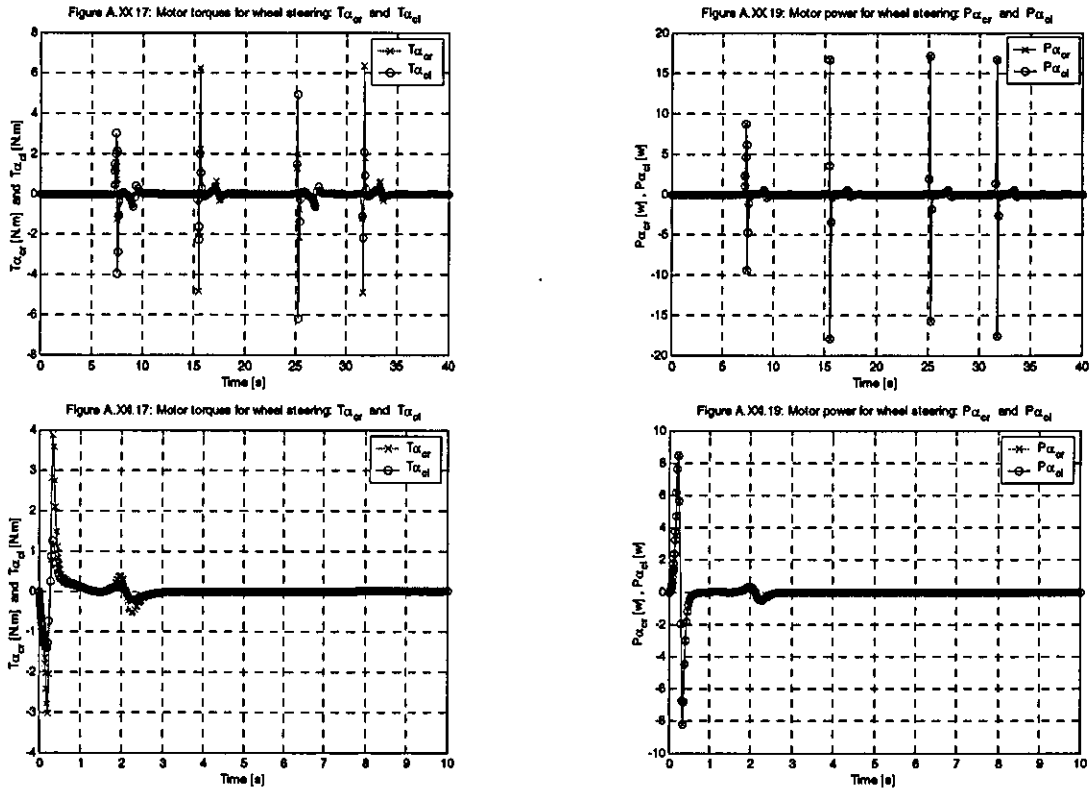


Figure VIII.8: Corridor deliberative path planning test: Motor torques and motor power for wheel steering. Figures A.XX.17 and A.XX.19: Line tracking moving backwards (Robot type(1,1)). Load=550kg, Figures A.XXI.17 and A.XXI.19: Docking moving forwards (Robot type(1,1)). Load=550kg.

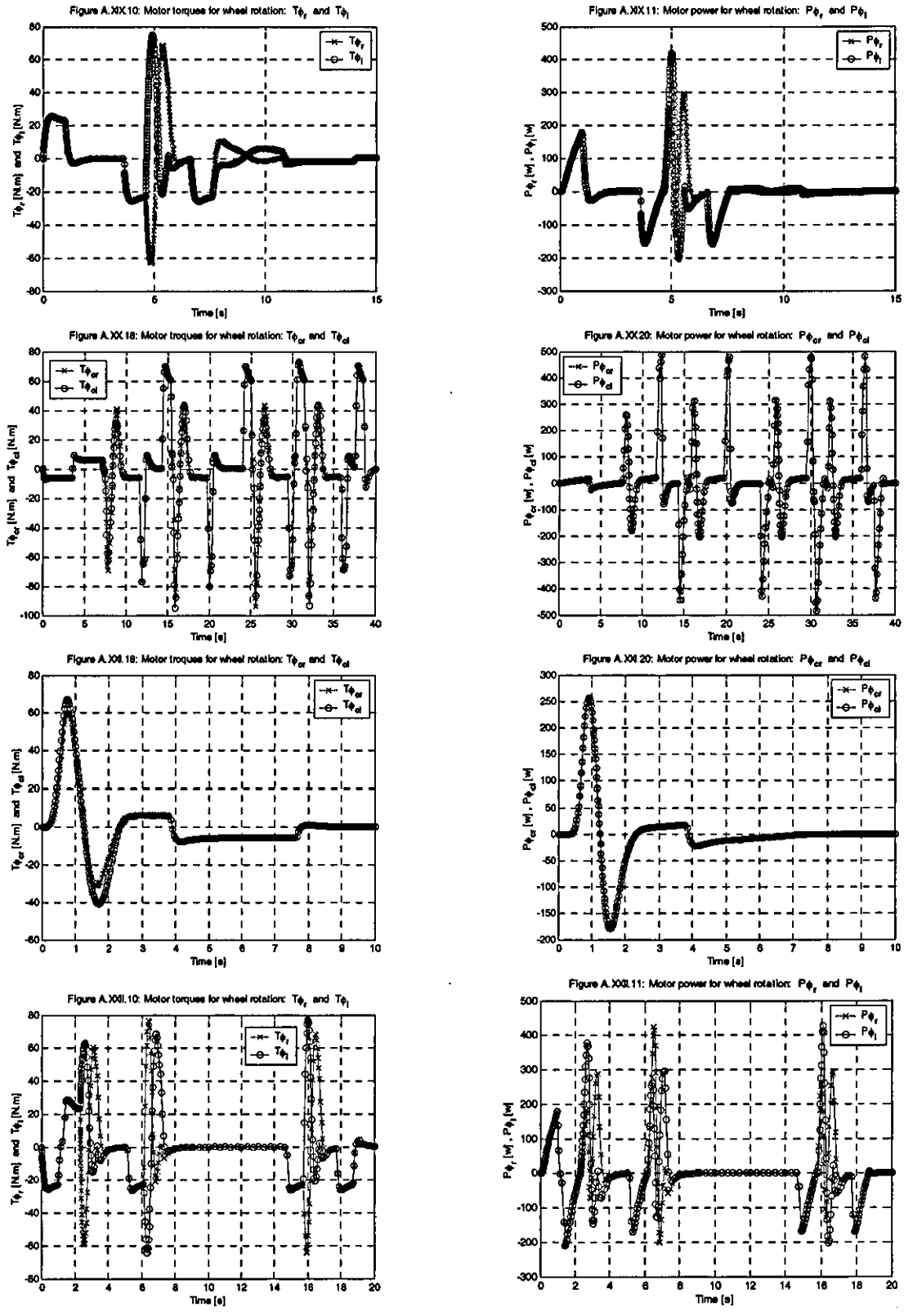


Figure VIII.9: Corridor deliberative path planning test: Motor torques and motor power for wheel rotation. Figures A.XIX.10 and A.XIX.10: Line tracking and docking (Robot type(2,0)), Figures A.XX.18 and A.XX.20: Line tracking moving backwards (Robot type(1,1)). Load=550kg, Figures A.XXI.18 and A.XXI.20: Docking moving forwards (Robot type(1,1)). Load=550kg, and Figures A.XXII.19: Undocking and line tracking (Robot type(2,0)).

Figure VIII.10: Corridor deliberative path planning test: Remaining generalized velocities.

Figure A.XIX.6: Line tracking and docking (Robot type(2,0)), Figures A.XX.7 and A.XX.8: Line tracking moving backwards (Robot type(1,1)), Figures A.XXI.7 and A.XXI.8: Docking moving forwards (Robot type(1,1)), and Figure A.XXII.6: Undocking and line tracking (Robot type(2,0)).

VIII.3.2 Reactive motion planning simulation results

The appendices XXIII to XXVI show the corridor and open area reactive path planning simulation results (where the figures of this section were extracted) for the single-robot and robot-trolley configuration respectively. **Figure VIII.11** shows the sequence of posture snapshots of the reactive path planning simulation tests for the single-robot and robot-trolley configuration respectively, where the robot loosely followed a straight line trajectory while avoiding any collision with the obstacles (trolleys out of position or objects moving along with the robot). The snapshot sample time selected for each test

was: 0.4 seconds for Figure A.XXIII.1, 0.5 seconds for Figure A.XXIV.1, 1 second for Figure A.XXV.1, and 1 second for Figure A.XXIV.1. Also, the colour of the robot and moving objects in the open area test was different each sample time in order to identify the position of each object at that time by the colour.

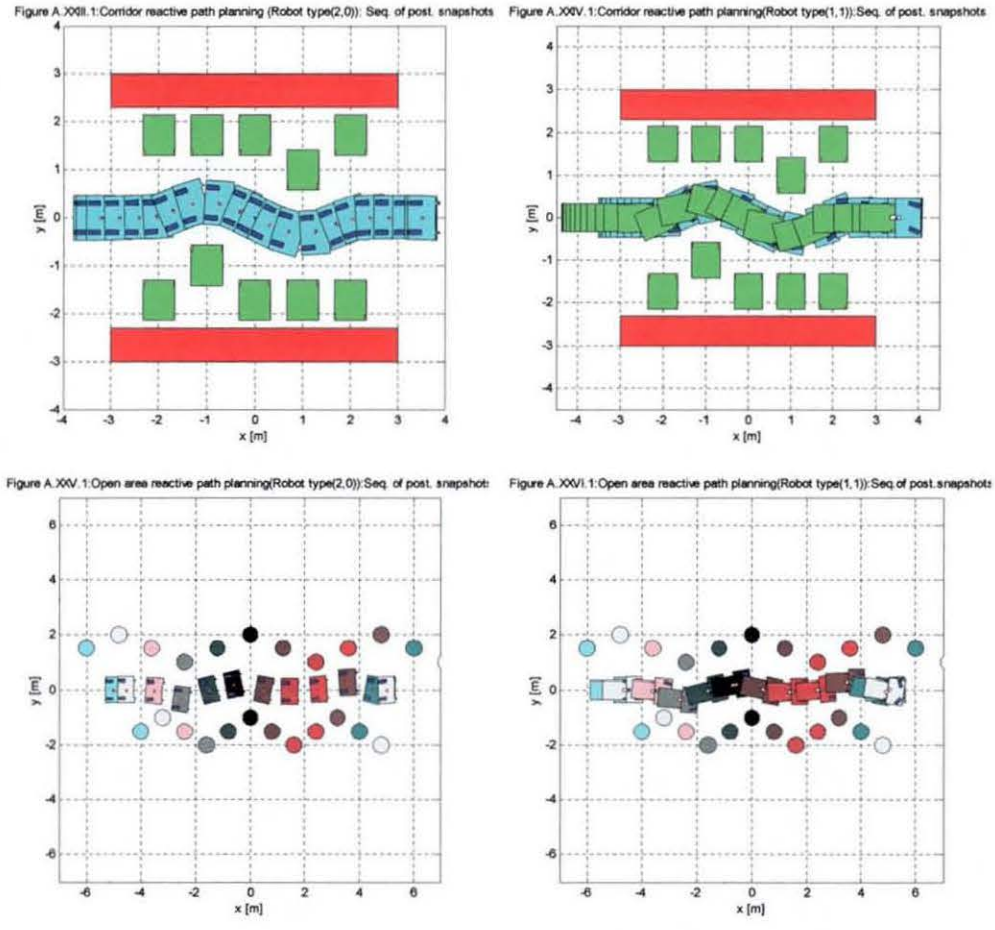


Figure VIII.11: Reactive path planning: Sequence of posture snapshots.

<u>Corridor</u>: Figure A.XXIII.1: Single-robot and Figure A.XXIV.1: Robot-trolley (moving backwards).
<u>Open area</u>: Figure A.XXV.1: Single-robot and Figure A.XXVI.1: Robot-trolley (moving backwards)

Table VIII.2 shows the reactive path planning test parameters: robot reference point initial position, robot reference point final position, posture of the fixed objects (trolleys), and trajectory of the moving objects (people). Each object was modelled with the following parameters (see equation (VII.11)): $\eta = 0.01$, $\rho_r = 1.3$, and $\rho_0 = 1$, which define objects with radius of 1 meter, with repulsive goal function influence of 1.3 meters, and constant gain of 0.01. The effect of walking along with the robot, where the moving objects are simultaneously travelling with the robot, was created by fixing the abscissa velocity component of the moving objects such that, the upper object was faster than the lower and positioned at a distance of +1.5 and -1.5 meters of the x-axis,

respectively. The amplitude of the oscillating movement on the ordinate was set at 0.5 meters in both trajectories.

Table VIII.2: Reactive path planning test parameters

	Robot reference point		Objects
	Initial position (x_r, y_r)	**Final position** (x_r, y_r)	
Corridor test	(-3.5, 0)	(3.5, 0)	Fixed object postures (x_l, y_l, θ_l) (Trolleys) (-2, 1.725, π /2), (-1, 1.725, π /2), (0, 1.725, π /2), (1, 1, π /2), (2, 1.725, π /2), (-2, -1.725, - π /2), (-1, 1, - π /2), (0, -1.725, - π /2), (1, -1.725, - π /2), (2, -1.7250, - π /2).
Open area test	(-5, 0)	(5, 0)	Moving object trajectories (people) Upper object $x_L = 1.2t - 6$ $y_L = 0.5\sin(\pi \cdot t/2) + 1.5$ Lower object $x_L = 0.8t - 4$ $y_L = 0.5\sin(\pi \cdot t/2) - 1.5$

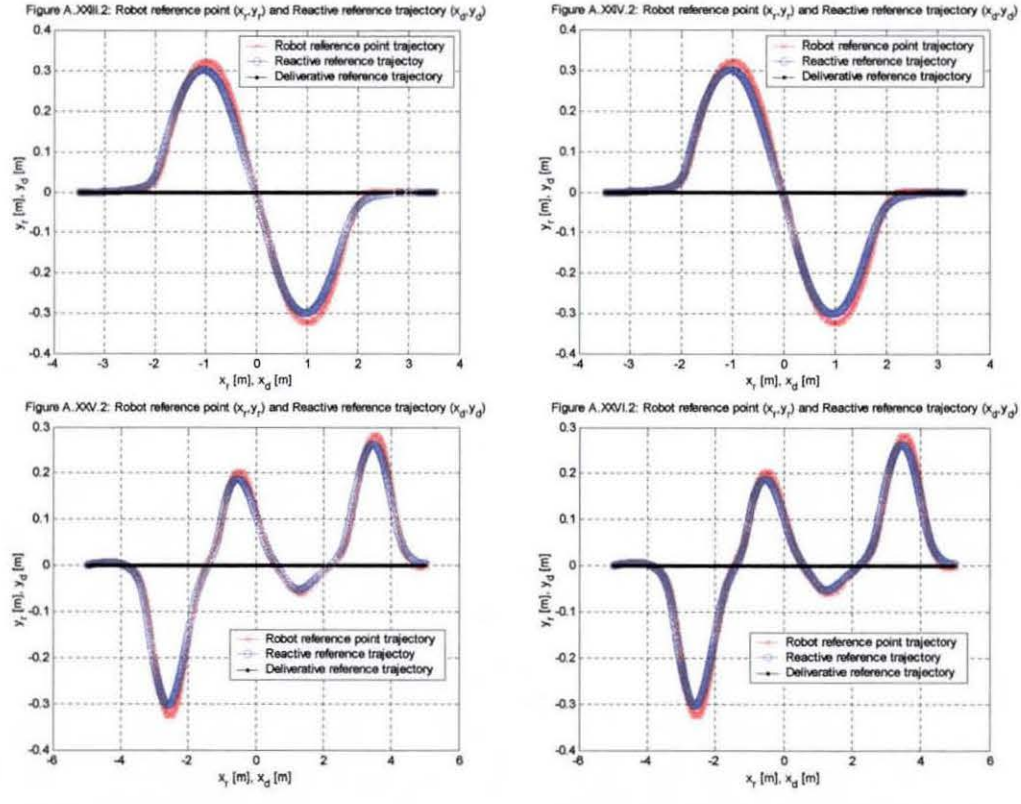


Figure VIII.12: Reactive path planning: 2D trajectory plotting.

Corridor: Figure A.XXIII.2: Single-robot and Figure A.XXIV.2: Robot-trolley (moving backwards).
Open area: Figure A.XXV.2: Single-robot and Figure A.XXVI.2: Robot-trolley (moving backwards)

On the other hand, Figure VIII.12 shows the robot reference point trajectory, the reactive trajectory, and the deliberative trajectory for the single-robot and the robot-trolley configuration, respectively, where the robot loosely followed a straight line trajectory (deliberative reference trajectory), when tracked the reactive trajectory produced by the reactive module to avoid any collision with the obstacles. For the single-robot, the deliberative reference trajectory was a "starting-ending trajectory". For the robot-trolley configuration, the deliberative reference trajectory was a "Starting-ending trajectory changing starting accelerations" for corridor path planning, and "starting-ending trajectory" for open area path planning (see Figure VII.3). The velocity and acceleration parameters were the following: $v_{max} = 1$ [m/s], $a_{max} = 1$ [m/s^2], and $a_{min} = 0.1$ [m/s^2], respectively (see section VI.3.2). Besides, Figure VIII.13 shows the robot reference point tracking errors for the single-robot and the robot-trolley configuration for these tests, which were less than ± 4 [cm].

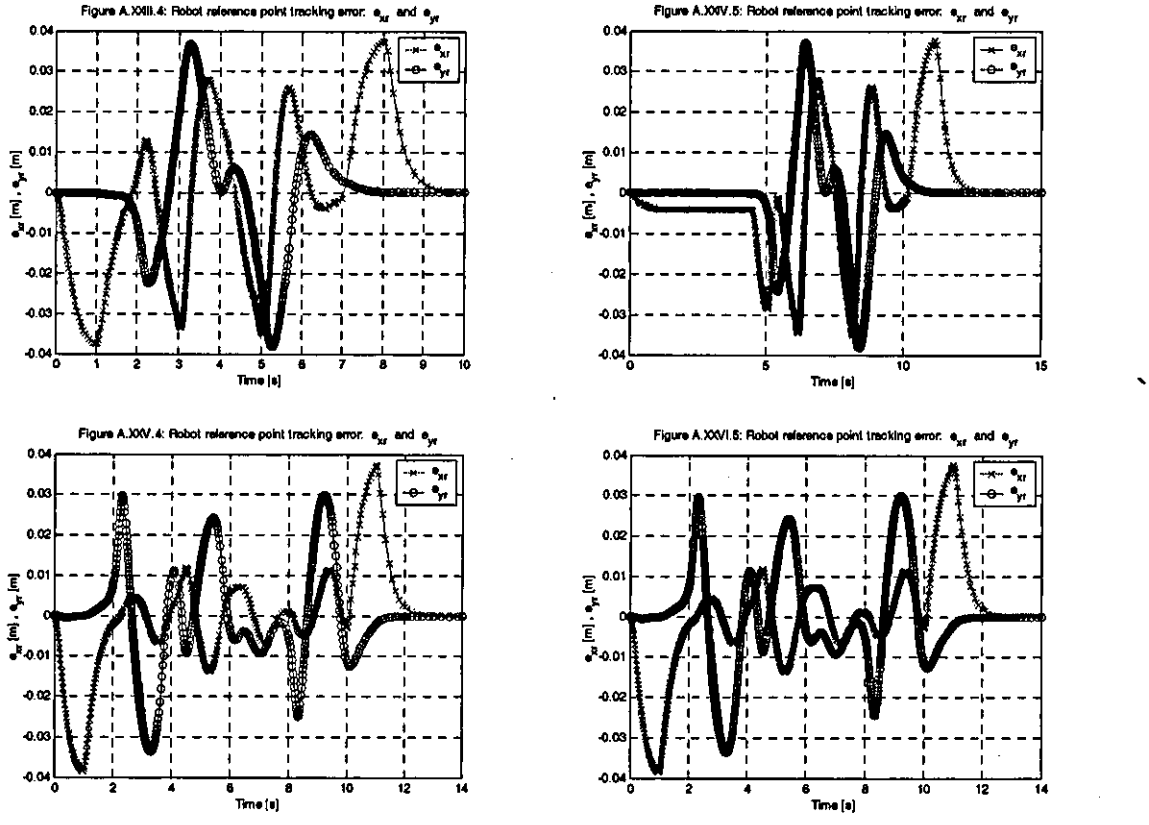
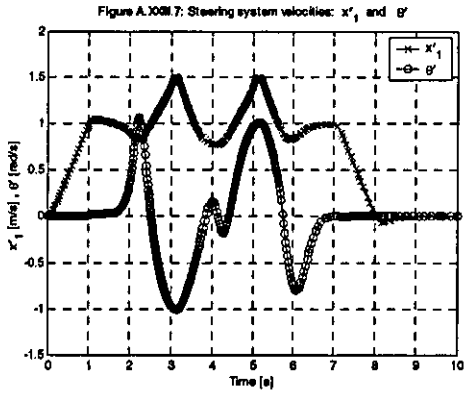
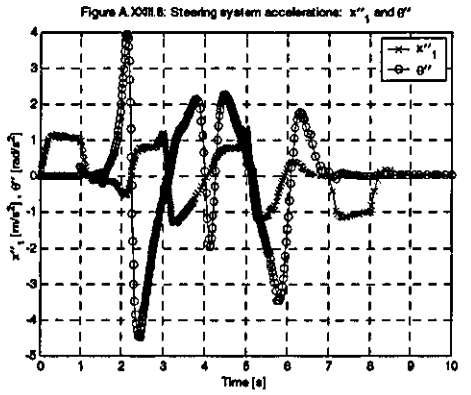
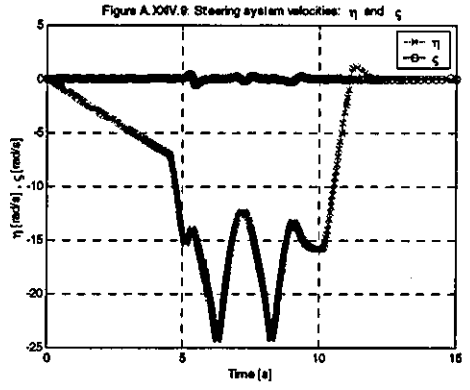


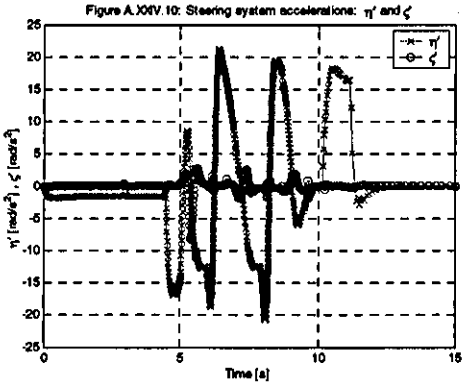
Figure VIII.13: Reactive path planning: Robot reference point tracking errors.

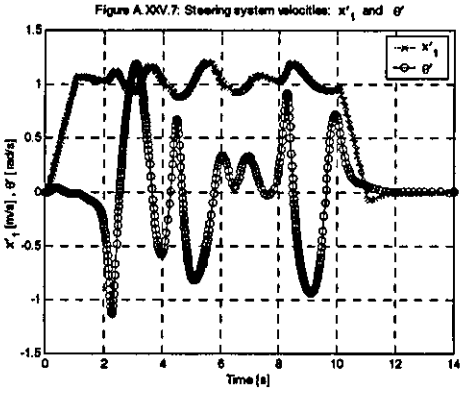
Corridor: Figure A.XXIII.4: Single-robot and Figure A.XXIV.5: Robot-trolley (moving backwards).
Open areas: Figure A.XXV.4: Single-robot and Figure A.XXVI.5: Robot-trolley (moving backwards).

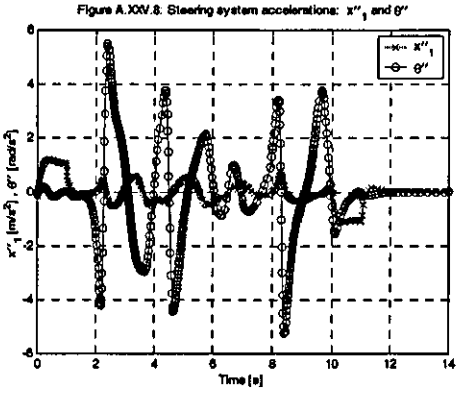


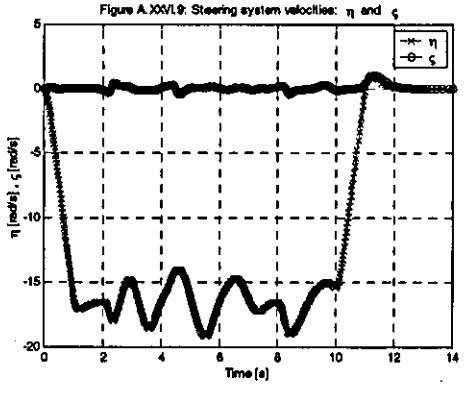












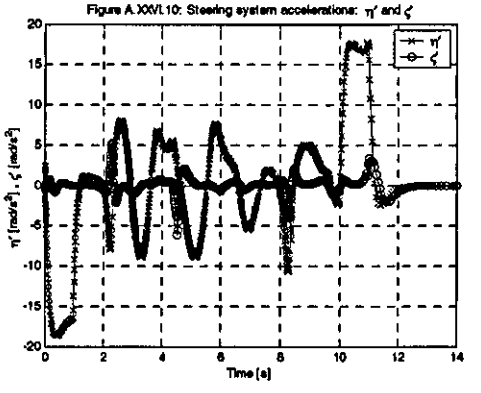


Figure VIII.14: Reactive path planning: Steering system velocities and accelerations.

<u>Corridor:</u> Figures A.XXIII.7 and A.XXIII.8: Single-robot and Figures A.XXIV.9 and A.XXIV.10: Robot-trolley (moving backwards). <u>Open area:</u> Figures A.XXV.7 and A.XXV.8: Single-robot and Figures A.XXVI.9 and A.XXVI.10: Robot-trolley (moving backwards)

Figure VIII.14 shows the steering system velocities and accelerations of each test respectively, where it can be observed that the maximum linear velocity reached was about 1.5 [m/s] for both the single-robot and the robot-trolley configuration. Notice that for the robot-trolley configuration, this velocity can be computed using the equation (VIII.1). On the other hand, the linear acceleration reached was 1.2 [m/s^2] for both the single-robot and the robot-trolley configuration, notice that for the robot-trolley configuration this linear acceleration can be computed using the equation (VIII.2).

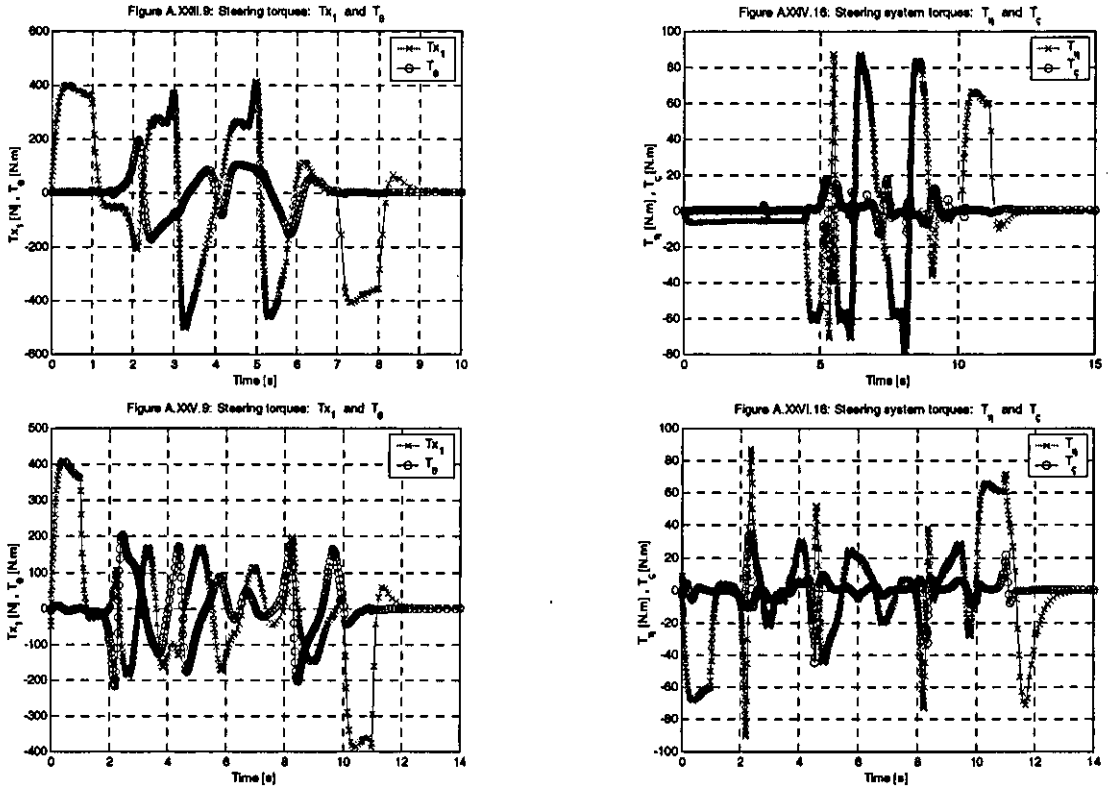


Figure VIII.15: Reactive path planning: Steering system torques.

Corridor: Figure A.XXIII.9: Single-robot and Figure A.XXIV.16: Robot-trolley (moving backwards). Load=550kg. Open area: Figure A.XXV.9: Single-robot and Figure A.XXVI.16: Robot-trolley (moving backwards). Load=550kg.

Figure VIII.15 shows the steering torque commands needed for the single-robot and robot-trolley configuration for the reactive path planning test. The force τ_{x_1} and torque τ_η must be interpreted as the force and torque needed to produce the required linear acceleration in the robot platform and in the virtual steerable wheel, respectively. For the single-robot, the maximum force τ_{x_1} was about ± 500 [N]. For the robot-trolley configuration, this torque τ_η must be interpreted as the torque needed to compute the rotational torques $\tau_{\phi_{cr}}$ and $\tau_{\phi_{cl}}$ of the centred wheels (see equations (VI.9)). The toque

τ_θ must be interpreted as the torque needed to produce the required angular acceleration of the robot platform and the torque τ_ζ must be interpreted as the torque needed to compute the steering torques $\tau_{\alpha_{cr}}$ and $\tau_{\alpha_{cl}}$ of the centred wheels (see equations (VI.9)), where for the single-robot, the maximum torque τ_θ was about ± 220 [Nm].

Figure VIII.16 shows the motor torques and motor power for wheel steering demanded by the task, whose maximum amounts were no more than 7 $[N \cdot m]$ and 10 $[w]$.

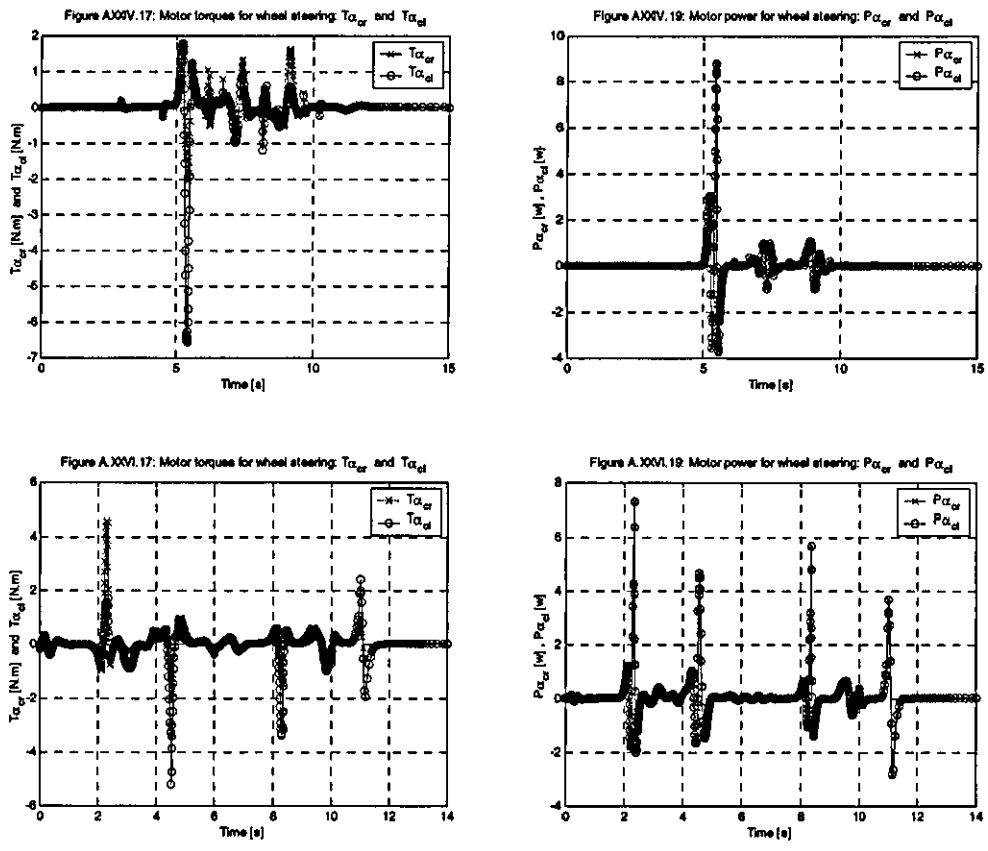


Figure VIII.16: Reactive path planning: Motor torques and motor power for wheel steering.

Corridor: Figures A.XXIV.17 and A.XXIV.19: Robot-trolley moving backwards. Open areas: Figures A.XXV.17 and A.XXV.19: Robot-trolley moving backwards. Load=500kg.

Figure VIII.17 shows the motor torques and motor power for wheel rotation demanded for the task, whose maximum torques were about 110 $[N \cdot m]$ and maximum power about 1000 $[w]$. Besides, Figure VIII.18 shows the motor steering and rotational velocities required to carry out the task, whose maximum values, considering both steering and rotational velocities, were no more than 6 $[rad/s]$ for wheel steering, and no more than 15 $[rad/s]$ for wheel rotation.

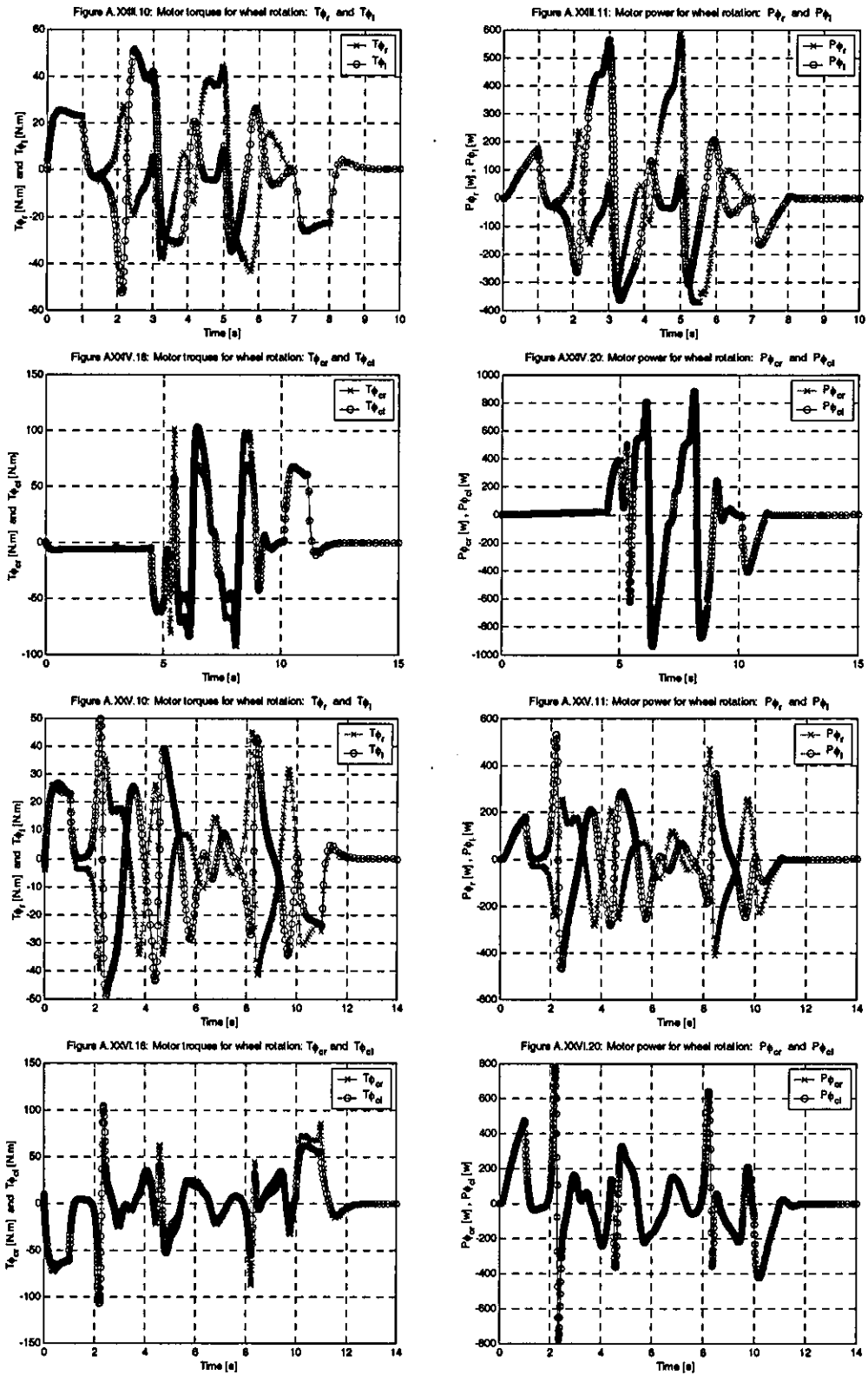
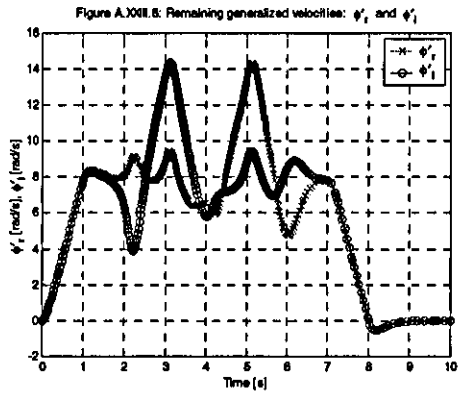
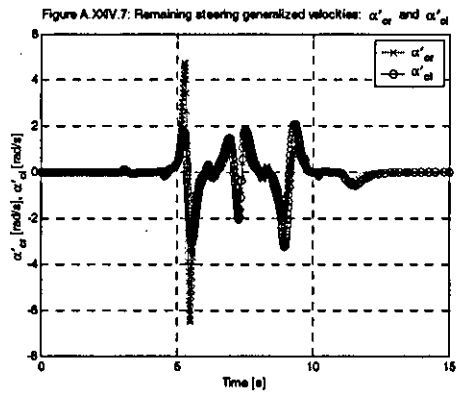
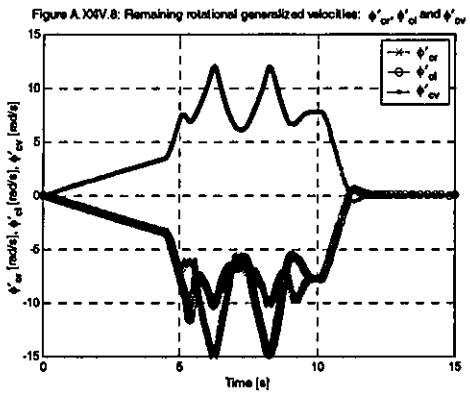


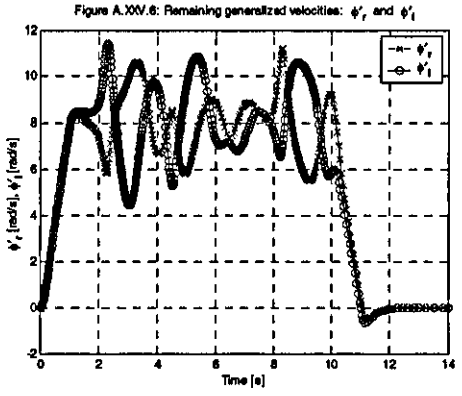
Figure VIII.17: Reactive path planning: Motor torques and motor power for wheel rotation.

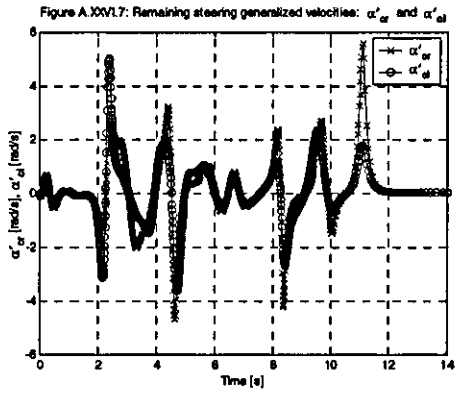
<u>Corridor:</u> Figures A.XXIII.10 and A.XXIII.11: Single-robot and Figures A.XXIV.18 and A.XXIV.20: Robot-trolley (moving backwards). <u>Open area:</u> Figures A.XXV.10 and A.XXV.11: Single-robot and Figures A.XXVI.18 and A.XXVI.20: Robot-trolley (moving backwards)











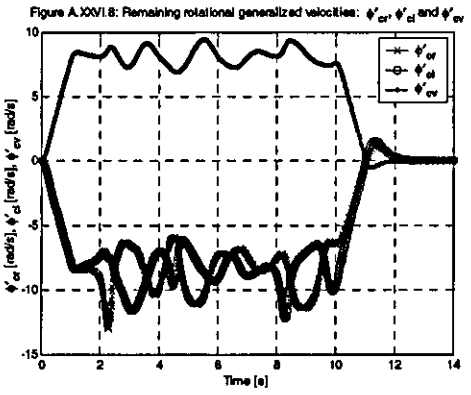


Figure VIII.18: Reactive path planning: Motor remaining generalized velocities.

Corridor: Figure A.XXIII.6: Single-robot and Figures A.XXIV.7 and A.XXIV.8: Robot-trolley (moving backwards). Load=550kg. Open area: Figure A.XXV.6: Single-robot and Figures A.XXVI.7 and A.XXVI.8: Robot-trolley (moving backwards). Load-550kg.

The characteristics of the motors needed for this kind of task are presented in Table VIII.3. As was cited in section VIII.3.1, these values must be considered with care, because they need to be incorporated with the torques associated to the motors inertia, joint frictions, disturbances, unmodelled dynamics, and external forces and moments. But, they can be considered later on when designing the controllers for the motors, whose design is out of the scope of this thesis.

Table VIII.3: Motor specifications for the reactive motion planning.

MOTOR ESPECIFICATIONS	Torque $[N \cdot m]$	Power [*Watts*]	Rotational velocity $[rad/seg]$
Steering motors	7	10	6
Rotational motors	110	1000	15

VIII.4 Summary

The single-robot and robot-trolley configuration were tested in a loosely structured environment specially designed to simulate the deliberative and reactive behaviour of the robot. In general, the tracking errors were about the following: maximum **travelling error** of ± 4 [cm] for constant acceleration or deceleration line tracking, maximum **docking error** of ± 4 [mm] for constant acceleration or deceleration line tracking, and zero **cruising speed error** for constant velocity line tracking in steady state. The wheel steering angles required for the mechanisms to accomplish the task were about $-135^0 \leq \alpha_{cr,cl} \leq 135^0$. The output characteristics required for the geared motors for wheel steering were: torque of 7 [Nm], power of 20 [W], and speed of 10 [rad/s]. The output characteristics for the geared motors for wheel rotation were: torque of 110 [Nm], power of 1000 [W], and speed of 15 [rad/s]. In general, the motion planning tests were satisfactory and all the results seem to be suitable for design purposes.

IX Conclusions, contributions and further work

IX.1 Conclusions

The development of a nonholonomic semiautonomous wheeled mobile robot able to apply stable pushes and pulls when holding a nonholonomic wheeled load firmly in two points while it moves about a loosely structured environment, arrived at the application of different techniques. The most important techniques used to solve the problem were the kinematic, dynamic, and state space representation modelling, the standard theory of nonlinear systems and nonlinear control, and the motion planning theory. As a result, two theoretical frameworks were introduced in this research work. The first one (chapter IV), for modelling conventional wheeled mobile robots, which presented the holonomic parameterized centred angular coordinates for coordinating the steering of multiple steerable wheels and the remaining generalized coordinates, which describe the evolution of the generalised coordinates of unmotorized joints and the motor control commands for motorized joints. The second one (chapter VII), for planning the motion of wheeled mobile robots in loosely structured environments based on a goal domain approach, which endows the robot with the capability avoid obstacles while tracking a desired trajectory.

The reformulation of the wheel kinematic descriptions of conventional wheeled mobile robots produced a set of kinematic constraints with a minimal number of variables, letting us introduce the external and internal kinematic models of the robot in a simpler way (see sections IV.3.4.b and IV.3.4.c).

The external kinematics model expressed by the tangent space of its configuration coordinates, showed to have holonomic constraints (integrable kinematic constraints) and nonholonomic constraints (nonintegrable kinematic constraints) (see sections V.6.1.a and V.6.2.a).

The absence of the centred angular generalized velocities $\dot{\alpha}_c$ in the kinematic constraints showed that the generalized velocities $\dot{\alpha}_c$ do not produce generalized reaction forces. As a result, the generalized velocities $\dot{\alpha}_c$ cannot produce accelerations on the robot's body.

Thus, when the generalized velocities $\dot{\alpha}_c$ are nonzero, and the other generalized velocities $\dot{\xi}$, $\dot{\alpha}_{oc}$ and $\dot{\phi}$ are zero, the posture of the robot remains stable. Also, The presence of the generalized coordinates θ, α_c, and α_{oc} in the kinematic constraints showed that their value defines the direction of the reaction forces produced by the generalized velocities $\dot{\xi}$ (see section IV.4.2 second paragraph and section VI.3.5 last paragraph).

The internal kinematics expressed by the tangent space of its internal configuration coordinates, showed that the kinematic properties of conventional wheeled mobile robots satisfy the planar motion of rigid bodies. Therefore, the linear velocity vector of any point on the robot frame is perpendicular to the straight line joining the point at the instant centre of rotation (ICR) at every instant (see section IV.4.2). Thus, when a mobile robot is equipped with more than r_s centred steerable wheels, the motion of extra wheels (remaining centred angular coordinates) must be coordinated to guarantee the existence of the instantaneous centre of rotation (ICR) at every time (see section IV.5 and V.6.2.a, and Figure VI.14 and Figure VI.15).

The dynamic modelling of conventional wheeled mobile robots showed to have several representations that can emerge from the first dynamic equation of motion obtained at the configuration space. The following aspects are worthily to mention: 1) Depending on the steering system selected to find the configuration kinematic state space representation of the robot, other dynamic equations of motion at the steering space can be obtained (se section IV.6.4). 2) The motor configuration, associated to the torque input vector of the dynamic equation of motion at the steering space, introduced new torque variables associated to the parameterized centred angular generalized velocities $\dot{\alpha}_p$, useful for selecting the correct motor configuration and computing the required torques for that kind of steerable wheels (see section IV.6.5 and the end of section VI.3.5).

The posture dynamical state space representation of restricted mobility robots is a differentially flat system, which can be expressed by a set of outputs y (flat outputs) such that the state and the input can be expressed algebraically in terms of y and a certain number of its derivates (see sections IV.8, V.6.1.d and V.6.2.d).

The modelling and the simulation of the system was found to be a very effective tool in defining and testing the system parameters in a modular way (see section VI.2). The use of the nonlinear tools to model and control the robot provided a medium to linearize the system model and control a point defined somewhere on the robot platform (see sections IV.8, V.6.1.d and V.6.2.d). On the other hand, the selection of the single-robot and robot-trolley configurations faced three basic problems: symmetric configurations, highest degree of mobility, and no singular configurations. The symmetric configurations permitted to manoeuvre in less space and have less turning radius, which imply more configuration space to navigate (see section V.3.1). The highest degree of mobility was associated to the selection of the best wheel configurations, which allowed to manoeuvre using simple motion planning approaches and selecting the correct motorization of the robot (see sections V.3 and V.5). The selection of non-singular motor configurations allowed to select the best robot wheel configuration (robot type(2,0)) and robot-trolley configuration (robot type(1,1)) according to the task (see sections V.3 and V.5). The selected single-robot platform is a flexible platform that can face the changes needed to adapt itself to the different situation imposed by the task of embodying and disembodying a wheeled load (see section VIII.3.1).

The linearized model of the single-robot and the robot-trolley configurations resulted in being linear models with stable unobservable zero dynamics, but not asymptotically stable, with internal dynamics exhibiting unstable behaviour when moving backwards (see sections V.6.1.d and V.6.2.d).

The addition of a virtual centred steerable wheel in the robot-trolley to indirectly control an specific point on the robot-trolley platform, linearize the robot model, coordinate the motorized centred steerable wheels, and control the movements when going forwards and backwards led us to the solution to the problem of tracking a desired trajectory (see sections V.5.2 and V.6.2).

The definition of trajectories in local coordinates, on a relative time framework, and transformed in absolute coordinates proved to be a useful flexible tool for the motion planning approach base on piecewise trajectories (see sections VI.3.1, VI.3.4 and VI.3.5).

The single-robot and robot-trolley configuration tests showed that straight lines and circular arcs trajectories can be used to design motion planning approaches, having relatively low robot reference point tracking errors (see Figure VI.10). The orientation of

the robot was shown to be asymptotically stable for line tracking and stable or bounded for circle tracking. The orientation of the centred steerable wheels was shown to be stable (see Table VI.5 and Table VI.8). The starting rotational torque and power demands are higher when the robot is misaligned with respect to the direction of the trajectory (see Figures A.II.10, A.II.11, A.VIII.18, A.VIII.20, A.XIV.18 and A.XIV.20). The torques and power demanded showed that the robot-trolley configuration was more efficient (compared to the single-robot), since it did not demand power when moving in steady state or on a circular trajectory of constant velocity (compare figure A.VI.11 with A.XII.20 and A.VIII.20).

The motion planning approach useful for avoiding static and moving obstacles, while the robot is tracking a desired trajectory or stabilized about an equilibrium point, was defined in a goal domain space with reactive and deliberative motion planning approaches. The reactive motion planning approach was constituted of both goal reaching and avoiding obstacles, and was based on goals and tracking a desired trajectory. The deliberative motion planning approach was defined for loosely structured environments, and constituted of straight line trajectories defined in a relative time domain (see chapter VII).

The motion planning approach presented in this research work made the robot behave itself in a semiautonomous way, semidependent on the motion planner, in a loosely structured environment (see section II.2). This approach was based on "artificial goals" in a "goal domain space." In consequence, it is not necessary to find a potential function, being only necessary to define the "attractive and repulsive goals" as smooth functions. Also, this approach can be defined as a "nonholonomic navigation approach with goals and tracking" (see chapter VII)

The reference trajectory generator based on piecewise straight lines, defined in a relative time framework and computed based on the distance between nodes (see section VII.5.1), produced trajectories that required less space to navigate. Also, in order to keep the docking tracking errors lower than travelling tracking errors, the docking acceleration and deceleration were set lower than travelling (see Figure VIII.4). The single-robot, tested on constant acceleration line tracking, constant deceleration line tracking, and constant velocity line tracking showed that it was suitable for these kind of trajectory tracking tasks (see appendices II, III, and IV). The robot-trolley configuration, tested on constant deceleration line tracking and constant velocity line tracking (see appendices XV and

XVI), showed that it was suitable for this kind of trajectory tracking tasks, but for constant acceleration line tracking it did not, because it can demand both high motor torques and high motor power for wheel rotation when tracking a new trajectory (see appendix XIV). So, the suitable trajectory is that that starts with a low acceleration command and speed up after stabilizing the orientation of the robot-trolley configuration (see sections VI.3.4 and VI.3.5).

The single-robot and the robot-trolley configuration models were tested on the following two simulation tasks: corridor deliberative path planning test and reactive motion planning test. The tests performed were oriented to evaluate the performance of the robot in these tasks, and to find the design parameters of the motors. In the deliberative test the robot reference point tracking errors showed exponential stability for constant velocity line tracking and were stable for constant acceleration and deceleration line tracking (see section VIII.3.1). The corridor and open area reactive path planning simulation results, for the single-robot and robot-trolley configuration respectively, showed that the robot loosely followed a straight line trajectory while avoiding any collision with the obstacles (trolleys out of position or objects moving along with the robot) (see section VIII.3.2).

The simulation results confirmed that it is possible to develop that kind of semiautonomous wheeled mobile robot, for material handling applications, able to apply stable pushes and pulls when holding a wheeled load firmly (mail trolleys), while it moves them about a loosely structured environment and between the load and unload stations. Since, the tracking errors were about the following (see Figure VIII.4 and Figure VIII.13): travelling error ± 4 [cm], docking error ± 4 [mm], and zero cursing speed error in steady state. The wheel steering angles required for the mechanisms to accomplish the task was about $-135^0 \leq \alpha_{cr,cl} \leq 135^0$ (see Figure VIII.8). The output characteristics required for the geared motors for wheel steering were: torque of 7 [Nm], power of 20 [W], and speed of 10 [rad/s]. The output characteristics for the geared motors for wheel rotation were: torque of 110 [Nm], power of 1000 [W], and speed of 15 [rad/s] (see Table VIII.1 and Table VIII.3).

As a general conclusion, the goal of this research has been met, the research questions were satisfactorily answered and the proposed platform, unifying theory and motion planning approach is expected to contribute to the development of new technology in the

area of conventional wheeled mobile robots with multiple centred steerable wheels for material handling applications.

IX.2 Summary of contributions

The literature reviewed revealed that no such system for moving wheeled loads with a wheeled mobile robot, ***grasping without lifting*** by applying stable pushes and pulls to the wheeled load, has been developed or investigated. This research work is novel in developing and investigating such kind of systems from many perspectives.

The work of Campion et al. (1996) presented a methodology for modelling the kinematics and dynamics of wheeled mobile robots in order to study the kinematic and dynamics properties of that kind of robots. This research work proposes a method for computing the kinematic constraints, the internal kinematics, and the external kinematics of the robot using fewer variables (see sections IV.3 and IV.4). Also, it introduces for the first time the holonomic parameterized centred angular coordinates α_p for coordinating the steering of multiple steerable wheels (see section IV.5). Besides, it proposes a method for computing the dynamics of the robot at the configuration space and at the steering space, adding to the dynamic equation of motions elements to compute the torques needed for the motorization of multiple steerable wheels (see section IV.6 and the end of section VI.3.5). In addition, the formulation of the remaining generalized velocities allows the computation of the velocity commands to the motorized joints (see section IV.5.2, V.6.1.a, and V.6.2.a).

So far, the dynamics equation of motion of mobile robots has been modelled by various authors such that it is not possible to locate the centre of mass of the robot in any place [e.g., Yamamoto & Yun (1993), Sarkar et al. (1994), Fierro & Lewis (1997), Laumond (1998), Tan et al. (2003)]. This research adds to the dynamic equation of motion the possibility of locating the centre of the mass of the robot in any place (see sections V.6.1.b and V.6.2.b). Also, the dynamic equation of motion at the steering space for robots type(2,0) has been computed (see section V.6.1.c).

Several authors have proposed a method for partially linearizing the posture dynamical state space representation of restricted mobility robots of type(2,0) using input-output static state feedback [e.g., the work of Yamamoto & Yun (1993), Sarkar et al. (1994), and Fierro & Lewis (1997)], also a method for partially linearizing the posture dynamical

state space representation of restricted mobility robots with steerable wheels using input-output static state feedback [e.g., d'Andréa-Novel et al. (1995)]. This research proposes a methodology that partially linearizes the posture dynamical state space representation of restricted mobility robots with or without steerable wheels using input-output static state feedback. Also, it presents the model description of the input-output map of the robots type(2.0) and type(1,1) (see sections V.6.1.d and V.6.2.d).

Some authors have used virtual wheeled elements for controlling the movements of the robot [e.g., Burke & Durrant-Whyte (1993), Lamiraux et al. (1999), Mutambara & Durrant-Whyte (2000), Wang & Qi (2001)]. This research work has introduced the concept of virtual wheel based on the kinematic properties of the robot and they are used to linearize and control the model of the robot (see sections V.6.2.d and section VI.3.5).

This research work has demonstrated that wheeled robot manipulators lose mobility or degrees of freedom when the robot applies stable pushes or pulls to a nonholonomic wheeled load (see section V.3.1).

This research work has demonstrated that the centred angular generalized velocities $\dot{\alpha}_p$ of restricted mobility robots with steerable wheels are holonomic, using nonholonomic theory to probe it (see section V.6.2.a).

Most of the literature, in motion planning for obstacle avoidance has demonstrated how to guide a robot to a desired goal robot without colliding with obstacles using virtual potential field methods [e.g., Khatib (1986), Krogh & Thorpe (1986), Khosla & Volpe (1988), Rimon & Koditschek (1992), Kyriakopoulos et al. (1996), Ulrich & Borenstein (2000)]. This research work introduced the concept of virtual goals, which do not need the definition of a potential function to be applied, to guide a robot to a desired goal without colliding with obstacles (see section VII.3).

The work of Kyriakopoulos et al. (1996) presented a navigation method for nonholonomic vehicles in complex environments with potential fields and tracking using the posture kinematics model of the robot. This research work introduces a navigation method for nonholonomic wheeled mobile robots with virtual goals and tracking applied to the partially linearized input-output map of the robot model (see sections VI.2 and VIII.3).

The work of Lamiraux et al. (2004) presented a reactive navigation method that given a collision-free path to be followed by the robot, in presence of obstacles, the method deforms the initial path in order to move it away from the obstacles, where the method computes iteratively admissible nonholonomic paths deformations for the robot based on a potential field function. This research introduces a novel reactive trajectory deformation method that uses virtual goals and tracking to deform the collision-free trajectory to avoid collision in the presence of obstacles (see section VIII.3.2).

IX.3 Suggestions for further work

In order to take into account the torques associated to the motors and transmission inertia, joint frictions, disturbances, unmodelled dynamics, and unmodelled external forces and moments that were not considered in the modelling of the robots. Here is suggested to design nonlinear motor controllers that include these terms to have a more realistic robot behaviour.

In order to couple the different behaviours of the models of single-robot and robot-trolley configurations in the motion planning approach. It is suggested to design a reconfigurable architecture with a switching coordinator of behaviours.

In this research work the analysis is restricted to having a robot that can grasp the cart using the front side of the robot. It is proposed to design a robot able to hold the load using the four sides of the robot.

This research work considered the analysis of one robot. It is proposed to design an architecture for multi-robot environments that takes into account the complete problem of given a series of tasks it produces the task scheduling program and computing the trajectories of each robot for that kind of system.

The robot must hold the trolley in at least two ponts. It is suggested to desing the mechanis (end efectors) to hold the trolley and the mechanisms to operate the manual brake of the trolley. Also, to built the robot and test it.

References

Agrachev, A. A. (2002). ***Mathematical Control Theory.*** ICTP Lecture Notes Series, Volume VIII: Parts 1 & 2. Trieste, Italy, ICTP - The Abdus Salam International Centre for Theoretical Physics. http://www.ictp.trieste.it/~pub_off/lectures/. ISBN: 92-95003-11-X.

AGV-Products (2004). ***Automated Guided Vehicles For The World's Best Companies.*** <http://www.agvp.com >. 10 January, 2004.

Alexander, J. C. and Maddocks, J. H. (1989). On kinematics of wheeled mobile robots. ***International Journal of Robotics Research.*** *Vol: 8(5): 15-27.* Oct, 1989. ISSN: 0278-3649.

Altafini, C. (2001). Some properties of the general n-trailer. ***International Journal of Control.*** *Vol: 74(4): 409-24.* 10 March 2001. ISSN: 00207179.

Althaus, P. and Christensen, H. I. (2003). Behavior coordination in structured environments. ***Advanced Robotics.*** *Vol: 17(7): 657-674.* ISSN: 0169-1864.

Arkin, R. C. (1988). ***Intelligent mobile robots in the workplace: leaving the guide behind.*** Proceedings of the first international conference on Industrial and engineering applications of artificial intelligence and expert systems. Tullahoma, Tennessee, United States. June 1988. ACM Press New York, NY, USA. Vol: 1. pp. 553 - 561. ISBN: 0-89791-271-3.

Arkin, R. C. and Balch, T. (1997). AuRA: principles and practice in review. ***Journal of Experimental and Theoretical Artificial Intelligence.*** *Vol: 9(2-3): 175-189.* April-Sept. 1997. ISSN: 0952-813X.

Arkin, R. C. (1998). ***Behavior-based robotics.*** Intelligent robots and autonomous agents. Cambridge, Mass., London : MIT Press. ISBN: 0262011654.

Astolfi, A., Bolzern, P. and Locatelli, A. (2004). Path-tracking of a tractor-trailer vehicle along rectilinear and circular paths: a Lyapunov-based approach. ***Robotics and Automation, IEEE Transactions on.*** *Vol: 20(1): 154-160.* ISSN: 1042-296X.

Bayle, B., Fourquet, J.-Y. and Renaud, M. (2003a). Manipulability of Wheeled Mobile Manipulators: Application to Motion Generation. ***International Journal of Robotics Research.*** *Vol: 22(7-8): 565-581.* July 1, 2003. ISSN: 0278-3649.

Bayle, B., Renaud, M. and Fourquet, J.-Y. (2003b). Nonholonomic Mobile Manipulators: Kinematics, Velocities and Redundancies. ***Journal of Intelligent and Robotic Systems.*** *Vol: 36(1): 45-63.* Jan. 2003. ISSN: 09210296.

Becerra, V. M. (2004). ***Advanced Nonlinear Control CY4A2, Lecture notes.*** Department of Cybernetics, The University of Reading. <http://www.personal.rdg.ac.uk/~shs99vmb/notes/ >. 27 February 2004.

Ben-Shahar, O. and Rivlin, E. (1995). Part I: To push or not push: On the Rearrangement of Movable Objects by a Mobile Robot. ***Technical Report: CIS9516***. Computer Science Department, Technion. Haifa, Israel.

Ben-Shahar, O. and Rivlin, E. (1998a). Practical pushing planning for rearrangement tasks. ***Robotics and Automation, IEEE Transactions on***. *Vol: 14(4): 549-565.* Aug. 1998. ISSN: 1042-296X.

Ben-Shahar, O. and Rivlin, E. (1998b). To push or not to push: on the rearrangement of movable objects by a mobile robot. ***Systems, Man and Cybernetics, Part B, IEEE Transactions on***. *Vol: 28(5): 667 - 679.* Oct. 1998. ISSN: 1083-4419.

Bentivegna, D. C., Ali, K. S., Arkin, R. C., et al. (1998). ***Design and Implementation of a Teleautonomous Hummer.*** Mobile Robot Laboratory College of Computing Georgia Institute of Technology Atlanta, Georgia. CiteSeer.IST - Penn State and NEC. <http://citeseer.ist.psu.edu/67418.html >. 10 January 2005.

Berman, S. and Edan, Y. (2002). Decentralized autonomous AGV system for material handling. ***International Journal of Production Research***. *Vol: 40(15): 3995-4006.* Oct. 2002. ISSN: 00207543.

Bloch, A. and Drakunov, S. (1994). ***Stabilization of a nonholonomic system via sliding modes***. Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on. Lake Buena Vista, FL. 14-16 Dec. 1994. Vol: 3. pp. 2961 - 2963. ISSN: 0191-2216.

Bloch, A. M., McClamroch, N. H. and Reyhanoglu, M. (1990). ***Controllability and stabilizability properties of a nonholonomic control system***. Decision and Control, 1990., Proceedings of the 29th IEEE Conference on. Honolulu, HI. 5-7 Dec. 1990. Vol: 3. pp. 1312 - 1314. ISSN: 0191-2216.

Bloch, A. M., Reyhanoglu, M. and McClamroch, N. H. (1992). Control and stabilization of nonholonomic dynamic systems. ***Automatic Control, IEEE Transactions on***. *Vol: 37(11): 1746-1757.* Nov 1992. ISSN: 0018-9286.

Bloss, D. (2002). Robotic mailman who does not ring twice, yet. ***Industrial Robot: An International Journal***. *Vol: 29(6): 507-510.* 2002. ISSN: 0143991X.

Bolzern, P., DeSantis, R. M. and Locatelli, A. (2001). An Input-Output Linearization Approach to the Control of an n-Body Articulated Vehicle. ***Journal of Dynamic Systems, Measurement, and Control***. *Vol: 123(3): 309-316.* September 2001.

Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots. ***Systems, Man and Cybernetics, IEEE Transactions on***. *Vol: 19(5): 1179-1187.* Sep/Oct 1989. ISSN: 0018-9472.

Borenstein, J. and Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. ***Robotics and Automation, IEEE Transactions on***. *Vol: 7(3): 278-288.* Jun 1991. ISSN: 1042-296X.

Borenstein, J., Wehe, D., Feng, L., et al. (1995). ***Mobile robot navigation in narrow aisles with ultrasonic sensors***. ANS 6th Tropical Meeting on Robotics and Remote Systems. Monterey, California. 5-10 February 1995. Vol. pp. 1-9.

Britannica-Corporate-Site (2004). ***Encyclopædia Britannica Online.*** Encyclopædia Britannica, Inc. <http://www.britannica.com >. 27 Julio, 2004.

Brockett, R. W. (1983). ***Asymptotic stability and feedback stabilization***. Differential geometric control theory, Proceedings of the conference. Michigan Technological University. June 28-July 2, 1982. Boston, Mass. : Birkhauser. Vol. pp. 181-191. ISBN / ISSN: 0364330910.

Brooks, R. A. (1985). A Robust Layered Control System for a Mobile Robot. ***MIT AI Lab Memo No.: 864.*** Massachusetts Institute of Technology. Massachusetts.

Brooks, R. A. (1990a). Elephants don't play chess. ***Robotics and Autonomous Systems.*** *Vol: 6(1): 3-15.* 1990. ISSN: 0921-8890.

Brooks, R. A. (1990b). The Behavior Language; User's Guide. ***MIT AI Lab Memo No.: 1227.*** Massachusetts Institute of Technology. Massachusetts.

Brooks, R. A. (1991). Intelligence without Reason. ***MIT A.I. Memo No.: 1293.*** Massachusetts Institute of Thechnology. Cambridge, Massachusetts.

Buhmann, J. M., Burgard, W., Cremers, A. B., et al. (1995). The Mobile Robot RHINO. ***AI Magazine***. Vol: **16:** 31-38. ISSN: 0738-4602.

Burke, T. and Durrant-Whyte, H. F. (1993). ***Kinematics for modular wheeled mobile robots***. IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems. Intelligent Robots for Flexibility. Yokohama, Japan. New York, NY, USA: IEEE. Vol: **2.** pp. 1279-86. ISBN: 0780308239.

Cameron, S. (1994). Obstacle avoidance and path planning. ***Industrial Robot: An International Journal.*** *Vol: 21(5): 9-14.* 1994. ISSN: 0143991X.

Campion, G., d'Andrea-Novel, B. and Bastin, G. (1991). ***Modelling and state feedback control of nonholonomic mechanical systems***. Decision and Control, Proceedings of the 30th IEEE Conference on. Brighton, UK. 11-13 Dec. 1991. Vol: **2.** pp. 1184 - 1189. ISBN: 0-7803-0450-0.

Campion, G., Bastin, G. and d'Andrea-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. ***Robotics and Automation, IEEE Transactions on.*** *Vol: 12(1): 47-62.* Feb. 1996. ISSN: 1042-296X.

Canudas de Wit, C. and Sordalen, O. J. (1992). ***Examples of piecewise smooth stabilization of driftless NL systems with less inputs than states***. Nonlinear Control Systems Design 1992. Selected Papers from the 2nd IFAC Symposium. Bordeaux, France. 24-26 June 1992. Oxford, UK: Pergamon Press. Vol. pp. 57-61. ISBN: 0-08-041901-1.

Canudas de Wit, C., Khennouf, H., Samson, C., et al. (1993). Nonlinear control design for mobile robots. In: Y. F. Zheng (Ed.) ***Recent trends in mobile robots***. Series in robotics and automated systems; v.11: 121-165. Singapore ; London. World Scientific. ISBN: 9810215118.

Canudas de Wit, C., Siciliano, B. and Bastin, G., eds. (1996). ***Theory of robot control.*** Communications and control engineering. London, Springer. ISBN: 3540760547.

Carlisle, B. (2000). ***Robot mechanisms***. Robotics and Automation. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: 1. pp. 701-708. ISBN: 0-7803-5886-4.

Chatila, R. (1995). Deliberation and reactivity in autonomous mobile robots. ***Robotics and Autonomous Systems***. *Vol: 16(2-4): 197-211*. Dec., 1995. ISSN: 0921-8890.

Chen, M. W. and Zalzala, A. M. S. (1997). Dynamic modelling and genetic-based trajectory generation for non-holonomic mobile manipulators. ***Control Engineering Practice***. *Vol: 5(1): 39-48*. Jan., 1997. ISSN: 09670661.

Chengqing, L., Ang, M. H., Jr., Krishnan, H., et al. (2000). ***Virtual obstacle concept for local-minimum-recovery in potential-field based navigation***. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: 2. pp. 983 - 988. ISBN: 0-7803-5886-4.

Choi, C. and Lee, J.-J. (1996). Dynamical path-planning algorithm of a mobile robot: Local minima problem and nonstationary environments. ***Mechatronics***. *Vol: 6(1): 81-100*. 1996/2. ISSN: 09574158.

Chung, J., Ryu, B.-S. and Yang, H. S. (1998). Integrated control architecture based on behavior and plan for mobile robot navigation. ***Robotica***. *Vol: 16: 378 - 399*. July-Aug., 1998. ISSN: 02635747.

Coron, J.-M. (1992). Global asymptotic stabilization for controllable systems without drift. ***Mathematics of control, signals, and systems MCSS***. *Vol: 5(3): 295-312*. 1992. ISSN: 09324194.

Craig, J. J. (1989). ***Introduction to robotics : mechanics and control.*** Addison-Wesley series in electrical and computer engineering: control engineering. Reading, Mass., Addison-Wesley. ISBN: 0201095289.

Crown (2002-2005). ***Crown Equipment Corporation.*** <http://www.crown.com/ >. 27 October 2005.

Cyberbotics-Ltd (2003). ***Webots 4.*** <http://www.cyberbotics.com >. 15 July, 2004.

d'Andréa-Novel, B., Campion, G. and Bastin, G. (1995). Control of nonholonomic wheeled mobile robots by state feedback linearization. ***International Journal of Robotics Research***. *Vol: 14(6): 543-559*. December 1995. ISSN: 0278-3649.

Dane Industries (2003). ***Power assist solutions.*** <http://www.daneindustries.com >. 16 February 2005.

De Luca, A. and Oriolo, G. (1995). Modelling and Control of Nonholonomic Mechanical Systems. In: J. Angeles and A. Kecskemethy (Ed.) ***Kinematics and Dynamics of Multi-Body Systems,***. CISM Courses and Lectures. Vol: **360:** 277-342. Wien. New York : Springer-Verlag. ISBN/ISSN: 3211827315.

Dong, W., Xu, Y. and Wang, Q. (2000). ***On tracking control of mobile manipulators.*** Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: **4.** pp. 3455 - 3460. ISBN: 0-7803-5886-4.

Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. ***American Journal of Mathematics.*** *Vol: 79: 497-516.* 1957.

Envosort (2004). ***York Roll Container.*** <http://www.envosort.com >. 12 January 2005.

Esquivel, W. D. and Chiang, L. E. (2002). Nonholonomic path planning among obstacles subject to curvature restrictions. ***Robotica.*** *Vol: 20: 49-58.* Jan 2002. ISSN: 02635747.

Fantoni, I. and Lozano, R. (2001). ***Non-linear control for underactuated mechanical systems.*** Communications and control engineering. London, Springer. ISBN: 1852334231.

Felner, A., Stern, R. and Kraus, S. (2002). ***PHA*: Performing A* in unknown physical environments.*** Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems. Bologna, Italy. 15-19 July 2002. New York, NY, USA: ACM. Vol. pp. 240-247.

Fierro, R. and Lewis, F. L. (1997). Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. ***Journal of Robotic Systems.*** *Vol: 14(3): 149-163.* 1997. ISSN: 07412223.

Fliess, M., Levine, J., Martin, P., et al. (1995). Flatness and defect of nonlinear systems: introductory theory and examples. ***International Journal of Control.*** *Vol: 61(6): 1327-61.* June 1995. ISSN: 00207179.

Fraichard, T. and Scheuer, A. (2004). From Reeds and Shepp's to continuous-curvature paths. ***Robotics, IEEE Transactions on.*** *Vol: 20(6): 1025- 1035.* Dec. 2004. ISSN: 1552-3098.

Fujita, M., Kitano, H. and Kageyama, K. (1998). ***Reconfigurable physical agents.*** Proceedings of the Second International Conference on Autonomous Agents. Minneapolis, MN, USA. ACM: New York, NY, USA. Vol. pp. 54-61. ISBN / ISSN: 0-89791-983-1.

Fujita, M., Kitano, H. and Kageyama, K. (1999). A reconfigurable robot platform. ***Robotics and Autonomous Systems.*** *Vol: 29(2-3): 119-132.* 30 November 1999. ISSN: 0921-8890.

Gat, E. (1998). On three-layer architectures. In: D. Kortenkamp, R. P. Bonasso and R. Murphy (Ed.) ***Artificial intelligence and mobile robots: case studies of***

successful robot systems. Cambridge, Mass. Menlo Park, Calif. ; London : AAAI Press : MIT Press. ISBN: 0262611376.

Goldberg, K., Lee, K. M., Akella, S., et al. (2002). ***IEEE Robotics and Automation Society Intent to establish the: Transactions on Automation Sciences and Engineering (TASE).*** IEEE RAS TASE Committee. <http://www.ncsu.edu/IEEE-RAS/TRA/TASEproposal.html >. 14 March 2005.

Gomez Ortega, J. and Camacho, E. F. (1996). Mobile robot navigation in a partially structured static environment, using neural predictive control. ***Control Engineering Practice. Vol: 4(12): 1669-1679.*** 1996/12. ISSN: 0967-0661.

Gonzalez-Villela, V. J. (2002). Identifying opportunities of research and development projects at Royal Mail: Report of the visit to the Royal Mail plc. ***Report.*** Mechatronics Research Centre, Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University. Loughborough.

Gonzalez-Villela, V. J., Parkin, R. M. and Lopez-Parra, M. (2003a). ***A model for mechatronics design with embedded micro-controllers***. ICOM 2003: International Conference on Mechatronics. Loughborough, UK. 19-20 June 2003. Professional Engineering Publishing Limited. Vol. pp. 91-96. ISBN: 1860584209.

Gonzalez-Villela, V. J. and Dorador-Gonzalez, J. M. (2003b). ***The mechatronics engineering career at UNAM.*** ICOM 2003: International Conference on Mechatronics. Loughborough, UK. 19-20 June 2003. Professional Engineering Publishing. Vol. pp. 649 - 654. ISBN: 1860584209.

Gonzalez-Villela, V. J., Parkin, R. M., Lopez-Parra, M., et al. (2004). A wheeled mobile robot with obstacle avoidance capability. ***Ingeniería Mecánica Tecnología y Desarrollo. Revista de la Sociedad Mexicana de Ingeniería Mecánica (SOMIM). Vol: 1(5): 159-166.*** Septiembre 2004. ISSN: 1665-7381.

Gonzalez-Villela, V. J. and Parkin, R. M. (2005). Evadiendo obstáculos con robots móviles. ***Revista Digital Universitaria [en línea].*** Vol: **6**. ISSN:1607-6079, Available on the Internet: http://www.revista.unam.mx.

Gonzalez-Villela, V. J., Parkin, R. M., Lopez-Parra, M., et al. (2005). ***Dynamic torque control extension for a wheeled mobile robot with obstacle avoidance capability.*** Memorias del XI Congreso Internacional Anual SOMIM & IV Congreso Bolivariano de Ingeniería Mecánica. Morelia, Mich. México. 21-23 de Septiembre. Vol: **OT**. pp. 38-45.

Groover, M. P. (1987). ***Automation, production systems, and computer-integrated manufacturing.*** Englewood Cliffs, NJ : Prentice-Hall, c1987. ISBN: 0130546526.

Halperin, D., Kavraki, L. E. and Latombe, J. C. (2004). Robotics. In: J. E. Goodman and J. O'Rourke (Ed.) ***Handbook of Discrete and Computational Geometry***: Chapter 47, pp. 755-778. Boca Raton, FL. CRC Press. ISBN: 0849385245.

Henson, M. A. and Seborg, D. E. (1997). ***Nonlinear process control.*** London : Prentice-Hall International, Upper Saddle River, N.J. : Prentice Hall PTR. ISBN: 013625179x.

Hoff, E. B. and Sarker, B. R. (1998). An overview of path design and dispatching methods for automated guided vehicles. ***Integrated Manufacturing Systems, , Emerald Group Publishing Limited.*** *Vol: 9(5): 296 - 307.* 1998. ISSN: 0957-6061.

Holmberg, R. and Khatib, O. (2000). Development and control of a holonomic mobile robot for mobile manipulation tasks. ***International Journal of Robotics Research.*** *Vol: 19(11): 1066 - 1074.* Nov. 2000. ISSN: 0278-3649.

Hoozemans, M. J. M., van der Beek, A. J., Fringsdresen, M. H. W., et al. (1998). Pushing and pulling in relation to musculoskeletal disorders: a review of risk factors. ***Ergonomics.*** *Vol: 41(6): 757 - 781.* June 1, 1998. ISSN: 0014-0139.

Huntsberger, T. (2001). Biologically Inspired Autonomous Rover Control. ***Autonomous Robots.*** *Vol: 11(3): 341-346.* 2001. ISSN: 0929-5593.

Huo, W. and Ge, S. S. (2001). Exponential stabilization of non-holonomic systems: an ENI approach. ***International Journal of Control.*** *Vol: 74(15): 1492-1500.* 15 Oct. 2001. ISSN: 0020-7179.

Hwang, Y. K. and Ahuja, N. (1992). Gross motion planning — A survey. ***ACM Comput. Surv., ACM Press.*** *Vol: 24(3): 219-291.* ISSN: 0360-0300.

Iannitti, S. (2002). Motion Planning and Control of a Class of Underactuated Robots. ***Ingegneria dei Sistemi.*** Roma, Universita Degli Studi Di Roma "La Sapienza": 143.

Isidori, A. (1995). ***Nonlinear control systems.*** Communications and control engineering series. Berlin, Springer Verlag. ISBN: 1852331887.

Jacoff, A., Messina, E. and Evans, J. (2002). Performance evaluation of autonomous mobile robots. ***Industrial Robot: An International Journal.*** *Vol: 29(3): 259-267.* 2002. ISSN: 0143991X.

Jantapremjit, P. and Austin, D. (2001). ***Design of a Modular Self-Reconfigurable Robot.*** Proceedings of Australian Conference on Robotics and Automation (ACRA 2001), http://wwwsyseng.rsise.anu.edu.au/rsl/rsl_papers.html. Sydney, Australia. November, 2001. Vol. pp. 6.

Ji, M., Zhang, Z., Biswas, G., et al. (2003). Hybrid fault adaptive control of a wheeled mobile robot. ***Mechatronics, IEEE/ASME Transactions on.*** *Vol: 8(2): 226-233.* ISBN: 1083-4435.

Jiang, Z.-P. and Nijmeijer, H. (1999). A recursive technique for tracking control of nonholonomic systems in chained form. ***Automatic Control, IEEE Transactions on.*** *Vol: 44(2): 265-279.* 0018-9286.

Kambhampati, S. and Davis, L. (1986). Multiresolution path planning for mobile robots. ***Robotics and Automation, IEEE Journal of.*** *Vol: 2(3): 135- 145.* Sep 1986. ISSN: 0882-4967.

Kanayama, Y., Kimura, Y., Miyazaki, F., et al. (1990). ***A stable tracking control method for an autonomous mobile robot.*** Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on. 13-18 May 1990. Vol: **1**. pp. 384-389. ISBN: 0-8186-9061-5.

Karkoub, M. A. and Zribi, M. (2002). Modelling and non-linear discontinuous feedback control of crane lifter systems. ***Proceedings of the Institution of Mechanical Engineers, Part I (Journal of Systems and Control Engineering). Vol: 216(12): 157-167.*** 2002. ISSN: 09596518.

Kelly, A. (1996). ***Introduction to Mobile Robots.*** The Robotics Institute, School of Computer Science, Carnegie Mellon University. <http://www.frc.ri.cmu.edu/~alonzo/course/course.html >. 27 July, 2004.

Keppens, J. and Shen, Q. (2001). On compositional modelling. ***The Knowledge Engineering Review. Vol: 16(2): 157-200.*** March 2001. ISSN: 02698889.

Khalil, H. K. (2002). ***Nonlinear systems.*** Upper Saddle River, N.J., Prentice Hall. ISBN: 0130673897.

Khalil, W. and Dombre, E. (2001). ***Modeling, identification & control of robots.*** London, Penton. ISBN: 1903996139.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. ***International Journal of Robotics Research. Vol: 5(1): 90-98.*** Spring 1986. ISSN: 0278-3649.

Khatib, O. (1999). Mobile manipulation: The robotic assistant. ***Robotics and Autonomous Systems. Vol: 26(2-3): 175-183.*** 1999/2/28. ISSN: 0921-8890.

Khosla, P. and Volpe, R. (1988). ***Superquadric artificial potentials for obstacle avoidance and approach.*** Robotics and Automation, Proceedings of the IEEE International Conference on. Philadelphia, PA. 24-29 April 1988. Vol: **3**. pp. 1380 - 1385. ISBN: 0-8186-0852-8.

Kim, J.-O. and Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. ***Robotics and Automation, IEEE Transactions on. Vol: 8(3): 338-349.*** Jun 1992. ISSN: 1042-296X.

Kingma, I., Kuijer, P. P. F. M., Hoozemans, M. J. M., et al. (2003). Effect of design of two-wheeled containers on mechanical loading. ***International Journal of Industrial Ergonomics. Vol: 31(2): 73-86.*** February, 2003. ISSN: 0169-8141.

Koff, G. A. and Demag, R. (1985). ***Automatic Guided Vehicles Systems "Basics of AGVS".*** Matherial Handling Institute. <http://www.mhia.org >. 12 January 2005.

Kolmanovsky, I. and McClamroch, N. H. (1995). Developments in nonholonomic control problems. ***IEEE Control Systems Magazine. Vol: 15(6): 20-36.*** Dec. 1995. ISSN: 02721708.

Koren, Y. and Borenstein, J. (1991). ***Potential field methods and their inherent limitations for mobile robot navigation.*** Robotics and Automation, 1991.

Proceedings., 1991 IEEE International Conference on. Sacramento, CA. 9-11 April 1991. Vol: 2. pp. 1398 - 1404. ISBN: 0-8186-2163-X.

Kortenkamp, D. (1998). ***Artificial intelligence and mobile robots : case studies of successful robot systems.*** Cambridge, Mass., Menlo Park, Calif. ; London : AAAI Press : MIT Press. ISBN: 0262611376.

Krippendorff, K. (1986). ***Klaus Krippendorff's Dictionary of Cybernetics.*** Principia Cybernetica. <http://pespmc1.vub.ac.be/ASC/Kripp.html >. 14 January 2005.

Krogh, B. H. (1984). ***A generalized potential field approach to obstacle avoidance control.*** Robotics Research: The Next Five Years and Beyond, SME Conference Proceedings on. Bethlehem, Pennsylvania. Vol. ISBN: 0-87263-152-4.

Krogh, B. H. and Thorpe, C. E. (1986). ***Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles.*** Proc. IEEE Int. Conf. Robotics and Automation. Vol. pp. 1664-1669. ISBN: 0-8186-0695-9.

K-Team-Corporation (2002-2004). ***Robot Khepera II.*** <http://www.k-team.com/robots/khepera/index.html >. 15 July, 2004.

Kuijer, P. P. F. M., van der Beek, A. J., van Dieen, J. H., et al. (2002). Effect of the number of two-wheeled containers at a gathering point on the energetic workload and work efficiency in refuse collecting. ***Applied Ergonomics.*** *Vol: 33(6): 571-577.* November, 2002. ISSN: 0003-6870.

Kyriakopoulos, K. J., Kakambouras, P. and Krikelis, N. J. (1996). ***Navigation of nonholonomic vehicles in complex environments with potential fields and tracking.*** Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on. Minneapolis, MN. 22-28 April 1996. Vol: 4. pp. 3389 - 3394. ISSN: 1050-4729.

Lamiraux, F., Sekhavat, S. and Laumond, J.-P. (1999). Motion planning and control for Hilare pulling a trailer. ***Robotics and Automation, IEEE Transactions on.*** *Vol: 15(4): 640 - 652.* Aug. 1999. ISSN: 1042-296X.

Lamiraux, F. and Lammond, J.-P. (2001). Smooth motion planning for car-like vehicles. ***Robotics and Automation, IEEE Transactions on.*** *Vol: 17(4): 498-501.* Aug 2001. ISSN: 1042-296X.

Lamiraux, F., Bonnafous, D. and Lefebvre, O. (2004). Reactive path deformation for nonholonomic mobile robots. ***Robotics, IEEE Transactions on.*** *Vol: 20(6): 967-977.* Dec. 2004. ISSN: 1552-3098.

Latombe, J. C. (1991). ***Robot motion planning.*** Boston, Kluwer Academic Publishers. ISBN: 0792391292.

Latombe, J. C. (1999). Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts. ***International Journal of Robotics Research.*** *Vol: 18(11): 1119-1128.* November 1999. ISSN: 0278-3649.

Latombe, J. C. (2003). ***Motion Planning. Course: CS326A, 2003 Spring Quarter.*** Stamford AI Lab, Stamford University, California. <http://robotics.stanford.edu/~latombe/ >. 24 January 2005.

Laumond, J. P. (1998). ***Robot motion planning and control.*** Lecture notes in control and information sciences ; 229. London, Springer. ISBN: 0780304047.

Laumond, J.-P. and Risler, J.-J. (1996). Nonholonomic systems: Controllability and complexity. ***Theoretical Computer Science. Vol: 157(1): 101-114.*** 1996/4/9. ISSN: 03043975.

Laursen, B. and Schibye, B. (2002). The effect of different surfaces on biomechanical loading of shoulder and lumbar spine during pushing and pulling of two-wheeled containers. ***Applied Ergonomics. Vol: 33(2): 167-174.*** March, 2002. ISSN: 0003-6870.

Lee, W. H. and Sanderson, A. C. (2001). Dynamic analysis and distributed control of the Tetrobot modular reconfigurable robotic system. ***Autonomous Robots. Vol: 10(1): 67-82.*** Jan. 2001. ISSN: 0929-5593.

Li, Z. and Canny, J. F. (1993). ***Nonholonomic motion planning.*** The Kluwer international series in engineering and computer science. Robotics: vision, manipulation and sensors. Boston, Mass, Kluwer Academic. ISBN: 0792392752.

Lima, P. and Ribeiro, M. I. (2002). ***Mobile Robotics. Course Handouts.*** Instituto Superior Técnico/Instituto de Sistemas e Robótica. <http://omni.isr.ist.utl.pt/~mir/cadeiras/robmovel/rmov0102.htm >. 14 Julio, 2004.

Lizárraga, D. A., Morin, P. and Samson, C. (2001). Chained form approximation of a driftless system. Application to the exponential stabilization of the general N-trailer system. ***International Journal of Control. Vol: 74(16): 1612 - 1629.*** November 10, 2001. ISSN: 0020-7179.

Looze, M. P. D., Greuningen, K. V., Rebel, J., et al. (2000). Force direction and physical load in dynamic pushing and pulling. ***Ergonomics. Vol: 43(3): 377-390.*** March 1, 2000. ISSN: 0014-0139.

Low, K. H., Low, W. K. and Ang, M. H. J. (2002). ***A hybrid mobile robot architecture with integrated planning and control.*** AAMAS '02: First International Joint Conference on Automomous Agents and Multi-Agent Systems. Bologna, Italy. July 15-19. New York, NY, USA: ACM. Vol. pp. 219-226.

Lozano-Perez, T. (1983). Spatial planning: a configuration space approach. ***IEEE Trans. Computers. Vol: C-32(2): 108-120.*** Feb., 1983. ISSN: 00189340.

Lynch, K. M. and Mason, M. T. (1996). Stable Pushing: Mechanics, Controlability, and Planning. ***International Journal of Robotics Research. Vol: 15(6): 533-556.*** December 1996. ISSN: 0278-3649.

Maaref, H. and Barret, C. (2002). Sensor-based navigation of a mobile robot in an indoor environment. ***Robotics and Autonomous Systems. Vol: 38(1): 1-18.*** 2002/1/31. ISSN: 0921-8890.

Mantel, R. J. and Landeweerd, H. R. A. (1995). Design and operational control of an AGV system. ***International Journal of Production Economics***. *Vol: 41(1-3): 257-266.* 1995/10. ISSN: 09255273.

Maravall, D., de Lope, J. and Serradilla, F. (2000). ***Combination of model-based and reactive methods in autonomous navigation***. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: **3**. pp. 2328 - 2333. ISBN: 0-7803-5886-4.

Marino, R. (1986). On the largest feedback linearizable subsystem. ***Systems & Control Letters***. *Vol: 6(5): 345-351.* 1986/1.

Mataric, M. J., Nilsson, M. and Simsarin, K. T. (1995). ***Cooperative multi-robot box-pushing***. Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on. Pittsburgh, PA. 5-9 Aug. 1995. IEEE Comput. Soc. Press. Vol: **3**. pp. 556-561. ISBN: 0-8186-7108-4.

McKerrow, P. (1991). ***Introduction to robotics.*** Electronic systems engineering series. New York, Wokingham : Addison-Wesley.

M'Closkey, R. (1998). Time-varying homogeneous feedback: design tools for the exponential stabilization of systems with drift. ***International Journal of Control.*** *Vol: 71(5): 837 - 869.* 20 Nov. 1998. ISSN: 0020-7179.

MHIA (2005). ***Material Handling Industry of America (MHIA).*** <http://www.mhia.org >. 12 January 2005.

Muir, P. F. and Neuman, C. P. (1987). Kinematic modelling of wheeled mobile robots. ***Journal of Robotic Systems***. *Vol: 4(2): 281-340.* Apr., 1987. ISSN: 07412223.

Murata, S., Yoshida, E., Kurokawa, H., et al. (2001). Self-Repairing Mechanical Systems. ***Autonomous Robots***. *Vol: 10(1): 7-21.* January 2001. ISSN: 0929-5593.

Murray, R. M. and Sastry, S. S. (1993). Nonholonomic motion planning: steering using sinusoids. ***Automatic Control, IEEE Transactions on***. *Vol: 38(5): 700 - 716.* May 1993. ISSN: 0018-9286.

Murray, R. M., Li, Z. and Sastry, S. S. (1994). ***A mathematical introduction to robotic manipulation.*** Boca Raton, Fla., London : CRC Press. ISBN: 0849379814.

Mutambara, A. G. O. and Durrant-Whyte, H. F. (2000). Fully decentralized estimation and control for a modular wheeled mobile robot. ***International Journal of Robotics Research***. *Vol: 19(6): 582-596.* June 2000. ISSN: 0278-3649.

Nehmzow, U. (2001). ***Mobile Robotics: Research, Applications and Challenges***. Proc. "Future Trends in Robotics". London. July 2, 2001. Institution of Mechanical Engineers, London. Vol.

Neimark, J. I. and Fufaev, N. A. (1972). ***Dynamics of nonholonomic systems.*** Translations of mathematical monographs ; v. 33. Providence, Rhode Island, American Mathematical Society. ISBN: 0821815830.

Nicolescu, M. N. and Mataric, M. J. (2002). ***A hierarchical architecture for behavior-based robots***. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1. Bologna, Italy. July 2002. ACM Press New York, NY, USA. Vol. pp. 227 - 233.

Nijmeijer, H. and van der Schaft, A. (1990). ***Nonlinear dynamical control systems.*** New York, Springer-Verlag. ISBN: 038797234x.

Nilsson, K. and Johansson, R. (1999). Integrated architecture for industrial robot programming and control. ***Robotics and Autonomous Systems***. *Vol: 29(4): 205-226.* 31 December 1999. ISSN: 0921-8890.

Orebäck, A. and Christensen, H. I. (2003). Evaluation of architectures for mobile robotics. ***Autonomous Robots***. *Vol: 14(1): 33-49.* Jan. 2003. ISSN: 09295593.

Oriolo, G., De Luca, A. and Vendittelli, M. (2002a). WMR control via dynamic feedback linearization: design, implementation, and experimental validation. ***Control Systems Technology, IEEE Transactions on***. *Vol: 10(6): 835 - 852.* Nov. 2002. ISSN: 10636536.

Papdopoulos, E., Poulakakis, I. and Papdimitriou, I. (2002). On Path Planning and Obstacle Avoidance for Nonholonomic Platforms with Manipulators: A Polynomial Approach. ***International Journal of Robotics Research***. *Vol: 21: 367-383.* April 1, 2002. ISSN: 0278-3649.

Pettersson, L. (1997). ***Control System Architectures for Autonomous Agents: A Survey Study.*** Mechatronics Division, Department of Machine Design, Royal Institute of Technology, Stockholm, Sweden. <CiteSeer.IST: Penn State and NEC: http://citeseer.ist.psu.edu/53581.html >. 10 Janury 2005.

Pimentel, B. S., Pereira, G. A. S. and Campos, M. M. F. M. (2002). ***On the development of cooperative behavior-based mobile manipulators***. Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems. Bologna, Italy. 15-19 July 2002. ACM: New York, NY, USA. Vol. pp. 234-239.

Pomet, J.-B. (1992). Explicit design of time-varying stabilizing control laws for a class of controllable system without drift. In:(Ed.) ***System and Control Letters***. Vol: **18:** 147-158. ISSN: 01676911.

Quinlan, S. (1994). ***Efficient distance computation between non-convex objects***. Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on. San Diego, CA. 8-13 May. Los Alamitos, CA, USA: IEEE Comput. Soc. Press. Vol: **4**. pp. 3324 - 3329. ISBN: 0-8186-5330-2.

Reeds, J. A. and Shepp, L. A. (1990). Optimal paths for a car that goes both forwards and backwards. ***Pacific Journal of Mathematics***. *Vol: 145(2): 367-393.*

Reyhanoglu, M., van der Schaft, A., Mcclamroch, N. H., et al. (1999). Dynamics and control of a class of underactuated mechanical systems. ***Automatic Control, IEEE Transactions on***. *Vol: 44(9): 1663-1671.* 0018-9286.

Rimon, E. and Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions. ***Robotics and Automation, IEEE Transactions on***. *Vol: 8(5): 501 - 518.* Oct. 1992. ISSN: 1042-296X.

Rollins, E., Luntz, J., Foessel, A., et al. (1998). ***Nomad: a demonstration of the transforming chassis***. Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. Leuven, Belgium. 16-20 May 1998. Vol: **1**. pp. 611-617. ISBN: 0-7803-4300-X.

Rooks, B. (2001). AGVs find their way to greater flexibility. ***Assembly Automation***. *Vol: 21(1): 38-43.* 2001. ISSN: 01445154.

Rooks, B. (2002). Mobile robots walk into the future. ***Industrial Robot: An International Journal***. *Vol: 29(6): 517-523.* 2002. ISSN: 0143-991X.

Rosell, J. and Iniguez, P. (2002). ***A hierarchical and dynamic method to compute harmonic functions for constrained motion planning***. Intelligent Robots and System, 2002. IEEE/RSJ International Conference on. Barcelona, Spain. Vol: **3**. pp. 2335- 2340. ISBN: 0-7803-7398-7.

Rus, D., Donald, B. and Jennings, J. (1995). ***Moving furniture with teams of autonomous robots***. Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings of thre IEEE/RSJ International Conference on. Pittsburgh, PA. Aug. 1995. Vol: **1**. pp. 235-242. ISBN: 0-8186-7108-4.

Rus, D., Butler, Z., Kotay, K., et al. (2002). Self-reconfiguring robots. ***Commun. ACM***. *Vol: 45(3): 39-45.* 2002. ISSN: 00010782.

Russell, S. J. and Norvig, P. (1995). ***Artificial intelligence : a modern approach.*** Upper Saddle River, N.J., Prentice Hall. ISBN: 0131038052.

Salichs, M. A. and Moreno, L. (2000). Navigation of mobile robots: open questions. ***Robotica***. *Vol: 18(3): 227-234.* May 2000. ISSN: 0263-5747.

Samson, C. (1993). Time-varying feedback stabilization of car-like wheeled mobile robots. ***International Journal of Robotics Research***. *Vol: 12(1): 55-64.* Feb. 1993. ISSN: 0278-3649.

Sarkar, N., Yun, X. and Kumar, V. (1994). Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots. ***International Journal of Robotics Research***. *Vol: 13(1): 55-69.* 1994. ISSN: 0278-3649.

Shim, H.-S. and Sung, Y.-G. (2003). Asymptotic control for wheeled mobile robots with driftless constraints. ***Robotics and Autonomous Systems***. *Vol: 43(1): 29-37.* 2003/4/30. ISSN: 0921-8890.

Simmons, R., Goodwin, R., Haigh, K. Z., et al. (1997). ***A layered architecture for office delivery robots***. Proceedings of the first international conference on Autonomous agents. Marina del Rey, California, United States. 1997. ACM Press New York, NY, USA. Vol. pp. 245 - 252. ISBN: 0-89791-877-0.

Simmons, R., Fernandez, J. L., Goodwin, R., et al. (2000). Lessons learned from Xavier. ***Robotics & Automation Magazine, IEEE. Vol: 7(2): 33 - 39.*** June 2000. ISSN: 1070-9932.

Stoy, K., Shen, W. M. and Will, P. (2002). ***How to make a self-reconfigurable robot run.*** Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2. Bologna, Italy. Vol. pp. 813-820.

Straker, L. M., Stevenson, M. G. and Twomey, T. (1997). A comparison of risk assessment of single and combination manual handling tasks: 2. Discomfort, Rating of Perceived Exertion and heart rate measures. ***Ergonomics. Vol: 40(6): 656-669.*** June 1, 1997. ISSN: 0014-0139.

Sugar, T. G. and Kumar, V. (2002). Control of cooperating mobile manipulators. ***Robotics and Automation, IEEE Transactions on. Vol: 18(1): 94-103.*** ISSN: 1042-296X.

Sussmann, H. J. (1983). Lie brackets, real analyticity and geometric control. In: R. S. M. Roger W. Brockett, Hector J. Sussman, (eds.) (Ed.) ***Differential geometric control theory.*** Progress in mathematics (Birkhäuser); 27: 1-116. Boston, MA: Birkhauser. ISBN/ISSN: 3764330910.

Sussmann, H. J. (1987). A general theorem on local controllability. ***SIAM Journal on Control and Optimization. Vol: 25(1): 158-194.*** Jan. 1987. ISSN: 03630129.

Sussmann, H. J. (1991). ***Local controllability and motion planning for some classes of systems with drift.*** Decision and Control, 1991., Proceedings of the 30th IEEE Conference on. Brighton. 11-13 Dec. 1991. Vol: **2.** pp. 1110 - 1114. ISBN: 0-7803-0450-0.

Tan, J., Xi, N. and Wang, Y. (2003). Integrated Task Planning and Control for Mobile Manipulators. ***International Journal of Robotics Research. Vol: 22(5): 337-354.*** May 2003. ISSN: 0278-3649.

Tanner, H. G. and Kyriakopoulos, K. J. (2000). ***Nonholonomic motion planning for mobile manipulators.*** Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: **2.** pp. 1233 - 1238. ISBN: 0-7803-5886-4.

Tanner, H. G., Loizou, S. G. and Kyriakopoulos, K. J. (2003). Nonholonomic navigation and control of cooperating mobile manipulators. ***Robotics and Automation, IEEE Transactions on. Vol: 19(1): 53- 64.*** Feb 2003. ISSN: 1042-296X.

Thuilot, B., d'Aandrea-Novel, B. and Micaelli, A. (1996). Modeling and feedback control of mobile robots equipped with several steering wheels. ***Robotics and Automation, IEEE Transactions on. Vol: 12(3): 375 - 390.*** June 1996. ISSN: 1042-296X.

Ulrich, I. and Borenstein, J. (1998). ***VFH+: reliable obstacle avoidance for fast mobile robots.*** Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. Leuven. 16-20 May. Vol: **2.** pp. 1572 - 1577. ISBN: 0-7803-4300-X.

Ulrich, I. and Borenstein, J. (2000). ***VFH*: local obstacle avoidance with look-ahead verification***. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA , USA. 04/24/2000 -04/28/2000. IEEE. Vol: **3**. pp. 2505-2511. ISBN: 0-7803-5886-4.

Unsal, C. and Khosla, P. K. (2000). ***Mechatronic design of a modular self-reconfiguring robotic system***. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: **2**. pp. 1742 - 1747. ISBN: 0-7803-5886-4.

Vadakkepat, P., Tan, K. C. and Ming-Liang, W. (2000). ***Evolutionary artificial potential fields and their application in real time robot path planning***. Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. La Jolla, CA. 16-19 July 2000. Vol: **1**. pp. 256 - 263. ISBN: 0-7803-6375-2.

Valavanis, K. P., Hebert, T., Kolluru, R., et al. (2000). Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field. ***Systems, Man and Cybernetics, Part A, IEEE Transactions on***. *Vol: 30(2): 187-196*. Mar 2000. ISSN: 1083-4427.

van der Beek, A. J., Hoozemans, M. J. M., Frings-Dresen, M. H. W., et al. (1999). Assessment of exposure to pushing and pulling in epidemiological field studies: an overview of methods, exposure measures, and measurement strategies. ***International Journal of Industrial Ergonomics***. *Vol: 24(4): 417-429*. August 23, 1999. ISSN: 0169-8141.

van der Beek, A. J., Kluver, B. D. R., Frings-Dresen, M. H. W., et al. (2000). Gender differences in exerted forces and physiological load during pushing and pulling of wheeled cages by postal workers. ***Ergonomics***. *Vol: 43(2): 269-281*. February 1, 2000. ISSN: 0014-0139.

Victorino, A. C., Rives, P. and Borrelly, J.-J. (2003a). Safe navigation for indoor mobile robots. Part I: a sensor-based navigation framework. ***International Journal of Robotics Research***. *Vol: 22(12): 1005-1018*. Dec. 2003. ISSN: 0278-3649.

Victorino, A. C., Rives, P. and Borrelly, J.-J. (2003b). Safe navigation for indoor mobile robots. Part II: exploration, self-localization and map building. ***International Journal of Robotics Research***. *Vol: 22(12): 1019-1039*. Dec. 2003. ISSN: 02783649.

Virk, G. S. (2003). CLAWAR Modularity for Robotic Systems. ***International Journal of Robotics Research***. *Vol: 22(3/4): 265 - 277*. March - April 2003. ISSN: 0278-3649.

Volpe, R., Nesnas, I., Estlin, T., et al. (2001). ***The CLARAty architecture for robotic autonomy***. Aerospace Conference, 2001, IEEE Proceedings on. Big Sky, MT. 10-17 March 2001. IEEE. Vol: **1**. pp. 1/121 - 1/132. ISBN: 0-7803-6599-2.

Wang, D. and Qi, F. (2001). ***Trajectory planning for a four-wheel-steering vehicle***. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. Seul, Korea. 24-26 May, 2001. Vol: **4**. pp. 3320-3325. ISBN: 0-7803-6576-3.

Wang, Y. and Cartmell, M. P. (1998). Trajectory generation for a four wheel steering tractor-trailer system: a two-step method. ***Robotica***. *Vol: 16(4): 381-386.* July 1998. ISSN: 02635747.

Wang, Y. and Chirikjian, G. S. (2000). ***A new potential field method for robot path planning***. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: 2. pp. 977 - 982. ISBN: 0-7803-5886-4.

Xu, F., Van Brussel, H., Nuttin, M., et al. (2003). Concepts for dynamic obstacle avoidance and their extended application in underground navigation. ***Robotics and Autonomous Systems***. *Vol: 42(1): 1-15.* Jan 31, 2003. ISSN: 0921-8890.

Yamamoto, Y. and Yun, X. (1993). Coordinating locomotion and manipulation of a mobile manipulator. In: Y. F. Zheng (Ed.) ***Recent trends in mobile robots***. Series in robotics and automated systems; v.11: 157-181. Singapore ; London. World Scientific. ISBN: 9810215118.

Yamamoto, Y. and Yun, X. (1996). Effect of the dynamic interaction on coordinated control of mobile manipulators. ***Robotics and Automation, IEEE Transactions on***. *Vol: 12(5): 816-824.* Oct. 1996. ISSN: 1042-296X.

Yang, G., Chen, I.-M., Lim, W. K., et al. (2001). Kinematic design of modular reconfigurable in-parallel robots. ***Autonomous Robots***. *Vol: 10(1): 83-89.* Jan. 2001. ISSN: 0929-5593.

Yim, M., Roufas, K., Duff, D., et al. (2003). Modular Reconfigurable Robots in Space Applications. ***Autonomous Robots***. *Vol: 14(2/3): 225 - 237.* 2003. ISSN: 0929-5593.

Young, B. J., Lawton, J. R. and Beard, R. W. (2000). ***Two hybrid control schemes for nonholonomic robots***. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on. San Francisco, CA. 24-28 April 2000. Vol: 2. pp. 1824 - 1829. ISBN: 0-7803-5886-4.

Yun, X. and Yamamoto, Y. (1997). Stability analysis of the internal dynamics of a wheeled mobile robot. ***Journal of Robotic Systems***. *Vol: 14(10): 697-709.* Oct. 1997. ISSN: 07412223.

Yun, X. and Sarkar, N. (1998). Unified formulation of robotic systems with holonomic and nonholonomic constraints. ***Robotics and Automation, IEEE Transactions on***. *Vol: 14(4): 640-650.* ISSN: 1042-296X.

Zhao, Y. and BeMent, S. L. (1992). ***Kinematics, dynamics and control of wheeled mobile robots***. Robotics and Automation, Proceedings of the IEEE International Conference on. Nice, France. 12-14 May 1992. Vol: 1. pp. 91 - 96. ISBN: 0-8186-2720-4.

Zheng, Y. F. (1993). ***Recent trends in mobile robots.*** World Scientific series in robotics and automated systems. Singapore, London : World Scientific. ISBN/ISSN: 9810215118.

Zorriassatine, F., Gonzalez-Villela, V. and Crusem, J.-P. (2002a). Possible ideas and projects for improvement of the Royal Mail Pipelines. ***Report***. Mechatronics Research Centre, Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University. Loughborough.

Zorriassatine, F., Gonzalez-Villela, V. J. and Crusem, J.-P. (2002b). A Tour of the Royal Mail Pipelines. ***Report***. Mechatronics Research Centre, Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University. Loughborough.

APPENDIX I ROBOT TYPE (2,0): Point stabilization

Simulation results

Figure A.I.1: Point stabilization: Sequence of posture snapshots

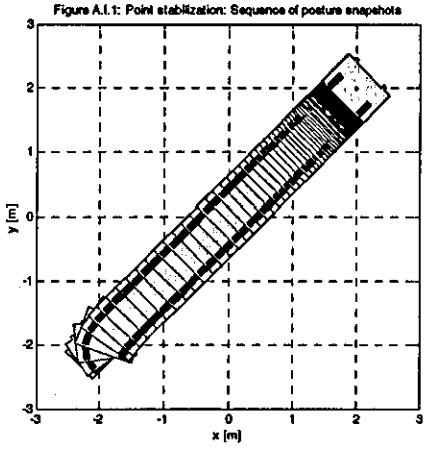


Figure A.I.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

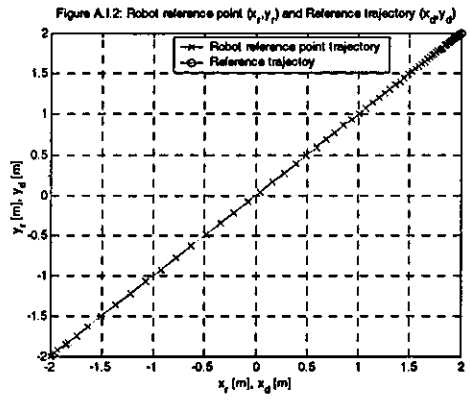


Figure A.I.3: Robot reference point posture variables: x_r, y_r and θ

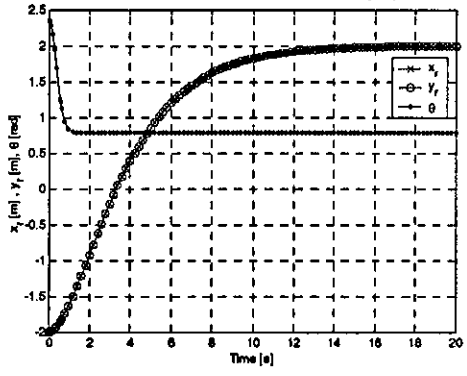


Figure A.I.4: Robot reference point tracking error: e_{xr} and e_{yr}

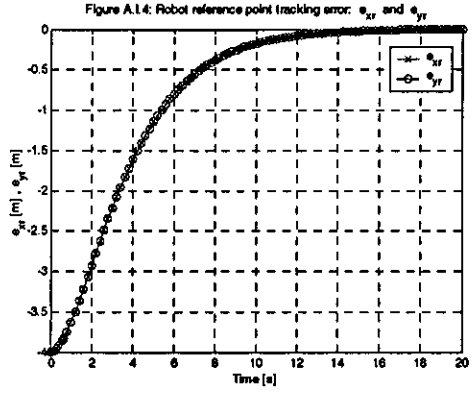


Figure A.I.5: Posture generalized velocities: x'_p, y'_p and θ'

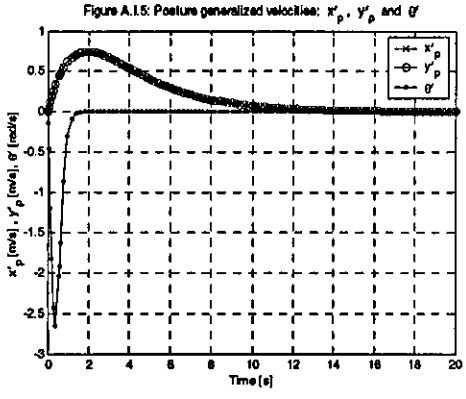


Figure A.I.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

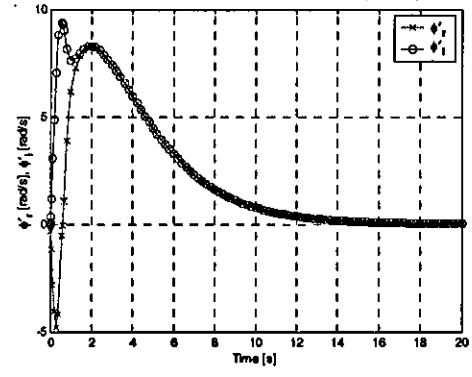


Figure A.I.7: Steering system velocities: x'_1 and θ'

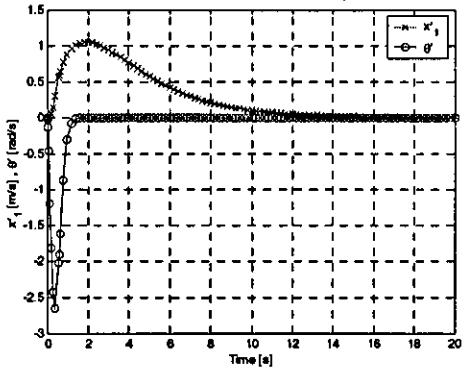


Figure A.I.8: Steering system accelerations: x''_1 and θ''

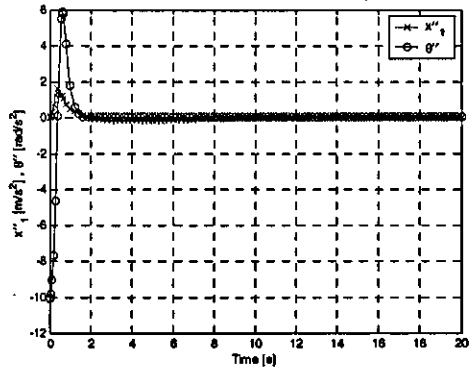


Figure A.I.9: Steering torques: Tx_1 and T_θ

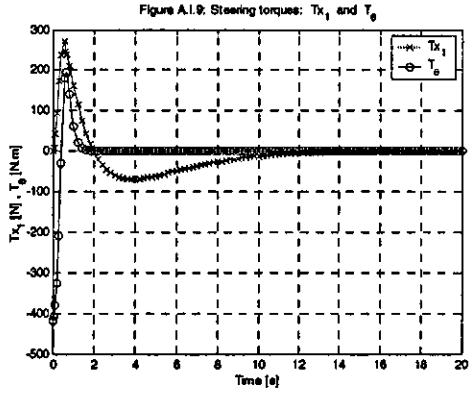


Figure A.I.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

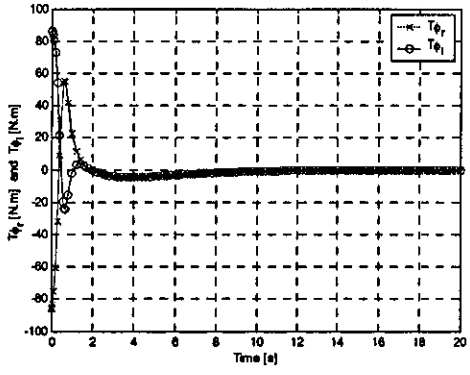
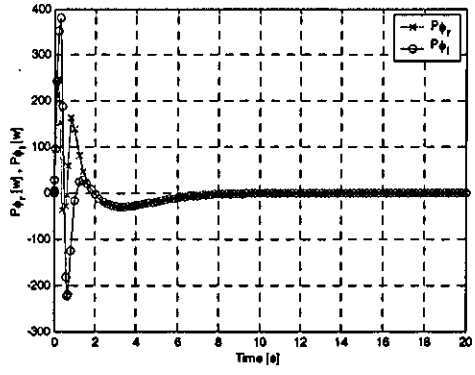


Figure A.I.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX II ROBOT TYPE (2,0): Line tracking. Constant acceleration

Simulation results

Figure A.II.1: Line tracking (constant acceleration): Sequence of posture snapshots

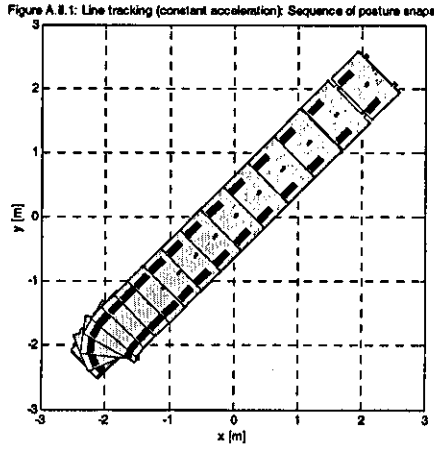


Figure A.II.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

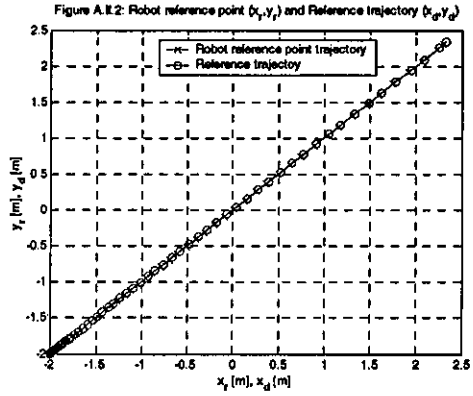


Figure A.II.3: Robot reference point posture variables: x_r, y_r and θ

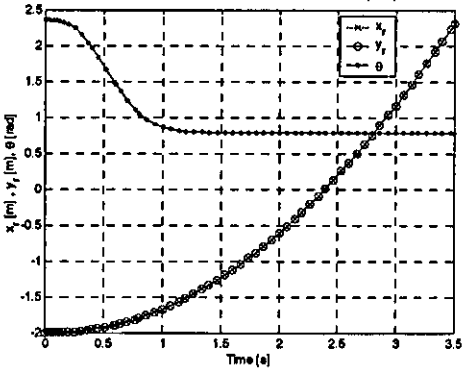


Figure A.II.4: Robot reference point tracking error: e_{xr} and e_{yr}

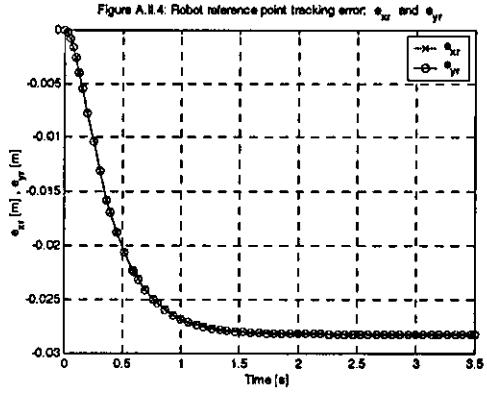


Figure A.II.5: Posture generalized velocities: x'_p, y'_p and θ'

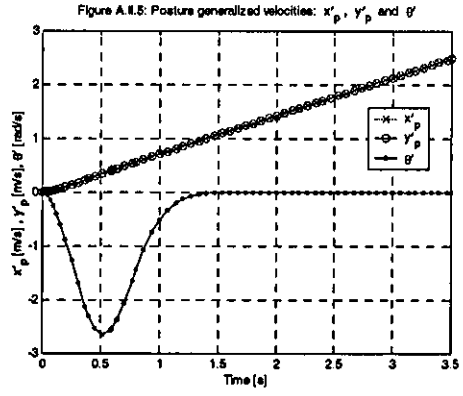


Figure A.II.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

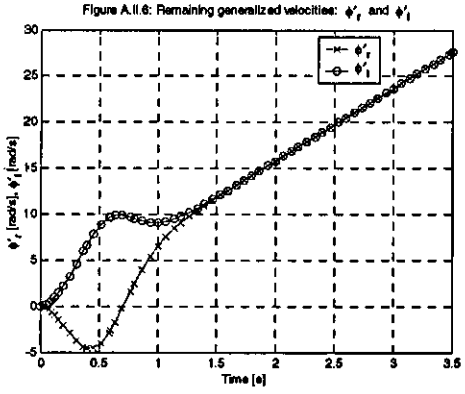


Figure A.II.7: Steering system velocities: x'_1 and θ'

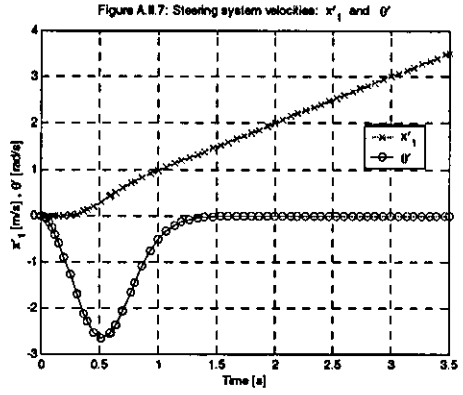


Figure A.II.8: Steering system accelerations: x''_1 and θ''

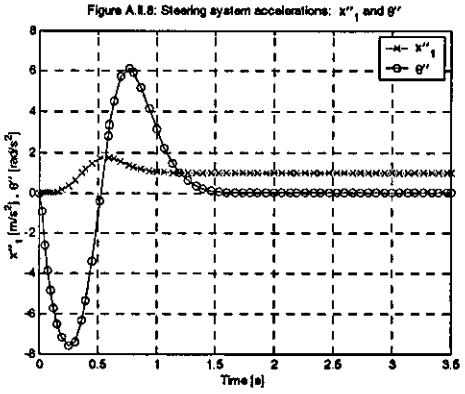


Figure A.II.9: Steering torques: Tx_1 and T_θ

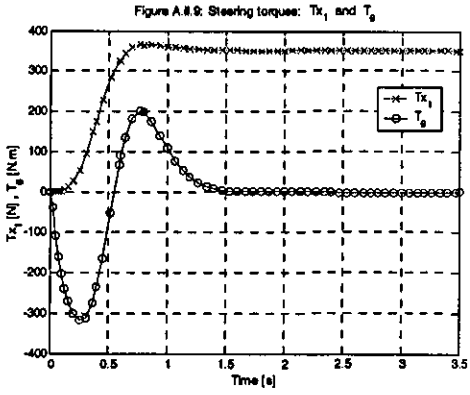


Figure A.II.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

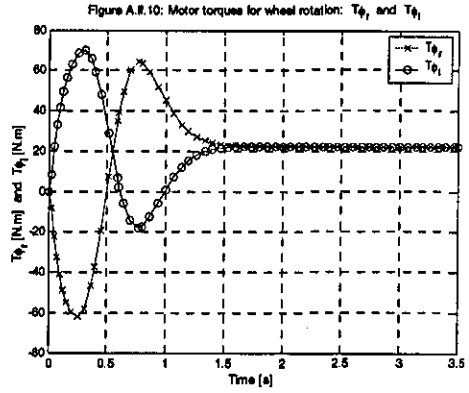
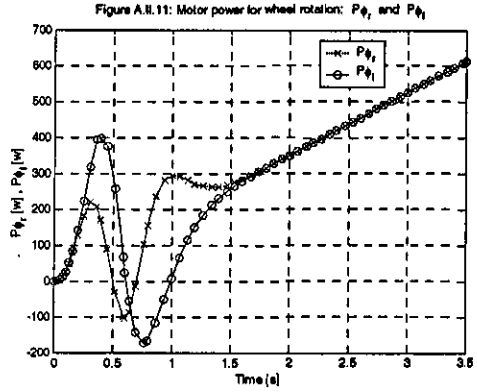


Figure A.II.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX III ROBOT TYPE (2,0): Line tracking. Constant deceleration

Simulation results

Figure A.III.1: Line tracking (constant deceleration): Sequence of posture snapshots

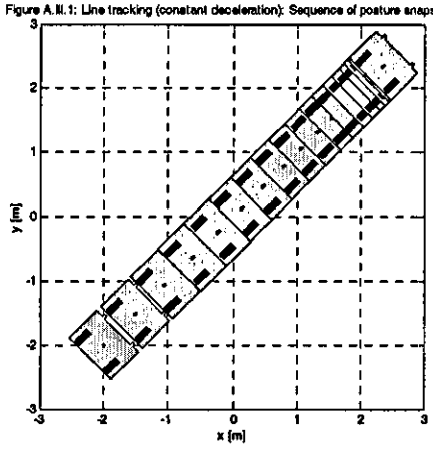


Figure A.III.2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

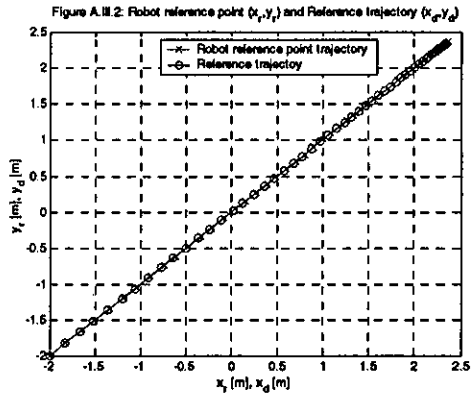


Figure A.III.3: Robot reference point posture variables: x_r, y_r and θ

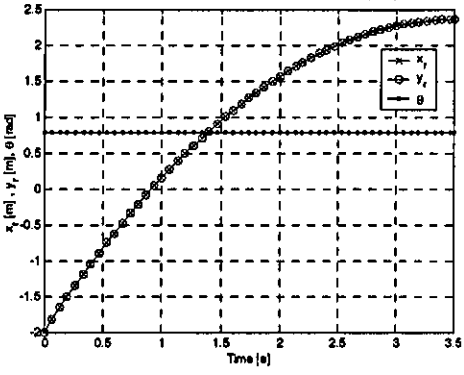


Figure A.III.4: Robot reference point tracking error: e_{xr} and e_{yr}

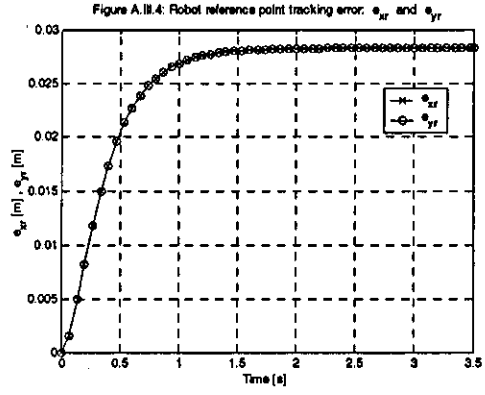


Figure A.III.5: Posture generalized velocities: x'_p, y'_p and θ'

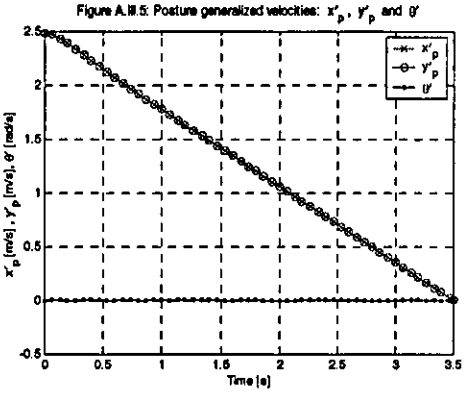


Figure A.III.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

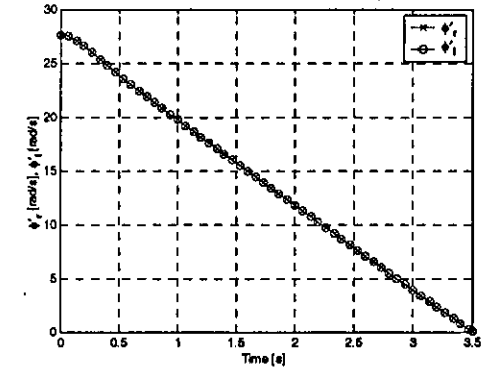


Figure A.III.7: Steering system velocities: x'_1 and θ'

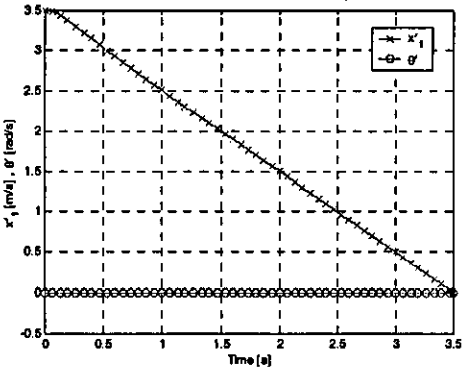


Figure A.III.8: Steering system accelerations: x''_1 and θ''

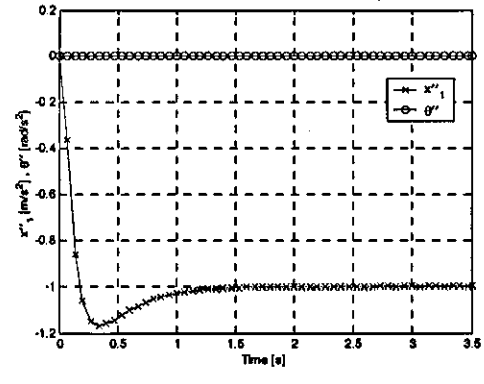


Figure A.III.9: Steering torques: Tx_1 and T_s

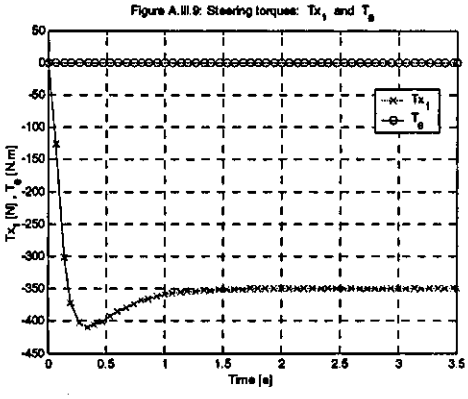


Figure A.III.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

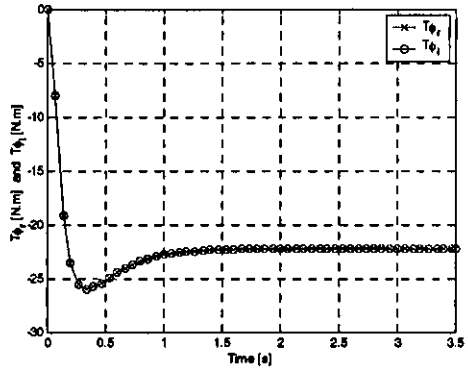
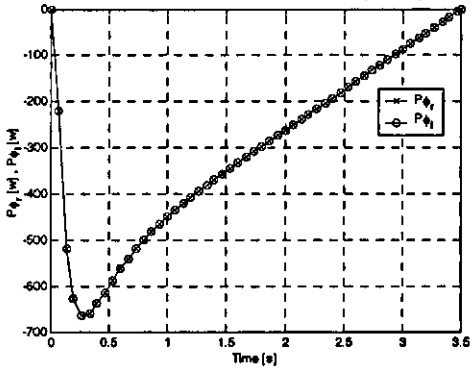


Figure A.III.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX IV ROBOT TYPE (2,0): Line tracking. Constant velocity

Simulation results

Figure A.IV.1: Line tracking (constant velocity): Sequence of posture snapshots

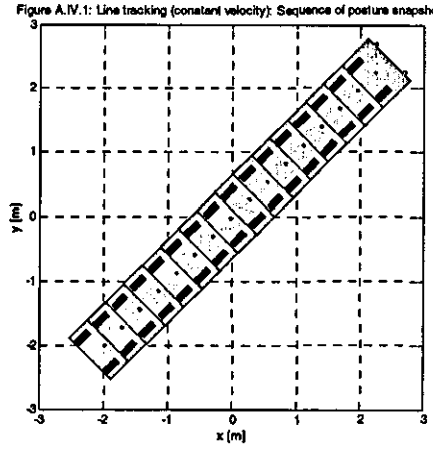


Figure A.IV.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

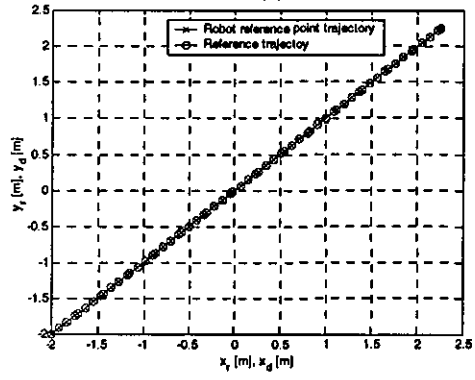


Figure A.IV.3: Robot reference point posture variables: x_r, y_r and θ

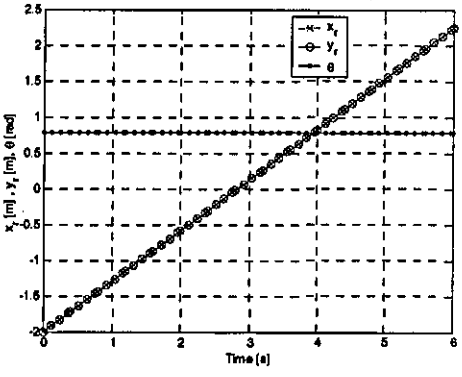


Figure A.IV.4: Robot reference point tracking error: e_{xr} and e_{yr}

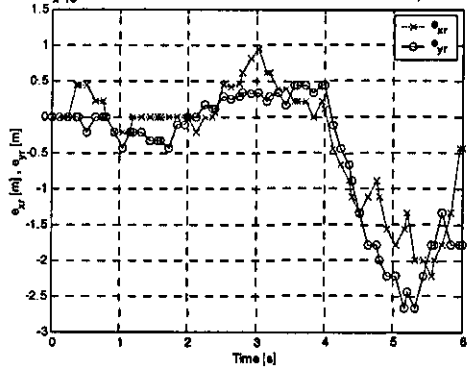


Figure A.IV.5: Posture generalized velocities: x'_p, y'_p and θ'

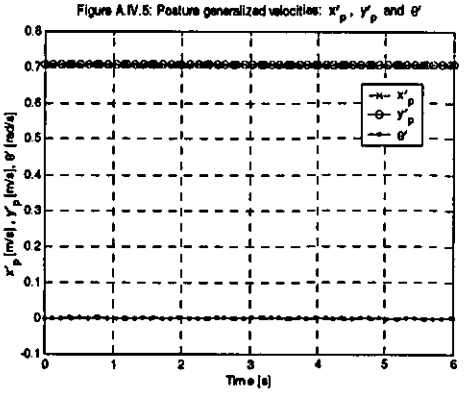


Figure A.IV.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

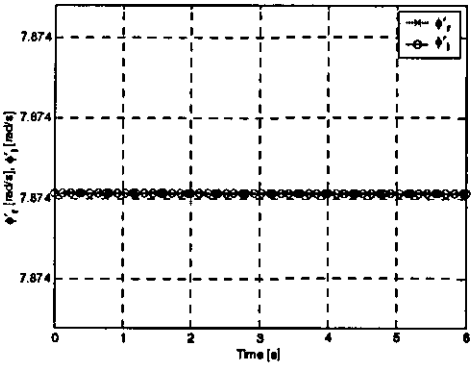


Figure A.IV.7: Steering system velocities: x'_1 and θ'

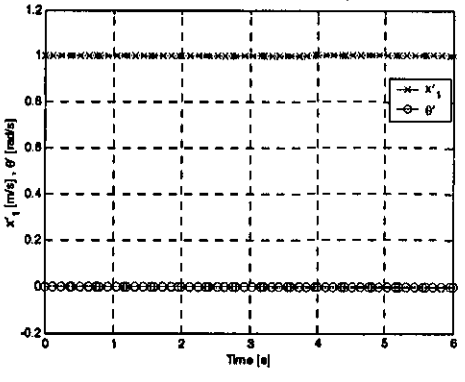
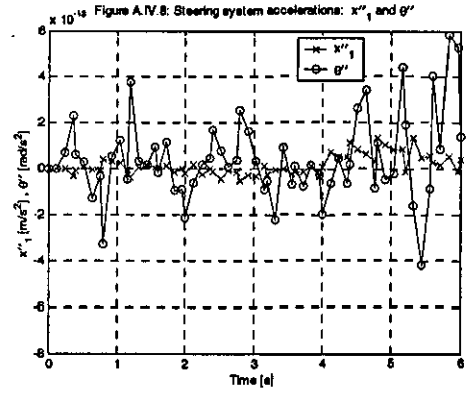


Figure A.IV.8: Steering system accelerations: x''_1 and θ''



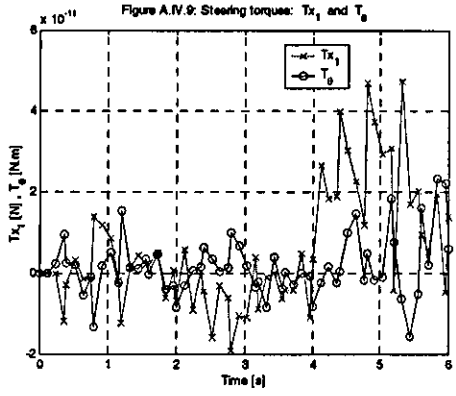


Figure A.IV.9: Steering torques: Tx_l and T_θ

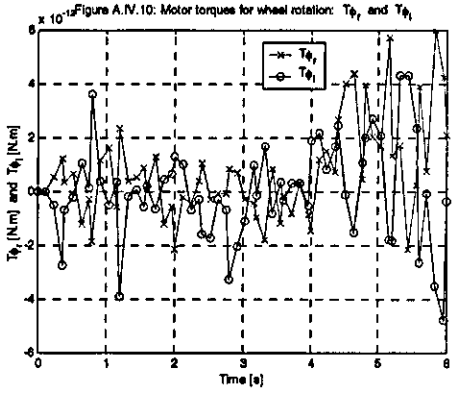


Figure A.IV.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

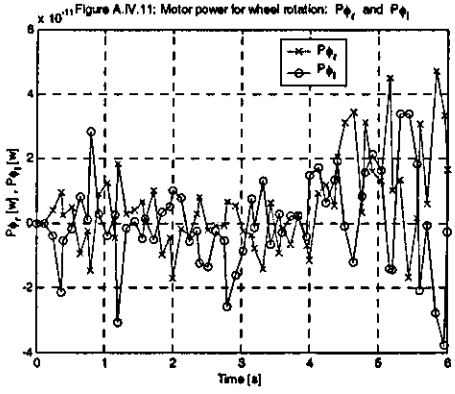


Figure A.IV.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$

APPENDIX V ROBOT TYPE (2,0): Circle tracking. Constant velocity

Simulation results

Figure A.V.1: Circle tracking (constant velocity): Sequence of posture snapshots

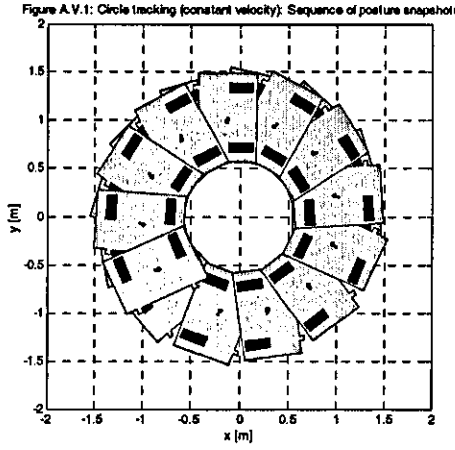


Figure A.V.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

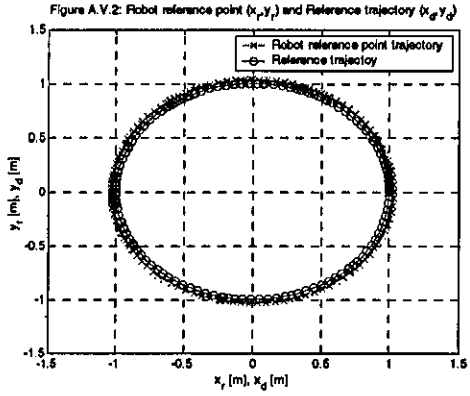


Figure A.V.3: Robot reference point posture variables: x_r, y_r and θ

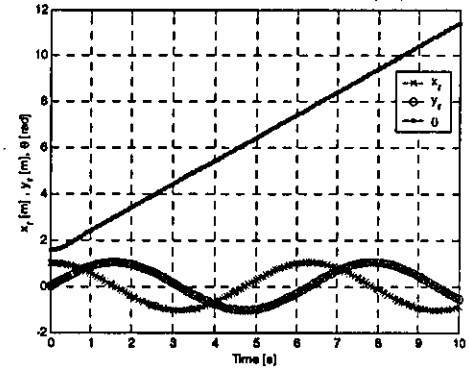


Figure A.V.4: Robot reference point tracking error: e_{xr} and e_{yr}

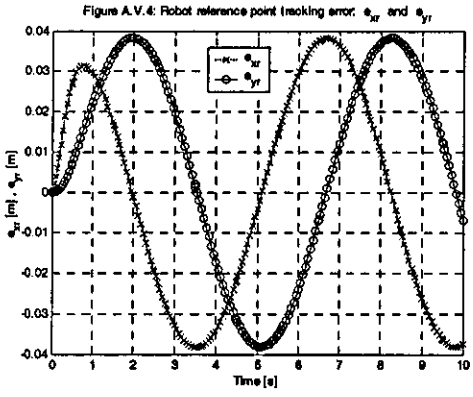


Figure A.V.5: Posture generalized velocities: x'_p, y'_p and θ'

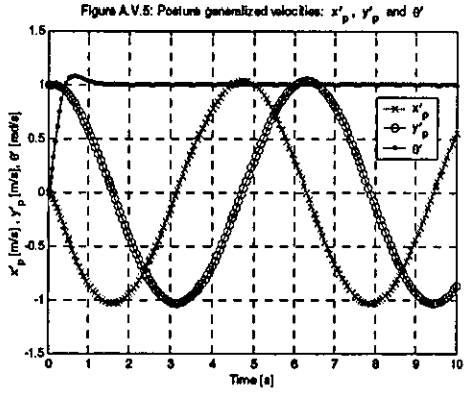


Figure A.V.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

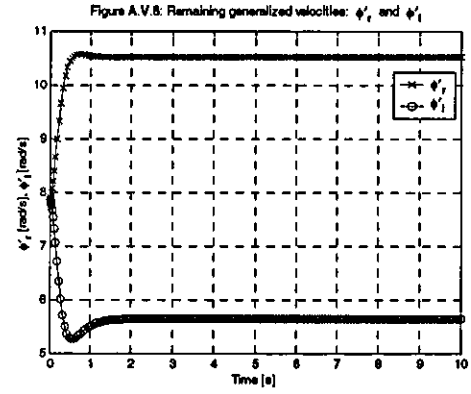


Figure A.V.7: Steering system velocities: x'_1 and θ'

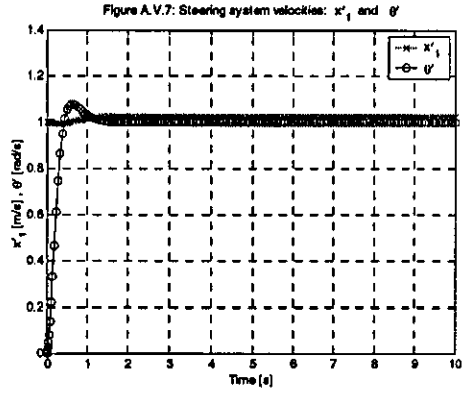


Figure A.V.8: Steering system accelerations: x''_1 and θ''

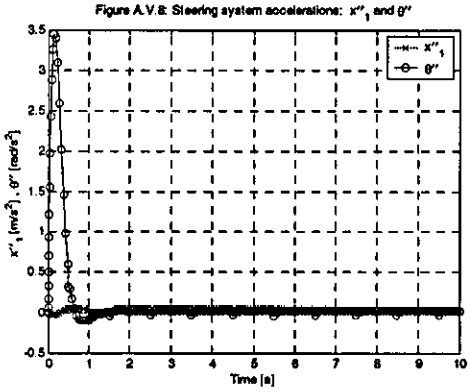


Figure A.V.9: Steering torques: Tx_1 and T_θ

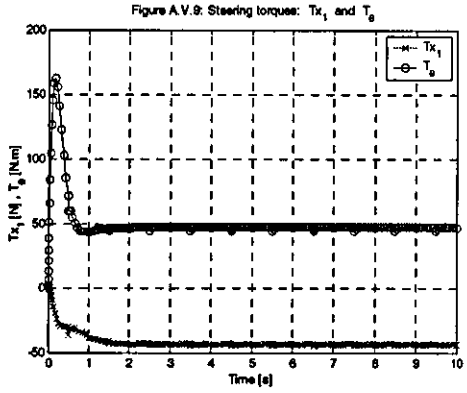


Figure A.V.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

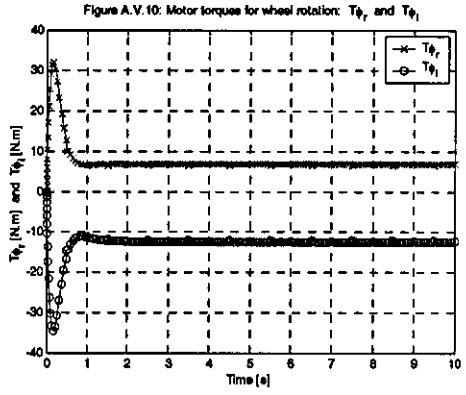
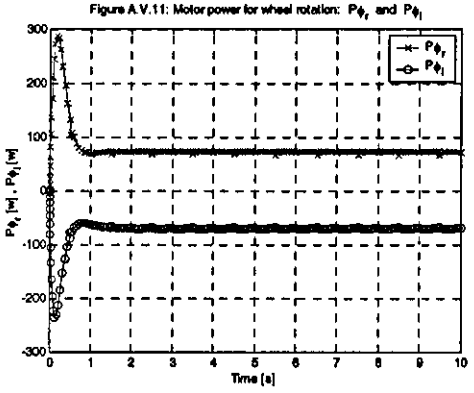


Figure A.V.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX VI ROBOT TYPE (2,0): Circle-line tracking. Constant velocity

Simulation results

Figure A.VI.1: Circle-line tracking (constant velocity): Sequence of posture snapshots

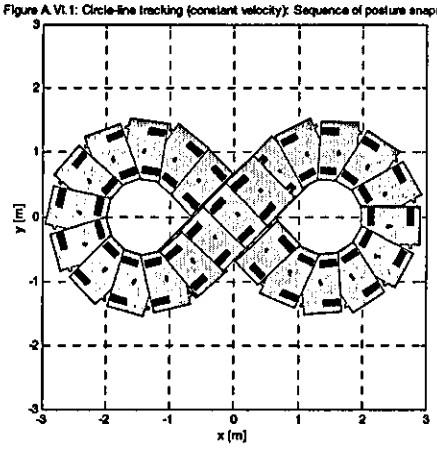


Figure A.VI.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

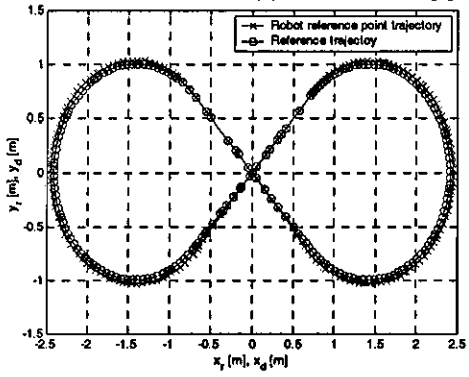


Figure A.VI.3: Robot reference point posture variables: x_r, y_r and θ

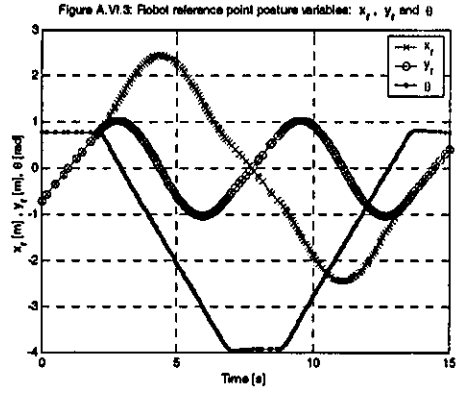


Figure A.VI.4: Robot reference point tracking error: e_{xr} and e_{yr}

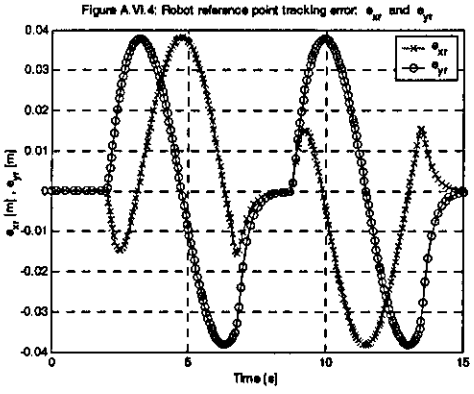


Figure A.VI.5: Posture generalized velocities: x'_p, y'_p and θ'

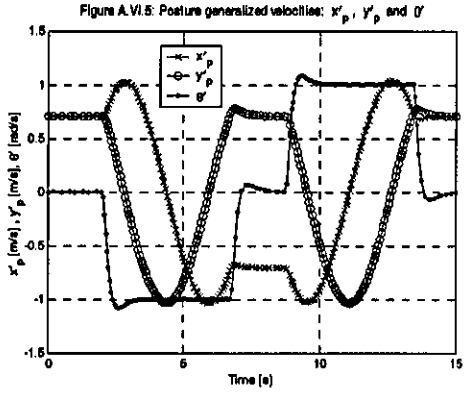


Figure A.VI.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

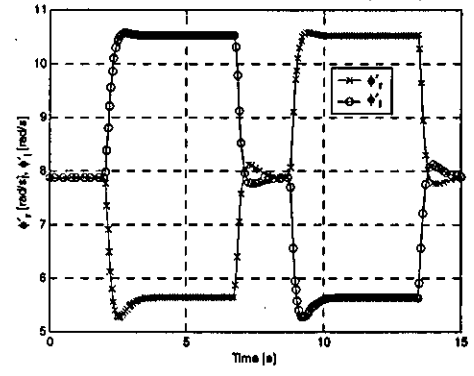


Figure A.VI.7: Steering system velocities: x'_1 and θ'

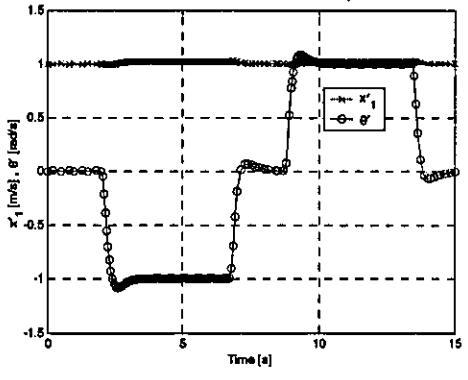


Figure A.VI.8: Steering system accelerations: x''_1 and θ''

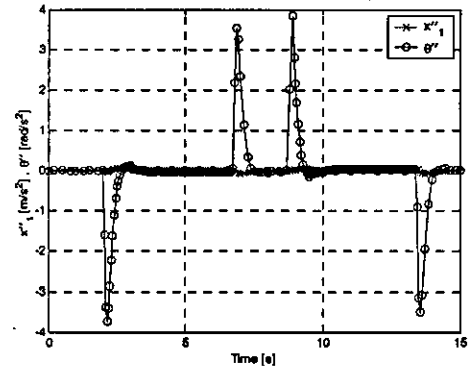


Figure A.VI.9: Steering torques: Tx_1 and T_θ

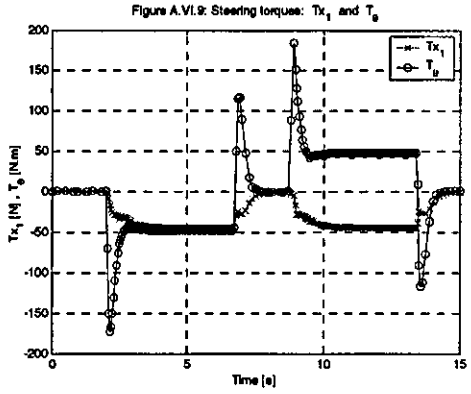


Figure A.VI.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

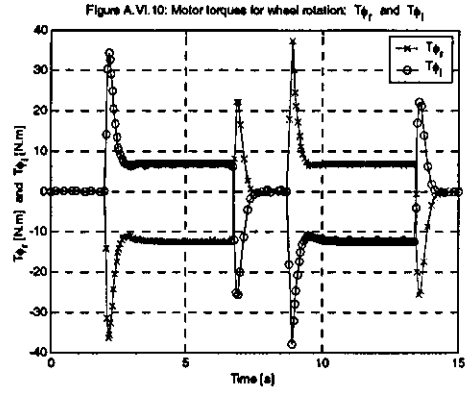
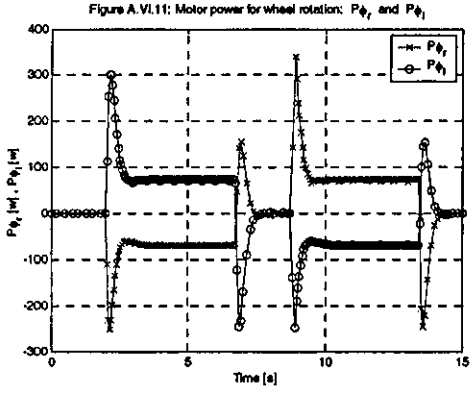


Figure A.VI.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX VII ROBOT TYPE (1,1): Point stabilization. Moving backwards. Load: 50 and 550

Simulation results

Figure A.VII.1: Point stabilization: Sequence of posture snapshots

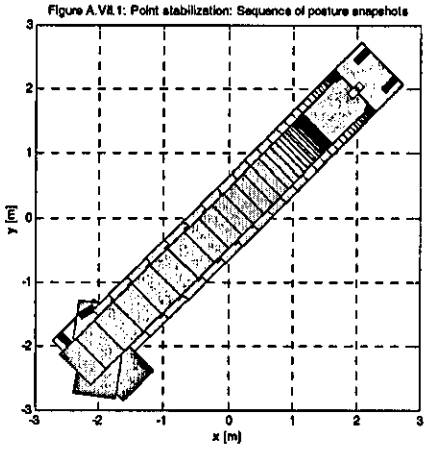


Figure A.VII.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

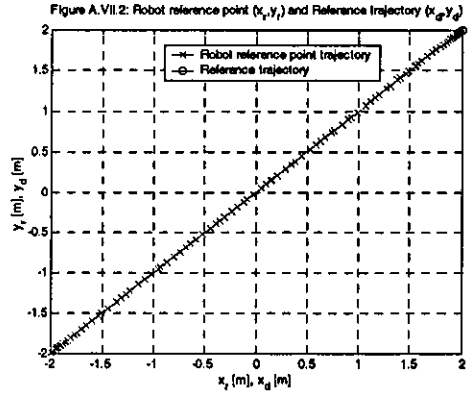


Figure A.VII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

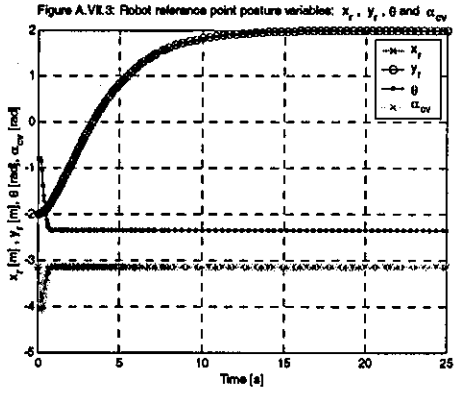


Figure A.VII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

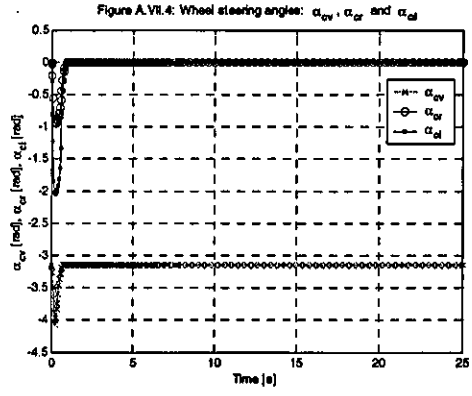


Figure A.VII.5: Robot reference point tracking error: e_{xr} and e_{yr}

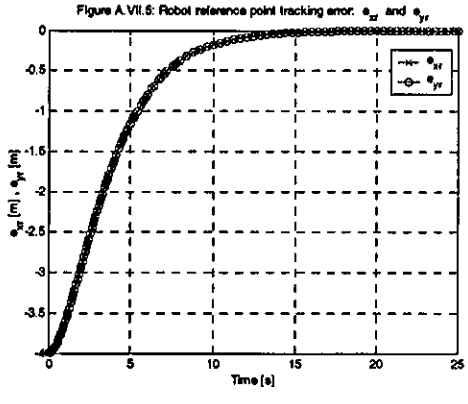


Figure A.VII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

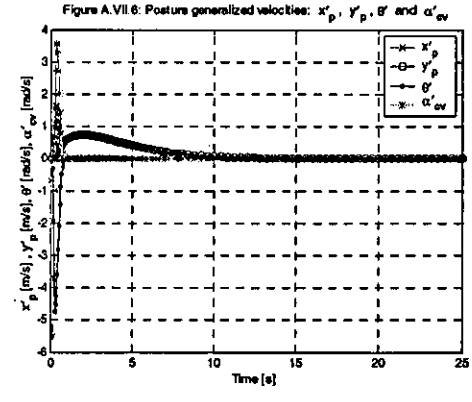


Figure A.VII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

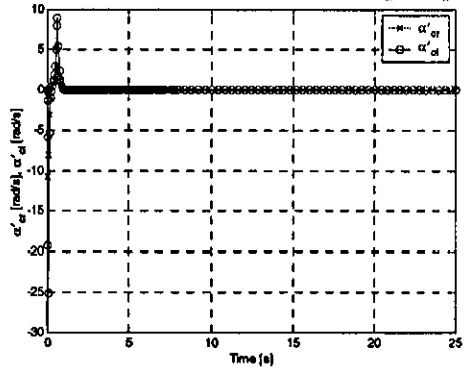


Figure A.VII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

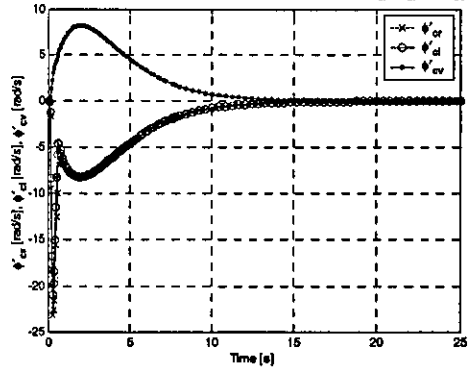


Figure A.VII.9: Steering system velocities: η and ς

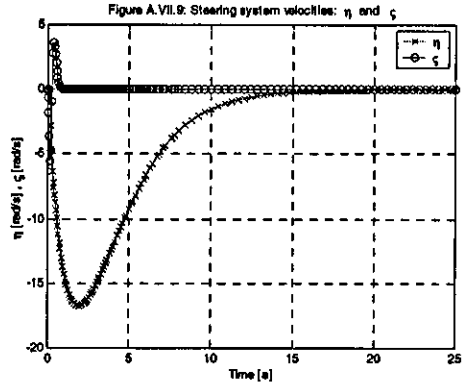
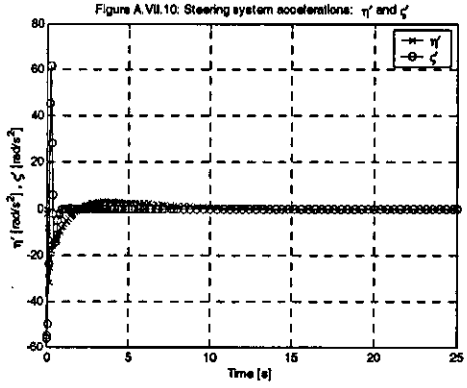


Figure A.VII.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.VII.11: Steering system torques: T_η and T_ς

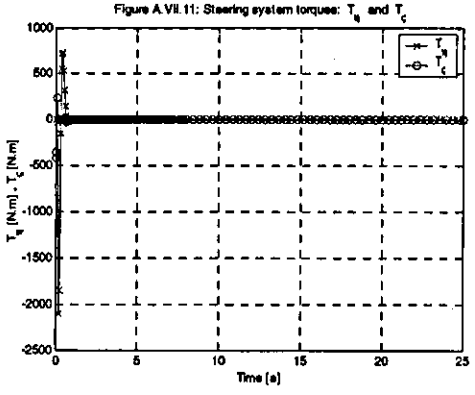


Figure A.VII.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

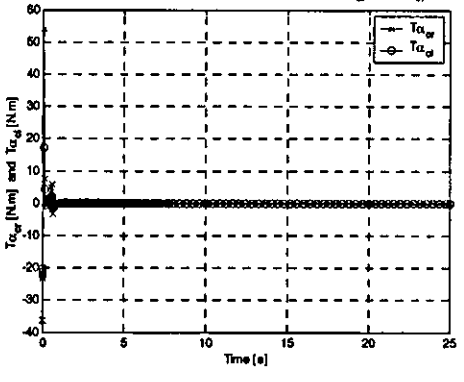


Figure A.VII.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

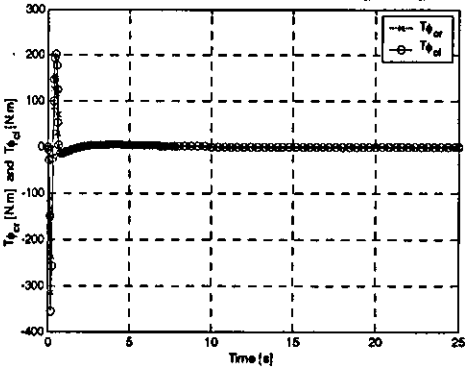


Figure A.VII.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

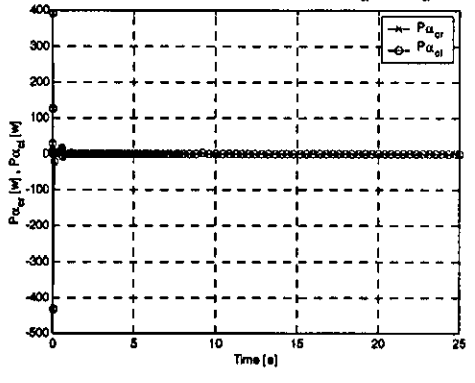
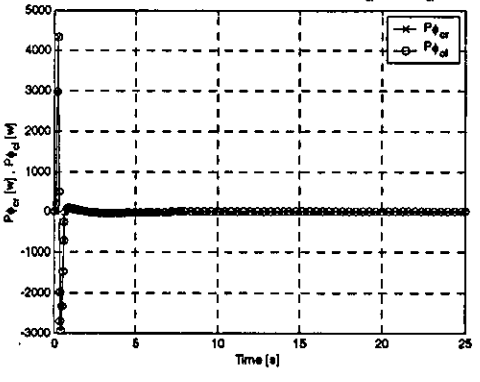


Figure A.VII.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.VII.16: Steering system torques: T_s and T_c

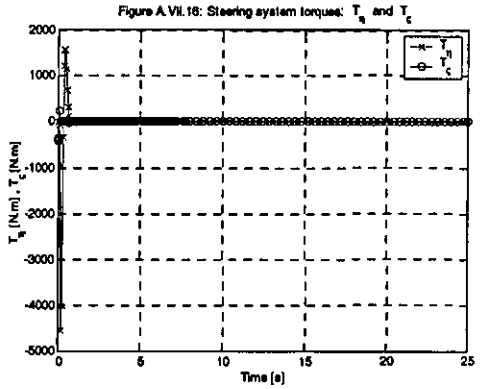


Figure A.VII.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

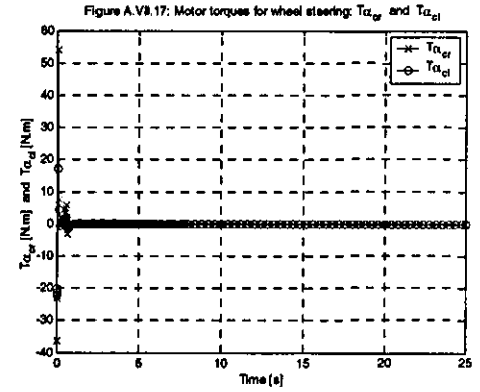


Figure A.VII.18: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

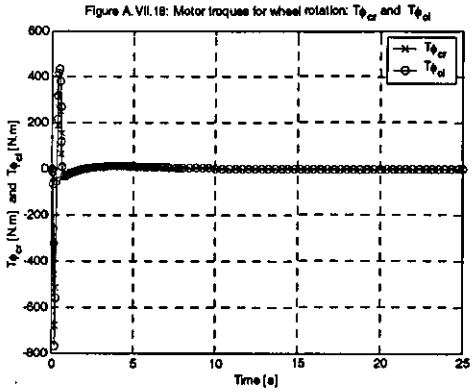


Figure A.VII.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

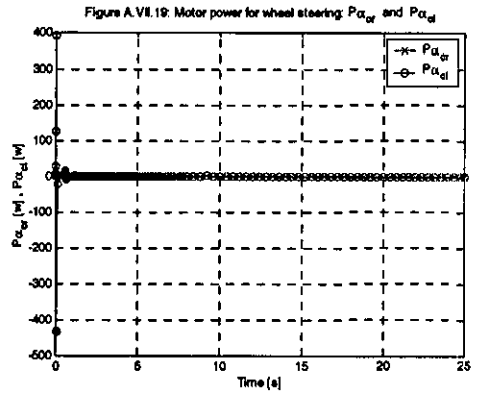
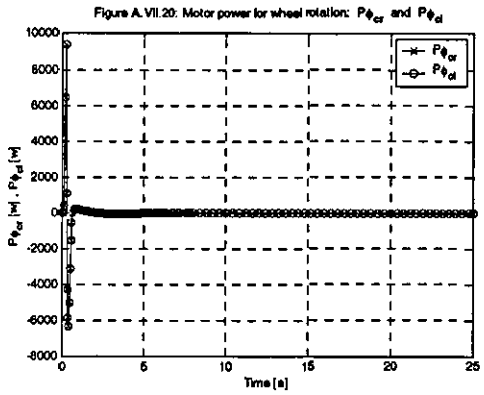


Figure A.VII.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX VIII ROBOT TYPE (1,1): Line tracking. Constant acceleration. Moving backwards. Load: 50 and 550

Simulation results

Figure A.VIII.1: Line tracking (constant acceleration): Sequence of posture snapshots

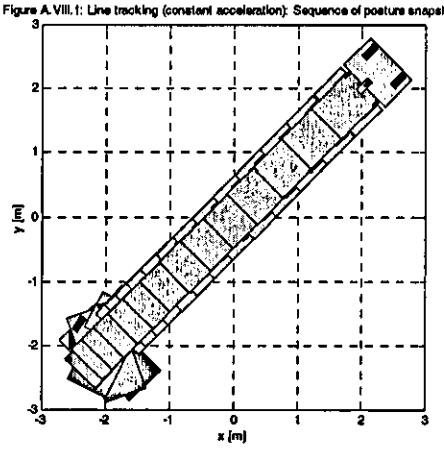


Figure A.VIII.2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

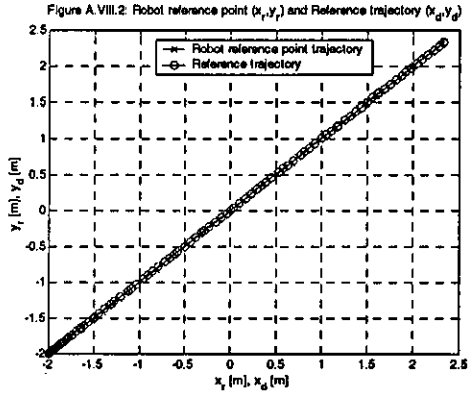


Figure A.VIII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

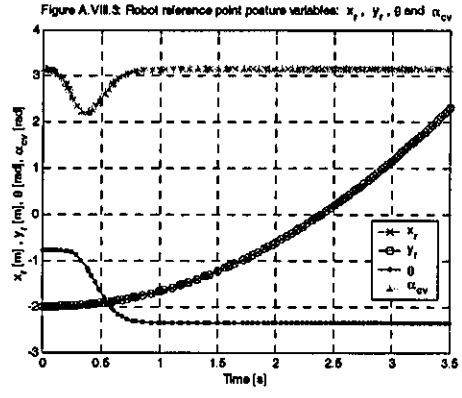


Figure A.VIII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

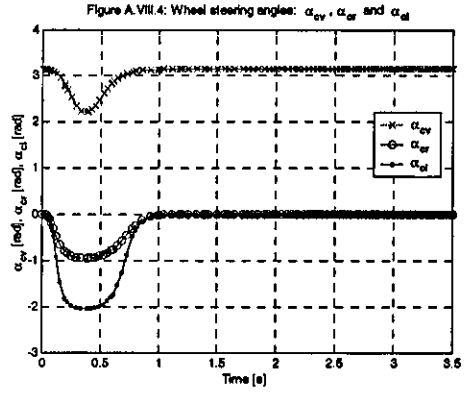


Figure A.VIII.5: Robot reference point tracking error: e_{xr} and e_{yr}

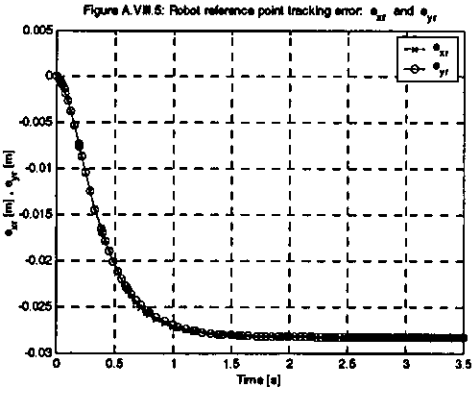


Figure A.VIII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

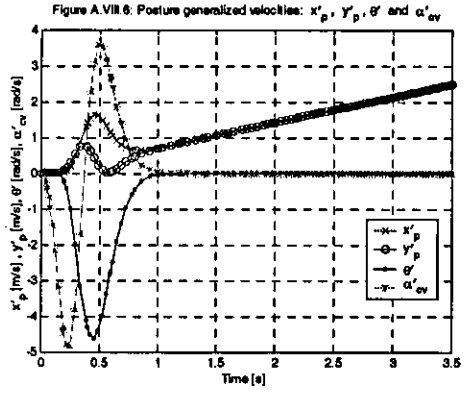


Figure A.VIII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

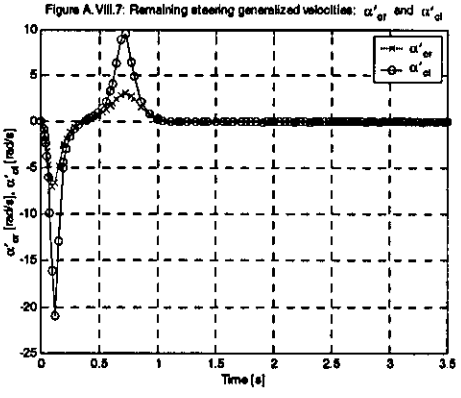


Figure A.VIII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

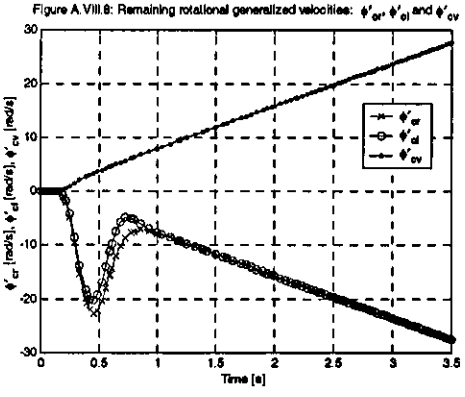


Figure A.VIII.9: Steering system velocities: η and ς

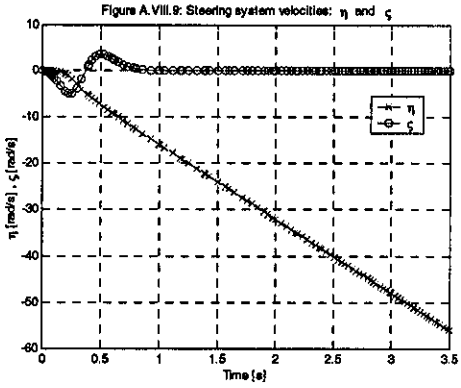
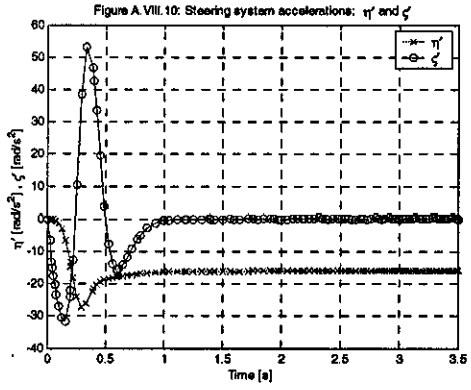


Figure A.VIII.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.VIII.11: Steering system torques: T_η and T_ς

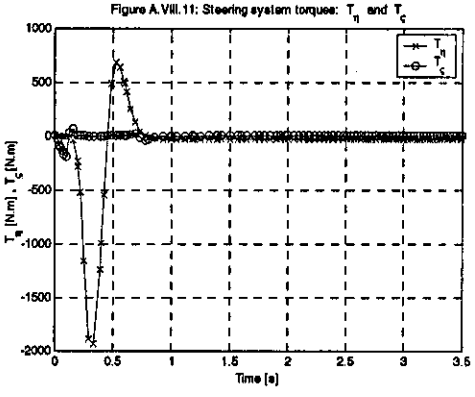


Figure A.VIII.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

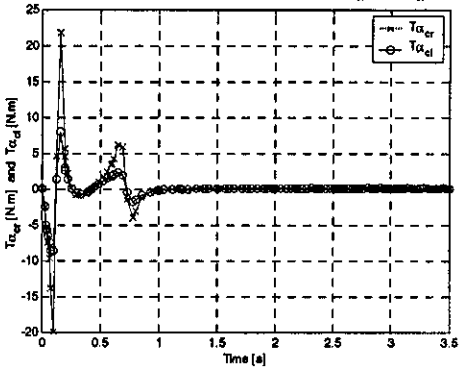


Figure A.VIII.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

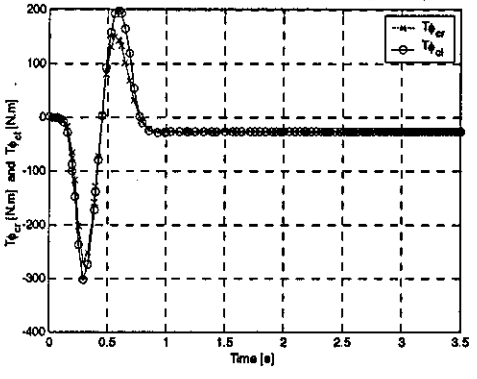


Figure A.VIII.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

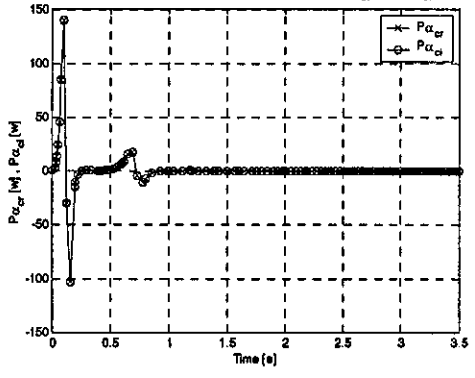
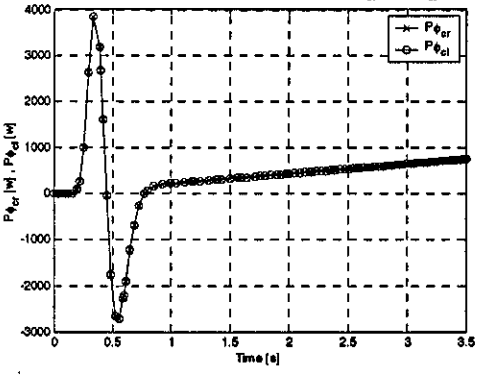


Figure A.VIII.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.VIII.16: Steering system torques: T_{ll} and T_c

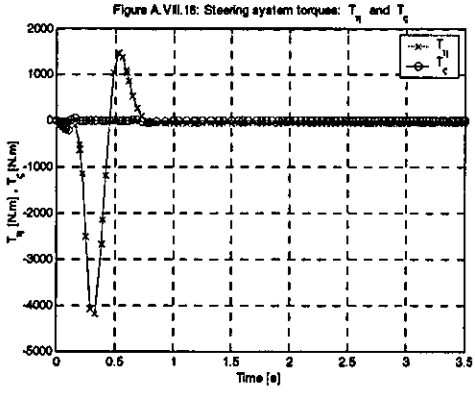


Figure A.VIII.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

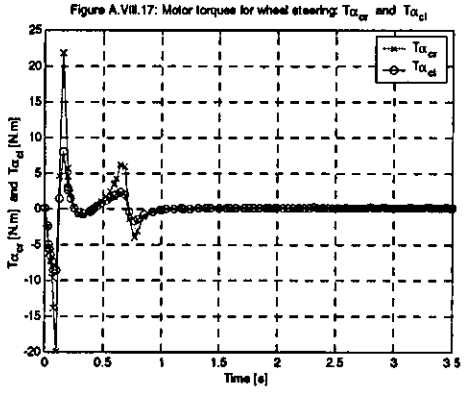


Figure A.VIII.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

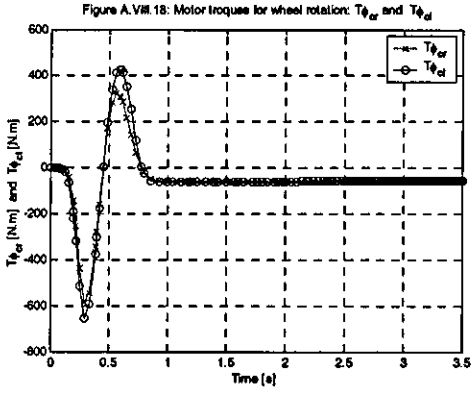


Figure A.VIII.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

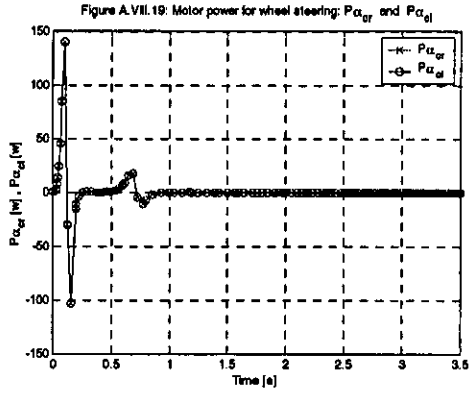
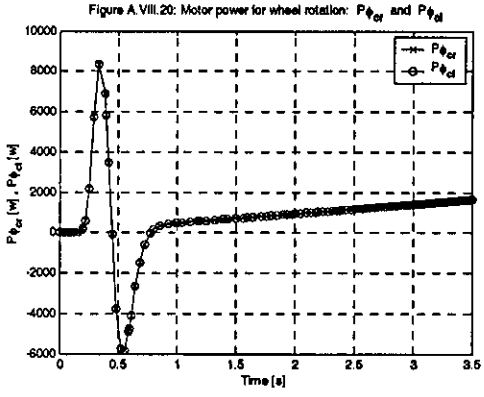


Figure A.VIII.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX IX ROBOT TYPE (1,1): Line tracking. Constant deceleration. Moving backwards. Load: 50 and 550

Simulation results

Figure A.IX.1: Line tracking (constant deceleration): Sequence of posture snapshots

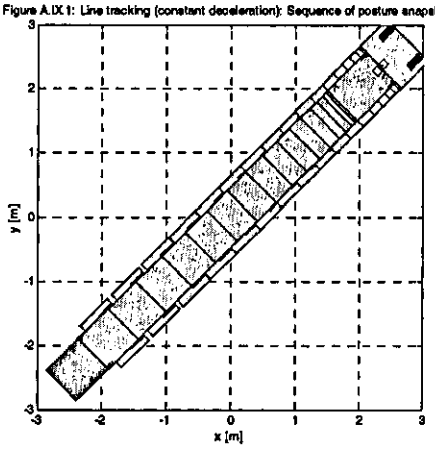


Figure A.IX.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

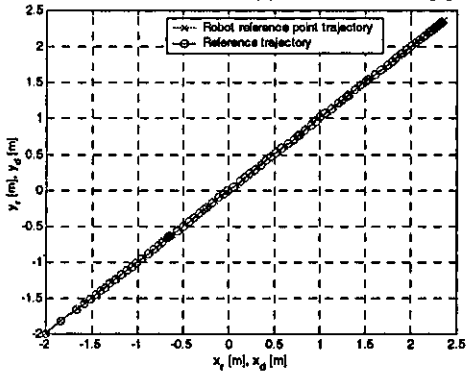


Figure A.IX.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

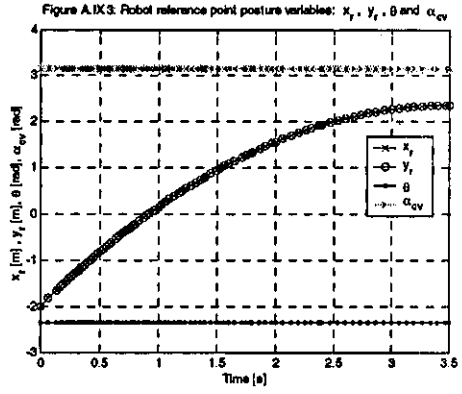


Figure A.IX.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

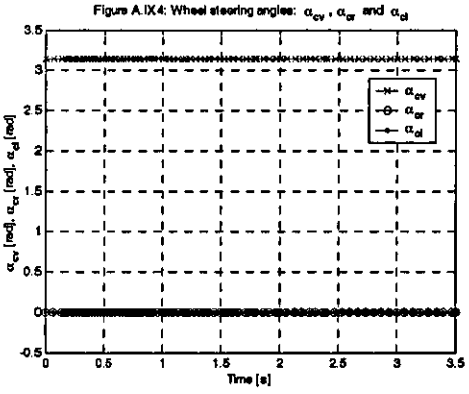


Figure A.IX.5: Robot reference point tracking error: e_{xr} and e_{yr}

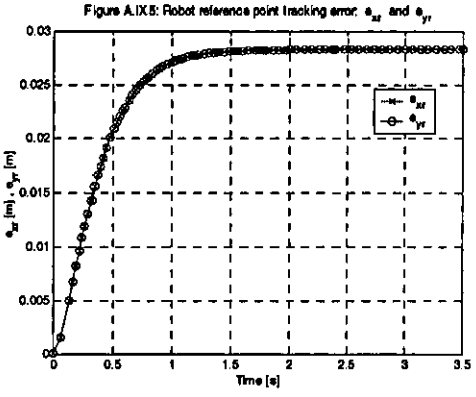


Figure A.IX.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

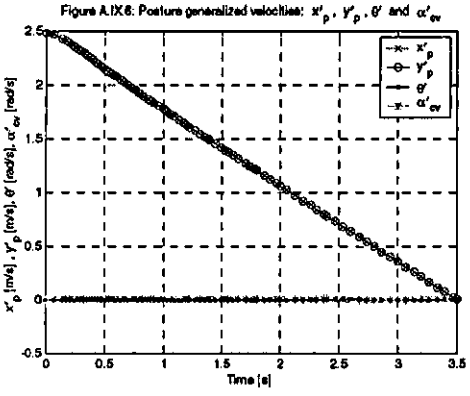


Figure A.IX.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

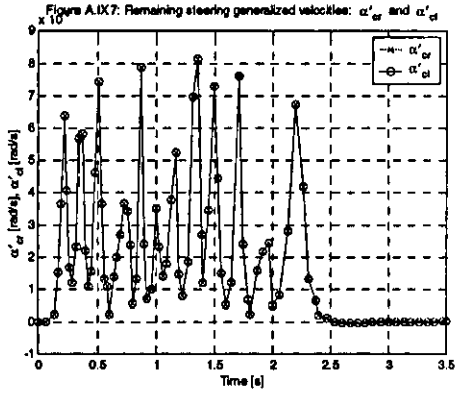


Figure A.IX.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

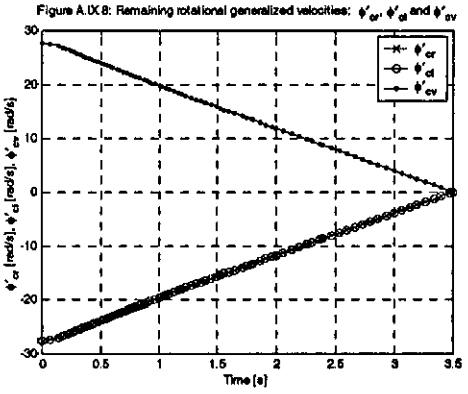


Figure A.IX.9: Steering system velocities: η and ζ

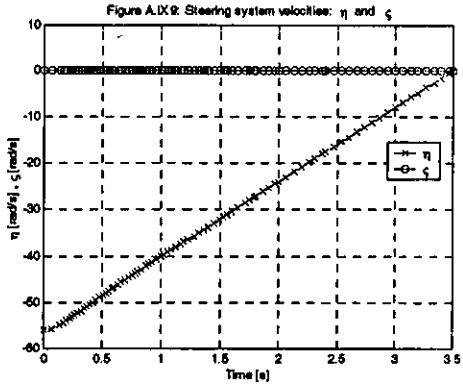
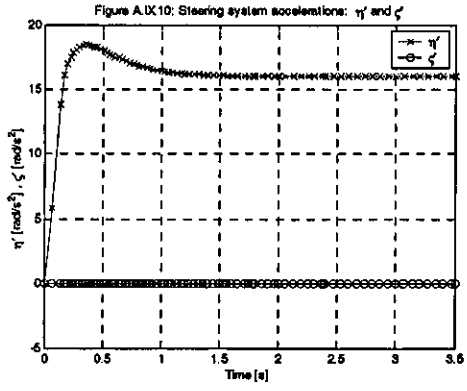


Figure A.IX.10: Steering system accelerations: η' and ζ'



Load = 50kg

Figure A.IX.11: Steering system torques: T_η and T_ζ

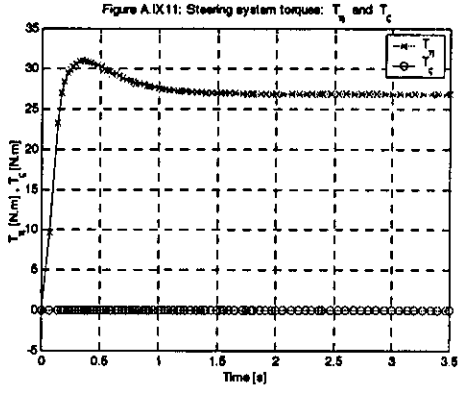


Figure A.IX.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

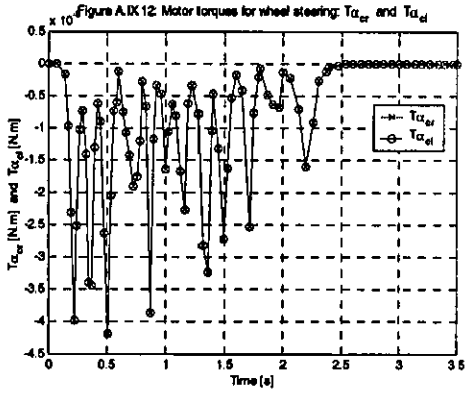


Figure A.IX.13: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

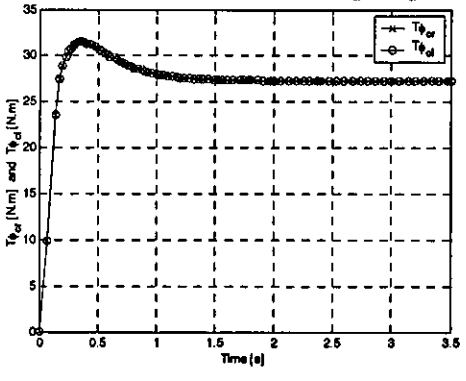


Figure A.IX.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

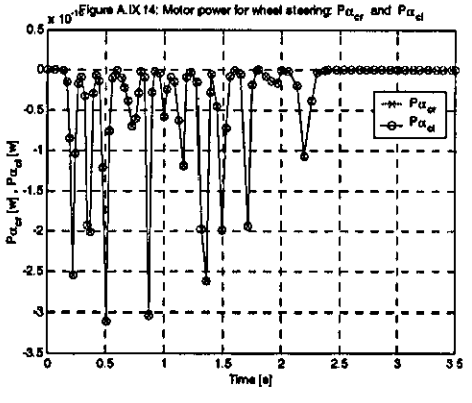
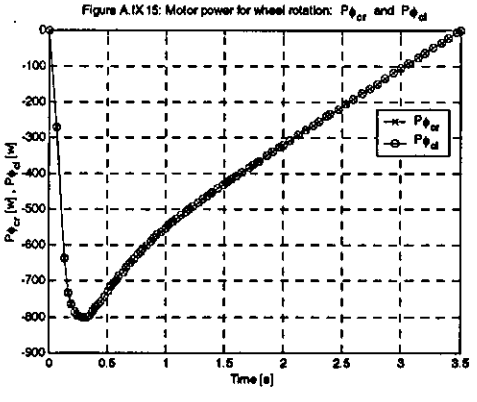


Figure A.IX.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.IX.16: Steering system torques: T_{sj} and T_c

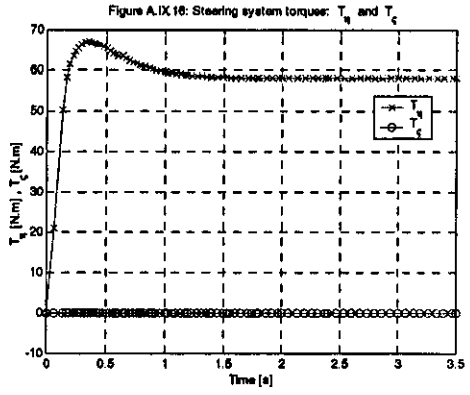


Figure A.IX.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

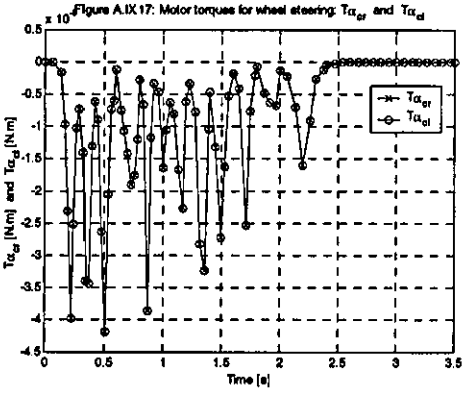


Figure A.IX.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

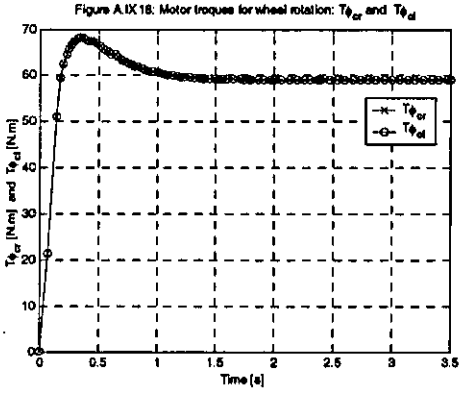


Figure A.IX.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

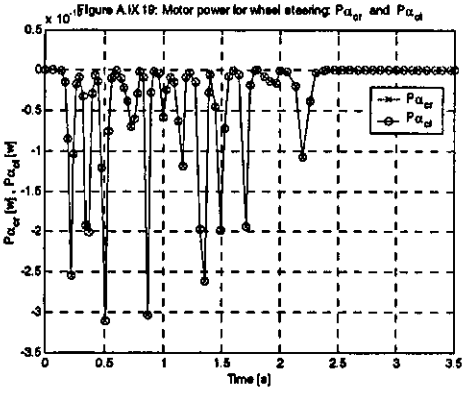
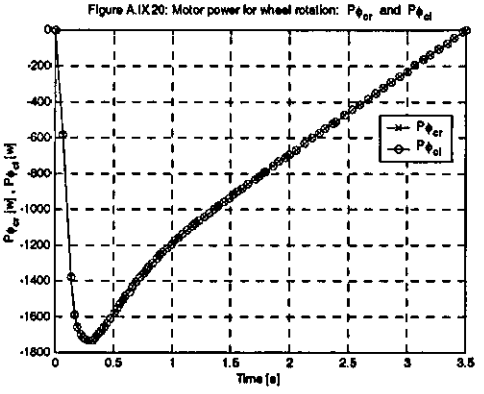


Figure A.IX.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX X ROBOT TYPE (1,1): Line tracking. Constant velocity. Moving backwards. Load: 50 and 550

Simulation results

Figure A.X1: Line tracking (constant velocity): Sequence of posture snapshots

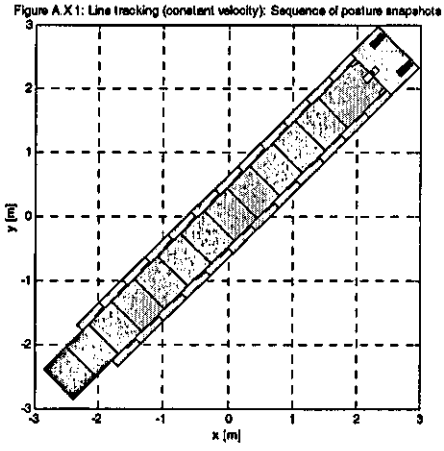


Figure A.X2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

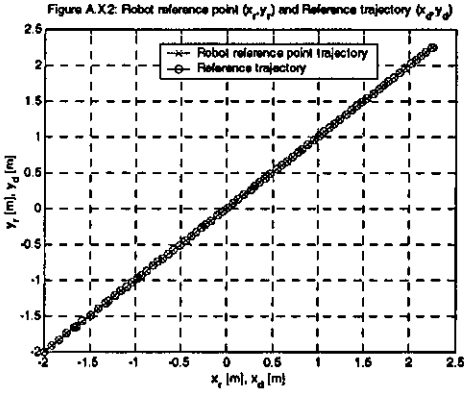


Figure A.X3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

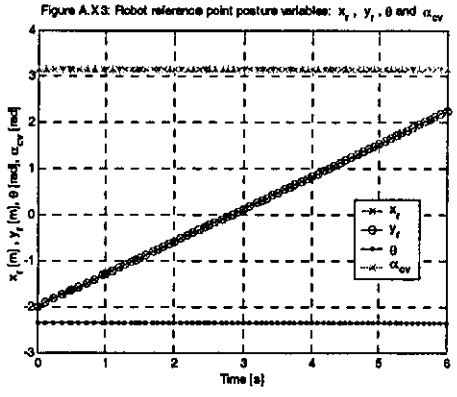


Figure A.X4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

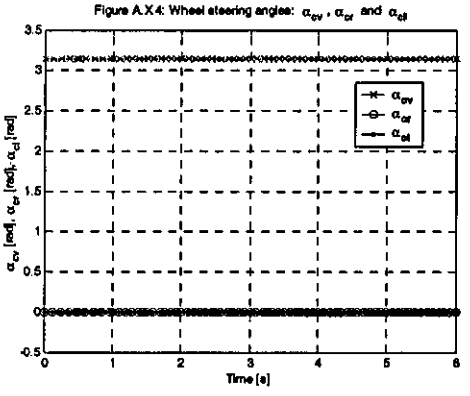


Figure A.X5: Robot reference point tracking error: e_{xr} and e_{yr}

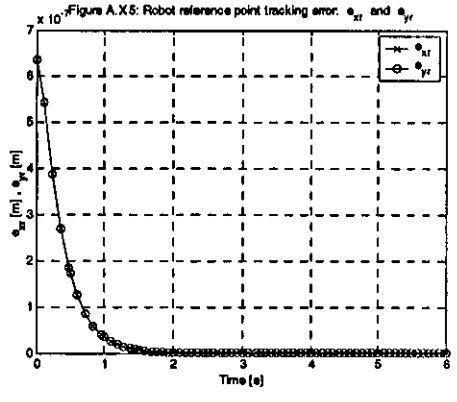


Figure A.X6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

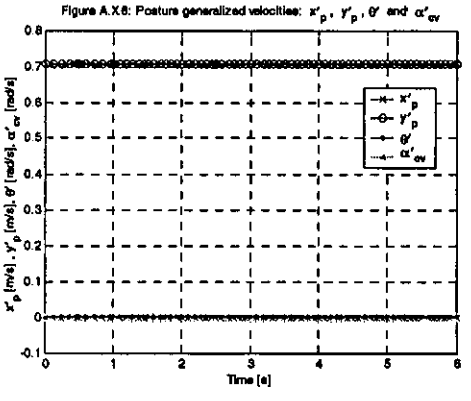


Figure A.X7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

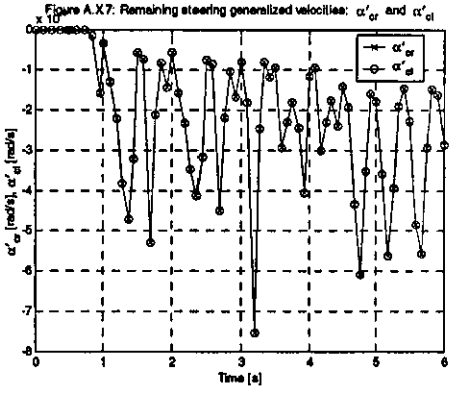
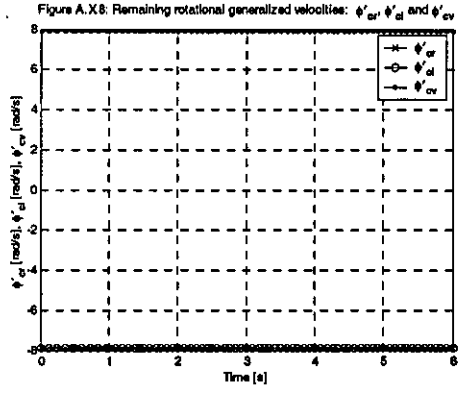


Figure A.X8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}



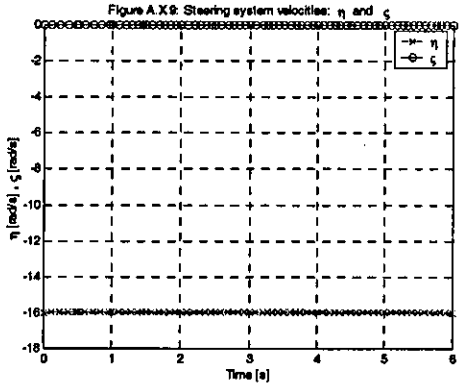


Figure A.X.9: Steering system velocities: η and ς
η [rad/s] , ς [rad/s]
Time [s]

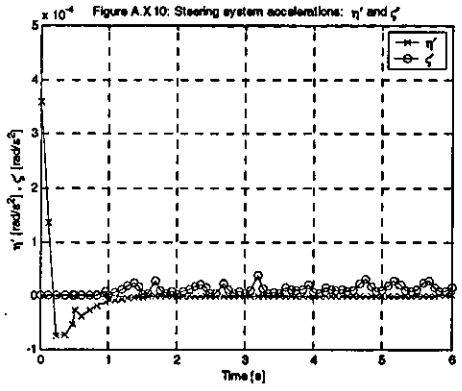


Figure A.X.10: Steering system accelerations: η' and ς'
η' [rad/s²] , ς' [rad/s²]
Time [s]

Load = 50kg

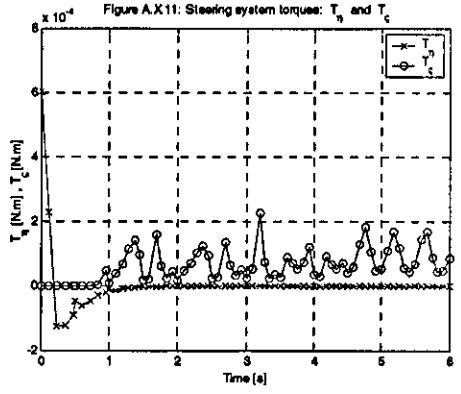


Figure A.X.11: Steering system torques: Tη and Tς
Tη [N.m] , Tς [N.m]
Time [s]

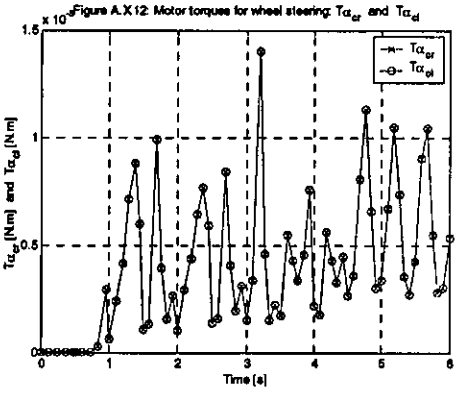


Figure A.X.12: Motor torques for wheel steering: Tαcr and Tαcl
Tαcr [N.m] and Tαcl [N.m]
Time [s]

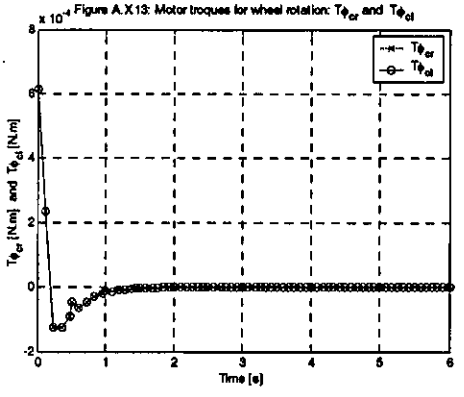


Figure A.X.13: Motor troques for wheel rotation: Tϕcr and Tϕcl
Tϕcr [N.m] and Tϕcl [N.m]
Time [s]

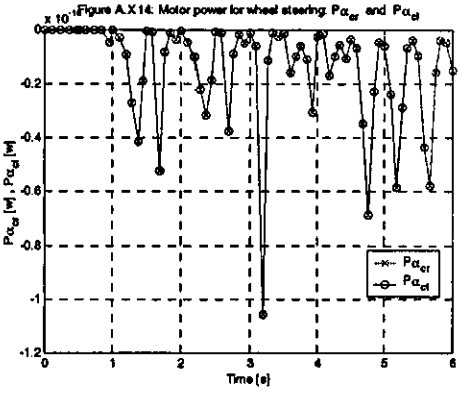


Figure A.X.14: Motor power for wheel steering: Pαcr and Pαcl
Pαcr [W] , Pαcl [W]
Time [s]

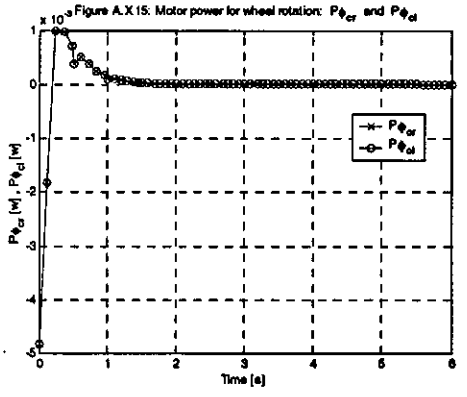


Figure A.X.15: Motor power for wheel rotation: Pϕcr and Pϕcl
Pϕcr [W] , Pϕcl [W]
Time [s]

Load = 550kg

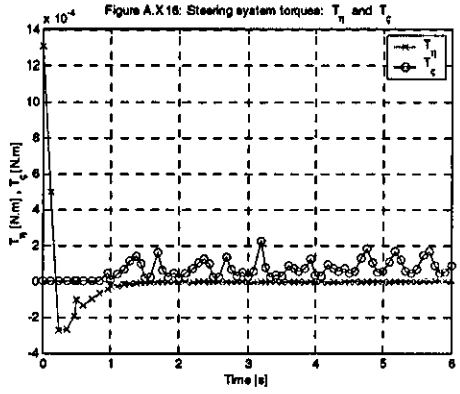


Figure A.X16: Steering system torques: T_η and T_ζ

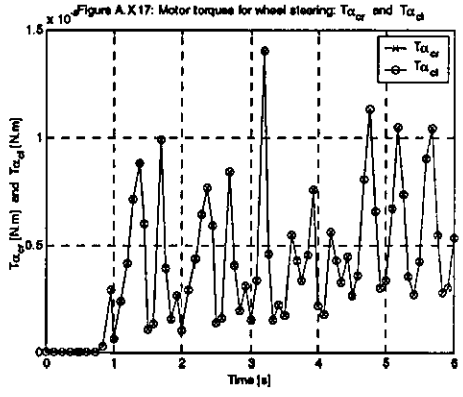


Figure A.X17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

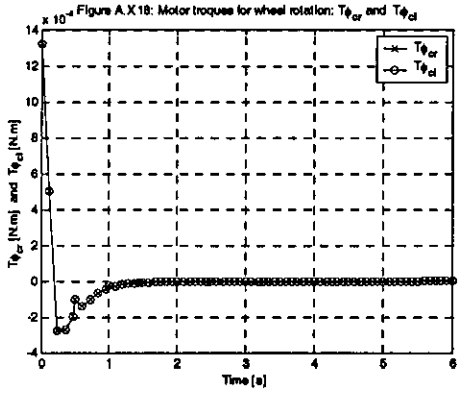


Figure A.X18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

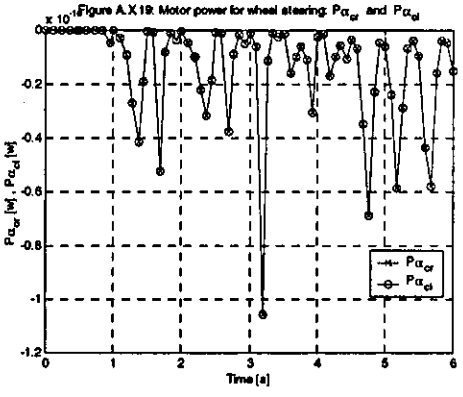


Figure A.X19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

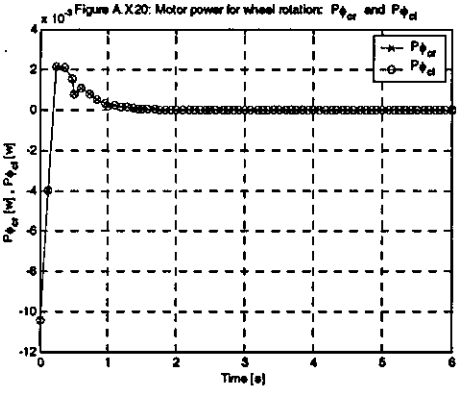


Figure A.X20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$

APPENDIX XI ROBOT TYPE (1,1): Circle tracking. Constant velocity. Moving backwards. Load: 50 and 550

Simulation results

Figure A.XI.1: Circle tracking (constant velocity): Sequence of posture snapshots

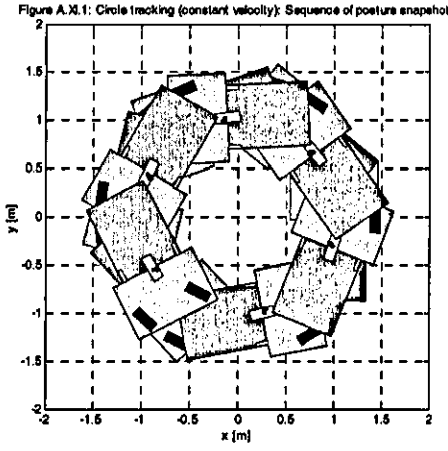


Figure A.XI.2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

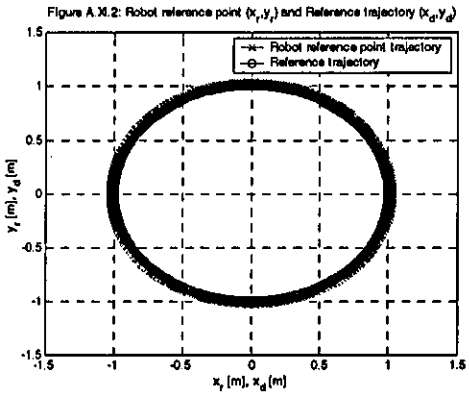


Figure A.XI.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

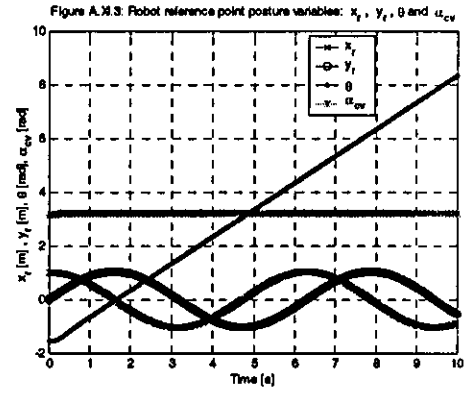


Figure A.XI.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

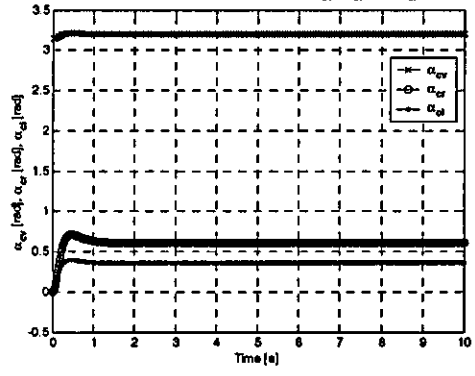


Figure A.XI.5: Robot reference point tracking error: e_{xr} and e_{yr}

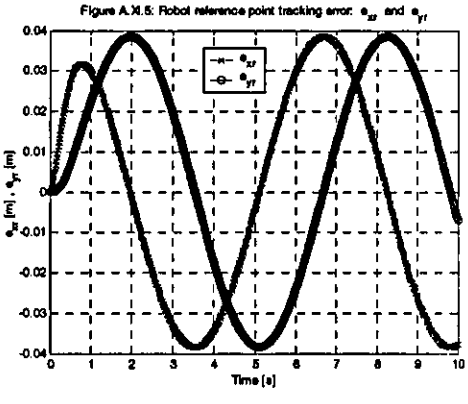


Figure A.XI.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

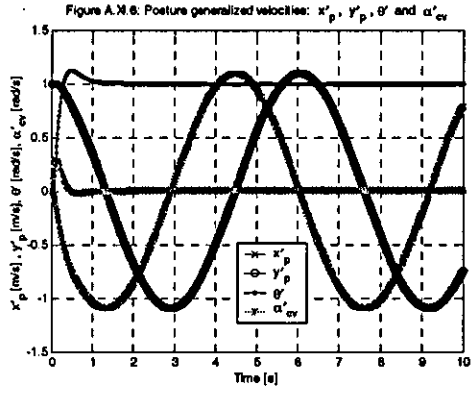


Figure A.XI.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

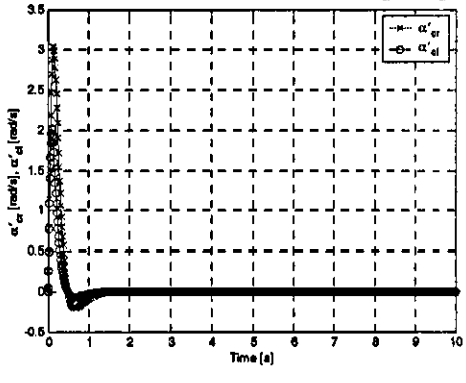


Figure A.XI.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

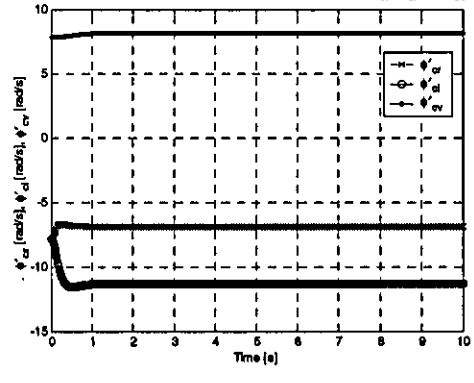


Figure A.XI.9: Steering system velocities: η and ς

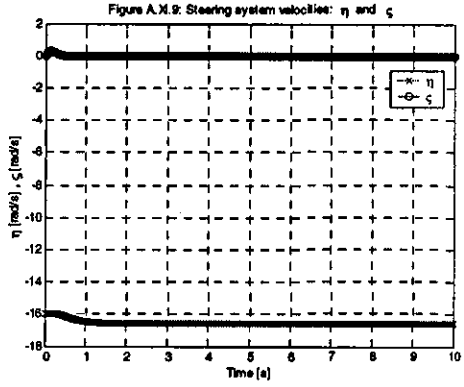
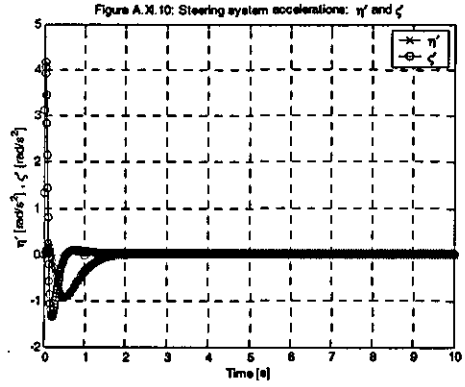


Figure A.XI.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.XI.11: Steering system torques: T_η and T_ς

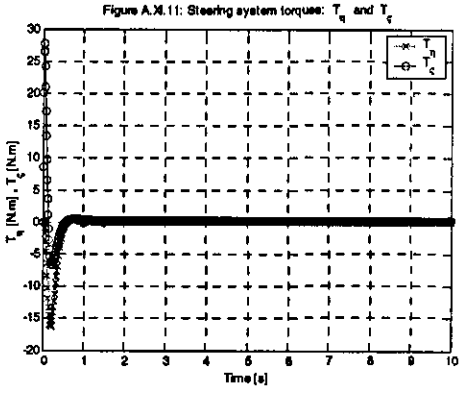


Figure A.XI.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

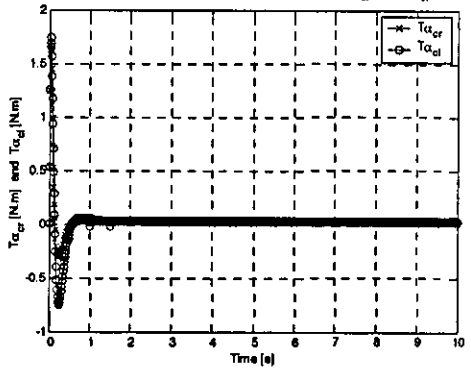


Figure A.XI.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

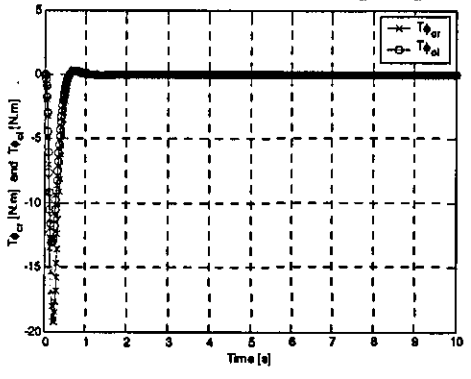


Figure A.XI.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

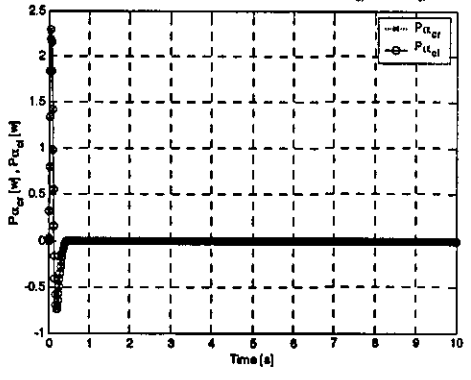
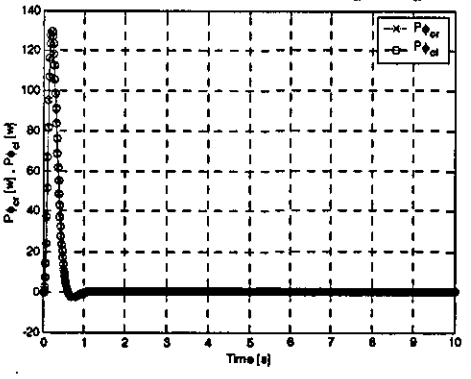


Figure A.XI.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XI.16: Steering system torques: T_{η} and T_{ζ}

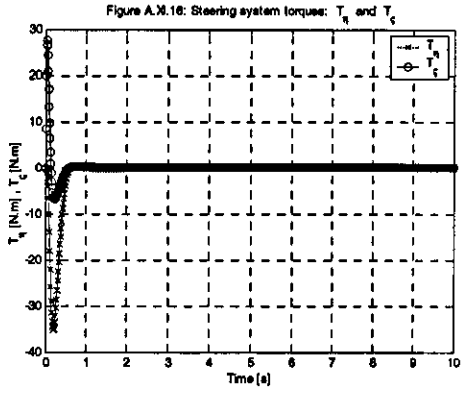


Figure A.XI.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

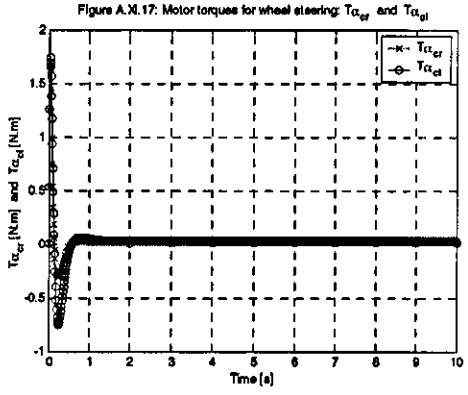


Figure A.XI.18: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

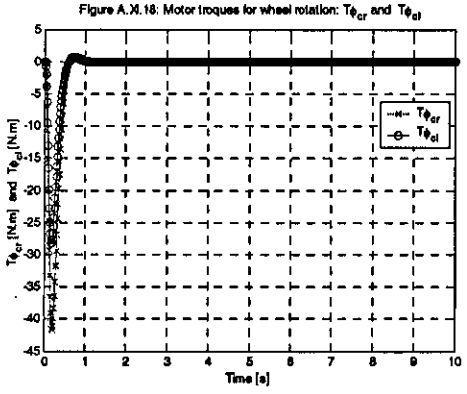


Figure A.XI.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

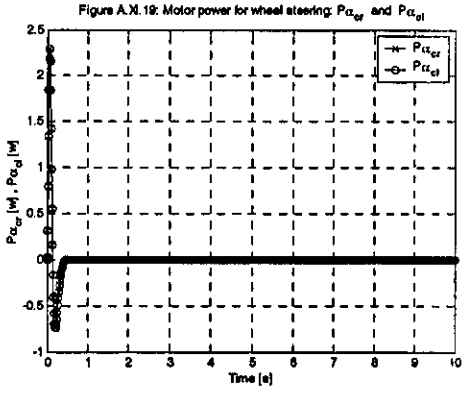
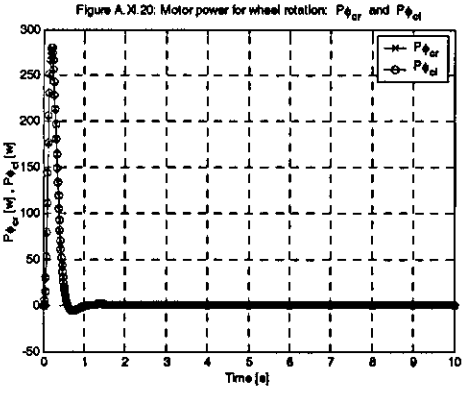


Figure A.XI.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XII ROBOT TYPE (1,1): Circle-line tracking. Constant velocity. Moving backwards. Load: 50 and 550

Simulation results

Figure A.XII.1: Circle-line tracking (constant velocity): Sequence of posture snapshots

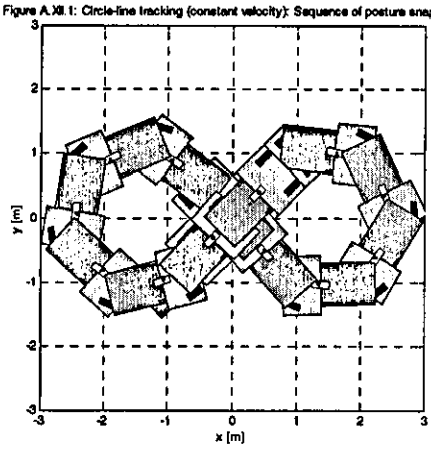


Figure A.XII.2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

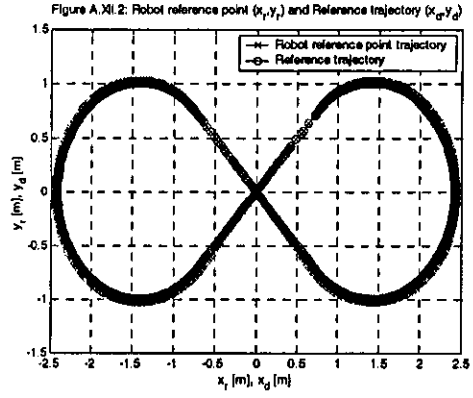


Figure A.XII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

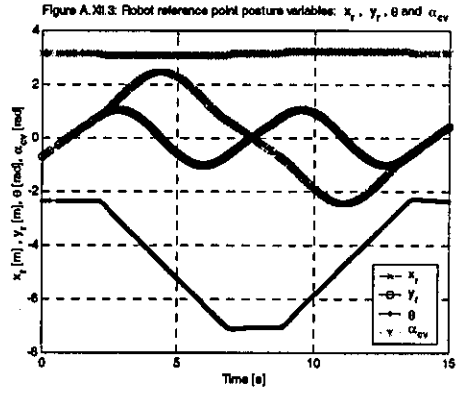


Figure A.XII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

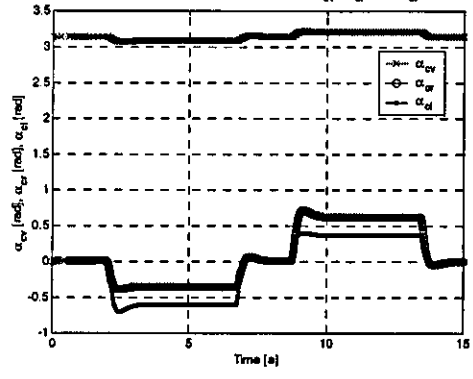


Figure A.XII.5: Robot reference point tracking error: e_{xr} and e_{yr}

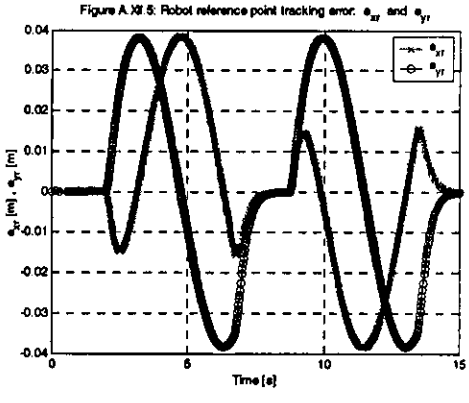


Figure A.XII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

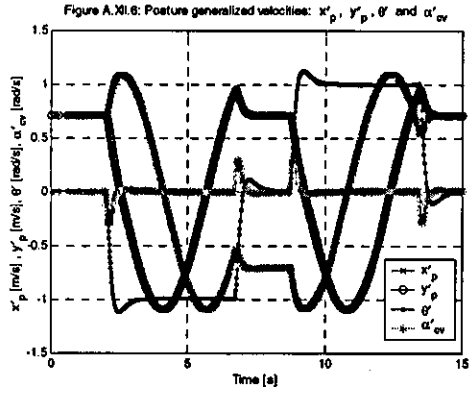


Figure A.XII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

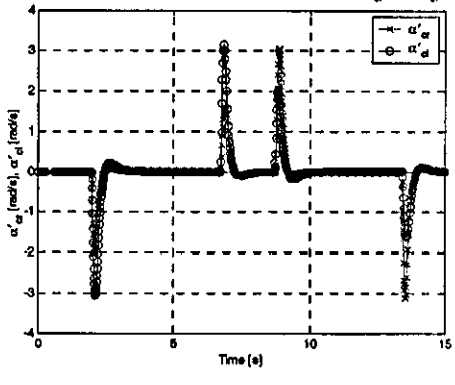


Figure A.XII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

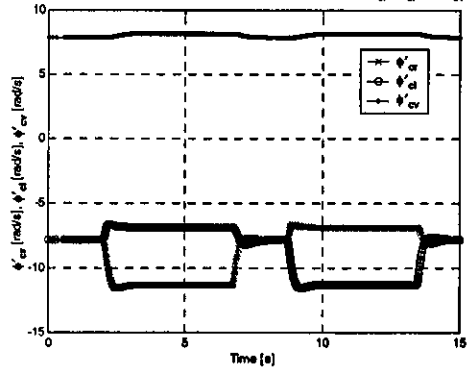


Figure A.XI.9: Steering system velocities: η and ζ

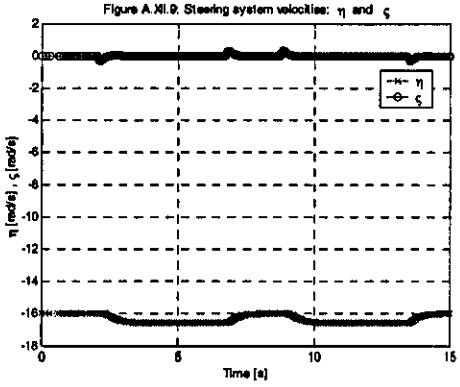
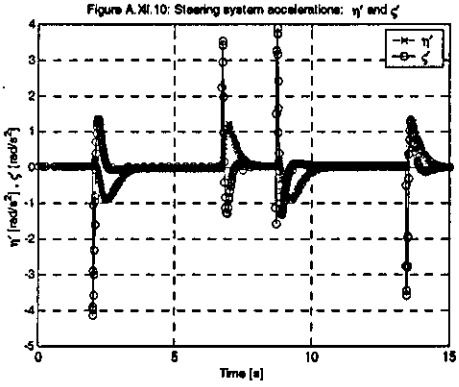


Figure A.XI.10: Steering system accelerations: η' and ζ'



Load = 50kg

Figure A.XI.11: Steering system torques: T_η and T_ζ

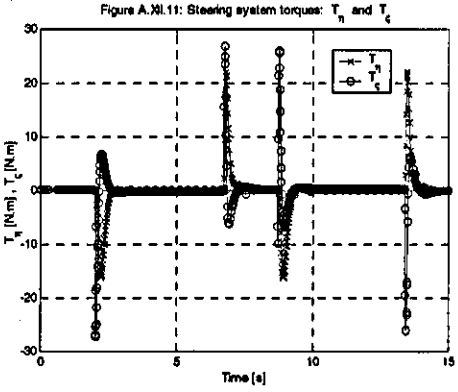


Figure A.XI.12: Motor torques for wheel steering: $T\alpha_{or}$ and $T\alpha_{ol}$

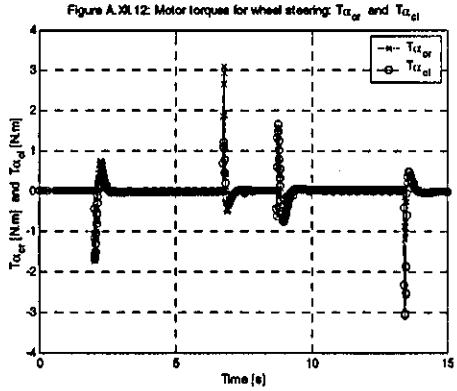


Figure A.XI.13: Motor troques for wheel rotation: $T\phi_{or}$ and $T\phi_{ol}$

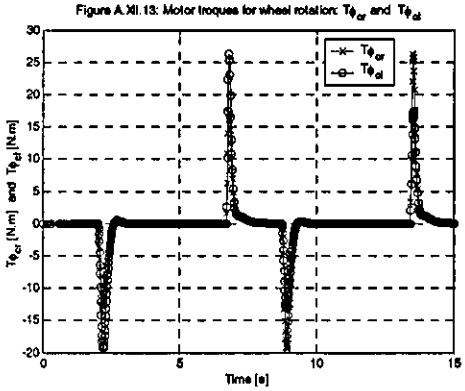


Figure A.XI.14: Motor power for wheel steering: $P\alpha_{or}$ and $P\alpha_{ol}$

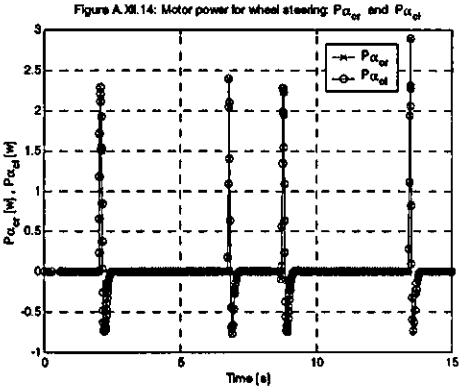
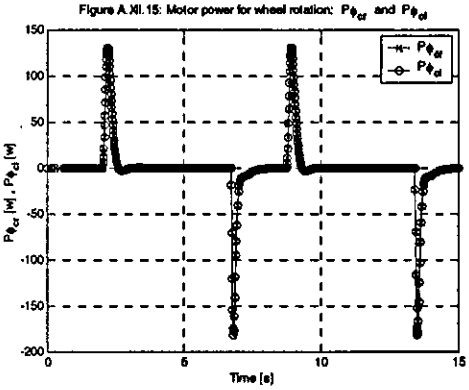


Figure A.XI.15: Motor power for wheel rotation: $P\phi_{or}$ and $P\phi_{ol}$



Load = 550kg

Figure A.XII.16: Steering system torques: T_s and T_c

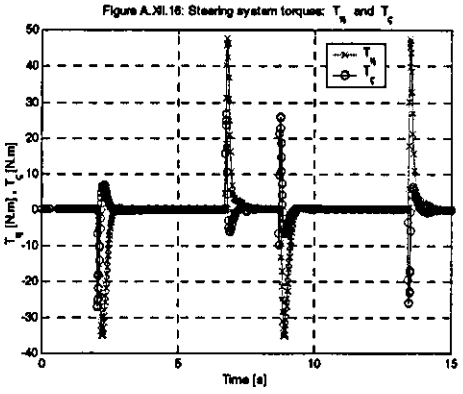


Figure A.XII.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

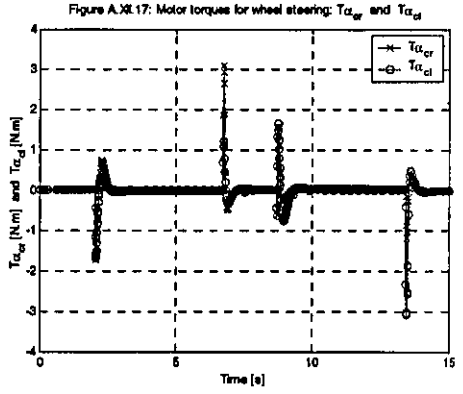


Figure A.XII.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

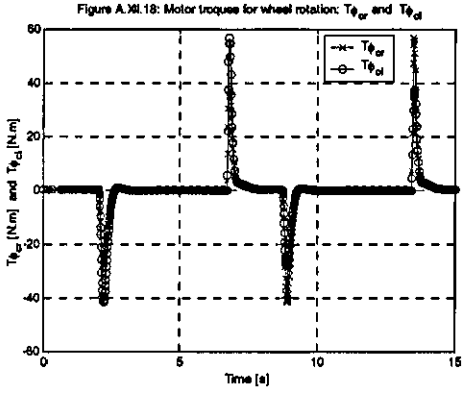


Figure A.XII.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

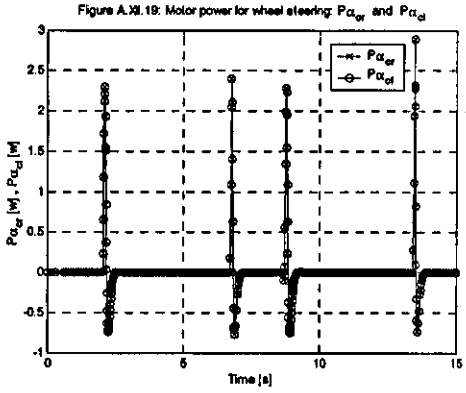
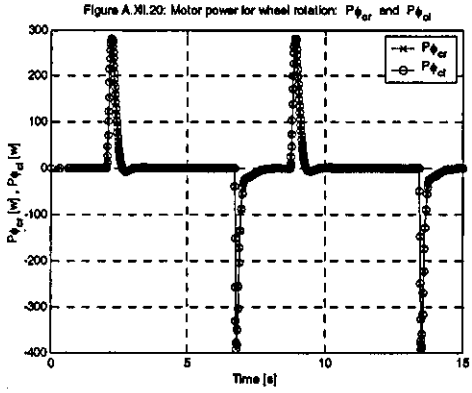


Figure A.XII.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XIII ROBOT TYPE (1,1): Point stabilization. Moving forwards. Load: 50 and 550

Simulation results

Figure A.XII.1: Point stabilization: Sequence of posture snapshots

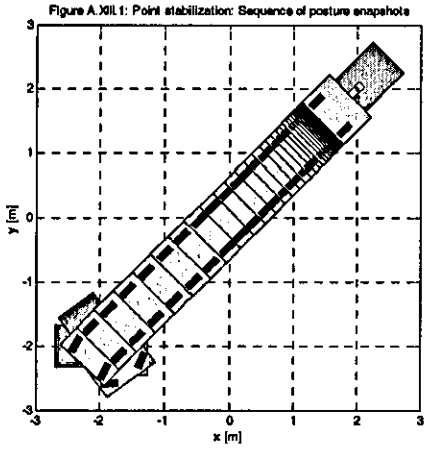


Figure A.XII.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

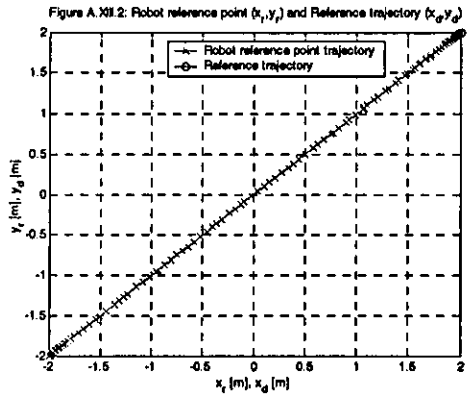


Figure A.XII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

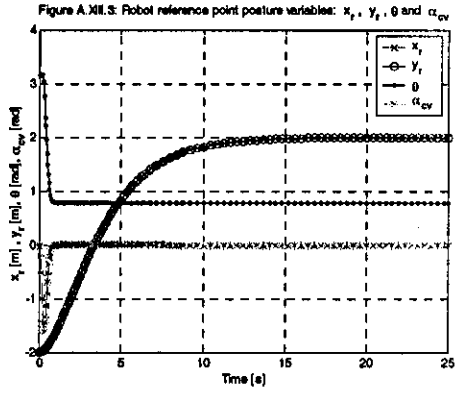


Figure A.XII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

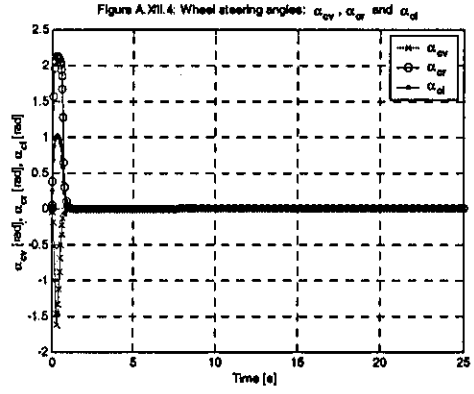


Figure A.XII.5: Robot reference point tracking error: e_{xr} and e_{yr}

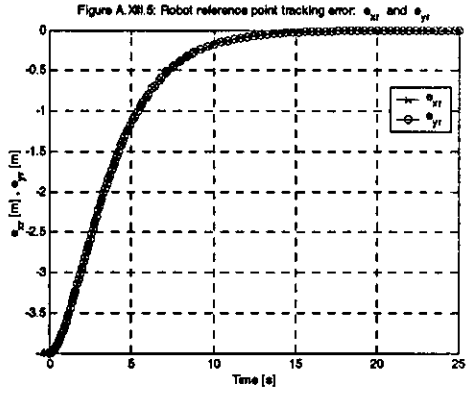


Figure A.XII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

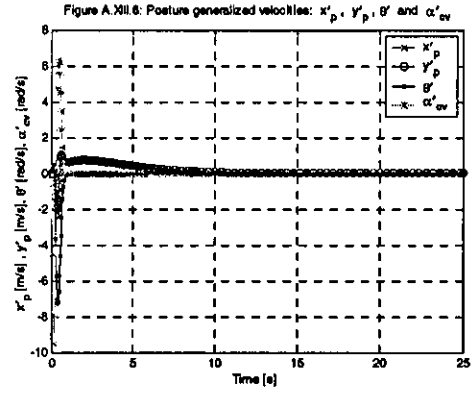


Figure A.XII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

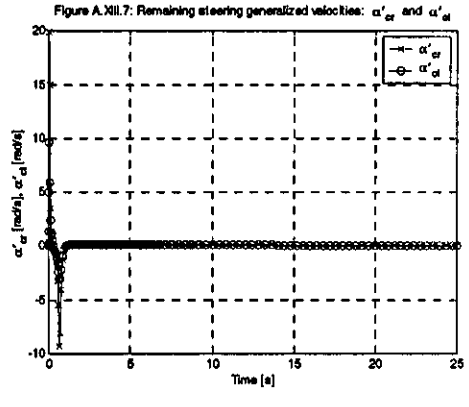


Figure A.XII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

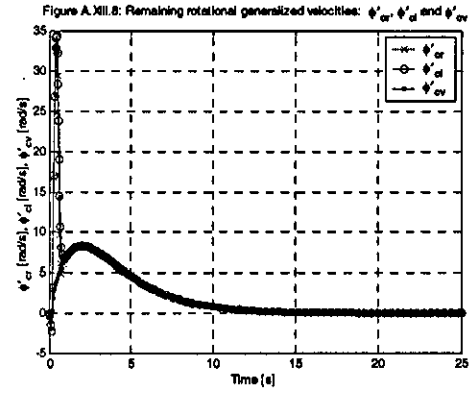


Figure A.XIII.9: Steering system velocities: η and ς

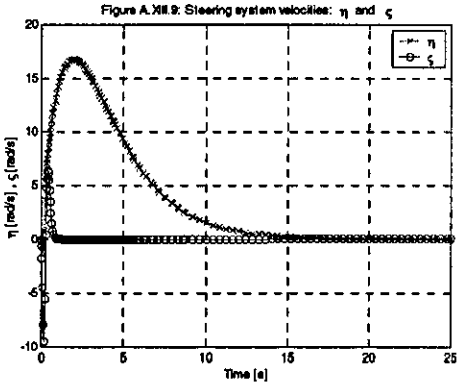
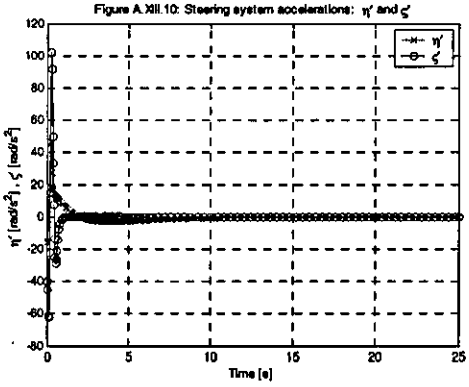


Figure A.XIII.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.XIII.11: Steering system torques: T_η and T_ς

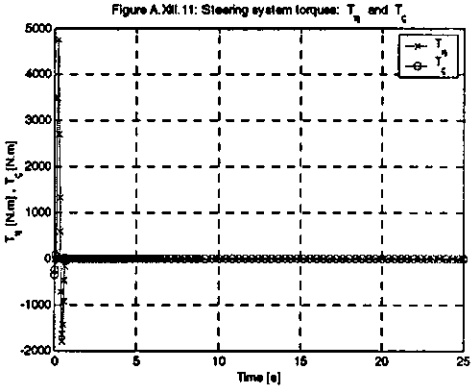


Figure A.XIII.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

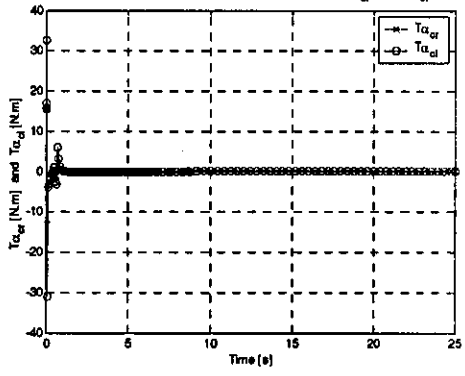


Figure A.XIII.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

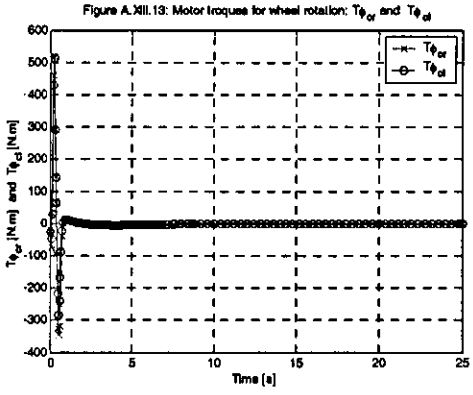


Figure A.XIII.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

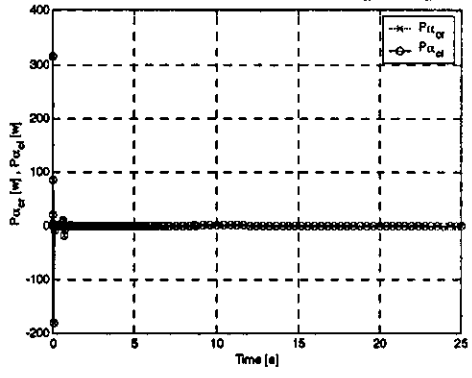
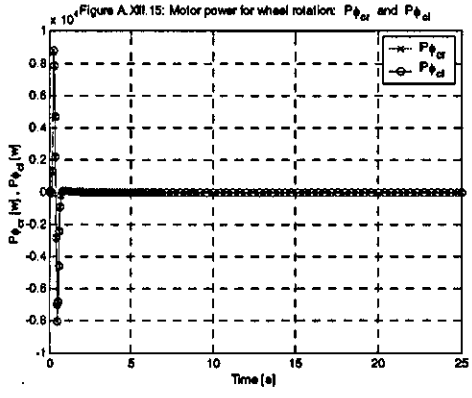


Figure A.XIII.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XII.16: Steering system torques: T_{η} and T_{c}

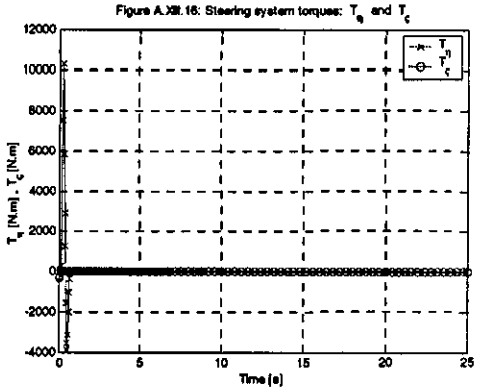


Figure A.XII.17: Motor torques for wheel steering: $T\alpha_{or}$ and $T\alpha_{ol}$

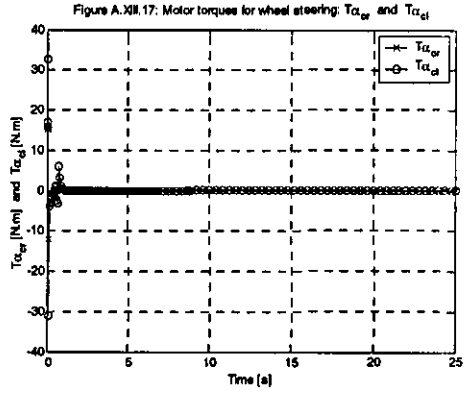


Figure A.XII.18: Motor troques for wheel rotation: $T\phi_{or}$ and $T\phi_{ol}$

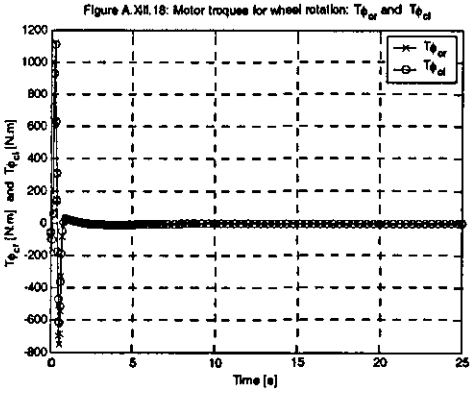


Figure A.XII.19: Motor power for wheel steering: $P\alpha_{or}$ and $P\alpha_{ol}$

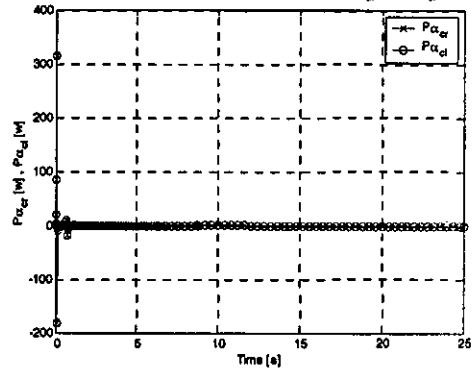
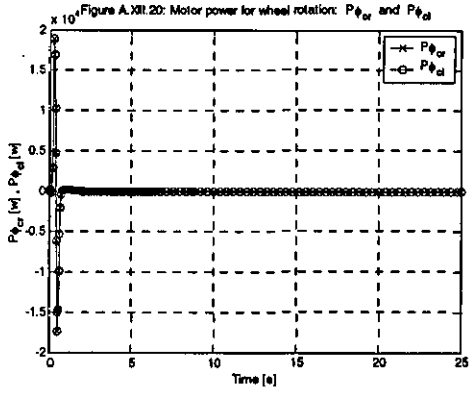


Figure A.XII.20: Motor power for wheel rotation: $P\phi_{or}$ and $P\phi_{ol}$



APPENDIX XIV ROBOT TYPE (1,1): Line tracking. Constant acceleration. Moving forwards. Load: 50 and 550

Simulation results

Figure A.XIV.1: Line tracking (constant acceleration): Sequence of posture snapshots

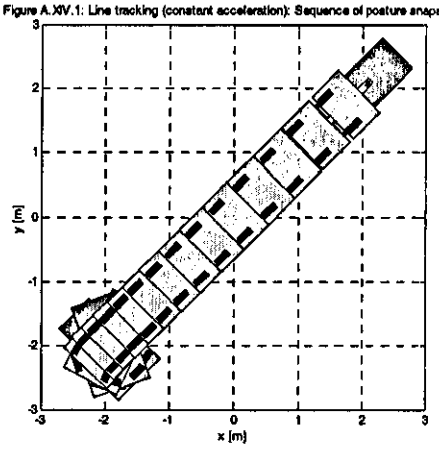


Figure A.XIV.2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

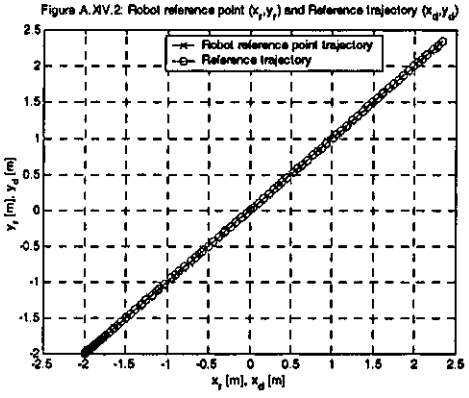


Figure A.XIV.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

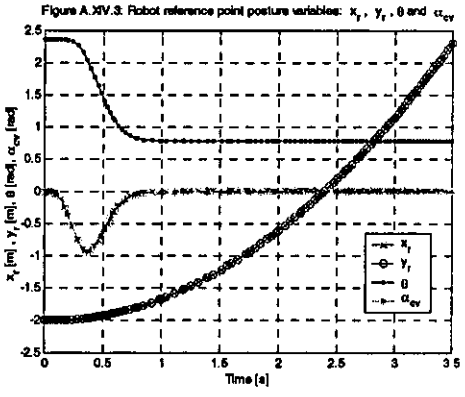


Figure A.XIV.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

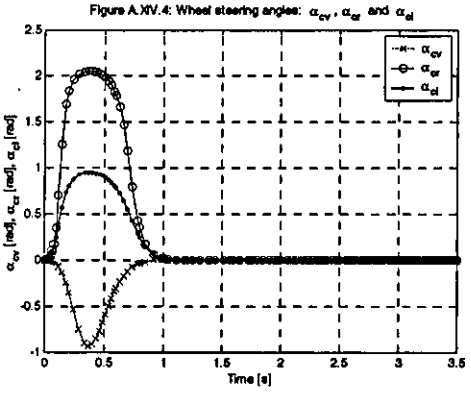


Figure A.XIV.5: Robot reference point tracking error: e_{xr} and e_{yr}

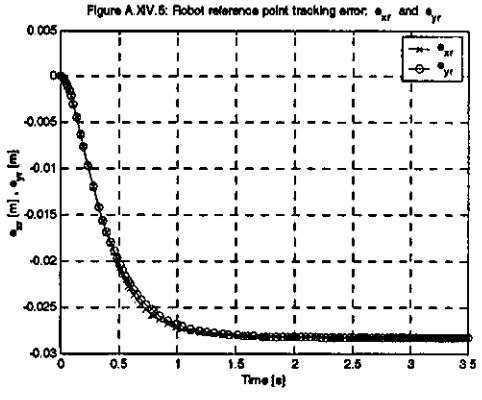


Figure A.XIV.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

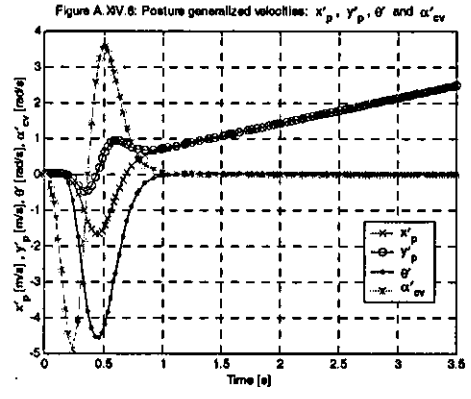


Figure A.XIV.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

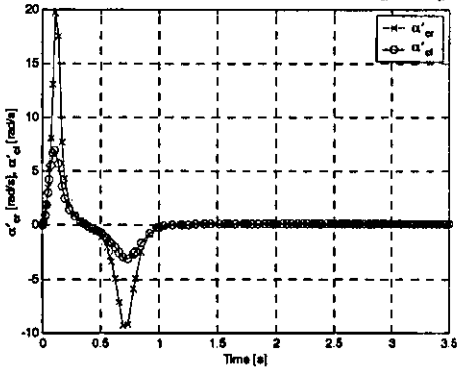


Figure A.XIV.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

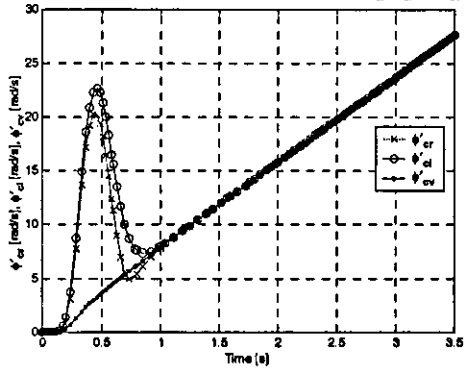


Figure A.XIV.9: Steering system velocities: η and ς

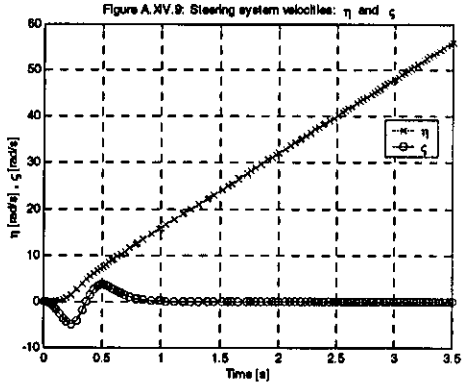
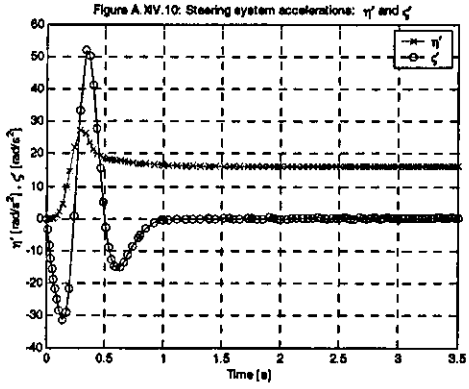


Figure A.XIV.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.XIV.11: Steering system torques: T_η and T_ς

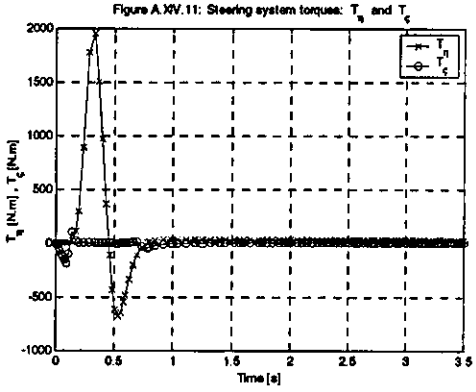


Figure A.XIV.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

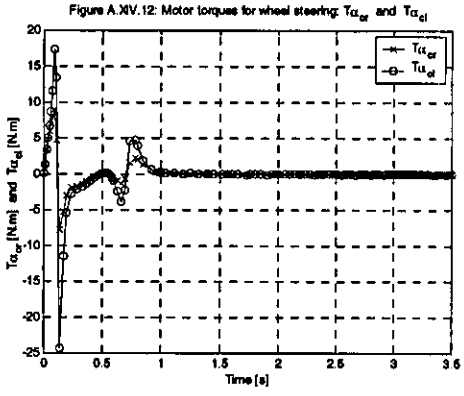


Figure A.XIV.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

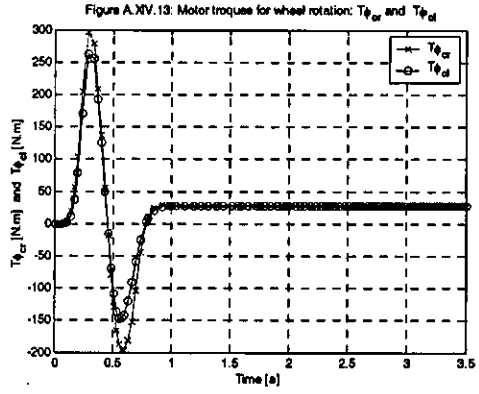


Figure A.XIV.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

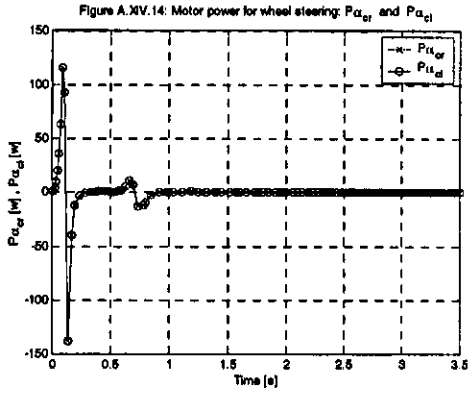
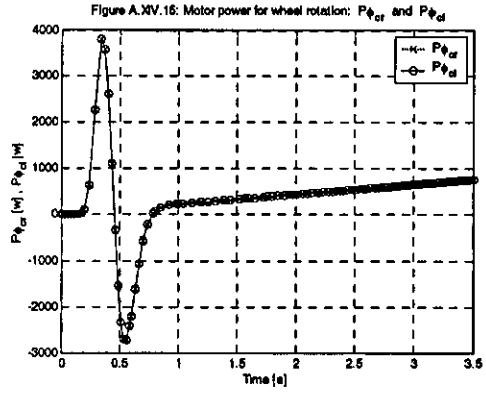


Figure A.XIV.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XIV.16: Steering system torques: T_s and T_c

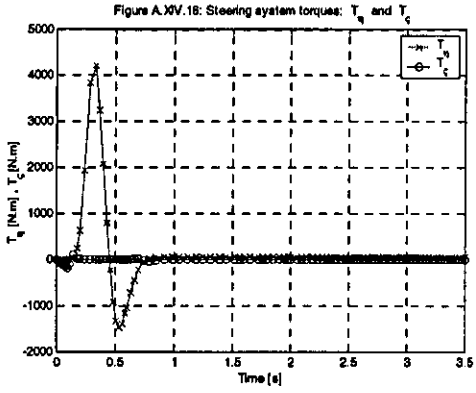


Figure A.XIV.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

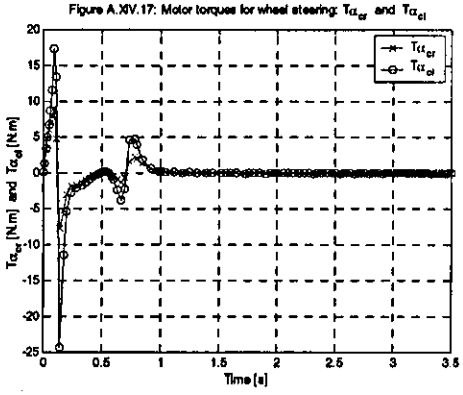


Figure A.XIV.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

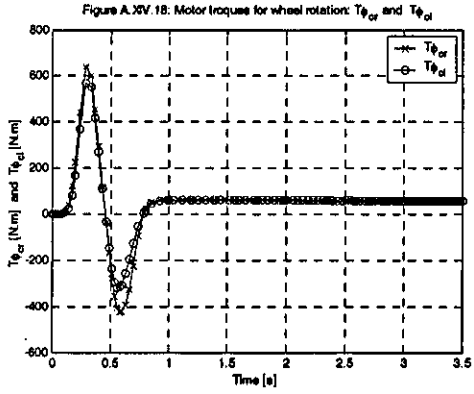


Figure A.XIV.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

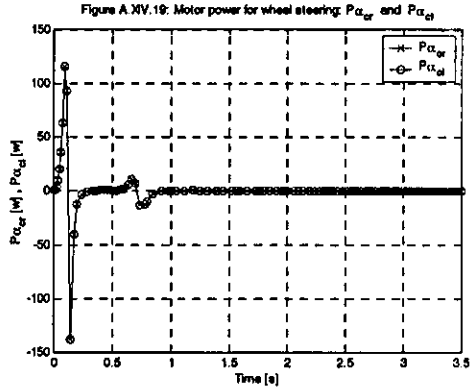
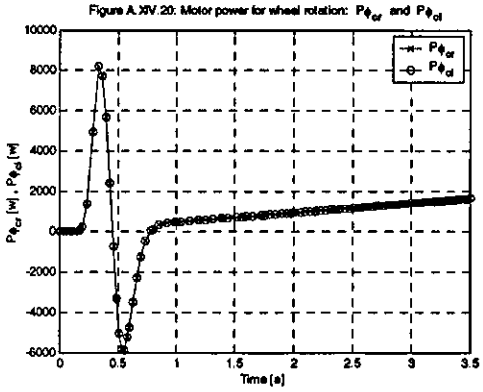


Figure A.XIV.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XV ROBOT TYPE (1,1): Line tracking. Constant deceleration. Moving forwards. Load: 50 and 550

Simulation results

Figure A.XV.1: Line tracking (constant deceleration): Sequence of posture snapshots

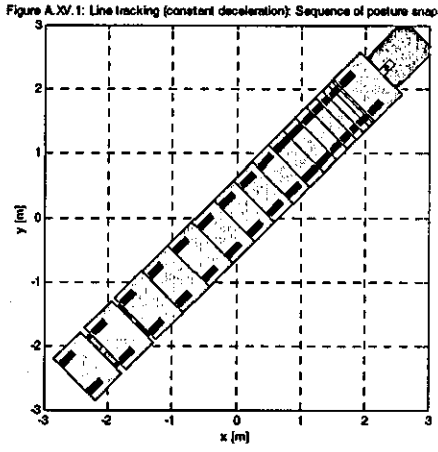


Figure A.XV.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

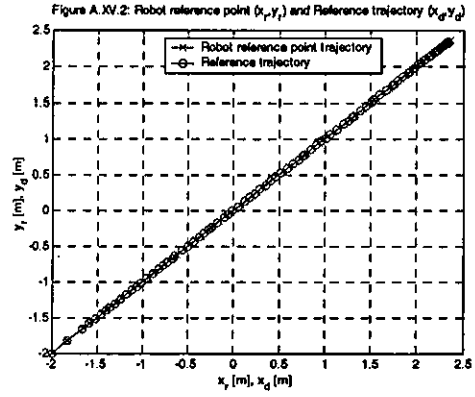


Figure A.XV.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

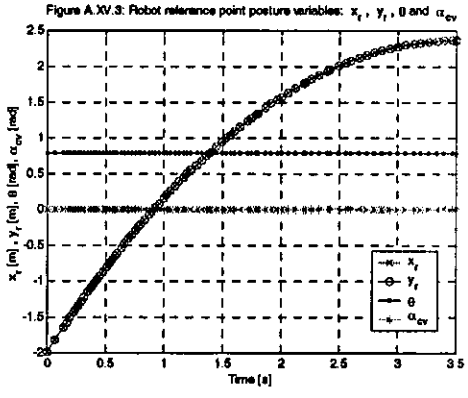


Figure A.XV.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

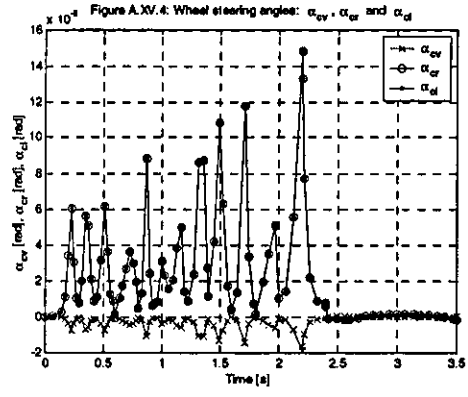


Figure A.XV.5: Robot reference point tracking error: e_{xr} and e_{yr}

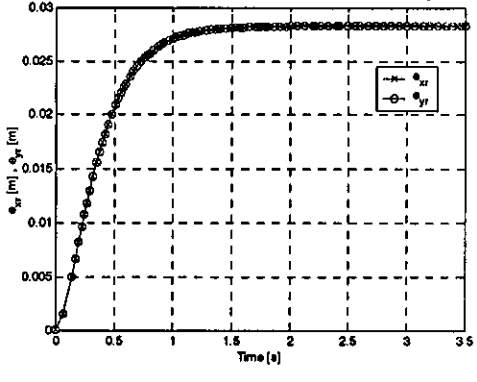


Figure A.XV.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

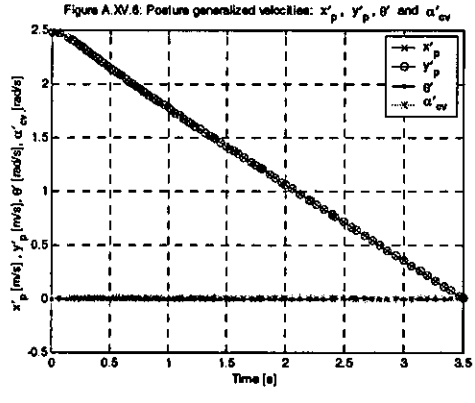


Figure A.XV.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

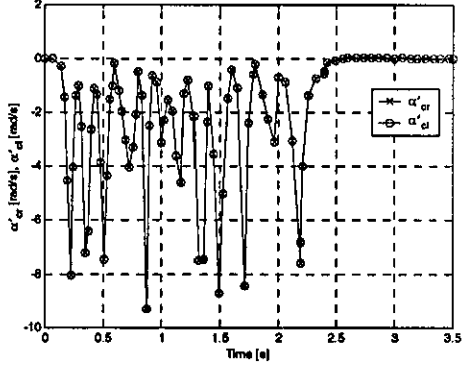
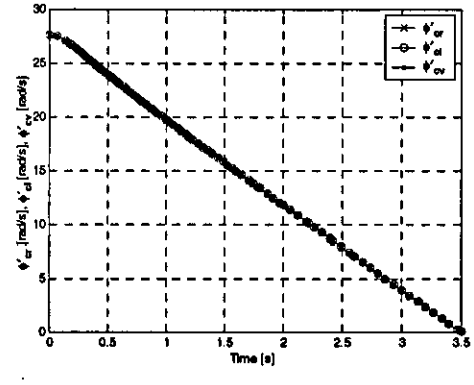


Figure A.XV.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}



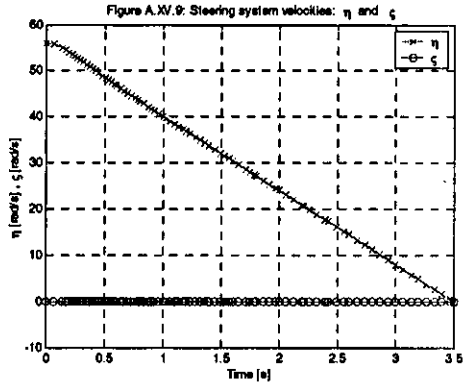


Figure A.XV.9: Steering system velocities: η and ζ

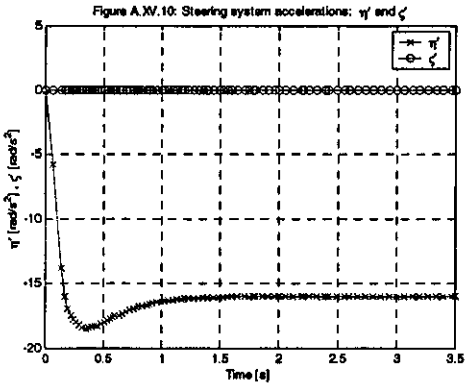


Figure A.XV.10: Steering system accelerations: η' and ζ'

Load = 50kg

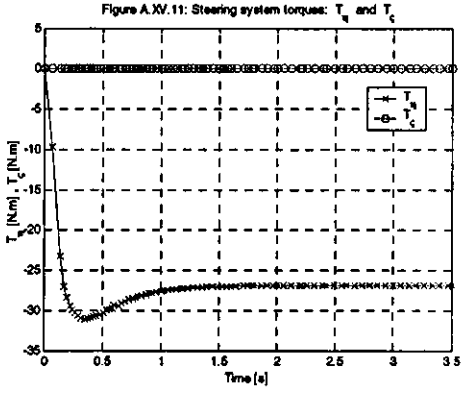


Figure A.XV.11: Steering system torques: T_η and T_ζ

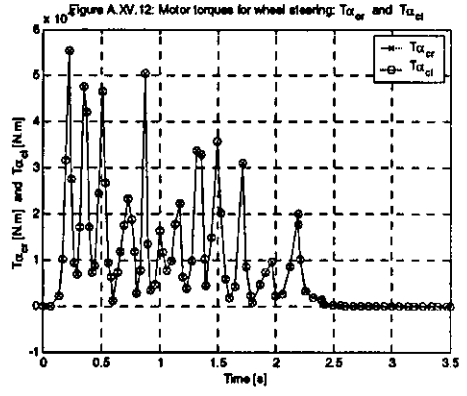


Figure A.XV.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

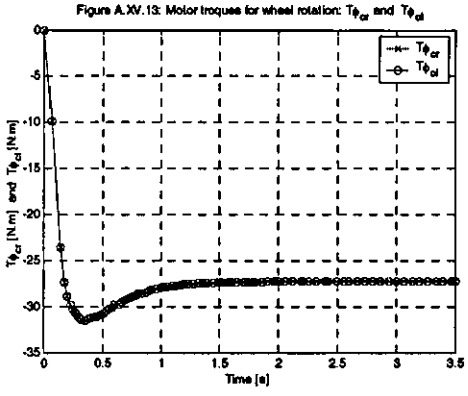


Figure A.XV.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

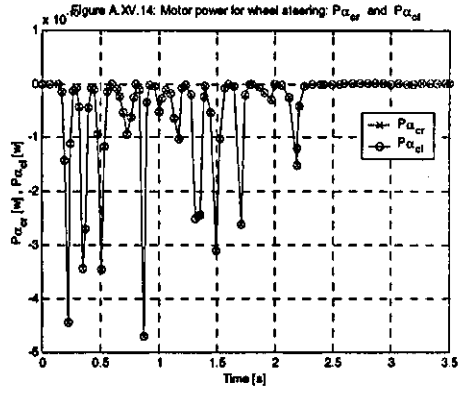


Figure A.XV.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

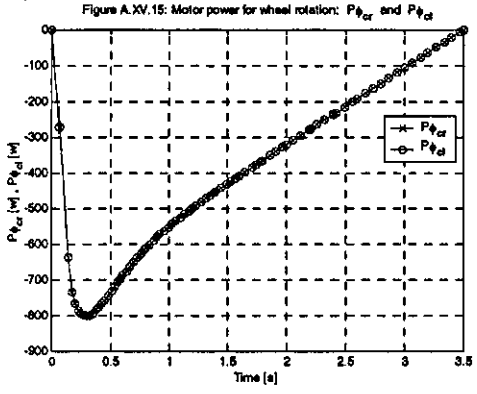


Figure A.XV.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$

Load = 550kg

Figure A.XV.16: Steering system torques: T_q and T_c

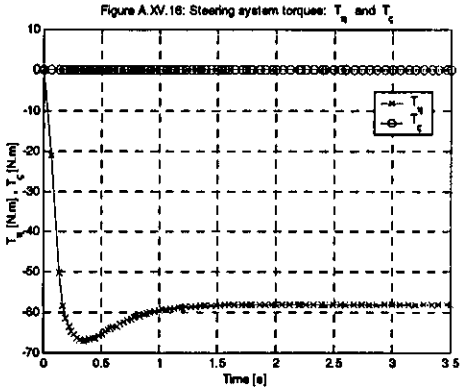


Figure A.XV.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

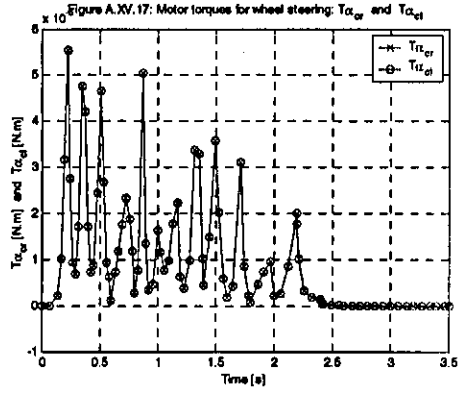


Figure A.XV.18: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

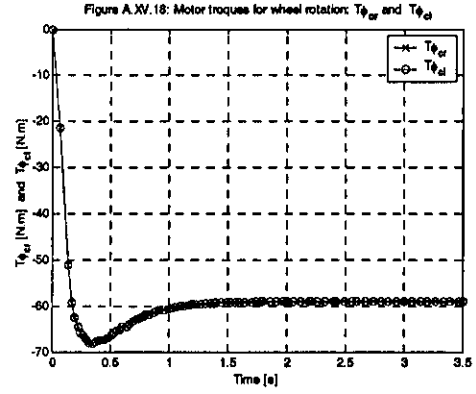


Figure A.XV.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

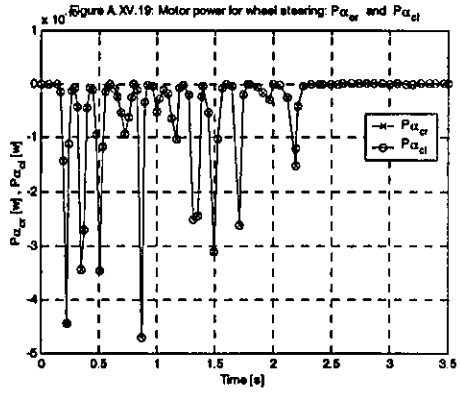
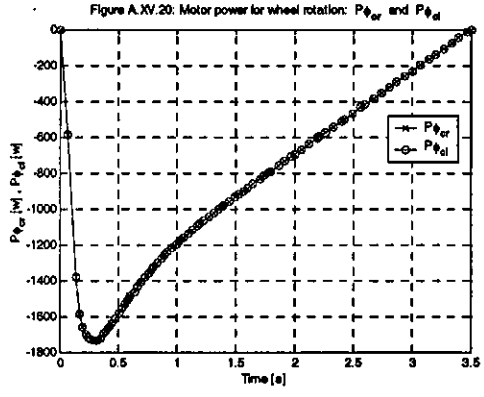


Figure A.XV.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XVI ROBOT TYPE (1,1): Line tracking. Constant velocity. Moving forwards. Load: 50 and 550

Simulation results

Figure A.XVI.1: Line tracking (constant velocity). Sequence of posture snapshots

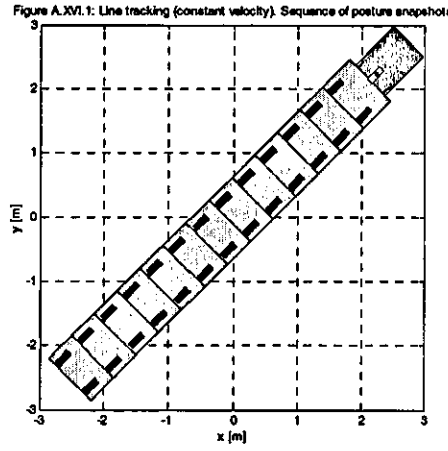


Figure A.XVI.2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

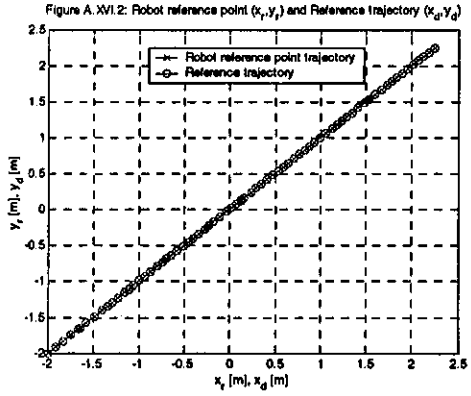


Figure A.XVI.3: Robot reference point posture variables: x_r , y_r , θ and α_{cv}

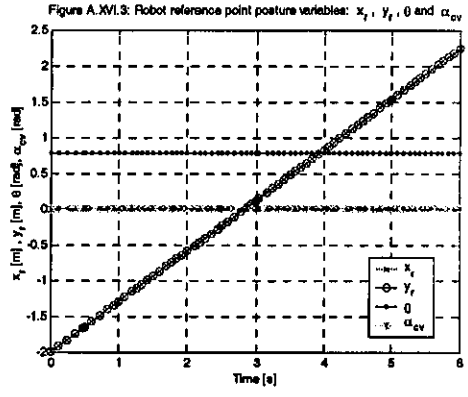


Figure A.XVI.4: Wheel steering angles: α_{cv} , α_{cr} and α_{cl}

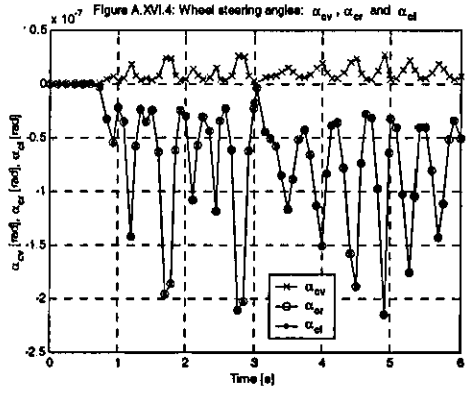


Figure A.XVI.5: Robot reference point tracking error: e_{xr} and e_{yr}

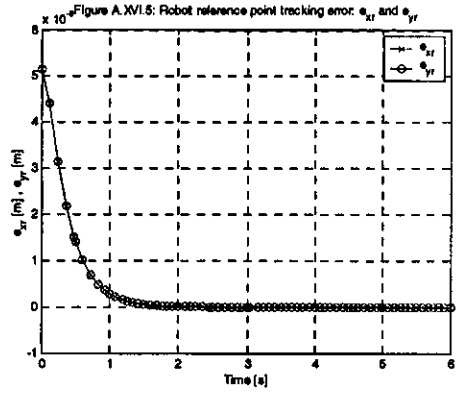


Figure A.XVI.6: Posture generalized velocities: x'_p , y'_p , θ' and α'_{cv}

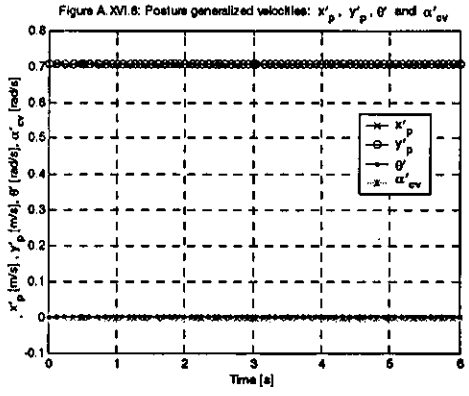


Figure A.XVI.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

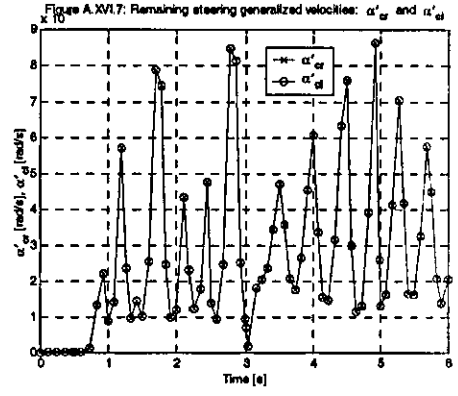
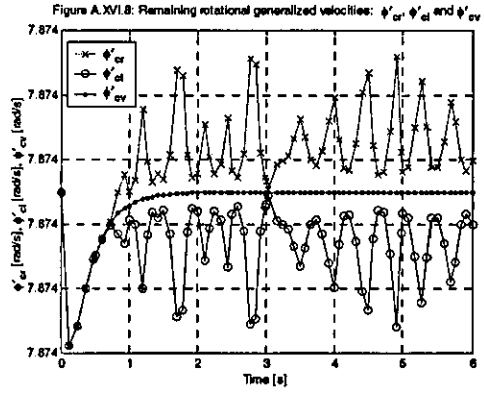


Figure A.XVI.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}



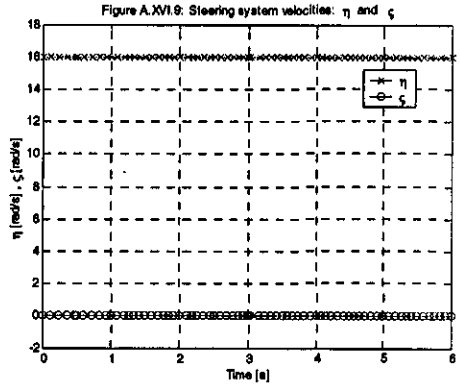


Figure A.XVI.9: Steering system velocities: η and ς
η [rad/s] , ς [rad/s]
Time [s]
η
ς

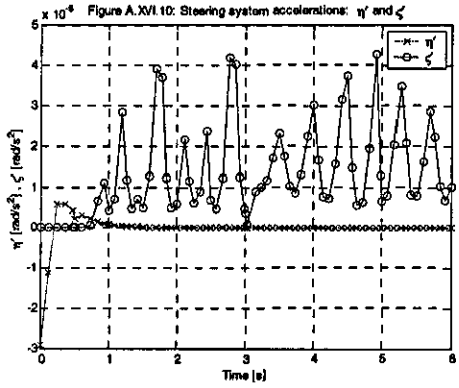


Figure A.XVI.10: Steering system accelerations: η' and ς'
η' [rad/s²] , ς' [rad/s²]
Time [s]
η'
ς'

Load = 50kg

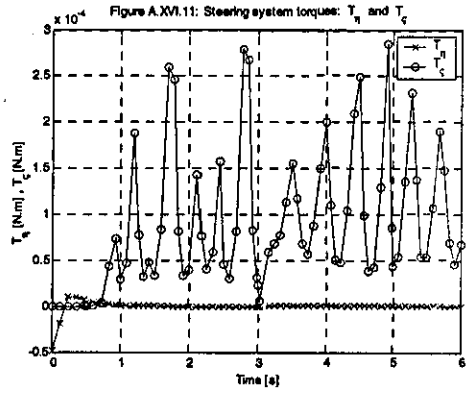


Figure A.XVI.11: Steering system torques: Tη and Tς
Tη [N.m] , Tς [N.m]
Time [s]
Tη
Tς

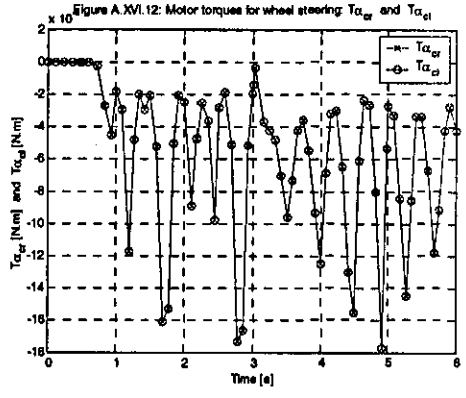


Figure A.XVI.12: Motor torques for wheel steering: Tαcr and Tαcl
Tαcr [N.m] and Tαcl [N.m]
Time [s]
Tαcr
Tαcl

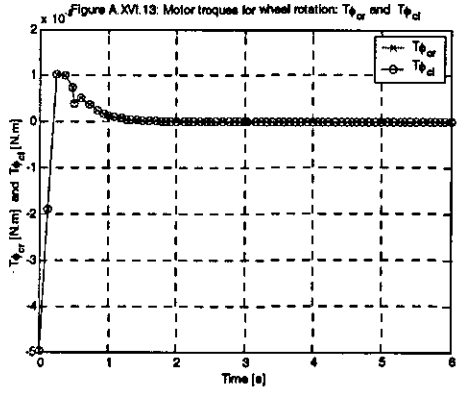


Figure A.XVI.13: Motor troques for wheel rotation: Tϕcr and Tϕcl
Tϕcr [N.m] and Tϕcl [N.m]
Time [s]
Tϕcr
Tϕcl

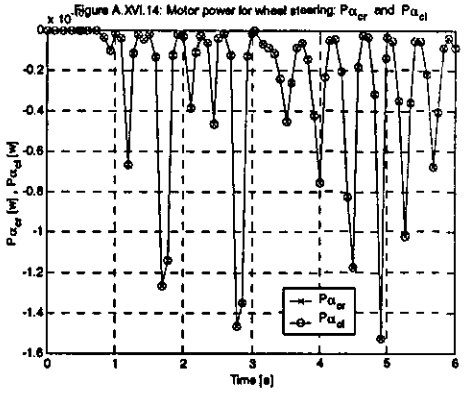


Figure A.XVI.14: Motor power for wheel steering: Pαcr and Pαcl
Pαcr [w] , Pαcl [w]
Time [s]
Pαcr
Pαcl

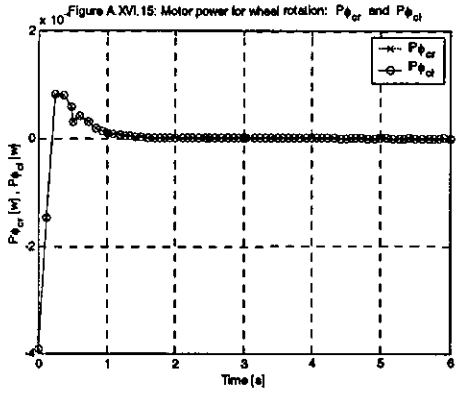


Figure A.XVI.15: Motor power for wheel rotation: Pϕcr and Pϕcl
Pϕcr [w] , Pϕcl [w]
Time [s]
Pϕcr
Pϕcl

Load = 550kg

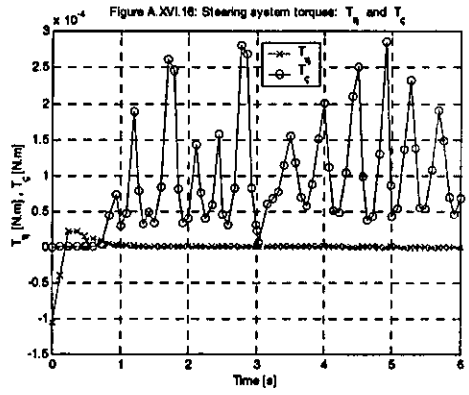


Figure A.XVI.16: Steering system torques: T_s and T_c
x 10^-4
T_s
T_c
T_s [N.m], T_c [N.m]
Time [s]

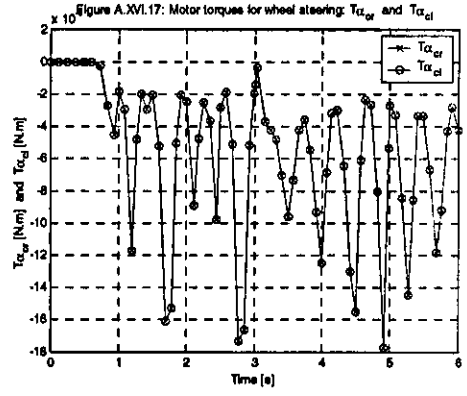


Figure A.XVI.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$
x 10^2
$T\alpha_{cr}$
$T\alpha_{cl}$
$T\alpha_{cr}$ [N.m] and $T\alpha_{cl}$ [N.m]
Time [s]

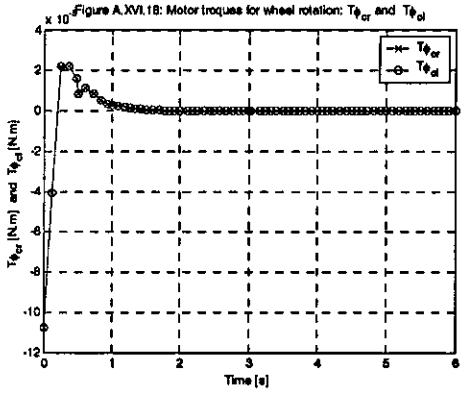


Figure A.XVI.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$
x 10^2
$T\phi_{cr}$
$T\phi_{cl}$
$T\phi_{cr}$ [N.m] and $T\phi_{cl}$ [N.m]
Time [s]

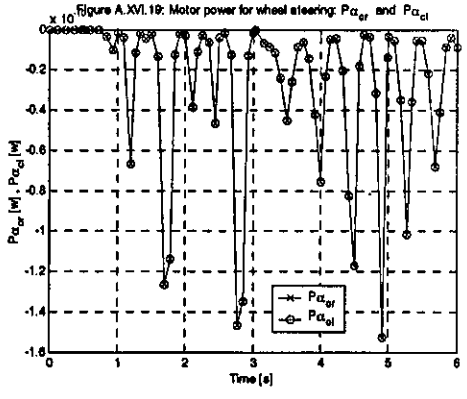


Figure A.XVI.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$
x 10^2
$P\alpha_{cr}$
$P\alpha_{cl}$
$P\alpha_{cr}$ [W], $P\alpha_{cl}$ [W]
Time [s]

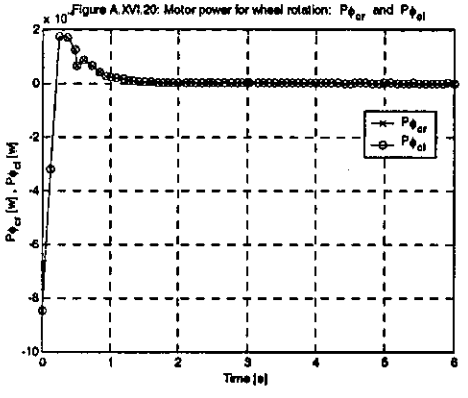


Figure A.XVI.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$
x 10^2
$P\phi_{cr}$
$P\phi_{cl}$
$P\phi_{cr}$ [W], $P\phi_{cl}$ [W]
Time [s]

APPENDIX XVII ROBOT TYPE (1,1): Circle tracking. Constant velocity. Moving forwards. Load: 50 and 550

Simulation results

Figure A.XVII.1: Robot circle tracking (constant velocity): Sequence of posture snapshots

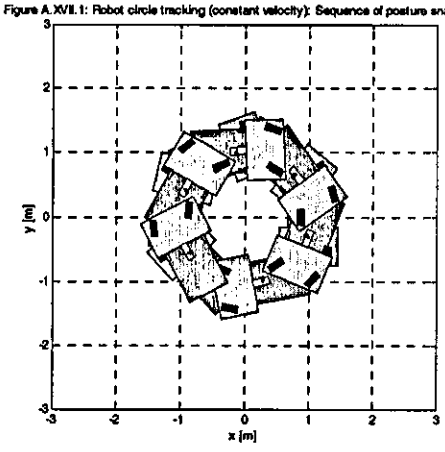


Figure A.XVII.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

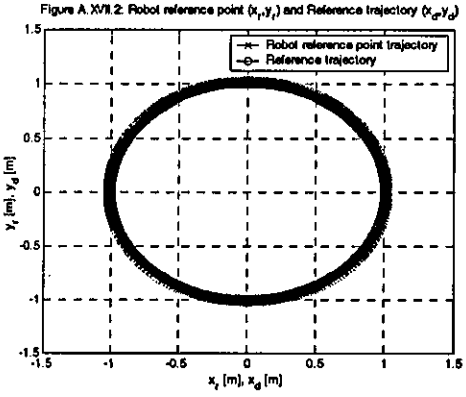


Figure A.XVII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

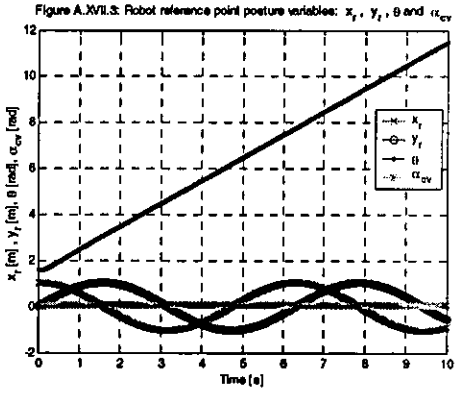


Figure A.XVII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

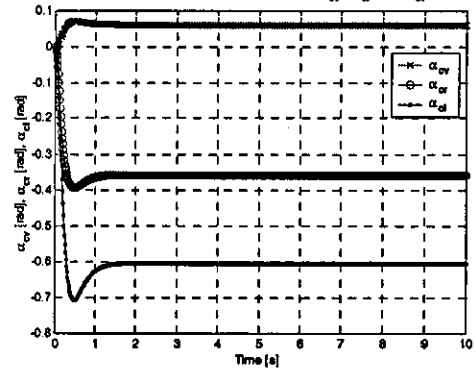


Figure A.XVII.5: Robot reference point tracking error: e_{xr} and e_{yr}

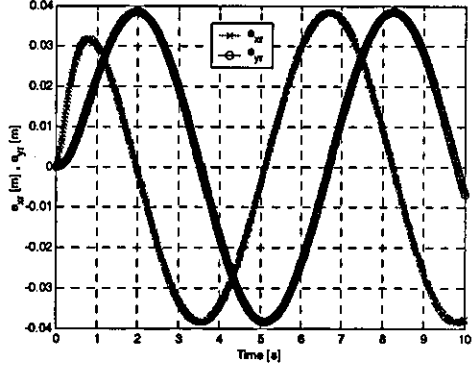


Figure A.XVII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

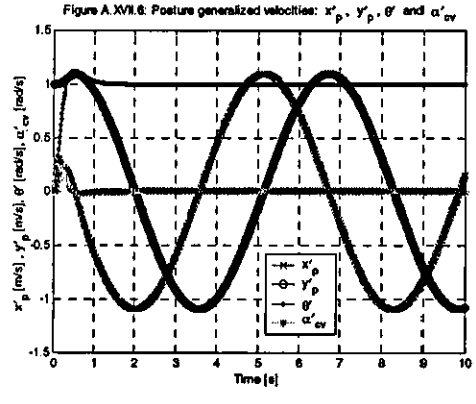


Figure A.XVII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

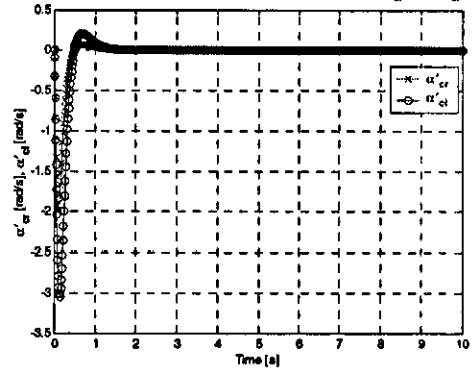


Figure A.XVII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

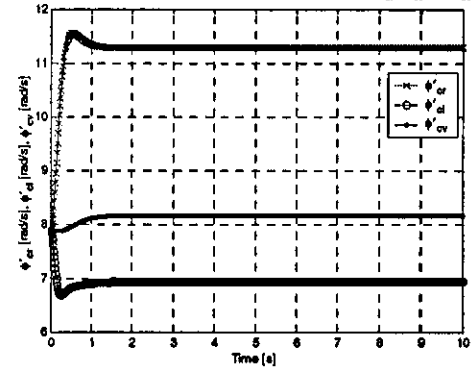


Figure A.XVII.9: Steering system velocities: η and ς

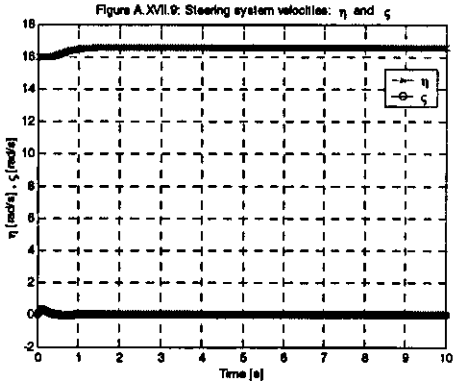
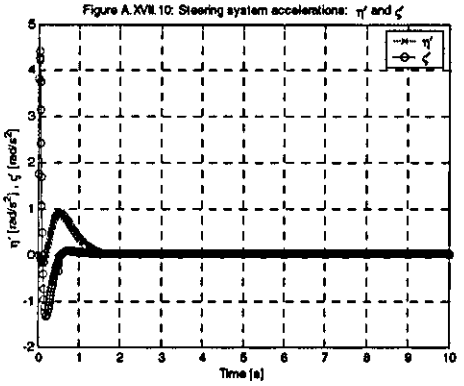


Figure A.XVII.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.XVII.11: Steering system torques: T_η and T_ς

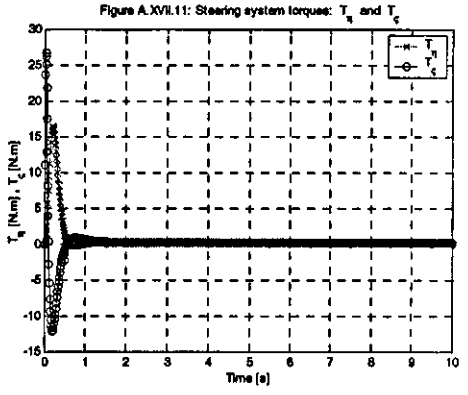


Figure A.XVII.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

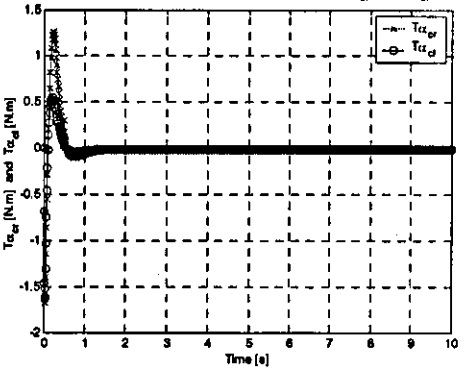


Figure A.XVII.13: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

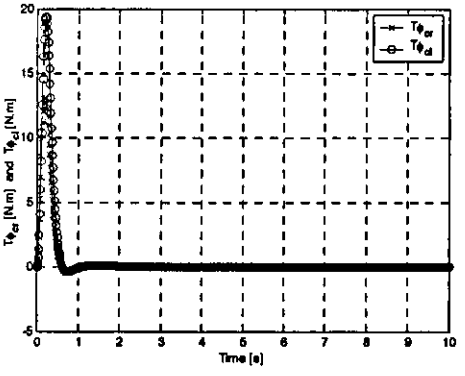


Figure A.XVII.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

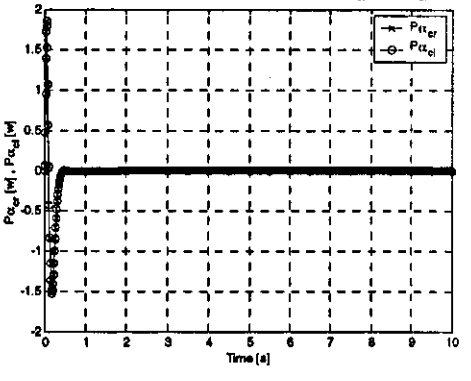
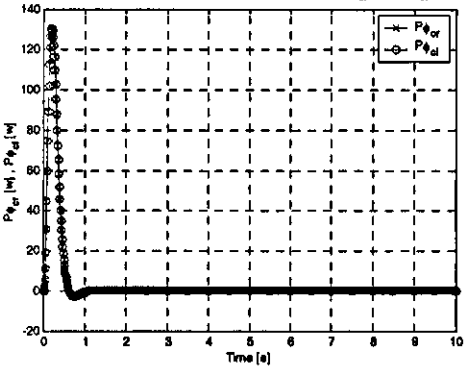


Figure A.XVII.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XVII.16: Steering system torques: T_η and T_ς

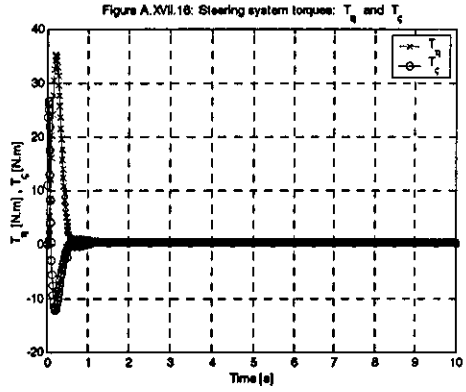


Figure A.XVII.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

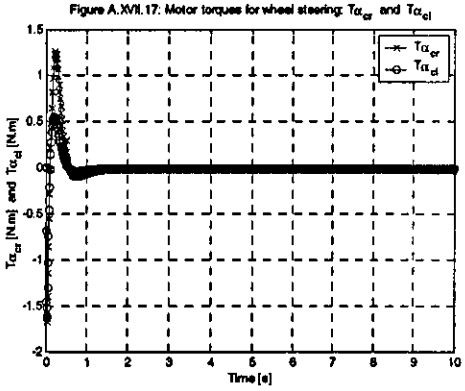


Figure A.XVII.18: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

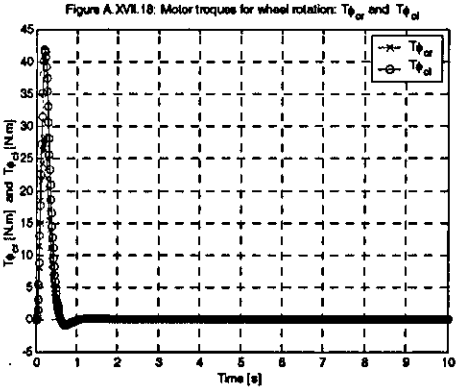


Figure A.XVII.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

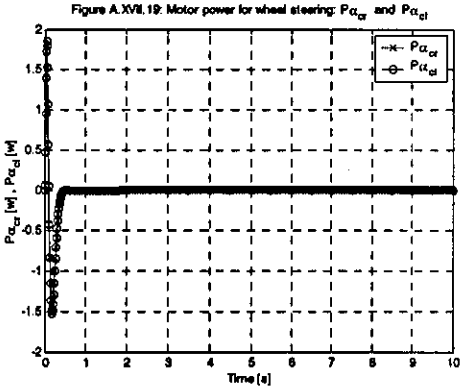
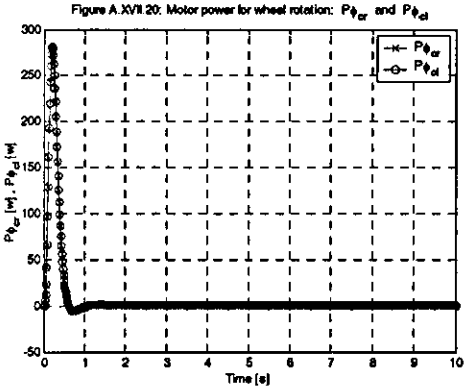


Figure A.XVII.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XVIII ROBOT TYPE (1,1): Circle-line tracking. Constant velocity. Moving forwards. Load: 50 and 550

Simulation results

Figure A.XVIII.1: Circle-line tracking (constant velocity): Sequence of posture snapshots

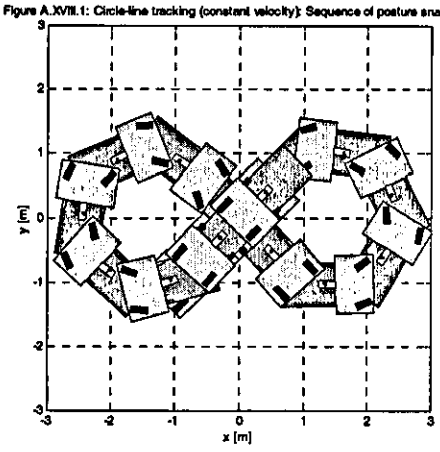


Figure A.XVII.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

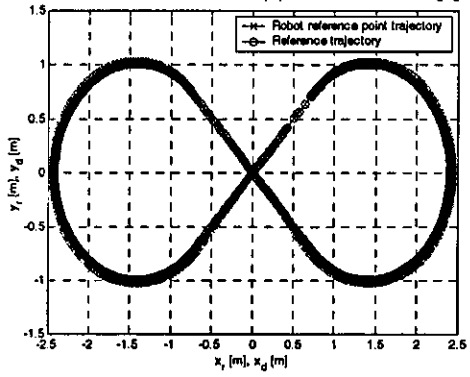


Figure A.XVIII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

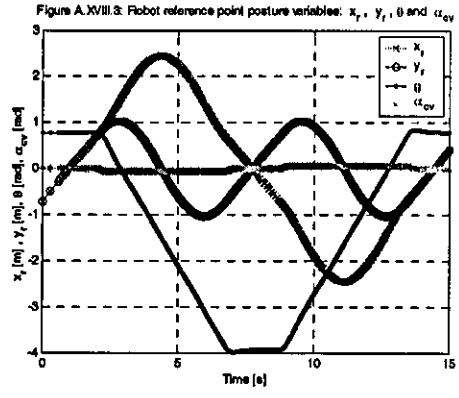


Figure A.XVIII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

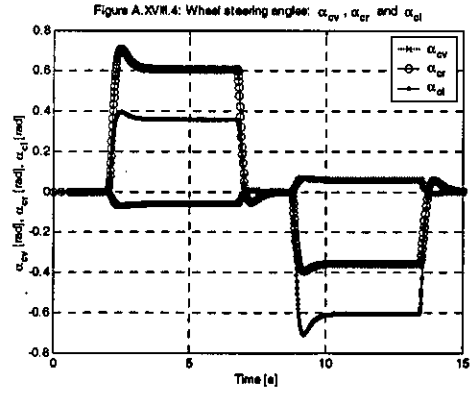


Figure A.XVIII.5: Robot reference point tracking error: e_{xr} and e_{yr}

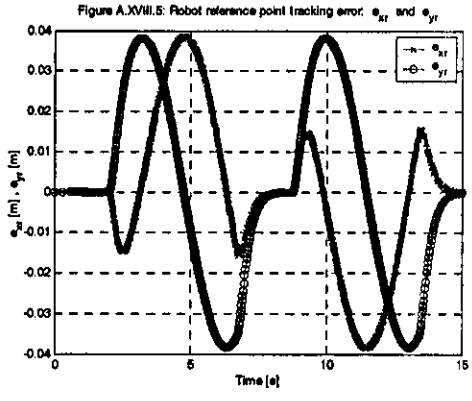


Figure A.XVIII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

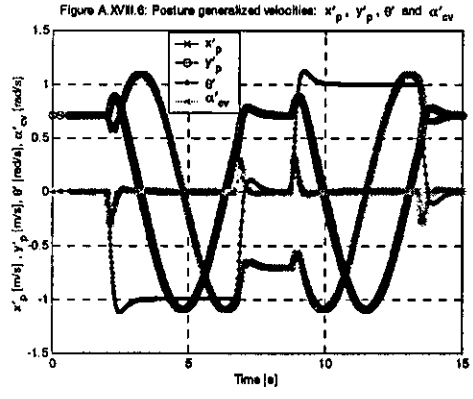


Figure A.XVIII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

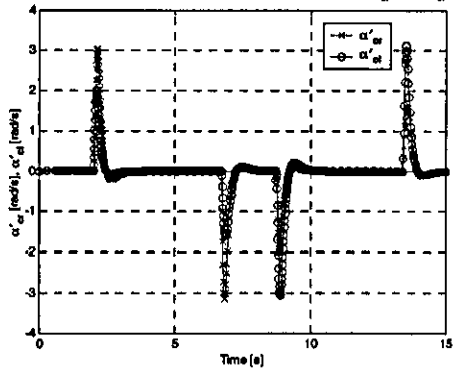


Figure A.XVIII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

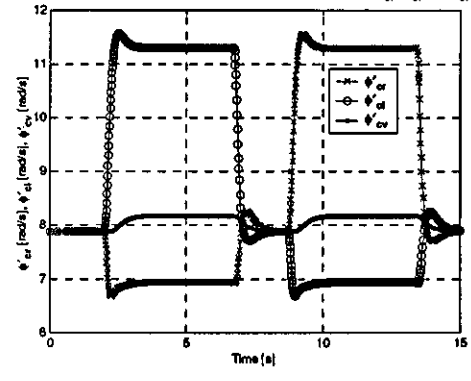


Figure A.XVIII.9: Steering system velocities: η and ς

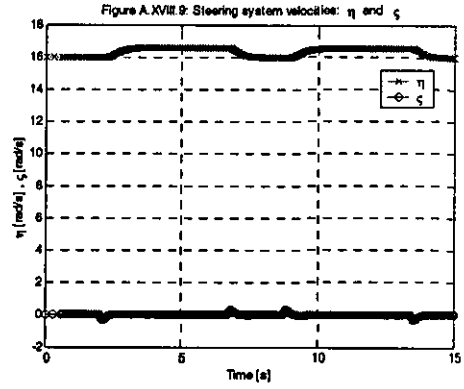
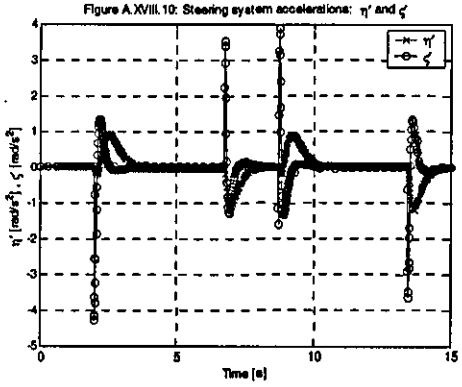


Figure A.XVIII.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.XVIII.11: Steering system torques: T_η and T_ς

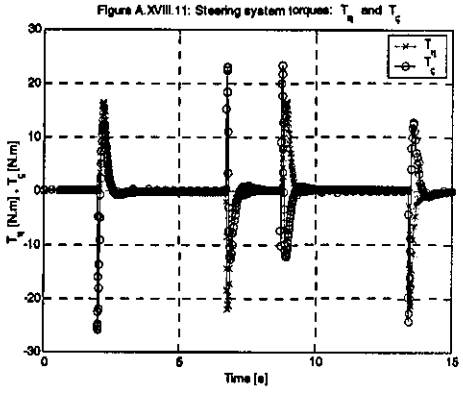


Figure A.XVIII.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

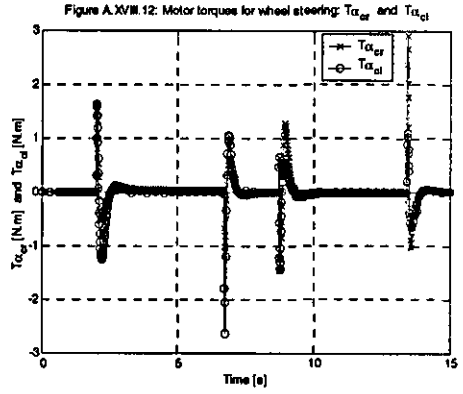


Figure A.XVIII.13: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

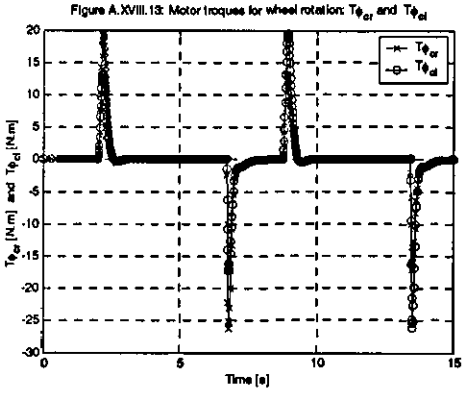


Figure A.XVIII.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

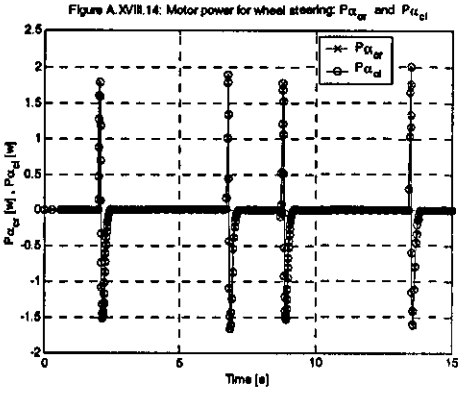
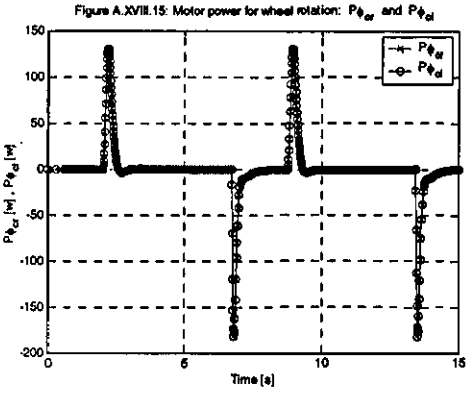


Figure A.XVIII.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XVIII.16: Steering system torques: T_{sl} and T_{sr}

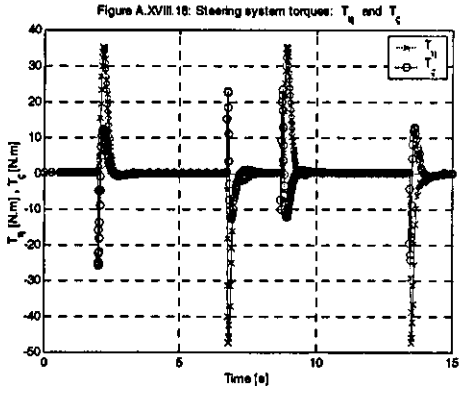


Figure A.XVIII.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

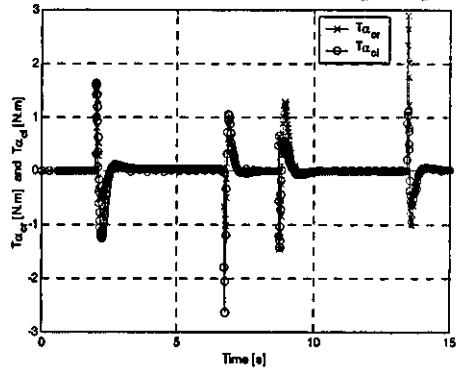


Figure A.XVIII.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

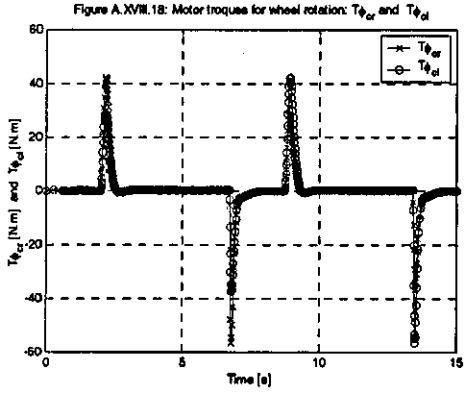


Figure A.XVIII.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

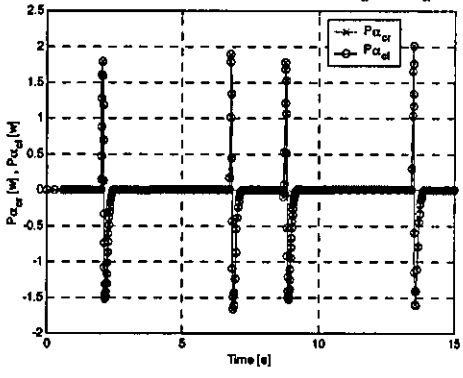
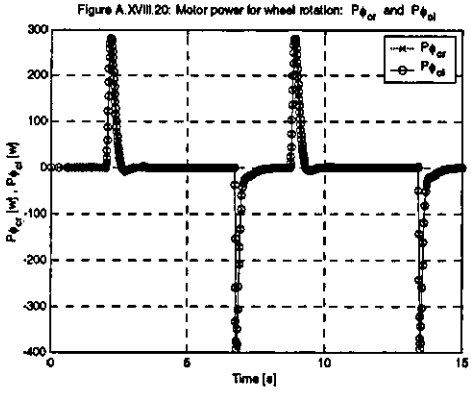


Figure A.XVIII.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XIX ROBOT TYPE (2,0): Corridor deliberative path planning. Line tracking and docking. Single robot

Simulation results

Figure A.XIX.1: Line tracking and docking(Robot type(2,0)): Sequence of posture snapshot

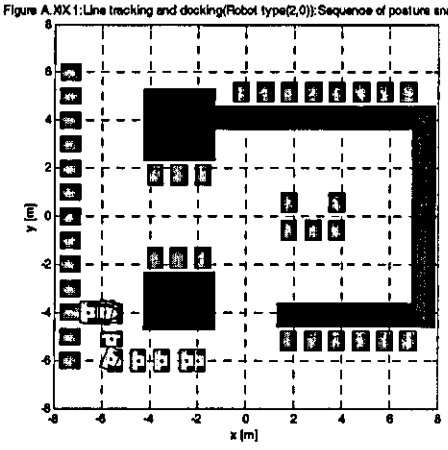


Figure A.XIX.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

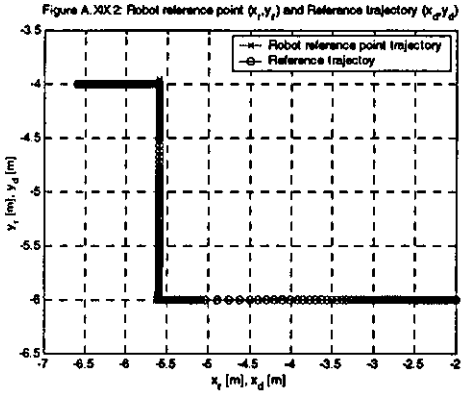


Figure A.XIX.3: Robot reference point posture variables: x_r, y_r and θ

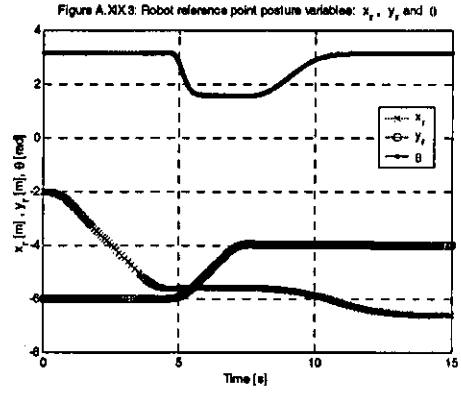


Figure A.XIX.4: Robot reference point tracking error: e_{xr} and e_{yr}

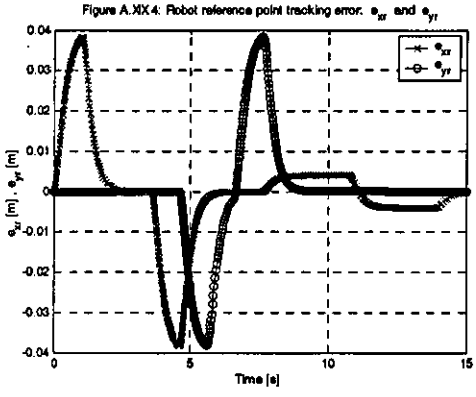


Figure A.XIX.5: Posture generalized velocities: x'_p, y'_p and θ'

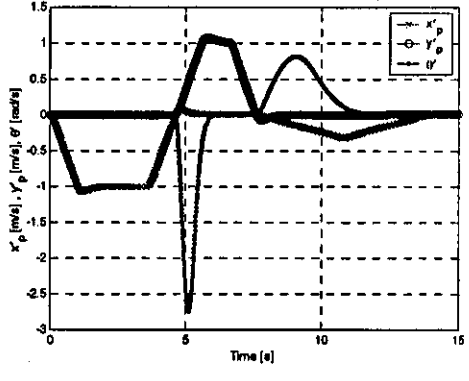


Figure A.XIX.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

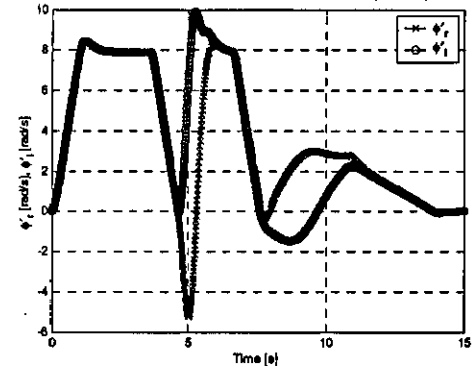


Figure A.XIX.7: Steering system velocities: x'_1 and θ'

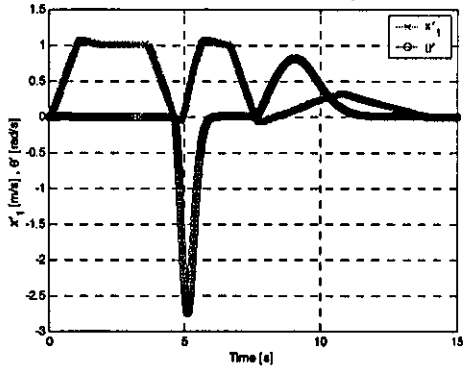


Figure A.XIX.8: Steering system accelerations: x''_1 and θ''

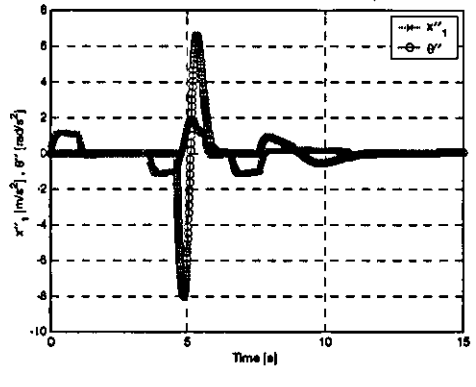


Figure A.XIX.9: Steering torques: Tx_1 and T_θ

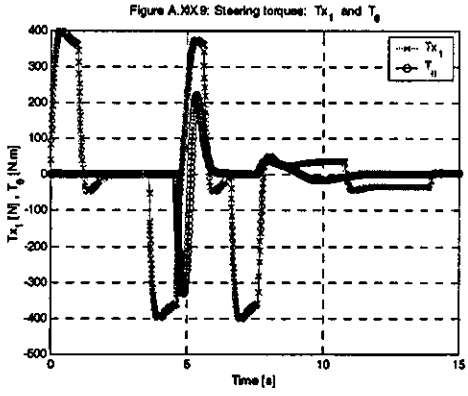


Figure A.XIX.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

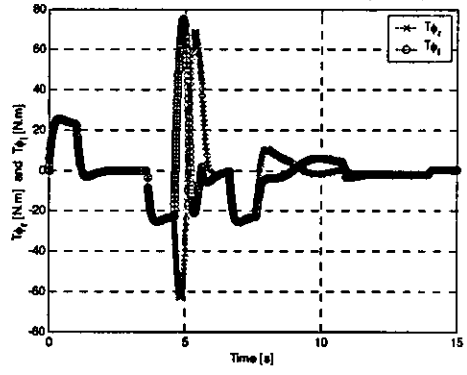
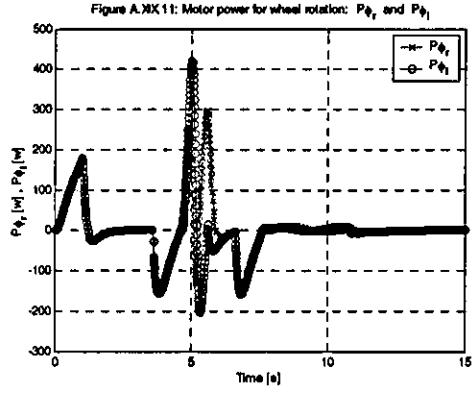


Figure A.XIX.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX XX ROBOT TYPE (1,1): Corridor deliberative path planning. Tracking moving backwards. Robot-trolley configuration. Load: 50 and 550

Simulation results

A.XX1: Line tracking moving backwards (Robot tipe(1,1)): Sequence of posture snapshots

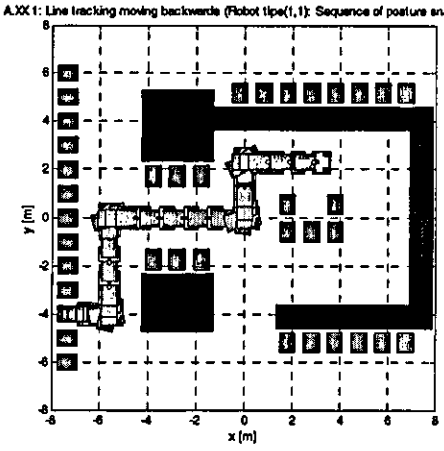


Figure A.XX2: Robot reference point (x_r,y_r) and Reference trajectory (x_d,y_d)

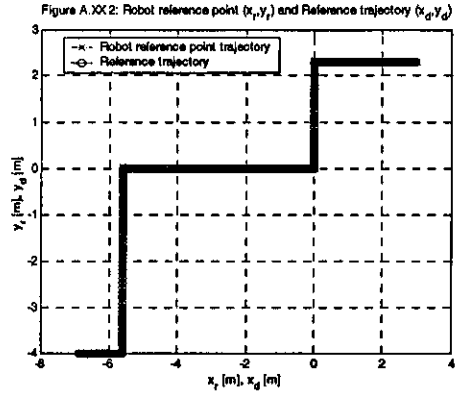


Figure A.XX3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

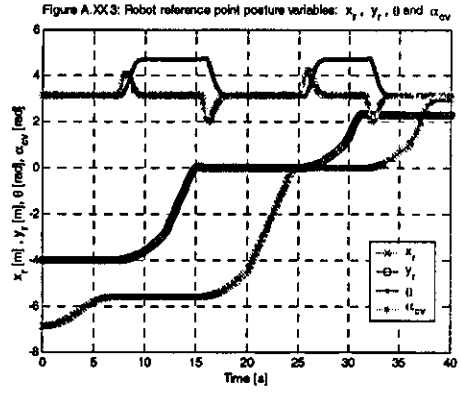


Figure A.XX4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

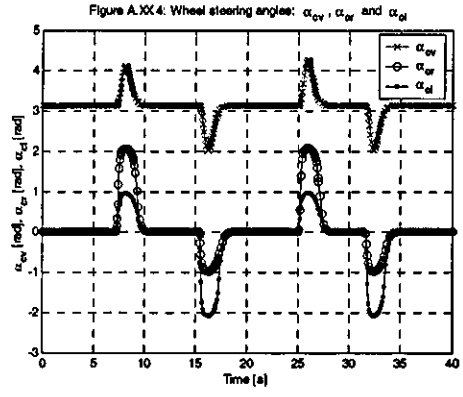


Figure A.XX5: Robot reference point tracking error: e_{xr} and e_{yr}

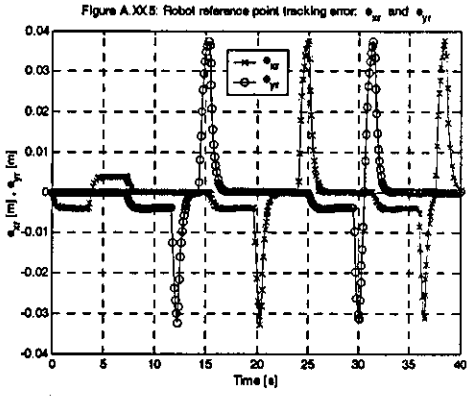


Figure A.XX6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

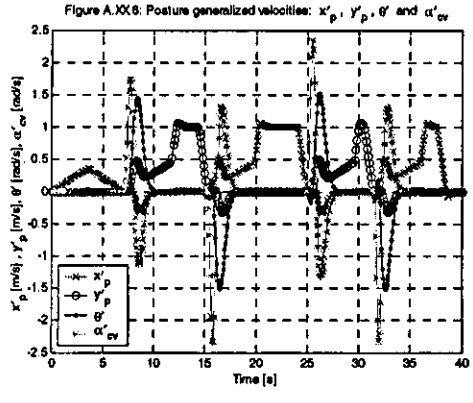


Figure A.XX7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

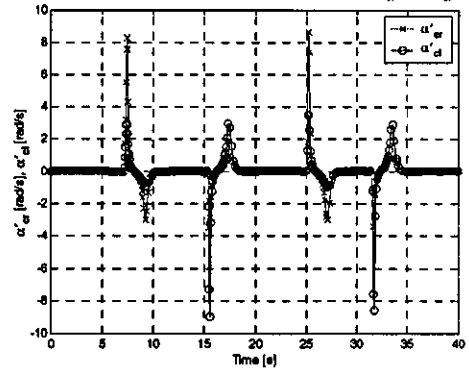


Figure A.XX8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

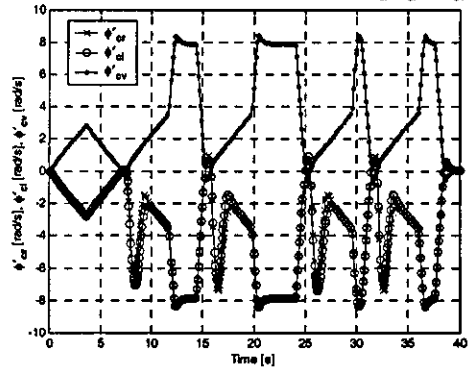


Figure A.XX.9: Steering system velocities: η and ζ

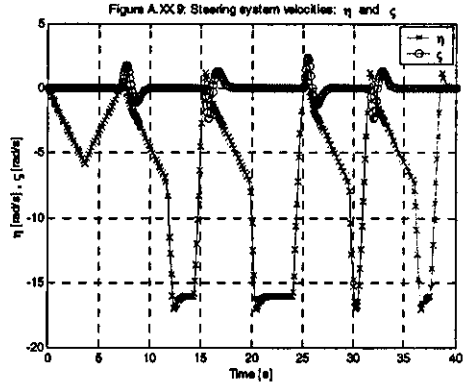
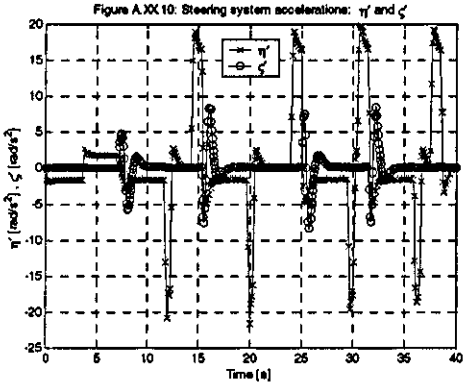


Figure A.XX.10: Steering system accelerations: η' and ζ'



Load = 50kg

Figure A.XX.11: Steering system torques: T_η and T_ζ

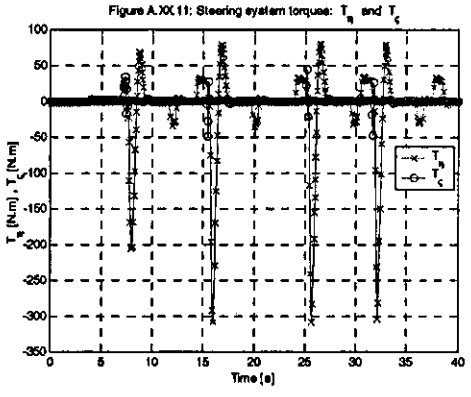


Figure A.XX.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

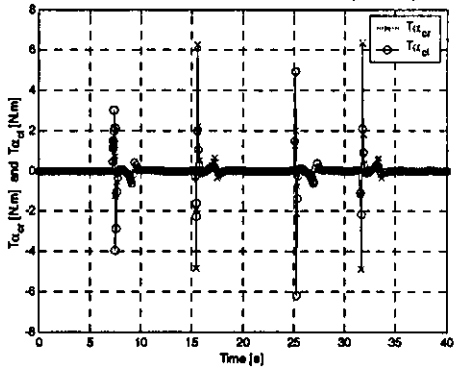


Figure A.XX.13: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

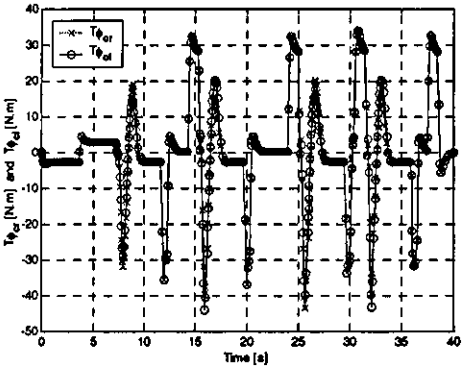


Figure A.XX.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

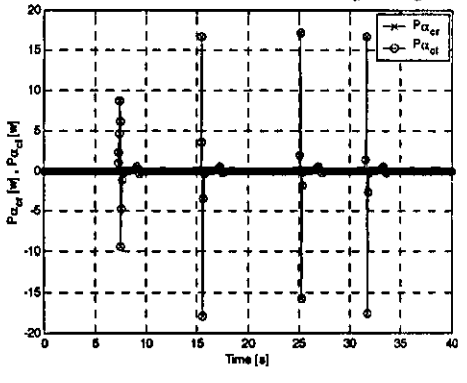
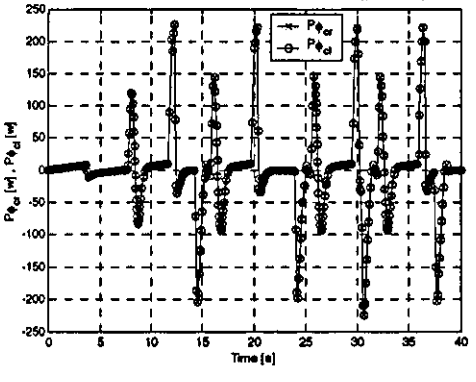


Figure A.XX.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XX.16: Steering system torques: T_s and T_c

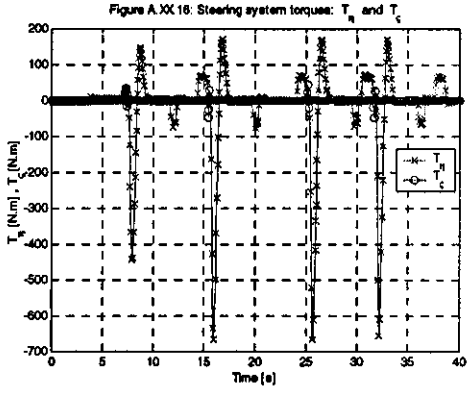


Figure A.XX.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

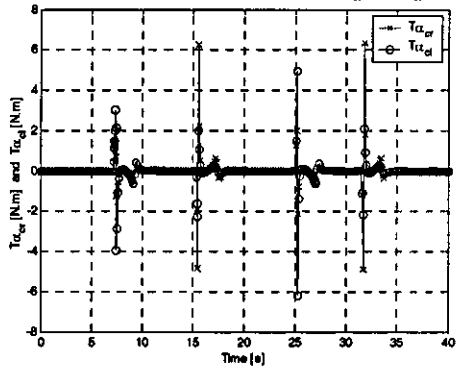


Figure A.XX.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

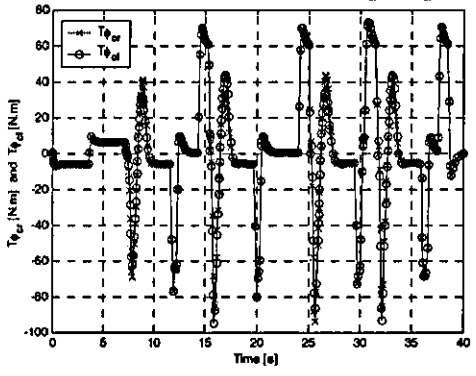


Figure A.XX.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

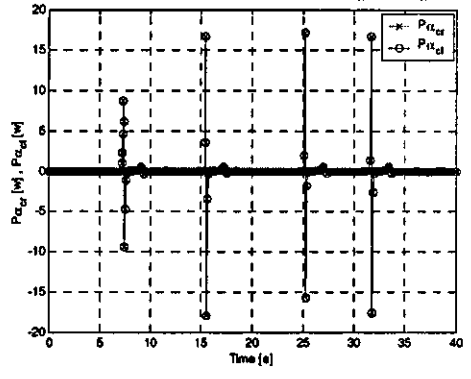
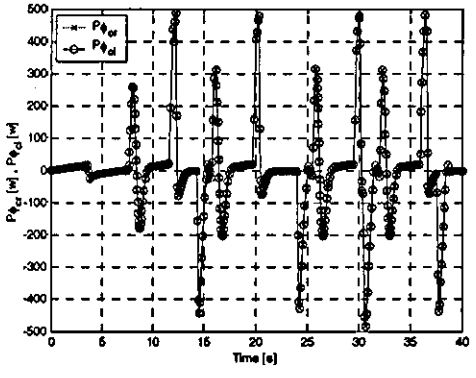


Figure A.XX.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XXI ROBOT TYPE (1,1): Corridor deliberative path planning. Docking moving forwards. Robot-trolley configuration. Load: 50 and 550

Simulation results

A.XII.1: Docking moving forwards (Robot tipe(1,1): Sequence of posture snapshots

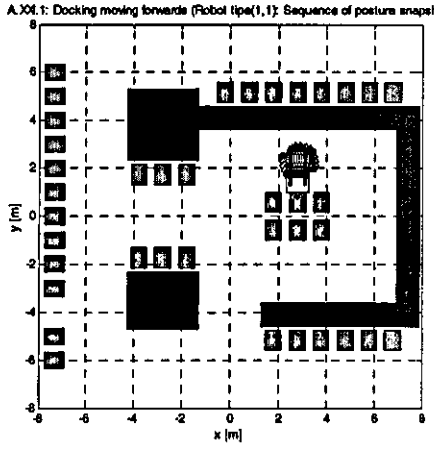


Figure A.XII.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

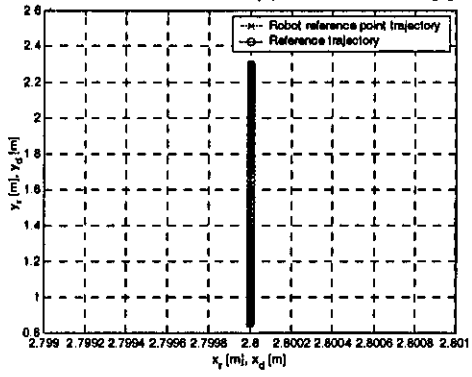


Figure A.XII.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

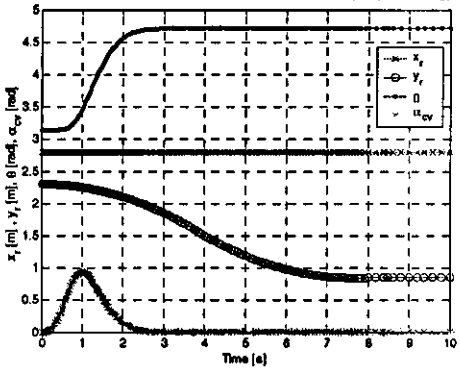


Figure A.XII.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

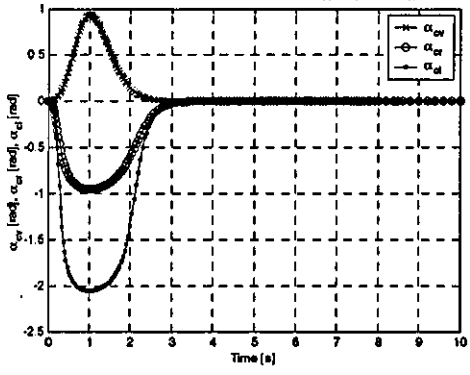


Figure A.XII.5: Robot reference point tracking error: e_{xr} and e_{yr}

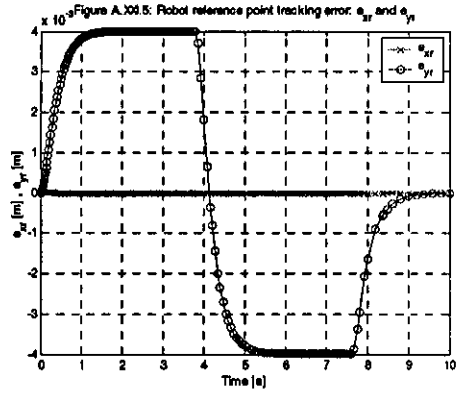


Figure A.XII.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

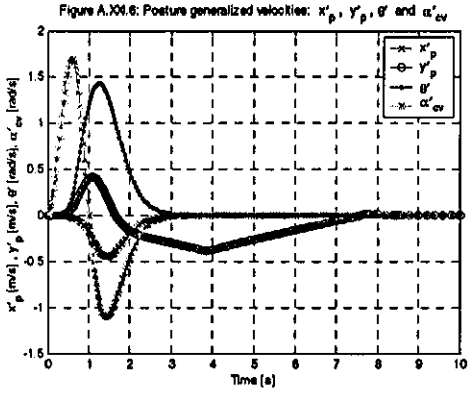


Figure A.XII.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

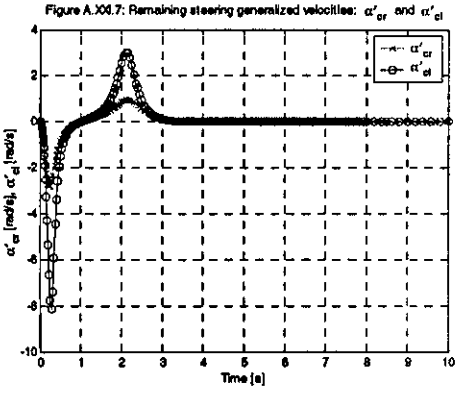


Figure A.XII.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

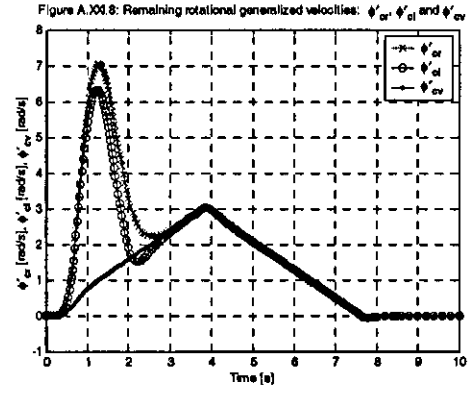


Figure A.XI4.9: Steering system velocities: η and ζ

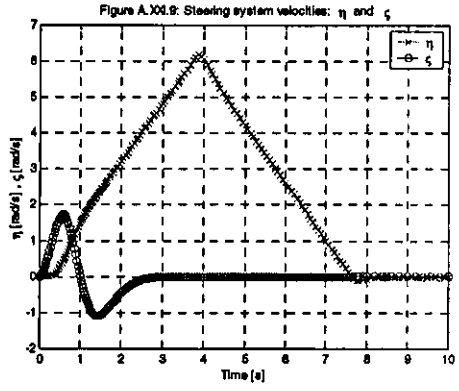
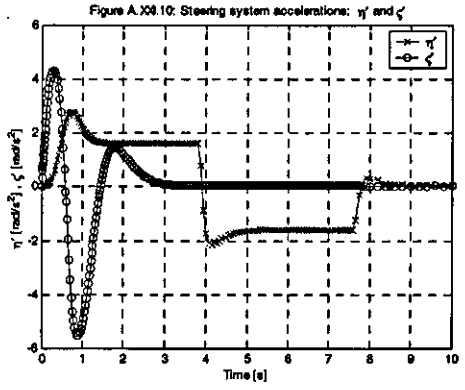


Figure A.XI4.10: Steering system accelerations: η' and ζ'



Load = 50kg

Figure A.XI4.11: Steering system torques: T_η and T_ζ

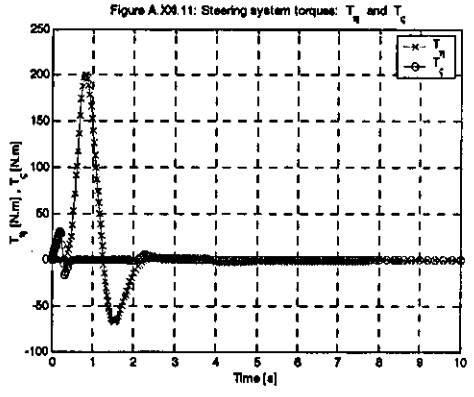


Figure A.XI4.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

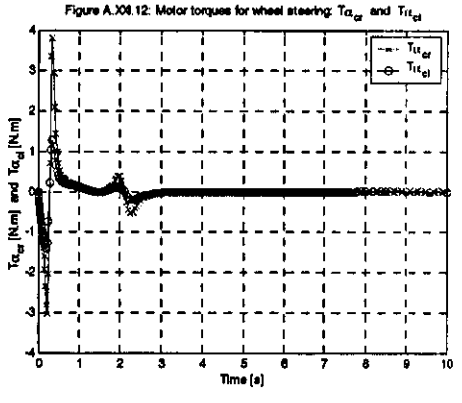


Figure A.XI4.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

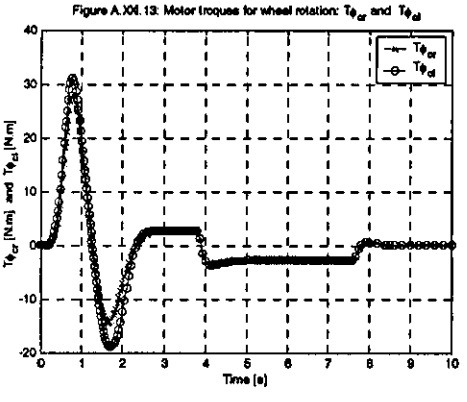


Figure A.XI4.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

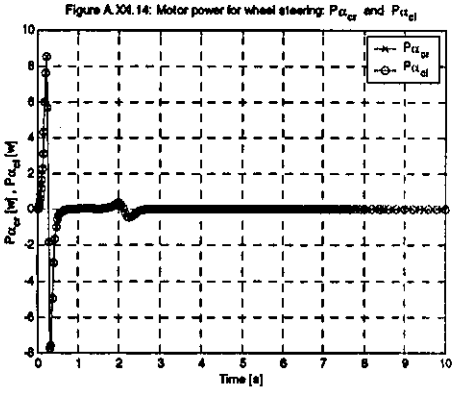
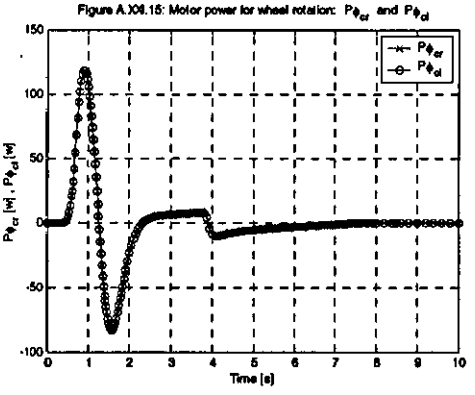


Figure A.XI4.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure A.XXI.16: Steering system torques: T_{η} and T_{ς}

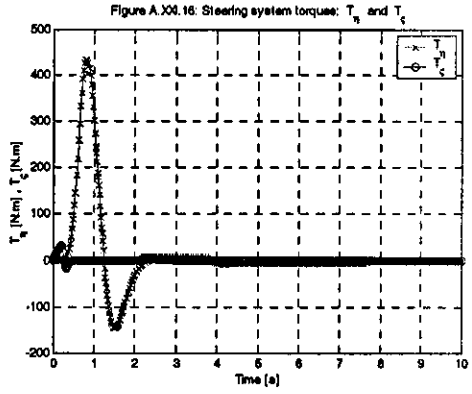


Figure A.XXI.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

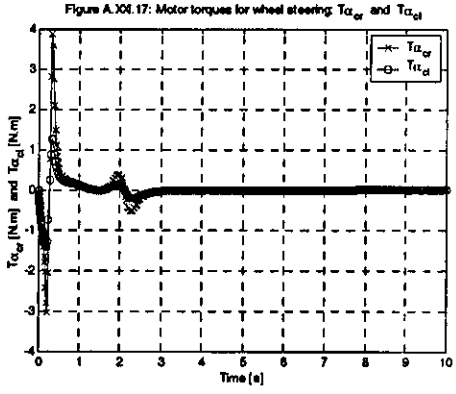


Figure A.XXI.18: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

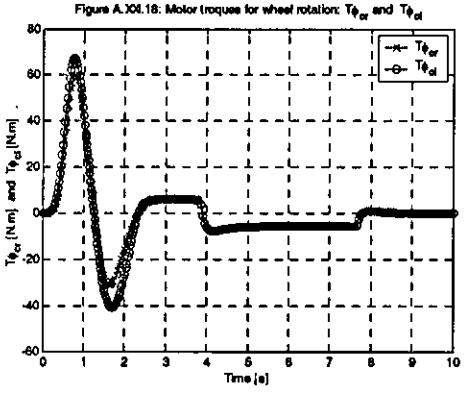


Figure A.XXI.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

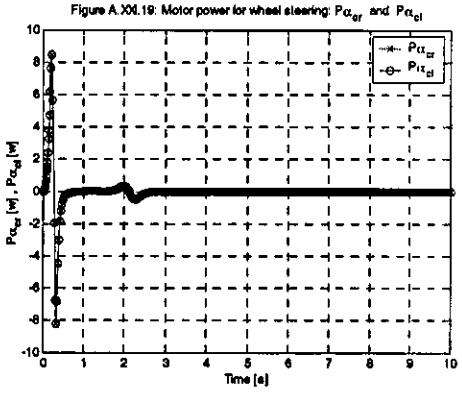
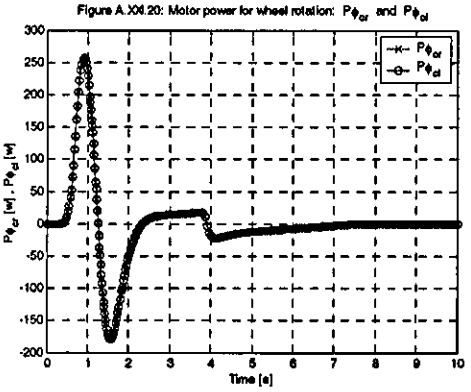


Figure A.XXI.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XXII ROBOT TYPE (2,0): Corridor deliberative path planning. Undocking and line tracking. Single-robot

Simulation results

Figure A.XXII.1:Undocking and line tracking(Robot type(2,0)) Seq. of posture snapshots

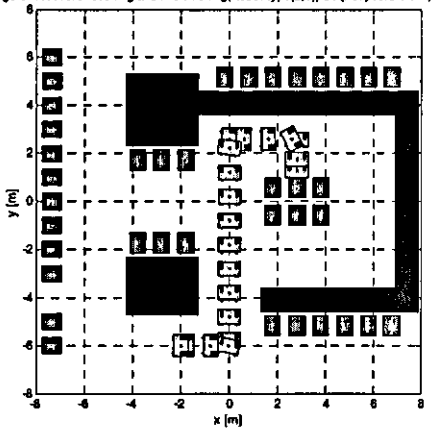


Figure A.XXII.2: Robot reference point (x_r, y_r) and Reference trajectory (x_d, y_d)

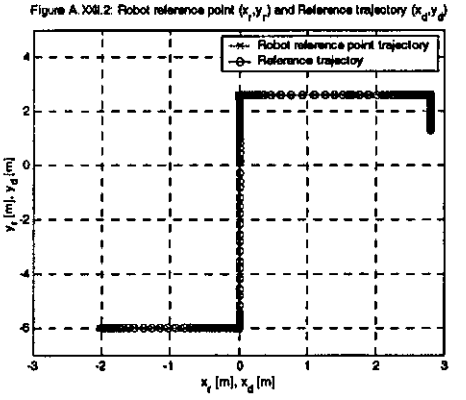


Figure A.XXII.3: Robot reference point posture variables: x_r, y_r and θ

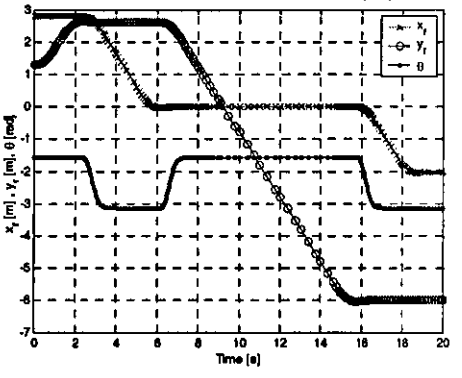


Figure A.XXII.4: Robot reference point tracking error: e_{xr} and e_{yr}

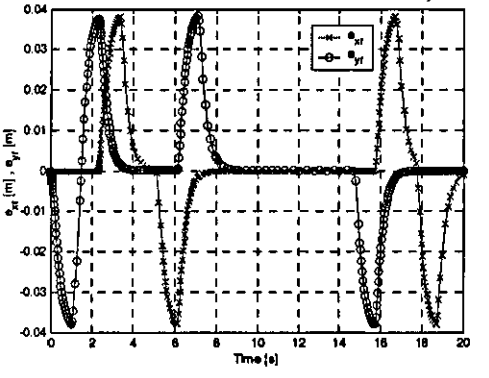


Figure A.XXII.5: Posture generalized velocities: x'_p, y'_p and θ'

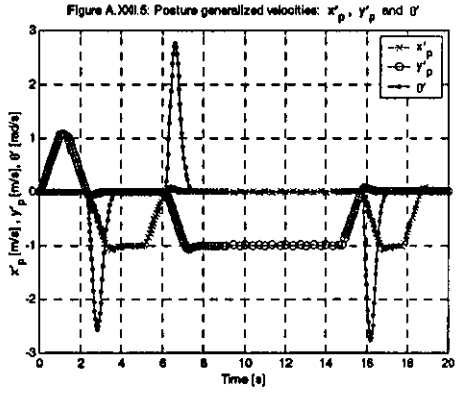


Figure A.XXII.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

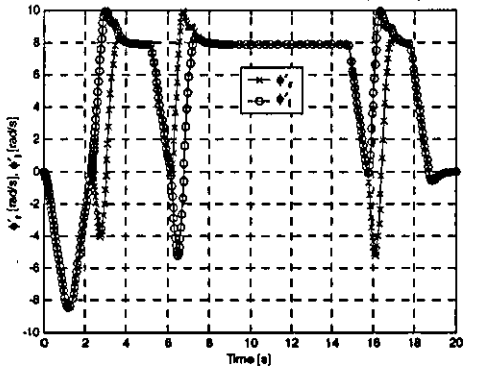


Figure A.XXII.7: Steering system velocities: x'_t and θ'

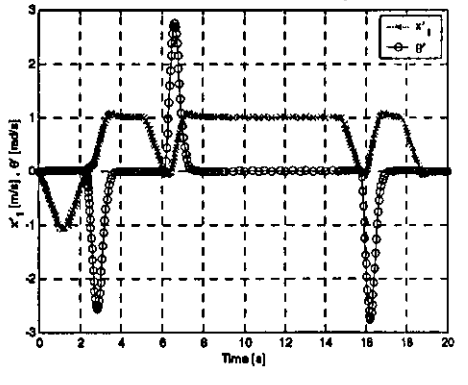


Figure A.XXII.8: Steering system accelerations: x''_t and θ''

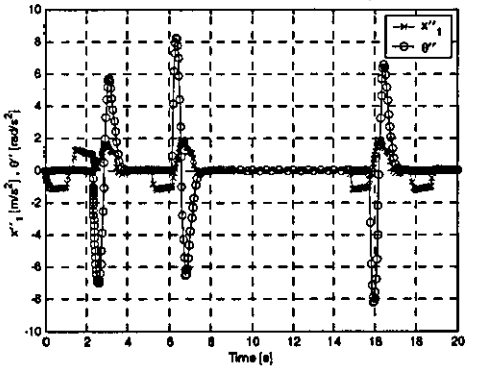


Figure A.XXII.9: Steering torques: Tx_1 and T_θ

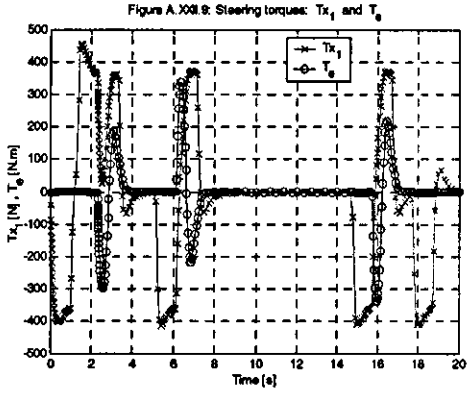


Figure A.XXII.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

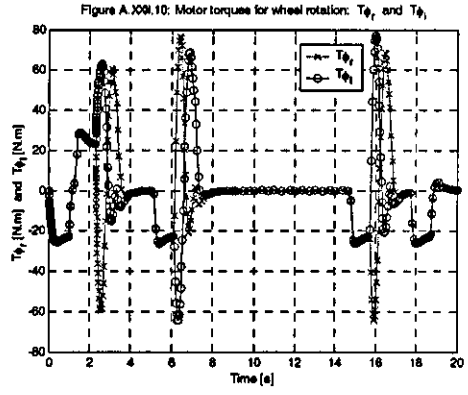
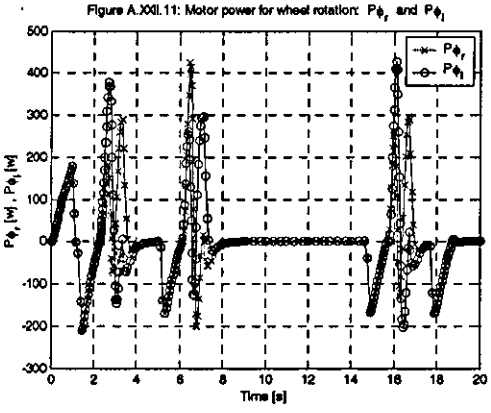


Figure A.XXII.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX XXIII ROBOT TYPE (2,0): Corridor reactive path planning. Single robot

Simulation results

Figure A.XXIII.1:Corridor reactive path planning (Robot type(2,0)): Seq. of post. snapshots

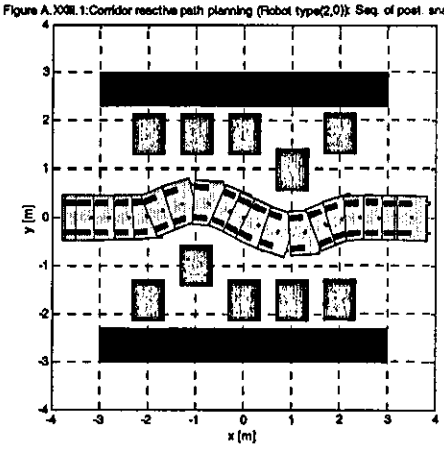


Figure A.XXIII.2: Robot reference point (x_r,y_r) and Reactive reference trajectory (x_d,y_d)

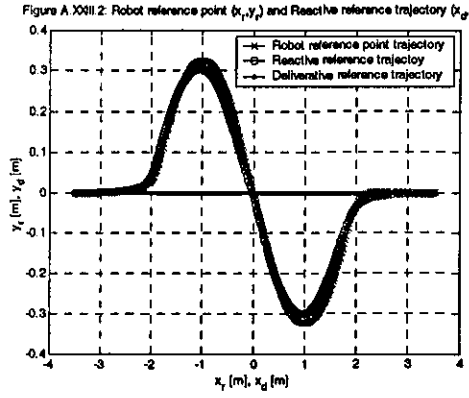


Figure A.XXIII.3: Robot reference point posture variables: x_r, y_r and θ

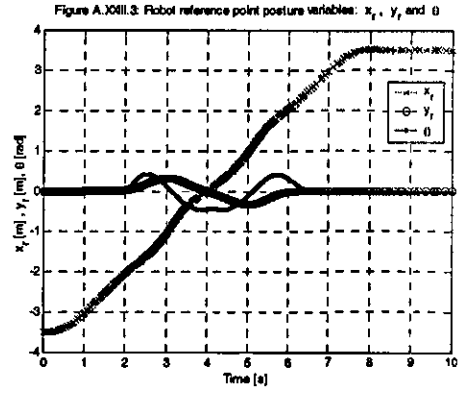


Figure A.XXIII.4: Robot reference point tracking error: e_{xr} and e_{yr}

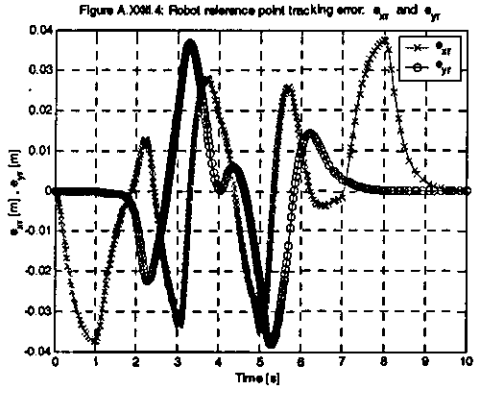


Figure A.XXIII.5: Posture generalized velocities: x'_p, y'_p and θ'

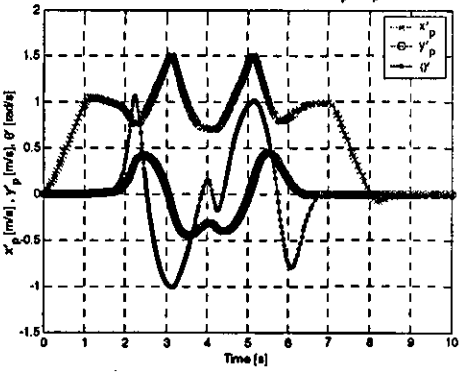


Figure A.XXIII.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

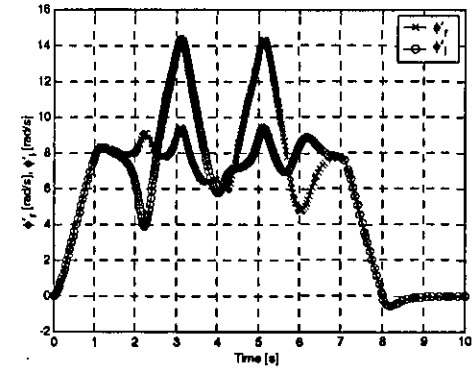


Figure A.XXIII.7: Steering system velocities: x'_1 and θ'

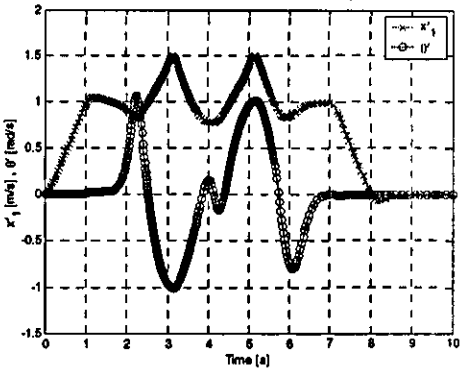


Figure A.XXIII.8: Steering system accelerations: x''_1 and θ''

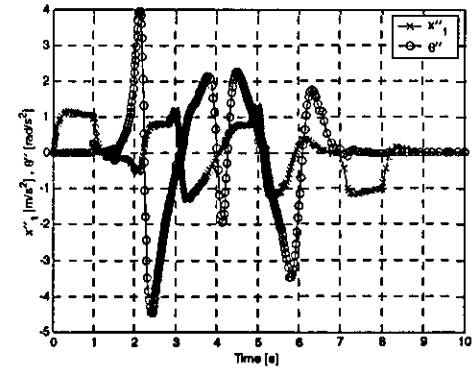


Figure A.XXIII.9: Steering torques: Tx_1 and T_θ

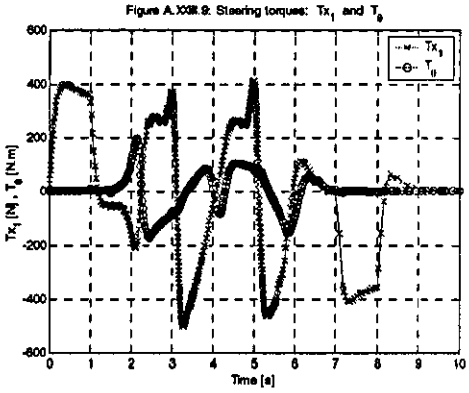


Figure A.XXIII.10: Motor torques for wheel rotation: T_{ϕ_r} and T_{ϕ_l}

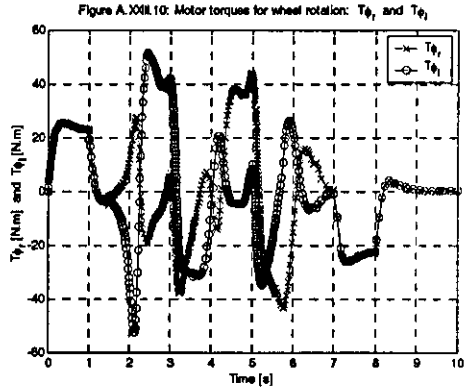
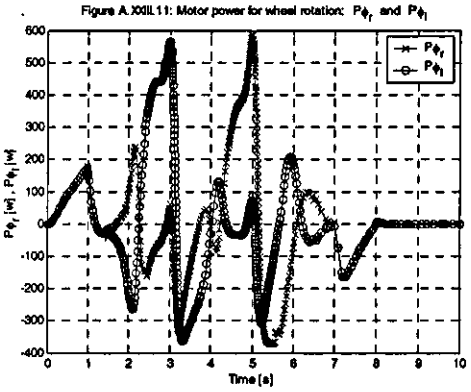


Figure A.XXIII.11: Motor power for wheel rotation: P_{ϕ_r} and P_{ϕ_l}



APPENDIX XXIV ROBOT TYPE (1,1): Corridor reactive path planning. Robot-trolley configuration. Load: 50 and 550

Simulation results

Figure A.XXIV.1:Corridor reactive path planning(Robot type(1,1)):Seq. of post. snapshots

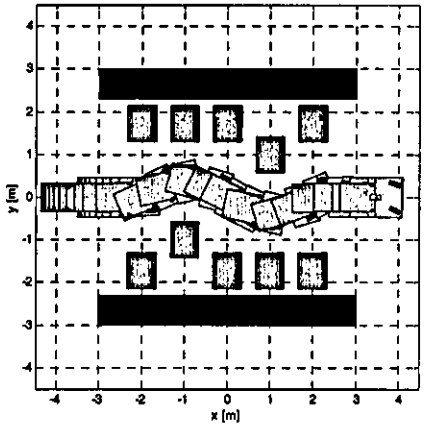


Figure A.XXIV.2: Robot reference point (x_r,y_r) and Reactive reference trajectory (x_d,y_d)

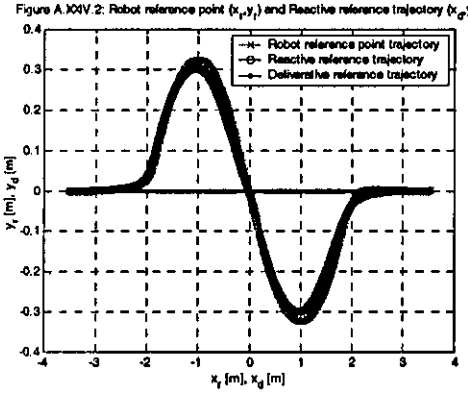


Figure A.XXIV.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

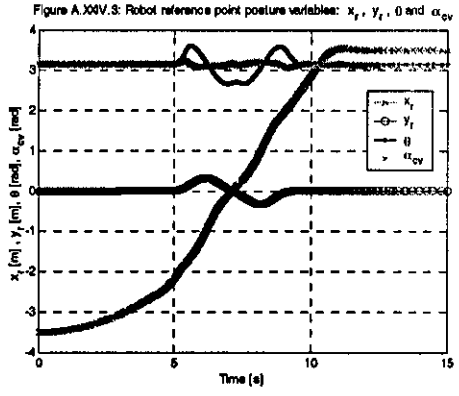


Figure A.XXIV.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

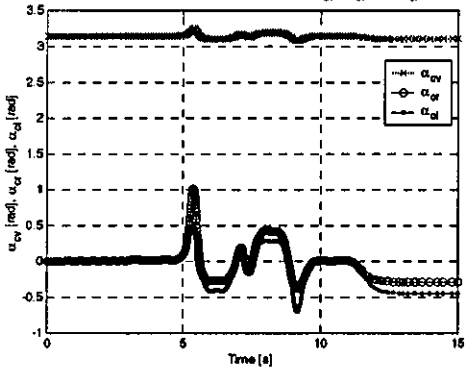


Figure A.XXIV.5: Robot reference point tracking error: e_{xr} and e_{yr}

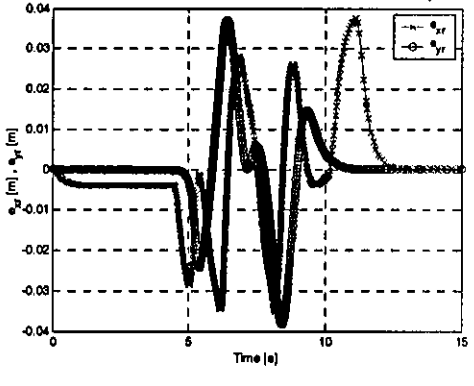


Figure A.XXIV.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

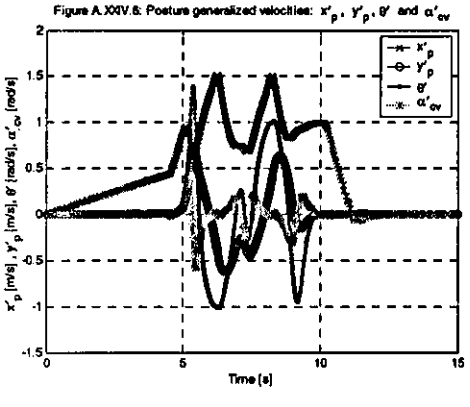


Figure A.XXIV.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

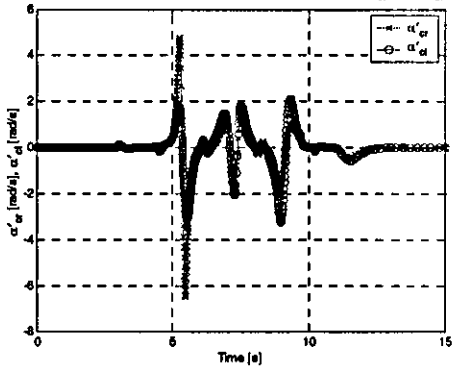


Figure A.XXIV.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

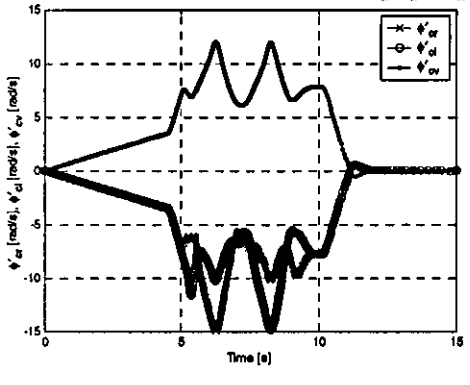


Figure A.XXIV.9: Steering system velocities: η and ς

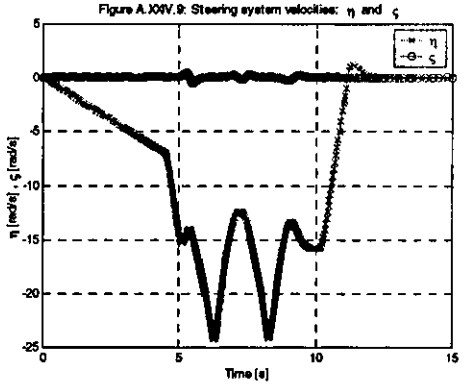
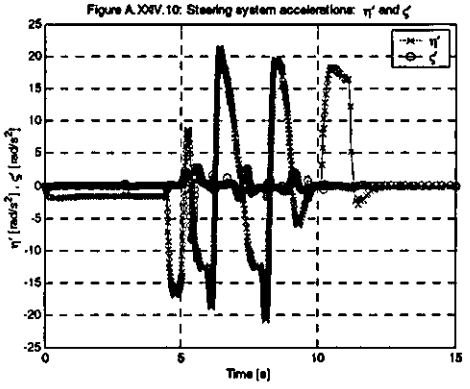


Figure A.XXIV.10: Steering system accelerations: η' and ς'



Load = 50kg

Figure A.XXIV.11: Steering system torques: T_η and T_ς

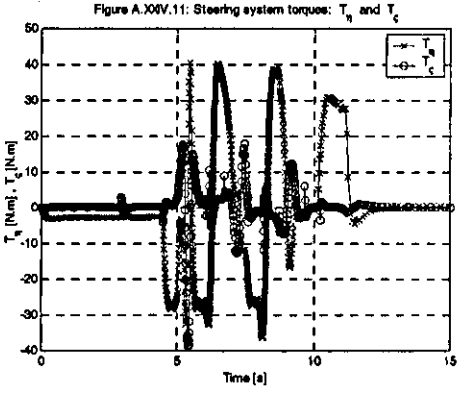


Figure A.XXIV.12: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

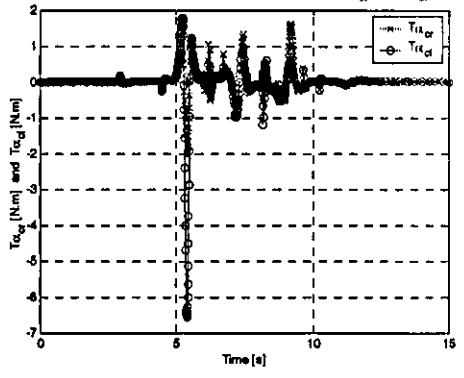


Figure A.XXIV.13: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

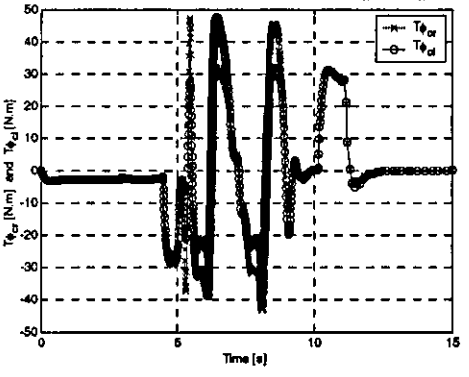


Figure A.XXIV.14: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

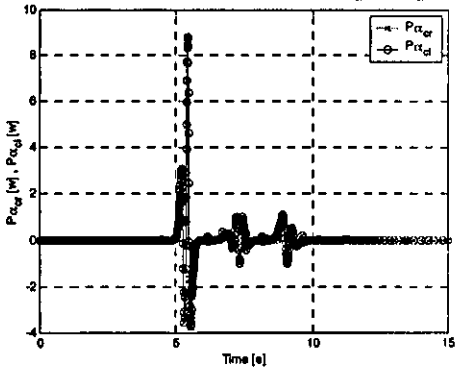
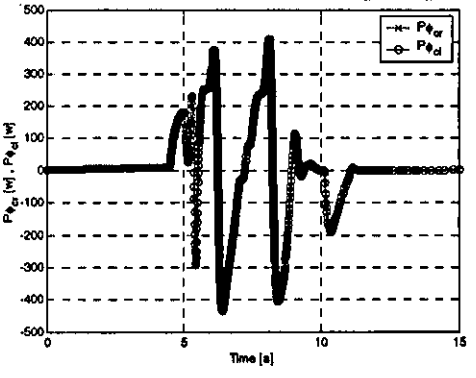


Figure A.XXIV.15: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



Load = 550kg

Figure AXXV.16: Steering system torques: T_η and T_ζ

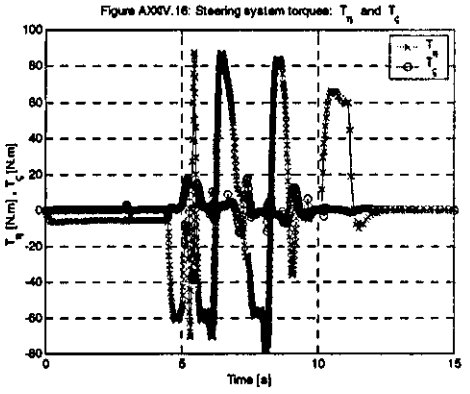


Figure AXXV.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

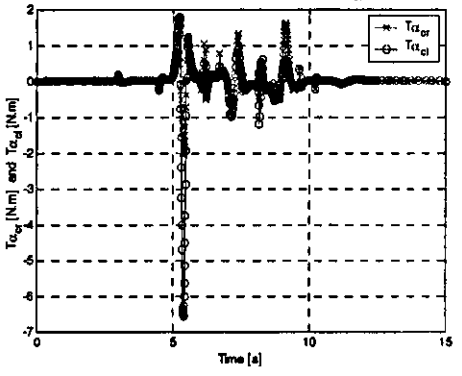


Figure AXXV.18: Motor torques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

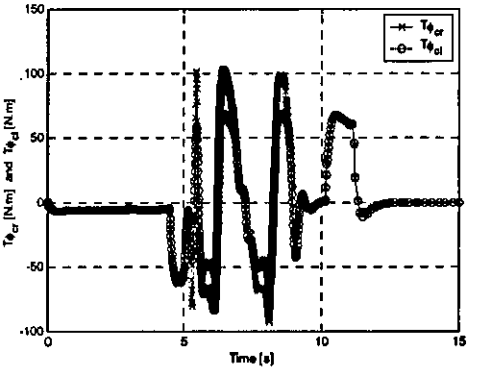


Figure AXXV.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

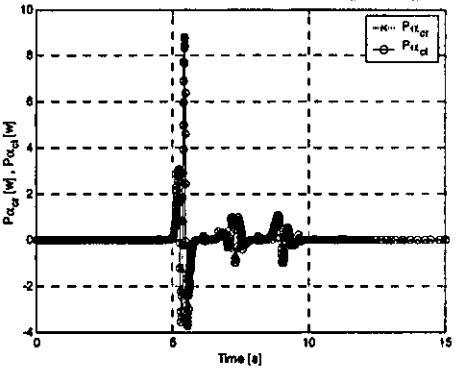
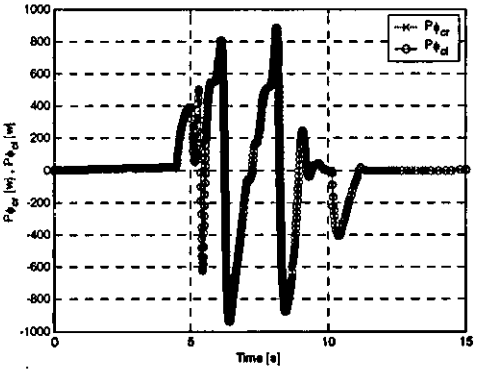


Figure AXXV.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XXV ROBOT TYPE (2,0): Open area reactive path planning. Single-robot

Simulation results

Figure A.XXV.1:Open area reactive path planning(Robot type(2,0)):Seq. of post. snapshot

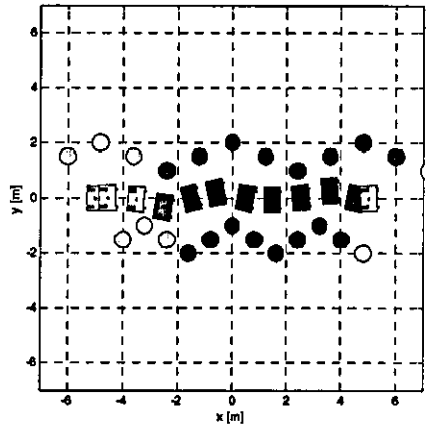


Figure A.XXV.2: Robot reference point (x_r,y_r) and Reactive reference trajectory (x_d,y_d)

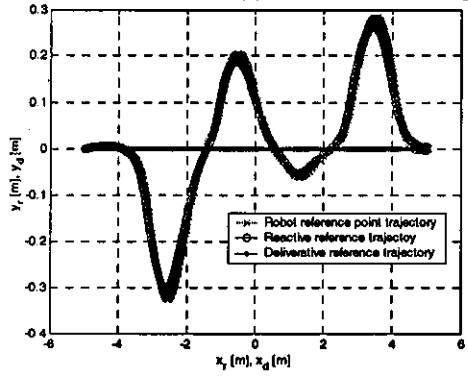


Figure A.XXV.3: Robot reference point posture variables: x_r, y_r and θ

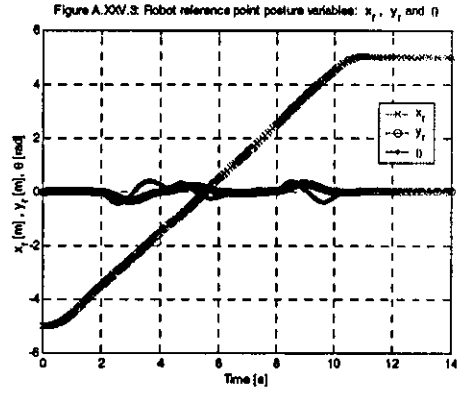


Figure A.XXV.4: Robot reference point tracking error: e_{xr} and e_{yr}

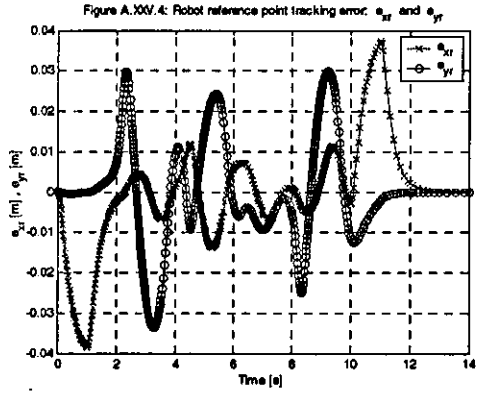


Figure A.XXV.5: Posture generalized velocities: x'_p, y'_p and θ'

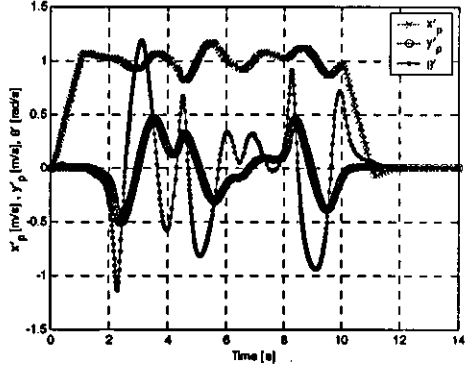


Figure A.XXV.6: Remaining generalized velocities: ϕ'_r and ϕ'_l

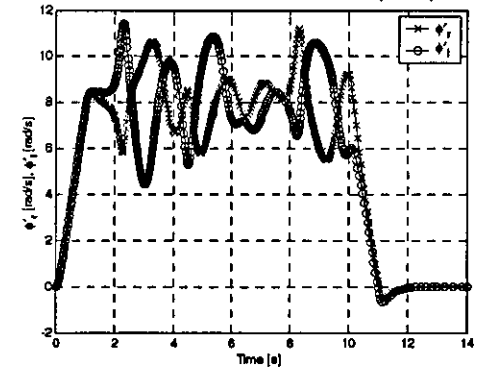


Figure A.XXV.7: Steering system velocities: x'_1 and θ'

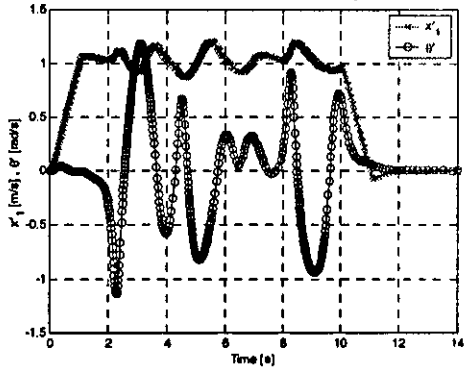


Figure A.XXV.8: Steering system accelerations: x''_1 and θ''

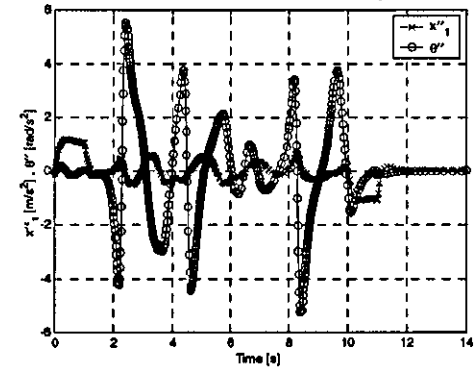


Figure A.XXV.9: Steering torques: Tx_1 and T_θ

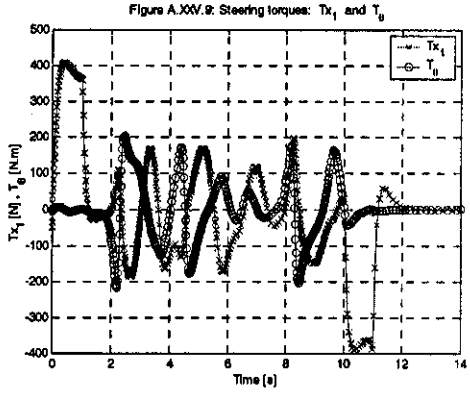


Figure A.XXV.10: Motor torques for wheel rotation: $T\phi_r$ and $T\phi_l$

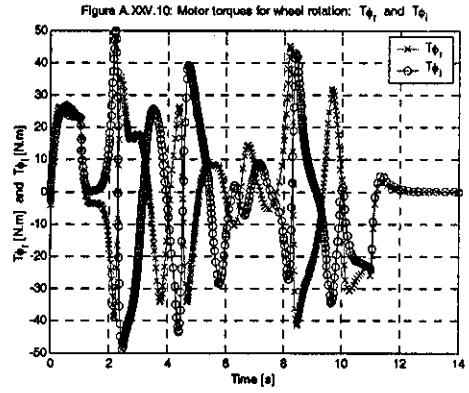
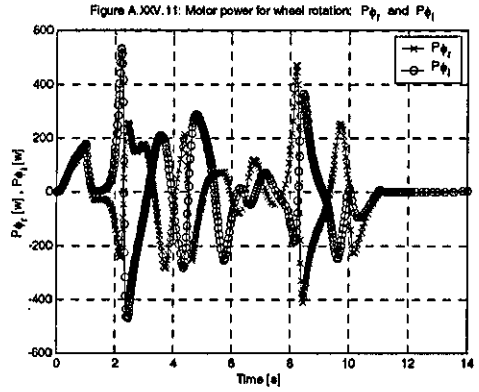


Figure A.XXV.11: Motor power for wheel rotation: $P\phi_r$ and $P\phi_l$



APPENDIX XXVI ROBOT TYPE (1,1): Open area reactive path planning. Robot-trolley configuration. Load: 50 and 550

Simulation results

Figure A.XXVI.1:Open area reactive path planning(Robot type(1,1)) Seq.of post.snapshots

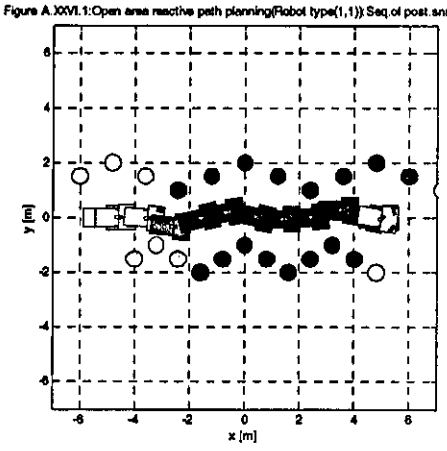


Figure A.XXVI.2: Robot reference point (x_r, y_r) and Reactive reference trajectory (x_d, y_d)

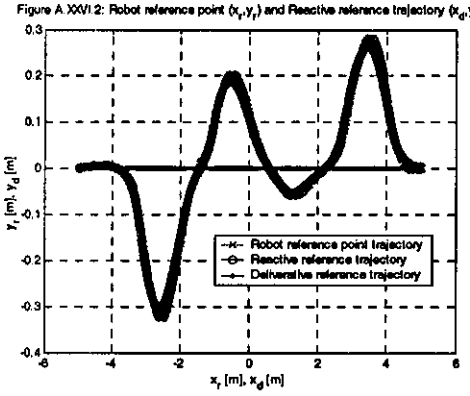


Figure A.XXVI.3: Robot reference point posture variables: x_r, y_r, θ and α_{cv}

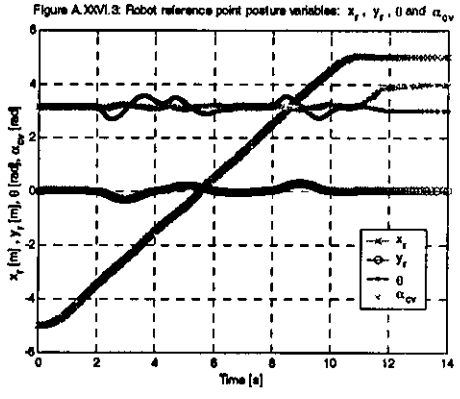


Figure A.XXVI.4: Wheel steering angles: α_{cv}, α_{cr} and α_{cl}

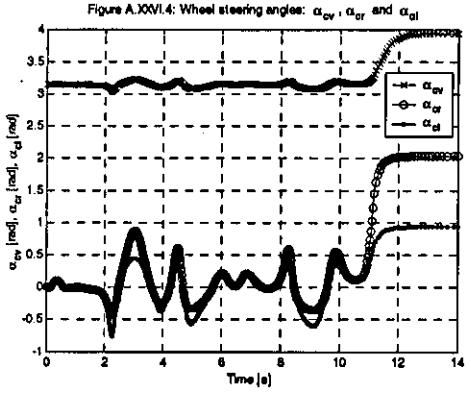


Figure A.XXVI.5: Robot reference point tracking error: e_{xr} and e_{yr}

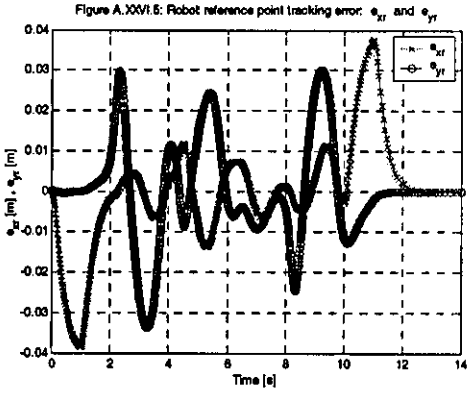


Figure A.XXVI.6: Posture generalized velocities: x'_p, y'_p, θ' and α'_{cv}

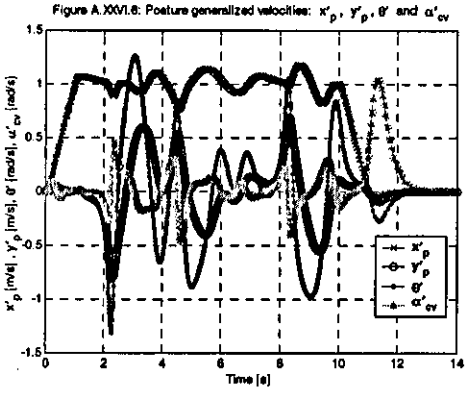


Figure A.XXVI.7: Remaining steering generalized velocities: α'_{cr} and α'_{cl}

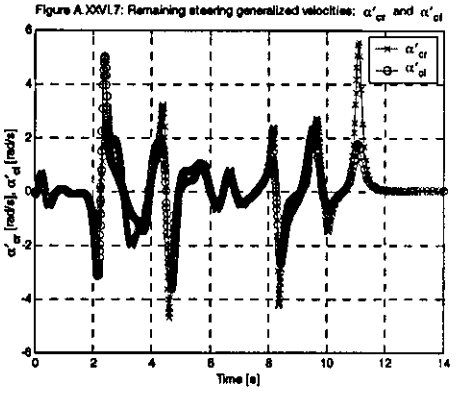


Figure A.XXVI.8: Remaining rotational generalized velocities: ϕ'_{cr}, ϕ'_{cl} and ϕ'_{cv}

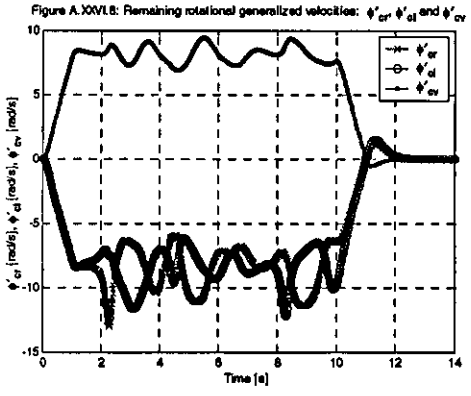


Figure A.XXVI.9: Steering system velocities: η and ζ

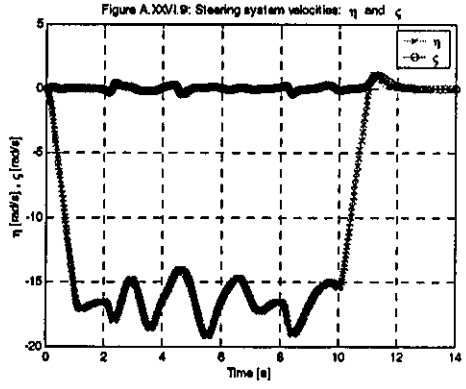
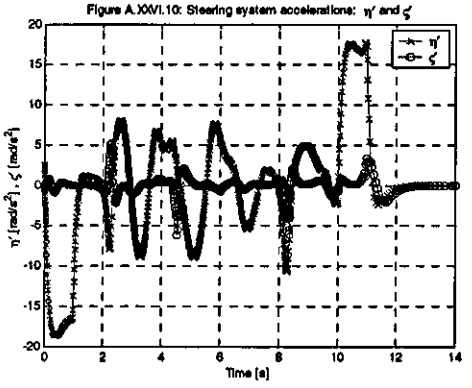


Figure A.XXVI.10: Steering system accelerations: η' and ζ'



Load = 50kg

Figure A.XXVI.11: Steering system torques: T_η and T_ζ

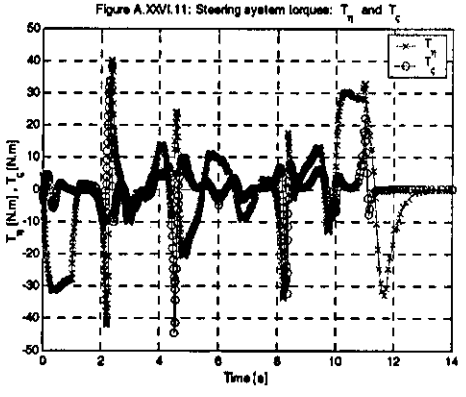


Figure A.XXVI.12: Motor torques for wheel steering: $T\alpha_{or}$ and $T\alpha_{ol}$

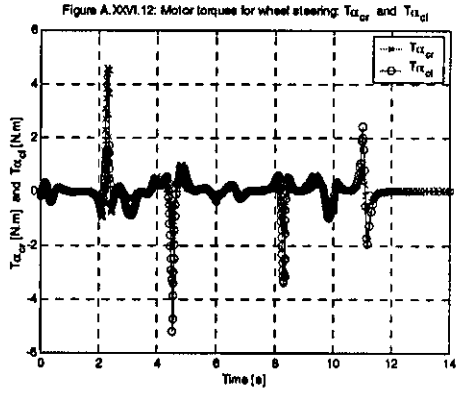


Figure A.XXVI.13: Motor torques for wheel rotation: $T\phi_{or}$ and $T\phi_{ol}$

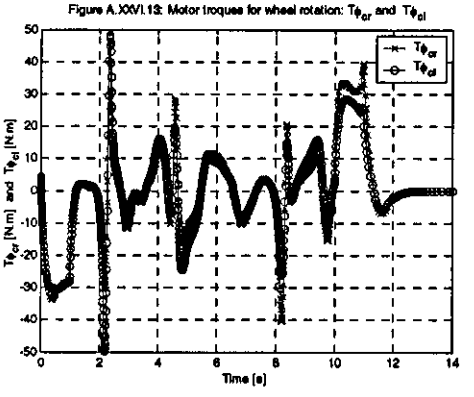


Figure A.XXVI.14: Motor power for wheel steering: $P\alpha_{or}$ and $P\alpha_{ol}$

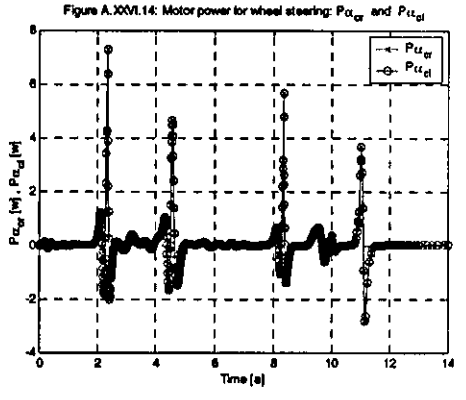
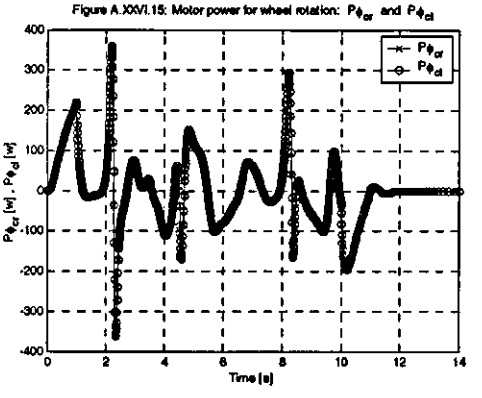


Figure A.XXVI.15: Motor power for wheel rotation: $P\phi_{or}$ and $P\phi_{ol}$



Load = 550kg

Figure A.XXVI.16: Steering system torques: T_s and T_c

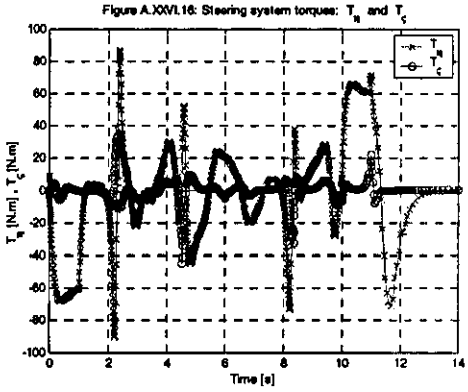


Figure A.XXVI.17: Motor torques for wheel steering: $T\alpha_{cr}$ and $T\alpha_{cl}$

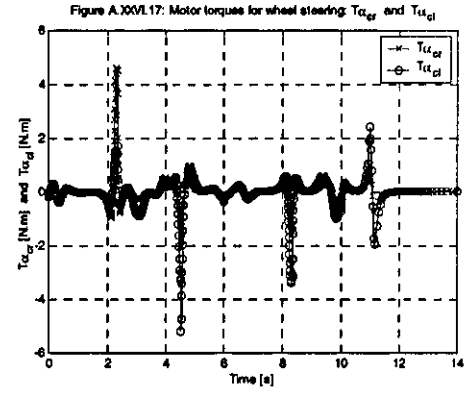


Figure A.XXVI.18: Motor troques for wheel rotation: $T\phi_{cr}$ and $T\phi_{cl}$

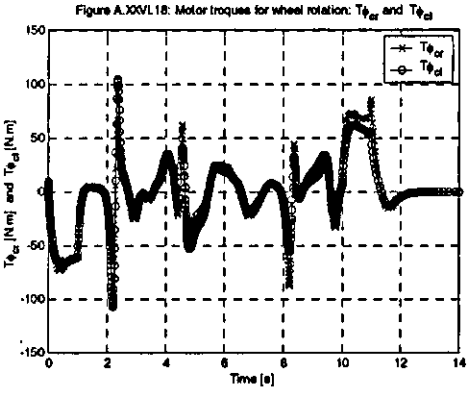


Figure A.XXVI.19: Motor power for wheel steering: $P\alpha_{cr}$ and $P\alpha_{cl}$

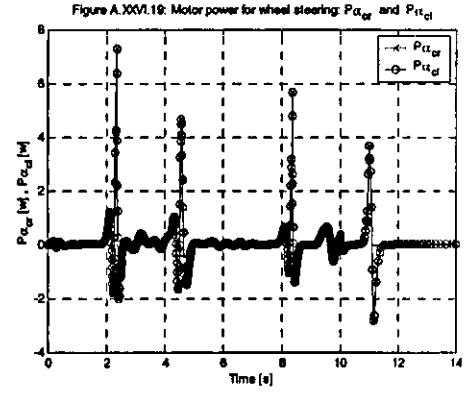
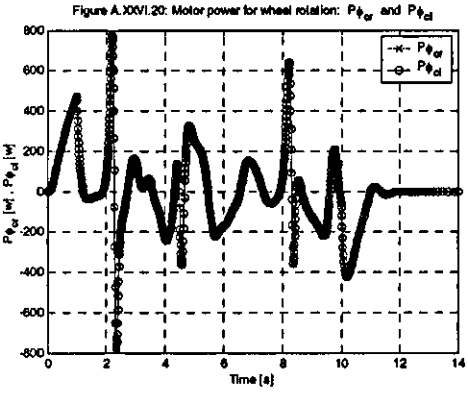


Figure A.XXVI.20: Motor power for wheel rotation: $P\phi_{cr}$ and $P\phi_{cl}$



APPENDIX XXVII Publications

1) Gonzalez-Villela, V. J., Parkin, R. M. and Lopez-Parra, M. (2003a). "A model for mechatronics design with embedded micro-controllers." ***ICOM 2003: International Conference on Mechatronics***. Loughborough, UK. Professional Engineering Publishing Limited. Vol. pp. 91-96. ISBN: 1860584209.

2) Gonzalez-Villela, V. J., Parkin, R. M., Lopez-Parra, M., Dorador-Gonzalez, J. M., et al. (2004). "A wheeled mobile robot with obstacle avoidance capability." ***Ingeniería Mecánica Tecnología y Desarrollo. Revista de la Sociedad Mexicana de Ingeniería Mecánica (SOMIM).*** Vol: 1(5): 159-166. Septiembre 2004. ISSN: 1665-7381.

3) González-Villela, V. J. and Parkin, R. M. (2005). "Evadiendo obstáculos con robots móviles." ***Revista Digital Universitaria [en línea].*** Vol: **6.** ISSN:1607-6079, Available on the Internet: http://www.revista.unam.mx.

4) Gonzalez-Villela, V. J., Parkin, R. M., Lopez-Parra, M., et al. (2005). ***Dynamic torque control extension for a wheeled mobile robot with obstacle avoidance capability.*** Memorias del XI Congreso Internacional Anual SOMIM & IV Congreso Bolivariano de Ingeniería Mecánica. Morelia, Mich. México. Vol: **OT.** pp. 38-45.

Options

A model for mechatronics design with embedded micro-controllers

V J GONZALEZ-VILLELA and **R M PARKIN**
Mechatronics Research Centre, Loughborough University, UK
M LOPEZ-PARRA
Dipartamento de Ingeniería Mecatrónica, Universidad Nacional Autónoma de México

SYNOPSIS

This work presents a general theoretical design approach to the components, subsystems and systems of mechatronic systems according to the needs of mechatronic designer and focusing on the application of embedded micro-controllers. The mechatronic design model has also been developed as a tool able to help to the mechatronic designer in the development of mechatronics products during the early stages of the design process. Besides here is defined the progress and evolution of the mechatronics systems as a part of this model. The paper also reports successful examples of systems designed from this point of view, e.g. wheeled, legged and fingered robots.

1 INTRODUCTION

Mechatronics is the synergistic combination of precision mechanical engineering, electronic control, information technologies, and systematic thought applied to development of intelligent products and processes [adapt. 1]. Mechatronics technology is unique and will be oriented in the efforts to find out the best solution to the required necessities through the efficient combination of many classes of knowledge of different areas [2]. Designing mechatronics systems could be so complex since they involve the combination of different technologies in one system in different hierarchical levels; some literature has modelled this products [3][4], but a model base on the General System Theory [5][6] makes easier to understand and to design a mechatronics system. Besides, this complexity should be modelled in one of multitude of rigorous mathematical formalisms. The construction of such mathematical models and their application forms a difficult problem that should be resolved by the designer [7]. When this complexity is divided in subsystems or subtasks the problem is reduced to design primary, secondary and coordination tasks [8] where embedded micro-controllers gain an important role. In this manner, not only a MS model is a good tool for the mechatronics designers to accomplish the mathematical formalism but certainly is a good tool to design with embedded micro-controllers as well. In this work is defined the structure of the mechatronic systems from the point of view of their constructive complexity, the mechatronics system model, and finally some design examples base on this model.

2 STRUCTURE OF THE MECHATRONIC SYSTEMS FROM THE POINT OF VIEW OF THEIR CONSTRUCTIVE COMPLEXITY.

The Mechatronic System (MS) presents a constructive complexity that allows it to achieve a purpose, this complexity is formed by a group of base components (an example of these components is: screws, nuts, gear, cables, resistances, capacitors, transistors, etc.). The dispositions of these elements generate the structure of the system, so they are called "Base Elements". A **Base Element:** 1) it is indivisible for analysis purposes, 2) the effects below their structural level are ignored in the analysis, 3) it is capable of generating all the superior levels of the system together with the other ones. This way, the whole structure of the MS could be integrated by base elements, mechanisms, machines, equipments, and systems: **mechanism** is a group of elements base that interact to each other; **machine** is a group of mechanisms that interact to each other; **equipment** is a group of machines that interact to each other; and **system** is a group of equipments that interact to each other [8].

3 MECHATRONIC SYSTEM MODEL

This Mechatronic System (MS) model is based on an Open System Model for Social Organization [5] that is based on the General Systems Theory [6]. It has been adapted to MSs. In this model, the MS should satisfy a mission to be useful (purpose or reason of being) it is created to satisfy necessities of its environment (the environment is waiting for goods or services form the MS); the MS is structured in such a way that their components and base elements are designed to satisfy some final cause.

In order to fulfil the final cause, the MS is structured with a particular disposition, interaction, control and coordination of their elements that makes it able to maintain two types of processes: static processes and dynamic processes.

3.1 Static Process

The static processes or static regulations have to do with the behaviour of the MS called steady state (or approaching to it) where the MS is considered in stationary state or in dynamic equilibrium, characterized by the disappearance of changes or variations in its elements, but really far from the static equilibrium. In the static process can be distinguished the following fundamental parts:

3.1.1 The environment, limit, input and output.

All that is outside of MS limit constitutes the ***environment.*** To preserving the order and control, the MS should be connected with different segments to environment (a connection is an input/output exchange of mass, energy, and information). In fact, the environment should provide the inputs, accept the outputs, and support the purpose, as well as to provide feedback to the MS. In consequence the environment influence on the order and control of the system is fundamental. All MS have a border or ***limit*** that differentiate to others. This limit can consist of the following ones: 1) Physical limit: These are the material limit and the structural characteristics; 2) Temporary limit: These are the operating time and the expected life of the MS; 3) Social limit: It has to do with people that interact with the MS. The social limit defines the ergonomic characteristics of the MS; 4) Psychological limit: These are the psychological effects caused by the MS in the working environment. This limit defines the appearance and operating way of the MS. The limit has openings that allow the interaction I/O with environment. The grade of permeability (or opening) of the limit is fundamental to preserving

order and control of the MS. The MS takes from the environment mass, energy, or/and information through its ***inputs***. The inability to collecting enough mass, energy resources and information results in inefficient use or faulty operation of the MS. The mass, energy and information ***outputs*** or results of the process (products, services, information, etc.) are taken toward the environment with the hope of fulfilling the purpose or the implied contract between the system and environment. In this point they are also included the secondary unwanted products (contamination, waste, errors, etc.).

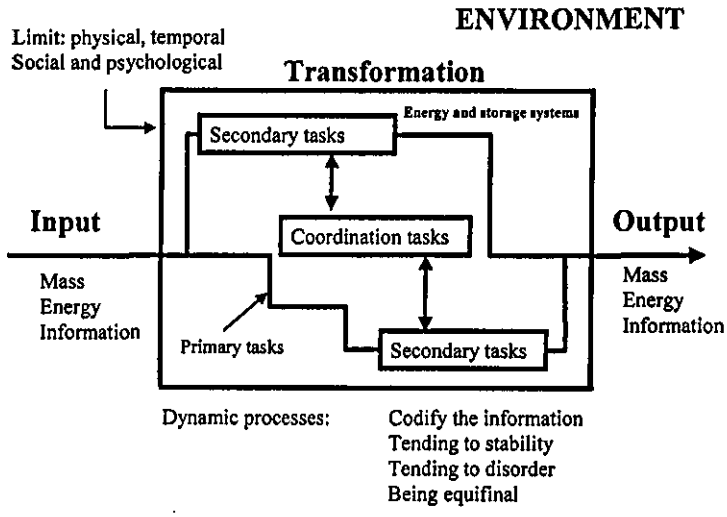


Figure 1: MS model

3.1.2 Transformation

During the transformation process the mass, energy and information inputs should turn into other things (products, services, results, information, etc.) to returning to the environment. The MS takes mass, energy and information of environment to carrying out the constitution and demolition activities (fig.1). The transformation is achieved by means of combined interactions of three processes, these are: primary tasks, secondary tasks and coordination tasks. **The Primary Tasks** refer to primary regulations or fundamental tasks to achieving the purpose, they are of type: 1) Static teleology or adaptation, when a particular disposition are useful for an specific propose and 2) Directedness based on the reach of a final state; all the systems arrive to a final condition and guide the events towards the final state. They are opened loop structures. **The Secondary Tasks** refer to the form in which system constituent elements concentrate their energy on the execution of the primary tasks. They have to do with the secondary regulations; these reside in the directedness based on the structure that is governed by the equifinality. In other words, they are closed loop structures that conduct the process in order to achieve the goal from different initial conditions. **The coordination tasks** refer to the way in that the systems divide the tasks, communicate, and interact with its subsystems. The coordination tasks are in charge of good link between the elements of the system, and the primary and secondary tasks. It has to do with the intention of the system.

Besides all these structures, in the static process is needed to add the mass, energy, and information storage systems.

3.2 The Dynamic processes

During its operation the MS parts interact each other (they maintain dynamic interactions). This way the MS is considered as a changing net of connections in movement through the time like a chain of events. The four fundamental dynamic processes are: 1) **Codifying the information:** Only the systems should react to the specific signals, which means to be programmed to responding to certain signals and ignore others. 2) **Tending to stability:** Stability means something that remains equally. The stable state is the tendency of the system to stabilize its transformation processes between certain limits, with the purpose of conserving order and control. An initial disturbance in the system causes the mobilization of energy to recovering the balance. 3) **Tending to disorder:** The physical systems go toward the simple random distribution of their elements. So the MS should act, correct or stop the process of disorganization to conserving order and control during its life. 4) **To show equifinality:** This term refers to the fact that the systems can reach the same final state starting from different initial conditions and through different ways. This is a capacity of adaptation or flexibility to carry out the work. There are many forms of achieving a result, depending on the individual circumstances. The key point is that the open systems are auto-regulated according to the capacity that they have to organize the process of their tasks to reach its purpose.

3.3 The progress and evolution in MS

The progress in MS is given by subdivision of a unitary action in various specialized parts (by differentiation) [6]. As long as MS is developed it can progresses and become more complex, and could perform new functions to face the growth, and to maintain the state of stability. So its subsystems could be divided, multiplied, and developed to perform specialized functions and additional functions. The MS could show evolution, expressed in their improvements and new designs. It is consequence of a general principle of **Progressive Mechanization** [6]. At the beginning the MS should be designed in such a way that it is governed by primary regulations and dynamic interactions among their components; later on it becomes established with physical dispositions and restriction conditions that make more efficient to the system and their parts, but, in passing, their components diminish the capacity to carry out multiple tasks, until ending up to abolish it. Associated to the Progressive Mechanization the **Progressive Centralization** [6] is presented, through which some parts win a dominant role and determine the system behaviour that makes it more unified and indivisible, as consequence the **Individualization Progressive** and the **Progressive Segregation** [6] appear. So the dynamic interaction is the widest aspect, since always it is possible to arrive up to automated function, imposing appropriate conditions of restriction using general systems laws. Then, without dropping the efficiency, a better design of a MS is carried out by designing the regulation functions in such a way that the dynamic interactions prevail to minimized control and specialized elements.

4 THE EMBEDDED MICRO-CONTROLLERS INSIDE THE MECHATRONICS SYSTEM MODEL

In this work the primary tasks type 1 are performed by boards, aesthetic elements, support and assemble structures, connectors, integrated circuits sockets, printed circuits boards, etc., and the primary tasks type 2 are performed by elements that are arranged in certain disposition and with certain shape that interact each other to producing the process, these are: gears,

bands, chains, motors, drivers, sensors, resistances, transistors, triac's, logic gates, amplifiers, etc. Superimposed in the primary tasks, the secondary tasks are in charge of reaching the same final state from different initial condition in order to restore the disturbances generated by the environment. They are feedback mechanisms for material and energetic regulation e.g. pressure regulators, sequential circuits, flip-flops, electronic filters, PID controllers, microprocessors, micro-controllers, etc. The coordination tasks are in charge of the processes and exchange information, sequence control (internal operations), communication protocol (interface man machine), and communication with other electronic systems, they can be electronic sequencers, microprocessors, micro-controllers, personal computers, main frames, etc.

In this model the secondary and coordination tasks are suited to carry out by micro-controllers, they have the ability to transfer, process, storage and manipulate information to control the actions of the different subsystems. In fact the embedded micro-controllers can be classified as a machine structure as defined in this work. Explanation of some examples can help to understand this model.

For example in figure 2 we can see a model of forklift, it is an automatic guided vehicle that can follow a path painted on the floor, it has tree motors to move the lift, and one motor for steering and one motor with mechanical differential to moving it on the path, in this case the tasks are programmed in the main station using a computer and then the task are transfer to the forklift by using infrared communication between the main computer and the forklift. In the figure 3 an automatic guided vehicle (AGV) with electronic differential is use to follow the path printed on the floor, it has two motors to moving. In the figure 4 there is an explorer who has in front an infrared sensor to seeking an infrared source in an unstructured environment, it use four driving wheels. In these cases is used only one micro-controller and the secondary and coordination tasks are performed in it. But when the system tasks become more complex, like in figure 5 and figure 6, is better if the tasks are divided in secondary and coordination tasks. In the figure 5 is shown a four-legged robot that consist of two micro-controllers, one that has the whole algorithm (coordination task) and another that control the twelve motors (tree in each leg), the communication between micro-controllers is done by the serial port at highspeed. In the figure 6 is shown a teleoperated mechanical hand with 13 degrees of freedom, each motor is servo controlled and one micro-controller makes the coordination tasks, it can communicate with the sensor glove or with the computer using the serial port.

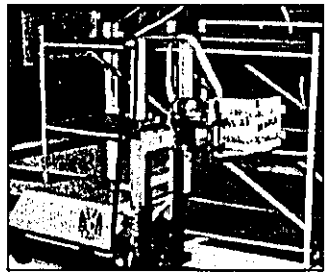


Figure 2: Forklift

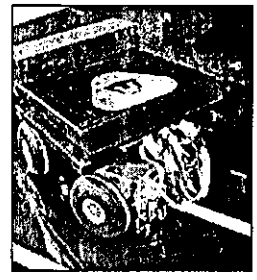


Figure 3: AGV

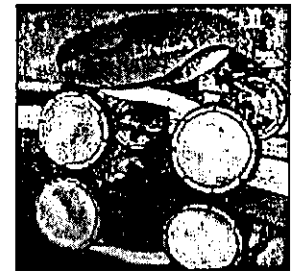


Figure 4: Explorer

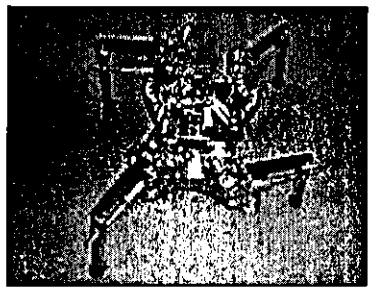


Figure 5: Four-legged robot

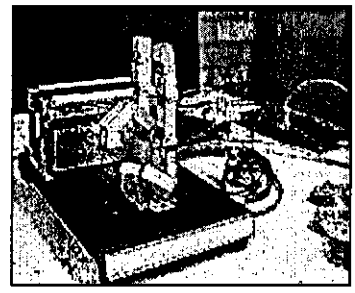


Figure 6: Teleoperated mechanical hand

CONCLUSIONS

This work reports the structure of mechatronic systems from the point of view of their constructive complexity. The paper also presents a generic mechatronics system model base on the General System Theory. The micro-controllers are suited mainly for secondary and coordinating tasks. Finally some design examples based on this model have been explained, but the model can be extended for any design of mechatronic system. The progress and evolution establish the guidelines to decide when and how structure the mechatronic system. The MS model is a good tool for accomplishing mathematical formalism and designing with embedded micro-controllers. With this model the complexity of the mechatronic systems and the application of the embedded micro-controllers will be easer, since the design problem could be divided in subsystems without losing the unity of the whole system.

ACKNOWLEDGMENTS

The author acknowledges to the General Direction of Academic Issues (DGAPA) and the School of Engineering of the National Autonomous University of Mexico (UNAM) for the support received to undertaking PhD studies at Loughborough University, UK.

REFERENCES

1 **Roberts, G.,** Intelligent mechatronics, *Engineering Science and Education Journal,* Volume: 8 Issue: 2, April 1999, p 66 –72.

2 **Tanie, K.,** Mechatronics - past, present and future, *Advanced Intelligent Mechatronics, 1999,* Proceedings of the 1999 IEEE/ASME International Conference on, 1999, 19-23 Sep, 1999, p. 2.

3 **Buur, J.,** *A theoretical approach to mechatronics design,* Lyngby: Institute for Engineering Design, 1990, Technical Unviersity of Denmark. Institute for Engineering Design.

4 **Bradley, D.,** [et al.], *Mechatronics : electronics in products and processes,* London: Chapman and Hall, 1991.

5 **Hanna, D.,** *Designing organizations for high performance,* Reading, Mass.:Addison-Wesley Pub. Co, c1988.

6 **Bertalanffy, L. von.,** *General system theory : foundations, development, applications,* Rev. ed., New York:Braziller, c1968.

7 **Keppens, J.; Shen Q.,** Cambridge University Press, On compositional modelling, *The Knowledge Engineering Review (2001),* Vol: 16:2, 2001, 157-200.

8 **Gonzalez-Villela, V.,** *Metodología de diseño electónico en un proyecto de desarrollo tecnológico,* Maestría en Ingeniería Eléctrica, Universidad Nacional Autónoma de México (UNAM), Facultad de Ingeniería, Feb.-93.

INGENIERÍA MECÁNICA
TECNOLOGÍA Y DESARROLLO

Vol. 1 No. 5 (2004) 159 - 166

A wheeled mobile robot with obstacle avoidance capability

Víctor J. González Villela[1], Robert Parkin[2], Marcelo López Parra[3], Jesús M. Dorador González[4], and M. Jaqueline Guadarrama Liho[5]

[1, 2] Mechatronics Research Centre, Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University. Holywell Building, Holywell Way, Loughborough, Leicestershire, LE11 3UZ, UK.
Tel: + 44 (0) 150 922 6570. Fax: + 44 (0) 150 928 6577.
Email:[1] V.J.Gonzalez-Villela@lboro.ac.uk, [2] R.M.Parkin@lboro.ac.uk.

[1, 4, 5] Departamento de Ingeniería Mecatrónica, Facultad de Ingeniería, UNAM, CU, México.
[3, 4] Departamento de Ingeniería Mecánica, Facultad de Ingeniería, UNAM, CU, México.
Labs. de Ing. Mecánica "Ing. Alberto Camacho Sánchez", Edificio Anexo de la Facultad de Ingeniería, Circuito Exterior, Ciudad Universitaria, UNAM, 04510, México D.F. México.
Tel. + (52 55) 56 22 80 50 y 51. Fax + (52 55) 56 22 80 55.
Email: [1] vjgv@servidor.unam.mx, [3] lopezp@servidor.unam.mx, [4] dorador@servidor.unam.mx, [5] jgl@servidor.unam.mx.

ABSTRACT

The ability of mobile robots to avoid mobile obstacles using artificial potential fields with the only knowledge of the kinematic model of the robot is addressed in this paper. Here, the artificial potential fields have been taken as both a repulsive distance vector (to the obstacle) and an attractive distance vector (to the goal) to compute velocity commands, in order to prove that artificial potential field methods can be applied with the only knowledge of the robot kinematic model. The effectiveness of the method is validated using MatLab and SIMULINK as simulation framework.

RESUMEN

Este artículo trata de la habilidad de esquivar obstáculos en movimiento de un robot móvil usando campos potenciales artificiales con el sólo conocimiento del modelo cinemático del robot. Donde los campos potenciales artificiales han sido tomados como vectores que representan una distancia repulsiva (al objeto) y una distancia atractiva (a la meta) para luego ser utilizados para calcular comandos de velocidad, comprobando que los métodos de campos potenciales artificiales pueden ser aplicados con el sólo conocimiento del modelo cinemático del robot. La eficacia del método es validada usando MatLab y SIMULINK como marco de simulación.

INTRODUCCIÓN

Industrial robots must be able to move about the environment, from one position an orientation (initial posture) to another position and orientation (final posture), without colliding with objects or their environment. The methods to avoid collision can be classified in two types: 1) Path planning techniques, and 2) obstacle avoidance techniques.

Generally speaking, the *path planning techniques* are related to those techniques that are done before the robot moves. These are associated with the fact that any autonomous vehicle must be able to move around without colliding with the most existing installations. Given a well-designed path or trajectory, allowing only one vehicle to be on the path segment, and using the graph-search algorithm A* [1], we can find the shortest route with collision free. On the other hand, the *obstacle avoidance techniques* are associated to those techniques that are done when the robot is moving. The most popular technique used to avoid obstacles is the potential field approach [2]. These techniques are used to guide the robot to a given goal at the same time that the robot is avoiding obstacles. These control methods use the vectorial sum of the virtual repulsive force (due to the obstacle) and the virtual attractive force (due to the goal) to guide the robot [3]. The use of these algorithms assumes the existence of a dynamic model, in which virtual repulsive and attractive forces are modelled as part of the external forces applied to the robot frame to produce acceleration commands [4]. If a kinematic model is available but the dynamic model is unknown, the use of these methods is unfeasible. One way to solve this problem is by taking the potential fields as distance vectors to be used to produce velocity commands, which will guide the robot thru the environment.

Different kinematic models for wheeled mobile robots can be found in the literature, e.g., [3], [5], [6]. These are associated not only with the dimensions of robot and type of wheels configurations, but also with the position of the robot frame reference system and the selection of the steering system. One drawback on the kinematic models of wheeled mobile robots with restricted mobility is that they are subject to the nonholonomic kinematic constraints, which means that they cannot be integrated. As a consequence, coordinates cannot be eliminated by integrating the kinematic equations of constraints [3].

On the other hand, on artificial potential field approaches we can find useful object models, which provide a real-time obstacle avoidance capability to the robot. Some examples are Force Inducting an Artificial Repulsion from the Surface (FIRAS, from the French) functions and Superquadratic functions, some examples can be can be found in [4], and [7].

The wheeled mobile robots have been classified in omnidirectional and restricted mobility robots [5]. Omnidirectional robots are those that have full mobility in the plane. Restricted mobility robots have less than three degrees of freedom in the plane. Moreover, due to the Brockett necessary condition [8], a restricted mobility robot can be controlled only in the same degrees of freedom in the plane as inputs it has to drive the system. This property does not prevent the restricted mobility robots for being controllable, in concordance with the physical intuition [5]. In this case, the potential field approaches are useful to drive the robot from one position to another, and the path planning techniques are one medium of using the physical intuition in order to finally direct a restricted mobility robot into a specific point with a specific orientation. Then, we can join the initial posture and the final posture by dividing the path in strategic consecutive goal points that the robot should reach. In this case the problem is reduced to reach the next goal after reaching the actual one until arriving at the final goal.

In this work is presented a wheeled mobile robot with two independent motorized rear wheels that use a potential field method technique to operate with obstacle avoidance capability, where the potential fields are taken as distance vectors useful for computing velocity commands in order to direct the robot to the goal whilst avoiding obstacles from one initial posture to a final point on the plane, with the only knowledge of the kinematic model. The method is very useful for directing wheeled mobile robots in structured environments with the presence of fixed and moving obstacles, and has been modelled in MatLab and SIMULINK environment to prove the effectiveness of the method. Here, the moving object is taken as a person who is moving in a relative briskly walking speed.

The remaining of this paper is organized in the following way. First, the general posture and the kinematics of wheeled mobile robots are defined, and a particular kinematic model of a wheeled mobile robot with two independent motorized rear wheels is shown. Then, the problems of "Reaching the goal", "Obstacle modelling", and "Avoiding the obstacle" are defined. Finally, the proposed solution, and their simulation results and conclusions are shown.

ROBOT POSTURE

A wheeled mobile robot is equipped with motors that are driven by on board computer. It is assumed that the wheeled mobile robot is made of a rigid frame, non-deformable wheels that do not slip, and it is moving on a horizontal plane. The position of the robot is fully described by the variables x, y, and θ as shown in Figure 1.

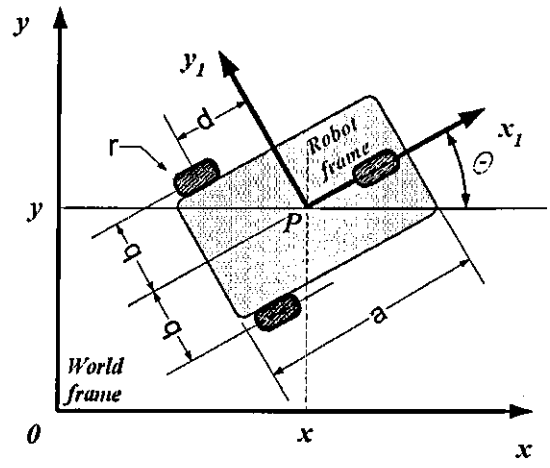


Figure 1: Posture definition

P is a fixed point on the robot's platform, which position is represented by the coordinates (x, y) with respect to the fixed world frame $\{0, x, y\}$ inertial system. The moving robot frame system $\{P, x_1, y_1\}$ is firmly attached to the point P. θ is the orientation angle of the robot frame system $\{P, x_1, y_1\}$ with respect to the world system $\{0, x, y\}$ measured from the x axis to the x_1 axis. The posture of the robot is fully described by the vector $\xi = (x \quad y \quad \theta)^T$. The orthonormal rotation matrix used to map the world frame into the robot frame $R(\theta)$, and vice versa $R^T(\theta)$ is given by:

$$R(\theta) \equiv \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

KINEMATICS OF WHEELED MOBILE ROBOTS

A wheeled mobile robot system having n-dimensional configuration space C with n generalized coordinates $q=(q_1,...,q_n)$ and subject to m bilateral kinematic constraints can be expressed of the following form [9]:

$$A(q)\dot{q}=0 \tag{2}$$

where $A(q)\in R^{m\times n}$ is the matrix associated with the constraints. Let $N(A)$ be the null space of $A(q)$. Then, by spanning $N(A)$ we can find a set of smooth and linearly independent vector fields $s_1(q),...,s_{n-m}(q)$. If we let $S(q)$ be a full rank matrix consisted of this vectors $S(q)=[s_1(q),...,s_{n-m}(q)]$, it is always possible to define $n-m$ input velocities $v(t)=[v_1 \quad v_2 \quad ... \quad v_{n-m}]^T$, where $v(t)\in R^{n-m}$ is called the steering system or the auxiliary velocity vector of the vehicle, such that, for all t

$$\dot{q}=S(q)v(t) \tag{3}$$

where $v(t)$ is the pre-selected input velocity vector for the kinematic model.

KINEMATICS OF A WHEELED MOBILE ROBOT WITH TWO INDEPENDENT MOTORIZED REAR WHEELS

One kinematic model for this type of wheeled mobile robot (Figure 1) can be found in [9], where the front wheel is a free wheel (castor wheel) that does not work for kinematic purposes [5]. The kinematic constraint due to the robot must move in the direction of the symmetry axis can be written as:

$$\dot{y}\cos\theta-\dot{x}\sin\theta-d\dot{\theta}=0 \tag{4}$$

and the rolling constraints due to the driving wheels do not slip can be written as:

$$\begin{aligned}\dot{x}\cos\theta+\dot{y}\sin\theta+b\dot{\theta}&=r\dot{\phi}_r\\ \dot{x}\cos\theta+\dot{y}\sin\theta-b\dot{\theta}&=r\dot{\phi}_l\end{aligned} \tag{5}$$

where $\dot{\phi}_r$ and $\dot{\phi}_l$ are the corresponding angular velocities of the right and left wheel, r is the radius of the wheels, d is the distance from the rear wheels' axle to the point P, b is the positive distance from each rear wheel to the x_1 axis, and $\dot{\theta}$ is the angular velocity of the robot frame.

Defining the generalized coordinates vector as $q=(x \quad y \quad \theta \quad \phi_r \quad \phi_l)^T$ and the generalized velocities vector as $\dot{q}=[\dot{x} \quad \dot{y} \quad \dot{\theta} \quad \dot{\phi}_r \quad \dot{\phi}_l]^T$, we can rewrite the constraints in the form of $A(q)\dot{q}=0$, where:

$$A(q)\dot{q}=\begin{bmatrix}-\sin\theta & \cos\theta & -d & 0 & 0\\ -\cos\theta & -\sin\theta & -b & r & 0\\ -\cos\theta & -\sin\theta & b & 0 & r\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{\phi}_r\\ \dot{\phi}_l\end{bmatrix} \tag{6}$$

Then, instead of finding a solution for $S(q)$ to the system given in (6) by selecting the velocities of the rear wheels as the steering system $v(t)=[\dot{\phi}_r \quad \dot{\phi}_l]^T$, as in [9], we can find a set of smooth and linear independent vector fields for $S(q)$ that span the null space of $A(q)$ when it is given the $n-m$ steering system $v(t)=[v_1 \quad v_2]^T=[\dot{x}_1 \quad \dot{\theta}]^T=[v \quad \omega]^T$, where $v_1=\dot{x}_1=v$ is the heading linear velocity of the robot at the point P and $v_2=\dot{\theta}=\omega$ is the angular velocity of the robot frame, then $\dot{q}=S(q)v(t)$ can be written as:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{\phi}_r\\ \dot{\phi}_l\end{bmatrix}=\begin{bmatrix}\cos\theta & -d\sin\theta\\ \sin\theta & d\cos\theta\\ 0 & 1\\ 1/r & b/r\\ 1/r & -b/r\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \tag{7}$$

With (7) we can compute the whole velocities at the work space $[\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$ and at the joint space $[\dot{\phi}_r \quad \dot{\phi}_l]^T$ due to the heading linear velocity v of the point P and angular velocity w of the robot frame, where the steering system $v(t)=[v \quad w]^T$ is the input of the kinematic model.

THE PROBLEM DEFINITION

Since a two independent motorized rear wheel mobile robot is defined as a restricted mobility robot [5] that has only two inputs, we only can control two variables on the plane (as imposed by the Brockett necessary condition [8]). In our case we will use (x_f, y_f) to define the final position of the robot, and then leave the final orientation of the robot to be reached to the path planning techniques. Then, in this paper, the problem to direct the robot from an initial posture to a final position (goal) while avoiding a mobile obstacle is defined as: given a specific posture (x_i, y_i, θ_i) reach the position (x_f, y_f) while avoiding moving obstacles. This problem can be described for the following aspects.

REACHING THE GOAL

The problem of reaching the goal (Figure 2) is defined as the direction and velocity in which the robot should be directed in order to reach the goal. First of all, let us define the distance vector to the goal as $d_g = (x_f - x_i, y_f - y_i)$, where the distance magnitude to the goal is defined as $d_g = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$, and the angle to the goal as $\theta_g = \tan^{-1}[(y_f - y_i)/(x_f - x_i)]$, where the error angle to the goal is defined as $\theta_e = \theta_g - \theta_i$. Then, the robot is directed from initial posture $\xi_i = [x_i \quad y_i \quad \theta_i]$ to the final position using the following rules.

$$v = \begin{cases} v_{\max} & if \quad |d_g| > k_r \\ \dfrac{v_{\max}}{k_r} \cdot d_g & if \quad |d_g| \le k_r \end{cases} \qquad (8)$$

where $v_{\max}$ is the maximum heading velocity and k_r is the radius of the docking area.

$$\omega = \omega_{\max} \sin(\theta_e) \qquad (9)$$

where $\omega_{\max}$ is maximum angular velocity of the robot platform.

With these rules (8) and (9), the robot is directed to the final position at the maximum linear velocity when it is outside of the docking area, and it will be approaching slowing down when it is inside the docking area. On the other hand, the robot will be correcting the heading angle as function of the sine of the error angle θ_e, where the maximum angular velocity $\omega = \pm\omega_{\max}$ will be achieved at $\theta_e = \pm 90^0$, and the minimum angular velocity $\omega = 0$ will be at $\theta_e = 0^0$.

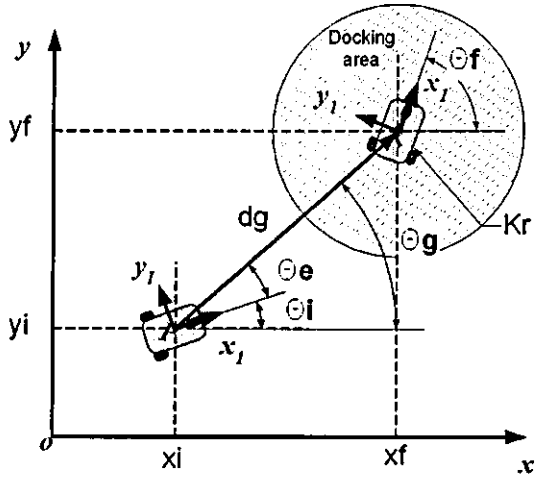


Figure 2: Problem definition of reaching the goal

OBSTACLE MODELLING

In this work the modelling of obstacles has been taken from [4], where the artificial potential field U_0 is taken as a repulsive distance vector. As a consequence we can use this artificial potential function with the only knowledge of the kinematic model. Then, using the shortest distance to the obstacle O, the potential function can be written as:

$$U_0 = \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{\rho} - \dfrac{1}{\rho_0}\right)^2 & if \quad \rho \le \rho_0 \\ 0 & if \quad \rho > \rho_0 \end{cases} \qquad (10)$$

where η is a constant gain, ρ_0 represents the limit distance of the potential field influence, and ρ the shortest distance from the robot to the obstacle O.

The selection of η and ρ_0 depends on the mobile robot $v_{\max}$ and $\omega_{\max}$ velocities that allows the robot to be able to avoid obstacles. When the obstacles are taken as circular obstacles with radius ρ_r, then, the distance ρ can be computed as the distance to the centre of the object ρ_c minus ρ_r. This way, the potential function can be rewriting as:

$$U_0 = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho_c-\rho_r}-\frac{1}{\rho_0}\right)^2 & if \quad \rho_c \le \rho_0+\rho_r \\ 0 & if \quad \rho_c > \rho_0+\rho_r \end{cases} \quad (11)$$

AVOIDING THE OBSTACLE

Avoiding objects while the robot is reaching a goal using artificial potential field methods has been defined as control method which uses the vectorial sum of virtual repulsive forces (due to the objects) and virtual attractive forces (due to the goals) to provide a resultant force to guide the vehicle, as shown in the Figure 3. In this case, again the use of the forces is unfeasible when the dynamical model is unknown; as consequence, in this work we assume that those potential forces are repulsive distance vectors, which can be used to produce velocity commands useful for directing the robot.

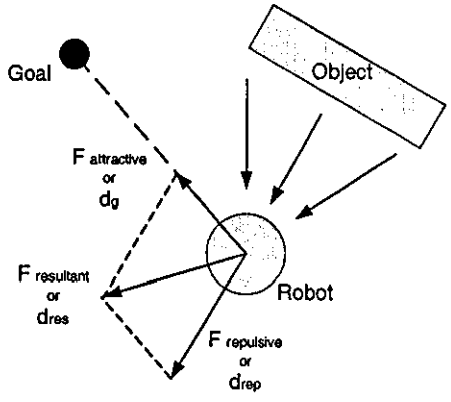


Figure 3: Potential-Field control approach

PROPOSED SOLUTION

The kinematic model of the wheeled mobile robot (7) can be used to find the actual position and orientation of the robot (x_i, y_i, θ_i) by integrating the first three generalized speeds $[\dot{x} \quad \dot{y} \quad \dot{\theta}]^T$. The final two generalized speeds $[\dot{\phi}_r \quad \dot{\phi}_l]^T$ can be used to produce the required velocity commands for the motors. In this way, we can compute at any instant time the actual robot posture (x_i, y_i, θ_i) and with the knowledge of the final position (x_f, y_f) compute the distance vector to the goal d_g. Then, in presence of obstacles, we can compute the distance resultant vector $d_{res} = d_g + d_{rep}$ as the total vector sum of repulsive distances due to objects d_{rep} and the distance vector to the goal . Besides, we can compute a new avoid angle error defined as , where is the angle of the vector, and is the steering velocity commands which direct the robot into the goal while avoiding obstacles. Finally, the new rules can be written as:

$$v = \begin{cases} v_{\max} & if \quad |d_{res}| > k_r \\ \frac{v_{\max}}{k_r} \cdot d_{res} & if \quad |d_{res}| \le k_r \end{cases} \quad (12)$$

$$\omega = \omega_{\max} \sin(\theta_{avoid,e}) \quad (13)$$

We can see that in absence of obstacles $d_{res} = d_g$, so that, the robot behaves in normal conditions. In presence of obstacles the new component d_{rep} will direct the robot far from the obstacle.

SIMULATION RESULTS

In the Figure 6 you will find the Simulik block diagram of the problem. The block I represents the kinematic of the robot associated to the plane, and corresponds to the first three generalized speeds of the equation (7), it includes the input initial conditions and the integration of the output variables $(\dot{x}, \dot{y}, \dot{\theta})$ to compute the vector (x, y, θ). The block II represents the problem definition of reaching the goal as shown in Figure 2. It computes the distance to the goal vector d_g, and normalize the θ_i as an angle defined between $-\pi$ and π. The block III computes the repulsive distance vector d_{rep} due to a moving rounded object as shown in (11) by taking the actual position of the robot (x, y), the actual position of the object (x_0, y_0), and the corresponding object modelling parameters η, ρ_r, and ρ_0. The block IV computes the potential field resultant distance d_{res} and the avoid angle error $\theta_{aviod,e}$. The block V computes the velocity commands (v, ω) according to the equations (12) and (13). The bock VI computes the robot wheels velocity commands $\dot{\phi}_r$ and $\dot{\phi}_l$ according to the second part of the kinematic model show in (7). The block VII is a S-function, which is the graphic simulation routine.

In this case we are assuming that we have a wheeled mobile to robot with the following dimensions: $b = 0.50m$, $a = 1.00P$, $d = 0.4m$, and $r = 0.075P$. That can move at $v_{max} = 1\ P/s$ and at $|\omega_{max}| = \pi/5\ \ 1/s$. The object has been modelled as a man with radius $\rho_r = 1/2\ \ m$, who is walking about the environment a relative briskly walking speed of $1.5\ P/s$, and with a field influence of $\rho_0 = 3\ \ m$. In this case the robot should reach the point $(x_f, y_f) = (10m, 10m)$ departing from the posture $(x_i, y_i, \theta_i) = (-10\,m, -10\,m, \frac{3}{2}\pi \cdot rad)$, while avoiding a moving obstacle. The Figure 5 shows the simulation results, where graphs (a) and (b) shows position of the robot (x, y) respectively. The graphs (c) and (d) represents the heading linear velocity and the angular velocity to the robot frame. And final two graphs (e) and (f) represents the angular velocity of the right wheel and the angular velocity of the left wheel. The Figure 4 shows the sequence of the simulation.

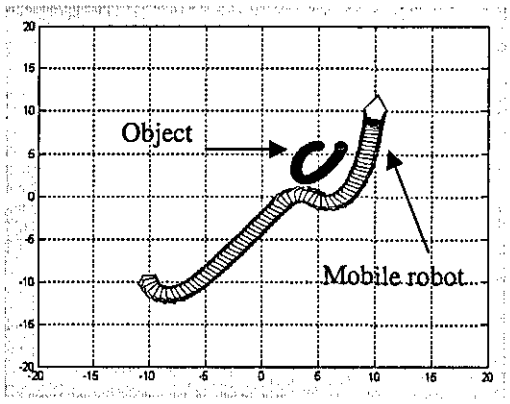


Figure 4: Graphic simulation sequence

CONCLUSIONS

Here has been proved that the use of artificial potential field methods can be useful for directing a restricted mobile robot from one posture into a position while it is avoiding obstacles, with the only knowledge of the robot kinematic model by taking the potential fields as a distance vectors in order to compute the velocity commands, as shown in the control rules (12) and (13). Selecting the steering system as $v(t) = [v \quad \omega]^T$ has been crucial for determining these control rules. The use of potential functions to model the objects has proved their efficacy. Besides, joining the initial posture and the final posture by dividing the path in strategic consecutive goal points can be a solution for the control of the robot trajectory and the final docking angle. In this case the problem is reduced to reach the next goal after reaching the actual one until arriving at the final goal. The method presented in this paper is very useful for directing wheeled mobile robots, which moves relatively slowly (e.g., forklifts), in structured environments with the presence of fixed or/and moving obstacles.

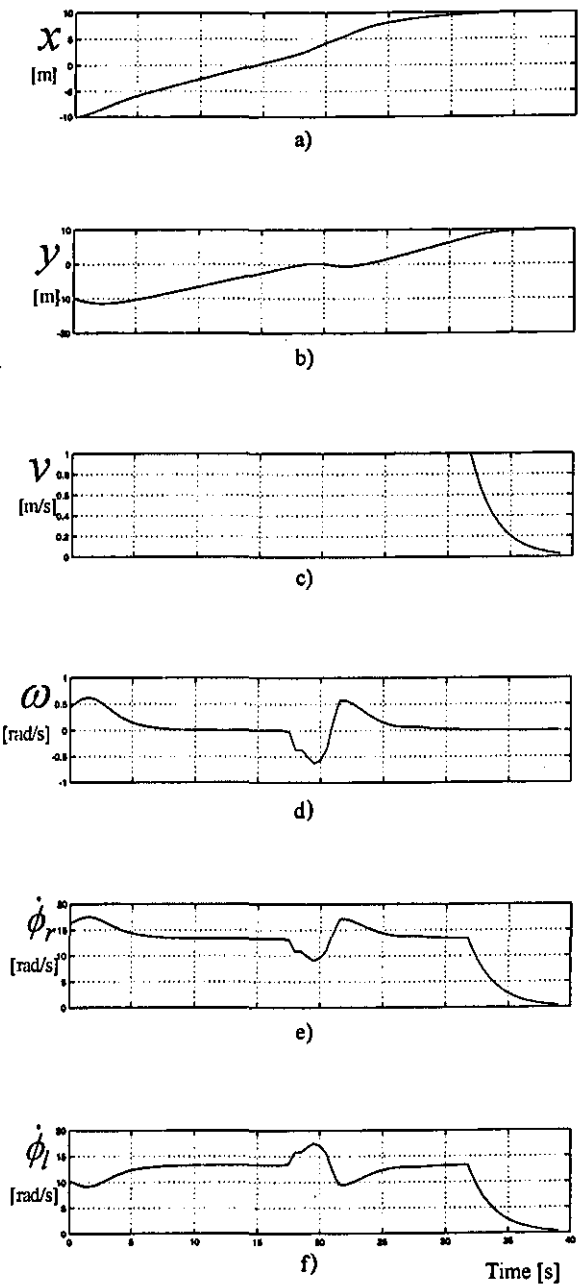


Figure 5: Simulation results

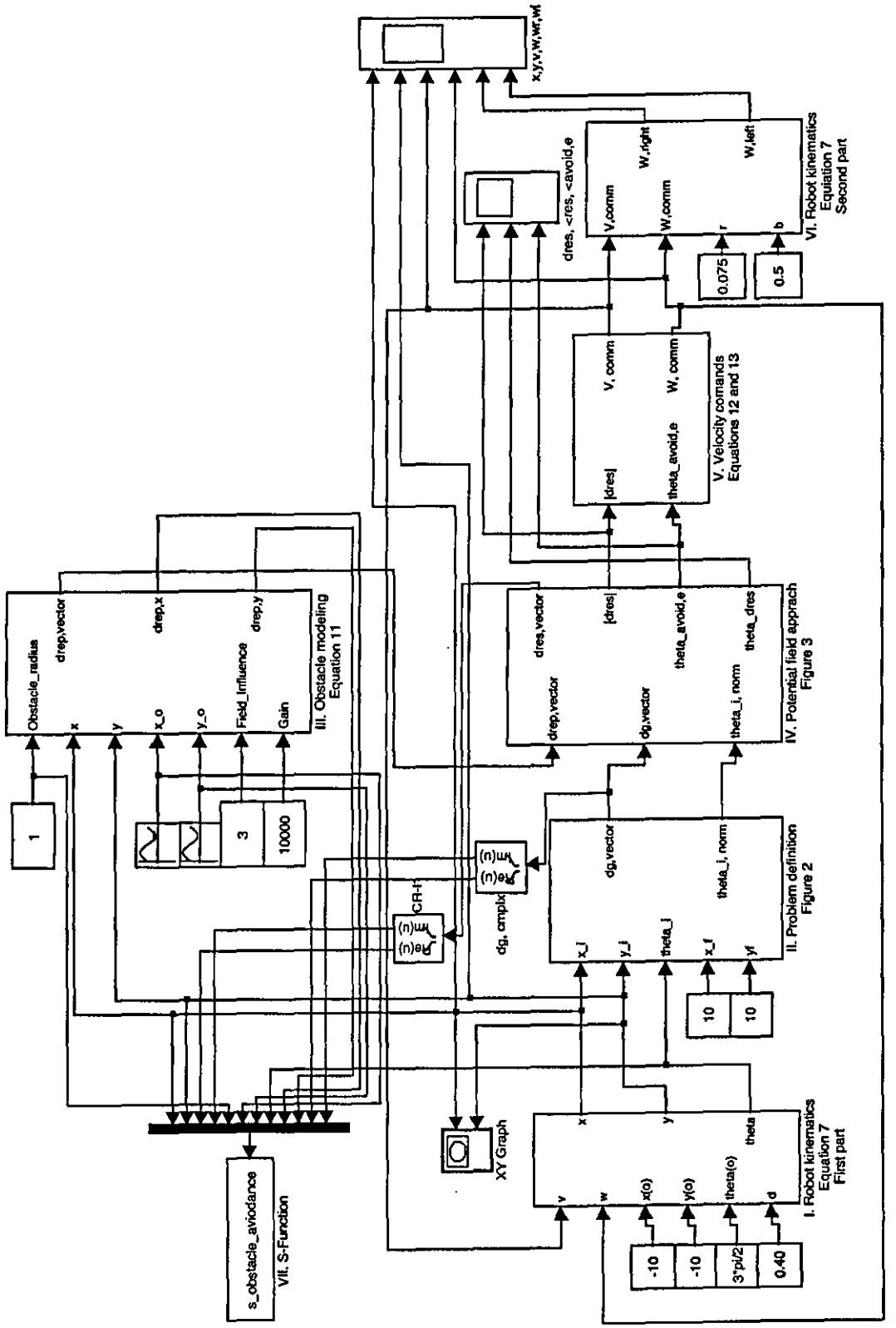


Figure 6: SIMULINK block diagram

ACKNOWLEDGEMENTS

The first author would like to acknowledge to the "Dirección General de Asuntos del Personal Académico" (DGAPA) and the "Facultad de Ingeniería" of the "Universidad Nacional Autónoma de México" (UNAM) for the support received to undertake PhD studies at the Mechatronics Research Centre, Loughborough University, UK.

REFERENCES

1. Cameron, S., "Obstacle avoidance and path planning". *Industrial Robot: An International Journal.* Vol: **21**(5): p. 9-14, 1994.

2. Latombe, J.C., *Robot motion planning.* Boston: Kluwer Academic Publishers. 1991.

3. Zhao, Y. and BeMent, S.L. *Kinematics, dynamics and control of wheeled mobile robots.* Robotics and Automation, Proceedings of the IEEE International Conference on. Nice. Vol: **1**: p. 91 - 96, 1992.

4. Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots". *qhe International Journal of Robotics Research.* Vol: **5**(1): p. 90-98, 1986.

5. Campion, G., Bastin, G., and d'Andrea-Novel, B., "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots". *Robotics and Automation, IEEE qransactions on.* Vol: **12**(1): p. 47-62, 1996.

6. Thuilot, B., d'Aandrea-Novel, B., and Micaelli, A., "Modeling and feedback control of mobile robots equipped with several steering wheels". *Robotics and Automation, IEEE qransactions on.* Vol: **12**(3): p. 375 - 390, 1996.

7. Khosla, P. and Volpe, R. *Superquadric artificial potentials for obstacle avoidance and approach.* Robotics and Automation, Proceedings of the IEEE International Conference on. Philadelphia, PA. Vol: **3**: p. 1380 - 1385, 1988.

8. Brockett, R.W. *Asymptotic stability and feedback stabilization.* Differential geometric control theory, Proceedings of the conference. Michigan Technological University: Boston, Mass. : Birkhauser: p. 181-191, 1993.

9. Yamamoto, Y. and Yun, X., *Coordinating locomotion and manipulation of a mobile manipulator,* in *Recent trends in mobile robots,* Y.F. Zheng, Editor. 1993, World Scientific: Singapore ; London. p. 157-181.

Revista Digital Universitaria
18 de enero 2005 • Volumen 6 Número 1 • ISSN: 1067-6079

EVADIENDO OBSTÁCULOS CON ROBOTS MÓVILES

Mtro. Víctor Javier González Villela
Profesor de Carrera Titular "A" del Departamento de Mecatrónica de la Facultad de Ingeniería, Universidad Nacional Autónoma de México
vjgv@servidor.unam.mx

Dr. Robert M. Parkin
Profesor of Mechatronics PhD, Mechatronics Research Centre, Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University
R.M.Parkin@lboro.ac.uk

http://www.revista.unam.mx/vol.6/num1/art02/art02.htm

EVADIENDO OBSTÁCULOS CON ROBOTS MÓVILES

RESUMEN

En este trabajo se presenta una síntesis de los métodos de anticolisión que se utilizan para evadir obstáculos con robots móviles, en particular, los métodos de planeación de rutas y métodos basados en campos potenciales artificiales.

Palabras clave: Robots móviles, Planeación de rutas, Evasión de obstáculos, Campos potenciales artificiales.

AVOIDING OBSTACLES WITH MOBILE ROBOTS

ABSTRACT

Here is presented a synthesis of anti-collision methods that are used to avoid obstacles with mobile robots, in particular, methods of path planning and methods based on artificial potential fields.

Keywords: Mobile robots, Path planning, Obstacle avoidance, Artificial potential fields.

Revista Digital Universitaria
18 de enero 2005 • Volumen 6 Número 1 • ISSN: 1067-6079

INTRODUCCIÓN

Desde que se crearon los robots industriales, se hizo evidente lo fácil que estos pueden llegar a chocar con su medio ambiente, o entre ellos mismos. Desde entonces, la búsqueda de métodos de anticolisión ha demandado la atención de los investigadores. En el primer caso están los métodos de planeación de rutas, y en el segundo caso, los métodos de evasión de obstáculos. Los métodos de planeación de rutas consisten en encontrar en forma automática las rutas libres de colisión, mientras que los métodos de evasión de obstáculos consisten en evadir los obstáculos que se presentan en la trayectoria del robot, mientras éste consigue alcanzar la meta.

PLANEACIÓN DE RUTAS

Cualquier vehículo autónomo debe poderse mover de un lado a otro sin chocar con los obstáculos que se encuentran en su ambiente de trabajo. Normalmente estos vehículos tienen definidos ciertos caminos por donde pueden moverse libremente. Por ejemplo, estos caminos libres de colisión están representados por las líneas punteadas en la Figura 1.a, mismas que definen el mapa de trayectorias.

Inicialmente, la planeación de rutas se logra traduciendo el mapa de trayectorias (Figura 1.a) en gráficos (Figura 1.b), donde las líneas continuas contienen alguna información relacionada con las trayectorias (por ejemplo, la distancia existente entre nodos). Por otro lado, los nodos (puntos verdes) representan la unión de trayectorias o cambios de dirección (Cameron 1994). Entonces, la ruta más corta puede encontrarse utilizando un "algoritmo de búsqueda gráfica estándar con respaldo de información" como puede ser el algoritmo A*, que hace uso de la información de la gráfica para calcular el costo de la ruta que va del nodo de partida al nodo *n* y el costo estimado de la ruta más barata que va del nodo *n* a la meta. La suma de estas dos cantidades proporciona el costo estimado de la solución más barata que pasa por el nodo *n*, de acuerdo a un criterio seleccionado. En este ejemplo, el criterio podría ser la distancia entre nodos, que es el único dato por ahora proporcionado. La explicación de este método y sus derivados puede encontrarse en (Russell et al. 2003). Además de todo esto, en un sistema multivehicular, es necesario tomar en cuenta otras consideraciones, tales como: estimar con precisión los tiempos de travesía, resolver los problemas de los cruces y los giros apropiadamente, programar un ambiente multivehicular, y enfrentar lo inesperado con rapidez.

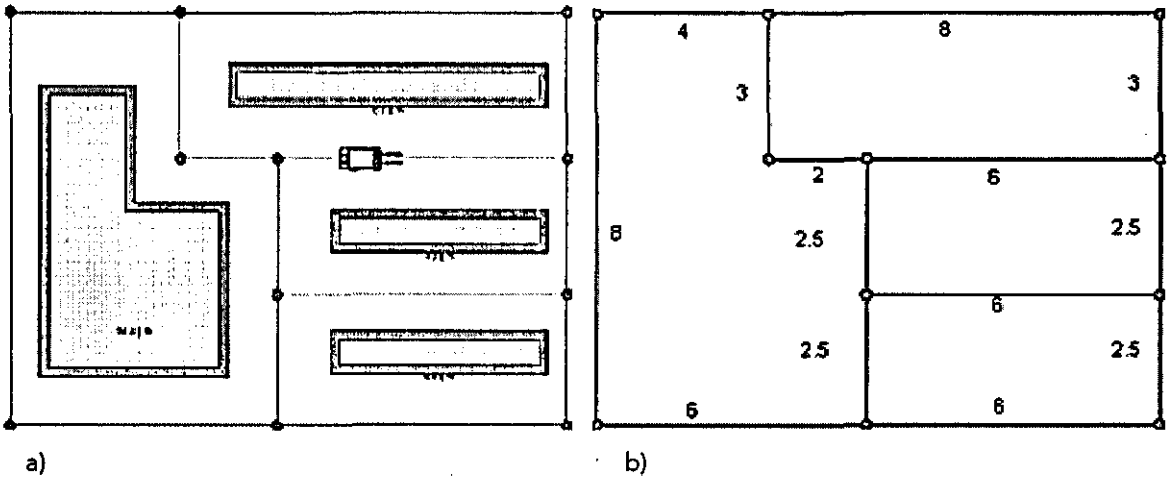


Figura 1: a) Espacio de configuración y trazo de rutas y b) Obtención de gráficas

La planeación de rutas se complica aun más cuando es necesario tomar en cuenta la forma del robot, debido al hecho de que todas las partes del robot deben evitar colisionar simultáneamente con el ambiente. En este último caso, el concepto de espacio de configuración nos permite separar matemáticamente los cuidados del calcular que parte de los objetos podrían chocar, de los cuidados de obtener los posibles caminos para alcanzar la meta. El espacio de configuración puede llegar a ser muy complejo, en otros puede resumirse a proyecciones de dos dimensiones como lo mostrado en la Figura 1.a.

En el caso más simple, los vehículos pueden ser considerados redondos, reduciendo el problema de "encontrar el camino que mantenga a todas las partes del robot lejos de los obstáculos" a "encontrar el camino que mantenga el centro del robot, a una distancia mayor que el radio del círculo que modela al robot". Entonces, es posible construir un nuevo mapa del ambiente donde se pueda ubicar al robot libre de colisiones haciendo crecer gráficamente los obstáculos, estas áreas pueden estar representadas de una manera burda por los contornos grises en la Figura 1.a. Una vez que está definido el espacio de configuración, es posible trazar trayectorias en ese espacio y usar, por ejemplo, el algoritmo A*.

En general, los métodos que han sido utilizados para encontrar este camino libre de colisiones, son los llamados métodos de Mapa de Caminos (Métodos de Visibilidad Gráfica, Diagramas de Voronoi, Freeway, y Silhouette), y los métodos de Descomposición de Celdas (Descomposición en Celdas Exactas, y Descomposición en Celdas Aproximadas), la explicación de estos métodos puede encontrarse en (Latombe 1991).

Sin embargo, el uso de estos métodos puede complicarse cuando el modelo del ambiente de trabajo y del robot es tridimensional, si el ambiente de trabajo cambia debido a la presencia de obstáculos inesperados, si el robot tiene una forma diferente a la circular, o cambia sus dimensiones debido a la presencia de mecanismos tales como manipuladores que el mismo robot utiliza para realizar tareas en su medio ambiente. En este caso, la introducción de métodos basados en el concepto de campos potenciales artificiales resulta más práctico.

EVASIÓN DE OBSTÁCULOS

Los algoritmos de planeación exactos, tales como los discutidos hasta ahora, se dedican a tratar de encontrar soluciones óptimas para problemas específicos, el concepto de optimización se reduce a optimizar tiempos, energía, distancia, costo, etc. Pero si estos métodos se desean aplicar a problemas prácticos, el concepto de optimización es mucho más vago. Por ejemplo, el vehículo debe de moverse rápido, interactuar con personas, alargar el tiempo de recarga de baterías, resolver rápidamente el camino a tomar, etc. Aun más, el problema se complica cuando se tienen muchas posibles rutas, y es entonces cuando los modelos de campos potenciales son usados para resolver el problema. Como un ejemplo de campos potenciales (ver Figura 2), imaginemos que tenemos un espacio de configuración donde los objetos están definidos por polígonos irregulares, dibujemos esos polígonos en una manta y hagamos que se levanten, poniendo objetos por debajo de estos polígonos, formando una superficie con altas (objetos) y bajas (trayectorias libres de colisión o la meta). Definamos un punto inicial y un punto final sobre la manta. Levantemos la manta lo necesario para que el punto inicial sea el más alto y el punto final sea el más bajo. Entonces, pongamos un balín en el punto inicial y observemos como el balín es atraído al punto más bajo (meta), evitando los obstáculos. De esta manera la manta funciona como la fuerza repulsiva que mantiene al balín lejos del los obstáculos y al mismo tiempo guía al robot al punto final.

18 de enero 2005 • Volumen 6 Número 1 • ISSN: 1067-6079

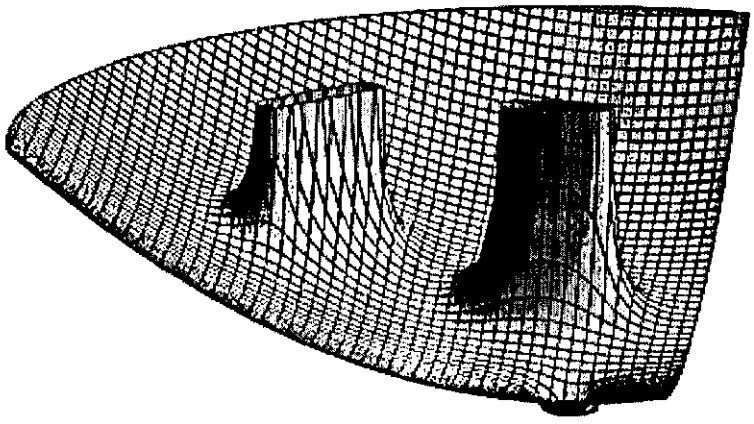


Figura 2. Ejemplo de un campo potencial artificial representado por dos obstáculos y la meta

Modelar estos campos potenciales no es fácil, sin embargo, es posible modelar las fuerzas de repulsión debidas a los objetos, con una función inversamente proporcional a la distancia a la que el robot se encuentra de ellos como lo mostrado en la Figura 3. Entonces, la fuerza repulsiva (suma total de las fuerzas de repulsión debidas a los objetos en el punto donde se encuentra el robot) sumada a la fuerza atractiva (fuerza asociada a la meta) produce una fuerza resultante. Esta fuerza resultante varía de acuerdo a la posición del robot y lo guía durante todo su trayecto, desde el punto inicial hasta el punto final. Un problema grande, en este método, es que pueden encontrarse hondonadas en el camino, como resultado de la ausencia de fuerzas actuantes en el robot, debido a la anulación de la fuerza resultante, ver por ejemplo (Mora Aguilar et al. 2004). Esto ocurre cuando la fuerza de atracción aplicada es de la misma magnitud a la fuerza de repulsión, pero exactamente en el sentido contrario a ésta, por lo que la suma vectorial es cero. Este problema es llamado mínimo local, y resulta en un estancamiento de robot en un punto no deseado. Algunos ejemplos de la construcción de estos campos, su problemática, y algunas aplicaciones interesantes puede encontrarse en (Khatib 1986), (González-Villela et al. 2004), y en la bibliografía reportada.

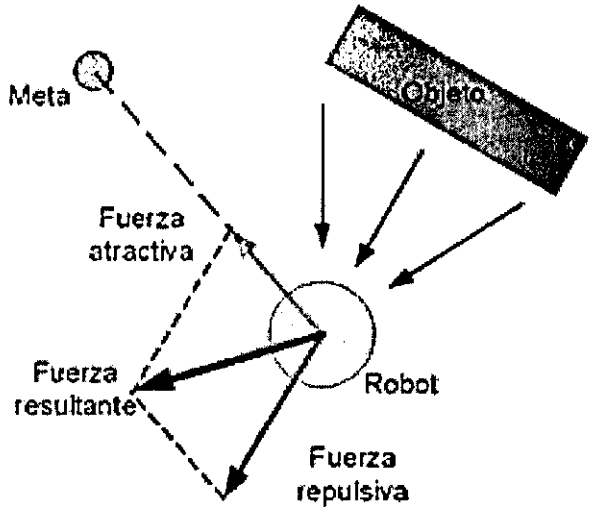


Figura 3. Modelado de fuerzas

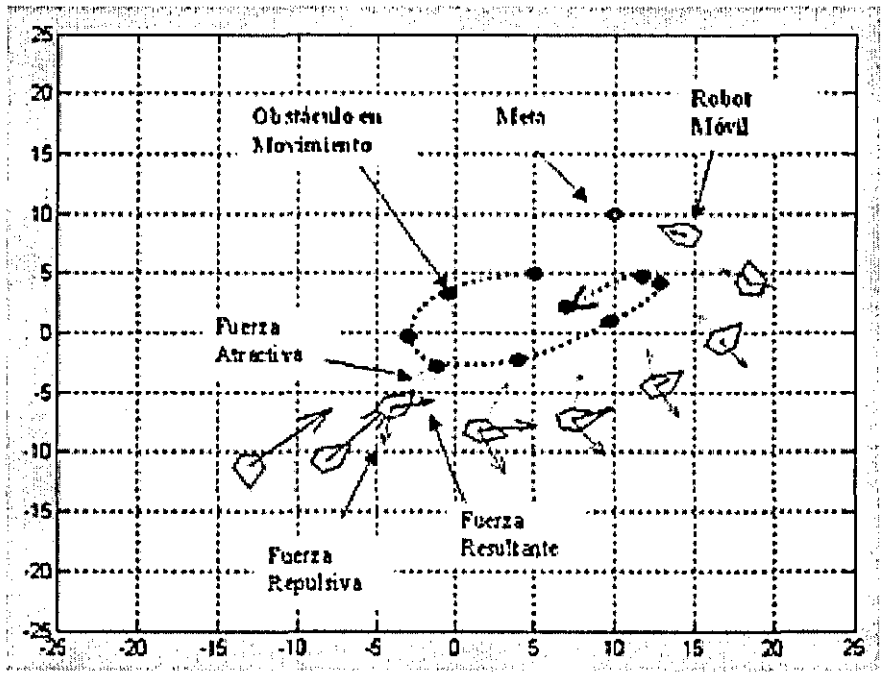
En general, estos métodos están limitados por la confiabilidad de los sensores que utilizan, por los modelos imprecisos que se tienen de los mecanismos del robot y por las características no ideales de las llantas que, hasta cierto punto, permiten el deslizamiento. Entonces, la idea básica para la utilización de estos algoritmos está constituida, y limitada al mismo tiempo, por estos tres puntos básicos (Cameron 1994):

- El método básico es restringido a robots representados por un punto y todo el campo potencial se construye alrededor del análisis de ese punto, por lo que no es aplicable a todos los tipos de robot.

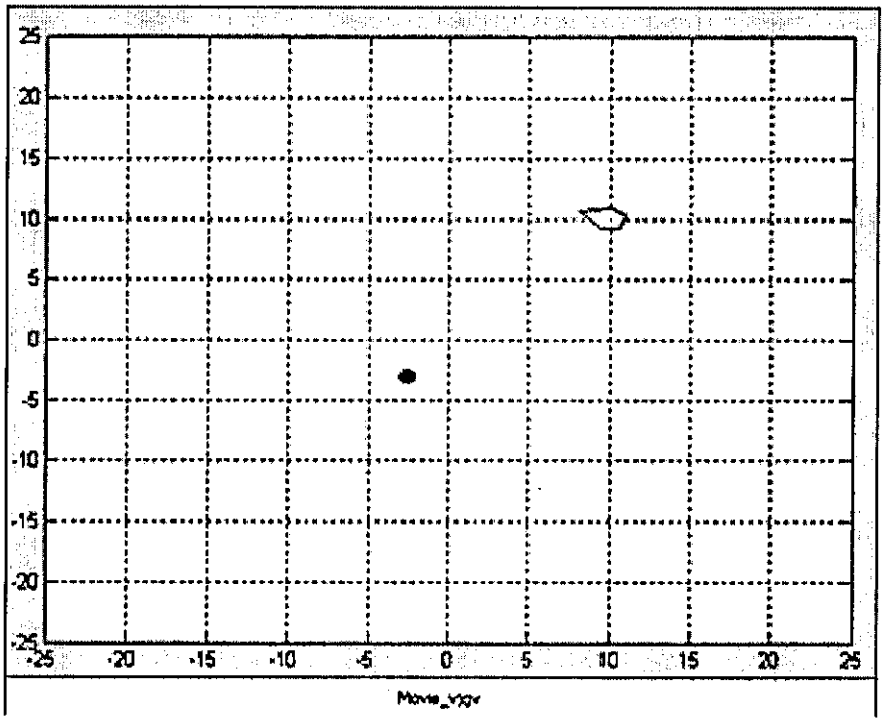
- Es difícil de evitar los mínimos locales en el campo potencial producidos por objetos cercanos al robot.

- La mayoría de los vehículos son de movilidad restringida (noholonómicos), que para cambiar su orientación deben moverse de su posición actual.

Un ejemplo de la utilización de estos métodos puede verse en la Figura 4, esta figura presenta una simulación hecha en SIMULINK de MatLab, donde se puede observar un robot móvil evadiendo un obstáculo en movimiento, éste se dirige a la meta guiado por la fuerza resultante, consecuencia de la suma vectorial de la fuerza de atracción (debida a la meta) y la fuerza de (repulsión debida al obstáculo en movimiento).



a) Secuencia gráfica



b)Película

Figura 4. Evasión de obstáculos en movimiento, simulación gráfica en SIMULINK:

DISCUSIÓN

Al tomar en cuenta los dos métodos descritos anteriormente, se puede decir que los métodos de planeación de rutas se utilizan antes de que el robot se mueva, y los métodos de evasión de obstáculos se utilizan cuando el robot está en movimiento. Además, el gradiente del potencial (fuerza resultante) que experimenta el robot en los métodos de campos potenciales está constituido de dos componentes: fuerza de repulsión generada por los obstáculos y fuerza de atracción generada por la meta. Sin embargo, estas fuerzas vectoriales no son reales y no existen en el medio ambiente real del robot, éstas son añadidas a los modelos matemáticos del robot para generar los comandos de aceleración en las ruedas del robot, son función de la distancia a la que se encuentra el robot de la meta (para la fuerza atractiva) y de la distancia a la que se encuentra de los obstáculos (fuerza repulsiva), es por eso que estos métodos se ha hecho llamar de Campos Potenciales Artificiales. Además, los datos de las distancias provienen de los sensores externos e internos del robot que le permiten saber donde se encuentra, a que distancia está de la meta y de los objetos de su alrededor.

También, la planeación de rutas, presupone la existencia de un modelo estable y confiable del medio ambiente, que por el contrario, en los métodos de evasión de obstáculos no es necesario. Es por eso que las técnicas de planeación de rutas y de evasión de obstáculos son técnicas que son utilizadas con diferente propósito. El primero está asociado al comportamiento deliberativo (preplaneado) y el segundo al comportamiento reactivo (responder a los estímulos provenientes del exterior), ver por ejemplo (Chatila 1995). Así que, los métodos de planeación de rutas están relacionados con los algoritmos de planeación y los métodos de evasión de obstáculos con la toma de decisiones del tipo sensar-reaccionar.

CONCLUSIÓN

Para contar con un sistema de evasión de obstáculos robusto es necesario balancear los dos comportamientos, el deliberativo y el reactivo, de tal manera que el robot móvil cuente con la capacidad de planear trayectorias óptimas libres de colisión y reaccionar ante la presencia de objetos inesperados en el camino. La aplicación de estas dos técnicas en un sólo robot da como resultado la creación de sistemas híbridos, los cuales usan las dos técnicas, de planeación de rutas y de evasión de obstáculos, para superar los problemas que se presentan en el transcurso de la conquista de la meta. En particular para encontrar un camino libre de colisión como en la Figura 1 y para evadir obstáculos que se presenten en el camino como en la Figura 3. De esta manera, el robot puede planear con anticipación la ruta que tomará, de acuerdo a un criterio de optimización (distancia, energía, tiempo, etc.), para luego dividir esa ruta específica en tramos constituidos por un conjunto de submetas consecutivas, que deberán ser alcanzadas una tras otra por el robot mientras utiliza un algoritmo basado en campos potenciales artificiales como se muestra en la Figura 4, de esta manera, el robot podrá seguir la trayectoria preplaneada y reaccionar rápidamente a los cambios que se presentan en su ambiente debido a la presencia de obstáculos inesperados.

AGRADECIMIENTOS

El primer autor desea agradecer a la Dirección General de Asuntos del Personal Académico (DGAPA) y a la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM) por el soporte recibido para realizar estudios de doctorado en el Holywell Mechatronics Research Centre, Loughborough University, Inglaterra.

18 de enero 2005 • Volumen 6 Número 1 • ISSN: 1067-6079

REFERENCIAS Y BIBLIOGRAFÍA

BECKHAUSS, Steffi; Felix Ritter y Thomas Strothotte. "Guided exploration with dynamic potential fields: the cubicalpath systems" [en línea] Fraunhofer Institut Medienkommunikation, 2004. http://viswiz.gmd.de/~steffi/Publications/CGJCubicalPath.pdf [Consulta: 9 diciembre 2004]

CAMERON, S. (1994). "Obstacle avoidance and path planning." *Industrial Robot: An International Journal* Vol: 21(5): 9-14.

CHATILA, R. (1995). "Deliberation and reactivity in autonomous mobile robots." *Robotics and Autonomous Systems* Vol: 16(2-4): 197-211.

GERDES, J. Christian and Eric J. Rossetter. "A unified approach to driver assistance systems based on artificial potential fields" [en línea]. Proceedings of ASME International Mechanical Engineering Congress and Exposition November 14-19, 1999, Nashville, TN, USA. http://wwwcdr.stanford.edu/dynamic/PF/papers/pfieldsIMECE99.pdf [Consulta: 9 diciembre 2004]

GONZÁLEZ-VILLELA, V. J., Parkin, R. M., Lopez-Parra, M., Dorador-González, J. M., et al. (2004). "A wheeled mobile robot with obstacle avoidance capability." *Ingeniería Mecánica Tecnología y Desarrollo. Revista de la Sociedad Mexicana de Ingeniería Mecánica (SOMIM).* ISSN 1665-7381. Vol: 1(5): 159-166.

KHATIB, O. (1986). "Real-time obstacle avoidance for manipulators and mobile robots." *The International Journal of Robotics Research* Vol: 5(1): 90-98.

LATOMBE, J. C. (1991). *Robot motion planning.* Boston, Kluwer Academic Publishers.

LEARNING AND VISION MOBILE ROBOTICS GROUP [en línea]. [Barcelona, España]: Institut de Robótica i Informática Industrial, 2004. http://www-iri.upc.es/groups/lrobots [Consulta: 9 diciembre 2004]

MORA AGUILAR, M. C., Armesto Ángel, L. and Tornero Montserrat, J. (2004). *Sistema de Navegación de Robots Móviles en Entornos Industriales,* XXV Jornadas de Automatica. http://www.isa-cr.uclm.es/xxvjornadas/ConfMan_1.7/SUBMISSIONS/115-arlarpvmor.pdf. Diciembre 9, 2004.

RUSELL, S. J. and Norvig, P. (2003). *Artificial intelligence: a modern approach.* Upper Saddle River, N.J., Prentice Hall.

PRAHLAD, Vadakeppat; Kay Chen Tan y Wang Ming-Liang. "Evolutionary artificial potential fields and their application in real time robot path planning" [en línea]. Documento presentado en el IEEE Congress on Evolutionary Computation 2000, 2000, La Jolla, CA, United States. Department of Electrical Engineering, The National University of Singapur http://www.ee.nus.edu.sg/stfpage/elepv/publication/cec2000.pdf [Consulta: 9 diciembre 2004]

DYNAMIC TORQUE CONTROL EXTENSION FOR A WHEELED MOBILE ROBOT WITH OBSTACLE AVOIDANCE CAPABILITY

Víctor J. González Villela[1], Robert M. Parkin[2],
Marcelo López Parra[3], and Jesús M. Dorador González[4].

[1, 2] Mechatronics Research Centre, Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University.
Holywell Building, Holywell Way, Loughborough, Leicestershire, LE11 3UZ, UK.
.Tel: + 44 (0) 1509 227 505. Fax: + 44 (0) 1509 286 577.
Email:[1] V.J.Gonzalez-Villela@lboro.ac.uk; [2] R.M.Parkin@lboro.ac.uk.

[1, 4] Departamento de Ingeniería Mecatrónica, Facultad de Ingeniería, UNAM, CU, México.
[3, 4] Departamento de Ingeniería Mecánica, Facultad de Ingeniería, UNAM, CU, México.
Labs. de Ing. Mecánica "Ing. Alberto Camacho Sánchez", Edificio Anexo de la Facultad de Ingeniería, Circuito Exterior, Ciudad Universitaria, UNAM, 04510, México D.F. México.
Tel. + (52 55) 5622 8050 y 51. Fax + (52 55) 5622 8055.
Email: [1] vjgv@servidor.unam.mx; [3] lopezp@servidor.unam.mx; [4] dorador@servidor.unam.mx.

ABSTRACT

The ability of mobile robots to avoid static and moving obstacles using artificial potential fields on the robot's kinematic model, which is endowed with dynamic extension for torque control is addressed in this paper. The effectiveness of the method is validated using MatLab and SIMULINK as simulation framework.

RESUMEN

Este trabajo trata de la habilidad de evadir obstáculos estáticos y móviles con robots móviles que usan campos potenciales artificiales sobre el modelo cinemático del robot y que está dotado con extensión dinámica para el control de torque. La efectividad del método es validada usando como marco de simulación MatLab y SIMULINK.

INTRODUCTION

Mobile robots must be able to move about the environment without colliding with objects or their environment. Methods of avoiding collision can be classified into two types [1]: 1) Path planning methods, and 2) obstacle avoidance methods. The *path planning methods* are those techniques that find the shortest collision free route before the robot moves. The *obstacle avoidance methods* are techniques that are employed whilst the robot is moving.

The artificial potential field approach has been the most popular technique used to avoid obstacles [2]. This technique is used to guide the robot to a given goal at the same time as the robot is avoiding obstacles. Control methods based on artificial potential fields use the vectorial sum of the virtual repulsive force (due to the obstacle) and the virtual attractive force (due to the goal) to guide the robot though the environment [3]. They use the dynamic model of the robot, in which the virtual repulsive and attractive forces are modelled as external forces applied to the robot frame to produce acceleration commands [4].

In [5] an artificial potential field algorithm was proposed, which was able to direct a wheeled mobile robot through the environment using only the kinematic model of the robot. The artificial potential fields were taken as both a repulsive distance vector (to the obstacle) and an attractive distance vector (to the goal) to compute velocity commands. However, this method did not take into account the dynamic aspects of the robot. Moreover, the question about the torques needed to perform the task still remains.

In order to solve the question, it was found a method to compute input acceleration based on a nonlinear feedback control law [6] derived from a previous work [7], where the accelerations of the robot are computed based on a reference mobile robot of the same class. These accelerations are applied to the robot dynamic extension model to compute torque commands. This way, it is possible to control the kinematic model of the robot and extend the effects to the dynamic model by computing motor torque commands for the robot. The method in [6] shows asymptotic stability assured by Lyapunov theory.

This work presents a wheeled mobile robot with two independent motorized rear wheels, which used the potential field method showed in [5] and has been provided with dynamical extension for torque control as showed in [6]. As a result, this method takes into account the dynamic characteristics of the robot and overcome the pure assumption of the perfect velocity tracking of the control method presented in [5].

The method is very useful for directing wheeled mobile robots in structured environments with the presence of fixed and moving obstacles, where the dynamics of the robot has an important role. This method has been modelled in the MatLab and SIMULINK environment to prove the effectiveness of the method.

The remainder of this paper is organized as follows: First of all, the robot posture, the robot's kinematic and dynamic models, and the problem definition are given. Afterwards, the solution with the simulation results is shown. Finally, the conclusions are presented.

ROBOT POSTURE

A wheeled mobile robot is equipped with motors that are driven by an on board computer. It is assumed that the wheeled mobile robot is made of a rigid frame, non-deformable wheels that do not slip, and moves on a horizontal plane. The position of the robot is fully described by the variables x, y and θ as shown in Figure 1.

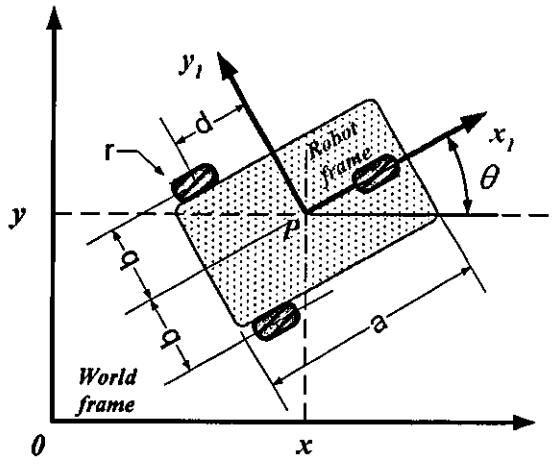


Figure 1: Posture definition

Where P is a fixed point on the robot's platform, whose position is represented by the coordinates (x, y) with respect to the fixed world frame $\{0, x, y\}$ inertial system. The moving robot frame system $\{P, x_1, y_1\}$ is firmly attached to the point P. θ is the orientation angle of the robot frame system $\{P, x_1, y_1\}$ with respect to the world system $\{0, x, y\}$ measured from the x axis to the x_1 axis. The posture of the robot is fully described by the vector $[x \quad y \quad \theta]^T$.

KINEMATIC AND DYNAMIC MODELS FOR A WHEELED MOBILE ROBOT WITH TWO INDEPENDENT MOTORIZED REAR WHEELS

The kinematic constraints of a wheeled mobile robot having n -dimensional configuration space can be expressed in the form:

$$A(q)\dot{q} = 0 \tag{1}$$

A set of these kinematic constraints, for this type of wheeled mobile robot (Figure 1), was taken form [8]:

$$\begin{bmatrix} -\sin\theta & \cos\theta & -d & 0 & 0 \\ -\cos\theta & -\sin\theta & -b & r_w & 0 \\ -\cos\theta & -\sin\theta & b & 0 & r_w \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = 0 \tag{2}$$

where $q \in R^n$ is the vector of the generalized coordinates, $\dot{q} \in R^n$ is the vector of the generalized velocities, $\dot{x}$ and $\dot{y}$ are the linear velocities of the point P along the x and y axis respectively, $\dot{\theta}$ is the angular velocity of the robot frame, $\dot{\phi}_r$ and $\dot{\phi}_l$ are the corresponding angular velocities of the right and left wheel, r_w is the wheel radius, d is the distance from the rear wheels' axle to the point P, and b is the positive distance from each rear wheel to the x_1 axis. The front wheel is a castor wheel that was ignored for the analysis, because it only plays the role of the robot's body stabilizer.

A controllable kinematic state space model of the robot obtained from e.q (1) can be expressed in the form:

$$\dot{q} = S(q)u \tag{3}$$

A model for this type of robot (Figure 1) was taken form [5] and is showed in eq. (4). It was obtained by finding a set of smooth and linear independent vector fields for $S(q)$ that span the null space of $A(q)$ when the $r = n - m$ dimensional steering system $u(t) = [v_1 \quad v_2]^T$ is given, where $v_1 = \dot{x} = v$ is the heading linear velocity of the robot at the point P and $v_2 = \dot{\theta} = \omega$ is the angular velocity of the robot frame.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \\ 1/r_w & b/r_w \\ 1/r_w & -b/r_w \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4}$$

Using the eq. (4) it is possible to compute the whole velocities at the work space $[\dot{x} \;\; \dot{y} \;\; \dot{\theta}]^T$ and at the joint space $[\dot{\phi}_r \;\; \dot{\phi}_l]^T$ due to the heading linear velocity v of the point P and angular velocity ω of the robot frame, where the steering system $u(t) = [v \;\; \omega]^T$ is the input of the kinematic model shown in eq. (4).

A dynamic equation of motion for this type of robot at the configuration space can be written as:

$$M(q)\ddot{q} + C(q,\dot{q}) = P\tau + A^T(q)\lambda \tag{5}$$

where $M(q) \in R^{n\times n}$ is a symmetric positive definite inertia matrix, $C(q,\dot{q}) \in R^{n\times 1}$ is the vector of centripetal and Coriolis forces, $P \in R^{n\times r}$ is the input transformation matrix, $\tau \in R^{r\times 1}$ is the input vector, $A(q) \in R^{m\times n}$ is the matrix associated with the kinematic constraints, and $\lambda \in R^{m\times 1}$ is the vector of the langrage multipliers. The components of eq. (5) were taken from [9]:

$$M(q) = \begin{bmatrix} m & 0 & 2m_w d\sin\theta & 0 & 0 \\ 0 & m & -2m_w d\cos\theta & 0 & 0 \\ 2m_w d\sin\theta & -2m_w d\cos\theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}$$

$$m = m_c + 2m_w$$

$$I = I_c + 2m_w(d^2 + b^2) + 2I_m$$

$$C(q,\dot{q}) = \begin{bmatrix} 2m_w d\dot{\theta}^2\cos\theta \\ 2m_w d\dot{\theta}^2\sin\theta \\ 0 \\ 0 \\ 0 \end{bmatrix}; \qquad P(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix};$$

$$A(q) = \begin{bmatrix} -\sin\theta & \cos\theta & -d & 0 & 0 \\ -\cos\theta & -\sin\theta & -b & r_w & 0 \\ -\cos\theta & -\sin\theta & b & 0 & r_w \end{bmatrix};$$

$$\lambda = [\lambda_1 \;\; \lambda_2 \;\; \lambda_3]^T$$

(6)

where m_c and I_c are the mass and the moment of inertia of the robot's body respectively, m_w, I_m and I_w are the mass, the moment of inertia at the wheel's z axis and the moment of inertia at the wheel's axle of each wheel respectively, and $\lambda \in R^{m\times 1}$ is the vector of the Lagrange multipliers.

In order to find the dynamic equation of motion at the steering space let us differentiate eq. (3), it yields:

$$\ddot{q} = S(q)\dot{u} + \dot{S}(q)u \tag{7}$$

Substituting eq. (7) in eq. (5) and premultiplying both sides by $S^T(q)$. The desired model can be written as:

$$\bar{M}(q)\dot{u} + \bar{C}(q,\dot{q}) = \bar{P}\tau \equiv \bar{\tau} \tag{8}$$

where $\bar{M}(q) = S^T M S$, $\bar{C}(q,\dot{q}) = S^T(M\dot{S}\dot{u} + C)$, and $\bar{P} = S^T P$. Besides, $\bar{M}(q) \in R^{r\times r}$ is a symmetric positive definite inertia matrix, $\bar{C}(q,\dot{q}) \in R^{r\times 1}$ is the vector of centripetal and Coriolis forces, and $\bar{\tau} \in R^{r\times 1}$ is the input vector. $\bar{P}$ is a constant nonsingular matrix that depends on the design specification of the mobile robot. Since $S(q)$ belongs to the null space of $A(q)$ then $S^T(q)A^T(q)$ vanishes.

PROBLEM DEFINITION

This type of wheeled mobile robot has restricted mobility. So, due to the Brockett necessary condition [10], a restricted mobility robot can only be controlled in the same degrees of freedom as inputs has the system to be controlled [5]. Because of the system only has two inputs, as is shown in eq. (4), the chosen two control outputs were the position (x, y). The control of the orientation of the robot can be achieved by using path planning methods, as proposed in [1], but is not treated here.

Thus, the problem of directing the robot from an initial posture to a final position (goal) while avoiding mobile obstacles is defined as follows: given an initial posture

(x, y, θ), reach the position (x_f, y_f) while avoiding moving obstacles. This problem can be divided into the following aspects [5]: reaching the goal, obstacle modelling, and avoiding the obstacle.

Reaching the goal

The problem of reaching the goal (see Figure 2) is defined as the direction and velocity in which the robot should be directed in order to reach the goal. First of all, let us define $e_g = [e_x \quad e_y]$ as the error vector to the goal, $e_d = \sqrt{e_x^2 + e_y^2}$ as the distance error to the goal, and $e_\theta = \theta_g - \theta$ as the angle error to the goal, where $e_x = x_f - x$, $e_y = y_f - y$, and $\theta_g = \tan^{-1}[e_y / e_x]$. Then one can define the position error vector $e_p = [e_d \quad e_\theta]^T$.

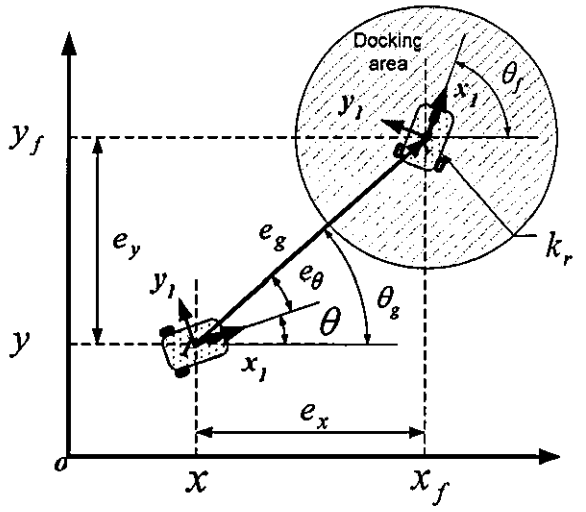


Figure 2: Problem definition of reaching the goal

Thus, the robot is directed from the initial posture (x, y, θ) to the final position (x_f, y_f) using the following rules:

$$v = \begin{cases} v_{\max} & \text{if} \quad |e_d| > k_r \\ \dfrac{v_{\max}}{k_r} \cdot e_d & \text{if} \quad |e_d| \le k_r \end{cases};$$

$$\omega = \omega_{\max} \sin(e_\theta) \qquad (9)$$

where $v_{\max}$ is the maximum heading velocity, $\omega_{\max}$ is the maximum angular velocity of the robot, and k_r is the radius of the docking area. With these rules eq. (9) the robot is directed to the final position at the maximum linear velocity when it is outside of the docking area, and it will be slowing down when it is inside the docking area. On the other hand, the robot will be correcting the heading angle as a function of the sine of the error angle e_θ, where the maximum angular velocity $\omega = \pm\omega_{\max}$ will be achieved at $e_\theta = \pm 90^o$, and the minimum angular velocity $\omega = 0$ will be at $e_\theta = 0^o$.

Obstacle modelling

The model of the obstacle was taken from [4] and adapted to be useful for modelling circular objects in [5], where the artificial potential field U_0 was taken as a repulsive distance vector. Then, using the distance to the centre of the circular obstacle, the potential function can be written as:

$$U_0 = \begin{cases} \dfrac{1}{2}\eta\left(\dfrac{1}{\rho_c - \rho_r} - \dfrac{1}{\rho_0}\right)^2 & \text{if} \quad \rho_c \le \rho_0 + \rho_r \\ 0 & \text{if} \quad \rho_c > \rho_0 + \rho_r \end{cases} \qquad (10)$$

where η is a constant gain, ρ_0 represents the limit distance of the potential field influence, ρ_r is the radius of the circular obstacle, and ρ_c is the distance to the centre of the obstacle. The selection of η and ρ_0 depends on the mobile robot $v_{\max}$ and $\omega_{\max}$ velocities that allows the robot to be able to avoid obstacles.

Avoiding the obstacle

Avoiding obstacles while the robot is reaching a goal using artificial potential field methods has been defined as [5]:

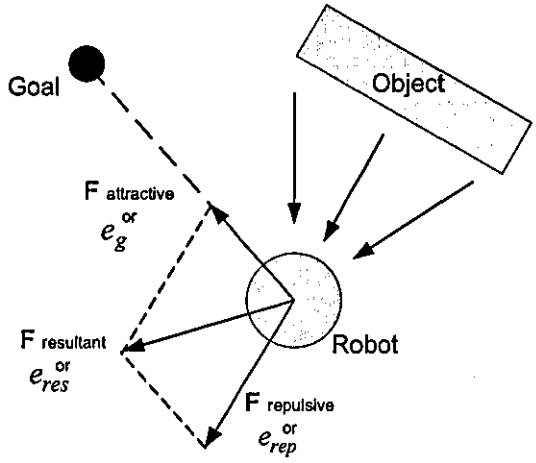


Figure 3: Artificial potential field control approach

the control method which uses the vectorial sum of virtual repulsive forces (due to the objects) and virtual attractive force (due to the goal) to provide a resultant force to be applied to the robot dynamic model to guide the vehicle, as shown in Figure 3. In this case we are assuming that those potential forces are attractive and repulsive distance vectors, which can be used to produce velocity commands useful to be applied on the robot kinematic model to guide de vehicle.

PROPOSED SOLUTION

Computing velocity commands

By integrating the first three generalized speeds $[\dot{x} \; \dot{y} \; \dot{\theta}]^T$ of the kinematic model eq. (4) one can compute the desired reference position and orientation of the robot (x_r, y_r, θ_r). So, with the knowledge of the final position (x_f, y_f) the distance vector to the goal e_g is computed.

The distance resultant vector $e_{res} = e_g + e_{rep}$ is computed in the presence of obstacles, as the total vector sum of repulsive distances due to objects e_{rep} and the distance vector to the goal e_g. Besides, we can compute a new avoid angle error defined as $\theta_{avoid,e} = \theta_{res} - \theta$, where θ_{res} is the angle of the e_{res} vector. Finally, the new rules can be written as:

$$v_r = \begin{cases} v_{\max} & if \quad |e_{res}| > k_r \\ \dfrac{v_{\max}}{k_r} \cdot e_{res} & if \quad |e_{res}| \le k_r \end{cases};$$
$$\omega_r = \omega_{\max} \sin(\theta_{avoid,e})$$

(11)

where $[v_r \quad \omega_r]^T$ is the reference steering system, which directs the robot into the goal while avoiding obstacles

Computing acceleration commands

The computation of the acceleration commands can be done by using eq. (12), which were taken form [6] and are Lyapunov stables. They use the current position and velocities of a reference robot to compute the corresponding acceleration commands for the robot, they are:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix};$$

$$\dot{e}_p = \begin{bmatrix} w \cdot e_2 - v + v_r \cos e_3 \\ -w \cdot e_1 + v_r \sin e_3 \\ w_r - w \end{bmatrix};$$
$$v_c = \begin{bmatrix} v_r \cos(e_3) + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix};$$
$$\dot{v}_c = \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{e}_p;$$
$$\dot{v} = \dot{v}_c + K_4 (v_c - v)$$

(12)

where the subscript r implies reference robot variables, $K_4 = k_4 I$, k_1, k_2, k_3 and k_4 are positive constants, I is the identity matrix, x, y, θ, v, w and x_r, y_r, θ_r, v_r, w_r are the position, orientation, and linear and angular velocities of both the robot and the reference robot respectively, and $\dot{v} = [\dot{v} \; \dot{w}]^T$ is the vector of the acceleration commands.

Computing the torque commands

To compute the motor torque commands τ is needed to compute the acceleration commands $\dot{v}$ from eq. (12) and substitute them in the left side of eq. (8), in order to find the torque commands $\bar{\tau}$. Thus, the motor torque commands τ are computed by solving the right side of eq. (8) as:

$$\tau \equiv \bar{P}^{-1} \bar{\tau}$$

(13)

SIMULATION RESULTS

In this case we are assuming that we have a wheeled mobile robot with the following dimensions [9]: $b = 0.75m$, $a = 2.00m$, $d = 0.30m$, $r_w = 0.075m$, $m_c = 30kg$, $m_w = 1kg$, $I_c = 15.625kg \cdot m^2$, $I_m = 0.0025kg \cdot m^2$, $I_w = 0.005kg \cdot m^2$. That can move at $|v_{\max}| = 1 \cdot m/s$ and at $|\omega_{\max}| = \pi/5 \cdot 1/s$. The object has been modelled as a man with radius $\rho_r = 1/2 \cdot m$, who is walking about the environment at a walking speed of $1 \pm 0.5 \cdot m/s$, with a field influence of $\rho_0 = 3m$, and $\eta = 4000$. In this case the robot should reach the point $(x_f, y_f) = (10m, 10m)$ departing from the posture $(x_i, y_i, \theta_i) = (-10m, -10m, (-1/2)\pi \cdot rad)$, while avoiding a moving obstacle. The value of the constants were $k_r = 2m$, $k_1 = 4$, $k_2 = 4$, $k_3 = 4$ and $k_4 = 8$.

The results of the simulation are presented in Figures 4 to 8. All of them can be approximately divided for the

analysis into four zones: 1) start up zone: the first 0 to 11 sec., 2) obstacle avoidance zone: the next 11 to 15 sec., 3) stabilization zone: the next 15 to 31 sec., and 4) docking zone: the final 31 to 40 sec. The nearest distance between the robot and the obstacle took place at 13.3 sec. and was of 1.85 m, which is good enough because it gives about 0.65 m of tolerance for control errors. Figure 4 shows the sequence of simulation results with a sample rate of 1 sec., where a rounded obstacle is approaching the robot's trajectory, while the robot is avoiding the obstacle. Figure 5 shows the reference robot variables associated with the computation of the velocity commands from eq. (11) and eq. (4), these are: x_r, y_r, θ_r, v_r, and w_r respectively, these variables presuppose perfect velocity tracking. Figure 6 shows the current variables of the robot as a result of applying the computed linear and angular accelerations $\dot{u}=[\dot{v}\ \dot{w}]$ from eq. (12), these are: x, y, θ, v, w, $\dot{v}$ and $\dot{w}$ respectively, these variables include the dynamic behaviour of the robot. Figure 7 shows the current angular velocity of the wheels, due to the application of the torque commands from eq. (13), these are: $\dot{\phi}_l$, $\dot{\phi}_r$, τ_l and τ_r respectively, where we can see that the value of the torques were reasonable according to the task assigned. Finally, Figure 8 shows the position and orientation errors between the value of the reference robot variables and the current robot variables, these are: $e_x = x_r - x$, $e_y = y_r - y$ and $e_\theta = \theta_r - \theta$ respectively, which showed asymptotic stability with reasonable values.

CONCLUSIONS

This work has presented a dynamic torque control extension to compute motor torque commands for a wheeled mobile robot that uses a potential field method on the kinematic model of the robot to avoid static and mobile obstacles.

This method takes into account the dynamic characteristics of the robot and overcomes the pure assumption of the perfect velocity tracking of the control method presented in [5]. The formulation of the dynamic extension model was necessary for the development of the algorithm. This method is based on computing robot accelerations from the kinematic model of a reference robot instead of computing virtual forces applied to the robot frame as showed in [4]. The position and orientation errors showed asymptotic stability with reasonable values.

With this method it is possible to compute torque commands for faster mobile robots that react timely to the dynamic changes of the environment with the presence of fixed or/and moving obstacles.

ACKNOWLEDGEMENTS

The first author would like to acknowledge the "Dirección General de Asuntos del Personal Académico" (DGAPA) and the "Facultad de Ingeniería" of the "Universidad Nacional Autónoma de México" (UNAM) for sponsoring this research.

REFERENCES

1. González Villela, V.J. and Parkin, R.M., "*Evadiendo obstáculos con robots móviles*", **Revista Digital Universitaria [en línea]**. 6(1). ISSN:1607-6079, Available on the Internet: http://www.revista.unam.mx. 2005.

2. Latombe, J.C., "*Robot motion planning*", Boston: Kluwer Academic Publishers. xviii, 651. ISBN: 0792391292. 1991.

3. Zhao, Y. and BeMent, S.L. "*Kinematics, dynamics and control of wheeled mobile robots*". **Robotics and Automation, Proceedings of the IEEE International Conference on**. Nice, France. ISBN: 0-8186-2720-4. 1992.

4. Khatib, O., "*Real-time obstacle avoidance for manipulators and mobile robots*". **International Journal of Robotics Research, 5(1)**: p. 90-98. ISSN: 0278-3649. 1986.

5. Gonzalez-Villela, V.J., Parkin, R.M., Lopez-Parra, M., Dorador-Gonzalez, J.M., et al., "*A wheeled mobile robot with obstacle avoidance capability*". **Ingeniería Mecánica Tecnología y Desarrollo. Revista de la Sociedad Mexicana de Ingeniería Mecánica (SOMIM). 1(5)**: p. 159-166. ISSN: 1665-7381. 2004.

6. Fierro, R. and Lewis, F.L., "*Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics*". **Journal of Robotic Systems, 14(3)**: p. 149-163. ISSN: 07412223. 1997.

7. Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. "*A stable tracking control method for an autonomous mobile robot*". **Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on**. ISBN: 0-8186-9061-5. 1990.

8. Yamamoto, Y. and Yun, X., "*Coordinating locomotion and manipulation of a mobile manipulator*", ***Recent trends in mobile robots***, Y.F. Zheng, Editor. World Scientific: Singapore ; London. p. 157-181. ISBN: 9810215118. 1993.

9. Sarkar, N., Yun, X., and Kumar, V., "*Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots*". **International Journal of Robotics Research, 13(1)**: p. 55-69. ISSN: 0278-3649. 1994.

10. Brockett, R.W. "*Asymptotic stability and feedback stabilization*". **Differential geometric control theory, Proceedings of the conference**. Michigan Technological University: Boston, Mass. : Birkhauser. 0364330910. 1983.

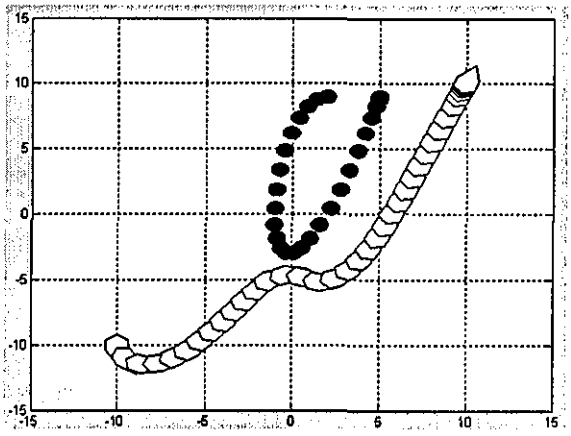


Figure 4: Simulation sequence

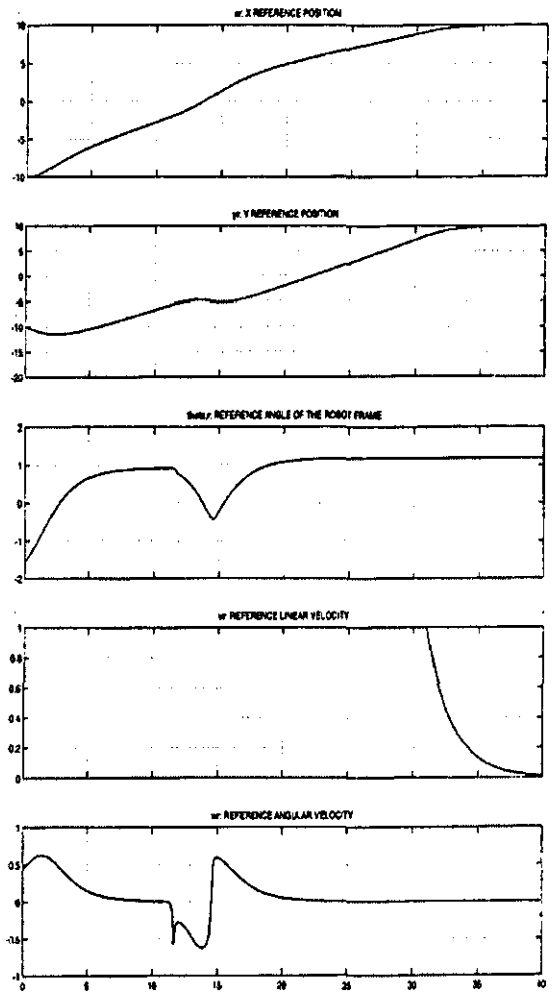


Figure 5: Reference robot variables

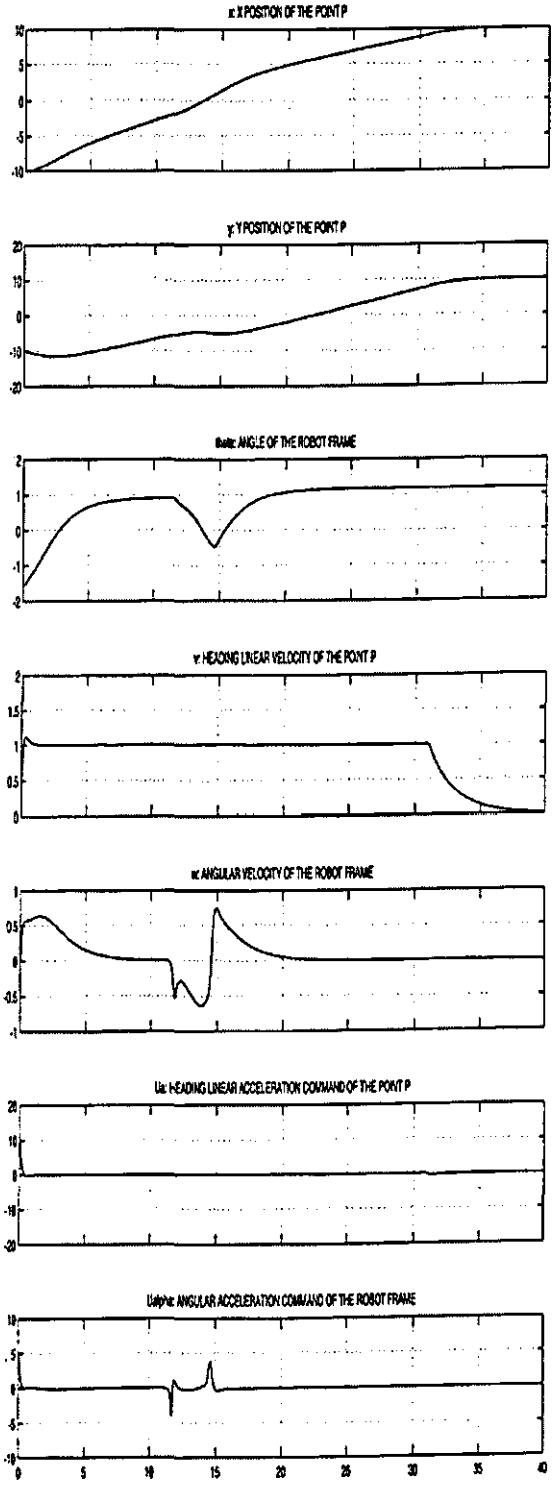


Figure 6: Robot variables

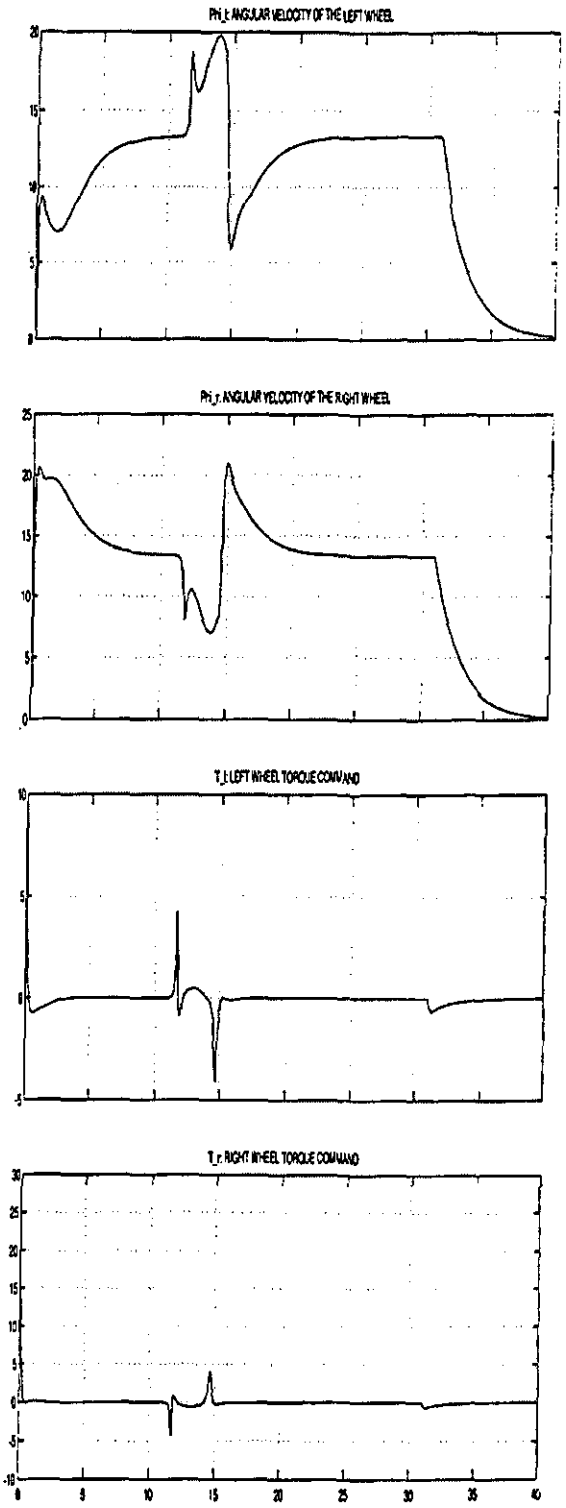


Figure 7: Angular velocities and torque commands of the wheels

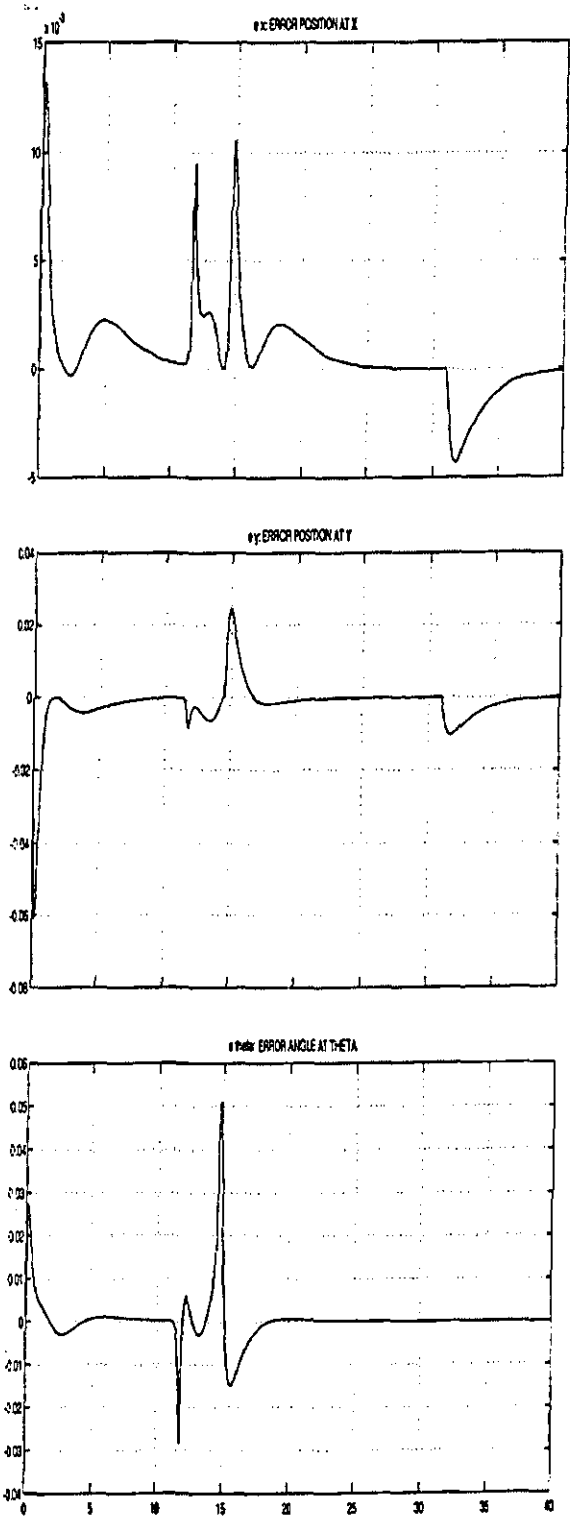


Figure 8: Position and orientation errors