

Pilkington Library

Author/Filing Title REEDMAN

Vol No Class Mark T

**Please note that fines are charged on ALL
overdue items.**

REEDMAN, R. J. 1988

0402693426



BADMINTON PRESS
UNIT 1 BROOK ST
SYSTON
LEICESTER LE7 1GD
ENGLAND
TEL 0116 260 2917
FAX 0116 269 6639

THE DESIGN AND CONTROL OF A MANIPULATOR FOR
SAFETY CRITICAL DEPLOYMENT APPLICATIONS

by

Adam Victor Creyke Reedman

A Doctoral Thesis


Submitted in partial fulfilment of the requirements

for the award of

The Degree of Doctor of Philosophy of Loughborough University

January 2002

© Adam Victor Creyke Reedman

| | |
|--|-----------|
|  Commonwealth of Massachusetts State Library | |
| Date | Jan 03 |
| Class | |
| Acc No. | 020269342 |

[illegible]

448

10

◆ ◆ ◆

Abstract

Development of manipulators that interact closely with humans has been a focus of research in fields such as robot-assisted surgery. The recent introduction of powered surgical-assistant devices into the operating theatre has meant that modified industrial robot manipulators have been required to interact with both patient and surgeon. Some of these robots require the surgeon to grasp the end-effector and apply a force while the joint actuators provide resistance to motion. In the operating theatre, the use of high-powered mechanisms to perform these tasks could compromise the safety of the patient, surgeon, and operating room staff.

In this investigation, a two degree-of-freedom (2-DOF) manipulator is designed for the purpose of following a pre-defined path under the direct control of the surgeon. In order to ensure safety and accuracy a novel application of non-backdriveable joint mechanism based upon a worm-wheel and worm gears is presented. The two motor-driven worm gears of each joint are controlled using state-of-the-art control methods. Namely, computed-torque and composite adaptive controllers are used in conjunction with a mathematical model of the joint mechanism to minimise backlash and regulate joint position. Simulation and experimental results show that for a 1-DOF system joint error was controlled to be less than ± 0.0015 rads using both control schemes. For the 2-DOF system, the end-effector can only be constrained to remain along the path (to within ± 3 mm) with a computed-torque controller despite the large effects of unmodelled dynamics such as joint flexibility. The composite adaptive controller was not shown to be robust in the face of large system flexibility and was unable to regulate end-effector position with the same accuracy as the computed-torque controller.

An in-depth analysis of the motion of the 2-DOF planar manipulator highlights the need to show the user how to apply a force to the control handle in order to accomplish the desired task. A control handle is presented that incorporates a pointing device for this purpose.

Acknowledgements

I would like to express my sincerest thanks to my Supervisor, Dr K Bouazza-Marouf, for his support and guidance throughout this research.

The opportunity to carry out this research has been made possible by the financial support of the Engineering and Physical Sciences Research Council (EPSRC) and Loughborough University. I am grateful for all of their support during the last three years.

I would like to thank Bob Ludlam and all the technical staff in the mechanical workshop for their patience and help in the construction of the manipulator. Similarly, I would like to thank Steve Retter and the staff in the Department's Electronics Workshop for their help and advice. I would also like to thank Neville Carpenter in the Department of Manufacturing Engineering for his help in the production of the circuit boards and R & T Mechatronics Ltd for the use of their machine shop facilities.

I would especially like to thank my wife and family for their continual support and encouragement.

Dedication

To my wife and parents:

Thank you for all of your patience, understanding and support

iv

| | |
|--|-----------|
| CHAPTER 4 MECHANICAL DESIGN OF A PROTOTYPE ROBOT FOR | |
| PASSIVE DEPLOYMENT APPLICATIONS | 52 |
| 4 1 INTRODUCTION | 52 |
| 4 2 BACKLASH CANCELLATION | 54 |
| 4 3 ROBOTIC JOINT DESIGN | 55 |
| 4 4 TWO DEGREE-OF-FREEDOM MANIPULATOR | 56 |
| 4 5 SENSORS AND ACTUATORS | 57 |
| 4 5 1 <i>Worm Drive Motors</i> | 57 |
| 4 5 2 <i>Position Measurement</i> | 58 |
| 4 5 3 <i>Dynamic Torque Measurement</i> | 58 |
| 4 6 CONTROL HANDLE | 61 |
| CHAPTER 5 A MECHATRONIC WORKSTATION FOR REAL-TIME | |
| CONTROL DEVELOPMENT | 64 |
| 5 1 INTRODUCTION | 64 |
| 5 2 SYSTEM OVERVIEW | 64 |
| 5 3 SINGLE-AXIS MOTION CONTROL INTERFACE | 66 |
| 5 4 POWER ELECTRONIC INTERFACE | 69 |
| 5 5 CONTROL DEVELOPMENT STUDIO | 71 |
| 5 5 1 <i>Description</i> | 71 |
| 5 5 2 <i>Principle of operation</i> | 72 |
| CHAPTER 6 MATHEMATICAL MODEL OF THE DUAL-WORM DRIVEN | |
| JOINT | 74 |
| 6 1 INTRODUCTION | 74 |
| 6 2 SYSTEM ANALYSIS | 74 |
| 6 3 MATHEMATICAL MODEL | 75 |
| CHAPTER 7 COMPUTED-TORQUE CONTROL OF A ONE DEGREE-OF- | |
| FREEDOM WORM DRIVEN MANIPULATOR | 80 |
| 7 1 INTRODUCTION | 80 |
| 7 2 CONTROL STRATEGY | 80 |
| 7 3 α -UNWINDING CONTROL | 81 |
| 7 4 β -UNWINDING CONTROL | 82 |
| 7 5 EXPERIMENTAL SET-UP | 83 |
| 7 6 RESULTS AND DISCUSSION | 84 |
| 7 6 1 <i>Simulation results</i> | 84 |
| 7 6 2 <i>Experimental results</i> | 86 |
| 7 7 CONCLUSIONS | 89 |

| | |
|--|-----------|
| CHAPTER 8 COMPOSITE ADAPTIVE CONTROL OF A ONE DEGREE-OF-FREEDOM WORM DRIVEN MANIPULATOR | 90 |
| 8 1 INTRODUCTION | 90 |
| 8 2 MATHEMATICAL MODEL | 90 |
| 8 3 CONTROL STRATEGY | 91 |
| 8 4 COMPOSITE ADAPTIVE CONTROLLER | 94 |
| 8 4 1 α -unwinding control | 94 |
| 8 4 2 β -unwinding control | 99 |
| 8 4 3 Stability Analysis | 99 |
| 8 4 4 Experimental Set-up | 102 |
| 8 5 RESULTS AND DISCUSSION | 102 |
| 8 5 1 Simulation Results | 102 |
| 8 5 2 Experimental Results | 108 |
| 8 5 3 Discussion | 115 |
| 8 6 CONCLUSIONS | 120 |

| | |
|--|------------|
| CHAPTER 9 MATHEMATICAL MODEL OF THE TWO DEGREE-OF-FREEDOM MANIPULATOR | 122 |
| 9 1 INTRODUCTION | 122 |
| 9 2 DENAVIT-HARTENBURG REPRESENTATION | 122 |
| 9 3 LINK JACOBIAN | 124 |
| 9 4 LAGRANGE-EULER FORMULATION OF THE MANIPULATOR DYNAMICS | 126 |
| 9 5 INVERSE KINEMATICS | 128 |
| 9 6 PATH REPRESENTATION | 130 |

| | |
|--|------------|
| CHAPTER 10 COMPUTED-TORQUE CONTROL OF A TWO DEGREE-OF-FREEDOM WORM-DRIVEN MANIPULATOR | 133 |
| 10 1 INTRODUCTION | 133 |
| 10 2 CONTROL STRATEGY | 133 |
| 10 3 CONTROL OF THE POINTER MECHANISM | 135 |
| 10 4 MATHEMATICAL MODEL | 137 |
| 10 5 COMPUTED-TORQUE CONTROLLER | 138 |
| 10 5 1 Joint 1 α -unwinding controller | 138 |
| 10 5 2 Joint 1 β -unwinding controller | 139 |
| 10 5 3 Joint 2 α -unwinding controller | 139 |
| 10 5 4 Joint2 β -unwinding controller | 140 |
| 10 6 EXPERIMENTAL SET-UP | 140 |
| 10 7 RESULTS AND DISCUSSION | 142 |
| 10 7 1 Discussion | 149 |
| 10 8 CONCLUSIONS | 153 |

CHAPTER 11 COMPOSITE ADAPTIVE CONTROL OF A TWO DOF WORM-DRIVEN MANIPULATOR 154

| | |
|---|-----|
| 11 1 INTRODUCTION | 154 |
| 11 2 CONTROL STRATEGY | 154 |
| 11 3 EXPERIMENTAL SET-UP | 154 |
| 11 4 MATHEMATICAL MODEL | 155 |
| 11 5 COMPOSITE ADAPTIVE CONTROLLER | 155 |
| 11 5 1 α -unwinding control of joint n | 156 |
| 11 5 2 β -unwinding control of joint n | 157 |
| 11 5 3 Stability Analysis | 158 |
| 11 5 4 α -unwinding control algorithm | 160 |
| 11 6 RESULTS AND DISCUSSION | 161 |
| 11 7 DISCUSSION | 165 |
| 11 8 CONCLUSIONS | 165 |

CHAPTER 12 CONCLUSIONS 166

| | |
|-------------------------------------|-----|
| 12 1 INTRODUCTION | 166 |
| 12 2 CONCLUSIONS FROM THIS RESEARCH | 166 |
| 12 3 PUBLICATIONS | 170 |
| 12 3 1 Published | 170 |
| 12 3 2 Submissions | 170 |

CHAPTER 13 RECOMMENDATIONS FOR FURTHER WORK 171

| | |
|---|-----|
| 13 1 INTRODUCTION | 171 |
| 13 2 CONTROL DEVELOPMENT STUDIO | 171 |
| 13 3 SINGLE AXIS MOTION CONTROL INTERFACE | 171 |
| 13 4 MECHANICAL SYSTEM | 172 |
| 13 4 1 Joint Configuration | 172 |
| 13 4 2 Joint Actuators | 173 |
| 13 4 3 Joint Mechanism | 174 |
| 13 4 4 Control Strategy | 174 |

REFERENCES 175

APPENDIX A SINGLE AXIS MOTION CONTROL INTERFACE CARD USER MANUAL 189

APPENDIX B POWER ELECTRONICS INTERFACE CONNECTION DIAGRAMS 229

| | |
|---|-----|
| B 1 INPUT AND OUTPUT CONNECTOR DIAGRAMS | 230 |
| B 1 1 Control Card Input | 230 |
| B 1 2 Encoder Input Connector | 231 |
| B 1 3 Motor Output Connector | 231 |
| B 1 4 Digital I/O Connector | 231 |
| B 1 5 Analogue Input Connector | 232 |
| B 2 INTERNAL CONNECTOR DIAGRAMS | 232 |
| B 2 1 Isolator Connectors | 232 |
| B 2 2 Motor Amplifier Connector | 234 |

| | | |
|-------------------|---|------------|
| APPENDIX C | CONTROL DEVELOPMENT STUDIO | |
| | PROGRAMMING/USER GUIDE | 235 |
| C 1 | GETTING STARTED | 235 |
| C 2 | GENERATING A NEW CONTROL PROGRAM | 236 |
| C 3 | BUILDING A CONTROL PROGRAM | 236 |
| C 4 | RUNNING A CONTROL PROGRAM | 237 |
| C 5 | SAVING LOGGED DATA | 237 |
| C 6 | EDITING A CONTROL PROGRAM | 237 |
| | C 6 1 <i>int InitControl(void)</i> | 238 |
| | C 6 2 <i>void StartControl(void)</i> | 238 |
| | C 6 3 <i>void DoControl(long count)</i> | 238 |
| | C 6 4 <i>void StopControl(void)</i> | 238 |
| | C 6 5 <i>int AddLogVariable(char *buffer, double *pfVar, long clr)</i> | 239 |
| | C 6 6 <i>void LogVariables(void)</i> | 239 |
| | C 6 7 <i>int AddControlVariable(char *buffer, double *pfVar)</i> | 239 |
| | C 6 8 <i>void UpdateControlVariable(double *pfVar)</i> | 239 |
| | C 6 9 <i>void DisplayUserMessage(char *pstr)</i> | 240 |
| C 7 | ADDING LIBRARY AND C++ SOURCE FILES | 240 |
| | C 7 1 <i>Library Files</i> | 240 |
| | C 7 2 <i>C++ Source Files</i> | 240 |
| | C 7 3 <i>An example make file</i> | 242 |
| | C 7 4 <i>An example link file</i> | 242 |
| APPENDIX D | ENGINEERING DRAWINGS | 243 |
| D 1 | WORM DRIVE UNITS | 243 |
| D 2 | LINK ASSEMBLY | 253 |
| D 3 | JOINT ASSEMBLY | 259 |
| D 4 | CONTROL HANDLE ASSEMBLY | 270 |
| APPENDIX E | DERIVATION OF MATHEMATICAL MODEL | 281 |
| E 1 | CASE I | 281 |
| E 2 | CASE II | 285 |
| E 3 | CASE III | 286 |
| E 4 | CASE IV | 288 |
| APPENDIX F | COMMON DEFINITIONS IN STABILITY ANALYSIS | 290 |
| F 1 | BARBALAT'S LEMMA | 290 |
| F 2 | RAYLEIGH-RITZ THEOREM | 290 |
| F 3 | UNIFORM CONTINUITY | 290 |
| F 4 | PROPERTIES OF A STABLE TRANSFER FUNCTION | 291 |
| APPENDIX G | MOTOR/GEARBOX MODEL | 292 |
| APPENDIX H | PUBLICATIONS | 294 |
| H 1 | CONTROL OF AN ACTIVELY CONSTRAINED ROBOTIC JOINT FOR PASSIVE DEPLOYMENT APPLICATIONS | 294 |
| H 2 | COMPOSITE ADAPTIVE CONTROL | 306 |

Table of Figures

| | |
|--|----|
| FIGURE 1 – MECHANICAL AXIS OF RIGHT LEG | 3 |
| FIGURE 2 – TOTAL KNEE REPLACEMENT (TKR) BONE REMOVAL AND PROCEDURE FOR FITTING THE FEMORAL PROSTHETIC COMPONENT | 5 |
| FIGURE 3 –TOTAL KNEE REPLACEMENT (TKR) BONE REMOVAL AND FITTING OF TIBIAL PROSTHETIC COMPONENT | 6 |
| FIGURE 4 - DEVIATION OF SAW FROM CUTTING PATH | 7 |
| FIGURE 5 - PADYC JOINT ARCHITECTURE | 18 |
| FIGURE 6 – CEMENTLESS TOTAL HIP REPLACEMENT PROCEDURE | 21 |
| FIGURE 7 - ROBODOC SYSTEM | 22 |
| FIGURE 8 - STEWART PLATFORM | 26 |
| FIGURE 9 - FIXATION OF TROCHANTERIC AND NECK FRACTURES | 29 |
| FIGURE 10 - FIXATION OF SHAFT FRACTURE | 29 |
| FIGURE 11 - OSTEOTOMY PROCEDURE | 31 |
| FIGURE 12 – 2-DOF MANIPULATOR FOLLOWING A CIRCULAR PATH | 41 |
| FIGURE 13 - TRAJECTORY FOLLOWING USING A 2-DOF MANIPULATOR | 44 |
| FIGURE 14 - AREAS OF MOTION FOR A 2-DOF MANIPULATOR | 47 |
| FIGURE 15 - ROBOT CONFIGURATION FOR $\theta_1 = 0$, $\theta_2 = \pi/4$ AND $\gamma = -7\pi/36$ RAD | 48 |
| FIGURE 16 - REQUIRED JOINT TORQUE FOR $F_L=5N$, $\phi=\pi/6$, $\theta_1=0$, AND $\theta_2=\pi/4$ | 49 |
| FIGURE 17 – DUAL-WORM DRIVE JOINT | 53 |
| FIGURE 18 - METHODS OF FIXING THE WORM WHEEL TO THE MAIN SHAFT | 55 |
| FIGURE 19 - WORM DRIVEN JOINT | 56 |
| FIGURE 20 – 2-DOF MANIPULATOR WITH CONTROL HANDLE | 57 |
| FIGURE 21 - ACROBOT CONTROL HANDLE | 59 |
| FIGURE 22 - FORCE MEASUREMENT FOR THE WORM-DRIVEN MANIPULATOR | 60 |
| FIGURE 23 - STRAIN GAUGE MODULE | 60 |
| FIGURE 24 – MANIPULATOR CONTROL HANDLE SCHEMATIC | 61 |
| FIGURE 25 - CONTROL HANDLE AND POINTER MECHANISM | 62 |
| FIGURE 26 - CONTROL HANDLE ASSEMBLY DRAWING | 63 |
| FIGURE 27 - MANIPULATOR CONTROL SYSTEM OVERVIEW | 65 |
| FIGURE 28 - MOTION CONTROL COMPUTER | 66 |
| FIGURE 29 - SINGLE AXIS MOTION CONTROL INTERFACE CARD BLOCK DIAGRAM | 68 |
| FIGURE 30 – SINGLE AXIS MOTION CONTROL INTERFACE CARD | 68 |
| FIGURE 31 - DC SERVO MOTOR AMPLIFIER CIRCUIT DIAGRAM | 69 |
| FIGURE 32 - DC SERVO MOTOR AMPLIFIER CARD | 70 |
| FIGURE 33 - POWER ELECTRONICS AND OPTO-ISOLATOR CARDS | 70 |

| | |
|--|-----|
| FIGURE 34 - CONTROL DEVELOPMENT STUDIO SCREEN SHOT | 72 |
| FIGURE 35 - CONTROL DEVELOPMENT STUDIO MODE OF OPERATION | 73 |
| FIGURE 36 - DUAL-WORM JOINT MODES OF CONTACT | 75 |
| FIGURE 37 - FREE-BODY DIAGRAM OF WORM DRIVE JOINT CASE I | 76 |
| FIGURE 38 - FREEBODY DIAGRAM OF STATIC CONDITIONS | 78 |
| FIGURE 39 - A SCHEMATIC OVERVIEW OF THE 1-DOF CONTROL SYSTEM | 83 |
| FIGURE 40 - DUAL-WORM JOINT SIMULATION RESULTS | 86 |
| FIGURE 41 - SINGLE WORM JOINT EXPERIMENTAL RESULTS | 87 |
| FIGURE 42 - DUAL-WORM JOINT EXPERIMENTAL RESULTS | 88 |
| FIGURE 43 - USER DEFINED VELOCITY DEMAND | 92 |
| FIGURE 44 - THE THREE USER-INPUT TORQUE COMMANDS USED IN THE SIMULATION | 105 |
| FIGURE 45 - DESIRED POSITION DEMAND FOR THE THREE USER-INPUTS | 105 |
| FIGURE 46 - COMPARISON OF TRACKING ERRORS DURING THE SIMULATION OF THE THREE USER-INPUT TORQUE COMMANDS | 106 |
| FIGURE 47 - SIMULATION PARAMETER ESTIMATES WITH USER-INPUT 1 | 106 |
| FIGURE 48 - SIMULATION PARAMETER ESTIMATES WITH USER-INPUT 2 | 107 |
| FIGURE 49 - SIMULATION PARAMETER ESTIMATES WITH USER-INPUT 3 | 107 |
| FIGURE 50 - EXPERIMENTAL USER-INPUT TORQUE COMMANDS | 110 |
| FIGURE 51 - EXPERIMENTAL DESIRED AND ACTUAL POSITION | 111 |
| FIGURE 52 - COMPUTED-TORQUE CONTROL VERSUS COMPOSITE ADAPTIVE CONTROL | 113 |
| FIGURE 53 - EXPERIMENTAL PARAMETER ESTIMATES WITH USER-INPUT 1 | 114 |
| FIGURE 54 - EXPERIMENTAL PARAMETER ESTIMATES WITH USER-INPUT 2 | 114 |
| FIGURE 55 - EXPERIMENTAL PARAMETER ESTIMATES WITH USER-INPUT 3 | 115 |
| FIGURE 56 - PARAMETER ESTIMATES WITH LESS EXCITING TRAJECTORY | 119 |
| FIGURE 57 - PARAMETER ESTIMATES WITH MORE EXCITING TRAJECTORY | 119 |
| FIGURE 58 - COMPARISON OF TRACKING ERROR WITH ERRONEOUS MODEL | 120 |
| FIGURE 59 - 2-DOF MANIPULATOR | 122 |
| FIGURE 60 - ROBOT GEOMETRY | 129 |
| FIGURE 61 - A PIECE-WISE REPRESENTATION OF A CARTESIAN PATH | 131 |
| FIGURE 62 - A SIMPLE CARTESIAN PATH | 132 |
| FIGURE 63 - USER DEFINED VELOCITY DEMAND | 134 |
| FIGURE 64 - REGIONS OF USER-INPUT FORCE AND PATH DIRECTION | 135 |
| FIGURE 65 - MAPPING POINTER DIRECTION TO DESIRED PATH | 136 |
| FIGURE 66 - A SCHEMATIC OVERVIEW OF THE 2-DOF MANIPULATOR CONTROL SYSTEM | 141 |
| FIGURE 67 - IMPLEMENTATION OF THE FOOT-SWITCH | 141 |
| FIGURE 68 - TRACKING OF A CIRCULAR PATH | 144 |
| FIGURE 69 - END-EFFECTOR POSITION ERROR | 145 |
| FIGURE 70 - MEASURED JOINT TORQUE FOR CIRCULAR PATH | 145 |
| FIGURE 71 - APPLIED USER-INPUT FORCE | 146 |

| | |
|---|-----|
| FIGURE 72 – TRACKING OF A COMPOSITE PATH | 147 |
| FIGURE 73 - END-EFFECTOR POSITION ERROR | 148 |
| FIGURE 74 - APPLIED USER-INPUT FORCE | 149 |
| FIGURE 75 - END-EFFECTOR POSITION WITH $V_{d\max} = 0.01 \text{ M/S}$ | 150 |
| FIGURE 76 - END-EFFECTOR POSITION ERROR WITH $V_{d\max} = 0.01 \text{ M/S}$ | 151 |
| FIGURE 77 - APPLIED USER-INPUT FORCE | 152 |
| FIGURE 78 - BLOCK DIAGRAM OF α -UNWINDING CONTROL ALGORITHM | 161 |
| FIGURE 79 - END-EFFECTOR POSITION | 162 |
| FIGURE 80 - END-EFFECTOR ERROR UNDER COMPOSITE ADAPTIVE CONTROL | 163 |
| FIGURE 81 - MEASURED JOINT TORQUE | 163 |
| FIGURE 82 - CALCULATED EQUIVALENT END-EFFECTOR FORCE | 164 |
| FIGURE 83 - PARAMETER ESTIMATES | 164 |
| FIGURE 84 - CONFIGURATIONS OF A 2-DOF ROBOT AND WORKSPACE | 172 |
| FIGURE 85 - BACK PLATE DIMENSIONS | 229 |
| FIGURE 86 - 44 WAY D-TYPE CONTROL CARD CONNECTOR | 230 |
| FIGURE 87 - 9 WAY D-TYPE ENCODER CONNECTOR | 231 |
| FIGURE 88 - 15 WAY D-TYPE MOTOR CONNECTOR | 231 |
| FIGURE 89 - 25 WAY DIGITAL IO CONNECTOR | 231 |
| FIGURE 90 - 9 WAY ANALOGUE INPUT CONNECTOR | 232 |
| FIGURE 91 - ISOLATOR CARD CONNECTORS | 233 |
| FIGURE 92 - MOTOR AMPLIFIER CONNECTOR | 234 |
| FIGURE 93 – FREE-BODY DIAGRAM CASE I | 281 |
| FIGURE 94 – FREE-BODY DIAGRAM OF STATIC CONDITIONS CASE I | 284 |
| FIGURE 95 - FREE-BODY DIAGRAM CASE II | 285 |
| FIGURE 96 – FREE-BODY DIAGRAM CASE III | 286 |
| FIGURE 97 – FREE-BODY DIAGRAM CASE IV | 288 |
| FIGURE 98 - GEARBOX MODEL | 292 |

List of Tables

| | |
|---|-----|
| TABLE 1 - COMBINATIONS OF JOINT MOTION | 47 |
| TABLE 2 - DIRECTION OF JOINT TORQUE TO BALANCE USER-INPUT FORCE | 49 |
| TABLE 3 - SYSTEM PARAMETERS | 85 |
| TABLE 4 - SIMULATION GAINS | 85 |
| TABLE 5 - STATIC ERROR MEASUREMENTS | 89 |
| TABLE 6 - SIMULATION MODEL PARAMETERS | 103 |
| TABLE 7 - SIMULATION CONTROLLER GAINS | 103 |
| TABLE 8 - EXPERIMENTAL CONTROLLER GAINS | 109 |
| TABLE 9 - DENAVIT-HARTENBERG REPRESENTATION | 123 |
| TABLE 10 - SYSTEM PARAMETERS | 142 |
| TABLE 11 - COMPUTED-TORQUE PID GAINS | 142 |
| TABLE 12 - COMPOSITE ADAPTIVE CONTROLLER PARAMETERS | 162 |

Notation

The following notation is used throughout this thesis

| | |
|------------------------|---|
| a_n | Position of centre of mass of link n |
| \mathbf{B} | Torque conversion matrix |
| B_{ij} | Element ij of \mathbf{B} |
| \mathbf{C} | Manipulator viscous damping matrix |
| C_{ij} | Element ij of \mathbf{C} |
| \mathbf{C}_r | Referred manipulator viscous damping matrix from worm mechanisms |
| C_{rij} | Element ij of \mathbf{C}_r |
| C_α | Equivalent viscous friction coefficient of α -worm system |
| C_β | Equivalent viscous friction coefficient of β -worm system |
| C_θ | Equivalent viscous friction coefficient of worm-wheel system |
| C_{fa} | Viscous friction coefficient of α -worm system |
| C_{fb} | Viscous friction coefficient of β -worm system |
| $C_{f\theta}$ | Viscous friction coefficient of worm-wheel system |
| \mathbf{E} | Joint error vector |
| e_n | Joint error of link n |
| $F_{\backslash\alpha}$ | Reaction force at interface between α -worm and worm wheel |
| $F_{\backslash\beta}$ | Reaction force at interface between β -worm and worm wheel |
| $F_{\backslash\alpha}$ | Coulombic friction force between α -worm and worm wheel |
| $F_{\backslash\beta}$ | Coulombic friction force between β -worm and worm wheel |
| \mathbf{F}_u | Applied user-input force |
| $F_{u\max}$ | User-input force at which maximum end-effector speed is reached |
| $F_{u\min}$ | Minimum user-input force required to generate motion |
| F_{ux}, F_{uy} | Components of \mathbf{F}_u in x and y directions |
| \mathbf{F}_t | Acceleration force along the desired path |
| F_{tx}, F_{ty} | Components of \mathbf{F}_t in x and y directions |
| F_α, F_β | Equivalent force generated by torque from motors |
| \mathbf{J} | Manipulator arm Jacobian matrix |
| \bar{J}_n | Mass moment of inertia about centre of mass of link n |
| $J_{T\alpha}$ | System inertia with respect to α -motor torque |
| $J_{T\beta}$ | System inertia with respect to β -motor torque |
| J_α | Inertia of the α -worm and motor |

| | |
|---|---|
| $J_{\beta n}$ | Inertia of the β -worm and motor |
| $J_{\theta n}$ | Residual inertia of link n |
| $K_{d\alpha n}, K_{i\alpha n}, K_{p\alpha n}$ | Derivative, integral, and proportional controller gains for α -unwinding |
| $K_{d\beta n}, K_{i\beta n}, K_{p\beta n}$ | Derivative, integral, and proportional controller gains for β -unwinding |
| $K_{v\alpha n}, K_{v\beta n}$ | Composite adaptive controller gain for α and β -unwinding |
| l_n | Length of link n |
| m_n | Mass of link n |
| M | Manipulator inertia matrix |
| M_r | Referred inertia matrix from worm mechanisms |
| M_α | Equivalent mass of J_α |
| M_β | Equivalent mass of J_β |
| M_θ | Residual inertia matrix |
| M_ϕ | Equivalent mass of J_ϕ |
| $P(x, y)$ | Desired end-effector path in Cartesian co-ordinates |
| $P_{\alpha n}, P_{\beta n}$ | Parameter estimate gain matrix for α and β -unwinding |
| Q | Joint space representation of Cartesian path |
| q | Vector of joint positions |
| q_d | Vector of desired joint positions |
| r | Filtered tracking error vector |
| r_n | Filtered tracking error of joint n |
| $r_{\alpha n}$ | Radius of contact of α -worm with worm wheel for joint n |
| $r_{\beta n}$ | Radius of contact of β -worm with worm wheel for joint n |
| $r_{\theta n}$ | Radius of contact of worm wheel with α & β worms for joint n |
| T | Actuator torque vector |
| V_m | Matrix of Coriolis and Centrifugal terms |
| $W_{\alpha n}, W_{\beta n}$ | Regression matrix for α and β unwinding control of joint n |
| $W_{f\alpha n}, W_{f\beta n}$ | Filtered regression matrix of joint n |
| x | Cartesian position and equivalent linear displacement of α -worm |
| y | Cartesian position and equivalent linear displacement of β -worm |
| $Y_{\alpha n}, Y_{\beta n}$ | Regression matrix for filtered tracking error system of joint n |
| z | Equivalent linear displacement of worm wheel |
| γ_n | Worm lead-angle of joint n |
| ε_n | Friction constant for joint n |
| θ_n | Position of joint n |
| θ_{nd} | Desired position of joint n |
| $\mu_{d\alpha n}, \mu_{d\beta n}$ | Coefficient of dynamic friction of α and β worms of joint n |
| $\mu_{s\alpha n}, \mu_{s\beta n}$ | Coefficient of static friction of α and β worms of joint n |
| τ | Joint actuator torque vector |
| τ_{act} | Vector of joint torques obtained from strain gauges measurements |
| τ_{actn} | Measured torques of joint n |

| | |
|---|---|
| τ_d | Vector of torque disturbances |
| τ_{dn} | Torque disturbance of joint n |
| $\tau_{d\alpha n}, \tau_{d\beta n}$ | Torque disturbance of joint n expressed in terms of the α and β -motor torque respectively |
| τ_m | Vector of α and β -motor torques |
| $\tau_{m\alpha n}, \tau_{m\beta n}$ | α and β motor torque of joint n |
| τ_u | Equivalent torque generated by applied user-input force |
| $\Phi_{\alpha n}, \Phi_{\beta n}$ | Vector of system parameters for joint n |
| $\hat{\Phi}_{\alpha n}, \hat{\Phi}_{\beta n}$ | Vector of system parameter estimates for joint n |
| $\tilde{\Phi}_{\alpha n}, \tilde{\Phi}_{\beta n}$ | Vector of parameter estimate errors for joint n |

Chapter 1 Introduction

1.1. Background

Development of manipulators that interact closely with humans is increasingly a focus of research in fields such as robot-assisted surgery and haptic interfaces as there is a growing demand for robotic devices that directly interact with humans in these two fields. In particular, robotic devices have been introduced into the operating theatre to help the surgeon perform intricate invasive procedures with a higher degree of accuracy and reliability than would otherwise be possible with more traditional methods.

A wide variety of operations have benefited from the introduction of a robotic assistant. For example orthopaedic, neurological, cardiac, and cosmetic surgical procedures have all been enhanced by robotic assistance. However, many of the robotic devices currently in use are based on modified industrial manipulators and have typically been designed for high-speed, high-torque applications. The accuracy of such devices is unsurpassed but in the operating theatre, the use of high-powered mechanisms to perform these tasks has the potential to compromise safety of the patient, surgeon, and operating room staff.

Alternatives to these high powered robots have included custom built passive arms with dynamic constraints, which are unable to generate motion of their own. These systems require the surgeon to grasp a control handle mounted on or near the robot end-effector and provide the motive force. Computer controlled clutches or brakes limit the range of motion of the end-effector to constrain the tool to remain within a pre-defined safe working region. Although current devices have proved relatively successful in accomplishing this task, they have only had limited success in constraining the tool to remain along a pre-defined path as may be required by more complex machining processes such as bone milling and drilling involved in modern surgical procedures.

1.2. Aims of the Research

This research aims to address the issue of safety. By critically reviewing current developments in the field of surgical assistant devices it is possible to identify solutions to current difficulties in the areas of safety and control. Following the identification of these issues it has been possible to design a prototype device which could aid the surgeon in performing the Total Knee Replacement (TKR) procedure. The device and its associated control system demonstrate that the operation could be performed with greater accuracy and safety than is possible using traditional methods.

In the remainder of this chapter the TKR procedure is described. This is a typical orthopaedic procedure, which has been addressed a number of times with regard to robotic/mechatronic assistance. Shortfalls with current manual techniques are highlighted in order to illustrate the reasons why robotic devices are the preferred solution. The chapter concludes with a statement of the objectives of the research and the structural outline of this thesis.

1.3. Total Knee Replacement Surgery

Total Knee Replacement (TKR) surgery is a procedure in which the surgeon is required to cut a series of plane profiles into the surface of the tibia and femur in such a way that a prosthetic knee joint may be fitted. Accurate alignment and position of the prosthetic components is crucial to the success of the operation, i.e. painless and improved leg-motion for the patient. Misalignment or poor fitting of the prosthesis can also reduce the longevity of the replacement joint.

During TKR one of the common methods for orientation of the prosthesis components is to use joint-line theory [1]. Joint-line theory tries to ensure that after surgery the femur and tibia are aligned along the mechanical axis (a straight line that passes through the head of the femur, the centre of the knee, and the centre of the ankle), as shown in Figure 1. Some surgeons favour this method because of cruciate retention. In particular the Posterior Cruciate Ligament (PCL) is important to the stability of the knee and absorbs stress that might otherwise be transmitted to the bone interface.

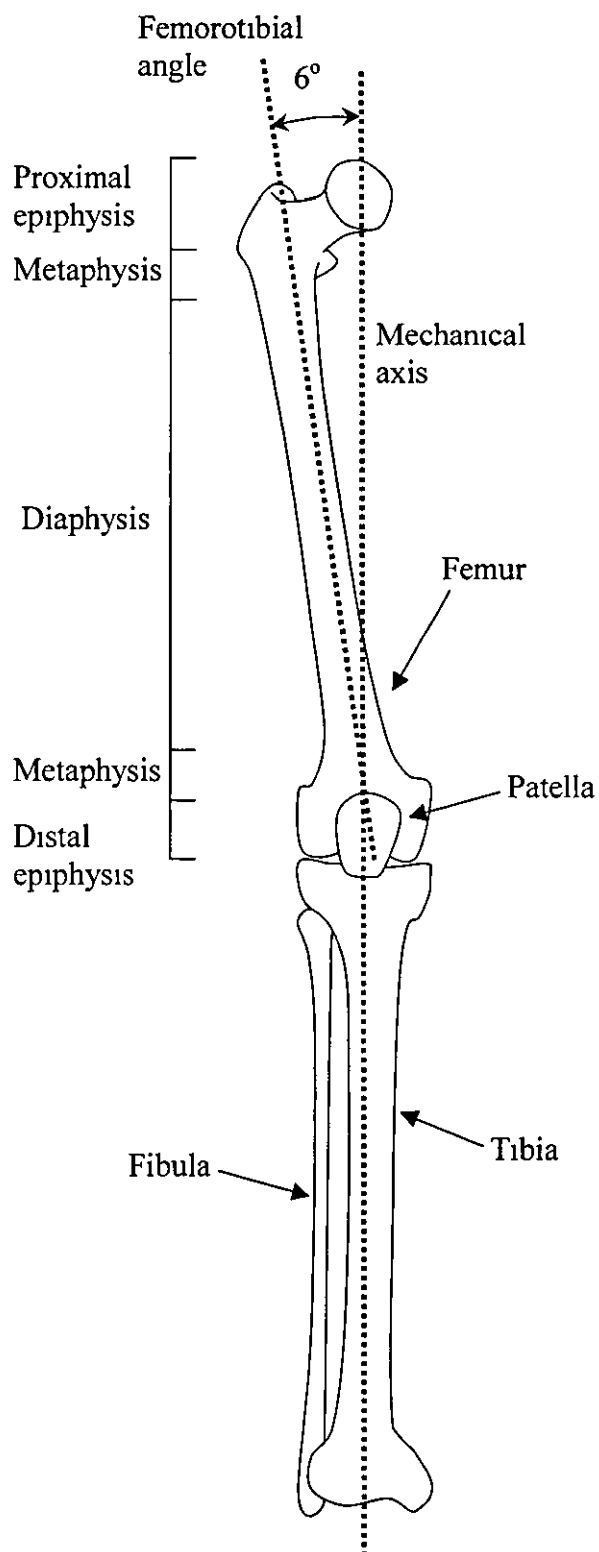


Figure 1 – Mechanical axis of right leg

Aside from the procedure required to gain access to the knee joint, the operation can be split into two parts: preparation of the femur for fitting of the femoral prosthesis, and preparation of the tibia for its associated component. The procedure for fitting the femoral component is shown in Figure 2. Firstly, the surgeon must drill a hole into the medullar cavity from the distal end of the femur, Figure 2a. Figure 2b shows that an intermedullary rod is inserted through the hole into the diaphyseal cavity, i.e. inside the shaft of the femur. The distal cutting jig is mounted on the rod perpendicular to the mechanical axis. Once the surgeon is satisfied that the jig has been aligned correctly, it is fixed in place with two Steinmann pins. The distal reference cut is then made using an oscillating saw, Figure 2c. This cut does not fix the position of the prosthesis; it generates a reference plane perpendicular to the mechanical axis from which all subsequent cuts will be made. It is at this point that the size of the prosthesis is selected and the surgeon aligns the appropriately sized chamfer-cutting jig in the correct position. Subsequently, the two anterior and two posterior chamfer cuts are made as shown in Figure 2d. The surgeon may also be required to mill small channels or drill holes for the locating features at the bone/prosthesis interface. The methods used to locate the prosthesis differ depending upon the manufacturer. In Figure 2e the femoral component is shown as having a single locating pin. In this case the surgeon would be required to drill a hole in the correct location, only then can the prosthesis be fitted.

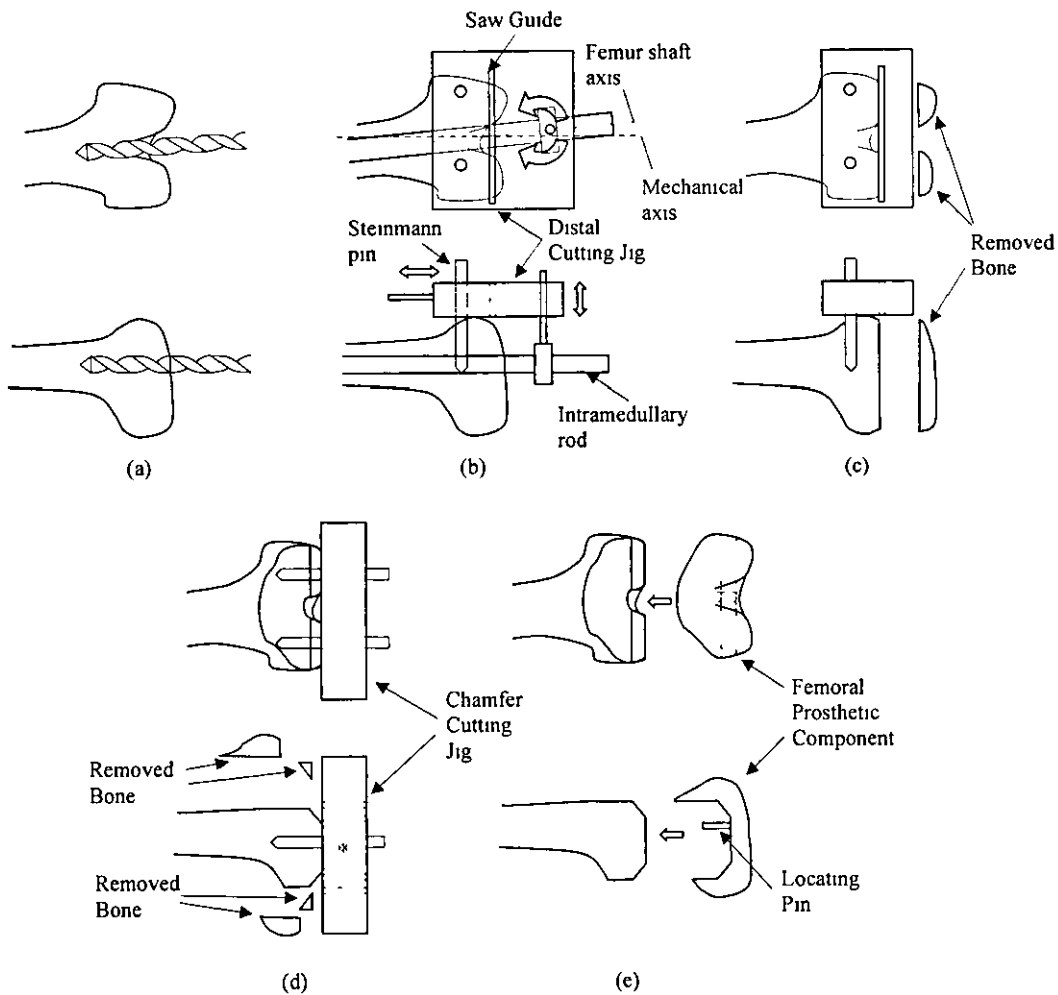


Figure 2 – Total Knee Replacement (TKR) bone removal and procedure for fitting the femoral prosthetic component

For placement of the tibial component, one plane surface cut is required. An external clamp is used to fix the distal end of an alignment rod to the centre of the ankle. The cutting jig is mounted at the proximal end and aligned to be perpendicular to the mechanical axis as shown in Figure 3a. Before the prosthesis can be fitted, the surgeon is required to mill the mating surface of the bone to accommodate the location profile of the prosthesis, Figure 3b.

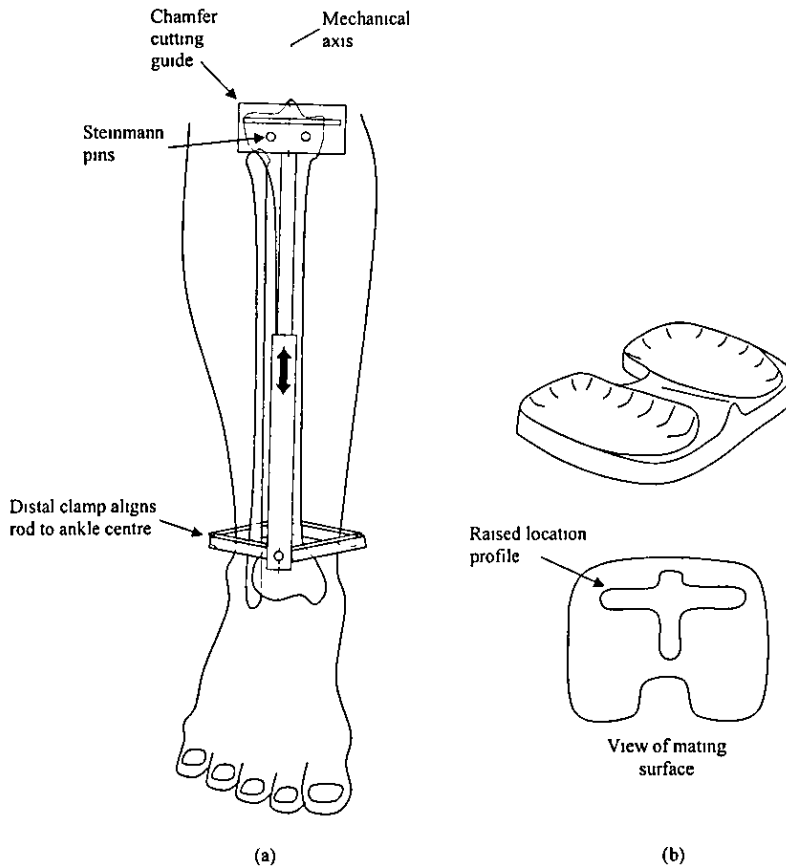


Figure 3 –Total Knee Replacement (TKR) bone removal and fitting of tibial prosthetic component

The above description of the TKR procedure is from the general point of view of bone removal. Issues concerned with preparing the joint for these procedures have not been detailed. The method is described with reference to the P.F.C.[®] Sigma Knee System from Johnson & Johnson Orthopaedic [2], which typifies in general the bone profiling requirement of TKR surgery.

1.4. Shortfalls of current procedures

The TKR procedure is made particularly difficult because of the jigs/templates and tools that the surgeon is required to use. Current methods used to place the new joint make misalignment of the prosthesis a common problem [3]. Poor jig-alignment is a

common problem and is compounded by the fact that accurate positioning of each subsequent jig is dependent upon the position of previous templates. It can be seen that in the case of Figure 2 above, the position and orientation of the chamfer cutting jig (Figure 2d) is wholly dependent upon the placement of the distal cutting template (Figure 2b)

Another critical aspect of the success of a TKR procedure is that of surface quality of the final cuts. This is particularly important in the implant of a cementless prosthetic component, where the bone is required to grow into the prosthesis in order to help maintain its position. A very common problem during the removal of bone is that even with jigs as a guide, the thin saw deflects away from the bone during cutting as illustrated in Figure 4

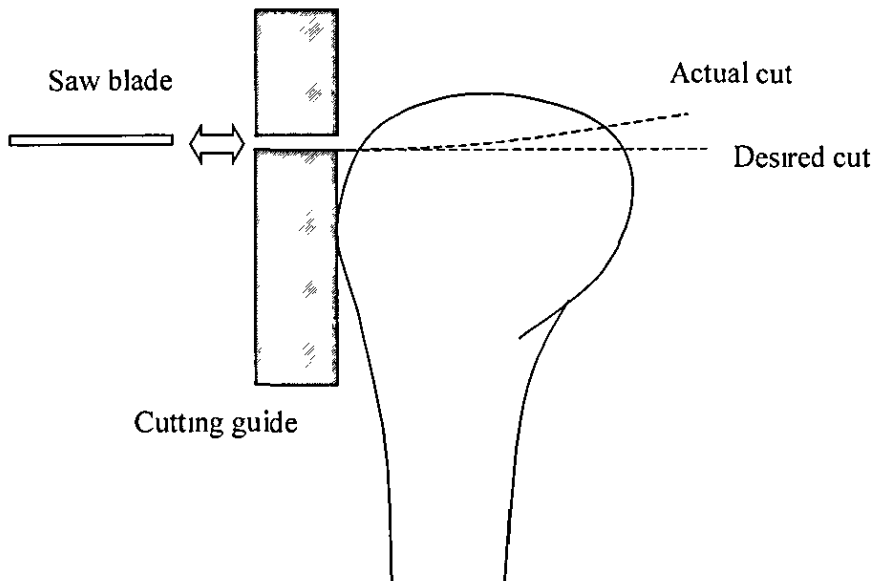


Figure 4 - Deviation of saw from cutting path

One solution to the problem of surface roughness and inaccuracy caused by the oscillating saw has been the introduction of milling tools. However, hand-held milling tools are very difficult to use because of the way in which the cutting force is transferred to the user. Even with jigs and guides the milling tool can 'bite' into the bone causing the surgeon's hand to be suddenly jolted. This can cause irreparable damage to the bone and surrounding tissue. Therefore, in order to increase the

accuracy with which the operation is performed robotic devices with milling tools mounted on the end-effector have been an attractive solution to these problems.

1.5. Organisation of the thesis

The review of the literature in robotic assistance in surgery is presented in Chapter 2. In this chapter, the current state of the art, both in terms of the solutions currently being developed and the procedures that are performed using existing devices, is illustrated. The review addresses the issue of safety of the patient, surgeon, and operating room staff, as well as the classification of the surgical assistant devices. Three passive robotic devices are identified in order to address the issue of safety. The chapter is concluded by a discussion on the limitations of the current state of the art and identification of the goals of this research.

In Chapter 3, three passive trajectory-enhancing robots are discussed in detail. The chapter aims to highlight current problems with the control and implementation of these devices. A mathematical analysis of a 2-DOF, revolute joint robot is performed in order to identify a solution to the problem of controlling the actuators to force the end-effector to remain along a pre-planned path.

Chapter 4 represents the design and development of the original concept into a 2-DOF manipulator. This is followed by a description of the electronic hardware and software in Chapter 5, which describes a mechatronic workstation that was developed as part of this research. The Personal-Computer (PC) based control system, running the QNX Real-Time Operating System (RTOS), is aimed at providing a platform for the rapid-prototyping of control algorithms and strategies for mechatronic systems.

In Chapter 6 the mathematical model of the joint mechanism is presented. Derivation of an accurate mathematical model enables the development of two types of control algorithms that are able to cancel backlash and regulate joint position. A computed-torque controller is developed in Chapter 7 and a composite adaptive controller is developed in Chapter 8. Both controllers are shown to accurately regulate position.

and cancel backlash despite the large friction forces inherent in the system. The two control algorithms and their results for a 1-DOF are discussed in Chapter 8

In Chapter 9, the mathematical model of a single joint is used to develop the dynamic equations for a 2-DOF manipulator. The control strategy for the 2-DOF system is presented in Chapter 10 for a computed-torque controller and in Chapter 11 for the composite adaptive controller

The conclusions of the research are stated in Chapter 12 followed by recommendations for future work in Chapter 13.

Chapter 2 Literature Survey

2.1. Introduction

The current state of the art of robotic devices designed to assist in surgical procedures is presented in this chapter. Firstly, the issue of safety is addressed. This is followed by a discussion on the classification of robotic assistant devices. Literature describing several devices relevant to this research is presented and critically reviewed. Furthermore, two haptic displays, used for providing tactile feedback to human operators, are reviewed because of their relevance to this work.

2.2. Safety Issues

Most of the robotic devices that are currently used to aid surgeons in performing operational tasks have been based on modified industrial manipulators. Industrial robots have typically been designed for high speed/high torque applications making them an inappropriate choice for use in the surgical environment without the introduction of complex safety strategies.

The manufacturers of industrial robot systems generally recommend that the manipulator be enclosed in its own cell thus protecting the human workers from the threat of injury by means of an interlocked safeguarding boundary [4]. The institution of such safety measures is not feasible for surgical applications. However, other issues on safety can be ported from existing industrial standards although the concerns about the reliability of these practices are augmented. To this end, a number of papers have been written in an attempt to address the issues of introducing a robot into this safety-critical environment.

Taylor et al [5, 6] describe the safety procedures implemented on the Total Hip Replacement (THR) system named ROBODOC which uses a modified industrial

manipulator to mill a cavity for the implant of a cementless prosthetic hip. They defined four safety requirements for a surgical robotic system. These are.

1. The robot should never lose control of its motions.
2. The robot should never exert excessive force on the patient
3. The robot's cutter should stay within a predefined workspace
4. The surgeon should be in complete control of the robot's actions at all times

Papers by Davies [7, 8] expressed concerns about the introduction of industrial robots into the surgical field. These articles do not only question the strength, speed and potential unreliability of the robots available on the market, but also questions the legal liability of robot manufacturers. Davies argued that when a surgeon uses a robot assistant and the surgery fails, the surgeon, (unless it can be proven that he was negligent), is not totally at fault. This may result in legal action being taken against the manufacturer of the robotic device. It can be seen that failure on the part of the robot raises issues other than human injury. Bearing this in mind, it is quite obvious why the robot industry would decline to let their robots be used in such a sensitive environment. A purely fictional, but pertinent, account of what might happen from a legal stand point is made by Epstein in "The Case of the Killer Robot" [9]. Epstein's story focuses on the ethics of the computer science industry. In this example a single programmer is indicted for manslaughter because a section of code that he wrote was alleged to have caused the robot operator's fatal accident.

The use of industrial robots in this field also raises the question of autonomy and how much the surgeon should be relieved of manual control. One method of safety enhancement is to make the surgeon confirm every action the robot is to make. This ensures that the surgeon has the final decision as to whether to proceed with the next step of the operation. This provokes the question, 'At what point should the surgeon confirm motion?' If the surgeon were required to confirm every minute motion, the time taken to perform the operation would increase. As a result, the patient would be under anaesthetic for a longer period and potentially at greater risk from infection. However, if the surgeon is not allowed to confirm small enough steps the system becomes more autonomous, thus escalating the problem of safety in favour of the

robot. A careful trade-off must be made in order to ensure that introduction of safety measures in one area does not cause the safety of the patient to be compromised

In 1994, the EEC Directive on Medical Devices put strict limits on the design of medical devices [10]. There are some very important points that can be directly applied to the design of robotic surgical assistant devices. The directive states

"Devices must be designed and manufactured in such a way as to protect the patient and user against mechanical risks connected with, for example, resistance, stability and moving parts "

Using these guidelines it should be possible to design a safe robotic system for surgery. Industrial robots are becoming more frequently used to perform tasks in an interactive manner with humans

One issue of safety that has not been discussed so far is that of maintaining a sterile environment in the operating theatre. Since the proposed research will produce only a laboratory prototype, it is not important at this stage to consider in depth the need for a sterile robot. However, there are two common methods of ensuring that the surgical tools are disinfected: these are physical and chemical. Typically only the physical methods are used owing to the fact that chemical methods cannot guarantee instruments are disinfected [11]. The physical methods include

1. Dry heat. The component to be sterilised is subjected to 160°C heat for 1 hour.
2. Dry steaming under pressure also known as autoclaving: The item is subjected to 135Kpa, 121°C for 15 minutes or 220Kpa, 134°C for 3 minutes

These physical methods of disinfecting surgical tools can be detrimental to most robotic actuators and sensors such as motors and encoders. There are a variety of components on the market that can withstand the high temperatures that need to be endured during the cleaning process. However, these components are generally more expensive than their non-autoclavable counterparts. A more common solution, in robot-assisted surgery, is to use a sterile drape that can be disposed of once the operation is complete. The drape (usually plastic, paper, or cloth) covers the robot

(except for the end-effector) ensuring that sterility is not compromised. The end-effector, the only component that actually comes into contact with the patient, is usually sterilised by autoclaving and attached to the robot without compromising the drape's ability to maintain sterility.

2.3. Classification of robotic surgical assistant devices

Many systems have been developed for a variety of different surgical procedures but until recently, classification of these aids appears to be somewhat of a grey area. Davies [7] proposed a potential hierarchy of systems for holding and manipulating surgical tools used the terms *passive* and *active* to describe a system's interaction with the patient and the way in which the tools are used. A powered robotic manipulator is used in a *passive* manner if it is implemented to align a mechanical guide or to place jigs. In this case the robot is relied upon for positional accuracy but the tool interaction is controlled and applied by the surgeon. Conversely, Davies states that a manipulator is used in an *active* manner if the robot makes physical contact with the patient, i.e. cutting bone with a milling tool or removing tissue with a probe.

However, Cinquin et al [12] used the terms *passive*, *semi-active* and *active* to describe the way in which the surgeon interacts with the system. Cinquin's classification of surgical aids stretches further than just robots. A *passive* system is depicted as one in which the surgeon guides the tools. Six degree-of-freedom (6-DOF) pointing devices, which are unpowered articulated arms that register position, fall into this category. *Passive* devices include hand-held instruments that are located by visual, ultrasonic and electro-magnetic techniques. Augmented Reality (AR) systems, in which video, Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) data is enhanced through computer graphics techniques, are used to display intraoperative data and aid in tool guidance are also classified as *passive*.

A *semi-active* system is described as one in which the surgeon is guided through a procedure aided by mechanical jigs or laser guidance. The jigs or laser could possibly be controlled by a powered robot. Cinquin et al also employ the term *semi-active* to describe a class of manipulator that uses mechanical constraints to limit the

workspace of the robot enabling the surgeon to move the robot around freely within a predefined region. As the end-effector approaches the boundary surface, freedom of motion in certain directions is either reduced or eliminated completely. The term *active* describes a system as one in which tasks within the operation are performed by an autonomous robot under the supervision of the surgeon.

From the point of view of robotic devices in general, neither Davies' nor Cinquin's use of the terms *passive*, *semi-active* and *active* is prudent. Like Cinquin et al, DiGioia, Jaramaz and Colgan [13] also state that surgical assistants can be separated into three distinct groups: *passive systems*, *semi-active systems*, and *active systems*. The meaning of each of the terms is stated in relation to the way in which the system physically interacts with its environment.

Davies later abandoned the six level hierarchy that was proposed in [7], adopting instead DiGioia's classification system and added two further definitions: *synergistic* and *telemanipulator* systems [14]. A *synergistic* system is one in which the surgeon physically interacts with the robot. By grasping a control handle mounted on the robot, the surgeon applies a force to move the manipulator. A computer controls the actuators of the manipulator in order to force the surgeon to remain within a predefined region or along a pre-defined path. A *telemanipulator* on the other hand is a manipulator that is remotely controlled by the surgeon. The surgeon uses a joystick (or other control mechanisms) to command the robot's motion.

It is DiGioia's classification system that will be adopted in this research with one adjustment: the term *semi-active* shall be replaced by *actively constrained*. Each category shall be discussed in more detail below offering examples of systems that represent the state of the art (i.e. systems that have been developed and those that are currently being researched). The addition of the terms *telemanipulator* and *synergistic* is superfluous as these systems all fall into one of the three categories given by DiGioia.

2.4. Passive Systems

A surgical assistant shall be described as *passive* if the system has no control over the position of the tool. The tool is guided under the power and judgement of the surgeon alone. The assistant system merely monitors tool position and offers advice on preferred procedure. This type of system is of little help for reaching a pre-determined position or following a trajectory

Simon et al [15] presented a system to aid the surgeon when placing the acetabular implant during THR surgery. The navigational aid contains elements typical of this genre of system regardless of the procedure. The system comprises an optical localiser, a visualisation aid (computer and monitor) and the surgical tools required to complete the procedure. The optical localiser is a system in which a tool is mounted with several LEDs and a series of cameras track and locate the tool. The visualisation equipment is used to display information to the surgeon about the patient (i.e. 3D model derived from CT data), tool position, desired tool position and trajectory. The system then guides the surgeon through the placement of the implant in the optimum predefined position and orientation with the aid of the optical localiser.

Delp et al [16] and Leitner et al [17] describe a passive system for Total Knee Replacement (TKR). Delp describes a system in which CT data is taken to generate a 3D model of the patient's femur and tibia. The planning system then orientates the prosthetic components with respect to the mechanical axis of the limb. An intra-operative system is then used to locate the tibia and femur in order to guide the cutting jigs into place. Once the jigs are in place the bones are cut to shape using a standard oscillating saw. The placement and tracking of the jigs are monitored via an optical localiser system. A graphical user interface (GUI) is employed to indicate position relative to a target point to ensure accurate placement of the jigs.

Tonetti et al [18] describe the application of an image-guided system to screw placement for repair of pelvic fractures in a non-invasive manner. The novel use of an ultra-sound scanner to register patient position to pre-operative CT data is the focus point. An optical localiser tracks the position of the scanner and sixty ultra-sound

images and scanner positions are recorded. Commercially available surface based registration software is used to relate the two types of data. The GUI displays two cross hairs to the surgeon. One cross hair represents the position of the end of the drill bit, while the second corresponds to alignment. Two small, concentric target circles are also shown to indicate the desired trajectory. The goal for the surgeon is to align the drill bit so that the two cross hairs are in the centre of the smallest circle. Once this has been accomplished the surgeon can start the drilling procedure. During the operation the surgeon keeps track of performance via the cross hairs to ensure that the trajectory stays true.

Langlotz et al [19] also demonstrate that an image guided system can be used to perform pelvic osteotomies. System registration is performed in a more common manner than is accomplished by Tonetti. The operation cannot be performed in a non-invasive manner, therefore the surgeon has visual and physical access to the bone so that it can be probed and matched to the CT model resulting in the registration of the system. Again, a custom GUI is used to guide the surgeon through the operation.

Augmented Reality (AR) is a new technique for surgical data visualisation. It is a method that involves combining information from the real world with computer generated graphics. Blackwell, Morgan and DiGioia [20] discuss the techniques that they used to implement a Three-Dimensional Image Overlay System at Carnegie Mellon University. In a similar manner to some of the systems that have been discussed so far an optical localiser is used to track tools, jigs and equipment. However, in this case the surgeon, wearing a pair of polarised glasses, views the patient through a semi-transparent 'window' (a half-silvered mirror) onto which the computer generated image is projected. The type of data that is typically displayed on such a screen is desired tool position, 3D images of the bone structure under the surface of the skin and tool penetration depth. To ensure that the surgeon's view of the patient and the projected images match up, the localiser must track the position of the surgeon with respect to the patient and the view screen.

2.5. *Actively constrained systems*

The term *actively constrained* generally refers to manipulators that restrict the surgeon's motions in some manner. The surgeon provides the motive force for the tool while computer controlled actuators limit the motion. The actuators cannot impart a force to move the tool, i.e. the active part of the system can only resist motion. Common examples of actuators for this genre of manipulators are electro-mechanical, pneumatic and hydraulic clutches and brakes.

The classic example of an actively constrained system, often referenced in the literature, is a 2-DOF prototype manipulator known as PADyC [21, 22]. In 1994, Lavalée and Troccaz obtained a European patent for PADyC [23], and in 1995, a US patent [24]. The manipulator is essentially passive except for a pair of motor driven over-running clutches on each joint. This original design also included a brake for limiting the speed of joint rotation and completely locking the joint. Troccaz also patented the free-wheel clutch in Europe [25] and in the United States [26]. The clutches allow free rotation in one direction but can constrain motion in the other [27]. Referring to Figure 5, which shows a single joint of PADyC, one clutch controls joint motion in the clockwise direction while its counterpart restrains motion in an anti-clockwise direction. The speed at which a clutch is driven determines the maximum velocity in that direction. The motors drive the clutch wheels through a worm gear to prevent the surgeon from back-driving the mechanism.

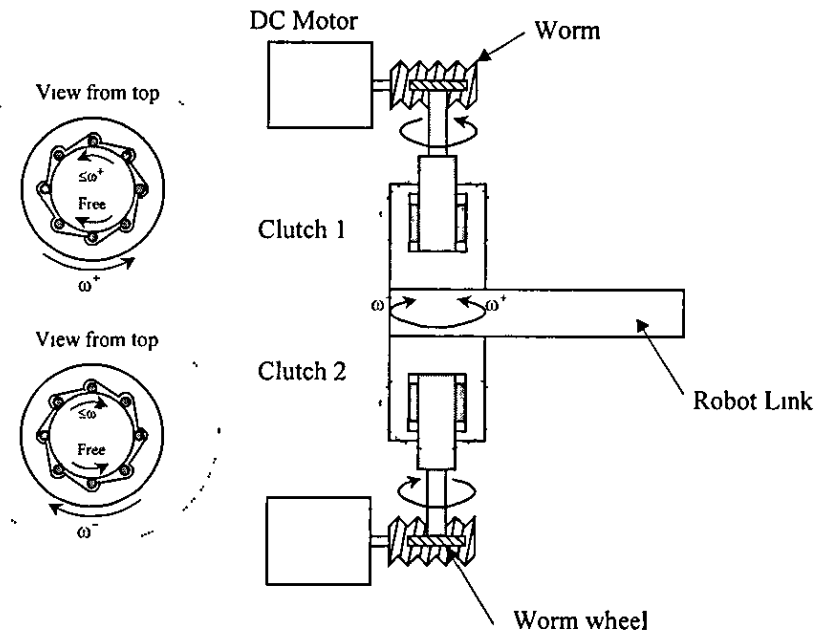


Figure 5 - PADyC joint architecture

Delnondedieu and Troccaz define four modes of operation for the manipulator. The first is 'free' mode in which the arm can be moved without any resistance. The second is 'position' mode, whereby the surgeon moves the manipulator to a pre-determined position. The clutches then lock, restraining any motion. 'Trajectory' mode provides the third manner in which the manipulator's motion may be constrained. In this mode the manipulator is allowed to move only along a specified trajectory. The authors actually state that the manipulator is constrained to remain within a corridor and that varying the width of the corridor increases the accuracy of execution of the path. The final mode is that of 'region' control. In this mode any number of control points on the tool (or arm itself) can be controlled to stay within a given boundary. Troccaz and Delnondedieu [28] state that the system had insufficient positional accuracy. The end-effector positional error was reported as being as high as a few millimetres. The inaccuracy of the system was attributed to four causes:

1. flexion in the axes between the freewheel and the link,
2. flexion in the freewheels,
3. backlash in the worm-gear drives, and
4. hysteresis of the freewheels.

If this is the case it is not just that the robot is imprecise; the robot is, essentially, free to move within a small area even though the clutches are attempting to restrain motion. If the accumulation of the errors across only two joints is a few millimetres, a 6-DOF system might exhibit errors of a few centimetres. Troccaz ran experiments that indicate that the error attains amplitudes of 10mm for an end-effector force of 10N and 20mm for an input force of 20N. In each of these cases, 5mm of the error is attributed to backlash with the remaining error coming from the elastic properties of the system. Owing to the relevance of this device to this research and the problems that are associated with forcing a manipulator of this type to track a desired path, PADyC is described in more detail in Chapter 3.

More recently, a 6-DOF system based on the PADyC prototype has been proposed for cardiac puncturing [29, 30]. In general the manipulator will only be used as a positioning device and not used in trajectory mode. The six axes will be used to attain the desired line of the trajectory, only then will the surgeon be asked to insert the needle. It was proposed that the speed of insertion would be controlled in part by only one axis of the system.

Phillips et al [31] and Viant et al [32] proposed CAOS (a Computer Assisted Orthopaedic System) to implement a straight-line trajectory typical of drilling operations. The system comprises three parts; CT data capture, trajectory planner and a 6-DOF, custom-built passive manipulator. Once the CT image has been taken the trajectory planner is used to decide upon a trajectory for the drill bit. The surgeon moves the manipulator into position then computer-controlled brakes lock the joints. At the tip of the robot is a mechanical guide through which the drill bit is fed. With the brakes turned off the surgeon is expected to align two cross hairs on a computer screen, a task which requires training and is not straightforward. These cross hairs represent a desired position and location.

The group looked at two designs of electromechanical brake: fail-safe and non-fail-safe. Due to a thirty percent increase in weight, the fail-safe option was discarded which introduces some interesting safety issues. For example, if a power failure

occurs during the drilling procedure, the weight of the falling manipulator may break the drill bit leaving part of it inside the patient

2.6. Active systems

An active system is defined as one that is able to perform position and trajectory tasks under its own power regardless of its proximity and interaction with the surgeon and patient. In the literature, ROBODOC is the most common example of an active system. However, there are many other active systems that warrant review if not only for the safety issues that they raise but also because of the variety of solutions that have been proposed.

ROBODOC [33, 34] is a five-axis industrial robot that has been modified to perform the machining of a femoral cavity to receive a cementless hip prosthesis. Cementless prostheses require high accuracy cavity machining compared to cemented prostheses. Figure 6 below shows how the surgeon is required to machine the femur in order to fit the prosthesis. The planning and operating procedure to machine the femoral cavity has five main steps. Firstly, the patient has three fiducial markers attached to the bone in a pre-operative procedure. A CT scan of the bone is then taken. In the third step, a 3-dimensional computer model of the femur is used to allow the surgeon to plan the position and orientation of the prosthesis with reference to the fiducial markers. During the operating procedure the position of the robot has to be determined with respect to the fiducial markers. Once this has been established the procedure can begin and the cavity automatically machined by the robot. The justification for the use of an automated system is that ROBODOC can reproduce the cavities to a dimensional accuracy far exceeding that of any surgeon. The increased accuracy alleviates problems of intraoperative fractures (caused by using excessive force to insert a prosthesis into a badly broached cavity) and reduces gaps between the prosthesis and the bone surface that may prevent bone growth.

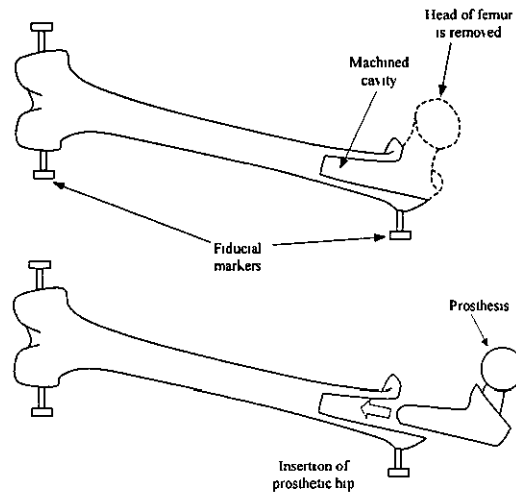


Figure 6 – Cementless total hip replacement procedure.

The safety features on the first generation ROBODOC assistant are described in detail throughout the texts by Taylor, Paul, Kazanzides et al [5, 6, 33, 34]. To ensure that the robot makes no unauthorised motion, the authors proposed a robot controller that performs consistency checks on position and velocity limits as well as monitoring external signals. A separate force monitor system was employed to ensure that the cutting tool was not exerting excessive force during the procedure. The force monitor then passed the information on to the robot controller so that erroneous conditions could be dealt with appropriately. Bone motion monitoring was employed in order to ensure that the bone does not move in relation to the robot base. If the bone is displaced during the surgical procedure the robot needs to adjust its frame of reference accordingly. A redundant position monitoring system that uses Light Emitting Diodes (LEDs) mounted on the robot and a series of cameras that track/monitor the LEDs was also proposed in order to track the robot end-effector and check that the cutting tool did not deviate from a predefined workspace.

After the successful trials on domestic canine subjects some of the main safety features implemented on ROBODOC had to be amended before human clinical trials could be performed. The original system used a visual-tracking system (OPTOTRAK™) to redundantly check the robot's position, ensuring that the robot end-effector was where it was reported to be. The surgeon concluded that the added complexity of the visual-tracking system was not worth the possible gain in safety in

the veterinary cases [5] but it was necessary in human clinical trials. The system was changed to accommodate a second set of position encoders on the robot arm, i.e. two encoders on each joint [35, 36]. A dedicated safety processor monitors the redundant encoders and compares the values with the motion controller readings from the original encoders to locate errors. Figure 7 shows the ROBODOC system in detail. The bone is attached to the robot base via a fixation device and the bone motion monitoring system checks for significant movement [37]. The safety processor also monitors bone-cutting force to ensure that the robot is performing correctly.

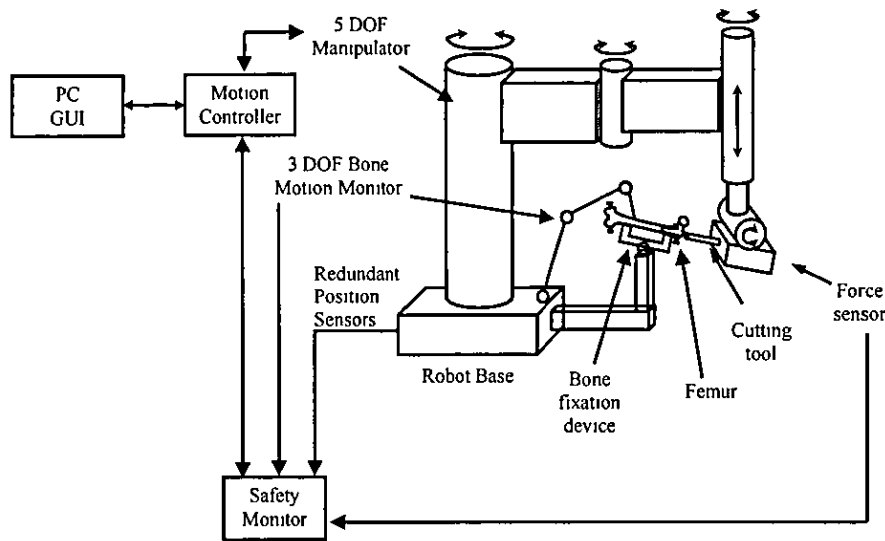


Figure 7 - ROBODOC system

In 1992, the ROBODOC Surgical Assistant System was allowed a 10-patient pilot study, authorised by the United States of America Food and Drug Administration (FDA). The study sufficiently demonstrated the safety of the system and after a review the FDA authorised a three hundred patient study in September 1993. The study comprised two randomly assigned groups of one hundred and fifty. A control group would undergo the hip replacement operation using standard manual techniques, while the study group would be operated on with the aid of ROBODOC. The robot system was also used in Germany. During the period of November 1994 to November 1997, 900 patients were operated upon [38]. Bargar et al [38] stated that during the first 100 operations the operating time decreased steadily because they were all performed by the same surgeon. However, four more surgeons joined the

team and the operation time remained in the range of 90-100 minutes. The procedure performed by the robot took approximately thirty minutes longer than the manual operation. This time loss was attributed to fixation of the patient's femur and registration of the robot using the fiducial markers. The extra time that must be taken to complete the operation is itself a danger to the patient; experiencing increased blood loss and exposure to infection. Since 1994, ROBODOC has performed over 7,300 THR operations and has also been used to perform over 100 Total Knee Replacement (TKR) procedures and has still not received FDA approval. There are currently 37 ROBODOC systems worldwide, including Japan and Europe. In December 2000, ROBODOC began clinical trials in the United States in order to achieve FDA approval.

Although the ROBODOC project successfully implemented an industrial robot as a surgical assistant many researchers in robot assisted surgery believe that custom made robotic devices are a safer way forward. Davies et al [39, 40] designed the Surgeon Assistant Robot for Prostatectomy (SARP) which was later to become known as PROBOT. It is claimed that this was the first active robotic device to perform tissue removal on a human patient in April 1991 [40]. A study was first carried out to determine whether a robotic prostatectomy was feasible by using a six-axis Unimation Puma industrial robot. The industrial robot was slightly modified to be able to reliably cut a conical cavity required for prostate resection. Davies demonstrated the feasibility of the robot by cutting a conical shape from a potato mounted in a Perspex box. A 3-DOF 'manual safety frame' was then designed that would allow a surgeon to make the cuts required to perform the operation manually. Wickham et al had previously obtained a European patent for this device [41]. The objective of designing a frame for a manual procedure was to test the kinematic geometry of the robotic device, and to identify problems with surgical procedure. The custom-made frame was completely motorised following clinical tests on 30 patients [40]. In its fully active form the system had performed five prostate resections on human subjects successfully. The first patient was operated upon in April 1991, which was reported to be the first robotically assisted removal of substantial quantities of tissue from a human patient [40].

CASPAR (Computer Assisted Surgical Planning and Robotics) system from OrthoMaquet also performs THR and TKR as well as Anterior Cruciate Ligament (ACL) reconstruction [42]. The robot is based on a PUMA industrial manipulator, which has a spherical workspace. It uses milling and drilling tools to perform the aforementioned types of surgery. The results of the first 43 TKR procedures performed with the CASPAR system showed that 97% of the prostheses were aligned to within the target $\pm 3^\circ$ compared to 61% in the manual procedure [43].

Davies et al [44] also designed a 4-DOF robotic system called ACROBOT (Active Constraint Robot) to enhance TKR surgery. ACROBOT constrains motions made by the surgeon to a pre-defined region. Geared DC-servo motors provide the necessary joint torque to constrain the end-effector to remain within the pre-planned region when the user applies a force on the end-effector/tool holder [45, 46]. The ACROBOT end-effector is moved via a control handle that senses the surgeon's input force and feeds the signal to the motor's control system. Depending on the proximity of the tool to the boundary the Motor Controller either allows motion of the tool/end-effector (far from the edge of the boundary) or resists motion (close to the periphery). The operation of this system does however, raise some important safety issues. Firstly, Davies classifies this system as semi-active because of the way in which the robot guides the surgeon. However, Harris et al stated that should the surgeon put enough force onto the end-effector to push the robot into a safety critical area, high-gain position control is used to move the cutter to the nearest point on the boundary, i.e. moving it out of the dangerous region [47]. This statement not only implies the patient may be injured by an over enthusiastic surgeon forcing the robot beyond the safe region, it also indicates that the surgeon is at risk because the ACROBOT will actually drive against any input force to return the robot to the safe region.

Kienzle et al [48, 49] also demonstrated the use of a 6-DOF Unimation PUMA 560 industrial robot, equipped with a 6-DOF force sensor and drill guide, for performing TKR surgery. In a similar manner to the ROBODOC system, the surgeon needs to place five fiducial markers in the patient's femur and tibia. A CT scan of the bone is taken to construct a 3-D model. The model is used to plan the placement of the prosthetic components. During the surgery the procedure requires that the femur and

tibia be immobilised using specially designed fixation devices. Registration of the robot with respect to the 3-D computer model is achieved by locating each of the five fiducial markers. Once this has been accomplished the robot is then used to drill holes in the femur and tibia so that mechanical jigs can be screwed into place. Using the jigs, the necessary cuts are made by the surgeon using the traditional hand-held oscillating saw. One point of discussion related to this research is the use of the original Unimation control system without any safety modifications. As has already been mentioned, industrial robots are prone to making unforeseen motions. Without implementing stringent safety measures this system cannot be expected to be used in an operating theatre.

MINERVA, a 5-DOF manipulator designed and built at the Swiss Federal Institute of Technology [50], is capable of performing biopsy, hematoma evacuation and living cell implantation upon the human brain. The objectives of the MINERVA robot are to cut operating times from two hours down to thirty minutes whilst at the same time increasing the accuracy of the operation. The whole procedure is performed by the MINERVA system, including skin incision, bone drilling and probe manipulation. Glauser et al [51], demonstrated the system performance with a series of eight biopsy procedures. The authors admitted that two of the operations were classed as failures. The first was attributed to the head of the patient being fixed too low in the stereotactic frame, thus making it impossible for the robot to reach the target area. The second operation failed because the robot retrieved tissue from around the tumour rather than tissue from the tumour itself. Operating on the brain is an extremely delicate procedure, robot failure could seriously injure or kill the patient. Injury caused by a robot such as ROBODOC, performing cutting operations on bone, may be corrected, unlike operations on critical soft tissue such as the heart or brain. However, the potential gain caused by the introduction of an active system with the positional accuracy demonstrated by that of MINERVA (ranging from 0.1mm to 0.5mm) outweighs fears of failure, assuming that a suitable safety strategy can be employed. A system similar to that of MINERVA, is NEUROMATE from Intuitive Surgical Systems [52]. Unlike ROBODOC, NEUROMATE has FDA approval and with 16 units worldwide the robot has supported over 3,000 neurosurgery procedures.

Brandt et al [53] presented a novel reduced workspace robot for general orthopaedic surgery named CRIGOS. The manipulator based on a Stewart Platform similar to that shown in Figure 8, has a much smaller workspace than the robots that have been discussed so far. The robot is manoeuvred into the general position required for performing the operation on a second floor-based mobile robot. Again the procedure requires a registration process to connect robot position to patient position. In comparison to more traditional robots the Stewart platform has a very limited workspace. The advantages of this particular robot, which make it a good candidate for implementation, are its size and weight. The robot is reported to weigh four kilograms and have a workspace of $100 \times 100 \times 50 \text{ mm}^3$ making it portable and easy to store. Due to the parallel nature of the system, CRIGOS appears to be safer than the other (serial) robots. Since more than one axis must move to significantly change position or orientation of the end-effector, the robot's ability to exhibit unauthorised motion is greatly reduced. However, the use of a large floor-based robot introduces large amounts of flexibility in to the system. In order to overcome this problem, a prop is fixed to the operating table once the floor robot is in position that ensures movement caused by flexibility is kept to a minimum.

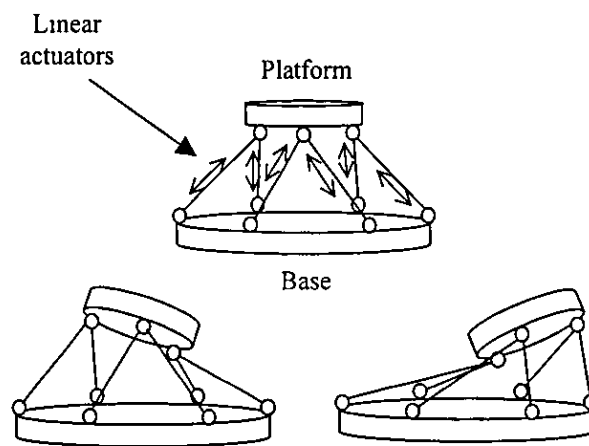


Figure 8 - Stewart platform

Brett et al [54, 55] showed the development of an automated device for drilling through small bones in the middle ear. A Stapedectomy is performed to restore hearing lost through bone tissue growth around the stapes (Osteosclerosis). The

stapes is the last bone in the ossicular chain, the series of three bones that transmit the acoustic vibration from the outer ear to the fluid of the inner ear. To restore hearing a small hole, 0.6mm in diameter, is drilled in the stapes bone and a prosthetic piston is inserted into the hole. The outermost end of the piston is mechanically attached to the incus bone and hearing is restored. The difficulty in this procedure is the detection of breakthrough of the drill bit as it passes through the bone. If the membranes behind the stapes bone are penetrated, loss of hearing could result. During manual drilling it is hard for the surgeon to detect and react to breakthrough thus making the procedure dangerous. The problem is made increasingly difficult by the deflection of the bone during drilling. Any force applied to the stapes causes the bone to move further into the ear also pushing the membranes back. Upon breakthrough the stapes and associated membranes spring back to their original position leaving the drill bit behind to penetrate the advancing tissue. Brett et al's 1-DOF 'micro-drill' monitors feed force, drilling torque and feed position to minimise breakthrough distance of the cutting bit. Once the onset of breakthrough has been detected the drill bit stops rotating and retreats until there is no feed force. The robot-assisted procedure involves visually positioning the drill, which is mounted on a passive arm equipped with pneumatic brakes. When the drill has been positioned correctly the brakes are locked and drilling can begin [56, 57, 58]. Results of operations performed on cadaver specimens showed that the protrusion of the bit was within 0.02mm of the bone surface. The authors admit that there are some issues to be resolved concerning hygiene, safety, and operating room compatibility. It has not been reported that clinical trials have been undertaken.

PAKY-RCM (Purcutaneous Access of the Kidney – Remote Centre of Motion) developed in part at the Brady Urological Institute, Baltimore, USA, is a system proposed for percutaneous renal access for biopsy [59, 60, 61]. The robotic mechanism uses a total of 11-DOF in order to position and orient a needle. The system consists of a 7-DOF passive arm, a 2-DOF remote centre of motion (RCM) module, and finally a two DOF PAKY end-effector. The first 8-DOF are used in the initial positioning of the device in the general vicinity of the target point and are firmly locked during the operation. The remaining 3-DOF are used to accurately target the desired needle trajectory. Once the robot has been positioned correctly, the

needle is automatically inserted. A CT scan is obtained in order to verify the needle position and the biopsy is taken

Bouazza-Marouf et al [62, 63], using a custom built 5-DOF manipulator, investigated the use of a robotic drilling device to aid the repair of trochanteric, neck and shaft fractures of the femur. The main difficulty is that the surgeon is operating in a 'blind' manner. In the case of trochanteric or neck fractures, the surgeon is required to attach a plate to the diaphysis of the femur and a sliding screw into the head of the bone as shown in Figure 9. The procedure requires that the sliding screw be inserted at the correct position, orientation and depth. The surgeon uses a guide wire to drill a pilot hole and is forced to take several X-ray images in order to monitor progress. Several attempts (depending on the experience of the surgeon) may be made before the surgeon is satisfied that the guide wire is correct. This may cause weakening of the structure of the cancellous bone inside the femur head whilst also exposing the patient and surgeon to excessive doses of X-ray radiation. The system of Bouazza-Marouf et al uses a C-arm X-ray unit and a calibration frame to take two nearly orthogonal views of the area to be drilled. The calibration frame is used for the computation of the transformations from the X-ray image frame of reference to the robot's frame of reference. The robotic assisted procedure requires the surgeon to highlight the desired drill bit trajectory in the two X-ray images. The drilling trajectory with respect to the robot frame of reference is computed and the robot end-effector/drill is automatically positioned. The surgeon, through the use of a dead-man's-handle, initiates the automatic drilling procedure. Additional safety enhancements involve the monitoring of the drilling force. Figure 10 shows the fixation of a shaft fracture; the procedure involves the surgeon inserting a long steel tube (intermedullary nail) into the medullary cavity of the femur. Following nail insertion, proximal and distal locking screws are to be 'blindly' placed in alignment with the respective holes in the nail. The proximal screw can be easily placed with the assistance of a jig, but the surgeon uses X-Ray images to obtain a view of the distal nail holes and aligns the drill bit. Again, the surgeon may take several attempts to line up the pilot hole with the hole in the nail increasing patient and surgeon exposure to X-ray radiation. Using the robotic drilling system developed by Bouazza-Marouf et al it is possible for the surgeon to drill the holes in the correct position at the first attempt using only two X-ray images.

No clinical trials have been carried out as the system is under going further improvements to enhance the Man-Machine Interface.

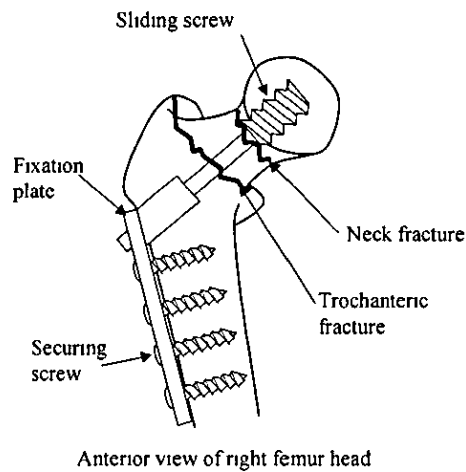


Figure 9 - Fixation of trochanteric and neck fractures

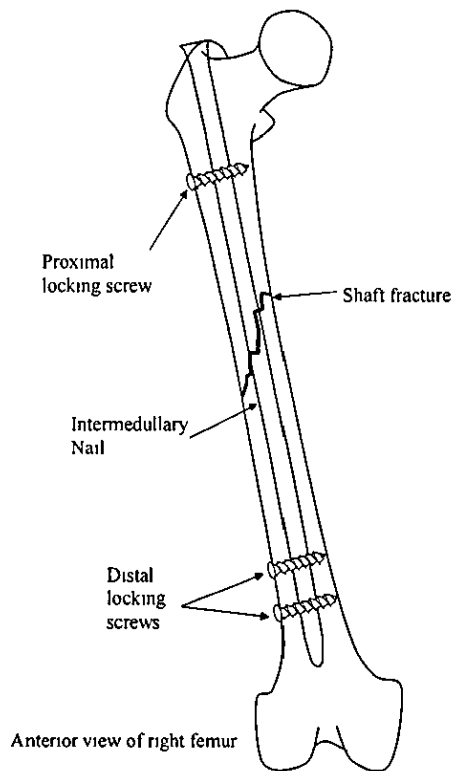


Figure 10 - Fixation of shaft fracture

Finlay also described a system for assisting in the drilling procedure [64] ORTHOSISTA was designed to place a guide cannula in a position and orientation that would allow the surgeon to drill the hole with the least possible exposure to X-ray radiation. The mechanism used a pair of 2-DOF Cartesian manipulators. Each manipulator manoeuvred an end of the guide cannula. By moving the Cartesian manipulators, the cannula could be placed at the desired position and orientation. The Cartesian frame was found to be too large and impeded the surgeon's view of the patient. The robot configuration was not developed any further.

Loughborough, Hull and Dundee Universities have also developed a robotic system for performing femoral and tibial osteomies as part of a research project involving the three institutions [65]. The tibial/femoral 'wedge' osteotomy procedure involves removing a section of bone in order to realign the tibia and/or femur. Figure 11 shows a typical femoral osteotomy procedure. Figure 11a shows the incorrect alignment of the tibia and femur. To correct the deformity a wedge is removed as shown in Figure 11a and the bone surfaces are closed together and the bone fixed with a staple as shown in Figure 11b. Traditional methods involve the surgeon cutting the wedge 'free-hand' using an oscillating saw. The saw vibrates causing a bad surface finish, which can cause further alignment problems. The feasibility study, involving the three universities and the Department of Traumatology and Orthopaedics of Royal Hull Hospitals, demonstrated that a custom-built 6-DOF active robot could be used to automatically cut the two plane profiles under close control of the surgeon.

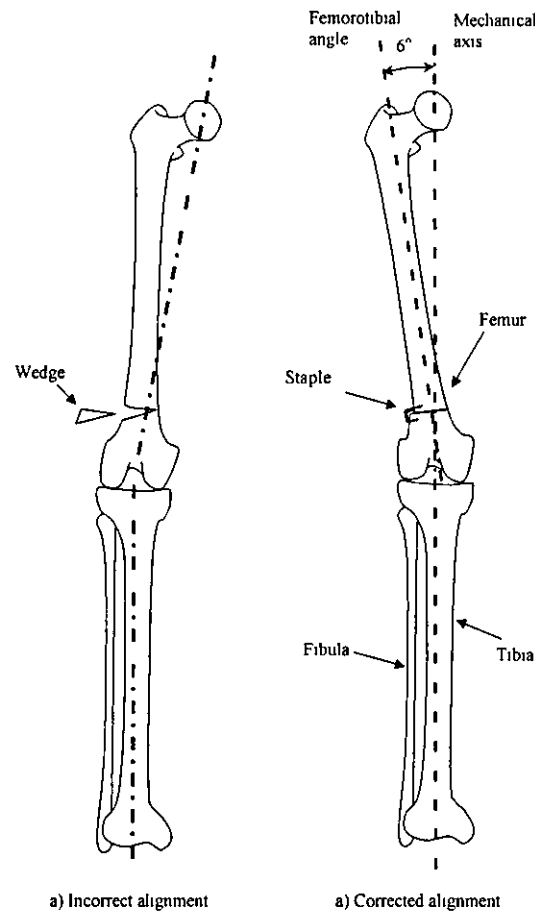


Figure 11 - Osteotomy procedure

In 1998, the 6-DOF robot assistant device OTTO was presented [66]. The robot, which hangs from the ceiling of the operating room, handles drills, saws, tappers, and screwdrivers, and many other tools that are required during surgery. OTTO is capable of helping the surgeon perform a wide range of procedures including plastic and reconstructive surgery, as well as dental and laser surgery. The robot was reportedly the first active system for performing Maxillofacial surgery.

Sackier and Wang [67] adapted a 6-DOF manipulator to holding a laparoscope (a device for viewing internal tissue through a small incision in the skin in minimally invasive surgery). The robot, known as AESOP, replaces a human assistant who would ordinarily hold and direct the scope under the surgeon's command. In some cases the surgeon might take manual control of the scope; leaving only one hand to perform the rest of the procedure. However, the robot was initially controlled via a

footpad allowing the surgeon to be in total control of the placement of the scope while both hands are free to perform the operation. Later developments included a voice command system for the control the robotic Laparoscope. Taylor et al also proposed a similar system, known as LARS, which was being developed as part of a joint study by IBM and John Hopkins University [68]. A third automated system for holding a Laparoscope is EndoAssist from Armstrong HealthCare [69]. The camera picture is displayed on a monitor and the surgeon, wearing a special headband, controls the position of the camera by looking at different areas of the image. Motion of the robot is not allowed unless a footswitch is activated. Another robot designed for the task of holding an endoscope is Endoxirob supported by Sintors S A. [70].

The AESOP system was further developed by Computer Motion Inc into the telemanipulator system known as ZEUS Robotic Surgical System [71]. The system has two robotic arms that operate in conjunction with AESOP Laparoscope. ZEUS operates in a master-slave mode; the surgeon sits at the remote terminal maneuvering the master robot. The slave copies the master's actions inside the patient through small incisions in the chest (in the case of heart surgery). The surgeon watches the operation on a monitor at the terminal displaying images captured by the endoscope. The ability of the robot to scale down the surgeon's motions into 'micro-movements' makes the task of stitching and other difficult procedures a more simple and accurate task.

Similarly, the da Vinci system [72, 73, 74] is a teleoperated minimally invasive surgical system comprising three robot arms and a separate control terminal. One arm is equipped with an endoscope, while the other two carry instruments. The system operates in a master-slave mode; the surgeon sits at the remote terminal manoeuvring the master robot. As with ZEUS, the slave end-effector, inserted inside the patient through small incisions in the chest, copies the master's actions inside the patient through small incisions in the chest. The surgeon watches the operation on a monitor at the terminal displaying images captured by the endoscope. On June 18, 1999, Intuitive Surgical Systems announced that the da Vinci system had been FDA approved. Up until this point, the system had performed nearly one hundred cardiac operations and over one hundred and fifty minimally invasive procedures of other

natures. There are currently 40 such systems operating world-wide performing a wide variety of minimally invasive surgery.

Smith et al [75] presented a robotic system for performing a cut in the cornea. The system uses a robotic mechanism to guide a water jet around the desired profile of the cut. Once the cornea has been removed it can be grafted on to another patient's eye. The need for accuracy and repeatability is obvious in this case. The two cuts have to be as similar as possible in order to avoid deforming the graft and distorting the vision of the recipient. The use of a robot in this way would allow the surgeon to plan the cut of complex shapes from both the donor and patient cornea. As a result, the cornea would not be distorted during the replacement procedure.

HIPPOCRATE from Sinters S A is a 6-DOF manipulator for placing ultrasound devices to an accuracy of 0.1mm [76]. The robot can apply a controlled pressure to the patient in order to obtain repeatable readings of blood pressure and to evaluate the onset of arteriosclerosis. It has also been proposed that HIPPOCRATE can be used for automatically taking skin samples from patient. Sinters S A have also been developing light weight robots known as TER and TERESA for the ultrasound scanner application.

Armstrong HealthCare Limited have also developed a robotic system for guiding the surgeon through neurosurgical procedures [77]. The 6-DOF device, known as PathFinder, is one of the most recent devices to be unveiled. The manipulator can be used as a tool for positioning drill guides with sub-millimetre accuracy. The manipulator can also be fitted with motorised tools so that it can perform operations automatically. Firstly, several titanium fiducial markers are attached to the patient's skin with surgical glue or attached directly to the skull. Once in the operating theatre, the surgeon uses X-ray, CT or MRI data to plan the operation selecting both the entry point and target area on the patient's skull. The robot, using a camera mounted on the end-effector moves towards the patient to search for the fiducial markers. Upon finding a marker the robot measures the position of the marker from several different angles in order to get an accurate reading of position. After the robot has found the fiducial markers the skull position relative to the robot base is known. The

robot can proceed to position the drill guide (or other surgical tool) above the desired entry point on the patient's skull [78]. The robot is reported to be under going accuracy and safety trials at the Queen's Medical Centre in Nottingham with clinical trials scheduled early in 2002

2.7. Other relevant works

Another area of interest in terms of using robots for physical interaction with humans is that of *haptic displays*. Haptic devices are mechanisms that provide tactile feedback to a user. As with surgical systems, many of the research devices have used modified industrial manipulators to provide this feedback. Much of the research has involved making the end-effector of a robot behave like a different object, for example a tennis ball or a spring. However, in machining processes it is more important that the robot is capable of following a pre-defined trajectory. There are two particularly interesting projects in the field of haptic displays that are relevant in the context of this research.

A group at the Georgia Institute of Technology proposed a robot known as PTER (Passive Trajectory Enhancing Robot) [79]. The robot is based on a motorised active haptic display known as HURBIRT [80]. The main difference between the two robots is that PTER uses four clutches and a differential gearbox to dynamically link the rotational velocities between successive revolute joints by varying the braking force of the clutches. Although several control methods were applied, the accuracy of the manipulator was not proven to be satisfactory. Book et al showed that the manipulator was not able to track the trajectory any closer than about 4mm [79]. It was also found that in order to achieve acceptable performance, the clutches needed to be switched on and off very quickly which caused undesirable discontinuous motion of the manipulator [81].

Another example of a passive haptic device is the COBOT (Collaborative Robot) designed by Colgate et al [82, 83]. The principle of the COBOT is to implement the active part of the system as a "steering" mechanism. The prototype COBOT is a single steering wheel, running on a plane surface, supported by a passive 2-DOF

cartesian frame. The user moves the end-effector via a control handle mounted on the manipulator. The direction of motion is controlled by sensing the force perpendicular to the direction of wheel rotation and controlling the direction of the wheel by a servomotor and gear train. Colgate et al described two modes of operation, virtual caster and virtual wall. In virtual caster mode the steering behaves in a fashion that follows the direction of the input force. However, in virtual wall mode as the user approaches a predefined boundary the direction of travel is controlled to constrain the motion to the "free-space", prohibiting movement beyond the boundary. Colgate describes a 3-DOF system using two steering wheels in which the COBOT (still supported by the cartesian frame) is able to rotate around any given point in the plane. The authors also mention the possible addition of a third wheel that would enable the COBOT to stand freely without external support. The COBOT was originally designed as an alternative to haptic displays, devices to provide physical feedback for virtual environments, but the potential of these collaborative robots in other safety-critical settings was quickly noted but only mentioned briefly. However, in reality the problem (certainly in the surgical environment) will be to ensure that the wheels do not slide on, or lose contact with the surface of the plane that they work on. A version of COBOT, called "scooter", was used on an automobile assembly line for material handling [84]. The COBOT was used for carrying the doors of the vehicle and while the operator pushed, "Scooter" steered the door towards the assembly point. The COBOT architecture has been used to develop a continuously variable transmission (CVT) for revolute robot joints [85]. Most recently, a 3-DOF manipulator using the CVTs has been proposed [86]. The joints of the robot are kinematically linked, that is if the user forces motion of one joint the rest of the joints are forced to move in a direction that generates motion along the desired path. However, although this passive device promises trajectory tracking the size of the CVT mechanisms is large compared to the maximum force that they can resist. The robot is only expected to be able to withstand 13N of force (in its worst configuration) at the end-effector before the drive rollers in the CVT slip and the robot loses its ability to maintain the current position. These devices are also discussed further in Chapter 3.

2.8. Discussion

As robotic systems have been developed the commercial advantages of such systems are quickly realised. The need to apply for patents and the commercial sensitivity of these systems means that authors stop publishing relevant material. This has been particularly evident in the recent developments of the ACROBOT system. The robot has performed a cadaver study and is believed to have started trials on live subjects. However, it is felt that in general the above literature review is good representation of the current state-of-the-art.

As it can be seen from the literature, safety is of the utmost importance. This highlights the concerns of using modified industrial or custom-built active systems no matter how advanced the measures taken to prevent unauthorised motion. It is believed that in certain surgical applications an actively constrained system is the best compromise between accuracy and safety. In the context of the proposed application that is the focus of this research, actively constrained devices could potentially offer all of the accuracy of an active device whilst reducing the associated risk of injury to patient and surgeon that are inherent in the ROBODOC, MINERVA, and da Vinci systems. An actively constrained device also has the advantage over passive systems of increased accuracy and steadiness as well as reduced time in the positioning of the end-effector. It should be noted that between the work of Troccaz's PADyC and Davies' ACROBOT there is an opening for a safe accurate trajectory following system that has, until this point, not been filled. Troccaz's PADyC suffers from large positional errors that can reach a magnitude of 20mm and are attributed to backlash, hysteresis, and flexibility. It is also shown in Chapter 3 that the user requires an in depth knowledge of the behaviour of the system in order to efficiently track a path, i.e. the user needs to be trained. However, while Davies' active ACROBOT offers high positional accuracy the issues associated with the risk of injury to both the surgeon and the patient do not make it a very attractive solution.

2.9. Research Objectives

The objective of this research is to develop an inherently safe robotic surgical assistant. The device should be capable of performing path/trajectory following tasks that typify motions required during a TKR procedure with little error and no risk to the user. To this end the following seven requirements are defined in order to assess safety and performance of the manipulator and control algorithms

- Requirement 1 : The user should also not be able to overpower the manipulator with excessive force
- Requirement 2 : The robot should move only under the direct control of the operator
- Requirement 3 : The robot should not perform unauthorised motion.
- Requirement 4 : The manipulator should be unable to provide a motive force against the user.
- Requirement 5 : The control algorithms and strategies employed should be robust against external disturbances such as the user-input force or from the mechanical properties of the device itself.
- Requirement 6 : To prove performance of the control algorithm the manipulator should be shown to track a prescribed path with minimal error
- Requirement 7 : The manipulator must be easy to use with little or no training.

Chapter 3 The Limitations of Passive Arms with Dynamic Constraints

3.1. Introduction

Development of robotic devices that interact closely with humans has been a focus of research in fields such as robot-assisted surgery and haptic devices. The demand for robotic devices where direct interaction with humans is required is growing. The recent introduction of powered surgical-assistant devices into the operating theatre has meant that modified industrial robot manipulators have been required to interact with both patient and surgeon. Many companies and research facilities have developed completely autonomous, modified industrial manipulators for direct interaction with humans. In the operating theatre, this introduces many questions on safety of the patient, surgeon, and operating room staff [7]. There are many other applications where the accuracy afforded by a robotic mechanism coupled with the perception of a human could be an advantage, for example assembly tasks and material handling in the manufacturing industry are of particular interest.

Custom-built robotic-assistant devices, such as ACROBOT (Active Constraint ROBOT) [47] and PADyC (Passive Arm with Dynamic Constraints) [27], have been a more attractive alternative to the larger and more expensive industrial systems. ACROBOT is a four degree-of-freedom (4-DOF) manipulator designed to help the surgeon perform total-knee-replacement surgery. The surgeon controls the motions of the manipulator by applying a force to the control handle attached to the end-effector of the robot. All of the joints are backdriveable and a DC motor at each joint controls the resistive force that the surgeon feels using a force control strategy. The workspace of the robot is actively constrained to confine the end-effector to remain within a pre-planned safe working region. However, while the ACROBOT system requires physical input from the surgeon in order to move the manipulator, the force-controlled servomotors in each joint are still powerful enough to provide unaided motion of the robot. It is also possible for the surgeon to over-power the motors forcing the robot

end-effector to deviate from the desired path or leave the pre-defined safe working region. However, increasing the available motor torque raises many issues regarding surgeon and patient safety

Alternatives to these motor-driven devices are devices, such as PADyC, that cannot provide motion of their own, i.e. they can only provide a resistance to motion. It is shown in this chapter that for such devices a user is required to have a deep understanding of the mechanism in order to be able to perform computer-controlled constrained path following tasks in a smooth manner. A 2-DOF revolute joint, planar robot is used to demonstrate such a requirement. A geometric analysis of the conditions that the direction of the user-input force must satisfy in order for the robot to be able to follow a path is presented. It is shown that incorporation of these conditions into a control algorithm and implementation of a mechanism for presenting this information to the user would simplify trajectory following, thus allowing the user to concentrate on the end-effector task. It is also shown that a more appropriate choice, when possible, of robot kinematic configuration would alleviate these problems.

3.2. Actively-constrained devices for trajectory following

Haptic displays, used for providing tactile feedback to human operators, have also been the subject of recent research and again provoke many questions on the issue of user safety. The majority of robotic systems require a mechanism with some form of motor-driven joint that provides force-feedback to the user. However, in addition to PADyC there are only two other devices that make use of passivity to ensure user safety. These two devices, PTER (Passive Trajectory Enhancing Robot) [87] and COBOT (Collaborative Robot) [86] provide a very different approach to that of PADyC by forcing a kinematic relationship between joints, such that any user-input force in the general direction of the path will generate motion of the robot along the path.

3.2.1 PADyC

The prototype of PADyC is a 2-DOF robot incapable of motion of its own [27]. At every joint PADyC has a pair of over-running clutches, each running on a separate motor driven hub. By controlling the speed of each motor one clutch limits the maximum allowable speed in the clockwise direction while the second clutch restricts speed in the anti-clockwise direction. It was demonstrated that PADyC could be used to confine the motion of the surgeon to a pre-defined work area and, to a limited extent, follow a pre-defined path. With 20N of force applied on the control-handle, the 2-DOF prototype of PADyC exhibited up to 20mm of error at the tool-tip with link lengths of 0.25m [28]. The error was attributed to joint flexibility and backlash in the clutch mechanism. Errors of this magnitude cannot be tolerated in most surgical applications.

Further analysis of the control algorithm that is used to implement the path following mode reveals some interesting problems. Consider the circular path, $P(x, y)$, of Figure 12a, and its joint-space representation, Q , shown in Figure 12b. The method employed to control PADyC involves approximating the joint-space path with a straight line. By considering a new point a joint-space distance ΔQ further along the path, the amount that each joint needs to be moved to reach the new point can be calculated. Using this method, it is possible to generate an authorised rectangle (a region of allowable motion) as shown in Figure 12b. Figure 12a also shows the authorised region which is the mapped rectangle from joint-space into the task-space. The authorised region is calculated by the computer in real-time and is continually adjusted to compensate for the current position of the manipulator. It can be seen that if the choice of ΔQ is large then it is entirely possible that the current robot position may never lie on the path. However, if ΔQ is very small, the robot may follow the path with some small degree of error, i.e. guiding the user down a narrow corridor. It is claimed that this method has been proved to give satisfactory results despite the error induced by backlash and joint/link flexibility.

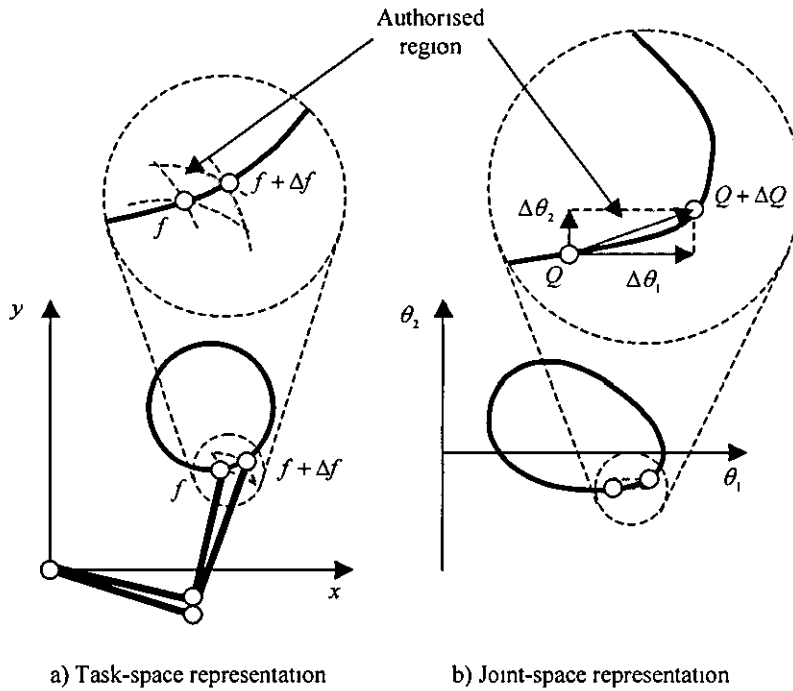


Figure 12 – 2-DOF manipulator following a circular path

In order for the robot end-effector to trace the desired path the clutches must impart a resistive force at one or more joints. In this case, if the user tries to exceed the maximum allowable speed of a given joint, the clutch suddenly engages and constrains the joint to remain at the given angular velocity. Any resistance to motion radically (and in this case instantaneously) changes the dynamic behaviour of the system transmitting an irregular 'feel' to the motion. The user is required to compensate for these changes and needs to adjust the direction that the driving force is applied in order to follow the path. In other words the user is expected to 'feel' their way along a narrow corridor, bouncing from wall to wall. In addition, some tasks, for example using a milling tool to machine bone or making an incision with a scalpel, require a smooth constrained motion to get the best results and much training would be required for the user to learn to operate the manipulator effectively.

3.2.2 PTER

PTER uses four clutches to control motion [87]. Two of the clutches connect each of the two controllable joints to a zero velocity source. A third clutch couples the rotation of these two joints in one direction, while a fourth clutch connects the rotation of the two joints in the opposite direction through a gear train. It was proposed that when the user applies an input force to the control handle, the voltages supplied to the clutches could be varied in order to allow motion in a given direction. PTER uses the clutches to dynamically generate forces/torques that regulate the joint velocities in an attempt to make a kinematic coupling between the joints. By controlling the clutches, each joint of PTER is coupled to all others kinematically. That is if the user forces motion of one joint the rest of the joints are also forced, using the coupling clutches, to move in a direction that generates motion of the end-effector along the desired path. However, it was reported that the method used was found to generate a discontinuous feel to the movements of the end-effector whilst tracking a circular path. The control of the manipulator was said to have a large trade-off between improved trajectory following and smooth performance due to the high-speed switching of the clutches. Preliminary results from the impedance controller of the redundantly actuated PTER mechanism showed that the end-effector attained errors in the order of 4mm [88].

3.2.3 COBOT

COBOT uses only a steering mechanism to guide the user along a desired path. A three-wheeled COBOT known as Scooter was designed and implemented to assist in material handling applications such as loading and unloading of vehicle doors in an automotive assembly line [89]. The vehicle door is loaded on to Scooter at a loading bay. The operator then moves the COBOT in the general direction of the target area. A computer, controlling the steering mechanism of the COBOT, is used to guide the user to the correct target point where the door is fixed to the vehicle. The COBOT concept was further developed to design a continuously variable transmission (CVT) [90]. It was proposed that the CVT could be used in a revolute-joint robot mechanism to control the relationship between joint velocities. The proposed method is particularly interesting in that each joint is coupled to the next by a drive shaft. By controlling two rubber steering wheels, the CVT gives the correct velocity coupling

between the input shafts. Using a CVT at each joint the user can be constrained to follow a path. The joints of the robot are mechanically constrained to give the kinematic solution for the required motion, i.e. by inducing motion in one joint the user forces the desired motion of other joints. More recently a 3-DOF, four link parallelogram arm using the CVT COBOT architecture has been proposed [86]. However, it is possible for the user to force the drive or steering rollers of the CVT to slip. Therefore each CVT has been designed to be quite large (using a 10.16cm sphere). Even with this redesign the maximum static loading torque that it will endure before slippage occurs is quite small. It is expected that the arm would only be capable of resisting approximately 13N of force in its worst configuration. Excessive wear of the rubber steering and drive rollers during normal operation is also of concern.

3.2.4. *Evaluation of existing actively-constrained devices*

Although there has been much research in this particular field, the success of trajectory following actively constrained manipulators has been limited. The COBOT principle gives good tracking performance. However, the ability to resist input forces normal to the path is limited, i.e. application of large input forces can lead to the end-effector leaving the path. On the other hand, while the clutch mechanisms of PADyC and PTER provide excellent holding torque, sufficient tracking performance and smooth operation has not been demonstrated. In comparison to the COBOT mechanism PTER provides excellent static holding torque in the order of 407Nm. The performance of PTER is degraded by the stick-slip problems associated with clutch/brake systems along with the dynamics of the actuators themselves. The PADyC system has exhibited mechanical problems attributed to backlash and flexibility inherent in the clutches of the prototype. There are also issues regarding ease of use for trajectory tracking applications. In the latest iteration of PADyC a 6-DOF SCARA type manipulator has been proposed for cardiac puncture [30]. However, the manipulator is not used for tracking trajectories using multiple axes.

In the remainder of this chapter an analysis of a 2-DOF revolute-joint manipulator is undertaken. It is shown that, for this configuration of passive manipulator with dynamic constraints, there are geometric conditions on the direction of the user-input force that should be applied in order to follow a pre-defined path. This method would allow the manipulator to track the desired path without complex gear and clutch systems or CVTs that impose a kinematic relationship between each of the joints

3.3. Analysis of a two link revolute-joint robot

Consider the 2-DOF, revolute-joint manipulator shown in Figure 13. The end-effector is required to track the path, $P(x, y)$, while the user applies a force F_u . The direction of F_u is restricted to some specific areas that are described in this section. The torques τ_1 and τ_2 applied at joints 1 and 2 respectively are generated by the joint actuators. Ignoring coulombic friction within the joints, it is possible to write the equation of motion for the manipulator as given by equation (3-1).

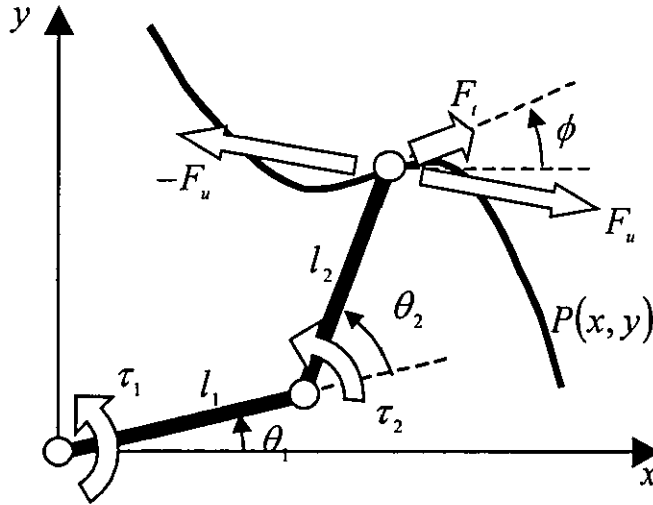


Figure 13 - Trajectory Following Using a 2-DOF Manipulator

$$\mathbf{M}\ddot{\theta} + \mathbf{C}\dot{\theta} + \mathbf{V}_n\theta = \mathbf{T} + \mathbf{J}^T \mathbf{F}_n \quad (3-1)$$

where

$$\mathbf{M} = \begin{bmatrix} a_1^2 m_1 + (l_1^2 + a_2^2 + 2l_1 a_2 \cos \theta_2) m_2 + \bar{J}_1 + \bar{J}_2 & (a_2^2 + l_1 a_2 \cos \theta_2) m_2 + \bar{J}_2 \\ (a_2^2 + l_1 a_2 \cos \theta_2) m_2 + \bar{J}_2 & a_2^2 m_2 + \bar{J}_2 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} c_{11} & 0 \\ 0 & c_{22} \end{bmatrix} \quad \mathbf{V}_n \theta = \begin{bmatrix} -m_2 l_1 a_2 (2\theta_1 \theta_2 + \theta_1^2) \sin \theta_2 \\ m_2 l_1 a_2 \theta_1^2 \sin \theta_2 \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

and

$$\mathbf{T} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad \mathbf{F}_n = \begin{bmatrix} F_{nx} \\ F_{ny} \end{bmatrix}$$

m_n , J_n , l_n , a_n represent the mass, inertia, link length, and position of centre of mass of link n respectively.

In order for the end-effector to move along the path, the joint actuators must each apply a torque that combine with the user-input force to result in a force along the desired trajectory. The torque applied by each joint can interpreted as a summation of torques that result in a force $-\mathbf{F}_n$ (that opposes \mathbf{F}_n) and another force that is directed along the desired path, \mathbf{F}_t , i.e.

$$\mathbf{T} = \mathbf{J}^T (-\mathbf{F}_n + \mathbf{F}_t) \quad (3-2)$$

where

$$\mathbf{F}_t = \begin{bmatrix} F_{tx} \\ F_{ty} \end{bmatrix}$$

If the manipulator is initially at rest, i.e. $\dot{\theta} = 0$, equation (3-1) may be written as

$$\begin{aligned} \mathbf{M}\ddot{\theta} &= \mathbf{T} + \mathbf{J}^T \mathbf{F}_n \\ &= \mathbf{J}^T \mathbf{F}_t \end{aligned} \quad (3-3)$$

The torque, T , that can be provided by the braking mechanism at each joint is, however, severely limited in the following way. If the sign of the joint torque is equal to the sign of the desired direction of motion of that joint, a motor would be required to generate the motion. However, if the sign of the joint torque is opposite to the sign of the desired direction of motion, a braking force is sufficient to control motion. This can also be expressed mathematically by stating that for a passive manipulator, motion can only be allowed if both of the conditions given by (3-4) hold true.

$$\text{sgn}(\tau_1) \neq \text{sgn}(\dot{\theta}_{1,d}) \quad \text{and} \quad \text{sgn}(\tau_2) \neq \text{sgn}(\dot{\theta}_{2,d}) \quad (3-4)$$

where $\dot{\theta}_{n,d}$ is the desired velocity of link n

It is possible to divide the workspace of the manipulator into four distinct areas based on the four combinations of joint velocities. Consider the manipulator shown in Figure 14. Each robot joint is allowed to move in either a positive or negative direction. Motion of the robot end-effector into the area labelled 1 in Figure 14 requires positive motion of joint 1 and negative motion of joint 2, i.e. $\dot{\theta}_1 > 0$ and $\dot{\theta}_2 < 0$. Area 2 illustrates negative motion of both joints, i.e. $\dot{\theta}_1 < 0$ and $\dot{\theta}_2 < 0$, and area 3 represents negative motion of joint 1 and positive motion of joint 2, i.e. $\dot{\theta}_1 < 0$ and $\dot{\theta}_2 > 0$. Finally, area 4 represents positive motion of both joints, i.e. $\dot{\theta}_1 > 0$ and $\dot{\theta}_2 > 0$. These four conditions are summarised in Table 1. Intuitively, it may be assumed that in order to constrain the end-effector to remain along the path by only applying a control torque with a brake/clutch mechanism it is sufficient that some component of the user-input force lies along the path. However, it can be shown that although this is the case, the conditions to deploy the end-effector along the desired path are more restrictive and are also dependent upon the position of the joints/links and the kinematic configuration of the robot.

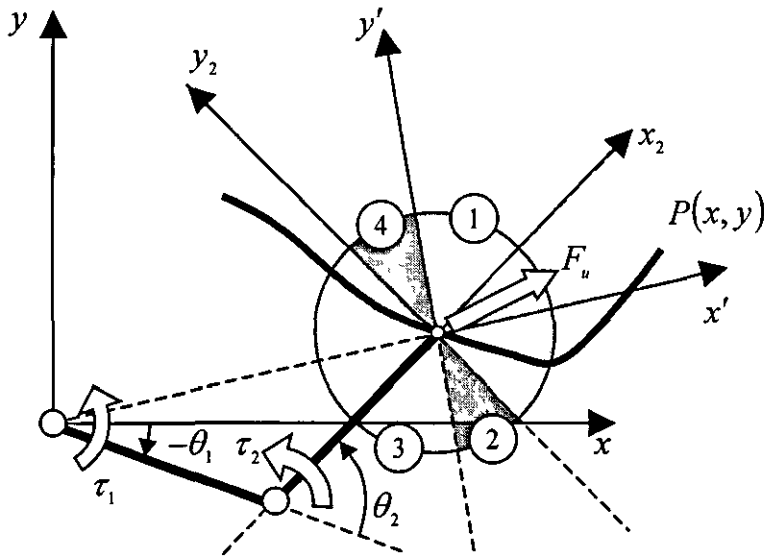


Figure 14 - Areas of motion for a 2-DOF manipulator

Table 1 - Combinations of joint motion

| Area of motion | θ_1 | θ_2 |
|----------------|------------|------------|
| 1 | + | - |
| 2 | - | - |
| 3 | - | + |
| 4 | + | + |

Using some examples, it is possible to show the conditions under which motion should or should not be allowed. Firstly, the magnitude and direction of the user-input force, F_u , is set to a constant value. Then by setting the magnitude of the desired acceleration force, F_a , to a constant and rotating it through 360° , it is possible to plot the joint torque required to generate the acceleration in the desired direction for any given joint configuration. Consider the manipulator with joint angles of $\theta_1 = 0$ and $\theta_2 = \pi/4$ rad and that the user force is directed at an angle $\gamma = -7\pi/36$ rad (or -35°) relative to the x-axis as shown Figure 15. Figure 16 shows the graphs of the joint torques required to cancel the user-input force and generate an acceleration force of

$|F_i| = \{0.01N, 1N, 2N, 5N\}$ in all possible path directions. Each graph of Figure 16 illustrates the areas 1, 2, 3, & 4 that represent the direction of motion of the joints. The graphs also show the division of the joint torque into four areas (A, B, C, & D) in a similar manner to the joint directions. These divisions are summarised in Table 2. It should be noted that the positions of the areas 1, 2, 3, & 4 in Figure 16 do not change because they are dependent only upon the joint positions. However, as the desired acceleration force increases the areas A, B, C, & D in Figure 16 do change. A cross on the x-axis of each graph shows the angular direction of the applied user-input force.

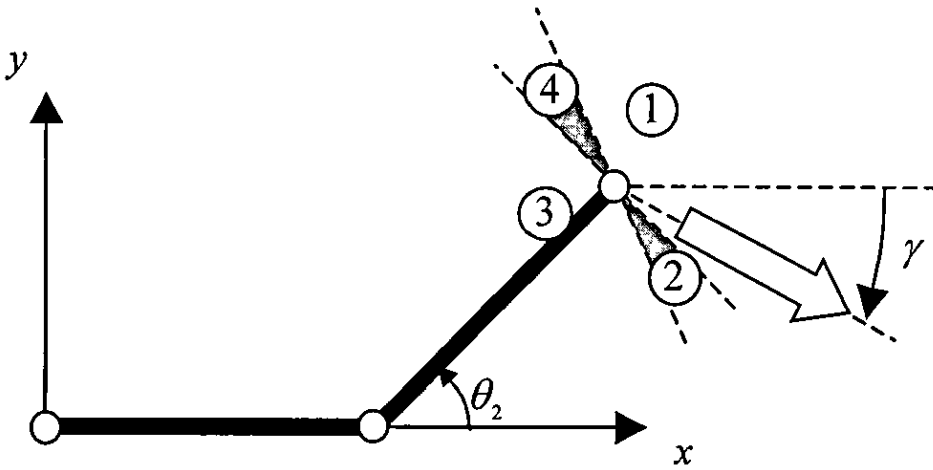


Figure 15 - Robot configuration for $\theta_1 = 0, \theta_2 = \pi/4$ and $\gamma = -7\pi/36$ rad

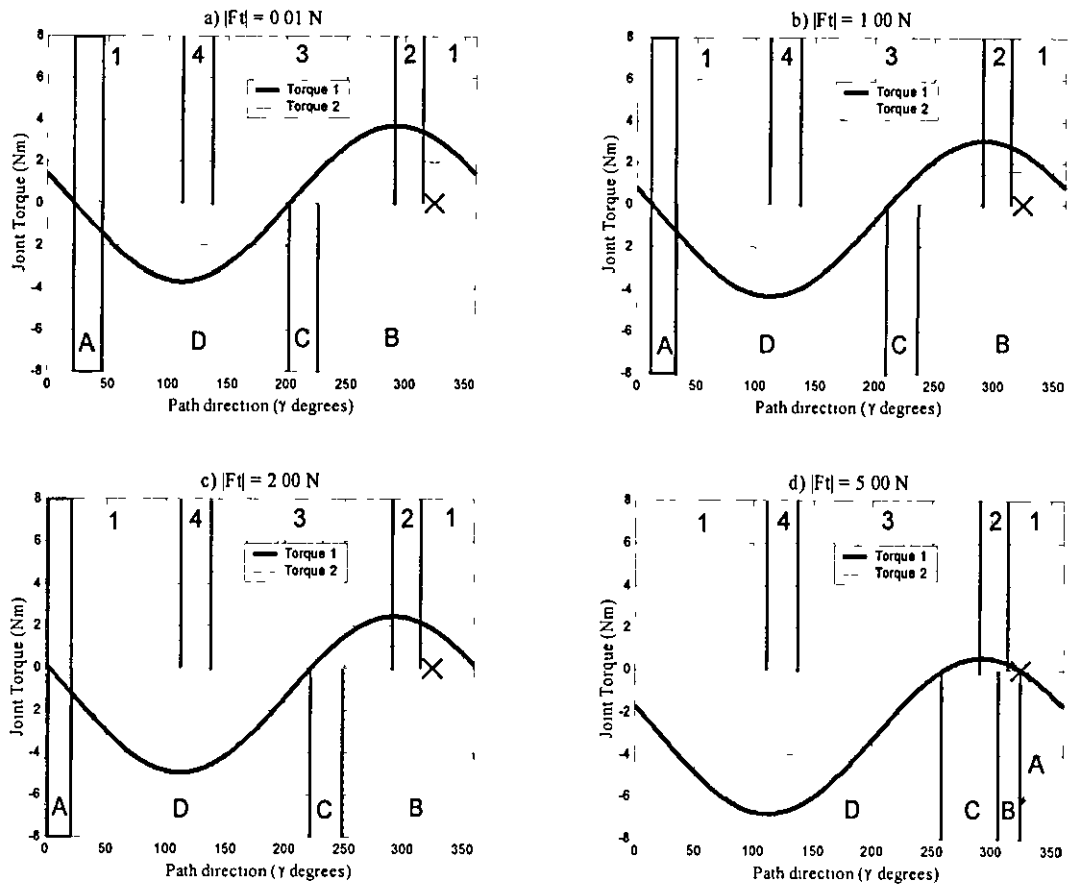


Figure 16 - Required joint torque for $F_u=5\text{N}$, $\phi=\pi/6$, $\theta_1=0$, and $\theta_2=\pi/4$

Table 2 - Direction of joint torque to balance user-input force

| Direction of joint torque | τ_1 | τ_2 |
|---------------------------|----------|----------|
| A | - | + |
| B | + | + |
| C | + | - |
| D | - | - |

By analysing the torque requirements given in Figure 16 and comparing these to the desired direction of motion it is possible to determine whether the torque required to restrict motion along the path is aiding or resisting motion. Using the conditions

given by equation (3-4) it must be concluded that, for $|F_u| = 0.01N$, motion of the passive manipulator can only be allowed where the areas A and 1 overlap as shown by the shaded area in Figure 16a. Similarly in Figure 16b and Figure 16c, the same condition must also hold true but for different ranges of path tangents. Figure 16a, Figure 16b, and Figure 16c are also interesting owing to the fact that motion in the direction of the applied user-input force is not allowed and can only be generated with the aid of motors. When the magnitudes of the desired acceleration force and the user-input force are equal as shown in Figure 16d, motion can only be allowed when the path direction is the same as that of the applied user-input force. In fact, it can be seen from Figure 16d that the only condition under which the user can push along the path is when the torque required from each joint actuator is zero, i.e. the desired acceleration is exactly equal (both magnitude and direction) to the user-input force. Hence, applying a resistive torque at the joints to direct the motion of the end-effector along the desired path is not always possible if the user-force is directed within a specific region.

3.4. Conclusions

The analysis of the 2-DOF revolute-joint robot has not included the dynamic forces induced by joint acceleration and coriolis effects. However, it has been shown that in this simple case, where these forces are small in comparison to the user-input force, it may not be possible to control the manipulator to follow the desired motion for a user-input that even lies in the general direction of the path. This result is significant in the context of the above-mentioned research. For the 2-DOF revolute-joint manipulator, it is more important to show the user how the force should be applied in order to achieve the desired motion than it is to show where the path lies. This problem is particularly evident in the control method of PADyC where the user is required to 'feel' their way along the path. The system suddenly changes from being a free pin-joint mechanism into a rigid body and it is the user that is required to calculate and supply a force that compensates for this. Such a problem can be avoided by showing the user the computed direction of the force.

The kinematic configurations of the manipulators depend on the application. However, it is often possible that different configurations could be used. Based on the analysis of the 2-DOF revolute-joint, it is possible to avoid the problem described above by redesigning the kinematic configuration of the manipulator when possible. For passive deployment it is best to use a combination of prismatic and revolute joints with intersecting joint axes, i.e. the joint z-axes. In this way the joint motion can be constrained independently, thus avoiding the deployment problem described above.

Chapter 4 Mechanical Design of a Prototype Robot for Passive Deployment Applications

4.1. Introduction

It has been shown in the previous chapters that the design of the joint mechanism and choice of actuators is critical to the overall safety design of the manipulator. In particular, the backdriveability of ACROBOT and the nonlinear clutching system of PADyC have been shown to present considerable problems in the areas of both safety and control. One of the design criteria of such a safety critical system is to ensure that the manipulator can maintain a desired position even under large external forces from the user with little or no power from the actuators.

It has also been shown that there are serious control issues in the use of brakes and clutches for a mechanism that is required to follow a given path. Therefore, by selecting a non-backdriveable gear train, the position of the end-effector can be maintained with little or no power from the drive actuator(s). By selecting a suitable control strategy (based on the results of the previous chapter) the end-effector can also be forced to track the desired path when the user-input force is in a specific direction. The deployment of the end-effector using the proposed system and control strategy is without risk to the user.

The proposed joint mechanism, shown schematically in Figure 17, consists of two worms each driven by a low power DC servomotor. The two worms follow a worm wheel that is fixed to the robot link. The user provides a force, F_u , on the end-effector and motion is allowed by controlling the two worms simultaneously. The non-backdriveability of the worm is achieved by the choice of lead angle, γ , which is less than the angle of friction [91]. That is

$$\gamma < \tan^{-1} \mu \quad (\text{F-1})$$

where μ is the coefficient of friction. In the case of a friction coefficient of $\mu = 0.1$, γ must be less than 5.71° . Therefore by selecting a worm with a lead-angle of 3.48° the mechanism is assured to be non-backdriveable for $\mu > 0.061$. Therefore no matter how much force the operator applies, motion is not allowed until both motors are controlled to move in the opposite direction. In Figure 17 the user-input force is shown as a torque τ_u , which is derived from the user-input force and the length of the link. The torque applied by the two worms α and β are indicated as $\tau_{m\alpha}$ and $\tau_{m\beta}$ respectively.

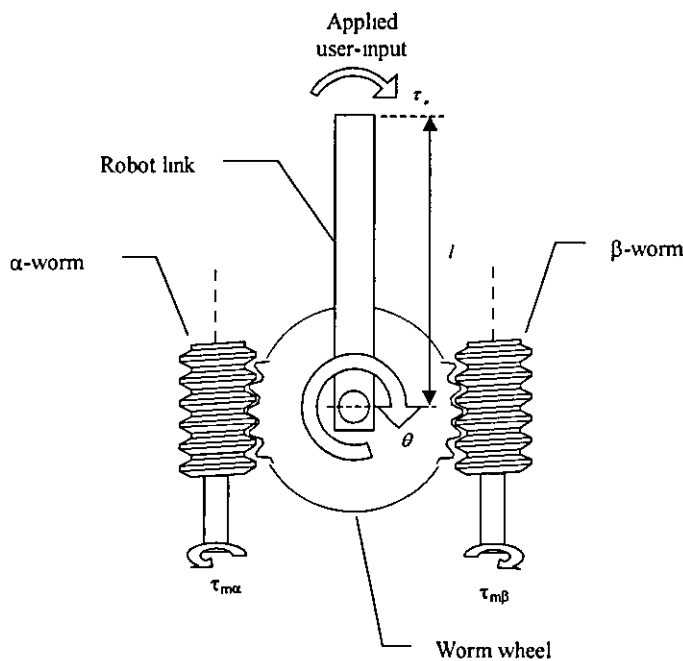


Figure 17 – Dual-worm drive joint

The parallel nature of this redundantly actuated joint structure makes this active system safer than modified industrial robots. If there was a failure, both motors would have to drive in the opposite direction for any motion to occur. However, if this failure occurs, motion of the end-effector would be slow, due to the high gear ratio giving the operator plenty of time to react, e.g. to release a dead-man's-handle.

The remainder of this chapter is dedicated to a discussion on the issues of backlash cancellation and design of 1-DOF and 2-DOF planar robotic manipulators. A control

handle with integrated pointing mechanism is also presented in order to indicate to the user the direction in which a force should be applied to accomplish the desired task. It is believed that this method of indicating to the user the direction the force should be applied with a control handle is novel

4.2. Backlash cancellation

In gear system design, a small amount of backlash is required in order to allow for thermal expansion, lubrication, and achieve acceptable frictional losses within a drive train [92]. In many applications large gear ratios are required which can mean that multi-stage gear trains are necessary. This can increase the effects of backlash. In many robotic applications, even very small amounts of backlash can cause errors in position that are not acceptable in demanding applications such as surgery. For example, if a single link of a robot arm exhibits ± 0.0087 rads (or $\pm 0.5^\circ$) of backlash there will be an error of ± 3.5 mm at the end of a 0.4m link. In order to reduce the backlash to an acceptable level, gear train housings must be manufactured to high engineering tolerances. This is expensive both in terms of the machinery required and time taken for manufacture. A common solution to this problem is the use of harmonic drives. Harmonic drives have excellent properties in terms of robotic systems with little and high efficiency. However, this type of gear system is reversible (or backdriveable) and therefore is not suitable for this application. Another solution would be to use anti-backlash gears, readily available from many gear manufacturers. However, these anti-backlash gears are spring-loaded systems and still allow relative motion between input and output shafts under certain loading conditions.

The dual-worm mechanism presented here has the ability to minimise backlash using a control strategy that is discussed in later chapters. Control of the gear system must be robust against the non-linear effects of inertia and frictional forces at the worm/wheel interface.

The mechanism has been presented here as a revolute joint for a low power robotic device. However, with suitably large actuators driving the α and β worms, the gear

system could be used for almost any application which would require non-backdriveability

4.3. Robotic Joint Design

The development of the design for the robotic joint has been a three-stage process. Firstly, a 1-DOF prototype system was built with only a single worm in order to test the principle of operation. Good results were achieved that helped gain confidence in the mathematical model. The attachment of the worm wheel to the main shaft proved problematic. Loosening of the wheel from the main shaft was unavoidable using grub screws and a taper pin in the original design as shown in Figure 18a. A better design was implemented as shown in Figure 18b. A redesign of the main shaft made it possible to machine a hub with a flange at one end. The worm wheel could then be rigidly fixed to the hub using three axial cap screws. A locating hub is used to ensure concentricity of the worm wheel. The component and assembly diagrams are included in Appendix D. Figure 19 shows the dual worm-driven joint.

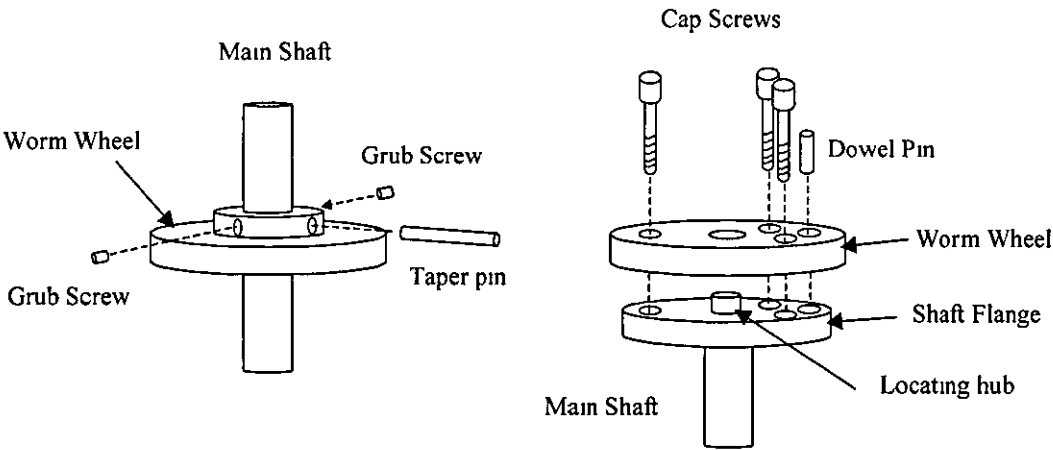


Figure 18 - Methods of fixing the worm wheel to the main shaft

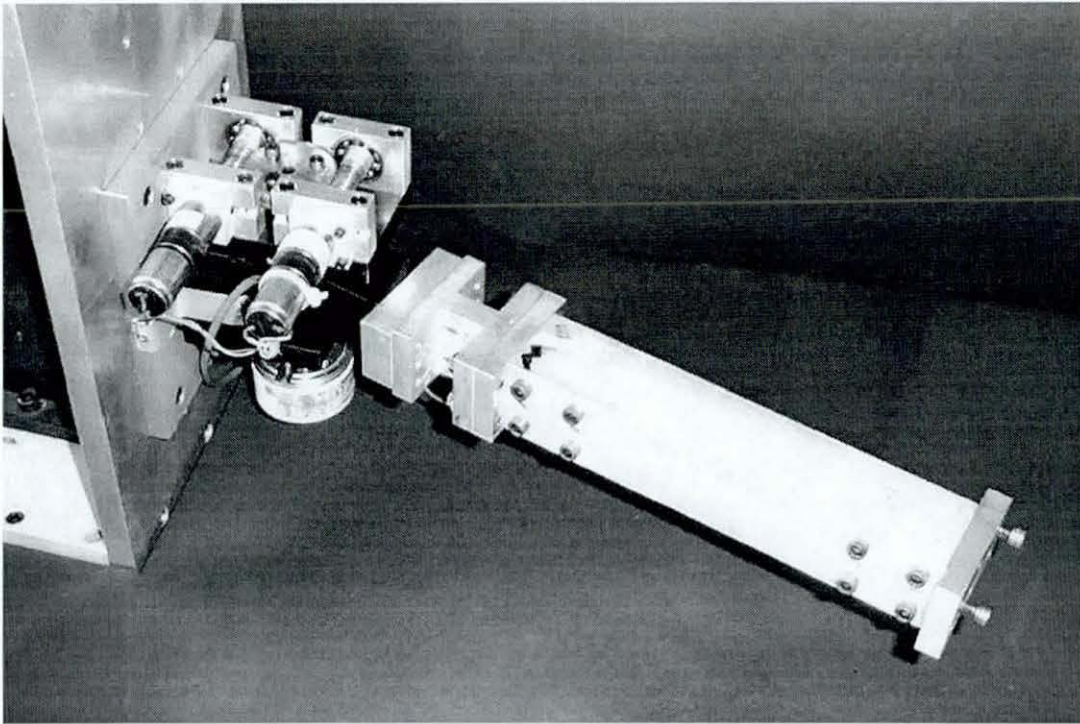


Figure 19 - Worm driven joint

4.4. Two degree-of-freedom manipulator

The 2-DOF manipulator simply consists of two 1-DOF mechanisms connected in series. The 2-DOF manipulator is shown, with a control handle, in Figure 20. The sensors and actuators used in the design of the manipulator are discussed in Section 4.5 and the design of the control handle is discussed in Section 4.6 below.

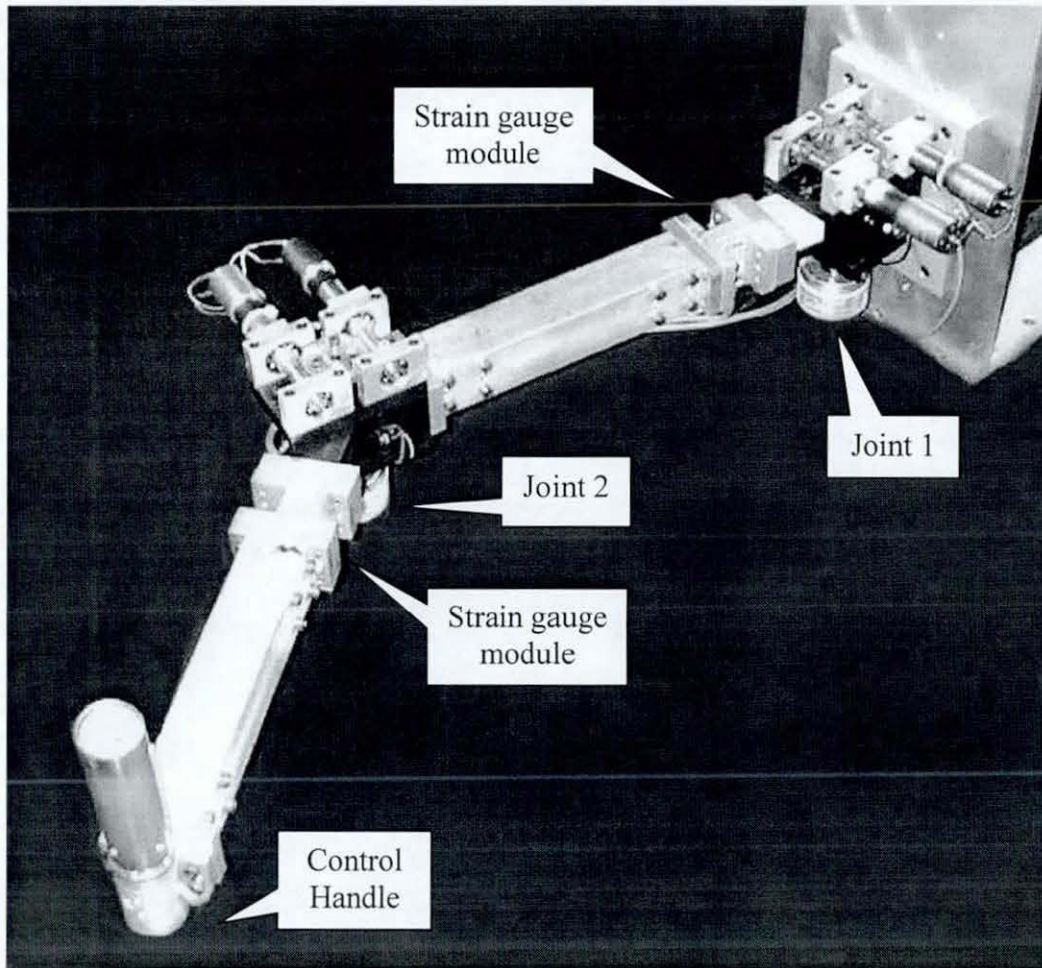


Figure 20 – 2-DOF manipulator with control handle

4.5. Sensors and Actuators

4.5.1. Worm Drive Motors

One of the original considerations was to use a stepper motor to drive each of the worms. One particular advantage of using stepper motors is safety, because the motor cannot 'runaway' if an electronic component in the amplifier fails. However, initial trials with such motors revealed that a discontinuous motion of the joint was apparent to the user. It is also difficult to control the torque applied by a stepper motor. In order to keep the mechanism as safe and as small as possible low power DC servo-motor/gearbox combination from Faulhaber DC Motors were chosen [93]. The original choice of motor/gearbox, a 12 volt, 3.27W motor (Part #: 2230T012S) with a

43.1 planetary gear-head (Part #.23/1), was found to be suitable for applications where the user-input torque did not exceed 10Nm. The unit was capable of providing a torque of 0.8Nm at the output of the gear-head taking into account the gearbox efficiency. However, although control of the robotic joint was successful with this unit, as shown by the simulation and experimental results in Chapter 7, in order to successfully implement a 2-DOF manipulator more powerful, 12 volt, 6.5W motors (Part #. 2842S012C) were used with the same 43.1 planetary gear-head. The selection of these motors was based on the need to be able to overcome friction in the worms under a 50Nm input-torque from the user.

4.5.2 Position Measurement

The rotation of each joint was measured using 5,000 pulse encoders from Hengstler. Using the HCTL-1100 Motion Control Interface of the Single Axis Motion Control Interface Card, see Chapter 5, the quadrature encoder signal can be decoded to give 20,000 counts per revolution.

4.5.3 Dynamic Torque Measurement

In order to control the manipulator measurement of user-input force it is required both to generate a velocity command for the manipulator and to compensate for the friction forces in the mechanism. The ACROBOT system uses a control handle mounted with strain gauges designed to measure the user-input force at the end-effector as shown in Figure 21. Getting an accurate measurement of the force acting on the manipulator could be difficult using this control handle, i.e. it is possible to get erroneous measurements. If the user were to apply the same force, F_x , to the control handle in two different places as shown in Figure 21a, the strain gauges would convey two substantially dissimilar readings. Therefore, as l is variable the user force obtained by the strain gauges is ambiguous. Similarly, it would also be possible for the user to apply a pure torque to the manipulator control handle as shown in Figure 21b. Although it is possible to obtain an accurate reading of user-input force direction the measured magnitude is dependent upon where the user grasps the handle. This would generate an incorrect force reading within the calibrated system. In terms of

the force-controlled ACROBOT system this could be a significant problem. For the worm-driven joint mechanism it will be necessary to accurately measure the torque applied to each joint in order to compensate for the effects of the friction forces acting at the gear interface

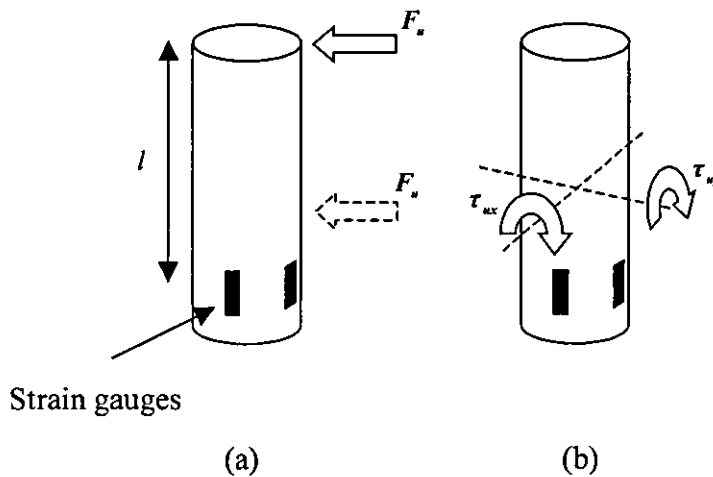


Figure 21 - ACROBOT Control Handle

For this reason the preferred method is to measure the strain in each link as shown in Figure 22. The placement of the strain gauges in this way has the distinct advantage that a more accurate bending strain of the link can be obtained. Assuming that the user is grasping the control handle, the distance to the point of application of the user force is always constant ensuring that an accurate calibration can be performed. It will be shown in Chapter 10 that another advantage of measuring the bending strain of each link is that the non-linear dynamics of the system (Coriolis and centrifugal coupling force) are measured by the strain gauge modules and do not need to be explicitly calculated. However, it should be noted that by measuring the joint torque in this way it is assumed that the link must bend in order to be able to measure the strain, i.e. adequate compliance is necessary. Similarly, if the link is not rigid enough this could lead to (un-measurable) errors in the end-effector position. The photograph of Figure 23 shows the strain gauge module in more detail.

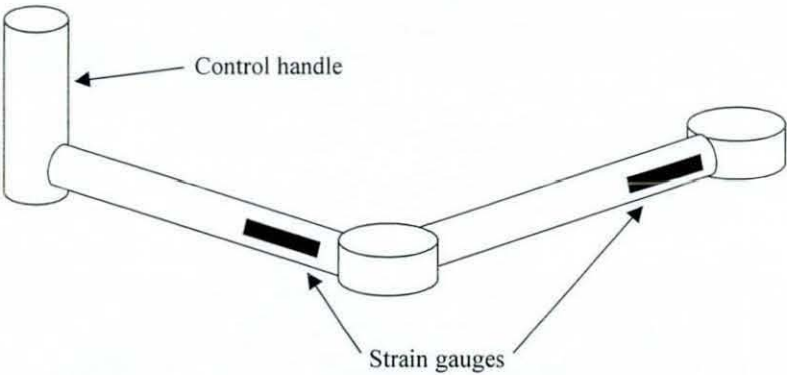


Figure 22 - Force measurement for the worm-driven manipulator



Figure 23 - Strain gauge module

4.6. Control Handle

It was mentioned in Chapter 3 that it would be necessary to provide the user with information on how to apply a force to the end-effector. Many surgical assistant systems use computer graphics to relate information to the surgeon. However, this can be distracting for the surgeon because of the need to be able to focus not only on the task in hand but also on the monitor that is giving directions. The use of a graphical interface in this manner requires the surgeon to transform mentally the information provided on the screen to the real-world system. This may not always be innate and would increase the time required to complete a task, contradicting the purpose of developing the manipulator in the first instance.

To overcome these problems, a control handle is presented here that incorporates a mechanism for informing the surgeon about the direction that the user-input force needs to be applied in order to allow motion. The direction indicator mechanism is mounted inside the control handle. Figure 24 shows a schematic diagram of the control handle. The handle consists of four main components: base unit, motor with integral encoder, direction indicator, and rotating grip. The direction of the indicator is calculated based on the user-input force and path direction.

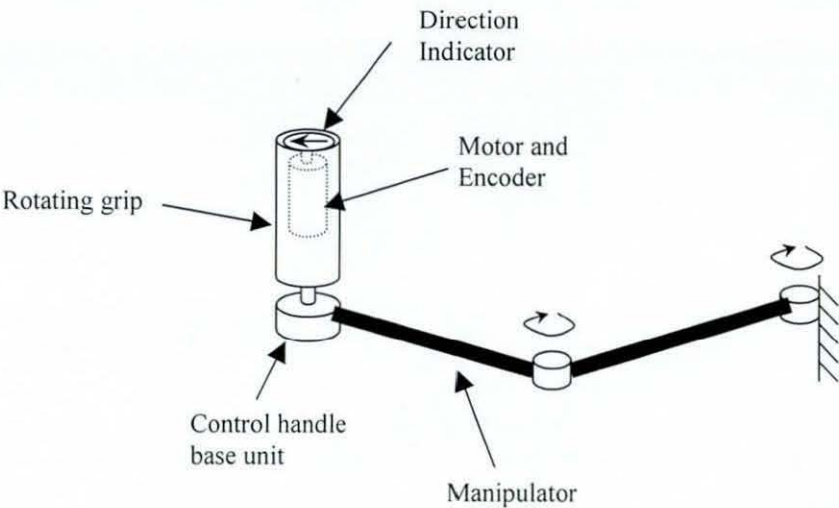


Figure 24 – Manipulator Control Handle Schematic

Mounting the indicator at a similar position to the robot end-effector has two distinct advantages. Firstly, the surgeon does not have to perform a co-ordinate transformation from the direction indicator to the real-world system. Secondly, the indicator is mounted in such a position that it can always be seen without large head movements and change of visual focus. This should help to reduce the need for training in the use of the manipulator and the speed at which operations can be performed. Figure 25 shows a close-up view of the control handle mounted on the robot. The motor, not shown in Figure 25, is a Faulhaber Minimotor (Part #: 1524T012SR) mounted with a 512 pulse magnetic encoder (Part #: IE2-512). Using a HCTL-1100 Motion Control Interface it is possible to decode the two channels to give 2048 counts per revolution. The control handle is shown in more detail in the assembly drawing of Figure 26.

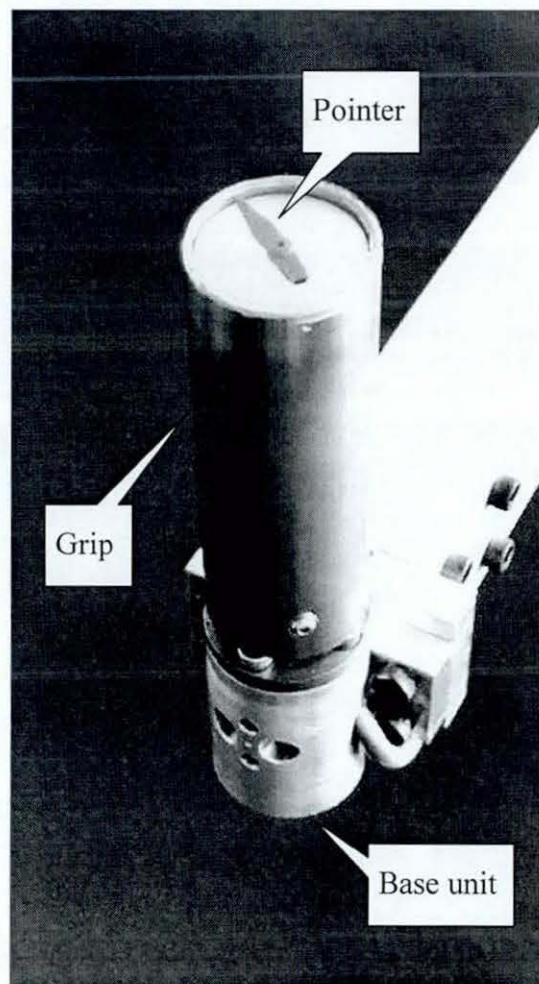
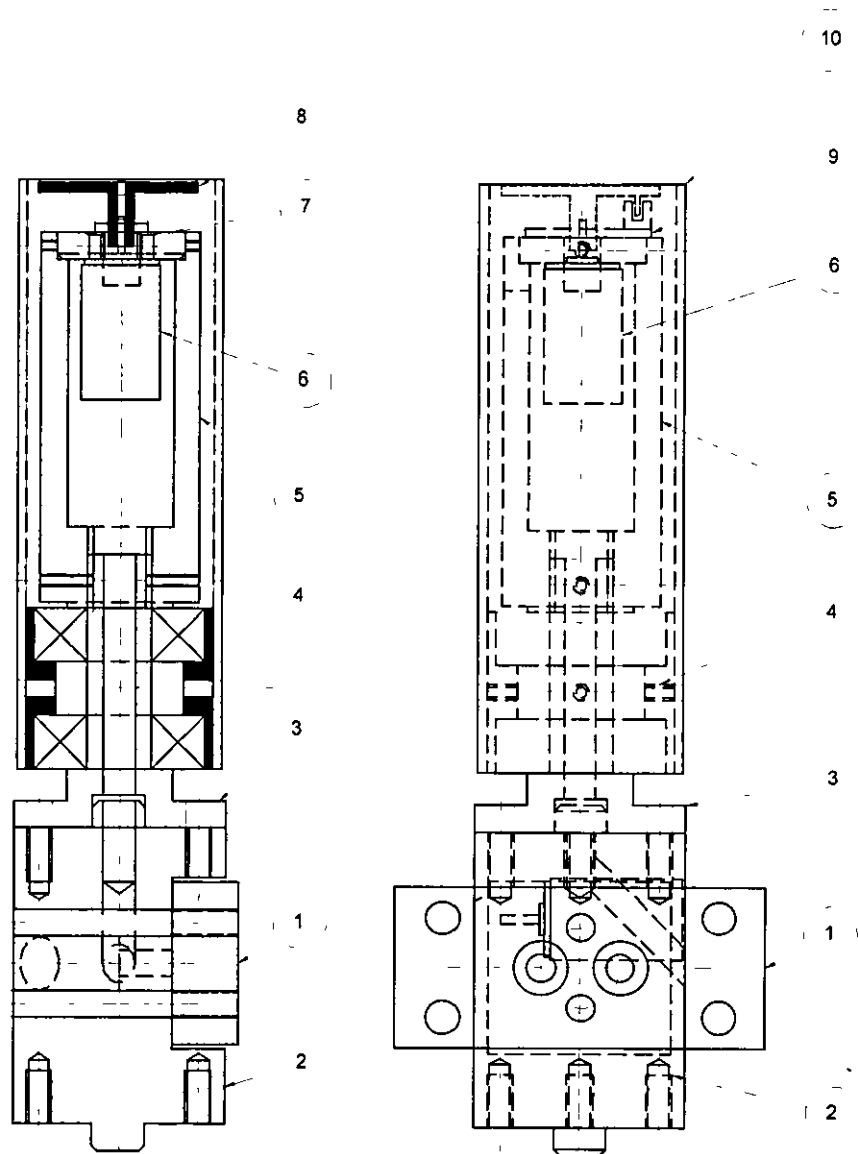


Figure 25 - Control handle and pointer mechanism



- 1 - End-effector mounting plate
- 2 - End-effector base
- 3 - Control handle shaft
- 4 - Control handle bearing housing
- 5 - Control handle motor housing
- 6 - Motor/Encoder
- 7 - Control handle motor mount
- 8 - Pointer rotor
- 9 - Opto-switch assembly
- 10 - Control handle sleeve

Figure 26 - Control handle assembly drawing

Chapter 5 A Mechatronic Workstation for Real-Time Control Development

5.1. Introduction

This chapter details the experimental hardware and software that has been used in order to control the deployment of the manipulator end-effector. The system is shown to be controlling the dual-worm joints of the manipulator, but can be used to control most electro-mechanical system given the correct power electronics. The computational workstation comprises an industrial single-board PC running the QNX 4.25 real-time operating system (RTOS). Potentially, the computer can be fitted with any commercial data acquisition hardware. However, as part of this work a Single Axis Motion Control Interface has been developed using HCTL-1100 Motion Controller devices from Agilent Technologies. All of the control algorithms described in this work have been developed and implemented using this workstation.

5.2. System Overview

One of the major problems in designing control systems is implementing real-time data capture and control. Many PC based control solutions involve the use of expensive digital signal processing (DSP) interface cards. This problem has been overcome in this instance by using the QNX4.25 RTOS from QNX Software Systems Ltd [94]. Unlike the QNX operating system, the Microsoft Windows operating system (OS) is designed to control and give priority to system hardware. As a result the interrupt latency, the time taken from the interrupt being triggered to the time when interrupt routine execution begins, is undefined. Another advantage of using this particular operating system is that QNX is packaged with a windows-based environment, which allows for the design of an attractive graphical user interface. However, unlike MS Windows 98/NT where interrupts (and in the case of NT low-level input/output (I/O)) are hidden from most users, this RTOS enables the user to program using interrupts and perform I/O without writing complicated device drivers.

The RTOS runs the control development software on a Pentium 233MHz single-board computer (SBC) from Professional Industrial Solutions with 64Mb of RAM that guarantees a response to an externally generated interrupt in under 4 μ S. The SBC is housed in an industrial rack-mounted case. All of the necessary low power electronics to read encoder positions, capture analogue/digital data and generate motor command signals have been designed and housed in this unit. The electronic hardware consists of a number of Single Axis Motion Control (SAMC) interface cards (described below). A PC14AT interface card from Amplicon is also used to generate the interrupt signal required for the control software. Figure 27 shows the system being used to control the 2-DOF robotic system with five SAMC interface cards. However, it must be noted that the same system could be used to control a 6-DOF robotic manipulator. Each Axis Control Interface controls a joint actuator and obtains sensor information for that joint.

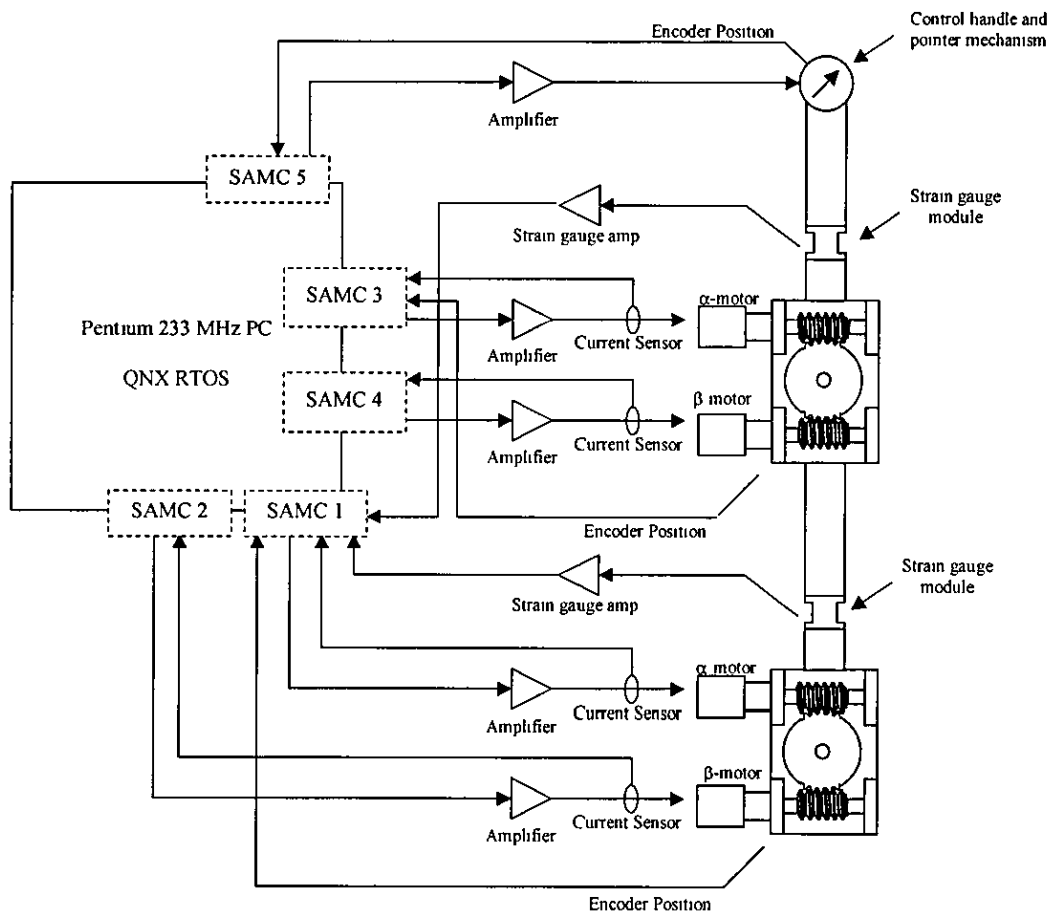


Figure 27 - Manipulator control system overview

The rack-mounting case is shown in Figure 28. The Single Axis Motion Control Interface Cards are shown in situ along with the Amplicon PC14AT and Pentium 233MHz SBC.

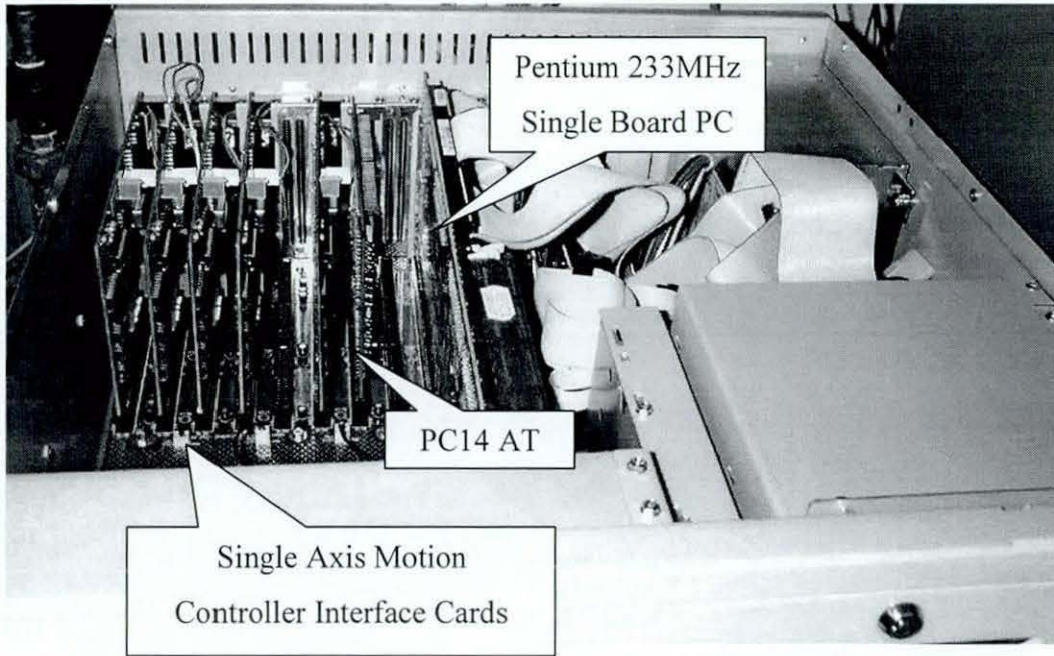


Figure 28 - Motion Control Computer

5.3. Single-axis motion control interface

As part of this work it has been necessary to develop a motion control interface. An EISA interface card has been designed to be versatile, i.e. drive and control not only the DC servo-motors of the dual-worm joints but also stepper and brushless DC motors. The interface card, produced on a double-sided printed-circuit board using the plated-through-hole technique, has been manufactured in this department. The interface card, incorporating the HCTL-1100 motion control interface (Agilent Technologies) [95], has been designed to generate the low power digital and analogue signals required to drive the three types of motor described above. The following list describes the attributes of each single axis controller:

- HCTL-1100 for controlling motors in one of four modes:
 1. Position Control,
 2. Proportional Velocity Control,
 3. Proportional + Integral Velocity Control, and
 4. Trapezoidal Profile Control.
- Incremental Encoder Decoding Hardware (HCTL-1100)
- 8 channels of 12 bit analogue input.
- 8 digital inputs.
- 8 digital outputs.
- 1 channel of 12 bit DAC output.
- 1 channel of 8 bit DAC output, HCTL-1100 motor command port.
- PWM motor command port.
- Stepper/Brushless DC motor command port

Figure 29 shows a block diagram of the interface. The interface card can be installed on any IBM compatible PC with an EISA bus. Figure 30 shows the completed interface card. A user manual, containing information for building and using the single-axis motion control interface is included in Appendix A. The user manual also contains information on the software libraries that have been provided for C and C++ programming environments.

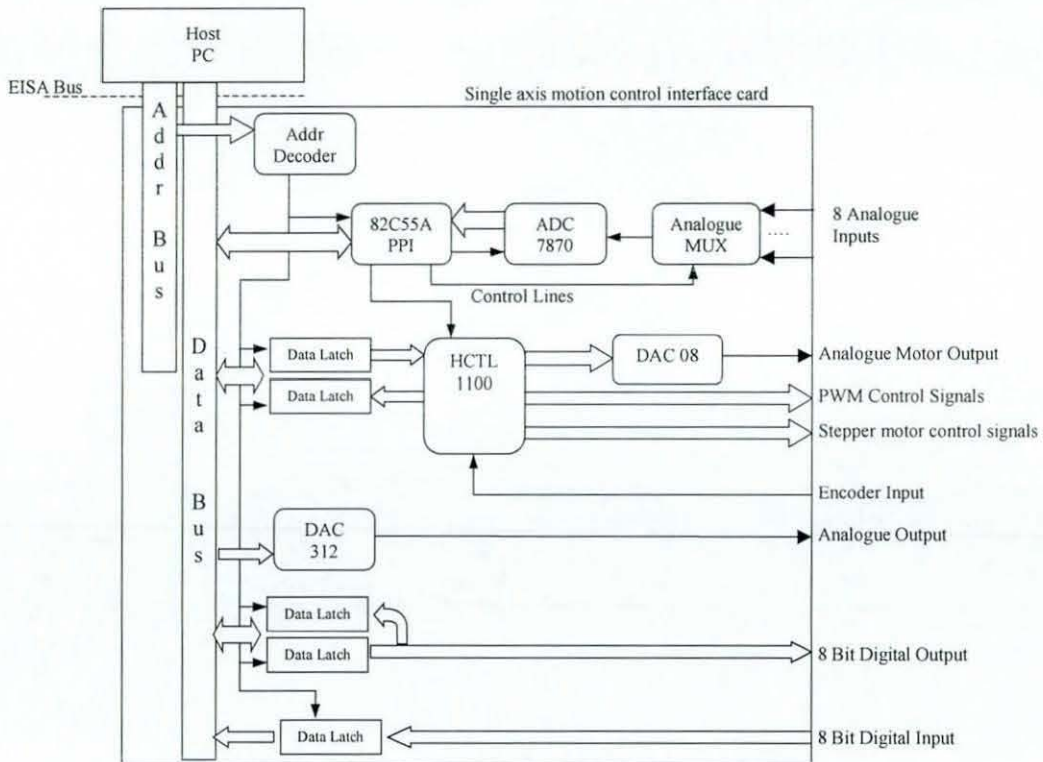


Figure 29 - Single Axis Motion Control Interface Card Block Diagram

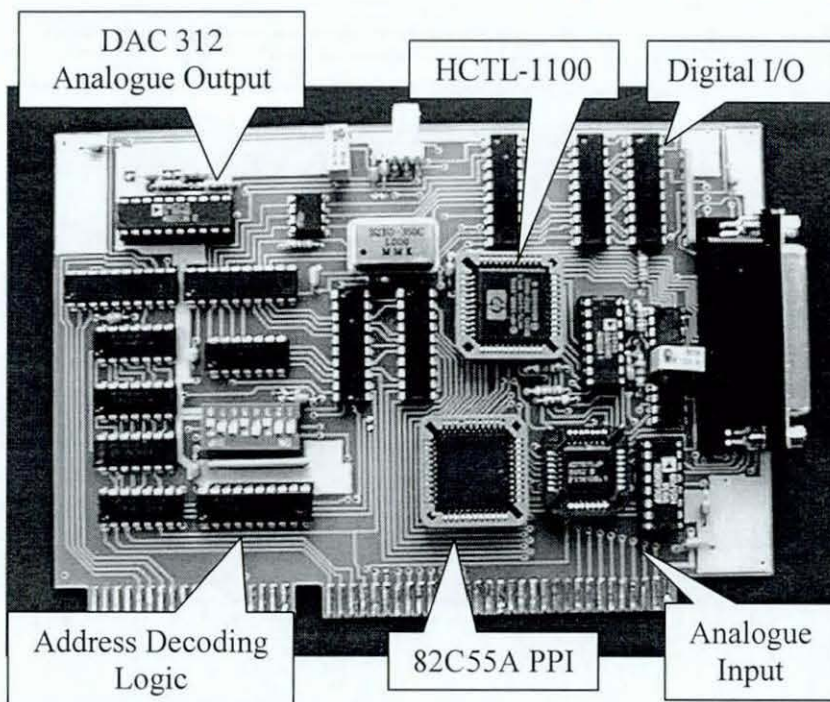


Figure 30 – Single Axis Motion Control Interface Card

5.4. Power Electronic Interface

In between each motion control interface and the actuators of the mechanism that is to be controlled, there exist some electronic hardware for amplification of analogue/digital signals and isolation of the low-power digital computer signals. As part of this research it was also necessary to design a DC servo amplifier card that included a current sensor. The circuit diagram for the DC servo motor amplifier is shown in Figure 31. The circuit includes a power supply filter and a Honeywell CSNE151-005 current sensor for monitoring armature current. The isolated signal from the current sensor is fed back to the SAMC interface card via channel 0 of the analogue input interface. To allow for further flexibility and the implementation of more complex control strategies than the HCTL-1100 can provide, a jumper is provided that has the ability to switch motor control between the analogue output of either the DAC-312 or the HCTL-1100 motor command port. The DC servo motor amplifier card is shown in Figure 32.

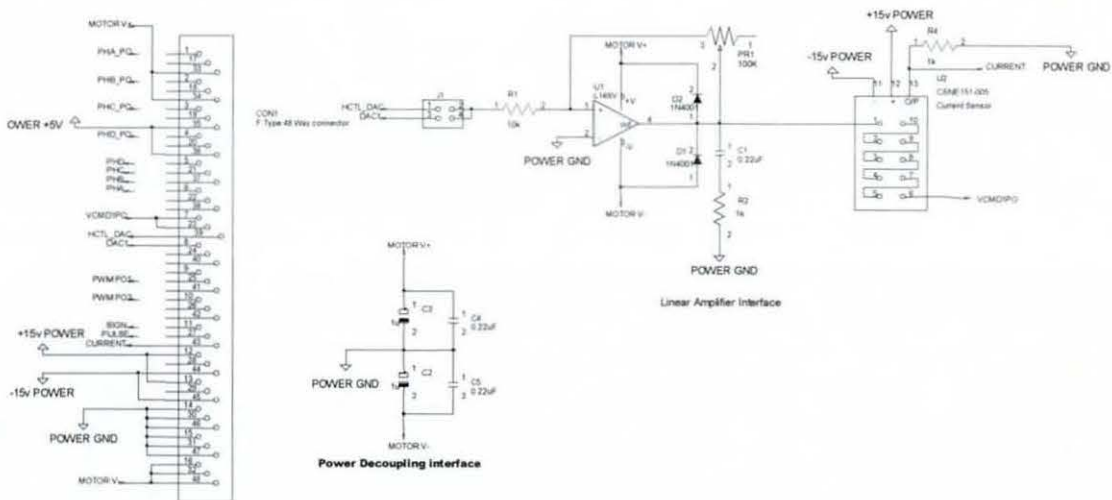


Figure 31 - DC Servo Motor Amplifier Circuit Diagram

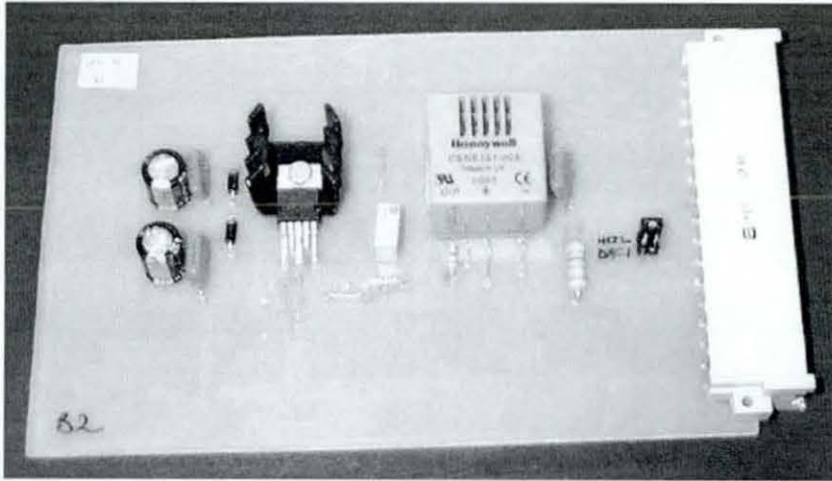


Figure 32 - DC Servo Motor Amplifier Card

To minimise the effect of noise the power electronic interfaces and isolation boards are housed separately from the PC as shown in Figure 33. Detailed diagrams of the internal and external connectors associated with the power interface are included in Appendix B.

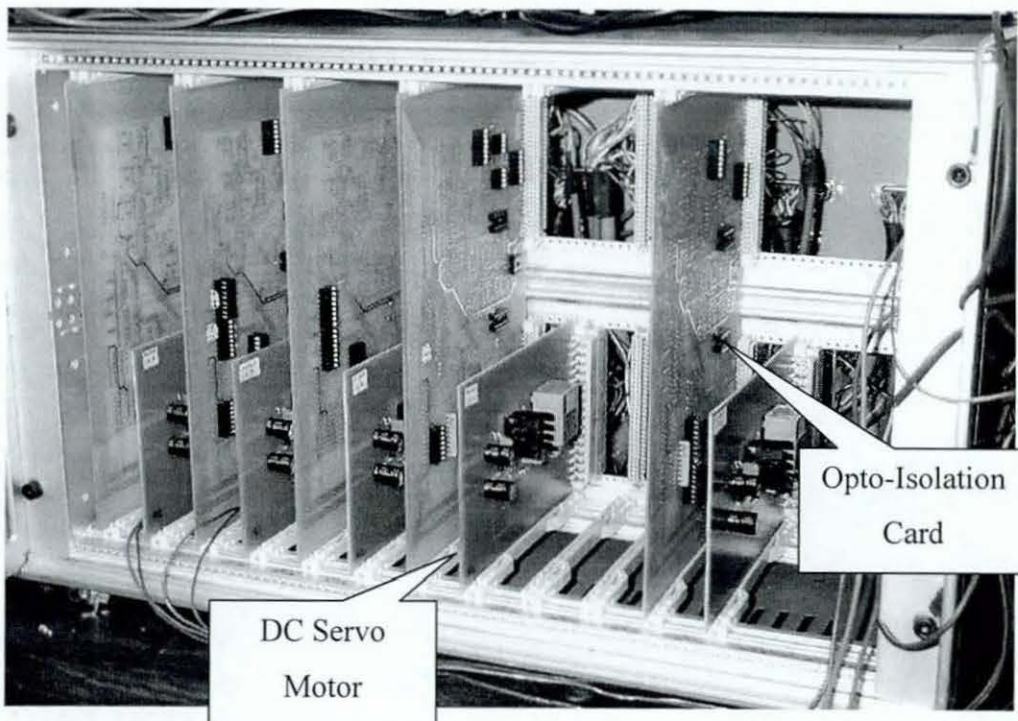


Figure 33 - Power electronics and opto-isolator cards

5.5. Control Development Studio

The QNX RTOS has a window-based environment, known as Photon, which can be used to generate attractive graphical user interfaces. The Photon application builder was used to create a generic control program that is used to automatically generate sub-programs that run the real-time control loops. This section is dedicated to describing the operation of this software known as the Control Development Studio (CDS)

5.5.1. Description

The CDS software is an environment in which the user can develop, debug and run control algorithms in the C++ programming language. CDS loads and runs a selected control program, communicating between the user and the control program. The developer, by implementing some of the functions provided by the CDS, can allow the user to change control variables, log control data such as position and error, and inform the user about the state of the control program for debugging purposes

A screen shot of the CDS User Interface is shown in Figure 34 below. The window displays four main features:

1. *Log message window* - used mainly for displaying debugging information.
2. *Log Variables window* - used to display buttons that control the display of data in the graph window.
3. *Control Variables window* - used to display the status of all adjustable parameters.
4. *Graph window* - displays the selected logged variables on a time base

The programmer of the control software regulates the variables that are to be logged and those that are to be adjusted by the user with the C++ functions described below

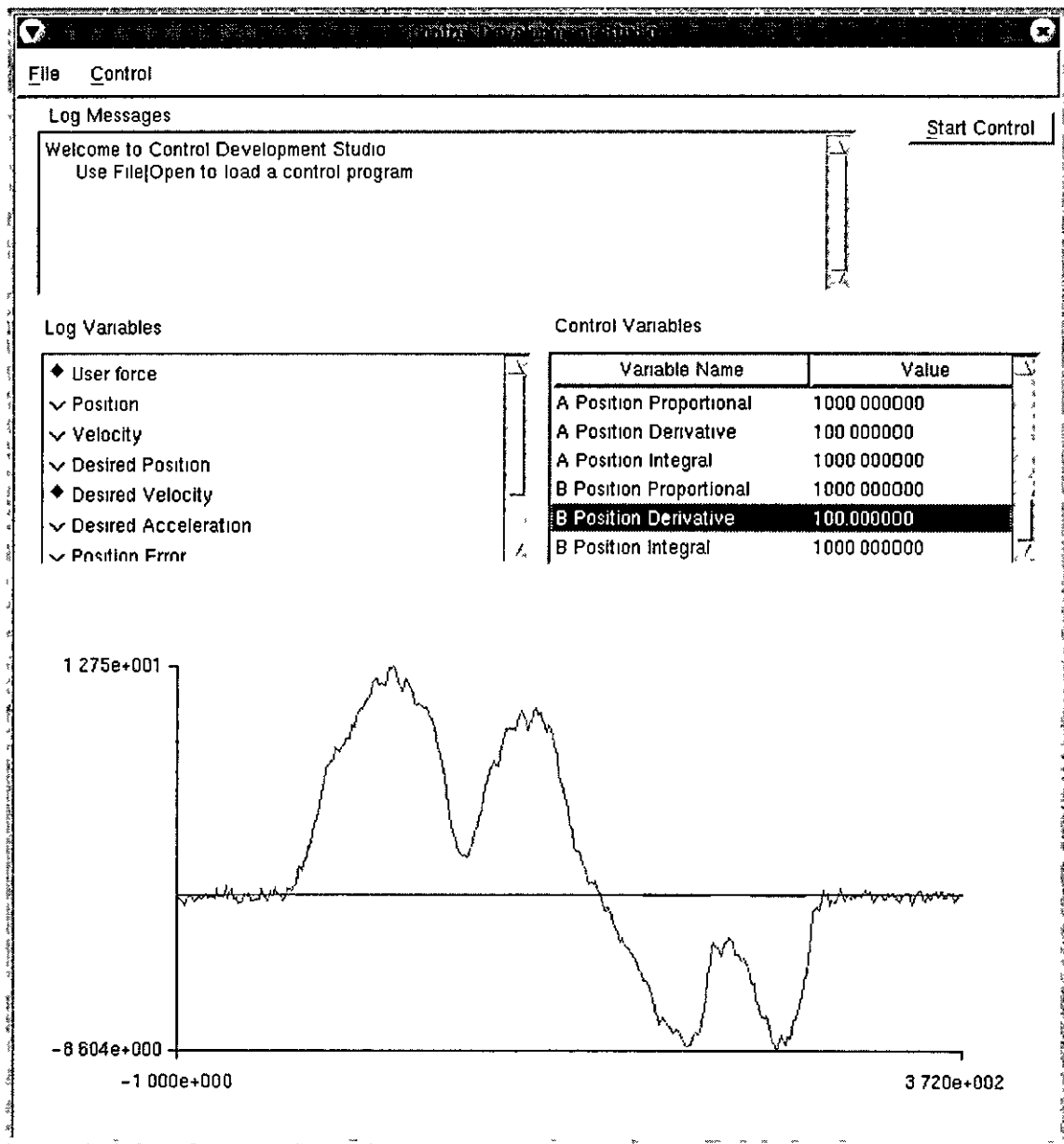


Figure 34 - Control Development Studio screen shot

5.5 2. Principle of operation

The CDS program allows control algorithms to be developed as plug-in utilities. The main CDS application is used to log data about the control program. A block diagram of the communication between the CDS and the control process is shown in Figure 35. The processes exchange data using two forms of inter-process communication (IPC). Messages are used to pass initialisation data and switch the control mode

between running and idle states. The QNX messaging system means that the process sending the message waits for the receiving process to acknowledge receipt. This is not an effective method of implementing data logging functions such as the one required here. Hence a form of asynchronous IPC is used known as a "photon pulses". A pulse is similar in effect to a software interrupt, the process receiving the pulse is informed that an event has occurred, in this case that some data is ready for processing. Using a data stream, known as a pipe, information such as logged data or messages can be sequentially fed down the stream by the control process and read at the other end by the CDS. The CDS only processes the data as and when it has the processor time to do so. No data is lost, even if the CDS cannot keep up with the control program. Care should be taken when designing a control to minimise the sampling frequency for data logging so that memory for system processes is preserved.

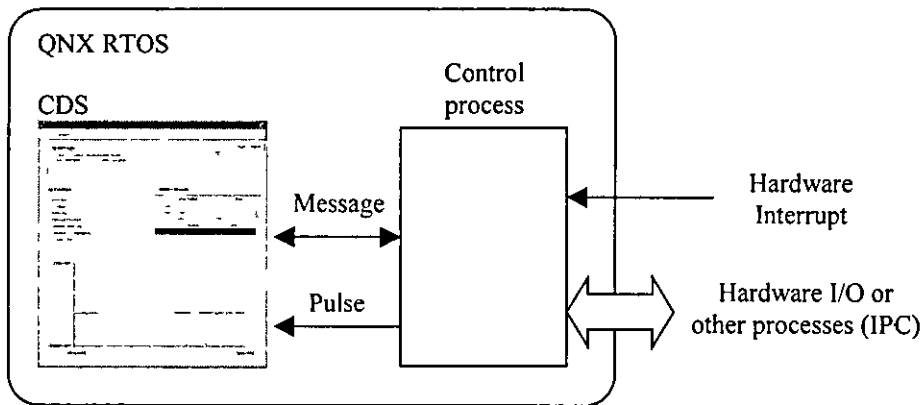


Figure 35 - Control Development Studio mode of operation

Chapter 6 Mathematical Model of the Dual-Worm Driven Joint

6.1. Introduction

In Chapter 4, a robot joint architecture designed to cancel backlash was presented. In this chapter a mathematical model of the mechanism is presented that will be used to develop a computed-torque and composite adaptive control schemes for path tracking applications.

6.2. System Analysis

Analysis of the dual-worm mechanism identifies four separate cases of contact between the two worms and the worm wheel. The first of these four cases (case I), shown in Figure 36a, represents the worm wheel in contact with the leading edges of the thread of both the α and β worms. In this condition the two worms cancel backlash. Figure 36b shows the second case (case II) where the worm wheel is in contact with the trailing edge of the thread of the α -worm and the leading edge of the thread of the β -worm. This situation is highly undesirable in terms of positional accuracy, as backlash is not cancelled. The third case (case III), illustrated Figure 36c, shows the trailing edges of the thread of the two worms in contact with the worm wheel. Backlash is cancelled in this condition. Finally, Figure 36d (case IV), the leading edge of the thread of the α -worm and the trailing edge of the thread of the β -worm are shown in contact with the wheel. As in case II, backlash is not cancelled in this condition. Therefore, backlash is minimised if either case I or case III is continuously maintained. In order to simplify the model, a control strategy is chosen that will maintain the conditions required for only case I. The mathematical model for case I is presented here, the derivation for all cases is included in Appendix E.

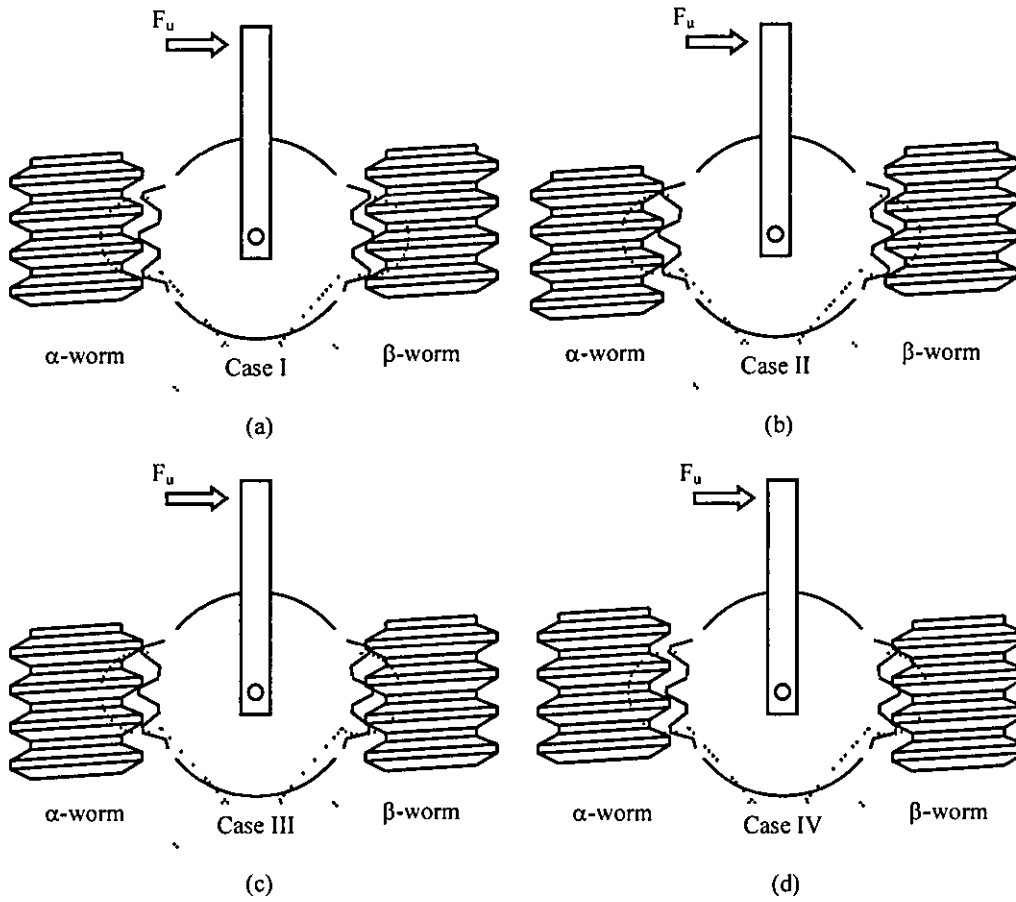


Figure 36 - Dual-worm joint modes of contact

6.3. Mathematical Model

Each of the three gears can be modelled as a plane inclined at an angle equal to the lead of the worms. A free-body diagram for case I is shown in Figure 37, where the upper wedge represents the α -worm, the middle wedge represents the worm wheel, and the bottom wedge represents the β -worm. In Figure 37, the terms M_θ , M_α , and M_β represent the equivalent mass of the worm-wheel inertia, the α -worm and β -worm respectively. Similarly the terms C_θ , C_α , and C_β represent the equivalent viscous friction present in the worm-wheel, α -worm and β -worm systems. The forces F_u , $F_{\alpha\alpha}$, $F_{\alpha\beta}$, $F_{\beta\alpha}$, and $F_{\beta\beta}$ refer to the equivalent user-input force applied to the end-effector, the contact force between the α -worm and worm-wheel, the contact force between the β -worm and worm-wheel, the friction force at the α -worm and worm-

wheel interface, and the friction force at the β -worm and worm-wheel interface respectively.

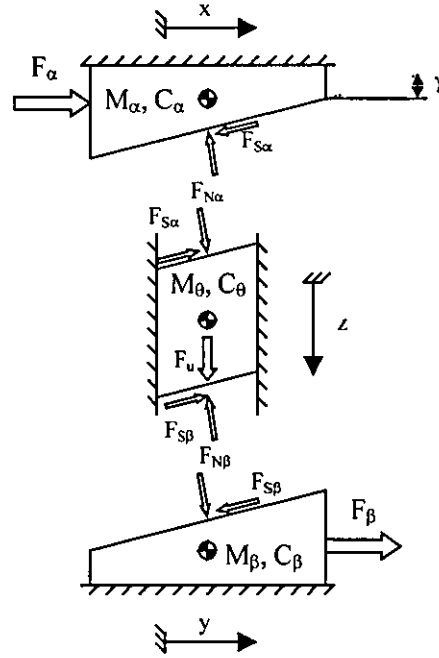


Figure 37 – Free-body diagram of worm drive joint: Case I

Using Figure 37, it is possible to write the dynamic equations of each wedge as equations (6-1), (6-2), and (6-3), respectively. The friction forces, $F_{s\alpha}$ and $F_{s\beta}$, between the two worms and the wheel are modelled using an exponential stick-slip friction model [96, 97] given in equations (6-4) and (6-5), respectively. In order to simplify the model it is assumed that neither the α -worm nor β -worm loses contact with the worm wheel, i.e. $F_{N\alpha} \neq 0$ and $F_{N\beta} \neq 0$.

$$M_{\alpha} \ddot{x} + C_{\alpha} \dot{x} = F_{\alpha} - F_{N\alpha} \sin \gamma - F_{s\alpha} \cos \gamma \quad (6-1)$$

$$M_{\theta} \ddot{z} + C_{\theta} \dot{z} = F_u + F_{s\alpha} \cos \gamma - F_{N\alpha} \sin \gamma - F_{s\beta} \cos \gamma + F_{N\beta} \sin \gamma \quad (6-2)$$

$$M_{\beta} \ddot{y} + C_{\beta} \dot{y} = F_{\beta} + F_{N\beta} \sin \gamma - F_{s\beta} \cos \gamma \quad (6-3)$$

$$F_{s\alpha} = \left(\mu_{da} + (\mu_{sa} - \mu_{da}) e^{\frac{-|x|}{\epsilon \cos \gamma}} \right) \text{sgn}(\dot{x}) F_{N\alpha} \quad (6-4)$$

$$F_{s\beta} = \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{\frac{-|y|}{\epsilon \cos \gamma}} \right) \text{sgn}(y) F_{v\beta} \quad (6-5)$$

Substitution of , (6-3), (6-4), and (6-5) into (6-2) yields

$$M_\theta z + C_\theta z = F_v + \frac{F_\alpha - M_\alpha x - C_\alpha x}{\Delta_\alpha} + \frac{F_\beta - M_\beta y - C_\beta y}{\Delta_\beta} \quad (6-6)$$

where

$$\Delta_\alpha = \frac{\sin \gamma + \left(\mu_{da} + (\mu_{sa} - \mu_{da}) e^{\frac{-|x|}{\epsilon \cos \gamma}} \right) \text{sgn}(x) \cos \gamma}{\cos \gamma - \left(\mu_{da} + (\mu_{sa} - \mu_{da}) e^{\frac{-|x|}{\epsilon \cos \gamma}} \right) \text{sgn}(x) \sin \gamma} \quad \Delta_\beta = \frac{\sin \gamma - \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{\frac{-|y|}{\epsilon \cos \gamma}} \right) \text{sgn}(y) \cos \gamma}{\cos \gamma + \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{\frac{-|y|}{\epsilon \cos \gamma}} \right) \text{sgn}(y) \sin \gamma}$$

Converting from the unwound threads to the rotational system by replacing F_α , F_β , F_v , M_α , M_β , M_θ , C_α , C_β , C_θ , z , x , and y in equation (6-6) with $\tau_{m\alpha}/r_\alpha$, $\tau_{m\beta}/r_\beta$, τ_v/r_θ , J_α/r_α^2 , J_β/r_β^2 , J_θ/r_θ^2 , $C_{f\alpha}/r_\alpha^2$, $C_{f\beta}/r_\beta^2$, $C_{f\theta}/r_\theta^2$, θ_θ , $\theta_\theta/\tan \gamma$, and $\theta_\theta/\tan \gamma$ respectively yields

$$J_\theta \theta + C_\theta \theta = \tau_v + \frac{\tau_{m\alpha} r_\theta}{\Delta_\alpha r_\alpha} + \frac{\tau_{m\beta} r_\theta}{\Delta_\beta r_\beta} \quad (6-7)$$

where $J_\tau = \left(J_\theta + \frac{J_\alpha r_\theta^2}{\Delta_\alpha r_\alpha^2 \tan \gamma} + \frac{J_\beta r_\theta^2}{\Delta_\beta r_\beta^2 \tan \gamma} \right)$ $C_\tau = \left(C_{f\theta} + \frac{C_{f\alpha} r_\theta^2}{\Delta_\alpha r_\alpha^2 \tan \gamma} + \frac{C_{f\beta} r_\theta^2}{\Delta_\beta r_\beta^2 \tan \gamma} \right)$

and

$$\Delta_\alpha = \frac{\sin \gamma + \left(\mu_{da} + (\mu_{sa} - \mu_{da}) e^{\frac{-|\theta|}{\epsilon \sin \gamma}} \right) \text{sgn}(\theta) \cos \gamma}{\cos \gamma - \left(\mu_{da} + (\mu_{sa} - \mu_{da}) e^{\frac{-|\theta|}{\epsilon \sin \gamma}} \right) \text{sgn}(\theta) \sin \gamma} \quad \Delta_\beta = \frac{\sin \gamma - \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{\frac{-|\theta|}{\epsilon \sin \gamma}} \right) \text{sgn}(\theta) \cos \gamma}{\cos \gamma + \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{\frac{-|\theta|}{\epsilon \sin \gamma}} \right) \text{sgn}(\theta) \sin \gamma}$$

For simulation purposes, it is necessary to ensure that there is no motion until the motor torque of either the α -worm or the β -worm exceeds the static friction forces. Figure 38a shows how the friction forces act on the three wedges during motion of the wheel in the anti-clockwise direction (indicated as α -unwinding) and Figure 38b shows the orientation of the forces during motion of the wheel in the clockwise direction (indicated as β -unwinding).

From Figure 38 it is possible to express the equilibrium conditions in the anti-clockwise direction, equation (6-8), and clockwise direction, equation (6-9). In Figure 38, $\mu_{s\alpha}F_{N\alpha}$ and $\mu_{s\beta}F_{N\beta}$ are substituted for $F_{s\alpha}$ and $F_{s\beta}$ respectively. In Figure 38 the magnitude of the static friction forces $F_{s\alpha}$ and $F_{s\beta}$ are equal to $\mu_{s\alpha}F_{N\alpha}$ and $\mu_{s\beta}F_{N\beta}$ respectively. The direction of $F_{s\alpha}$ and $F_{s\beta}$ are defined by whether α -unwinding or β -unwinding motion is about to occur.

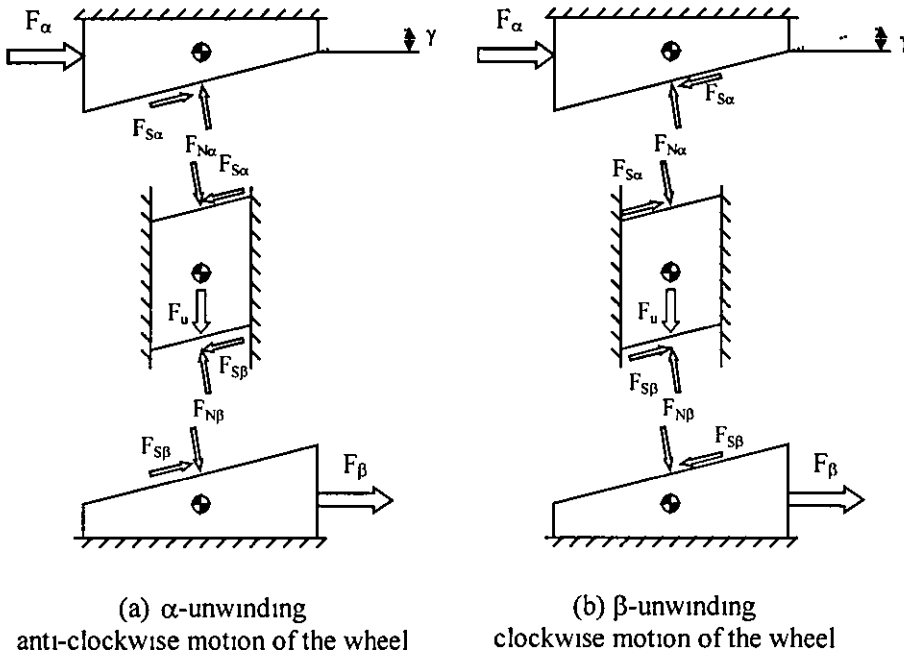


Figure 38 - Freebody diagram of static conditions

$$F_u + F_\alpha \frac{(\cos \gamma + \mu_{s\alpha} \sin \gamma)}{(\sin \gamma - \mu_{s\alpha} \cos \gamma)} + F_\beta \frac{(\cos \gamma - \mu_{s\beta} \sin \gamma)}{(\sin \gamma + \mu_{s\beta} \cos \gamma)} = 0 \quad (6-8)$$

$$F_u + F_\alpha \frac{(\cos \gamma - \mu_{s\alpha} \sin \gamma)}{(\sin \gamma + \mu_{s\alpha} \cos \gamma)} + F_\beta \frac{(\cos \gamma + \mu_{s\beta} \sin \gamma)}{(\sin \gamma - \mu_{s\beta} \cos \gamma)} = 0 \quad (6-9)$$

Using equation (6-8) for the system of unwound threads, motion of the wheel in the anti-clockwise direction will occur if the condition given by equation (6-10) is satisfied. Equation (6-11) gives this condition for the actual rotational system.

$$F_a < -F_u \frac{(\sin \gamma - \mu_{sa} \cos \gamma)}{(\cos \gamma + \mu_{sa} \sin \gamma)} - F_\beta \frac{(\sin \gamma - \mu_{sa} \cos \gamma)(\cos \gamma - \mu_{sb} \sin \gamma)}{(\cos \gamma + \mu_{sa} \sin \gamma)(\sin \gamma + \mu_{sb} \cos \gamma)} < 0 \quad \text{AND} \quad F_\beta < 0 \quad (6-10)$$

$$\frac{\tau_{ma}}{r_a} < -\frac{\tau_u}{r_\theta} \frac{(\sin \gamma - \mu_{sa} \cos \gamma)}{(\cos \gamma + \mu_{sa} \sin \gamma)} - \frac{\tau_{m\beta}}{r_\beta} \frac{(\sin \gamma - \mu_{sa} \cos \gamma)(\cos \gamma - \mu_{sb} \sin \gamma)}{(\cos \gamma + \mu_{sa} \sin \gamma)(\sin \gamma + \mu_{sb} \cos \gamma)} < 0 \quad \text{AND} \quad \tau_{m\beta} < 0 \quad (6-11)$$

Similarly, using equation (6-9) for the system of unwound threads, clockwise rotation of the wheel will be allowed if the condition given by equation (6-12) is satisfied. Equation (6-13) gives the same condition in terms of the actual rotational system.

$$F_\beta > -F_u \frac{(\sin \gamma - \mu_{sb} \cos \gamma)}{(\cos \gamma + \mu_{sb} \sin \gamma)} - F_a \frac{(\sin \gamma - \mu_{sb} \cos \gamma)(\cos \gamma - \mu_{sa} \sin \gamma)}{(\cos \gamma + \mu_{sb} \sin \gamma)(\sin \gamma + \mu_{sa} \cos \gamma)} > 0 \quad \text{AND} \quad F_a > 0 \quad (6-12)$$

$$\frac{\tau_{m\beta}}{r_\beta} > -\frac{\tau_u}{r_\theta} \frac{(\sin \gamma - \mu_{sb} \cos \gamma)}{(\cos \gamma + \mu_{sb} \sin \gamma)} - \frac{\tau_{ma}}{r_a} \frac{(\sin \gamma - \mu_{sb} \cos \gamma)(\cos \gamma - \mu_{sa} \sin \gamma)}{(\cos \gamma + \mu_{sb} \sin \gamma)(\sin \gamma + \mu_{sa} \cos \gamma)} > 0 \quad \text{AND} \quad \tau_{ma} > 0 \quad (6-13)$$

This model given by equation (6-7) will be used in subsequent chapters to develop suitable control algorithms for the joint mechanism.

Chapter 7 Computed-Torque Control of a One Degree-of-Freedom Worm Driven Manipulator

7.1. Introduction

In this chapter, a strategy for the control of a 1-DOF worm-driven manipulator is described. The computed-torque control strategy ensures safety of the user and cancels backlash. The control algorithm is shown to be effective in tracking a desired joint position generated from a velocity determined by the direction and magnitude of the user-input force. Simulation and experimental results are shown that illustrate the ability of the algorithm to make the end-effector track the trajectory. The results of the dual-worm mechanism are compared with those of a mechanism that includes only a single worm to highlight the success of the backlash cancellation strategy.

7.2. Control Strategy

The aim of the control algorithm is to make the manipulator track a desired position command whilst cancelling backlash at the worm interface. In order to ensure surgeon and patient safety the joint control and deployment strategy **must not** allow:

- (i) any motion against the user,
- (ii) any motion without direct control from the user,
- (iii) any backlash at the worm/wheel interface, or
- (iv) any motion caused by excessive user input-force

To this end, effective control of the dual-worm mechanism requires two algorithms. In the clockwise direction (i.e. the user-input torque $\tau_u > 0$) the motor command voltage for the α -worm is set to a constant value and the β -motor torque, $\tau_{m\beta}$, is controlled to unwind the β -worm to track the trajectory. In this condition the β -worm *leads* and the α -worm is used to *follow* the motion of the worm wheel without

applying unnecessary frictional forces to the system. Control of motion in this manner shall be termed β -unwinding control. However, in the anti-clockwise direction (i.e. $\tau_v < 0$) the β -worm motor command voltage is set to a constant value and τ_{ma} is used to control the unwinding of the α -worm to track the trajectory. This is termed α -unwinding control. In this instance, the α -worm *leads* and the β -worm *follows*. This dual-control strategy is shown to work well.

For the purpose of this preliminary work the desired velocity is proportional to the user-input torque. That is

$$\theta_d = K_v \tau_v \quad (7-1)$$

where K_v could be chosen as a function of position in order to restrict motion. For a manipulator with more than 1-DOF, this velocity-limiting algorithm would be used to constrain the user to remain along/within a pre-defined path/region within the robot's workspace.

7.3. α -unwinding control

The α -unwinding control law regulates the unwinding of the α -worm to follow a time varying trajectory. The error between the desired position and the actual position of the system is defined as

$$e = \theta_d - \theta \quad (7-2)$$

From equations (6-7) and (7-2) it is possible to derive the tracking error dynamics given by equation (7-3)

$$e = \theta_d + \frac{1}{J_r} \left(C_r \theta - \tau_v - \frac{\tau_{ma} r_{\theta}}{\Delta_a r_a} - \frac{\tau_{mb} r_{\theta}}{\Delta_b r_b} \right) \quad (7-3)$$

Using feedback linearisation, τ_{ma} can be chosen to cancel the dynamic effects of friction and the inertial forces τ_{ma} is obtained as given in equation (7-4) where $\hat{\Delta}_\alpha$, $\hat{\Delta}_\beta$, \hat{C}_τ , and \hat{J}_τ are the estimates of Δ_α , Δ_β , C_τ , and J_τ , respectively

$$\tau_{ma} = \frac{r_\alpha \hat{\Delta}_\alpha}{r_\theta} \left(\hat{J}_\tau u_\alpha + \hat{J}_\tau \theta_d + \hat{C}_\tau \theta - \tau_\alpha - \frac{\tau_{m\beta} r_\theta}{r_\beta \hat{\Delta}_\beta} \right) \quad (7-4)$$

In equation (7-4) the control u_α is chosen to drive the error to zero and $\tau_{m\beta}$ is the measured β -motor torque. Convergence of the error to zero is achieved using a Proportional+Integral+Derivative (PID) control law, equation (7-5).

$$u_\alpha = K_{va} e + K_{pa} e + K_{ia} \int e \, dt \quad (7-5)$$

It is important to note that if the components of equation (7-4) are not known exactly the system parameter estimates may be inaccurate and without the integral term in equation (7-5) there may be a non-zero steady-state error

7.4. β -unwinding control

A computed-torque controller for unwinding the β -worm can be found in a similar fashion as in section 7.3. Equation (6-7) can be used to derive the β -motor torque as given in equation (7-6)

$$\tau_{m\beta} = \frac{r_\beta \hat{\Delta}_\beta}{r_\theta} \left(\hat{J}_\tau u_\beta + \hat{J}_\tau \theta_d + \hat{C}_\tau \theta - \tau_\alpha - \frac{\tau_{ma} r_\theta}{r_\alpha \hat{\Delta}_\alpha} \right) \quad (7-6)$$

In this case the control u_β is chosen to drive the error to zero and τ_{ma} is the measured α -motor torque. Convergence of the error is achieved using the PID controller of equation (7-7)

$$u_\beta = K_{v\beta} e + K_{p\beta} e + K_{i\beta} \int e \, dt \quad (7-7)$$

7.5. Experimental Set-up

A schematic diagram of the control system for a 1-DOF worm-driven joint is shown in Figure 39. The α -unwinding and β -unwinding control algorithms require measurements of motor torques, user-input force, and position. The user-input force is measured using four strain gauges, in a Wheatstone bridge configuration, mounted on a specially designed section of the link. The 3 27W geared DC motors are capable of producing 0.8Nm of torque at the output of the gearbox. Closed-loop regulation of the motors output torque is accomplished by measuring the armature current and implementing a digital PI (Proportional+Integral) controller. All analogue signals are measured and generated using 12-bit analogue-to-digital converters and 12-bit digital-to-analogue converters. The link rotation is recorded using an encoder and appropriate electronics to generate 20000 counts per revolution, giving a resolution of 0.0003142 rads/count (or 0.018°/count). The encoder position is read as a 24 bit number from an HCTL-1100 Motion Control Interface. The frequency of the control loop is set at 600 Hz and link rotational velocity measurement is obtained in software by using a backward difference algorithm.

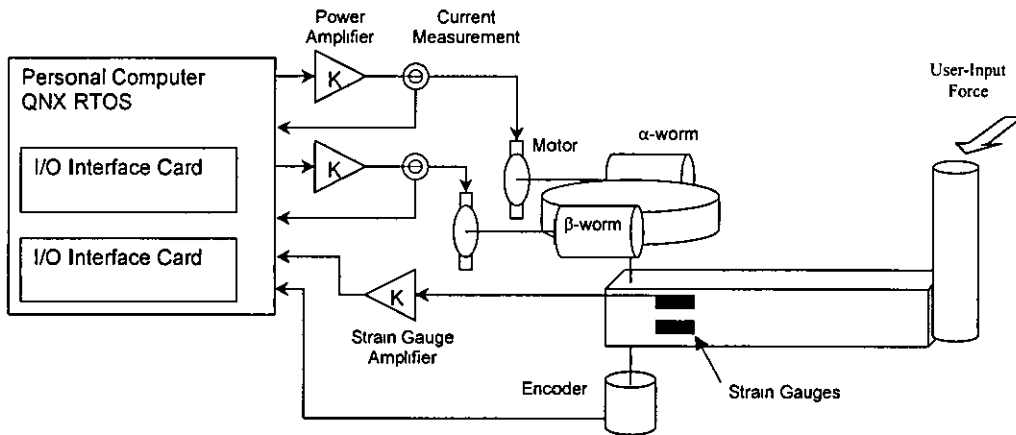


Figure 39 - A schematic overview of the 1-DOF control system

7.6. Results and Discussion

For the model, given by equation (6-7), and the controllers of equations (7-4) and (7-6), the estimated values of the system parameters are given in Table 3. The values of inertia were calculated from the joint geometry, the worm and worm wheel dimensions are assumed to be the pitch centre diameter of the respective gears, while the friction coefficients were determined experimentally. It should be noted that there will be variations in the friction coefficients due to temperature and wear. These variations may affect the implementation of this system in a multi-DOF robot. Therefore, the robot control system should be calibrated to update the estimates of the friction coefficients periodically, or whenever tracking performance is degraded. This could be done automatically.

7.6 1. Simulation results

In order to test the performance of the control strategy and algorithms an implementation of the model, given by equation (6-7), was developed using MATLAB and SIMULINK. Using the values in Table 3 and assuming that the control algorithms had exact knowledge of these parameters. Figure 40 shows the results obtained from the simulation. The user-input which was applied to the system is given by equation (7-8) below and is shown in Figure 40b. Figure 40a shows the desired position generated from the speed demand, where θ_d is given by equation (7-1) with $K_v = 0.01$. Sufficient tracking performance was achieved using the gains given in Table 4. The graph of position error given in Figure 40c highlights this. Figure 40d and Figure 40e show the applied α -motor and β -motor torque used to track the desired trajectory.

$$\tau_d = 2.5 \operatorname{sgn}(\sin(0.5t)) (\tanh(3 \cos(t)) - 1) \quad (7-8)$$

Table 3 - System parameters

| | Worm wheel | α-worm | β-worm |
|-------------------|---------------------------------------|---------------------------------------|--------------------------------------|
| Inertia | $J_{\theta}=0.062833 \text{ Kg m}^2$ | $J_{\alpha}=0.002855 \text{ Kg m}^2$ | $J_{\beta}=0.002855 \text{ Kg m}^2$ |
| Viscous Damping | $C_{f\theta}=0.005 \text{ N.m.s/rad}$ | $C_{f\alpha}=0.001 \text{ N.m.s/rad}$ | $C_{f\beta}=0.001 \text{ N.m.s/rad}$ |
| Gear geometry | $r_{\theta}=0.025 \text{ m}$ | $r_{\alpha}=0.0065 \text{ m}$ | $r_{\beta}=0.0065 \text{ m}$ |
| Static Friction | | $\mu_{s\alpha}=0.12$ | $\mu_{s\beta}=0.16$ |
| Dynamic Friction | | $\mu_{d\alpha}=0.10$ | $\mu_{d\beta}=0.14$ |
| Friction Constant | | $\varepsilon=0.001$ | $\varepsilon=0.001$ |
| Worm lead angle | | $\gamma=0.06071 \text{ rad}$ | $\gamma=0.06071 \text{ rad}$ |

Table 4 - Simulation gains

| Control Direction | Proportional | Integral | Derivative |
|--------------------------|---------------------|--------------------|-------------------|
| α -unwinding | $K_{p\alpha}=5000$ | $K_{a\alpha}=1000$ | $K_{d\alpha}=100$ |
| β -unwinding | $K_{p\beta}=5000$ | $K_{a\beta}=1000$ | $K_{d\beta}=100$ |

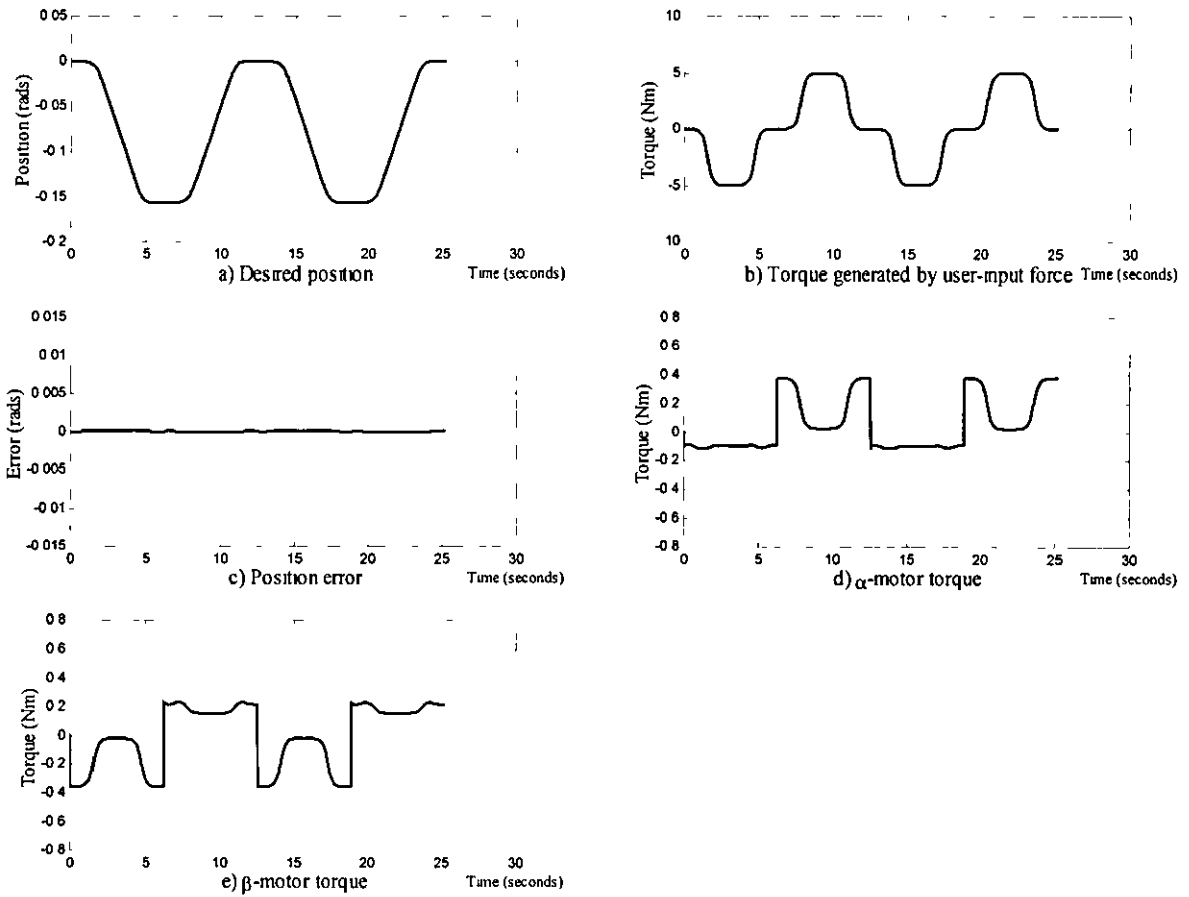


Figure 40 – Dual-worm joint simulation results

7.6.2. Experimental results

The experimental results for a joint with a single worm are compared with those of the dual-worm mechanism to illustrate backlash cancellation. In both experiments the trajectory is generated by manually applying a force to a control handle attached to the robot link. The velocity demand is generated based on equation (7-1) where $K_v = 0.01$. The desired velocity is then digitally integrated and differentiated to give the desired position and desired acceleration, respectively. Figure 41 presents the results for the single-worm mechanism. Figure 41a shows the actual and desired position generated from the user-input force of Figure 41b. Figure 41c displays the time history of the error during the movement and Figure 41d illustrates the motor

torque used to drive the α -worm. It can be seen from Figure 41c that the change in direction of the user-input force causes large position errors in the order of ± 0.01 rads (or $\pm 0.57^\circ$). These errors are attributed to the backlash between the worm and worm wheel present in the single-worm system.

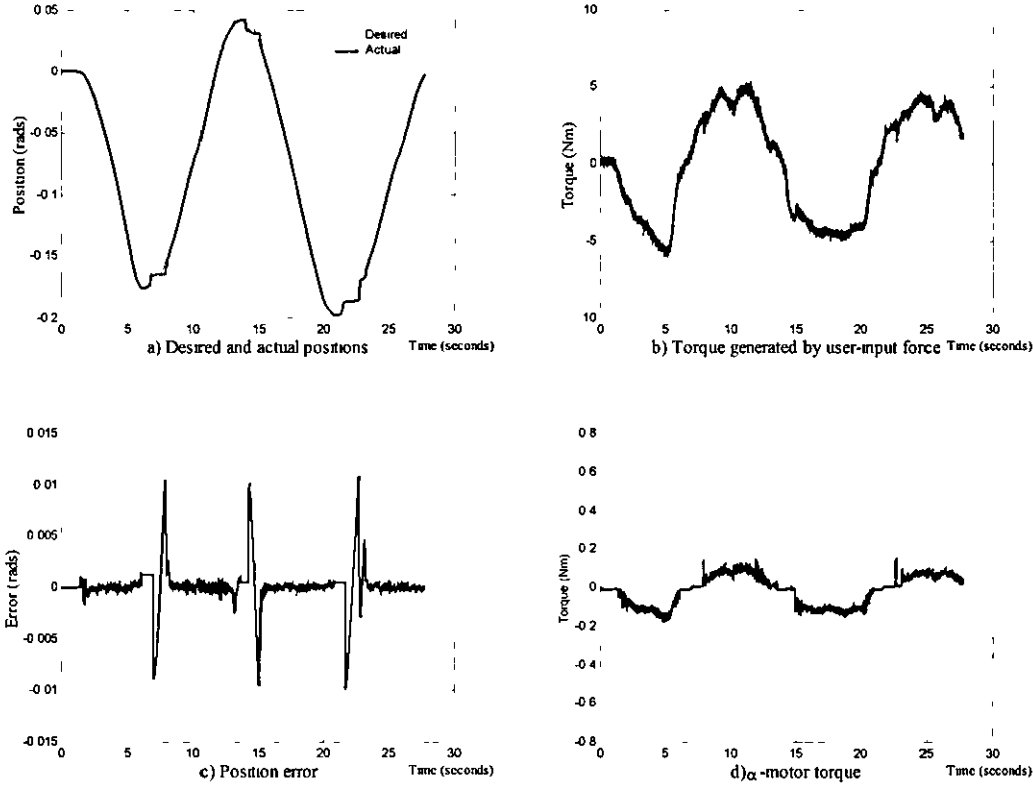
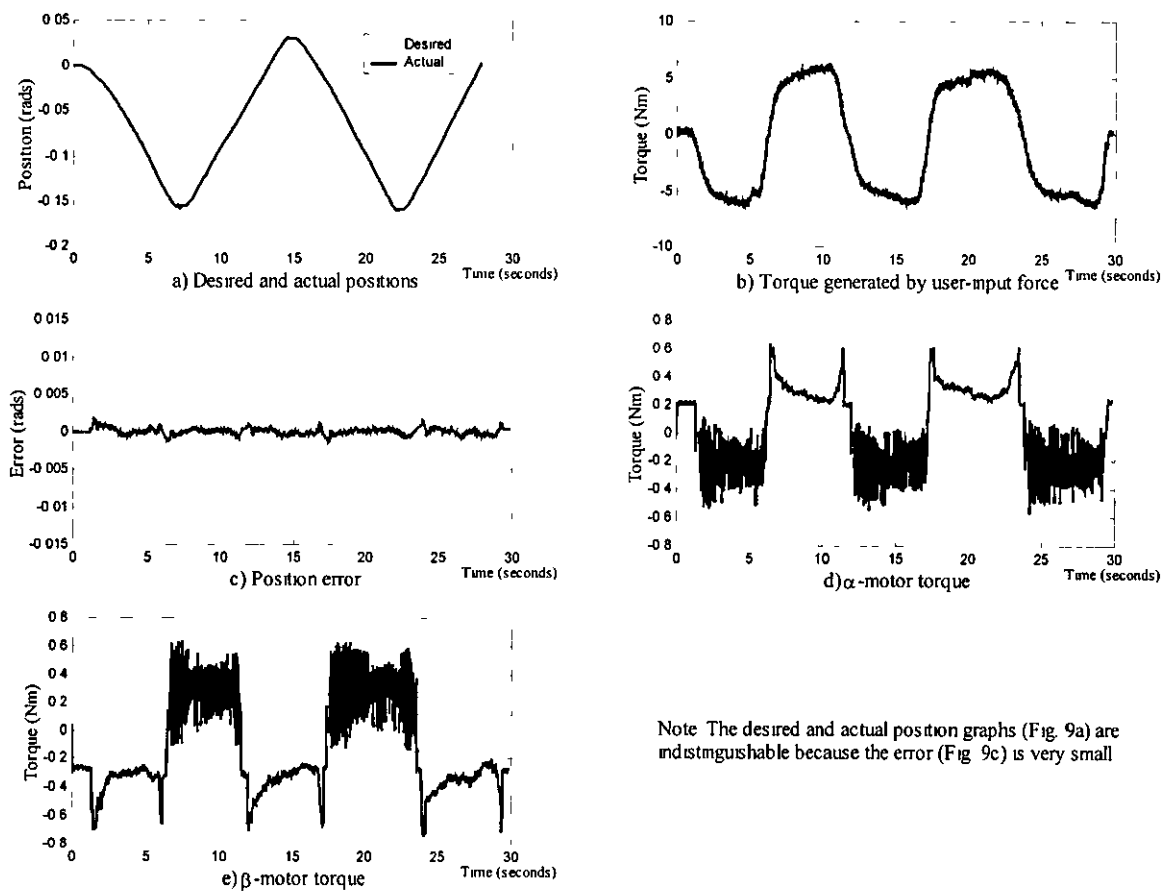


Figure 41 - Single worm joint experimental results

Figure 42 presents the experimental results of a similar motion to that of the simulation as given by Figure 40 with the dual-worm mechanism. Figure 42a shows the desired and actual positions of the joint. The torque generated by the user-input force is shown in Figure 42b, and the position error is illustrated in Figure 42c. It should be noted that the graphs of actual and desired positions in Figure 42a are indistinguishable because the error (shown in Figure 42c) is very small. Figure 42d and Figure 42e also show the motor output torque for the α and β motors, respectively. It is possible to see from Figure 42c that the backlash in the mechanism has been cancelled and the overall error has been reduced to less than ± 0.0015 rads (or $\pm 0.086^\circ$).

Tests have also been carried out to determine how the tool-tip position is affected by an external input from the user when the joint is stationary. It has been possible to measure two different sources of error, flexibility of the link and shaft and also end-float of the worm due to limited pre-loading of the worm angular contact bearings. Table 5 shows the deflection of the tool-tip due to the two kinds of error for user-input forces of 10N and 20N applied at the tool tip. This error was measured at the extremity of the link, a distance 0.4m from the centre of rotation. Although this 1-DOF robot does exhibit a total deflection 0.4mm of error per 10N of user-input force is not considered excessive when compared to similar devices. However, increasing the stiffness of the system could further reduce the link and shaft flexibility.



Note: The desired and actual position graphs (Fig. 9a) are indistinguishable because the error (Fig. 9c) is very small.

Figure 42 - Dual-worm joint experimental results

Table 5 - Static error measurements

| Applied Force | Deflection from zero (mm) | | |
|---------------|---------------------------|---------------------------|-------------------------------|
| | Total | Due to end-float in worms | Due to link/shaft flexibility |
| 10N | 0.36 | 0.16 | 0.20 |
| 20N | 0.74 | 0.32 | 0.42 |

7.7. Conclusions

A suitable control scheme to force the dual-worm mechanism to track a trajectory has been derived based on a mathematical model of the system. A computed-torque control algorithm has been shown to give good results whilst tracking a trajectory generated from the measured user-input torque. The control algorithm has been effective in cancelling backlash and compensating for the non-linear effects of friction generated at the interface of the two worms and worm wheel.

However, when the power to the motors is switched off, high frequency user-input (i.e. shaking of the end-effector) can cause the worms to unwind allowing the mechanism to exhibit backlash. For this reason, when the joint is stationary, each motor applies a small amount of opposing torque to keep the system backlash free.

Chapter 8 Composite Adaptive Control of a One Degree-of-Freedom Worm Driven Manipulator

8.1. Introduction

In the previous chapters, a mathematical model of the mechanism was presented and was used to develop a *computed-torque* control algorithm for tracking a desired path. The algorithm guarantees exact tracking when the robot parameters are known precisely. However, non-linear robot parameters (including frictional and inertial terms) are notoriously difficult to estimate. In this chapter, a composite adaptive control scheme is presented that is used to estimate system parameters such as inertia, damping coefficients and friction coefficients. A review of the concept of the dual-worm robot joint mechanism is given followed by a statement of the mathematical model of the gear system. The strategy for control of the mechanism is presented along with a discussion on the selection of an appropriate adaptive control scheme. Simulation and experimental results are presented showing a comparison between the performance of the computed-torque and composite adaptive controllers.

8.2. Mathematical Model

For simplicity the mathematical model given by equation (6-7) is restated in equation (8-1).

$$J_\tau \theta + C_\tau \dot{\theta} = \tau_u + \frac{\tau_{ma} r_\theta}{\Delta_a r_a} + \frac{\tau_{mb} r_\theta}{\Delta_b r_b} \quad (8-1)$$

where $J_\tau = \left(J_\theta + \frac{J_a r_\theta^2}{\Delta_a r_a^2 \tan \gamma} + \frac{J_b r_\theta^2}{\Delta_b r_b^2 \tan \gamma} \right)$ $C_\tau = \left(C_\theta + \frac{C_a r_\theta^2}{\Delta_a r_a^2 \tan \gamma} + \frac{C_b r_\theta^2}{\Delta_b r_b^2 \tan \gamma} \right)$

$$\Delta_a = \frac{\sin \gamma + \left(\mu_{ab} + (\mu_{ab} - \mu_{ab}) e^{\frac{-|\theta|}{\epsilon \sin \gamma}} \right) \operatorname{sgn}(\theta) \cos \gamma}{\cos \gamma - \left(\mu_{ab} + (\mu_{ab} - \mu_{ab}) e^{\frac{-|\theta|}{\epsilon \sin \gamma}} \right) \operatorname{sgn}(\theta) \sin \gamma}$$

$$\Delta_b = \frac{\sin \gamma - \left(\mu_{ab} + (\mu_{ab} - \mu_{ab}) e^{\frac{-|\theta|}{\epsilon \cos \gamma}} \right) \operatorname{sgn}(\theta) \cos \gamma}{\cos \gamma + \left(\mu_{ab} + (\mu_{ab} - \mu_{ab}) e^{\frac{-|\theta|}{\epsilon \cos \gamma}} \right) \operatorname{sgn}(\theta) \sin \gamma}$$

8.3. Control Strategy

The control strategy for the composite adaptive control scheme is required to perform in the same manner as that of the computed-torque controller. The aim of the control algorithm is to make the manipulator track a desired position command whilst cancelling backlash at the worm interface. The composite adaptive control scheme must also ensure surgeon and patient safety by not allowing:

- (i) any motion against the user,
- (ii) any motion without direct control from the user,
- (iii) any backlash at the worm/wheel interface, or
- (iv) any motion caused by excessive user input-force

In a similar manner to the computed-torque control scheme, effective control of the dual-worm mechanism requires two algorithms referred to as α -unwinding and β -unwinding. α -unwinding control is used to control the position of the joint when the equivalent torque generated by the user-input force $\tau_u < 0$, i.e. when the desired motion is in an anti-clockwise direction. However, when $\tau_u > 0$, the desired motion is in the clockwise direction and β -unwinding control is used to control the joint position. The velocity command is generated from the user-input force in the following way

$$\theta_d = \begin{cases} 0 & \forall |\tau_u| < \tau_{u \min} \\ \theta_{d \max} \operatorname{sgn}(\tau_u) \left(-\frac{2|\tau_u - \tau_{u \min}|}{(\tau_{u \max} - \tau_{u \min})^3} + \frac{3|\tau_u - \tau_{u \min}|^2}{(\tau_{u \max} - \tau_{u \min})^2} \right) & \forall \tau_{u \min} \leq |\tau_u| \leq \tau_{u \max} \\ \theta_{d \max} \operatorname{sgn}(\tau_u) & \forall |\tau_u| > \tau_{u \max} \end{cases} \quad (8-2)$$

where $\theta_{d \max}$, $\tau_{u \min}$, and $\tau_{u \max}$ are positive constants chosen to give a smooth resistance to the user-input. The velocity-limiting algorithm of equation (8-2) is shown graphically in Figure 43. The above velocity command is different from that used in Chapter 7 equation (7-1). This strategy has been employed to ensure sufficient smoothness of the desired position, velocity, and acceleration command signals.

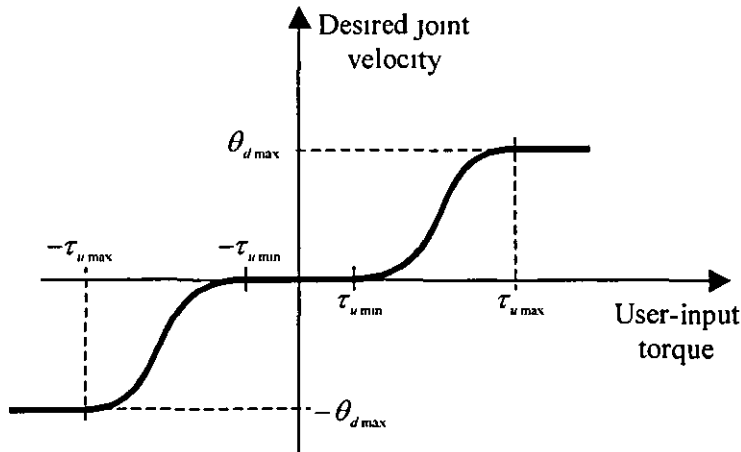


Figure 43 - User defined velocity demand

Craig, Hsu, and Sastry [98] presented an adaptive control scheme for the control of robotic systems that ensures convergence of the parameter error to zero under certain conditions on the desired trajectory, known as persistency of excitation. The control method of Craig, Hsu, and Sastry also requires inversion of the manipulator inertia matrix and measurement of acceleration. Although the parameter resetting technique used in [98] ensures the existence of the inverse of the inertia matrix, the poor acceleration measurement (derived by digitally differentiating the position with respect to time) is undesirable.

Psaltis et al [99] began to explore the idea of using Neural Networks (NN) to control dynamic systems. The paper showed simulation results of a NN trained, using a variety of update algorithms, to learn the behaviour of the plant. It was assumed that the NN would learn exactly the plant dynamics and no mention of system stability was made. In 1991, Ozaki et al [100] presented a NN controller for a 2-DOF manipulator. The results clearly showed the improvement in performance achieved over the computed-torque method. However, no discussion on stability of the overall system was given. In 1994, Safaric and Jezernik [101] presented a method for deriving the learning algorithms for NN control of a 2-DOF SCARA robot. The method is based on linearisation of the NN model and Lyapunov stability theory for adaptive controllers. Both NN and computed-torque controllers were tested on the manipulator and compared to each other. Although, the NN controller was not shown

to significantly outperform the computed-torque algorithm it would be able to compensate for variations in payload mass and friction coefficients. Kwan, Lewis, and Dawson [102] presented a NN controller for a 2-DOF robot where a back-stepping procedure was used to develop robust control algorithms based on a model that included motor dynamics. Two functional-link NNs were used to compensate for complex non-linear functions one NN compensated for non-linear terms in the robot dynamics while the second NN compensated for non-linear terms in the motor dynamics. It was shown that the NN controller considerably outperformed a Proportional-plus-Derivative (PD) controller in simulations. However, the stability results for all of these control methods are based on the need for a persistently exciting trajectory. The control method that has been proposed for the control of the worm-driven joint involves switching between two controllers: one controller for controlling motion in the positive direction and another for the negative direction. Due to the nature of the human input required to move the system, it is not possible to guarantee a persistently exciting trajectory for each controller.

Due to the requirement of acceleration measurement and the problems of persistent excitation described above, a control method that relies on a less restrictive condition than persistency of excitation, known as the infinite integral condition, is used. Slotine and Li [103] proposed a composite adaptive controller that extracts information from both the tracking error and a prediction error in the filtered joint torque. This method consists of filtering the joint torque, making an estimate of this filtered quantity, designing a controller to track the desired trajectory, and finally selecting a parameter update rule that ensures convergence of both the tracking and parameter errors to zero. This torque filtering method, which is analysed in detail by Lewis, Abdallah, and Dawson [104], removes the need for acceleration measurements and inversion of the manipulator inertia matrix.

8.4. Composite Adaptive Controller

In the development of the composite adaptive controller, the error between the desired position and the actual position of the link is defined by equation (8-3), and the quantity r , known as the filtered tracking error, is defined by equation (8-4). Both α -unwinding and β -unwinding control strategies are developed below followed by a stability analysis of the controllers

$$e = \theta_d - \theta \quad (8-3)$$

$$r = e + \lambda e \quad (8-4)$$

8.4.1. α -unwinding control

The equation of motion for the system of gears, given in equation (6-7), can be used to solve for the α -motor torque, $\tau_{m\alpha}$, by multiplying both sides by $\Delta_\alpha r_\alpha / r_\theta$. That is

$$\tau_{m\alpha} = J_{1\alpha} \ddot{\theta} + C_{1\alpha} \dot{\theta} - \tau_u \frac{\Delta_\alpha r_\alpha}{r_\theta} - \tau_{m\beta} \frac{\Delta_\alpha r_\alpha}{\Delta_\beta r_\beta} \quad (8-5)$$

where

$$J_{1\alpha} = J_1 \frac{\Delta_\alpha r_\alpha}{r_\theta} \quad C_{1\alpha} = C_1 \frac{\Delta_\alpha r_\alpha}{r_\theta}$$

The assumption is made that the transition between static and dynamic friction is very fast (because of the $1/\sin \gamma$ term in both Δ_α and Δ_β) hence once motion has started Δ_α and Δ_β are constant. Under this assumption it is possible to write $\tau_{m\alpha}$ as a linear combination of constant unknown parameters, Φ_α , and known functions, W_α . Therefore, equation (8-5) can be written as

$$\tau_{m\alpha} = W_\alpha \Phi_\alpha \quad (8-6)$$

where

$$W_\alpha = \begin{bmatrix} \ddot{\theta} & \dot{\theta} & \tau_u & \tau_{m\beta} \end{bmatrix} \quad \Phi_\alpha = \begin{bmatrix} J_{1\alpha} & C_{1\alpha} & -\frac{\Delta_\alpha r_\alpha}{r_\theta} & -\frac{\Delta_\alpha r_\alpha}{\Delta_\beta r_\beta} \end{bmatrix}^T$$

Equation (8-5) can also be written as

$$\tau_{m\alpha} = h_\alpha + g_\alpha \quad (8-7)$$

where

$$h_\alpha = \frac{dh_\alpha}{dt} = \frac{d}{dt}(J_{1\alpha}\theta)$$

and

$$g_\alpha = -J_{1\alpha}\theta + C_{1\alpha}\theta - \tau_u \frac{\Delta_\alpha r_\alpha}{r_\theta} - \tau_{m\beta} \frac{\Delta_\alpha r_\alpha}{\Delta_\beta r_\beta}$$

Equation (8-5) has now been separated in a way that allows the acceleration term, θ , to be filtered out. By filtering both sides of equation (8-7) it is possible to write the filtered torque equation given by equation (8-8). f is the impulse response of the linear stable, strictly proper filter, and $*$ is used to denote the convolution operation.

$$\tau_{f\alpha} = f * \tau_{m\alpha} = f * h_\alpha + f * g_\alpha \quad (8-8)$$

By the property of convolution, it is also possible to write

$$f * h_\alpha = \int_0^t f(t-v)h_\alpha(v)dv \quad (8-9)$$

and integrating by parts yields

$$f * h_\alpha = [f(t-v)h_\alpha(v)]_0^t - \int_0^t f(t-v)\dot{h}_\alpha(v)dv \quad (8-10)$$

which in turn yields

$$f * h_\alpha = -f * h_\alpha + f(0)h_\alpha - f * h_\alpha(0) \quad (8-11)$$

Substitution of equation (8-11) into equation (8-8), replacing h_α and g_α from equation (8-7) and noting that $J_{1\alpha}$ is constant and the filter and velocity are initialised to $f(0)=c$ and $\theta(0)=0$ respectively yields

$$\tau_{f\alpha} = -f * (J_{1\alpha}\theta) + cJ_{1\alpha}\theta + f * \left(C_{1\alpha}\theta - \tau_u \frac{\Delta_\alpha r_\alpha}{r_\theta} - \tau_{m\beta} \frac{\Delta_\alpha r_\alpha}{\Delta_\beta r_\beta} \right) = W_{f\alpha} \Phi_\alpha \quad (8-12)$$

where, $W_{fa} = \begin{bmatrix} -f^* \theta + c \theta & f^* \theta & f^* \tau_u & f^* \tau_{m\beta} \end{bmatrix}$ W_{fa} is referred to as the filtered regression matrix and Φ_a is the same as in (8-6). Assuming that τ_{ma} , $\tau_{m\beta}$, τ_u , and θ are measurable, i.e. that the filtered regression matrix is known, it is possible to define the estimate of the filtered α -motor torque, $\hat{\tau}_{fa}$, based on the estimate of the unknown parameters, $\hat{\Phi}_a$, such that

$$\hat{\tau}_{fa} = W_{fa} \hat{\Phi}_a \quad (8-13)$$

The error in the estimate of the filtered torque is defined as

$$\tilde{\tau}_{fa} = \tau_{fa} - \hat{\tau}_{fa} \quad (8-14)$$

In order to design a controller for the system consider the Lyapunov-like function given by equation (8-15) and its time derivative, equation (8-16), where $r = e + \lambda e$ is the filtered tracking error, $e = \theta_d - \theta$ is the tracking error, $\tilde{\Phi}_a = \Phi_a - \hat{\Phi}_a$ is the error in the parameter estimates and P_a is a time-varying symmetric matrix

$$V_a = \frac{1}{2} J_{\tau a} r^2 + \frac{1}{2} \tilde{\Phi}_a^T P_a^{-1} \tilde{\Phi}_a \quad (8-15)$$

$$\dot{V}_a = J_{\tau a} r \dot{r} + \tilde{\Phi}_a^T P_a^{-1} \dot{\tilde{\Phi}}_a + \frac{1}{2} \dot{\tilde{\Phi}}_a^T P_a^{-1} \tilde{\Phi}_a \quad (8-16)$$

Making the following observation

$$J_{\tau a} r = J_{\tau a} (\theta_d + \lambda e) - J_{\tau a} \theta \quad (8-17)$$

and replacing for $J_{\tau a} \theta$ from equation (8-5) yields

$$J_{\tau a} r = J_{\tau a} (\theta_d + \lambda e) + C_{\tau a} \theta - \tau_{ma} - \tau_u \frac{\Delta_a r_a}{r_0} - \tau_{m\beta} \frac{\Delta_a r_a}{\Delta_\beta r_\beta} \quad (8-18)$$

which can be written as

$$J_{\tau a} r = Y_a \Phi_a - \tau_{ma} \quad (8-19)$$

where $Y_a = \begin{bmatrix} (\theta_d + \lambda e) & \theta & \tau_u & \tau_{m\beta} \end{bmatrix}$ Selecting the control input, τ_{ma} , as

$$\tau_{ma} = Y_a \hat{\Phi}_a + K_{va} r \quad (8-20)$$

and substituting into equation (8-19) and subsequently into equation (8-16) yields

$$V_a = -K_{va} r^2 + \tilde{\Phi}_a^T \left(P_a^{-1} \tilde{\Phi}_a + Y_a^T r \right) + \frac{1}{2} \tilde{\Phi}_a^T P_a^{-1} \tilde{\Phi}_a. \quad (8-21)$$

The least-squares adaptive update law proposed Li and Slotine [105] is used here. This is given by equations (8-22) and (8-23).

$$\tilde{\Phi}_a = \Phi_a - \hat{\Phi}_a = -P_a Y_a^T r - P_a W_{fa}^T \tilde{\tau}_{fa} \quad (8-22)$$

$$P_a^{-1} = W_{fa}^T W_{fa} \quad (8-23)$$

Using equation (8-22) and replacing for $\tilde{\tau}_{fa}$ from equations (8-13) and (8-14) and noting that Φ_a is a vector of constant parameters, (i.e. $\dot{\Phi}_a = 0$), yields

$$\hat{\Phi}_a = P_a Y_a^T r + P_a W_{fa}^T (\tau_{fa} - W_{fa} \hat{\Phi}_a) \quad (8-24)$$

Using the matrix identity given by equation (8-25) [106],

$$P P^{-1} = I \quad (8-25)$$

and differentiating with respect to time yields

$$\dot{P} P^{-1} + P \dot{P}^{-1} = 0 \quad (8-26)$$

$$\dot{P} = -P \dot{P}^{-1} P \quad (8-27)$$

Subsequently, equation (8-23) may be written more conveniently as equation (8-28).

$$\dot{P}_a = -P_a W_{fa}^T W_{fa} P_a \quad (8-28)$$

Since \mathbf{P}_a is negative semi-definite, it follows that

$$\lim_{t \rightarrow \infty} \lambda_{\max} \{\mathbf{P}_a\} = 0 \quad (8-29)$$

and

$$\lim_{t \rightarrow \infty} \lambda_{\min} \{\mathbf{P}_a^{-1}\} = \infty \quad (8-30)$$

where $\lambda_{\min} \{\mathbf{A}\}$ and $\lambda_{\max} \{\mathbf{A}\}$ represent the minimum and maximum eigenvalues of the matrix \mathbf{A} respectively. Equation (8-30) is also referred to as the infinite integral condition. Substitution of the least-squares estimator and parameter update law given by equations (8-22) and (8-23) into leads to equation (8-21) yields

$$V_a = -K_{va} r^2 - \frac{1}{2} \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a \quad (8-31)$$

Equation (8-31) is obtained as follows. Substitution of equation (8-22) and (8-23) into equation (8-21) yields

$$V_a = -K_{va} r^2 + \tilde{\Phi}_a^T \left(\mathbf{P}_a^{-1} (-\mathbf{P}_a Y_a^T r - \mathbf{P}_a W_{fa}^T \tilde{\tau}_{fa}) + Y_a^T r \right) + \frac{1}{2} \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a. \quad (8-32)$$

Equation (8-32) reduces to

$$V_a = -K_{va} r^2 + \tilde{\Phi}_a^T (-Y_a^T r - W_{fa}^T \tilde{\tau}_{fa} + Y_a^T r) + \frac{1}{2} \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a \quad (8-33)$$

which in turn yields

$$V_a = -K_{va} r^2 - \tilde{\Phi}_a^T W_{fa}^T \tilde{\tau}_{fa} + \frac{1}{2} \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a \quad (8-34)$$

and noting $\tilde{\tau}_{fa} = W_{fa} \tilde{\Phi}_a$, it follows that

$$V_a = -K_{va} r^2 - \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a + \frac{1}{2} \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a \quad (8-35)$$

which can be simplified to

$$V_\alpha = -K_{v\alpha} r^2 - \frac{1}{2} \tilde{\Phi}_\alpha^T W_{f\alpha}^T W_{f\alpha} \tilde{\Phi}_\alpha \quad (8-36)$$

8.4.2 β -unwinding control

Similarly, for β -unwinding control, equation (8-1) can be used to solve for the β -motor torque by multiplying both sides by $\Delta_\beta r_\beta / r_o$ as given in equation (8-37)

$$\tau_{m\beta} = J_{\tau\beta} \theta + C_{\tau\beta} \theta - \tau_u \frac{\Delta_\beta r_\beta}{r_o} - \tau_{ma} \frac{\Delta_\beta r_\beta}{\Delta_\alpha r_\alpha} = W_\beta \Phi_\beta \quad (8-37)$$

where $W_\beta = \begin{bmatrix} \theta & \theta & \tau_u & \tau_{ma} \end{bmatrix}$ $\Phi_\beta = \begin{bmatrix} J_{\tau\beta} & C_{\tau\beta} & -\frac{\Delta_\beta r_\beta}{r_o} & -\frac{\Delta_\beta r_\beta}{\Delta_\alpha r_\alpha} \end{bmatrix}^T$

Following the method outlined above the control input, $\tau_{m\beta}$, can be selected as

$$\tau_{m\beta} = Y_\beta \hat{\Phi}_\beta + K_{v\beta} r \quad (8-38)$$

where $Y_\beta = \begin{bmatrix} (\theta_d + \lambda e) & \theta & \tau_u & \tau_{ma} \end{bmatrix}$

The least-squares adaptive update law is given in equations (8-39) and (8-40) and the time derivative of the Lyapunov function is given by equation (8-41).

$$\begin{aligned} \tilde{\Phi}_\beta &= -\dot{\hat{\Phi}}_\beta = -P_\beta Y_\beta^T r - P_\beta W_{f\beta}^T \tilde{\tau}_{f\beta} \\ \dot{\hat{\Phi}}_\beta &= P_\beta Y_\beta^T r + P_\beta W_{f\beta}^T (\tau_{f\beta} - W_{f\beta} \hat{\Phi}_\beta) \end{aligned} \quad (8-39)$$

$$\begin{aligned} P_\beta^{-1} &= W_{f\beta}^T W_{f\beta} \\ P_\beta &= -P_\beta W_{f\beta}^T W_{f\beta} P_\beta \end{aligned} \quad (8-40)$$

$$V_\beta = -K_{v\beta} r^2 - \frac{1}{2} \tilde{\Phi}_\beta^T W_{f\beta}^T W_{f\beta} \tilde{\Phi}_\beta \quad (8-41)$$

8 4 3 Stability Analysis

The Lyapunov functions for α and β -unwinding may be written as

$$V_i = \frac{1}{2} J_{\tau i} r^2 + \frac{1}{2} \tilde{\Phi}_i^T P_i^{-1} \tilde{\Phi}_i \quad (8-42)$$

where the subscript i should be replaced by α or β for α -unwinding and β -unwinding respectively. Similarly, the time-derivative of the Lyapunov function given by equations (8-31) or (8-41) can also be expressed as

$$\dot{V}_i = -K_w r^2 - \frac{1}{2} \tilde{\Phi}_i^T W_{\beta}^T W_{\beta} \tilde{\Phi}_i \leq 0 \quad (8-43)$$

By showing that the second derivative of the Lyapunov function is bounded, so proving that \dot{V}_i is uniformly continuous, Barbalat's lemma given in Appendix F can be invoked to show that the tracking error, r , and parameter error, $\tilde{\Phi}_i$, both converge to zero. Firstly, \dot{V}_i is obtained by differentiating equation (8-43) to give equation (8-44)

$$\dot{V}_i = -2K_w r \dot{r} - \tilde{\Phi}_i^T W_{\beta}^T W_{\beta} \tilde{\Phi}_i - \tilde{\Phi}_i^T W_{\beta}^T W_{\beta} \dot{\tilde{\Phi}}_i \quad (8-44)$$

Therefore, in order to prove that \dot{V}_i is bounded and V_i is uniformly continuous, it is necessary and sufficient to prove that r , \dot{r} , $\tilde{\Phi}_i$, $\dot{\tilde{\Phi}}_i$, W_{β} , and \dot{W}_{β} are bounded. \dot{V}_i is at least negative semi-definite implying that $\dot{V}_i \leq \dot{V}_i(0)$ and that r and $\tilde{\Phi}_i$ are also bounded. Subsequently, from the definition of r in equation (8-4), e and \dot{e} are also bounded. Considering Y_i as defined in equations (8-19) and (8-38), i.e. $Y_i = [\theta_d + \lambda e \quad \theta \quad \tau_u \quad \tau_m]^T$ where i should be replaced by β for α -unwinding and α for β -unwinding and assuming that the inputs, τ_u , τ_m , θ_d , θ , and $\dot{\theta}_d$ are bounded then θ , $\dot{\theta}$, and Y_i must also be bounded. Substitution of the control law (8-20) or (8-38) into equation (8-19) leads to

$$J_n \dot{r} = Y_i \tilde{\Phi}_i - K_w r \quad (8-45)$$

Noting that $J_n > 0$ then r and \dot{r} are also bounded. Subsequently, from equations (8-5) or (8-37) and from the assumption that Φ_i is bounded, τ_m , W_i , and $\dot{\Phi}_i$ are bounded. From the definition of the filter, f , and the properties of a stable transfer function given in Appendix F it can be seen that τ_{fa} , W_{β} and \dot{W}_{β} are also bounded. Finally, the parameter update law of equation (8-22) or (8-39), written as

$$\tilde{\Phi}_i = -P_i Y_i^T r - P_i W_{f_i}^T \tilde{\tau}_{f_i} \quad (8-46)$$

is used to prove that V_i is bounded. From equations (8-23) or (8-40), P_i is bounded by $P_i(0)$ since $P_i \leq 0$ is negative semi-definite. Using equations (8-46), (8-13), and (8-14) and the fact that the filter, f , has a stable transfer function, it can be seen that $\tilde{\Phi}_i$ is bounded. Application of Barbalat's lemma, given in Appendix F, yields

$$\lim_{t \rightarrow \infty} V_i = 0 \quad (8-47)$$

and

$$\lim_{t \rightarrow \infty} r = 0 \quad (8-48)$$

Since the relationship between r and e , from equation (8-4), may also be written in terms of the strictly proper, asymptotically stable transfer function, $H(s)$, such that

$$e(s) = H(s)r(s) \quad (8-49)$$

Therefore

$$\lim_{t \rightarrow \infty} r = 0 \quad \Rightarrow \quad \lim_{t \rightarrow \infty} e = 0 \quad (8-50)$$

It is also possible to define the type of stability for the parameter error. Since V_i is bounded by $V_i(0)$, if the infinite integral condition given in equation (8-51) holds, it must be concluded that the parameter error tends to zero, i.e. $\lim_{t \rightarrow \infty} \tilde{\Phi}_i = 0$. The infinite integral condition is less restrictive than the persistency of excitation condition discussed above, owing to the fact that as long as there is input to the system and that the system is capable of motion it is easy to prove that equation (8-51) holds true. A persistently exciting trajectory/input, on the other hand, it is not always easy to derive or generate in most robotic applications.

$$\lim_{t \rightarrow \infty} \lambda_{\min} \left\{ \int_0^t W_{f_i}^T(\sigma) W_{f_i}(\sigma) d\sigma \right\} = \infty \quad (8-51)$$

8.4.4 Experimental Set-up

The same equipment and configurations were used as given in Chapter 7. The only difference between the results presented here and the results of Chapter 7 are that 6.5W geared DC motors were used instead of the original 3.27W motors. The frequency of the control loop is set at 600 Hz.

8.5. Results and Discussion

8.5.1. Simulation Results

The mathematical model of the dual worm-driven joint was simulated using SIMULINK. The parameters of the model were set according to Table 6 and the control parameters set using Table 7. The dynamic effects of the α and β motors are cancelled using high gain PI (Proportional-plus-Integral) torque control feedback loop assuming that motor output is measured without error. During the simulation the β -motor is set to follow the worm with constant torque under α -unwinding control. Similarly, during β -unwinding the α -motor is also set to follow the worm-wheel with constant torque. The matrices P_α and P_β were chosen to give well damped parameter estimates, and the parameter estimates themselves, $\hat{\phi}_\alpha$ and $\hat{\phi}_\beta$, were initialised with arbitrary values according to equation (8-52). Torque filtering was performed using the linear filter given by the first-order transfer function of equation (8-53). The same filter was used in both the α -unwinding and β -unwinding controllers.

Table 6 - Simulation model parameters

| | Worm wheel | α -worm | β -worm |
|-------------------|---------------------------------------|---------------------------------------|--------------------------------------|
| Inertia | $J_\theta=0.25 \text{ Kg.m}^2$ | $J_\alpha=0.003698 \text{ Kg m}^2$ | $J_\beta=0.003698 \text{ Kg m}^2$ |
| Viscous Damping | $C_{f\theta}=0.005 \text{ N m s/rad}$ | $C_{f\alpha}=0.005 \text{ N m.s/rad}$ | $C_{f\beta}=0.005 \text{ N m s/rad}$ |
| Gear geoemtry | $r_\theta=0.025 \text{ m}$ | $r_\alpha=0.0065 \text{ m}$ | $r_\beta=0.0065 \text{ m}$ |
| Static Friction | | $\mu_{s\alpha}=0.14$ | $\mu_{s\beta}=0.14$ |
| Dynamic Friction | | $\mu_{d\alpha}=0.12$ | $\mu_{d\beta}=0.12$ |
| Friction Constant | | $\varepsilon=0.001$ | $\varepsilon=0.001$ |
| Worm lead angle | | $\gamma=0.05236 \text{ rad}$ | $\gamma=0.05236 \text{ rad}$ |

Table 7 - Simulation controller gains

| Control Direction | Tracking error filter gain | Controller gain |
|---------------------|----------------------------|------------------|
| α -unwinding | $\lambda_\alpha=50$ | $K_{v\alpha}=20$ |
| β -unwinding | $\lambda_\beta=50$ | $K_{v\beta}=20$ |

$$\mathbf{P}_\alpha(0) = \mathbf{P}_\beta(0) = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad \Phi_\alpha(0) = \begin{bmatrix} 0.4 \\ 0.1 \\ 0.1 \\ 0.5 \end{bmatrix} \quad \Phi_\beta(0) = \begin{bmatrix} 0.1 \\ 0.3 \\ 0.1 \\ 0.1 \end{bmatrix} \quad (8-52)$$

$$f(s) = \frac{1}{s+1} \quad (8-53)$$

Three user-input torque commands, τ_{u1} , τ_{u2} , and τ_{u3} , given by equations (8-54), (8-55), and (8-56) respectively were applied to the model. Approximately one cycle of each of the three torque inputs is shown in Figure 44.

$$\tau_{u1} = 5 \tanh(\cos(4t) - 0.76152) \operatorname{sgn}(\sin(t/2)) \quad (8-54)$$

$$\tau_{u2} = 8 \tanh(\cos(4t) - 0.76152) \operatorname{sgn}(\sin(t/2)) \quad (8-55)$$

$$\tau_{u3} = 8 (\cos(4t) + 1.5) \sin(t/2) \quad (8-56)$$

The desired velocity for system is calculated using equation (8-2) with $\theta_{d\max} = 0.1 \text{ rad/s}$, $\tau_{u\max} = 10 \text{ Nm}$, and $\tau_{u\min} = 1 \text{ Nm}$. $\tau_{u\max}$ and $\tau_{u\min}$ are shown in Figure 44 by dotted lines. Figure 45 shows the desired position generated from the integral of equation (8-2). With the system parameters given by Table 6, the actual parameter vectors Φ_a and Φ_p can be calculated for $|\theta| \neq 0$ as given in equation (8-57).

$$\Phi_a = \Phi_p = \begin{bmatrix} 0.162 \\ 0.223 \\ 0.017 \\ 0.387 \end{bmatrix} \quad (8-57)$$

Figure 46 shows the position error recorded during the simulations for each user-input. The various rates at which the magnitude of the tracking error decays can be clearly seen. Figure 47 shows the parameter estimates of the simulated system with the user-input defined by equation (8-54). This figure illustrates that for the first user-input command the estimated parameters do not converge on the expected values given by equation (8-57). The expected values of the parameters are shown by dotted lines in Figure 47, Figure 48, and Figure 49. Figure 48 and Figure 49 show the parameter estimates with the user-inputs of equations (8-55) and (8-56) respectively. However, with these two user-input commands the parameter error converges to zero. The more 'exciting' input given by equation (8-55) yields slower convergence of the parameter estimates to the true values than the input of equation (8-56) as shown in Figure 48 and Figure 49 respectively.

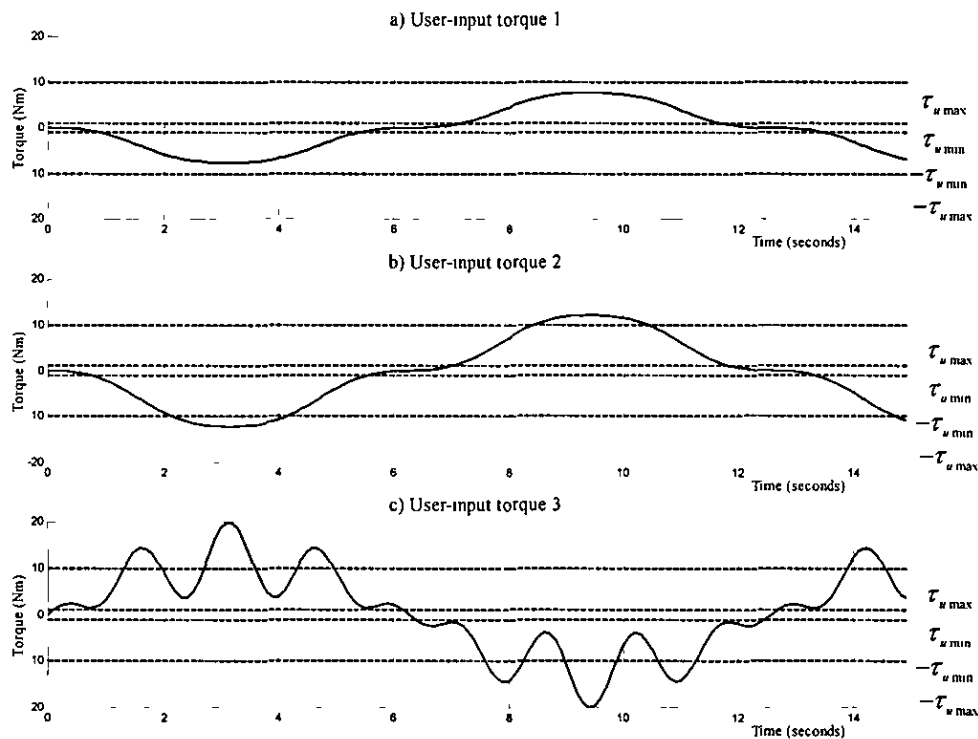


Figure 44 – The three user-input torque commands used in the simulation

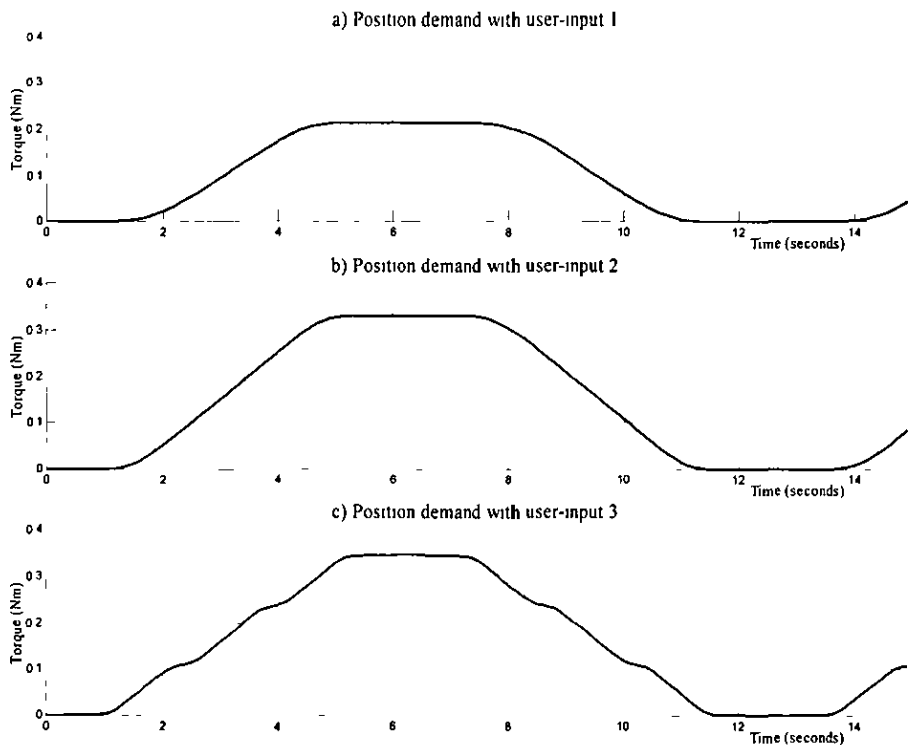


Figure 45 - Desired position demand for the three user-inputs

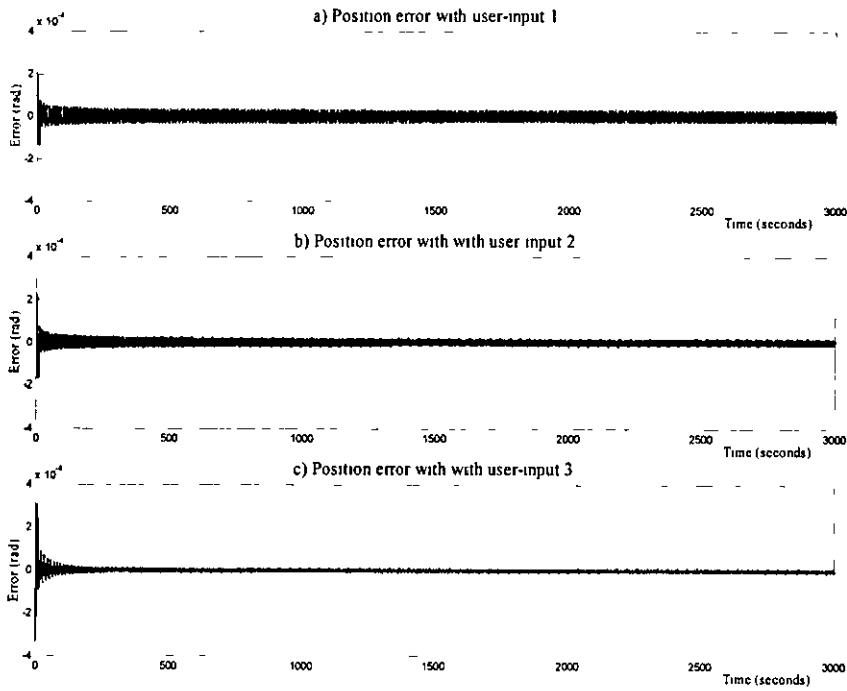


Figure 46 - Comparison of tracking errors during the simulation of the three user-input torque commands

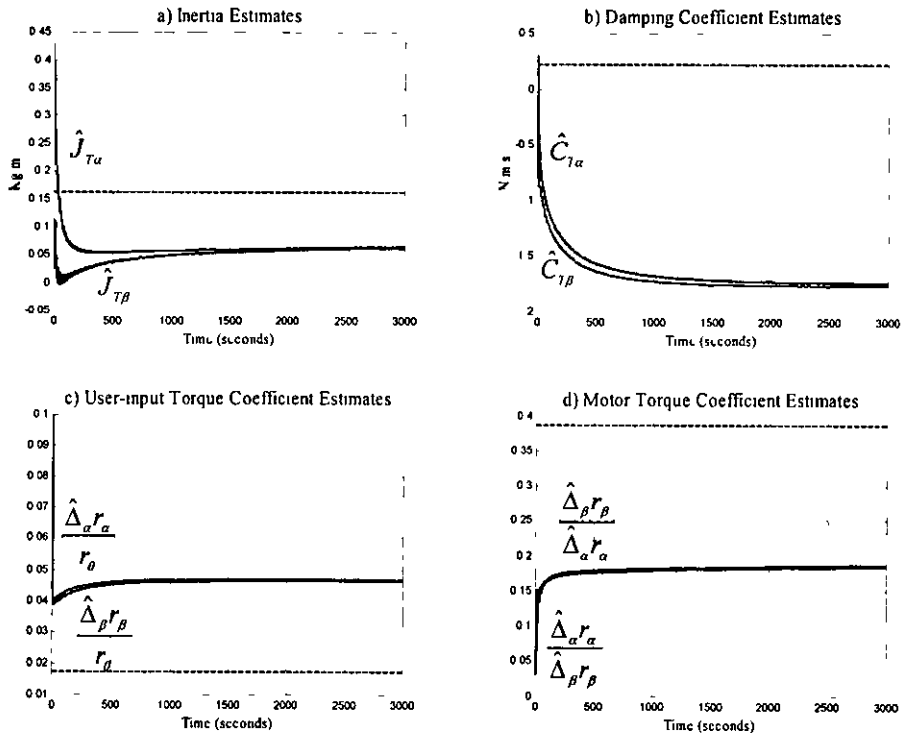


Figure 47 – Simulation parameter estimates with user-input 1

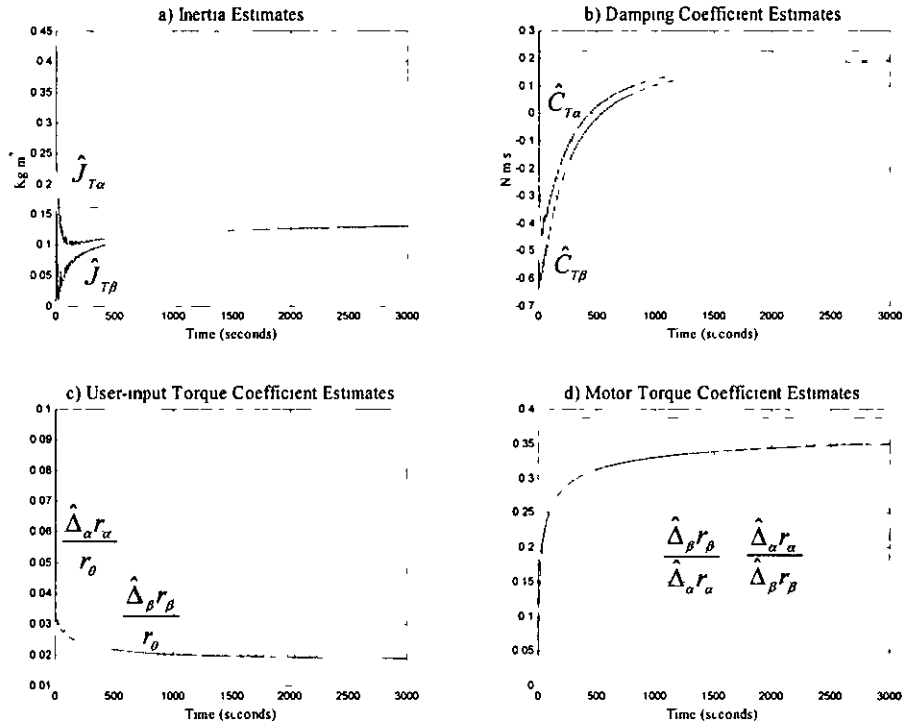


Figure 48 – Simulation parameter estimates with user-input 2

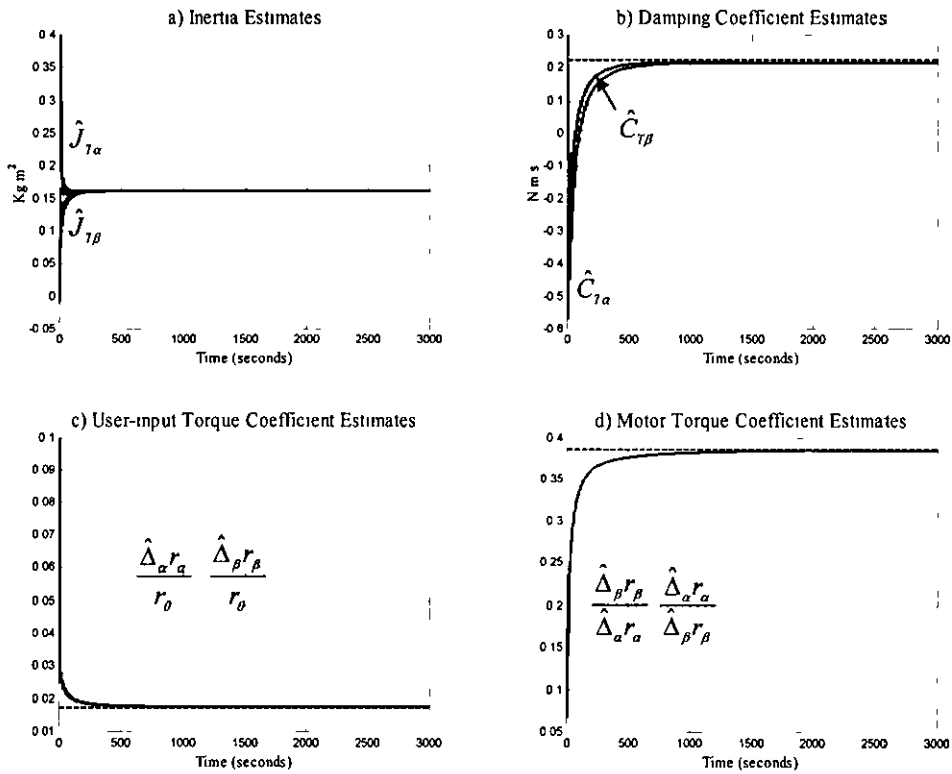


Figure 49 – Simulation parameter estimates with user-input 3

It should be noted that the magnitude of the user-input force affects the convergence of the parameters. The problem arises from the choice of θ_d in equation (8-2). With the user-input defined by equation (8-54), shown in Figure 44a, the desired velocity function never saturates, i.e. $\theta_d < \theta_{d\max}$. Thus the dependence of θ_d on τ_u is very strong and as inputs they appear very similar. Under these conditions the composite adaptive controller cannot extract enough information from these inputs to determine the correct parameter estimates. However, the parameter estimates and tracking error remain bounded at all times.

The user-inputs given by equations (8-54) and (8-55) differ only in magnitude. This difference significantly changes the appearance of θ_d with respect to τ_u in the region where $|\tau_u| > \tau_{u\max}$. The composite adaptive controller is able to extract more information about the system with the user-inputs defined by equations (8-55) and (8-56), and shown in Figure 44b and Figure 44c respectively. These effects must be taken into account during experimental work.

8.5.2 Experimental Results

The experimental results of the computed-torque controller are compared to the results obtained from the composite adaptive controller. In both experiments the applied user-input force generates a command velocity based on equation (8-2), where $\theta_{d\max} = 0.05$ rad/s, $\tau_{u\max} = 10$ Nm, $\tau_{u\min} = 1$ Nm. The desired velocity is then digitally integrated and differentiated to give the desired position and desired acceleration, respectively. The matrices P_α and P_β and the vectors $\hat{\phi}_\alpha$ and $\hat{\phi}_\beta$ were initialised as in equation (8-52) and the controller gains set as shown in Table 8. The torque filter that was used in the experiments for both α -unwinding and β -unwinding control is given by the first-order transfer function of equation (8-58). The filter of equation (8-58) used in these experiments differs from the filter used in the simulation tests (given in equation (8-53)) because it has been tuned to minimise the effects of quantisation and sensor noise that are not modelled in the simulation.

Table 8 - Experimental controller gains

| Control Direction | Tracking error filter gain | Controller gain |
|---------------------|----------------------------|------------------|
| α -unwinding | $\lambda_{\alpha}=50$ | $K_{v\alpha}=35$ |
| β -unwinding | $\lambda_{\beta}=50$ | $K_{v\beta}=35$ |

$$f(s) = \frac{10}{s+10} \quad (8-58)$$

In a similar manner to the simulation tests, three user-input torque commands were manually applied to the joint mechanism as shown by Figure 50. Figure 50a, Figure 50b, and Figure 50c show the user-input commands similar to those given in equations (8-54), (8-55), and (8-56) respectively. Figure 51 shows the desired and actual position, derived from the integration of equation (8-2), for the three user-input torque commands. The graphs of desired and actual position in each are indistinguishable because the error is very small.

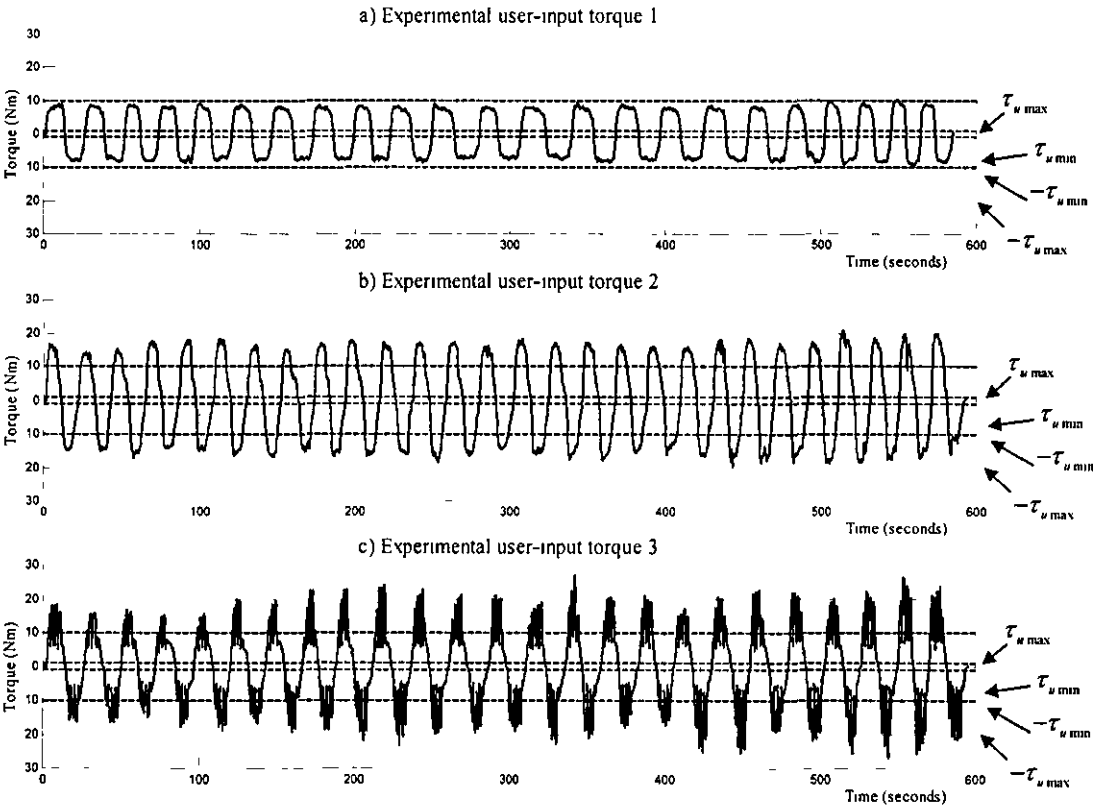


Figure 50 - Experimental user-input torque commands

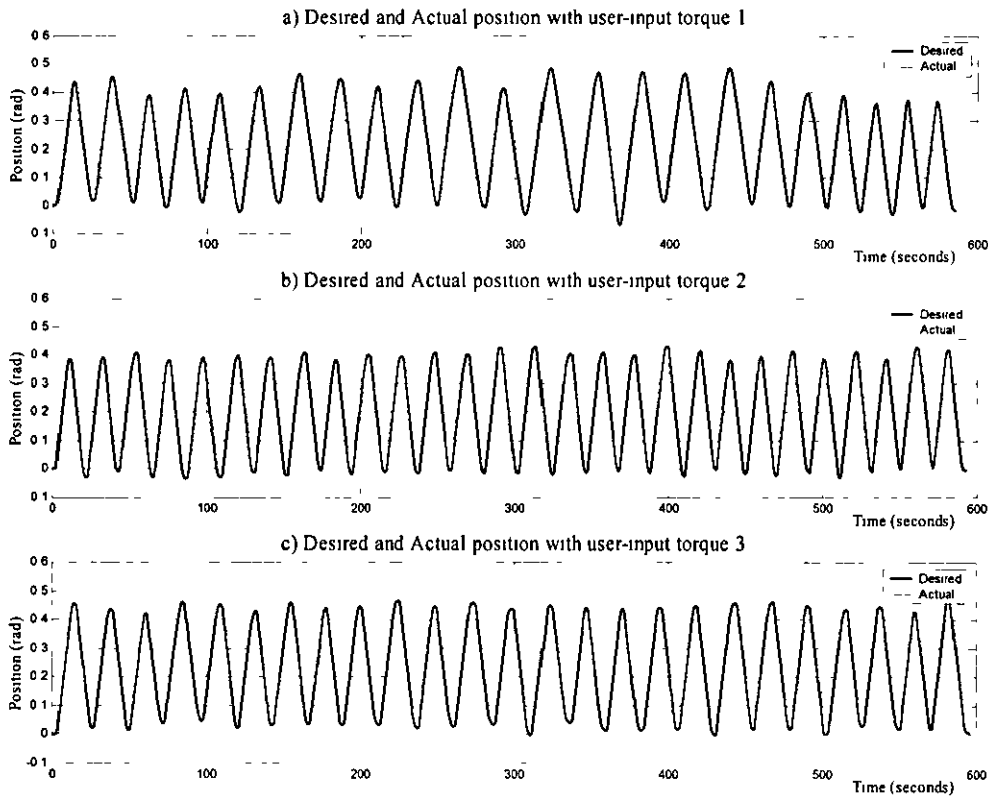


Figure 51 - Experimental desired and actual position

The system position error using the composite adaptive and computed-torque controllers is shown in Figure 52. Although it has not been possible to generate exactly the same trajectory (because the user force is applied manually) Figure 52 shows a comparison between the computed-torque method, presented in Chapter 7, and the composite adaptive control scheme developed in this chapter. Figure 52a shows the performance of the computed-torque controller compared to that of the composite adaptive controller (with the user-input command of Figure 50a). Similarly, Figure 52b and Figure 52c illustrate the comparison for the user-inputs given by Figure 50b and Figure 50c respectively. Computed-torque control was performed with user-inputs similar to those given in Figure 50. The torque command could not be reproduced accurately owing to the input being applied manually. The computed-torque controller error is shown offset by 0.005 rad for clarity. It can be seen that in the cases of Figure 52b and Figure 52c there is not a significant difference in the performance of the composite adaptive controller compared to that of the computed-torque method.

The results for the computed-torque controller shown in Figure 52 differ from those of Chapter 7 because of the change from 3.27W to 6.5W motors. The reasons for this change are due to the order in which this research was carried out. Firstly, the computed torque algorithm was developed for both the 1-DOF and 2-DOF systems. The 2-DOF system required the use of the larger motors in order to overcome the friction forces generated at the worm/wheel interface. The composite adaptive controller for the 1-DOF and 2-DOF systems were introduced after the analysis of the 2-DOF computed-torque controller. However, the comparison of the two control strategies is direct because the computed-torque experiments were repeated with the larger actuators for the results of this chapter.

The parameter estimates corresponding to the results of the user-input given by Figure 50a, Figure 50b, and Figure 50c are given by Figure 53, Figure 54, and Figure 55 respectively. Unlike the simulation, the parameter errors converge on very different values to those calculated in equation (8-57) and shown by a dotted line in each figure. In the case of the low magnitude user-input (Figure 53) the parameters are slow to reach a constant estimate. However, for the two larger magnitude user-input commands, Figure 54 and Figure 55, the parameter errors are shown to reach steady-state more rapidly.

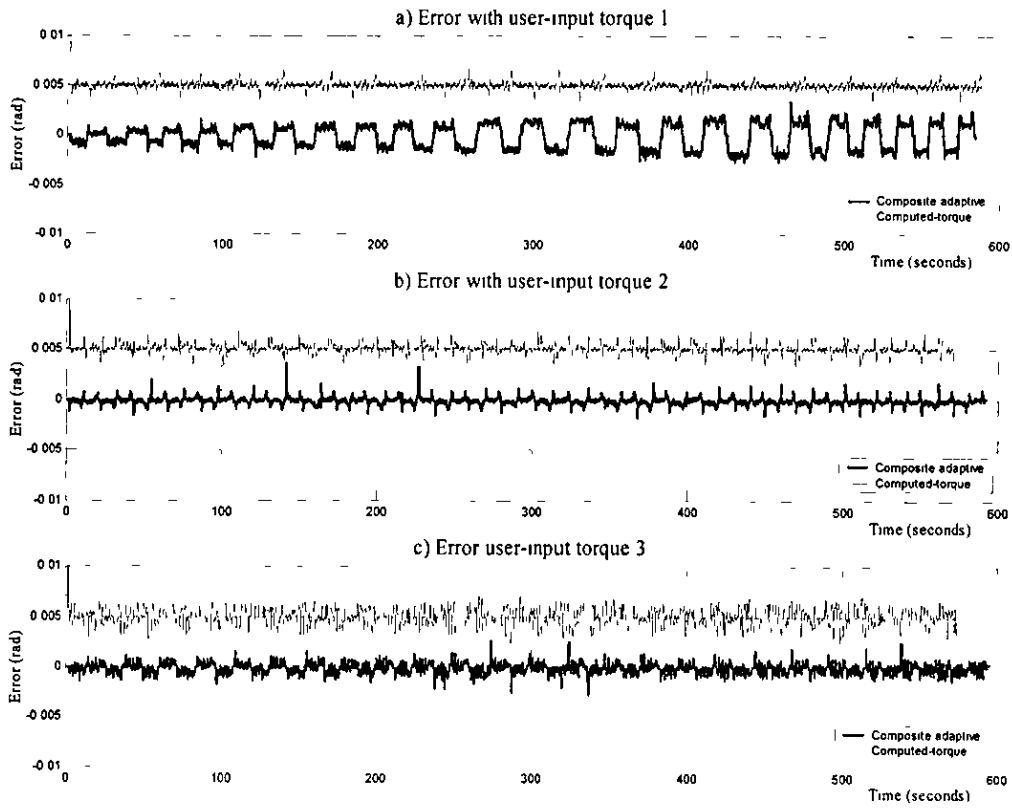


Figure 52 - Computed-torque control versus composite adaptive control

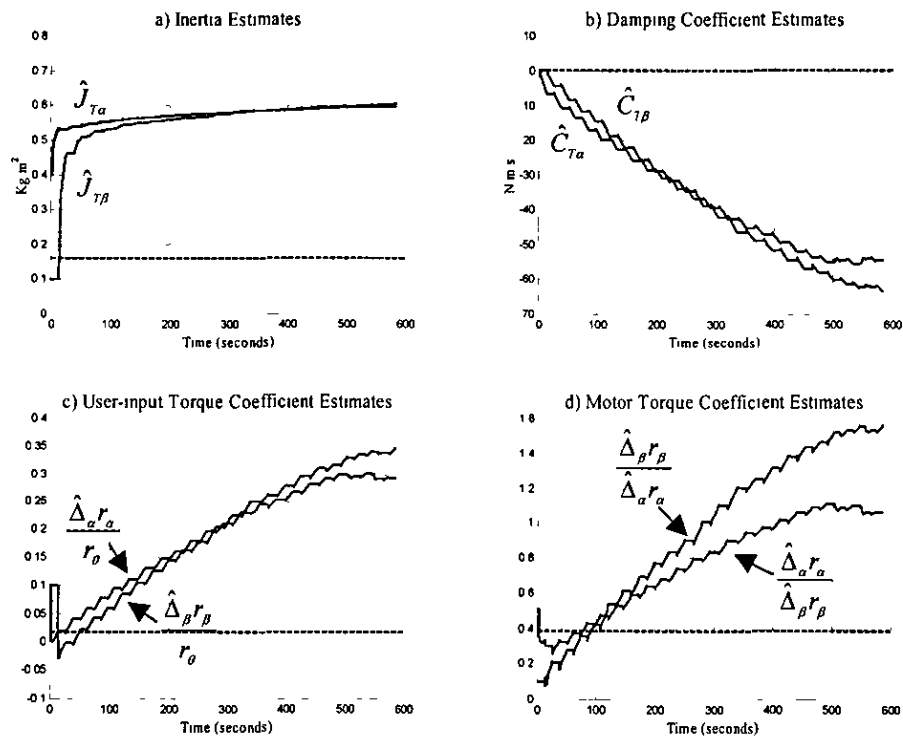


Figure 53 - Experimental parameter estimates with user-input 1

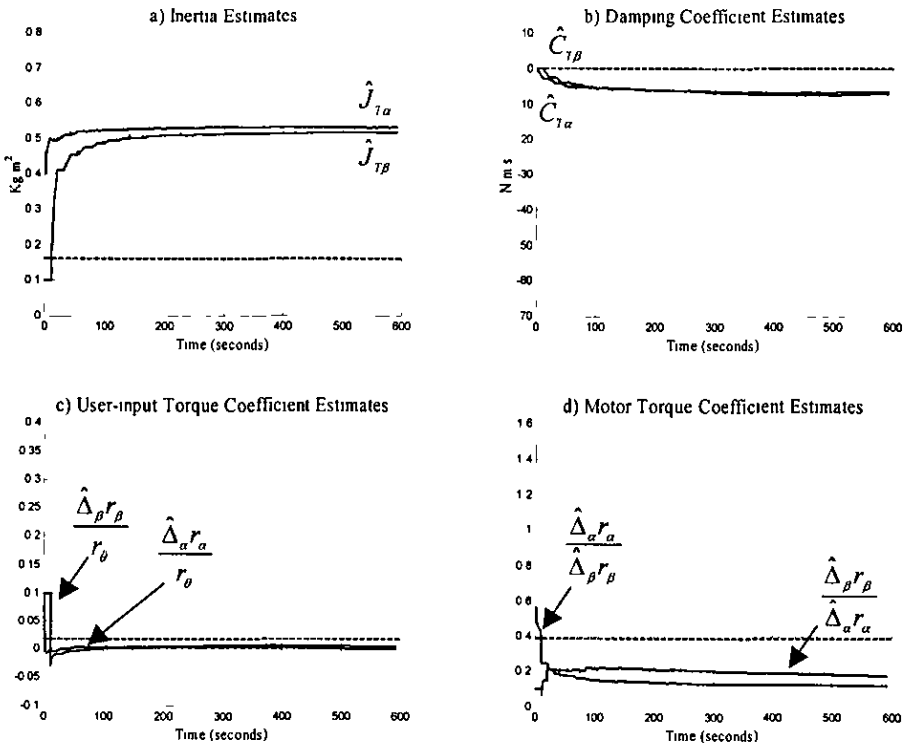


Figure 54 - Experimental parameter estimates with user-input 2

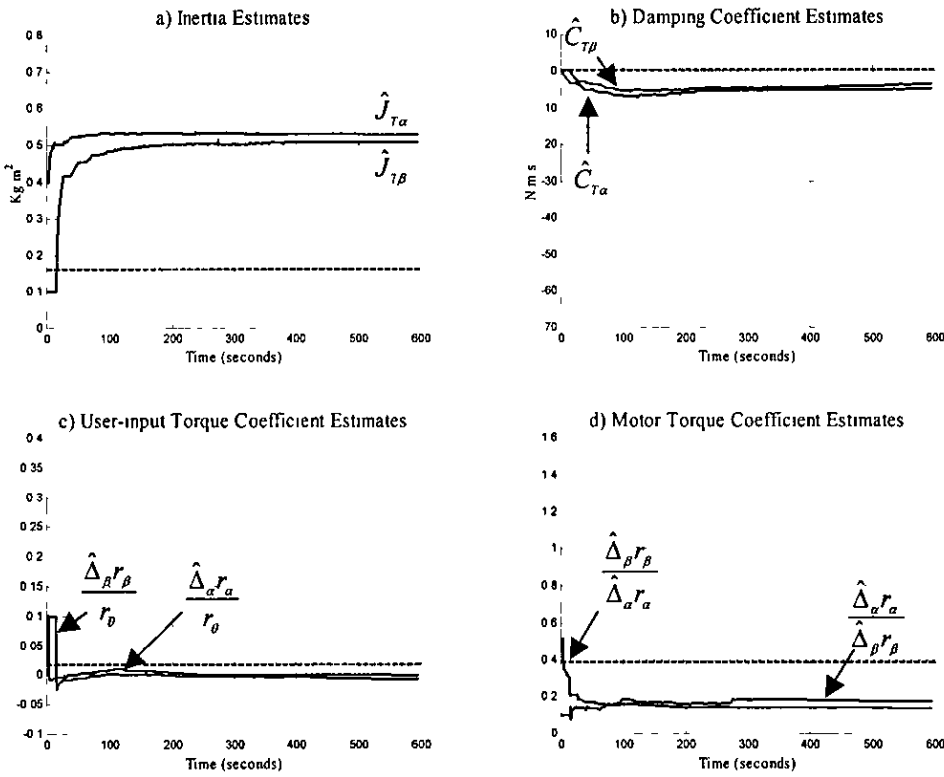


Figure 55 - Experimental parameter estimates with user-input 3

8.5 3 Discussion

There are a number of reasons for differences between simulation and experimental results. Firstly, the simulation assumes perfect torque control of both motors under both α and β -unwinding control schemes. However, practically the torque control method could not be used on the 'following' motor as this method generated 'jerky' motion of the link. To alleviate this problem the constant voltage technique, described in section 7.2, was used. This introduces a significant modelling error into the equations of motion for the system. Specifically, during i -unwinding the i' -motor command voltage is set to constant voltage, v_u . The terms i and i' should be replaced by α and β as in section 8.4.3. Hence the i' -motor torque can be expressed as in equation (8-59) using the model of an armature controlled DC motor and planetary gearbox as given in Appendix G. In equation (8-59), k_t , k_b , η_u , $n_{i'}$ and $R_{a'}$ represent the torque constant, the back EMF (electro-motive Force), planetary

gearbox efficiency, planetary gearbox ratio, and armature resistance of the i' -motor respectively. Armature inductance is assumed to be negligible.

$$\tau_m = \frac{\eta_i n_i k_u}{R_m} v_a - \frac{k_u k_b \eta_i n_i^2 r_o}{R_m r_a \tan \gamma} \theta \quad (8-59)$$

Using the system dynamics of equations (8-5) and (8-37) may in general be written as equation (8-60).

$$J_n \theta + C_n \theta = \tau_u \frac{\Delta_i r_i}{r_o} + \tau_m + \tau_{mi} \frac{\Delta_i r_i}{\Delta_i r_i} \quad (8-60)$$

Substitution of equation (8-59) into (8-60) yields

$$J_n \theta + \left(C_n + \frac{k_u k_b \eta_i n_i^2 r_o}{R_m r_a \tan \gamma} \frac{\Delta_i r_i}{\Delta_i r_i} \right) \theta = \tau_u \frac{\Delta_i r_i}{r_o} + \tau_m + \frac{\eta_i n_i k_u}{R_m} v_a \frac{\Delta_i r_i}{\Delta_i r_i} \quad (8-61)$$

Since the term $n_i^2 r_o / (r_a \tan \gamma)$ will be quite large due the speed ratio from the worm and wheel combination and from the planetary gearbox, equation (8-61) represents a significant modelling error. In fact it can be seen that the estimate of the damping coefficient is severely effected by the constant voltage technique as shown experimentally in section 8.5.2. There may also be errors in experimental determination of the system parameters given in Table 6 and errors in measurement of the user-input, motor torque and speed (caused by software filtering to remove noise) that could also affect the convergence of both tracking and parameter errors to zero.

The composite adaptive controller is based on the model of the system given by equation (8-1). The model does not include flexibility in the joint mechanism and assumes perfect measurement of motor torque and speed. The sum effect of these errors could be considered as an additive bounded torque disturbance, τ_d , i.e. equation (8-1) can be modified as shown in equation (8-62).

$$J_r \theta + C_r \theta + \tau_d = \tau_u + \frac{\tau_{ma} r_o}{\Delta_a r_a} + \frac{\tau_{mb} r_o}{\Delta_b r_b} \quad (8-62)$$

This disturbance also affects the parameter estimates as shown below in equation (8-63), i.e. equation (8-24) is replaced by equation (8-63) to take into account the unknown disturbance

$$\dot{\hat{\Phi}}_i = \mathbf{P}_i \mathbf{Y}_i^T r + \mathbf{P}_i \mathbf{W}_\beta^T (\tau_\beta - \mathbf{W}_\beta \hat{\Phi}_i + f * \tau_d) \quad (8-63)$$

where $\tau_d = \frac{\Delta_i r_i}{r_\beta} \tau_d$ and f represents the linear stable, strictly proper filter. More specifically, this disturbance manifests itself in the time derivative of the Lyapunov function as

$$\dot{V}_i = -K_v r^2 - \frac{1}{2} \tilde{\Phi}_i^T \mathbf{W}_\beta^T \mathbf{W}_\beta \tilde{\Phi}_i + r \tau_d + \tilde{\Phi}_i^T \mathbf{W}_\beta^T (f * \tau_d) \quad (8-64)$$

It can be seen that the introduction of this disturbance into the model effects the stability result defined above, i.e. the negative semi-definiteness of \dot{V} is no longer assured. It can be clearly seen that the tracking and parameter error may not be zero with the errors caused by un-modelled dynamics and incorrect measurements. Although the parameter error may not converge to zero, the boundedness of $\tilde{\Phi}_i$ can still be proven since the result that $\lim_{t \rightarrow \infty} \{\mathbf{P}\} = 0$ is unaffected by the introduction of the torque disturbance noting also that $\tilde{\Phi}_i$ will also eventually become constant.

Further insight into the behaviour of the system can be gained by analysing equation (8-64). If the condition given in equation (8-65) holds, then \dot{V}_i is negative and the Lyapunov function, V_i , will decrease. Using equation (8-42) and by momentarily fixing τ_d and \mathbf{W}_β , it can be determined that both r and $\tilde{\Phi}_i$ must decrease

$$K_v r^2 + \frac{1}{2} \tilde{\Phi}_i^T \mathbf{W}_\beta^T \mathbf{W}_\beta \tilde{\Phi}_i > r \tau_d + \tilde{\Phi}_i^T \mathbf{W}_\beta^T (f * \tau_d) \quad (8-65)$$

However, eventually r and $\tilde{\Phi}_i$ will decrease so that the following condition, in equation (8-66), holds true.

$$K_v r^2 + \frac{1}{2} \tilde{\Phi}_i^T \mathbf{W}_\beta^T \mathbf{W}_\beta \tilde{\Phi}_i \leq r \tau_d + \tilde{\Phi}_i^T \mathbf{W}_\beta^T (f * \tau_d) \quad (8-66)$$

Therefore if equation (8-66) holds true, \dot{V}_i becomes positive and V_i increases which will eventually lead to an increase in r and $\tilde{\Phi}_i$. This leads to the conclusion that as

long as τ_{di} and W_{fi} remain bounded, both r and $\tilde{\Phi}_i$ are also bounded by some function of r , $\tilde{\Phi}_i$, K_v , τ_{di} and W_{fi}

The robustness of the composite adaptive controller can be highlighted in simulation by adjusting the model to include the constant voltage control of the 'following' worm given by equation (8-61) and by introducing some measurement error of τ_u , τ_{na} , and τ_{mg} . In the following simulation tests on the system the two user-input torque commands applied to the system are given by equations (8-54) and (8-56). The simulation uses the same parameters defined in section 8.5.1. Figure 56 shows the parameter estimates when the less exciting user-input of equation (8-54) is used. Similarly, Figure 57 illustrates the parameter estimates when the more exciting input of equation (8-56) is applied to the system. As shown in section 8.5.1, the speed of convergence is affected by the properties of the user-input command. As anticipated in both cases the estimates do not converge on the actual values calculated in equation (8-57) and shown by dotted lines in both Figure 56 and Figure 57. Finally, the difference in tracking performance under the two separate conditions is illustrated in Figure 58. The less exciting trajectory shows considerably larger errors. However, in both cases the tracking error and parameter error are both shown to be bounded under these conditions illustrating that the composite adaptive controller is robust against disturbances caused by both un-modelled dynamics and measurement error.

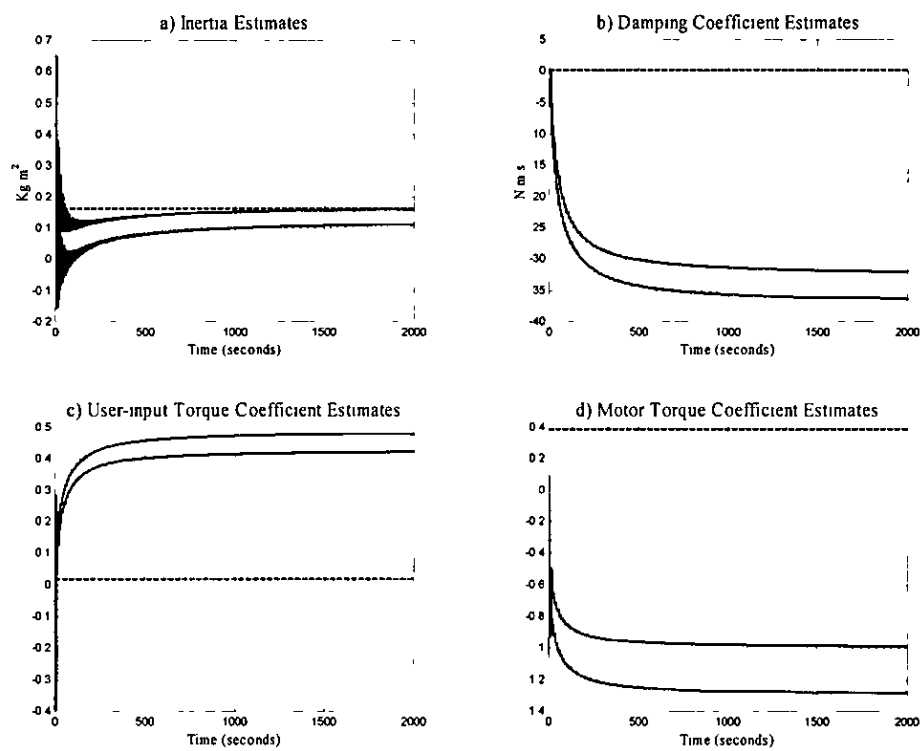


Figure 56 - Parameter estimates with less exciting trajectory

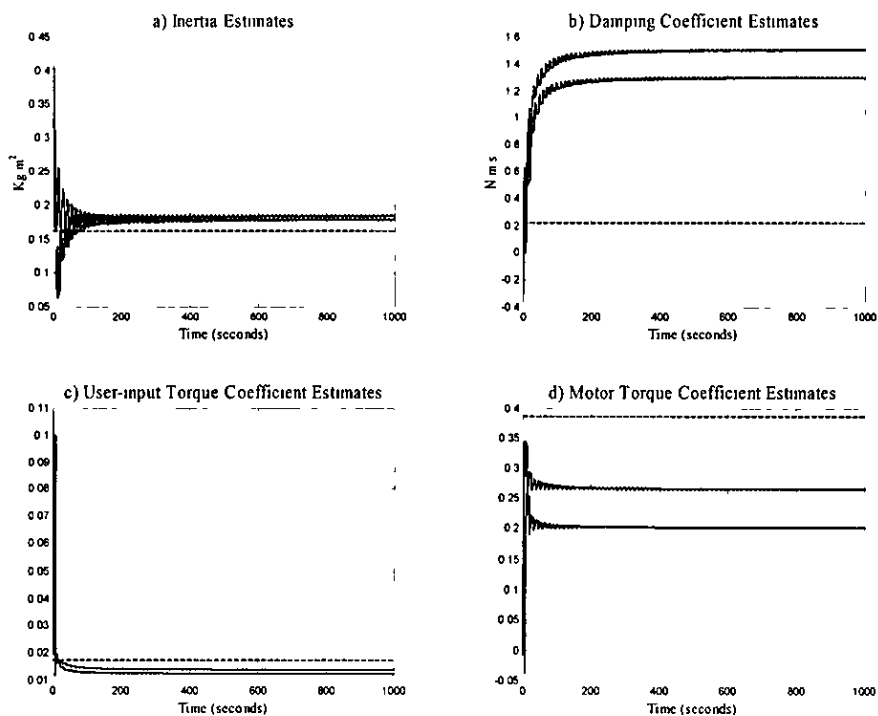


Figure 57 - Parameter estimates with more exciting trajectory

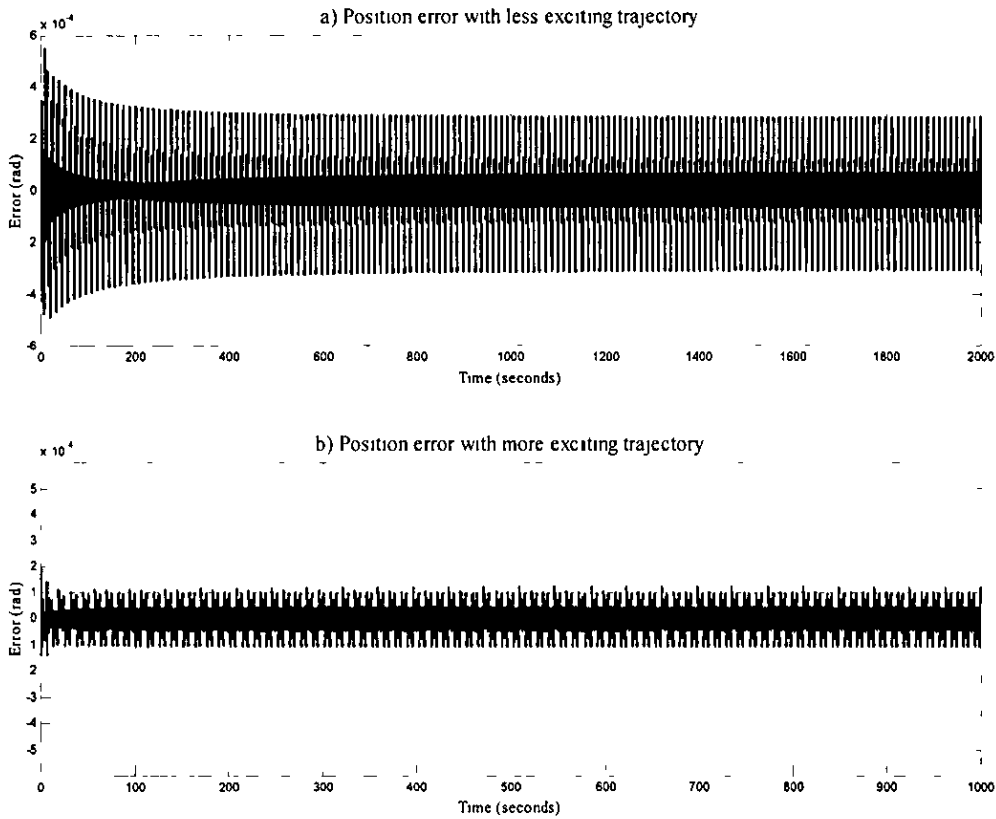


Figure 58 - Comparison of tracking error with erroneous model

8.6. Conclusions

Both simulation and experimental results have shown that a composite adaptive control scheme can be used to track a desired path. It has been shown that the dependence of the desired velocity on the user-input force and error in the model has a detrimental effect on the convergence of the tracking and parameter estimates to zero. The composite adaptive control method has been shown to give performance equivalent to the computed-torque algorithm, presented in Chapter 7, under certain conditions on the user-input torque command. However, it has been shown by experimental that if the desired velocity and user-input are not sufficiently independent of each other, the composite adaptive controller does not perform as well as the computed-torque algorithm. In both the simulation and experimental results, the tracking and parameter errors remain bounded under these conditions.

Unlike the composite adaptive controller, the ability of the computed-torque control algorithm to track a path relies on fixed estimates of the robot parameters. Therefore degradation of the computed-torque controller's performance due to wear of the gears or changes in load can be expected. This would require periodic re-tuning of the algorithm in order to ensure satisfactory performance, which can be a time-consuming process. The composite adaptive control scheme has been shown to be robust in the face of large modelling and measurement errors and as such can be used to compensate for changes in system parameters. Due to the nature of the proposed application, it may not be prudent to allow the controller to adjust parameters during operation. The composite adaptive controller should be used to train the manipulator off-line as part of a calibration procedure before it is used. Care must be taken to ensure that the user-input command consistently saturates the desired velocity function during this training period.

Chapter 9 Mathematical Model of the Two Degree-of-Freedom Manipulator

9.1. Introduction

In order to develop suitable algorithms for trajectory tracking control of the 2-DOF worm-driven manipulator it is necessary to derive a mathematical model of the system. This chapter details the formulation of the model as well as the derivation of the inverse kinematics

9.2. Denavit-Hartenberg Representation

Assigning the co-ordinate frames using the Denavit-Hartenberg (D-H) method as shown in Figure 59, it is possible to get the D-H representation given in Table 9.

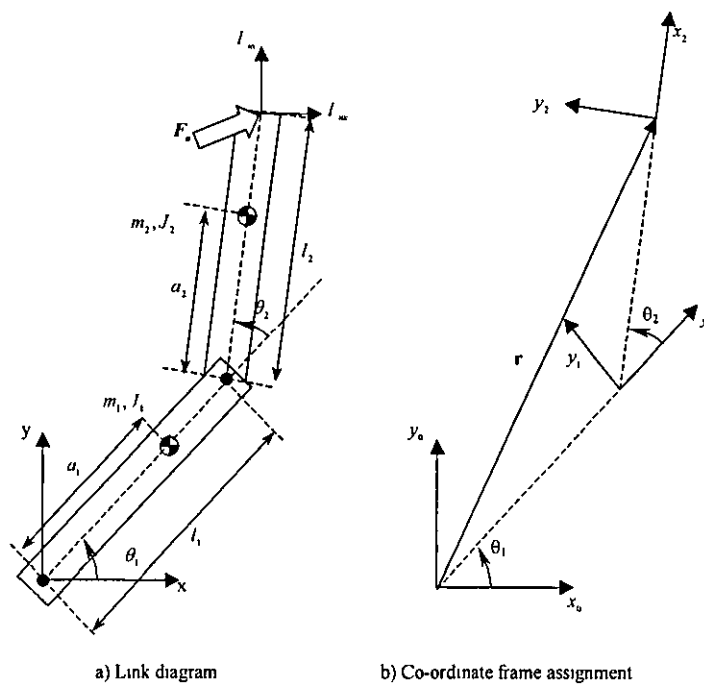


Figure 59 – 2-DOF Manipulator

Table 9 - Denavit-Hartenberg representation

| Axis | θ | d | a | α | Home |
|------|------------|-----|-------|----------|------|
| 1 | θ_1 | 0 | l_1 | 0 | 0 |
| 2 | θ_2 | 0 | l_2 | 0 | 0 |

In the following equations the notation given in equation (9-1) is used.

$$\begin{aligned} s_n &= \sin \theta_n & s_{nm} &= \sin(\theta_n + \theta_m) \\ c_n &= \cos \theta_n & c_{nm} &= \cos(\theta_n + \theta_m) \end{aligned} \quad (9-1)$$

Using Table 9 it is possible to obtain the link co-ordinate transformations given in equations (9-2) and (9-3)

$$T_0^1 = \begin{bmatrix} \mathbf{R}_0^1 & \mathbf{P}_0^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1 & -s_1 & 0 & l_1 c_1 \\ s_1 & c_1 & 0 & l_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9-2)$$

where \mathbf{R}_m^n represents the rotational transformation matrix from frame $\{m\}$ to frame $\{n\}$ and \mathbf{P}_m^n represents the position of frame $\{m\}$ with respect to frame $\{n\}$.

$$T_1^2 = \begin{bmatrix} \mathbf{R}_1^2 & \mathbf{P}_1^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 & -s_2 & 0 & l_2 c_2 \\ s_2 & c_2 & 0 & l_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9-3)$$

It is therefore possible to define the transformation from the base frame (frame 0) to frame 2 as given by equation (9-4)

$$T_0^2 = T_0^1 T_1^2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9-4)$$

9.3. Link Jacobian

Using the Langrage-Euler method [107], the translation from the base frame (frame 0) to the centre of mass of link 1 is defined as

$$\begin{aligned}\bar{c}' &= H_1 T_0^1 \begin{bmatrix} -l_1 + a_1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ \bar{c}' &= \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix}\end{aligned}\quad (9-5)$$

where H_1 is the homogenous transformation given by equation (9-6). It is used to convert a (4x1) vector to a (3x1) vector, i.e. the position of the centre of mass of a link with respect to frame $\{0\}$.

$$H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\quad (9-6)$$

Using equation (9-7), where z' is the third unit vector of R_0^1 , the next step is to compute the Jacobian, J^1 , given in equation (9-8)

$$J^k = \begin{bmatrix} \frac{\partial \bar{c}^k}{\partial \theta_1} & \frac{\partial \bar{c}^k}{\partial \theta_k} \\ \vdots & \vdots \\ \xi_0 z^0 & \xi_k z^{k-1} \end{bmatrix} = \begin{bmatrix} A^k(q) \\ \vdots \\ B^k(q) \end{bmatrix} \quad q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}\quad (9-7)$$

$$J^1(q) = \begin{bmatrix} -a_1 s_1 & 0 \\ a_1 c_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} A^1(q) \\ \vdots \\ B^1(q) \end{bmatrix}\quad (9-8)$$

Similarly, for frame 2, the translation from the base frame to the mass centre of the link is given by equation (9-9).

$$\begin{aligned}\bar{\mathbf{c}}^2 &= \mathbf{H}_1 \mathbf{T}_0^2 \begin{bmatrix} -l_2 + a_2 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ \bar{\mathbf{c}}^2 &= \begin{bmatrix} l_1 c_1 + a_2 c_{12} \\ l_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix}\end{aligned}\quad (9-9)$$

Using equation (9-7), the link Jacobian \mathbf{J}^2 can be calculated to give equation (9-10).

$$\mathbf{J}^2 = \begin{bmatrix} -l_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ l_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}^2(\mathbf{q}) \\ \mathbf{B}^2(\mathbf{q}) \end{bmatrix}\quad (9-10)$$

The inertia tensor of link1 in base co-ordinates is

$$\mathbf{D}_1 = \mathbf{R}_0^1 \bar{\mathbf{D}}_1 (\mathbf{R}_0^1)^T \quad (9-11)$$

where $\bar{\mathbf{D}}_1$ is the inertia tensor at the mass centre of link 1. The inertia tensor of link 2 with respect to the base frame is given by equation (9-12)

$$\mathbf{D}_2 = \mathbf{R}_0^2 \bar{\mathbf{D}}_2 (\mathbf{R}_0^2)^T \quad (9-12)$$

The manipulator inertia matrix, \mathbf{M} , is given by equation (9-15) and (9-16).

$$\mathbf{M} = \sum_{i=1}^2 \left((\mathbf{A}^i)^T m_i \mathbf{A}^i + (\mathbf{B}^i)^T \mathbf{D}_i \mathbf{B}^i \right) \quad (9-13)$$

When $i=1$ the inertia for the first link is given by equation (9-14)

$$(\mathbf{A}^1)^T m_1 \mathbf{A}^1 + (\mathbf{B}^1)^T \mathbf{D}_1 \mathbf{B}^1 = \begin{bmatrix} a_1^2 m_1 + J_{o1} & 0 \\ 0 & 0 \end{bmatrix} \quad (9-14)$$

Similarly for the second link, i.e. when $i = 2$, the inertia tensor is given by equation (9-15)

$$\begin{aligned} (\mathbf{A}^2)^T m_2 \mathbf{A}^2 + (\mathbf{B}^2)^T \mathbf{D}_2 \mathbf{B}^2 = \\ \begin{bmatrix} (l_1^2 + a_2^2 + 2l_1 a_2 c_2) m_2 + J_{\theta 2} & (a_2^2 + l_1 a_2 c_2) m_2 + J_{\theta 2} \\ (a_2^2 + l_1 a_2 c_2) m_2 + J_{\theta 2} & a_2^2 m_2 + J_{\theta 2} \end{bmatrix} \end{aligned} \quad (9-15)$$

Therefore the manipulator inertia matrix for the 2-DOF planar manipulator is

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} a_1^2 m_1 + J_1 + (l_1^2 + a_2^2 + 2l_1 a_2 c_2) m_2 + J_2 & (a_2^2 + l_1 a_2 c_2) m_2 + J_2 \\ (a_2^2 + l_1 a_2 c_2) m_2 + J_2 & a_2^2 m_2 + J_2 \end{bmatrix} \\ &= \begin{bmatrix} \alpha_0 + 2\alpha_1 c_2 & \alpha_2 + \alpha_1 c_2 \\ \alpha_2 + \alpha_1 c_2 & \alpha_2 \end{bmatrix} \\ \alpha_0 &= a_1^2 m_1 + (l_1^2 + a_2^2) m_2 + J_1 + J_2 \\ \alpha_1 &= m_2 l_1 a_2 \\ \alpha_2 &= a_2^2 m_2 + J_2 \end{aligned} \quad (9-16)$$

9.4. Lagrange-Euler Formulation of the Manipulator Dynamics

It is possible to write the kinetic co-energy as

$$\begin{aligned} T^* &= \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^2 M_{ij} \theta_i \theta_j \\ T^* &= \frac{1}{2} (M_{11} \theta_1^2 + M_{12} \theta_1 \theta_2 + M_{21} \theta_2 \theta_1 + M_{22} \theta_2^2) \end{aligned} \quad (9-17)$$

where M_{ij} is the ij element of \mathbf{M}

As the manipulator does not work against gravity and that joint/link flexibility has not been modelled the potential energy, V , is zero. However, there are forces from the user (F_u), viscous friction (b), and worm drives (τ) acting on the manipulator. The virtual work done by these forces is defined by

$$\begin{aligned} \partial W &= \mathbf{F}^T \partial \mathbf{q} \\ \partial W &= \boldsymbol{\tau}^T \partial \mathbf{q} - \mathbf{b}^T \partial \mathbf{q} + \mathbf{F}_u^T \frac{\partial \mathbf{X}}{\partial \mathbf{q}} \partial \mathbf{q} \end{aligned} \quad (9-18)$$

Noting that

$$\begin{aligned} \mathbf{F}_u &= \begin{bmatrix} F_{ux} \\ F_{uy} \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix}, \\ \frac{\partial \mathbf{X}}{\partial \mathbf{q}} &= \begin{bmatrix} \frac{\partial \mathbf{X}}{\partial \theta_1} & \frac{\partial \mathbf{X}}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \end{aligned} \quad (9-19)$$

the generalised forces are described by

$$\mathbf{F} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - \begin{bmatrix} C_{f\theta 1} \theta_1 \\ C_{f\theta 2} \theta_2 \end{bmatrix} + \begin{bmatrix} -F_{ux}(l_1 s_1 + l_2 s_{12}) + F_{uy}(l_1 c_1 + l_2 c_{12}) \\ -F_{ux} l_2 s_{12} + F_{uy} l_2 c_{12} \end{bmatrix} \quad (9-20)$$

where τ_1 and τ_2 are the torques applied with the worm driven gearboxes. τ_1 and τ_2 are given by equations (9-21) and (9-22), respectively

$$\tau_1 = \frac{\tau_{m\alpha 1} r_{\theta 1}}{r_{\alpha 1} \Delta_{\alpha 1}} + \frac{\tau_{m\beta 1} r_{\theta 1}}{r_{\beta 1} \Delta_{\beta 1}} - J_{\alpha 1} \frac{\theta_1 r_{\theta 1}^2}{\Delta_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} - J_{\beta 1} \frac{\theta_1 r_{\theta 1}^2}{\Delta_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} - C_{f\alpha 1} \frac{\theta_1 r_{\theta 1}^2}{\Delta_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} - C_{f\beta 1} \frac{\theta_1 r_{\theta 1}^2}{\Delta_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} \quad (9-21)$$

$$\tau_2 = \frac{\tau_{m\alpha 2} r_{\theta 1}}{r_{\alpha 1} \Delta_{\alpha 2}} + \frac{\tau_{m\beta 2} r_{\theta 1}}{r_{\beta 1} \Delta_{\beta 2}} - J_{\alpha 2} \frac{\theta_2 r_{\theta 1}^2}{\Delta_{\alpha 2} r_{\alpha 1}^2 \tan \gamma_2} - J_{\beta 2} \frac{\theta_2 r_{\theta 1}^2}{\Delta_{\beta 2} r_{\beta 1}^2 \tan \gamma_2} - C_{f\alpha 2} \frac{\theta_2 r_{\theta 1}^2}{\Delta_{\alpha 2} r_{\alpha 1}^2 \tan \gamma_2} - C_{f\beta 2} \frac{\theta_2 r_{\theta 1}^2}{\Delta_{\beta 2} r_{\beta 1}^2 \tan \gamma_2} \quad (9-22)$$

Using Lagrange's equation the equations of motion for this 2-DOF manipulator may be derived. The Lagrangian, L , is defined as

$$L = T^* - V \quad (9-23)$$

and the equations of motion can be expressed as

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{F}. \quad (9-24)$$

Substitution of equations (9-17) and (9-20) into equation (9-24) and noting that $D'(q)$ is symmetric yields

$$(\mathbf{M} + \mathbf{M}_r)\mathbf{q} + (\mathbf{C} + \mathbf{C}_r)\mathbf{q} + \mathbf{V}_m\mathbf{q} = \mathbf{B}\boldsymbol{\tau}_m + \mathbf{J}^T \mathbf{F}_e \quad (9-25)$$

where

$$\mathbf{M}_r = \begin{bmatrix} \frac{J_{\alpha 1} r_{\theta 1}^2}{\Delta_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} + \frac{J_{\beta 1} r_{\theta 1}^2}{\Delta_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} & 0 \\ 0 & \frac{J_{\alpha 2} r_{\theta 2}^2}{\Delta_{\alpha 2} r_{\alpha 2}^2 \tan \gamma_2} + \frac{J_{\beta 2} r_{\theta 2}^2}{\Delta_{\beta 2} r_{\beta 2}^2 \tan \gamma_2} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} C_{f\theta 1} & 0 \\ 0 & C_{f\theta 2} \end{bmatrix} \quad \mathbf{C}_r = \begin{bmatrix} \frac{C_{f\alpha 1} r_{\theta 1}^2}{\Delta_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} + \frac{C_{f\beta 1} r_{\theta 1}^2}{\Delta_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} & 0 \\ 0 & \frac{C_{f\alpha 2} r_{\theta 2}^2}{\Delta_{\alpha 2} r_{\alpha 2}^2 \tan \gamma_2} + \frac{C_{f\beta 2} r_{\theta 2}^2}{\Delta_{\beta 2} r_{\beta 2}^2 \tan \gamma_2} \end{bmatrix}$$

$$\mathbf{V}_m \mathbf{q} = \begin{bmatrix} -m_2 l_1 a_2 s_2 (2\theta_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\ m_1 l_1 a_2 s_2 \dot{\theta}_1^2 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{r_{\theta 1}}{\Delta_{\alpha 1} r_{\alpha 1}} & \frac{r_{\theta 1}}{\Delta_{\beta 1} r_{\beta 1}} & 0 & 0 \\ 0 & 0 & \frac{r_{\theta 2}}{\Delta_{\alpha 2} r_{\alpha 2}} & \frac{r_{\theta 2}}{\Delta_{\beta 2} r_{\beta 2}} \end{bmatrix} \quad \boldsymbol{\tau}_m = [\tau_{m\alpha 1} \quad \tau_{m\beta 1} \quad \tau_{m\alpha 2} \quad \tau_{m\beta 2}]^T$$

The model of the 2-DOF manipulator given by equation (9-25) will be used to develop computed-torque and composite adaptive control algorithms in later chapters. The remainder of this chapter is dedicated to the inverse kinematics of the 2-DOF system and an explanation of the path representation scheme used in this work

9.5. Inverse Kinematics

In general, desired paths and trajectories for robotic devices are specified in world-coordinates, (x_d, y_d, z_d) . Therefore it is necessary to calculate the desired joint angles that would achieve the specified position. For this 2-DOF manipulator only the position of the end-effector is important, i.e. orientation of the end-effector is not

specified. Using Figure 60, the desired cartesian position of the end-effector, x_d and y_d , may be written as equations (9-26) and (9-27) respectively.

$$x_d = l_1 c_1 + l_2 c_{12} \quad (9-26)$$

$$y_d = l_1 s_1 + l_2 s_{12} \quad (9-27)$$

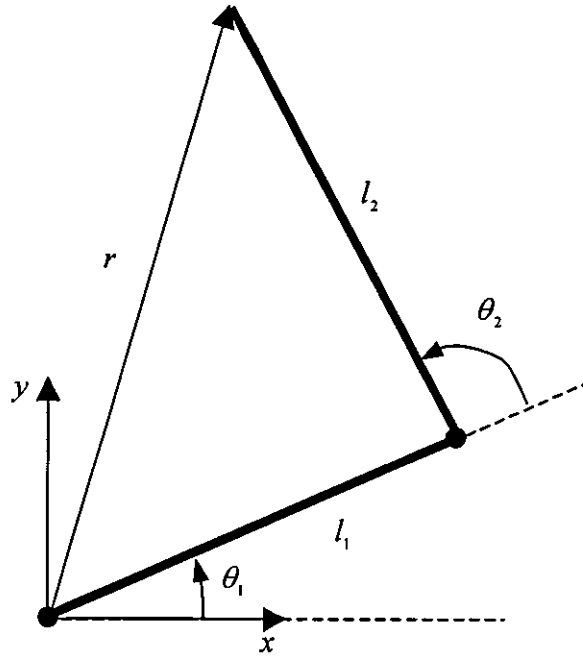


Figure 60 - Robot geometry

From Figure 60, it is possible to write

$$\begin{aligned} r^2 = x_d^2 + y_d^2 &= (l_1 c_1 + l_2 c_{12})^2 + (l_1 s_1 + l_2 s_{12})^2 \\ &= l_1^2 + 2l_1 l_2 c_2 + l_2^2 \end{aligned} \quad (9-28)$$

Therefore it is possible to explicitly solve for the desired joint angle θ_{2d} to give

$$\cos(\theta_{2d}) = \frac{x_d^2 + y_d^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (9-29)$$

and

$$\sin(\theta_{2d}) = \pm \sqrt{1 - \cos^2(\theta_{2d})} \quad (9-30)$$

θ_{2d} can then be chosen by selecting the sign in equation (9-30) and substituting the values from equations (9-29) and (9-30) into equation (9-31).

$$\theta_{2d} = \text{atan2} \left(\frac{x_d^2 + y_d^2 - l_1^2 - l_2^2}{2l_1l_2}, \pm \sqrt{1 - \frac{x_d^2 + y_d^2 - l_1^2 - l_2^2}{2l_1l_2}} \right) \quad (9-31)$$

$\sin(\theta_{2d})$ in (9-29) can be chosen to be either positive or negative to give a right or left handed robot configuration as desired. Expanding the expressions for x_d and y_d , given by equations (9-26) and (9-27) respectively, using the sine of the sum and the cosine of the sum trigonometric identities yields equations (9-32) and (9-33).

$$x_d = (l_1 + l_2 c_2) c_1 - l_2 s_2 s_1 \quad (9-32)$$

$$y_d = l_2 s_2 c_1 + (l_1 + l_2 c_2) s_1 \quad (9-33)$$

Simultaneous solution of (9-32) and (9-33) yields

$$c_1 = \frac{(l_1 + l_2 c_2)x_d + l_2 s_2 y_d}{(l_1 + l_2 c_2)^2 + (l_2 s_2)^2} \quad (9-34)$$

$$s_1 = \frac{(l_1 + l_2 c_2)y_d - l_2 s_2 x_d}{(l_1 + l_2 c_2)^2 + (l_2 s_2)^2} \quad (9-35)$$

Division of equation (9-35) by equation (9-34) yields the desired angle of the first joint

$$\theta_{1d} = \text{atan2}((l_1 + l_2 c_2)y_d - l_2 s_2 x_d, (l_1 + l_2 c_2)x_d + l_2 s_2 y_d) \quad (9-36)$$

9.6. Path Representation

In most robotic application a path is described in terms of a Cartesian co-ordinate system. A desired path $P(x, y)$ can be separated into a series of shorter continuous paths $p_1(x, y), \dots, p_n(x, y)$ by defining via points as shown in Figure 61

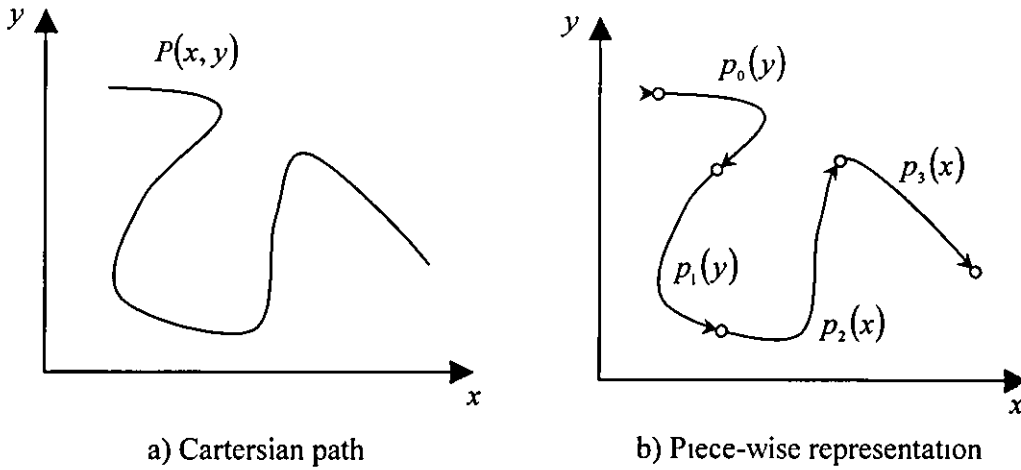


Figure 61 - A piece-wise representation of a Cartesian path

One common representation of these paths is in the form of piece-wise cubic polynomials [107]. A particularly attractive feature of the cubic polynomial representation is that the end-effector will (in theory) pass directly through the desired via points. However, other representations, such as β -splines do not force the end-effector to pass directly through the via-point.

In general, the polynomial functions usually describe the position of each Cartesian variable with respect to time. This allows the velocity and acceleration along the path to be specified. For this application only the path itself is pre-defined, the user-input force will be used to specify the velocity along the path and the method is described in more detail in **Chapter 10**. Therefore a series of cubic polynomials is used to describe one Cartesian variable with respect to the other and is represented by equation (9-37) or (9-38). The α_i 's and β_i 's are found from the known boundary conditions of each of equations (9-37) or (9-38), i.e. from the position and gradient at the start and end of the path. Continuity of the path is assured by making the gradient at the start of the path equal to the gradient at the end of the previous path. Care should also be taken to ensure that the path is split into small enough segments so that there is only one solution to equation (9-7) or (9-8) and that the coefficients remain finite.

Chapter 10 Computed-Torque Control of a Two Degree-of-Freedom Worm-Driven Manipulator

10.1. Introduction

This chapter represents the design of a computed torque controller for the 2-DOF manipulator. By measuring joint torque the computed-torque control strategy ensures safety of the user by never driving against the input force. The control algorithm is shown to force the end-effector to track a desired path with a velocity determined by the magnitude of the user-input force. Experimental results show that by directing the user to apply a force in the correct direction the end-effector can be forced to track the path without risk to the operator.

10.2. Control Strategy

In order to ensure that the robot does not exhibit motion against the user, the robot is allowed to move only if the torque acting on each joint is in the same direction as the desired motion of that joint. These conditions can be expressed mathematically as given by equation (10-1) and they must both be satisfied in order for the robot to be able to move as shown in Chapter 3. To implement this strategy successfully strain gauges have been mounted on each link in order to measure the link torques, τ_{act1} and

τ_{act2}

$$\begin{aligned}\text{sgn}(\dot{\theta}_{1d}) &= \text{sgn}(\tau_{act1}) \\ \text{sgn}(\dot{\theta}_{2d}) &= \text{sgn}(\tau_{act2})\end{aligned}\tag{10-1}$$

The user controls the motion of the end-effector along the path by grasping the control handle and applying a force. Only if the conditions of equation (10-1) are satisfied are the robot joints allowed to move along the path. The desired (Cartesian) velocity of the motion, V_d , of the end-effector is in a direction determined by the path and at a magnitude generated by the user input-force and the relationship given in equation (10-2). $V_{d\max}$, $F_{u\min}$, and $F_{u\max}$ in equation (10-2) are all positive constants. This

velocity-limiting algorithm is also shown graphically in Figure 43 and the relationship ensures that velocity and acceleration demands for the end-effector are continuous

$$|V_d| = \begin{cases} 0 & \forall |F_u| < F_{u \min} \\ V_{d \max} \left(-\frac{2(|F_u| - F_{u \min})^3}{(F_{u \max} - F_{u \min})^3} + \frac{3(|F_u| - F_{u \min})^2}{(F_{u \max} - F_{u \min})^2} \right) & \forall F_{u \min} \leq |F_u| \leq F_{u \max} \\ V_{d \max} & \forall |F_u| > F_{u \max} \end{cases} \quad (10-2)$$

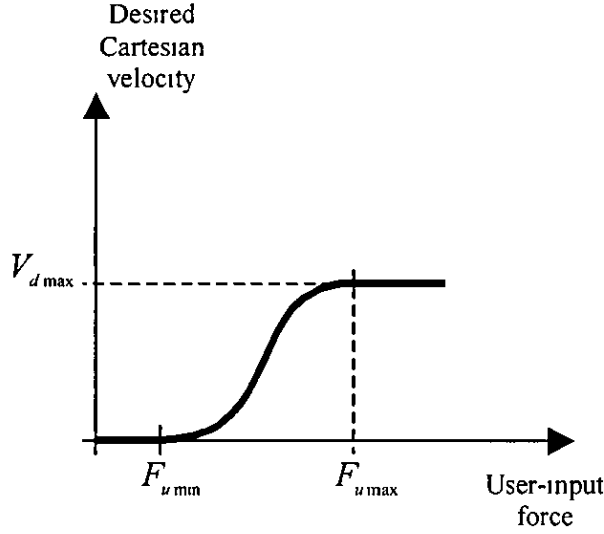


Figure 63 - User defined velocity demand

In order to cancel backlash in the worm mechanism the α and β -unwinding control strategy is used as in Chapter 7. For motion of a joint in the clockwise direction (i.e. $\theta_a > 0$) the motor command voltage for the α -worm is set to a constant value and the β -motor torque, $\tau_{m\beta}$, is controlled to unwind the β -worm. In this condition the β -worm *leads* and the α -worm is used to *follow* the motion of the worm wheel without applying unnecessary frictional forces to the system. Control of motion in this manner is termed β -unwinding control. However, in the anti-clockwise direction (i.e. $\theta_a < 0$) the β -worm motor command voltage is set to a constant value and $\tau_{m\alpha}$ is used to control the unwinding of the α -worm. This is termed α -unwinding control. In this instance, the α -worm *leads* and the β -worm *follows*.

10.3. Control of the pointer mechanism

A control handle was presented in Chapter 4 that incorporates a pointer mechanism for indicating to the user the direction in which a force should be applied to accomplish the desired task. The method for deciding the direction in which the pointer should be aimed is described below.

Firstly, the assumption is made that the desired joint acceleration remains small, i.e. the force, F , described in Chapter 3, tangential to the desired path is small. Under this assumption it is possible to map the pointer direction to the desired path direction by considering the diagram shown in Figure 64. The inner circle of Figure 64 represents the direction of the path divided into the four possible combinations of joint motion as described by Table 1 in Chapter 3. Similarly, the outer circle represents the four combinations of joint torque required to resist a user-input force described by Table 2 also in Chapter 3.

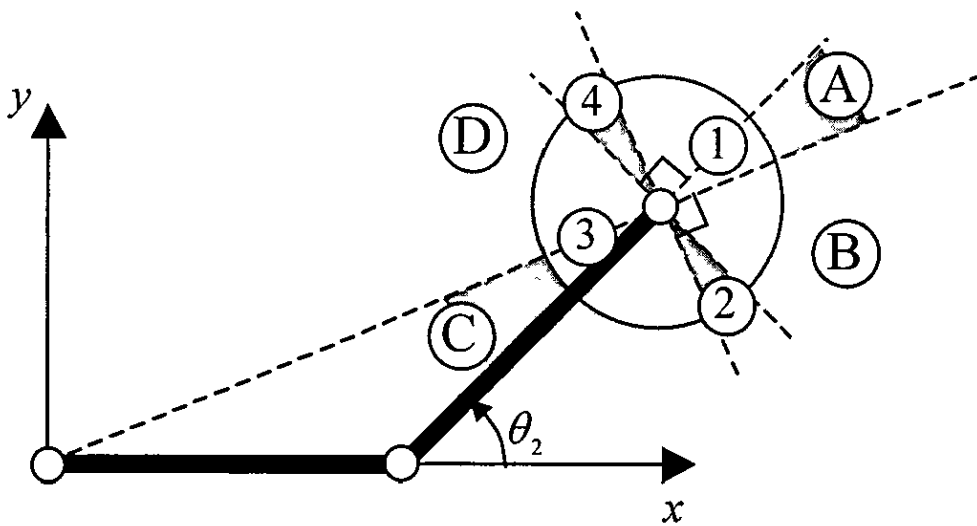


Figure 64 – Regions of user-input force and path direction

Using equation (10-1), Table 1, and Table 2 the conditions under which motion can be allowed are

1. If the path exists in region 1, the user-input force **must** be directed into region A
2. If the path exists in region 2, the user-input force **must** be directed into region B.
3. If the path exists in region 3, the user-input force **must** be directed into region C.
4. If the path exists in region 4, the user-input force **must** be directed into region D.

Using these conditions the pointer direction is linearly mapped into the desired regions A, B, C, or D if the path direction is in the regions 1, 2, 3, or 4 respectively as shown Figure 65. During this mapping no account is taken for the magnitude of the acceleration force, F_t , from equation (3-3)

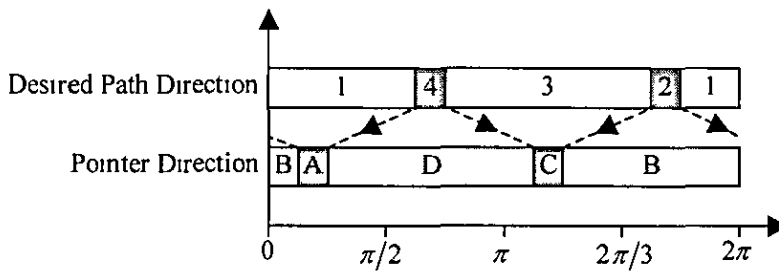


Figure 65 - Mapping pointer direction to desired path

10.4. Mathematical Model

The mathematical model of the 2-DOF worm-driven manipulator is given by equation (9-25) and is repeated here for clarity.

$$\mathbf{M}\mathbf{q} + \mathbf{M}_r\mathbf{q} + \mathbf{C}\mathbf{q} + \mathbf{C}_r\mathbf{q} + \mathbf{V}_m\mathbf{q} = \mathbf{B}\boldsymbol{\tau}_m + \mathbf{J}^T\mathbf{F}_e \quad (10-3)$$

where

$$\mathbf{M}_r = \begin{bmatrix} \frac{J_{\alpha 1} r_{\theta 1}^2}{\Delta_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} + \frac{J_{\beta 1} r_{\theta 1}^2}{\Delta_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} & 0 \\ 0 & \frac{J_{\alpha 2} r_{\theta 2}^2}{\Delta_{\alpha 2} r_{\alpha 2}^2 \tan \gamma_2} + \frac{J_{\beta 2} r_{\theta 2}^2}{\Delta_{\beta 2} r_{\beta 2}^2 \tan \gamma_2} \end{bmatrix}$$

$$\mathbf{C}_r = \begin{bmatrix} \frac{C_{f\alpha 1} r_{\theta 1}^2}{\Delta_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} + \frac{C_{f\beta 1} r_{\theta 1}^2}{\Delta_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} & 0 \\ 0 & \frac{C_{f\alpha 2} r_{\theta 2}^2}{\Delta_{\alpha 2} r_{\alpha 2}^2 \tan \gamma_2} + \frac{C_{f\beta 2} r_{\theta 2}^2}{\Delta_{\beta 2} r_{\beta 2}^2 \tan \gamma_2} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{r_{\theta 1}}{\Delta_{\alpha 1} r_{\alpha 1}} & \frac{r_{\theta 1}}{\Delta_{\beta 1} r_{\beta 1}} & 0 & 0 \\ 0 & 0 & \frac{r_{\theta 2}}{\Delta_{\alpha 2} r_{\alpha 2}} & \frac{r_{\theta 2}}{\Delta_{\beta 2} r_{\beta 2}} \end{bmatrix} \quad \boldsymbol{\tau}_m = \begin{bmatrix} \tau_{m\alpha 1} & \tau_{m\beta 1} & \tau_{m\alpha 2} & \tau_{m\beta 2} \end{bmatrix}^T$$

and

$$\mathbf{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$$

However, due to the placement of the strain gauges, i.e. on each link, it is possible to simplify equation (10-3) by introducing the terms $\tau_{\alpha c1}$ and $\tau_{\alpha c2}$ which represent measurements from the two sensors. The use of the sensors in this manner mathematically decouples the dynamic equations due to the explicit knowledge of the forces acting on each joint, including coriolis and inertial forces. The position of the sensors does not allow for the measurement of the total inertial force. The small amount of unmeasured inertial force generated by the *residual* inertia, \mathbf{M}_r , includes the inertia of the worm-wheel, the main shaft, and a small portion of the link up to the strain-gauge module. The total force measured by the strain gauge sensors is given by equation (10-4)

$$\tau_{act} = -(\mathbf{M} - \mathbf{M}_o)\ddot{\mathbf{q}} - \mathbf{V}_m\dot{\mathbf{q}} + \mathbf{J}^T \mathbf{F}_e \quad (10-4)$$

where

$$\mathbf{M}_o = \begin{bmatrix} J_{o1} & 0 \\ 0 & J_{o2} \end{bmatrix} \quad \tau_{act} = [\tau_{act1} \quad \tau_{act2}]^T$$

Therefore, it is possible to write the de-coupled system equation by substituting equation (10-4) into (10-3)

$$(\mathbf{M}_d + \mathbf{M}_r)\ddot{\mathbf{q}} + (\mathbf{C} + \mathbf{C}_r)\dot{\mathbf{q}} = \mathbf{B}\tau_m + \tau_{act} \quad (10-5)$$

where

$$\mathbf{M}_o = \begin{bmatrix} J_{o1} & 0 \\ 0 & J_{o2} \end{bmatrix} \quad \tau_{act} = [\tau_{act1} \quad \tau_{act2}]^T$$

and J_{o_n} represents the residual inertia of link n .

10.5. Computed-Torque Controller

The difference between the desired trajectory and the actual trajectory is defined as

$$\mathbf{E} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \mathbf{q}_d - \mathbf{q} \quad (10-6)$$

Using equations (10-5) and (10-6) it is possible to write the error dynamics as

$$\mathbf{E} = (\mathbf{M}_o + \mathbf{M}_r)^{-1}((\mathbf{M}_o + \mathbf{M}_r)\ddot{\mathbf{q}}_d + (\mathbf{C} + \mathbf{C}_r)\dot{\mathbf{q}} - \mathbf{B}\tau_m - \tau_{act}) \quad (10-7)$$

10.5.1 Joint 1 α -unwinding controller

During α -unwinding control of joint 1 the torque, $\tau_{m\alpha 1}$, is used to regulate position. In order to track the desired trajectory it is possible to select the computed-torque control law as

$$\tau_{m\alpha 1} = \frac{\hat{\Delta}_{\alpha 1} r_{\alpha 1}}{r_{\theta 1}} \left(\left(\hat{J}_{\theta 1} + \frac{\hat{J}_{\alpha 1} r_{\theta 1}^2}{\hat{\Delta}_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} + \frac{\hat{J}_{\beta 1} r_{\theta 1}^2}{\hat{\Delta}_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} \right) (\theta_{d1} + u_{\alpha 1}) + (\hat{C}_{11} + \hat{C}_{r11}) \dot{\theta}_1 - \frac{r_{\theta 1}}{\hat{\Delta}_{\beta 1} r_{\beta 1}} \tau_{m\beta 1} - \tau_{act1} \right) \quad (10-8)$$

where $u_{\alpha 1}$ is the Proportional+Integral+Derivative (PID) control law given by

$$u_{\alpha 1} = K_{p\alpha 1}e_1 + K_{v\alpha 1}\dot{e}_1 + K_{i\alpha 1}\int e_1 dt \quad (10-9)$$

The terms $\hat{J}_{\theta 1}$, $\hat{J}_{\alpha 1}$, $\hat{J}_{\beta 1}$, $\hat{\Delta}_{\alpha 1}$, $\hat{\Delta}_{\beta 1}$, \hat{C}_{11} , \hat{C}_{r11} , \hat{B}_{11} , and \hat{B}_{12} in equation (10-10) are the estimates of $J_{\theta 1}$, $J_{\alpha 1}$, $J_{\beta 1}$, $\Delta_{\alpha 1}$, $\Delta_{\beta 1}$, C_{11} , C_{r11} , B_{11} , and B_{12} respectively.

10 5 2 Joint 1 β -unwinding controller

Similarly $\tau_{m\beta 1}$ is used to regulate position for β -unwinding control of joint 1. Hence the computed-torque PID controller given by equations (10-10) and (10-11) is defined

$$\tau_{m\beta 1} = \frac{\hat{\Delta}_{\beta 1} r_{\beta 1}}{r_{\theta 1}} \left(\left(\hat{J}_{\theta 1} + \frac{\hat{J}_{\alpha 1} r_{\alpha 1}^2}{\hat{\Delta}_{\alpha 1} r_{\alpha 1}^2 \tan \gamma_1} + \frac{\hat{J}_{\beta 1} r_{\beta 1}^2}{\hat{\Delta}_{\beta 1} r_{\beta 1}^2 \tan \gamma_1} \right) (\theta_{d1} + u_{\beta 1}) + (\hat{C}_{11} + \hat{C}_{r11}) \theta_1 - \frac{r_{\theta 1}}{\hat{\Delta}_{\alpha 1} r_{\alpha 1}} \tau_{m\alpha 1} - \tau_{ac1} \right) \quad (10-10)$$

$$u_{\beta 1} = K_{p\beta 1}e_1 + K_{v\beta 1}\dot{e}_1 + K_{i\beta 1}\int e_1 dt \quad (10-11)$$

10 5 3 Joint 2 α -unwinding controller

For α -unwinding control of joint 2, the computed-torque PID controller is chosen as in equations (10-12) and (10-13)

$$\tau_{m\alpha 2} = \frac{\hat{\Delta}_{\alpha 2} r_{\alpha 2}}{r_{\theta 2}} \left(\left(\hat{J}_{\theta 2} + \frac{\hat{J}_{\alpha 2} r_{\alpha 2}^2}{\hat{\Delta}_{\alpha 2} r_{\alpha 2}^2 \tan \gamma_2} + \frac{\hat{J}_{\beta 2} r_{\beta 2}^2}{\hat{\Delta}_{\beta 2} r_{\beta 2}^2 \tan \gamma_2} \right) (\theta_{d2} + u_{\alpha 2}) + (\hat{C}_{22} + \hat{C}_{r22}) \theta_2 - \frac{r_{\theta 2}}{\hat{\Delta}_{\beta 2} r_{\beta 2}} \tau_{m\beta 2} - \tau_{ac2} \right) \quad (10-12)$$

$$u_{\alpha 2} = K_{p\alpha 2}e_2 + K_{v\alpha 2}\dot{e}_2 + K_{i\alpha 2}\int e_2 dt \quad (10-13)$$

The terms $\hat{J}_{\theta 2}$, $\hat{J}_{\alpha 2}$, $\hat{J}_{\beta 2}$, $\hat{\Delta}_{\alpha 2}$, $\hat{\Delta}_{\beta 2}$, \hat{C}_{22} , \hat{C}_{r22} , \hat{B}_{21} , and \hat{B}_{24} in equation (10-12) are the estimates of $J_{\theta 2}$, $J_{\alpha 2}$, $J_{\beta 2}$, $\Delta_{\alpha 2}$, $\Delta_{\beta 2}$, C_{22} , C_{r22} , B_{21} , and B_{24} respectively.

10.5.4 Joint2 β -unwinding controller

β -unwinding control of joint 2 is accomplished using the computed-torque PID controller of equations (10-14) and (10-15).

$$\tau_{m\beta 2} = \frac{\hat{\Delta}_{\beta 2} r_{\beta 2}}{r_{\theta 2}} \left(\left(\hat{J}_{\theta 2} + \frac{\hat{J}_{\alpha 2} r_{\theta 2}^2}{\hat{\Delta}_{\alpha 2} r_{\alpha 2}^2 \tan \gamma_2} + \frac{\hat{J}_{\beta 2} r_{\theta 2}^2}{\hat{\Delta}_{\beta 2} r_{\beta 2}^2 \tan \gamma_2} \right) (\theta_{d2} + u_{\beta 2}) + (\hat{C}_{22} + \hat{C}_{r22}) \theta_2 - \frac{r_{\theta 2}}{\hat{\Delta}_{\alpha 2} r_{\alpha 2}} \tau_{ma2} - \tau_{ac12} \right) \quad (10-14)$$

$$u_{\beta 2} = K_{p\beta 2} e_2 + K_{v\beta 2} \dot{e}_2 + K_{i\beta 2} \int e_2 dt \quad (10-15)$$

10.6. Experimental Set-up

The 2-DOF manipulator was controlled using the Mechatronic Workstation described in Chapter 5. A schematic overview of the system is shown in Figure 66. The computed-torque control method requires regulation of the motor torque. Closed-loop control of the motor torque was accomplished by measuring the motor current and using a digital Proportional-plus-Integral (PI) regulator. The pointer mechanism in the control handle was controlled using the HCTL-1100 Motion Control Interface. Joint position measurements were obtained from the 5,000 count encoders and separate HCTL-1100 interfaces were used to decode the quadrature signal into 20,000 counts/revolution. Velocity measurement was obtained using a standard backward difference algorithm. The control algorithm was run at frequency of 600Hz. In order to enhance user safety a foot-switch was introduced into the system. The foot-switch physically controls connection of all motors to the power amplifiers through relays as shown in Figure 67. With the foot-switch disabled the amplifiers cannot drive the motors. The control PC also monitors the state of the foot-switch disabling/enabling the control algorithm appropriately.

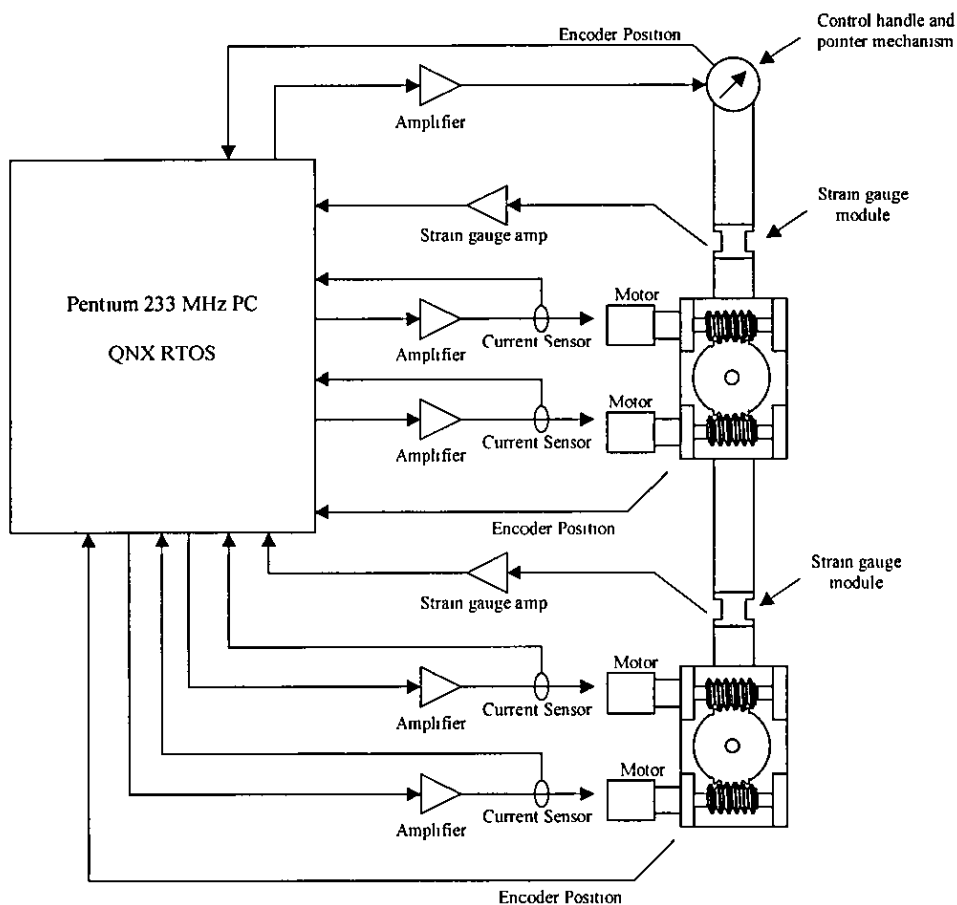


Figure 66 - A schematic overview of the 2-DOF manipulator control system

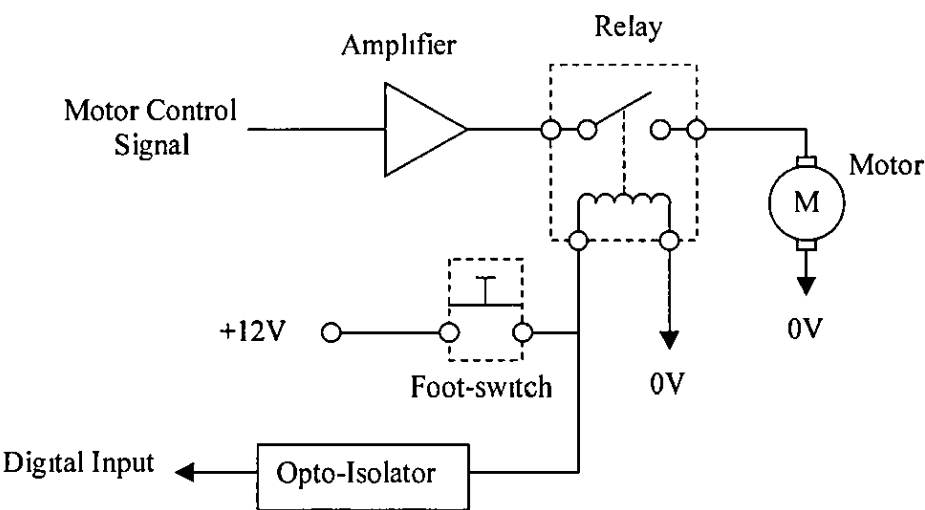


Figure 67 - Implementation of the foot-switch

10.7. Results and Discussion

To test the ability of the manipulator to track a trajectory the end-effector is required to track a circular path. The path is defined as 0.05m diameter circle that starts at the manipulators current position. The parameters used to compensate for system dynamics in the control algorithm are given in Table 10. The gains used in the α and β unwinding controllers are given in Table 11. The velocity along the path was generated from the user-input force using $F_{u\max} = 50\text{ N}$, $F_{u\min} = 10\text{ N}$, and $V_{d\max} = 0.005\text{ m/s}$ as given by equation (10-2).

Table 10 - System Parameters

| | Worm wheel | α -worm | β -worm |
|---------------------------------------|--|--|--|
| Inertia (Kg m^2) | $J_{\theta 1} = J_{\theta 2} = 0.25122$ | $J_{\alpha 1} = J_{\alpha 2} = 0.003698$ | $J_{\beta 1} = J_{\beta 2} = 0.003698$ |
| Viscous Damping N m s/rad | $C_{\theta 1} = C_{\theta 2} = 0.005$ | $C_{fa1} = C_{fa2} = 0.001$ | $C_{fb1} = C_{fb2} = 0.001$ |
| Gear PCD (m) | $r_{\theta 1} = r_{\theta 2} = 0.025$ | $r_{\alpha 1} = r_{\alpha 2} = 0.0065$ | $r_{\beta 1} = r_{\beta 2} = 0.0065$ |
| Static Friction | | $\mu_{s\alpha 1} = \mu_{s\alpha 2} = 0.12$ | $\mu_{s\beta 1} = \mu_{s\beta 2} = 0.16$ |
| Dynamic Friction | | $\mu_{d\alpha 1} = \mu_{d\alpha 2} = 0.10$ | $\mu_{d\beta 1} = \mu_{d\beta 2} = 0.14$ |
| Friction Constant | | $\varepsilon_1 = \varepsilon_2 = 0.001$ | $\varepsilon_1 = \varepsilon_2 = 0.001$ |
| Worm lead angle | | $\gamma_1 = \gamma_2 = 0.05236$ | $\gamma_1 = \gamma_2 = 0.05236$ |
| Joint link lengths | $l_1 = 0.450\text{ m}$ $l_2 = 0.415\text{ m}$ | | |

Table 11 - Computed-Torque PID Gains

| | a-unwinding | | b-unwinding | |
|--------------|-------------|---------|-------------|---------|
| | Joint 1 | Joint 2 | Joint 1 | Joint 2 |
| Proportional | 1000 | 1000 | 1000 | 1000 |
| Integral | 100 | 100 | 100 | 100 |
| Derivative | 1000 | 1000 | 1000 | 1000 |

The desired and actual paths of the end-effector are shown in Figure 68 for two laps of the path. The start position of the end-effector on the path is also illustrated. It can be seen that in general the end-effector of the robot tracks the path with little error.

Whilst the plot of x-position illustrates good tracking performance the graph of y-position shows some deviation from the path. Figure 69 illustrates the error in the position of the end-effector shown in Cartesian co-ordinates. The x-axis error, shown in Figure 69a, does not exceed 0.6mm at any time while the y-axis error, Figure 69b, remains under 2mm. Figure 70 shows the joint torque measured by the strain gauge modules. Figure 71 shows the Cartesian components of the equivalent end-effector force calculated from the measured joint torque. During the two laps of the path, four main points (1-4) have been marked that identify large tracking errors. These four points are shown in Figure 68, Figure 69, and Figure 70. It is clear to see that these errors coincide with changes in sign of the measured joint torque, τ_{act1} and τ_{act2} .

The end-effector error, owing to the flexibility of joint 1, is also dependent upon the manipulator configuration but can be highlighted in a static test. The test has shown that for every 10N of user-input force in the y-direction the end-effector is deflected by approximately 1.15mm in the y-axis in the given manipulator configuration. For the same force in the x-direction the end-effector is deflected only 0.1mm in the x-axis. By increasing the stiffness of the main shaft of joint 1 the error due to flexibility can be significantly reduced.

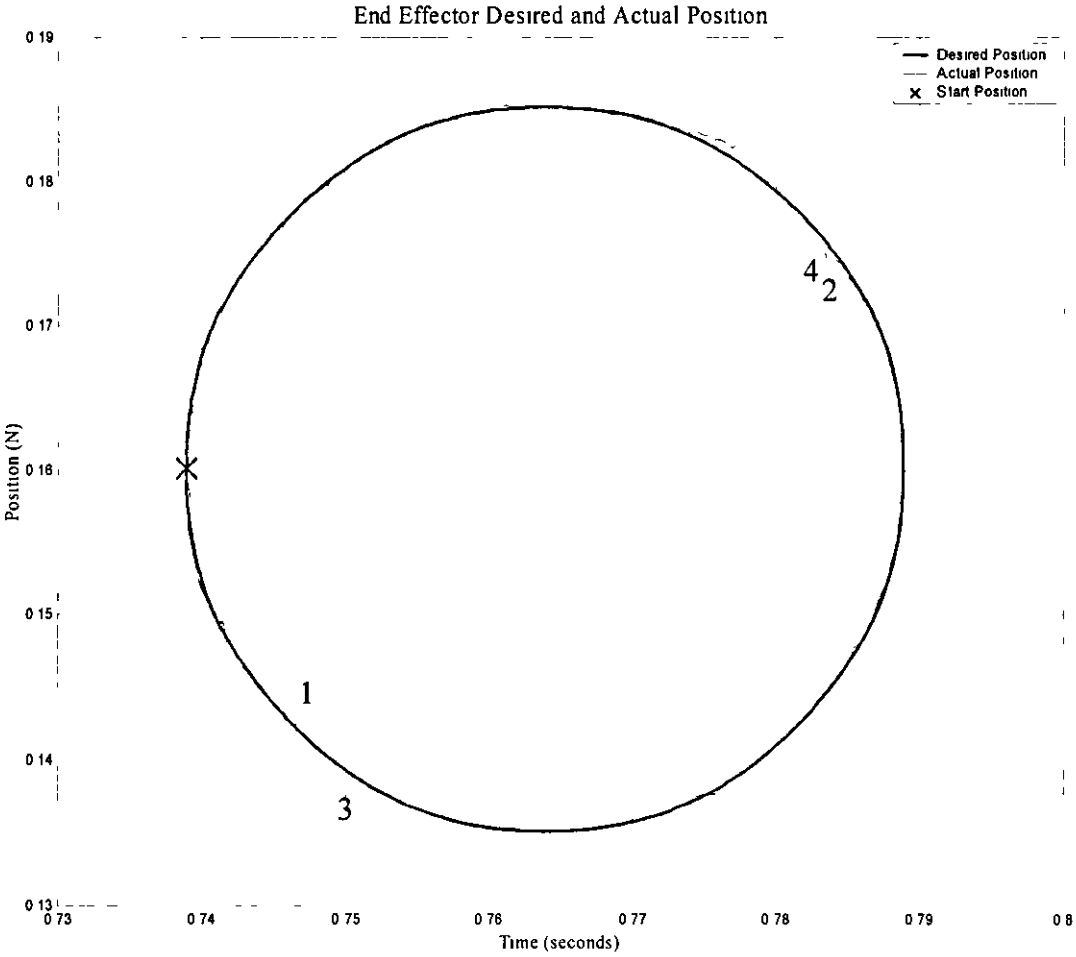


Figure 68 - Tracking of a circular path

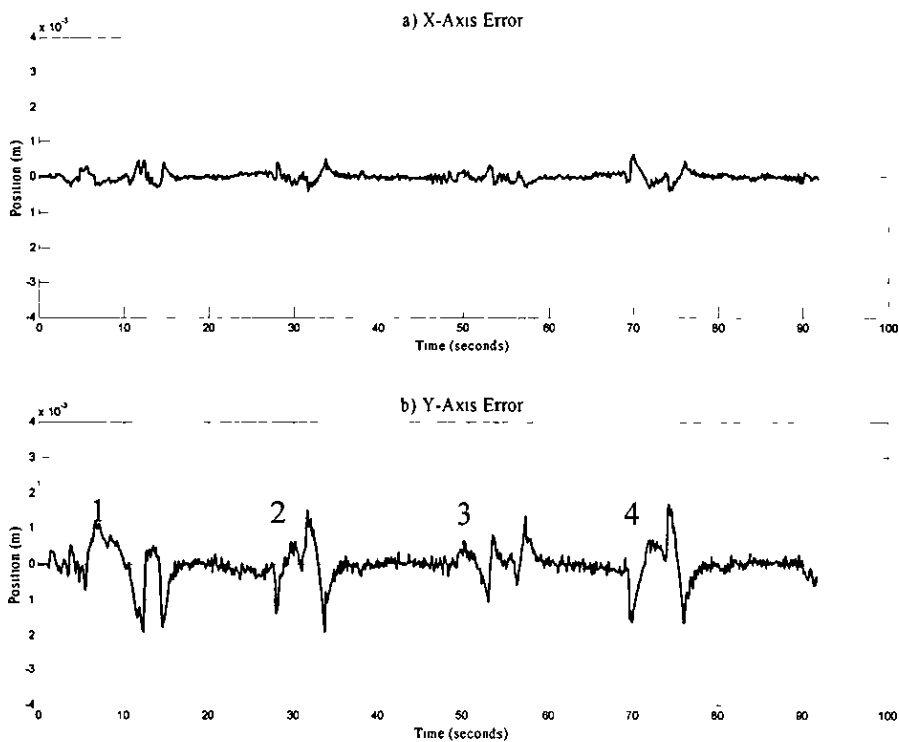


Figure 69 - End-effector position error

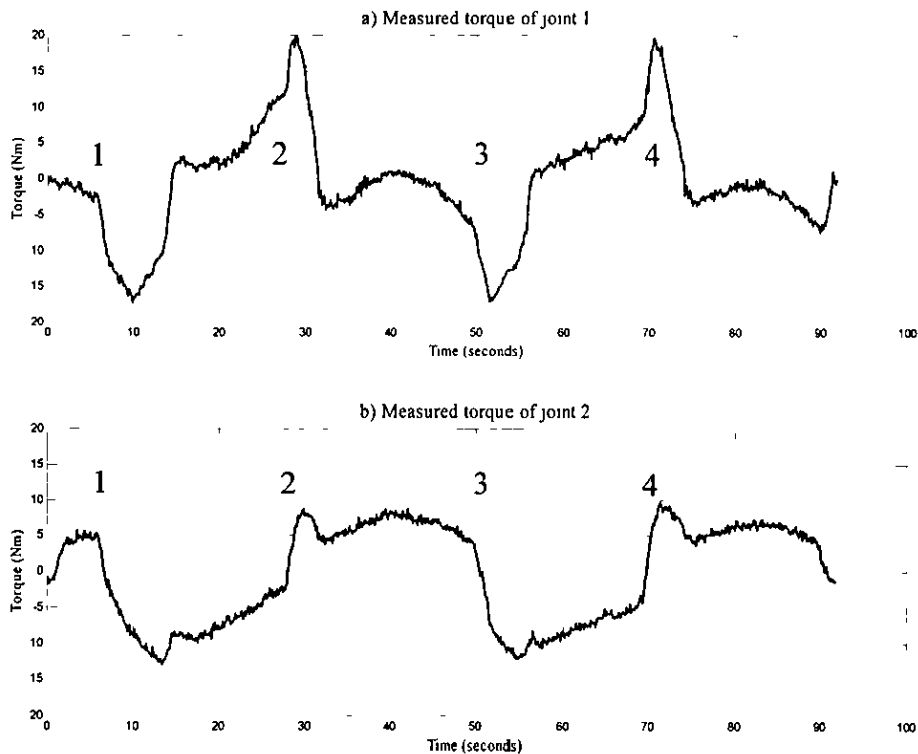


Figure 70 - Measured joint torque for circular path

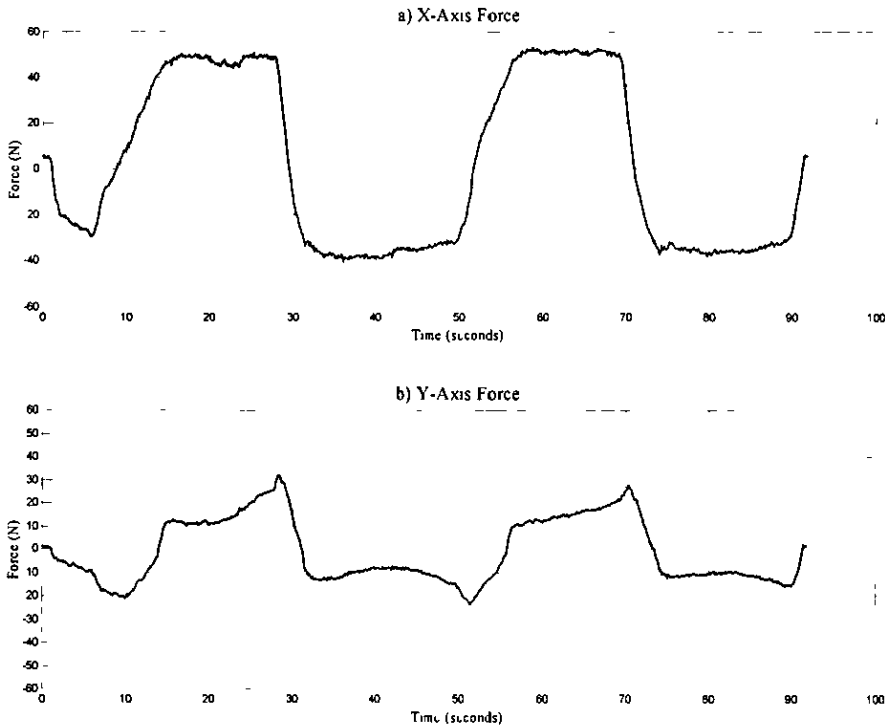


Figure 71 – Applied user-input force

In addition to the circular path of Figure 68, a more complicated path composed of both linear and curvilinear segments has been implemented. The desired and actual end-effector positions for one lap of the path (including the start point) are shown in Figure 72. Figure 73 and Figure 74 illustrate the end-effector position error and the applied user-input force respectively. The magnitude of the error in the x-axis is less than 0.8mm while in the y-axis the magnitude remains under 3mm.

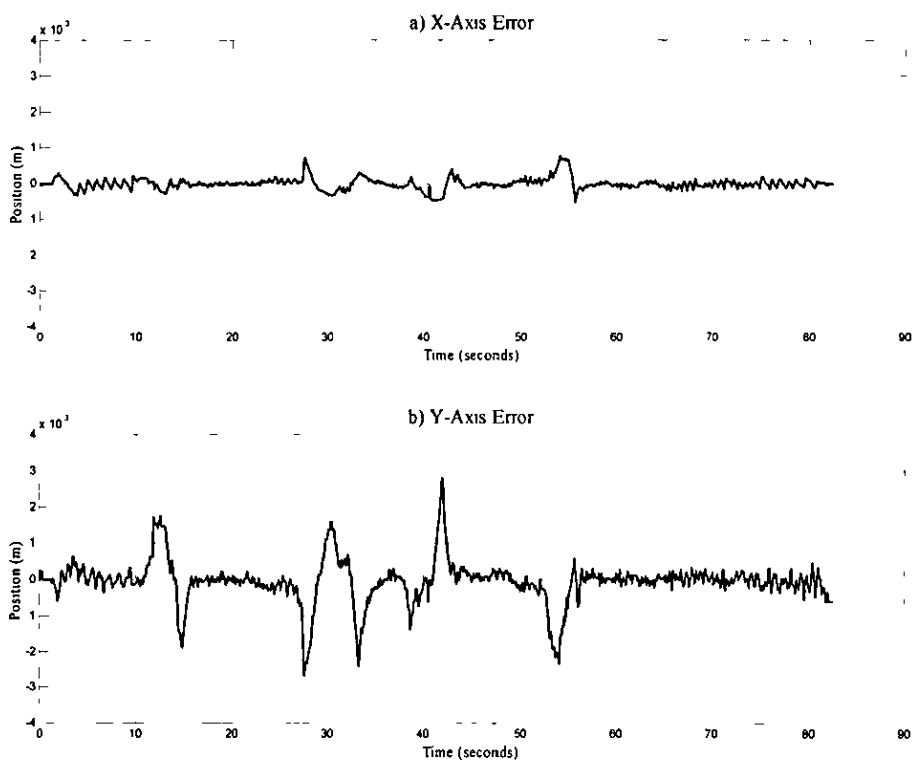


Figure 73 - End-effector position error

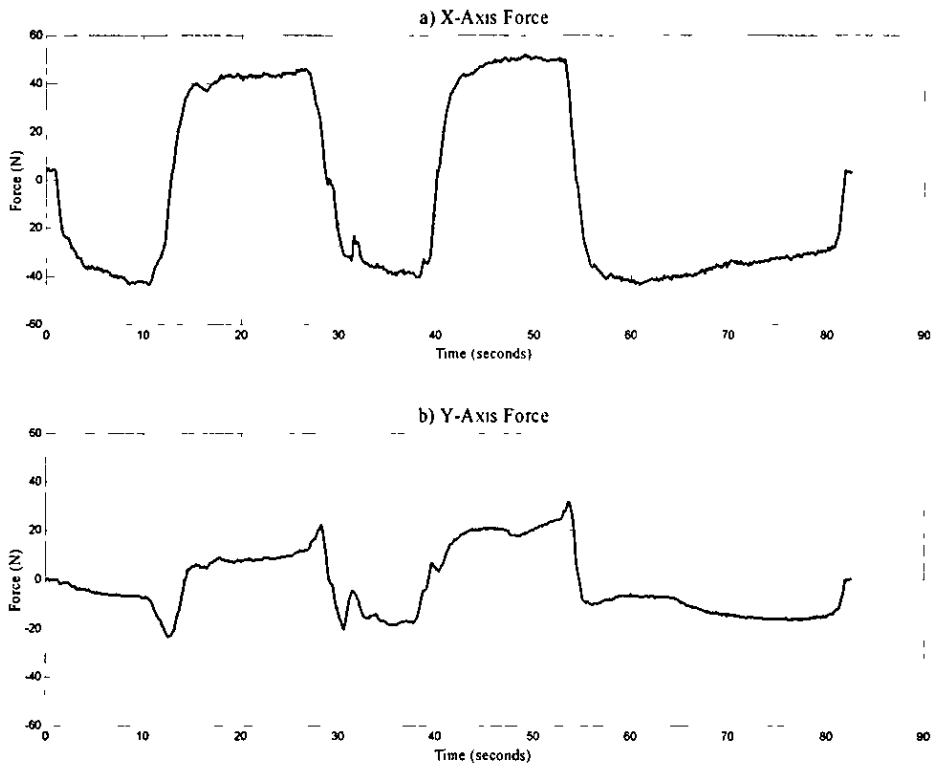


Figure 74 - Applied user-input force

10 7 1 Discussion

The results presented here are for a given velocity demand that is dependent upon the force that is applied by the user. In the simulations of the 1-DOF system the interaction of the user with the end-effector has been ignored. However, in reality the ability of the user to respond to the motion of the manipulator plays an important role in the ability of the manipulator end-effector to track the desired path. Thus the choices of $V_{d\max}$, $F_{u\min}$, and $F_{u\max}$ in equation (10-2) are critical to the performance of the controller, i.e. small motions of the first joint in this 2-DOF system can mean significant motion of the end-effector. If the user is unable to keep-up with the end-effector the user-force will drop and the manipulator will slow down in accordance with equation (10-2). The user then catches up with the end-effector, i.e. the force applied by the user increases, and the joint is required to speed up. This effect is compounded further by the stiffness problems of the first joint. The net effect of large $V_{d\max}$ is to generate a vibrating motion at the manipulator end-effector excited by both

the interaction with the user and oscillations generated from overall system stiffness. This is highlighted by the experimental results shown in Figure 75, Figure 76, and Figure 77.

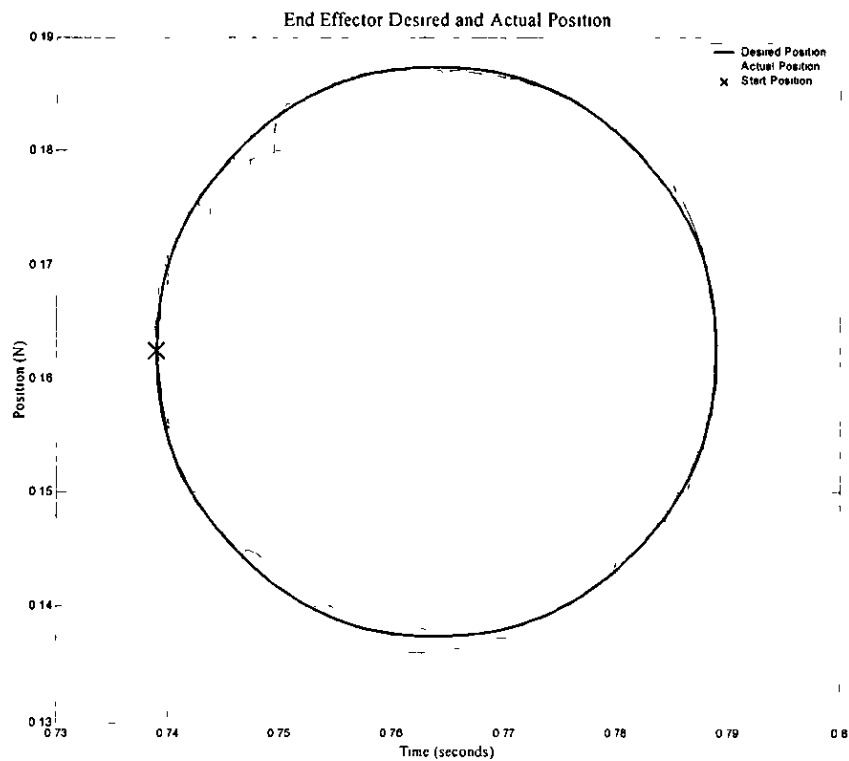


Figure 75 - End-effector position with $V_{d\max} = 0.01\text{ m/s}$

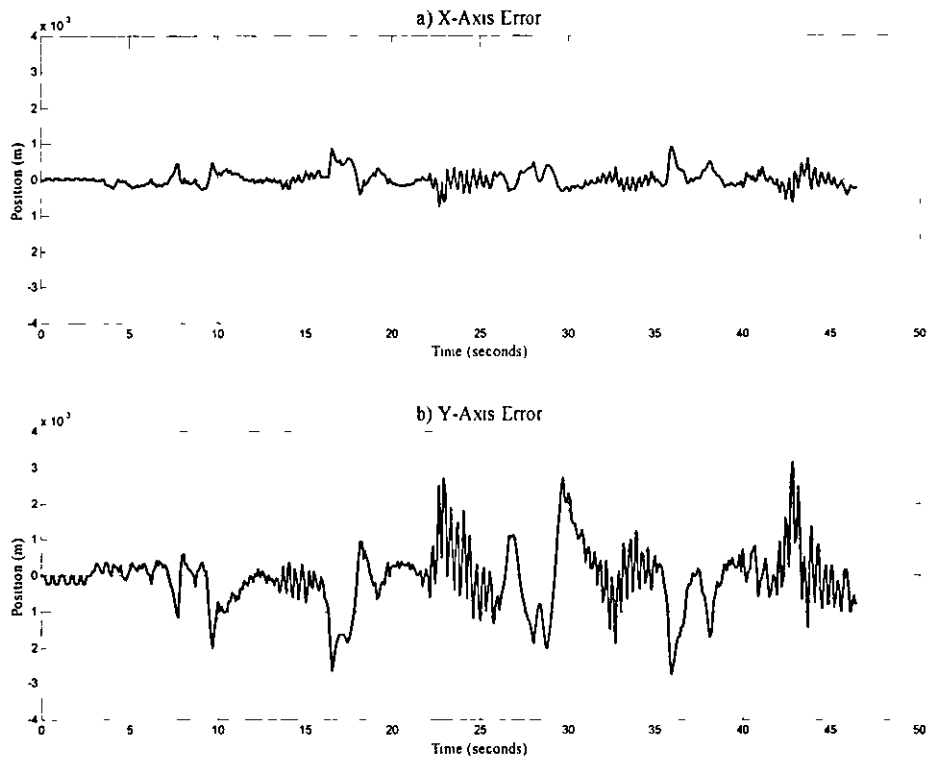


Figure 76 - End-effector position error with $V_{d\max} = 0.01$ m/s

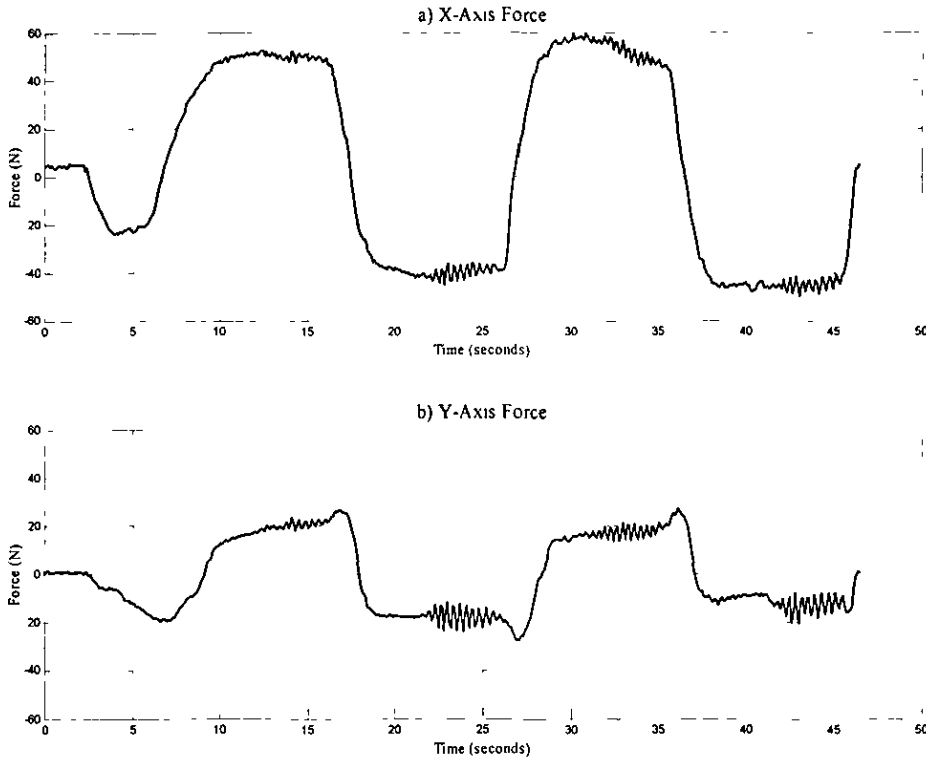


Figure 77 - Applied user-input force

A second problem, related to the user-input command signal has also been highlighted by the experimental results above. The large errors caused by the changing sign of τ_{act1} and τ_{act2} stem from the control strategy itself. The transition of the user-input across the boundaries defined by Table 2 causes the end-effector position to ‘jump’. This is explained by the fact that this boundary introduces a discontinuity to the velocity requirement of equation (10-2). Consider the point when the path crosses the boundary from region 3 to region 4 of Figure 64. Unless the direction of the user-input force happens to cross the boundary from C to D at exactly the same time the control algorithm decides that motion cannot be allowed and the desired velocity, V_d , is set to zero. The user compensates for this and changes the direction of the applied force to point in to region D. The controller realises that this condition is suitable to allow motion and sets the user velocity according to equation (10-2). However, the desired velocity has instantaneously changed from $V_d = 0 \text{ m/s}$ to a value dependent upon the magnitude of the applied force. This in turn means that the acceleration demand is very high. From the definition of the motor torque, equations (8-8), (8-10),

(8-12), and (8-14), a large torque is required from the motors. Therefore the system lurches, resulting in a large error. Although both axes are affected, the first joint is more susceptible owing to the configuration of the mechanism. However, in the regions where the sign of τ_{act1} and τ_{act2} are well established, tracking performance can be seen to be significantly under 1mm in both x and y axes as shown in Figure 73.

10.8. Conclusions

It has been shown that by directing the user to apply a force in the correct way, the manipulator can be forced to track a desired path without driving against the user. The error was shown to be less than 3mm in magnitude and in view of the large errors that can be obtained due to the application of a force by the user (approximately 4mm for 35N) this is considered to be reasonable. Since the user can easily apply forces to the control handle in excess of 50N, the error due to flexibility within this two DOF manipulator cannot be tolerated in most medical applications. However, a mechanical re-design of the system and reconfiguration of the manipulator would result in increased robustness to large user-input forces. Further development of the control strategy at the critical points where measured joint torque changes sign would also enhance performance.

Chapter 11 Composite Adaptive Control of a Two DOF Worm-Driven Manipulator

11.1. Introduction

In an attempt to improve on the performance of the computed-torque control algorithm, a composite adaptive controller for the 2-DOF manipulator is presented. In this chapter a control scheme based on the adaptive controller presented in Chapter 8 is presented. It is again shown that the location of the strain gauges significantly reduces the complexity of the control algorithm.

11.2. Control Strategy

The control strategy is the same as described in Chapter 10. Four controllers are defined, i.e. α and β unwinding controllers for the two joints. The velocity along the path is generated by the measured user-input force as given by equation (10-2) and shown in Figure 63.

11.3. Experimental Set-up

The 2-DOF manipulator was controlled using the Mechatronic Workstation as described in Chapter 10. A schematic overview of the system is shown in Figure 66. The composite adaptive control method requires regulation of the motor torque. Closed-loop control of the motor torque was accomplished by measuring the motor current using a digital Proportional-plus-Integral (PI) regulator.

11.4. Mathematical Model

The mathematical model of the 2-DOF worm-driven manipulator is given by (9-25) and is repeated here for clarity

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{M}_r\dot{\mathbf{q}} + \mathbf{C}\mathbf{q} + \mathbf{C}_r\dot{\mathbf{q}} + \mathbf{V}_m\mathbf{q} = \mathbf{B}\boldsymbol{\tau}_m + \mathbf{J}^T\mathbf{F}_u \quad (11-1)$$

As stated in Chapter 10, due to the placement of the strain gauges, i.e. on each link, it is possible to simplify equation (11-1) by introducing the terms τ_{act1} and τ_{act2} which represent measurements from the two sensors. The use of the sensors in this manner mathematically decouples the dynamic equations due to the explicit knowledge of the forces acting on each joint. This *residual* inertia includes the inertia of the worm-wheel, the main shaft, and a small portion of the link up to the strain-gauge module

$$\boldsymbol{\tau}_{act} = -(\mathbf{M} - \mathbf{M}_\theta)\ddot{\mathbf{q}} - \mathbf{V}_m\mathbf{q} + \mathbf{J}^T\mathbf{F}_u \quad (11-2)$$

where

$$\mathbf{M}_\theta = \begin{bmatrix} J_{\theta 1} & 0 \\ 0 & J_{\theta 2} \end{bmatrix} \quad \boldsymbol{\tau}_{act} = \begin{bmatrix} \tau_{act1} & \tau_{act2} \end{bmatrix}^T$$

Therefore, it is possible to write the system equation as

$$(\mathbf{M}_\theta + \mathbf{M}_r)\ddot{\mathbf{q}} + (\mathbf{C} + \mathbf{C}_r)\dot{\mathbf{q}} = \mathbf{B}\boldsymbol{\tau}_m + \boldsymbol{\tau}_{act} \quad (11-3)$$

and $J_{\theta i}$ is the 'residual' inertia that generates an acceleration force not measured by τ_{acti} .

11.5. Composite Adaptive Controller

The error, \mathbf{E} , and filtered tracking error, \mathbf{r} , are defined by equations (10-6) and (11-5) respectively.

$$\mathbf{E} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \mathbf{q}_d - \mathbf{q} \quad (11-4)$$

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \mathbf{E} + \lambda \mathbf{E} \quad (11-5)$$

11.5.1. α -unwinding control of joint n

The de-coupled equation of motion for the system of gears, given in (11-3), can be used to solve for the α -motor torque, $\tau_{m\alpha n}$, by multiplying both sides by $\Delta_{\alpha n} r_{\alpha n} / r_{\beta n}$. That is

$$\tau_{m\alpha n} = J_{\gamma \alpha n} \ddot{\theta}_n + C_{\gamma \alpha n} \dot{\theta}_n - \tau_{\alpha c n} \frac{\Delta_{\alpha n} r_{\alpha n}}{r_{\beta n}} - \tau_{m\beta n} \frac{\Delta_{\alpha n} r_{\alpha n}}{\Delta_{\beta n} r_{\beta n}} \quad (11-6)$$

where

$$J_{\gamma \alpha n} = \left(J_{\beta n} + \frac{J_{\alpha n} r_{\beta n}^2}{\Delta_{\alpha n} r_{\alpha n}^2 \tan \gamma} + \frac{J_{\beta n} r_{\beta n}^2}{\Delta_{\beta n} r_{\beta n}^2 \tan \gamma} \right) \frac{\Delta_{\alpha n} r_{\alpha n}}{r_{\beta n}}$$

and

$$C_{\gamma \alpha n} = \left(C_{\beta n} + \frac{C_{\alpha n} r_{\beta n}^2}{\Delta_{\alpha n} r_{\alpha n}^2 \tan \gamma} + \frac{C_{\beta n} r_{\beta n}^2}{\Delta_{\beta n} r_{\beta n}^2 \tan \gamma} \right) \frac{\Delta_{\alpha n} r_{\alpha n}}{r_{\beta n}}$$

Following the method outlined in Chapter 8, equation (11-6) can be written as

$$\tau_{m\alpha n} = W_{\alpha n} \Phi_{\alpha n} \quad (11-7)$$

where

$$W_{\alpha n} = \begin{bmatrix} \theta_n & \dot{\theta}_n & \tau_{\alpha c n} & \tau_{m\beta n} \end{bmatrix} \quad \Phi_{\alpha n} = \begin{bmatrix} J_{\gamma \alpha n} & C_{\gamma \alpha n} & -\frac{\Delta_{\alpha n} r_{\alpha n}}{r_{\beta n}} & -\frac{\Delta_{\alpha n} r_{\alpha n}}{\Delta_{\beta n} r_{\beta n}} \end{bmatrix}^T$$

Selecting the control input, $\tau_{m\alpha n}$, as

$$\tau_{m\alpha n} = Y_{\alpha n}^T \hat{\Phi}_{\alpha n} + K_{\alpha n} r_n \quad (11-8)$$

where $Y_{\alpha n} = \begin{bmatrix} (\theta_n + \lambda_n e_n) & \dot{\theta}_n & \tau_{\alpha c n} & \tau_{m\beta n} \end{bmatrix}$. The least-squares adaptive update law is given by equations (11-9) and (11-10) [104, 105]

$$\begin{aligned} \tilde{\Phi}_{\alpha n} &= -\dot{\hat{\Phi}}_{\alpha n} = -P_{\alpha n} Y_{\alpha n}^T r_n - P_{\alpha n} W_{f\alpha n}^T \tilde{\tau}_{f\alpha n} \\ \hat{\Phi}_{\alpha n} &= P_{\alpha n} Y_{\alpha n}^T r_n + P_{\alpha n} W_{f\alpha n}^T (\tau_{f\alpha} - W_{f\alpha n} \hat{\Phi}_{\alpha n}) \end{aligned} \quad (11-9)$$

and

$$\begin{aligned} P_{\alpha n}^{-1} &= W_{f\alpha n}^T W_{f\alpha n} \\ P_{\alpha n} &= -P_{\alpha n} W_{f\alpha n}^T W_{f\alpha n} P_{\alpha n} \end{aligned} \quad (11-10)$$

Substitution of the least-squares estimator given by equations (11-9) and (11-10) into the time derivative of the Lyapunov function leads to

$$V_{an} = -K_{van} r_n^2 - \tilde{\Phi}_{an}^T W_{fan}^T W_{fan} \tilde{\Phi}_{an} \quad (11-11)$$

11.5.2 β -unwinding control of joint n

Similarly, for β -unwinding control, (11-3) can be used to solve for the β -motor torque by multiplying both sides by $\Delta_{\beta n} r_{\beta n} / r_{\alpha n}$ as given in equation (11-12).

$$\tau_{m\beta n} = J_{\tau\beta n} \theta_n + C_{\tau\beta n} \dot{\theta}_n - \tau_{m\alpha n} \frac{\Delta_{\beta n} r_{\beta n}}{r_{\alpha n}} - \tau_{mcan} \frac{\Delta_{\beta n} r_{\beta n}}{\Delta_{\alpha n} r_{\alpha n}} = W_{\beta n} \Phi_{\beta n} \quad (11-12)$$

where

$$W_{\beta n} = \begin{bmatrix} \theta_n & \dot{\theta}_n & \tau_{m\alpha n} & \tau_{mcan} \end{bmatrix} \quad \Phi_{\beta n} = \begin{bmatrix} J_{\tau\beta n} & C_{\tau\beta n} & -\frac{\Delta_{\beta n} r_{\beta n}}{r_{\alpha n}} & -\frac{\Delta_{\beta n} r_{\beta n}}{\Delta_{\alpha n} r_{\alpha n}} \end{bmatrix}^T$$

$$J_{\tau\beta n} = \left(J_{\alpha n} + \frac{J_{\alpha n} r_{\alpha n}^2}{\Delta_{\alpha n} r_{\alpha n}^2 \tan \gamma} + \frac{J_{\beta n} r_{\beta n}^2}{\Delta_{\beta n} r_{\beta n}^2 \tan \gamma} \right) \frac{\Delta_{\beta n} r_{\beta n}}{r_{\alpha n}} \quad C_{\tau\beta n} = \left(C_{f\alpha n} + \frac{C_{f\alpha n} r_{\alpha n}^2}{\Delta_{\alpha n} r_{\alpha n}^2 \tan \gamma} + \frac{C_{f\beta n} r_{\beta n}^2}{\Delta_{\beta n} r_{\beta n}^2 \tan \gamma} \right) \frac{\Delta_{\beta n} r_{\beta n}}{r_{\alpha n}}$$

Following the method outlined above the control input, $\tau_{m\beta n}$, can be selected as

$$\tau_{m\beta n} = Y_{\beta n} \hat{\Phi}_{\beta n} + K_{\beta n} r_n \quad (11-13)$$

where $Y_{\beta n} = \begin{bmatrix} (\theta_{\alpha n} + \lambda_n e_n) & \dot{\theta}_n & \tau_{m\alpha n} & \tau_{mcan} \end{bmatrix}$ and $K_{\beta n}$ is a positive constant control parameter.

The least-squares adaptive update law is given in equations (11-9) and (11-10) and the time derivative of the Lyapunov function is given by equation (11-16).

$$\begin{aligned} \tilde{\Phi}_{\beta n} &= -\dot{\hat{\Phi}}_{\beta n} = -P_{\beta n} Y_{\beta n}^T r_n - P_{\beta n} W_{f\beta n}^T \tilde{\tau}_{f\beta n} \\ \dot{\hat{\Phi}}_{\beta n} &= P_{\beta n} Y_{\beta n}^T r_n + P_{\beta n} W_{f\beta n}^T (\tau_{f\beta n} - W_{f\beta n} \hat{\Phi}_{\beta n}) \end{aligned} \quad (11-14)$$

$$\begin{aligned} P_{\beta n}^{-1} &= W_{f\beta n}^T W_{f\beta n} \\ P_{\beta n} &= -P_{\beta n} W_{f\beta n}^T W_{f\beta n} P_{\beta n} \end{aligned} \quad (11-15)$$

$$V_{\beta n} = -K_{\beta n} r_n^2 - \tilde{\Phi}_{\beta n}^T W_{f\beta n}^T W_{f\beta n} \tilde{\Phi}_{\beta n} \quad (11-16)$$

11.5.3 Stability Analysis

The analysis of system stability follows the same method as given in Chapter 8. However, the analysis for the 2-DOF system is given for clarity and completeness due to the redefinition of some variables. The Lyapunov functions for α and β -unwinding may be written as

$$V_m = \frac{1}{2} J_{T_m} r_n^2 + \frac{1}{2} \tilde{\Phi}_m^T P_m^{-1} \tilde{\Phi}_m \quad (11-17)$$

where the subscript i should be replaced by α or β for α -unwinding and β -unwinding respectively and $n=1,2$ represents the joint number. Similarly, the time-derivative of the Lyapunov function can be expressed as

$$\dot{V}_m = -K_{vm} r_n^2 - \frac{1}{2} \tilde{\Phi}_m^T W_{fm}^T W_{fm} \tilde{\Phi}_m \leq 0 \quad (11-18)$$

By showing that the second derivative of the Lyapunov function is bounded, so proving that V_m is uniformly continuous, Barbalat's lemma can be invoked to show that the tracking error, r_n , and parameter error, $\tilde{\Phi}_m$, both converge to zero. Firstly, \dot{V}_m is obtained by differentiating equation (11-18) to give equation (11-19).

$$\dot{V}_m = -2K_{vm} r_n \dot{r}_n - \tilde{\Phi}_m^T W_{fm}^T W_{fm} \tilde{\Phi}_m - \tilde{\Phi}_m^T W_{fm}^T W_{fm} \tilde{\Phi}_m \quad (11-19)$$

Therefore, in order to prove that V_m is bounded and \dot{V}_m is uniformly continuous, it is necessary and sufficient to prove that r_n , \dot{r}_n , $\tilde{\Phi}_m$, $\dot{\tilde{\Phi}}_m$, W_{fm} , and \dot{W}_{fm} are bounded. \dot{V}_m is at least negative semi-definite implying that $V_m \leq V_m(0)$ and r_n and $\tilde{\Phi}_m$ are bounded. Subsequently, from the definition of r_n in equation (11-4), e_n and \dot{e}_n are also bounded. Considering Y_m as defined in equations (11-8) and (11-13), i.e. $Y_m = [\theta_{dn} + \lambda_n e_n \quad \theta_n \quad \tau_{actn} \quad \tau_{mnn}]$ where i' should be replaced by β for α -unwinding and α for β -unwinding and assuming that the inputs, τ_{actn} , τ_{mnn} , θ_{dn} , $\dot{\theta}_{dn}$, and $\dot{\theta}_{dn}$ are bounded then θ_n , $\dot{\theta}_n$, and Y_m must also be bounded. Substitution of the control law (11-8) or (11-13) into

$$J_{Tin} r_n = J_{Tin} (\theta_{dn} - \theta_n + \lambda_n e_n) \quad (11-20)$$

leads to

$$J_{Tin} r_n = Y_{in}^T \tilde{\Phi}_{in} - K_{vn} r_n \quad (11-21)$$

Noting that $J_{Tin} > 0$ then r_n , θ , and subsequently τ_{min} are also bounded. Since $\tau_{min} = W_{in}^T \Phi_{in}$, then W_{in} , Φ_{in} , and $\hat{\Phi}_{in}$ are bounded. From the definition of the filter, f , and the properties of a stable transfer function in Appendix F it can be seen that τ_{fin} , $\hat{\tau}_{fin}$, W_{fin} , and W_{fin} are also bounded. Finally, to prove that V_{in} is bounded, the parameter update law of equation (11-9) or (11-14) may be written

$$\tilde{\Phi}_{in} = -P_{in} Y_{in}^T r_n - P_{in} W_{fin}^T \tilde{\tau}_{fin} \quad (11-22)$$

From equation (11-10) or (11-15), P_{in} is bounded by $P_{in}(0)$ since $P_{in} \leq 0$ is negative semi-definite. From equation (11-22) it can be seen that $\tilde{\Phi}_{in}$ is bounded. Application of Barbalat's lemma yields

$$\lim_{t \rightarrow \infty} V_{in} = 0 \quad (11-23)$$

and

$$\lim_{t \rightarrow \infty} r_n = 0 \quad (11-24)$$

Since the relationship between r_n and e_n , equation (11-5), may also be written in terms of the strictly proper, asymptotically stable transfer function, $H_n(s)$, such that

$$e_n(s) = H_n(s) r_n(s) \quad (11-25)$$

it must also be concluded that

$$\lim_{t \rightarrow \infty} r = 0 \Rightarrow \lim_{t \rightarrow \infty} e = 0 \quad (11-26)$$

Since W_{fn} depends on θ_n , τ_{actn} , and the motor torque, it is always non-zero while the controller is operating. This implies that the infinite integral condition given by equation (11-27) holds, where λ_{\min} is the minimum eigenvalue.

$$\lim_{t \rightarrow \infty} \lambda_{\min} \left\{ \int_0^t W_{fn}^T(\sigma) W_{fn}(\sigma) d\sigma \right\} = \infty \quad (11-27)$$

Therefore it must be concluded that

$$\lim_{t \rightarrow \infty} \tilde{\Phi}_{\alpha} = 0 \quad (11-28)$$

11.5.4. α -unwinding control algorithm

Figure 78 shows the block diagram representation of the composite adaptive α -unwinding controller. The terms θ_{dn} , e_n , θ_n , r_n , τ_{actn} , τ_{mfn} , and τ_{actn} are assumed to be known or measurable quantities. By substituting the subscript α for β and vice versa the diagram represents the composite adaptive β -unwinding controller.

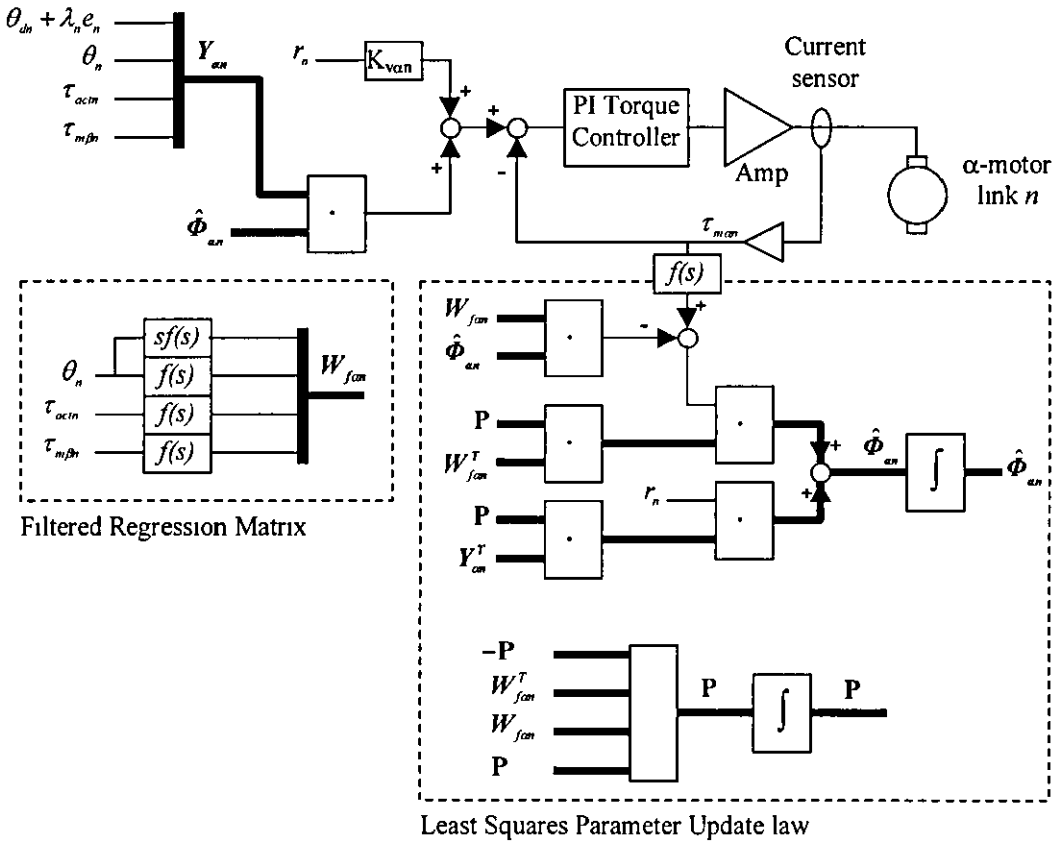


Figure 78 - Block diagram of α -unwinding control algorithm

11.6. Results and Discussion

Figure 79 shows the desired and actual path of the end-effector. The end-effector, controlled by the user, performs four 'laps' of the pre-defined path. Figure 80 illustrates the end-effector position error during composite adaptive control of the end-effector shown in Cartesian co-ordinates. The x-axis error, shown Figure 80a, exceeds 1mm and the y-axis error, Figure 80b, exceeds 10mm. Figure 81 and Figure 82 show the measured joint torque and calculated equivalent end-effector force respectively. It can be seen that the end-effector error is greatly affected by the change in sign of the measured joint torque. Figure 83 shows the estimates of the parameters for the two joints. The parameter estimates remain bounded at all times. The control parameters used in this scheme are shown in Table 12 and equations (11-29) and (11-30).

Table 12 - Composite adaptive controller parameters

| | α -unwinding | | β -unwinding | |
|--------------------|---------------------|---------|--------------------|---------|
| | Joint 1 | Joint 2 | Joint 1 | Joint 2 |
| λ_{α} | 60 | 60 | 50 | 50 |
| k_{α} | 10 | 10 | 50 | 50 |

$$\mathbf{P}_{\alpha 1}(0)=\mathbf{P}_{\beta 1}(0)=\mathbf{P}_{\alpha 2}(0)=\mathbf{P}_{\beta 2}(0)=\begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$
$$\Phi_{\alpha 1}(0)=\begin{bmatrix} 0.4 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad \Phi_{\beta 1}(0)=\begin{bmatrix} 0.4 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad \Phi_{\alpha 2}(0)=\begin{bmatrix} 0.1 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad \Phi_{\beta 2}(0)=\begin{bmatrix} 0.1 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

(11-29)

$$f(s)=\frac{10}{s+10}$$

(11-30)

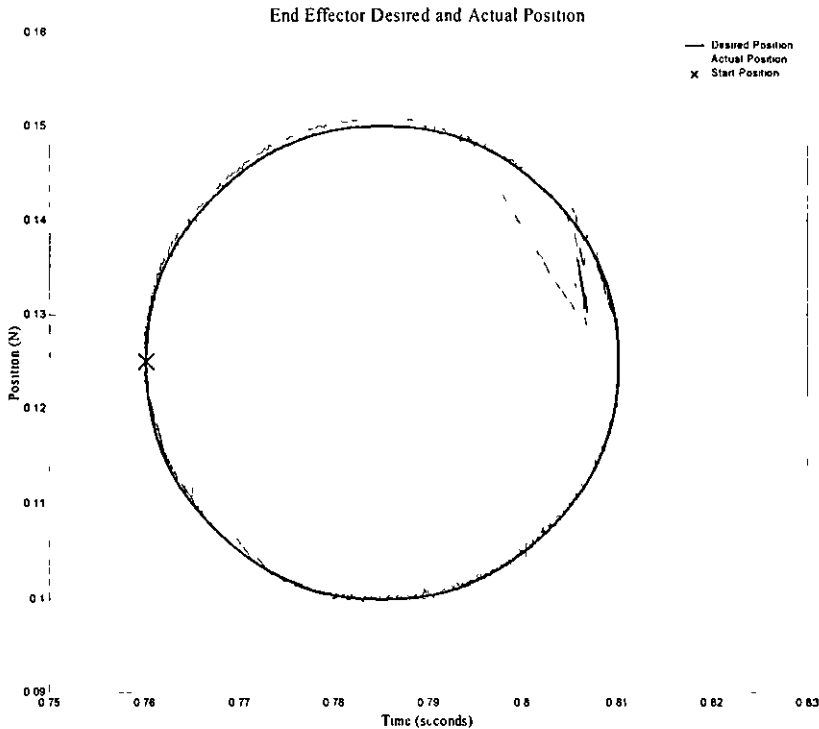


Figure 79 - End-effector position

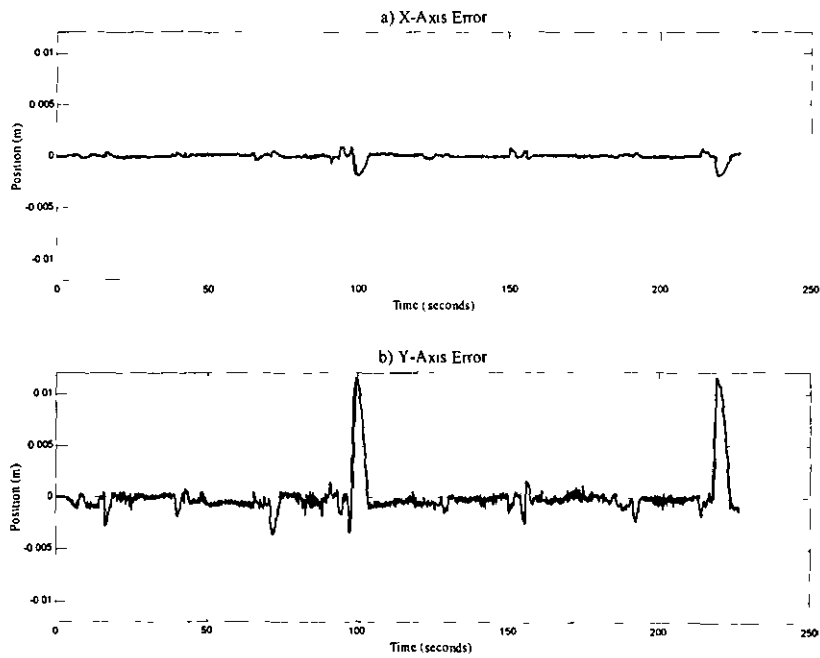


Figure 80 - End-effector error under composite adaptive control

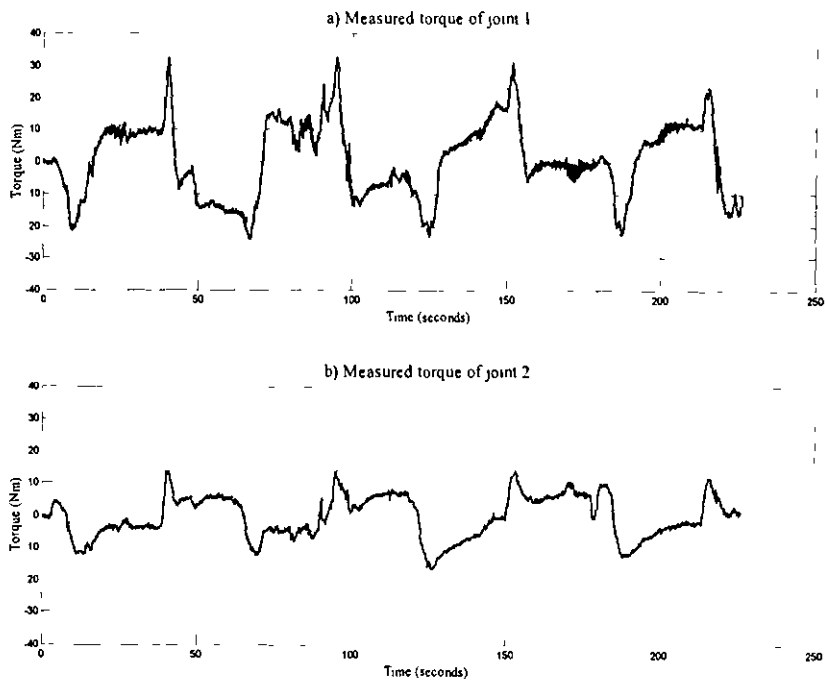


Figure 81 - Measured joint torque

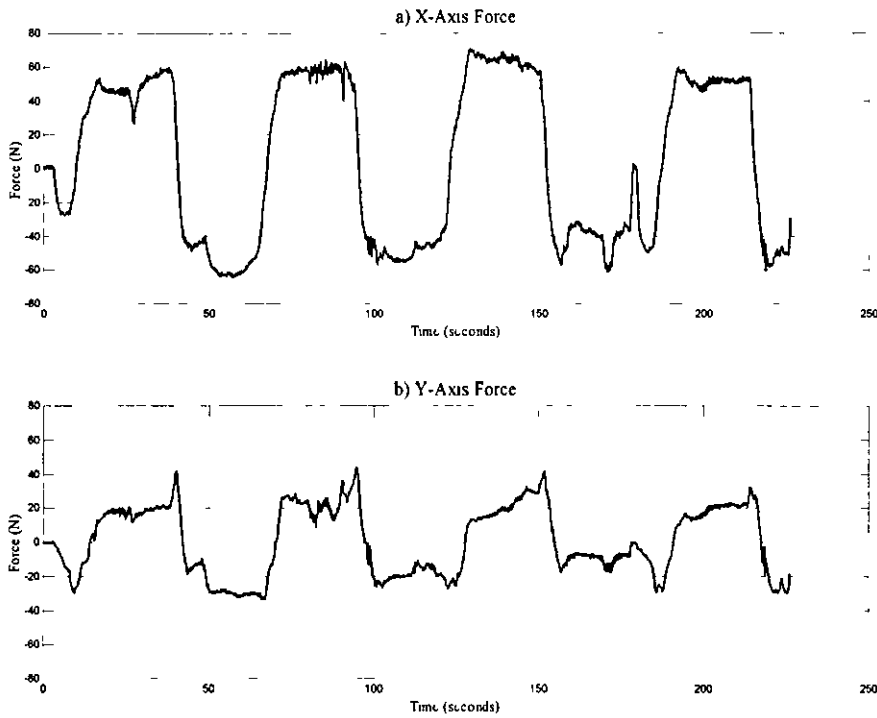


Figure 82 - Calculated equivalent end-effector force

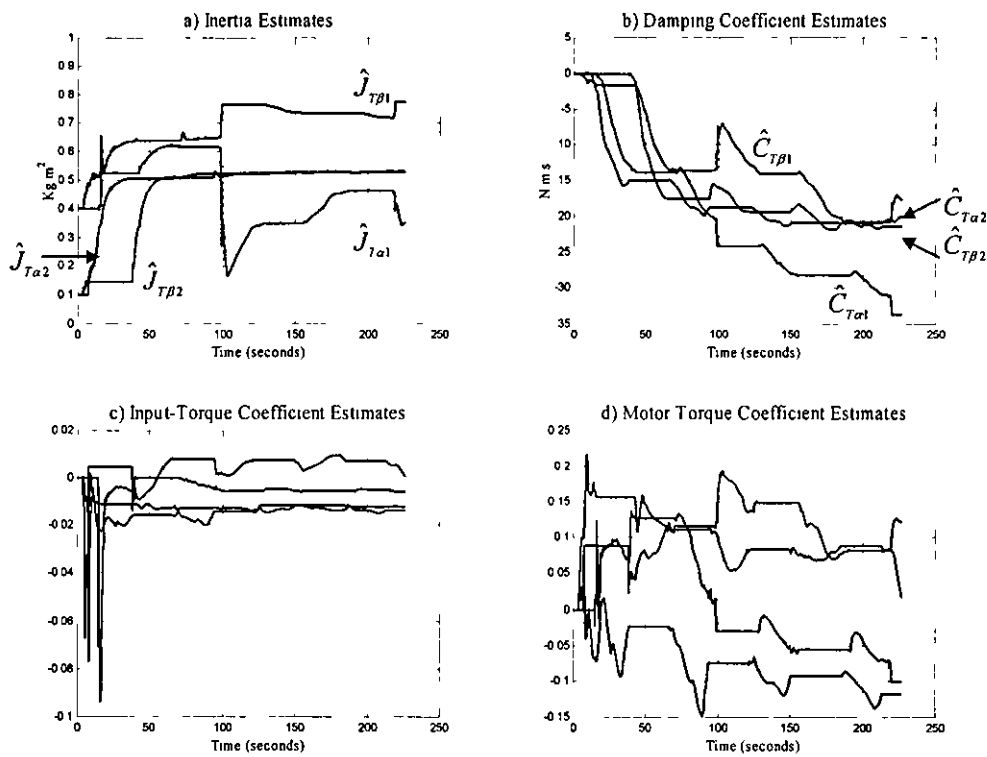


Figure 83 - Parameter estimates

11.7. Discussion

The composite adaptive controller does not perform well for the 2-DOF system. In fact, the error can be seen to be at least 5 times greater than under the computed-torque control scheme presented in Chapter 10. The composite adaptive controller is more severely affected by the change in sign of the measured joint torque. This is partially because the stability of the composite adaptive controller is based on the continuity of the inputs. It was shown in Chapter 10 that the continuity of the velocity cannot be guaranteed with the control strategy that has been presented here.

Performance of the composite adaptive controller is also degraded by joint flexibility within the 2-DOF system. The controller is easily excited by this flexibility which quickly causes the motion to become 'jerky' for any useful $V_{d\max}$. By increasing the stiffness of the main shaft of joint 1 it is expected that this effect could be significantly reduced. It can also be seen that the robot arm is in a configuration that yields poor system stiffness. The arm has a reach of 0.865m and is close to the periphery of the workspace. By redesigning the main shaft of joint 1 and reconfiguring the robot it would be possible to make the system much more rigid and in turn improve the tracking performance.

11.8. Conclusions

Using the composite adaptive controller the end-effector tracking error was not improved. However, both the tracking error and parameter estimates were shown to remain bounded at all times. There is a significant difference between the performance of the two control schemes presented in this work. If system stiffness could be increased and the control algorithm modified to give better results at the critical points where the measured joint torque changes sign, the tracking error could be significantly improved.

Chapter 12 Conclusions

12.1. Introduction

In this chapter, the conclusions that can be drawn from this work are summarised. The chapter aims to show how the objectives given in Chapter 2 have been met. The seven requirements for the manipulator and control strategy are repeated here for clarity and are defined as

- Requirement 1 : The user should also not be able to overpower the manipulator with excessive force
- Requirement 2 : The robot should move only under the direct control of the operator
- Requirement 3 : The robot should not perform unauthorised motion
- Requirement 4 : The manipulator should be unable to provide a motive force against the user.
- Requirement 5 : The control algorithms and strategies employed should be robust against external disturbances such as the user-input force or from the mechanical properties of the device itself.
- Requirement 6 : To prove performance of the control algorithm the manipulator should be shown to track a prescribed path with minimal error.
- Requirement 7 : The manipulator must be easy to use with little or no training

A list of current and proposed publications that are to be made as part of this work is also included

12.2. Conclusions from this research

This thesis has identified the limitations of passive trajectory enhancing devices such as PADyC based on a mathematical analysis of a 2-DOF planar robot mechanism. Safety issues associated with the introduction of active devices such as ROBODOC and ACROBOT have also been highlighted. As a result of this preliminary work an alternative robotic joint mechanism has been designed. The non-backdriveable design of the joint ensures that objective 1 is satisfied. The over-actuated robotic joint, in combination with a suitable control strategy, enhances user safety. The control

strategy is also used to cancel backlash at the gear interface. This particular result is not only significant in the context of medical robotics but also in terms of general servo-control problems where the elimination of backlash at an output shaft is paramount. Implementation of a footswitch, used as a dead-man's-handle ensures that the robot only moves under the control of the operator and cannot perform unauthorised motion. Therefore it is concluded that requirements 2 and 3 have been satisfied. In this particular application low power DC servo-motors have been used. However, the use of larger motors could broaden the range of applications of this gearbox mechanism.

Computed-torque and composite adaptive control algorithms have been designed for a 1-DOF manipulator based on a mathematical model of the system of gears. Simulation and experimental results show that for the 1-DOF mechanism, velocity and position demands can be generated from the measured joint torque. The error in joint position was shown to be less than ± 0.0015 rads. The composite adaptive controller did not significantly outperform the computed-torque scheme. However, the advantages of using the composite adaptive scheme include robustness to changes in system parameters and guaranteed controller stability. Use of a suitable control strategy and footswitch ensures that requirements 4, 5, and 6 are satisfied.

A 2-DOF manipulator has been designed based on the worm-driven revolute joint. A pointer mechanism in the control handle indicates the direction in which the user should apply a force and enhances the ease of use of the manipulator and safety of the operator. It was also shown that the location of the joint torque sensors (strain gauge modules) decreases the complexity of the mathematical model by explicitly measuring the Coriolis and Centripetal forces.

A mathematical model of the 2-DOF system was developed so that a computed-torque control algorithm could be designed to regulate joint position for tracking of a pre-defined path under direct control of the operator. Experimental results have shown that the computed torque control scheme was able to track the desired path to within ± 3 mm. Although this error is quite large, x-axis tracking was shown to be under 0.5 mm, most of the error appeared in y-axis tracking. It was shown that the robot arm

was in a configuration that yielded poor system stiffness in the y-direction. Hence, this tracking error has been attributed to the effects of the un-modelled joint flexibility and discontinuity in the velocity command generated by the control strategy. A second control scheme, namely a composite adaptive controller, has also been designed using the mathematical model. Experimental results have shown that the algorithm is not capable of tracking the desired path with accuracy similar to that of the computed-torque control scheme. Again performance of the controller was adversely affected by the lack of system stiffness x-axis error was shown to be under 2mm while the y-axis error was recorded at over 10mm. The computed torque controller has been shown to track the desired path in the face of disturbances from friction, user-input force and joint flexibility. However, both controllers have been adversely affected by changes in direction of user-input force. Although the computed-torque controller was shown to track the desired path, the transfer of the control strategy from 1-DOF to 2-DOF has been unsuccessful.

Despite the differences in the performance of the controllers for the 2-DOF, the 1-DOF control strategy was shown to work well. The composite adaptive scheme has two distinct advantages over the computed-torque method. Firstly, by performing a short test procedure before each use, the adaptive controller can be re-tuned (calibrated) for the current mechanical condition of the manipulator. That is, by systematically training of the control algorithm the control can compensate for the slow changes in system parameters associated with wear at the gear interface and the within the bearing of the mechanism. The computed-torque control algorithm on the other hand would require periodic re-tuning by a skilled control engineer in order to retain tracking performance. The second potential advantage of the composite adaptive control scheme presented above is that of safety. Despite the changes in system parameters (possibly large changes in friction coefficients due to temperature and wear) the stability of the composite adaptive controller is mathematically guaranteed. The stability of the computed-torque algorithm under such conditions is undefined and the tracking performance is likely to be reduced due to large variations in system parameters.

For robotic manipulators in general, errors in the order of 3mm are quite large. With this performance the manipulator is not suitable for use in medical applications. The stiffness of the joint mechanism has to be significantly increased in order to improve performance of both the computed-torque and adaptive controllers. However, in comparison to the prototype of PADyC, which exhibited errors 20mm for 20N of input force, the worm-driven manipulator performs well. The addition of a pointer mechanism inside the control handle also improves the usability of system, i.e. the operator can concentrate on the task in hand without having to mentally perform co-ordinate transformations and large changes in visual focus. This also reduces the need for training satisfying requirement 7.

The worm-driven manipulator should be classified as an active device. Under certain erroneous conditions caused by possible electronic hardware failure, it is entirely possible that the motors used in the mechanism could generate motion against the user. However, the over-actuated nature of the joint coupled with the low power of the motors minimises the ability to do harm should both motors fail in opposite directions. The operator has plenty of time to react, i.e. by releasing the foot-switch and disabling power to the motors.

Significant steps have been made towards developing a safe robotic device for direct human interaction. A 1-DOF robotic manipulator was shown to work well, but the transfer of the control strategy to 2-DOF has not been hugely successful. It has been shown that a planar revolute joint manipulator can be smoothly controlled to follow a pre-defined path under the direct control of a human operator. However, tracking performance is degraded in the critical regions where the measured joint torque changes sign. The derivation of the requirements for the control of a revolute joint serial robot (presented in Chapter 3) has played a large role in the advancement of a control strategy for this genre of device. Potentially, a motorised joint is not required and by designing a clutching mechanism that would enable control of the braking force system safety could be further enhanced.

12.3. Publications

As part of this research two journal papers (included in Appendix H) have been accepted for publication. Two other papers have also been submitted to reputable journals

12.3.1. Published

Reedman. A. V. C., and Bouazza-Marouf, K., Control of an Actively Constrained Robotic Joint for Passive Deployment Applications, *Proceedings of the Institution of Mechanical Engineers, Part K Journal of Multibody Dynamics*, **215**, 2001, pp187-197

Reedman. A. V. C., and Bouazza-Marouf, K., Composite Adaptive Control of a Robotic Joint for Passive Deployment Applications, *Proceedings of the Institution of Mechanical Engineers, Part I Journal of Control and Systems Engineering*, **216**, 2002.

12.3.2. Submissions

Reedman, A. V. C. and Bouazza-Marouf, K., Limitations on the deployment of Passive Arms with Dynamic Constraints, *Proceedings of the Institution of Mechanical Engineers, Part K Journal of Multi-body Dynamics*, Submitted Feb 2002

Reedman, A. V. C. and Bouazza-Marouf, K., An Actively Constrained Two Degree-of-Freedom Manipulator for Passive Deployment Applications, *International Journal of Control*, Submitted in Mar 2002.

Chapter 13 Recommendations for Further Work

13.1. Introduction

It has been shown that the robot mechanism and the control hardware/software perform well as a prototype system. However, there are certain enhancements that are required in order to help improve performance and further this research. This chapter describes certain enhancements to the system that would help to drive this work forward in order to realise a robotic device that could be used in a surgical environment.

13.2. Control Development Studio

Further work in the development of the CDS should involve the addition of project handling capability. The current method involves the use of automatically generated *make* and *link* files and requires some explicit knowledge of the Watcom C/C++ package in order to add new files to a control program's build list. Work should also include development of the control program to run independently of the CDS, i.e. as a standalone program if the user so desires.

13.3. Single Axis Motion Control Interface

The Single Axis Motion Control Interface has been proved to be a useful asset to this work. The capability of this to control a variety of motors coupled with the digital and analogue I/O that is available makes this interface incredibly flexible. However, in its current configuration the card uses the EISA bus, which is now being phased out of PC architecture. In order to keep up with current developments in the computer industry it is recommended that the card be redesigned for the PCI bus so that the speed of data acquisition could be increased.

13.4. Mechanical System

13.4.1. Joint Configuration

Due to the physical limits on the rotation of each joint and the way in which the manipulator has been designed, the work area is limited to the area shown in Figure 84a. The shaded area represents the working region of the end-effector based on the fact that the motion of each joint is limited by $\theta_1, \theta_2 \in (-70^\circ, 70^\circ)$. It can be seen that the workspace of the robot is very limited. Therefore, by rotating the second link through an angle of 90° it is possible to generate the workspace given by Figure 84b. This configuration also has the added advantage that the workspace singularities associated with the straight arm configuration of Figure 84a cannot exist in the bent-arm configuration.

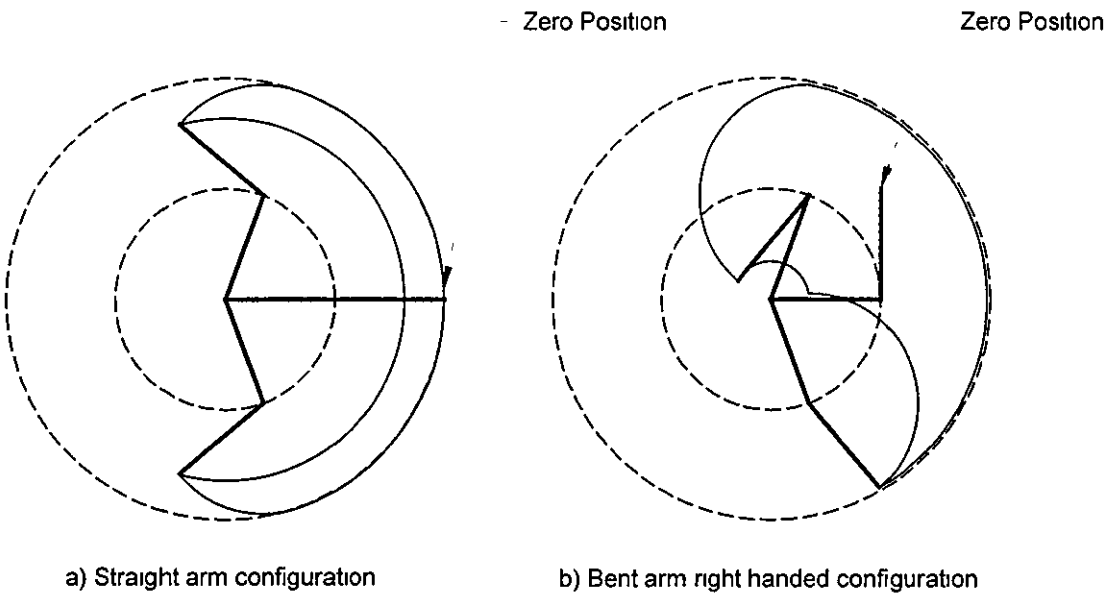


Figure 84 - Configurations of a 2-DOF Robot and workspace

The results of Chapter 3 also indicate that a better choice of robot configuration could be used. In particular, the use of combinations of revolute and prismatic joints, such as in a polar configuration are advantageous because the joint can be directly controlled in the direction of the force without the coupling associated with the type of device presented in this work. A second option would be to use a Cartesian

configuration which has the advantage that end-effector errors induced by joint errors are consistent over the whole workspace. However, the Cartesian manipulator has the disadvantage that the physical dimensions of manipulator are generally larger than its working volume

However, it is felt that a redesign of the joint mechanism coupled with the control strategies presented in this work, the end-effector position error of the 2-DOF revolute joint planar manipulator could be controlled to be less than 0.5mm in magnitude

A second point of interest is that the 2-DOF manipulator and control algorithms presented in this work are only capable of operating safely and successfully in a plane perpendicular to gravity. In many real-world applications it may not always be possible to operate the robot under these conditions. For this revolute-joint manipulator a gravity compensating controller or counter-balancing mechanism are two possible solutions. However, owing to the need for more powerful actuators for gravity compensation controllers and in light of the requirements set out above, the *passive* counter-balance method would be preferred in safety critical applications. Counter balancing a revolute-joint robot with even as few as 2-DOF is not a simple task. Therefore, a robot configuration with intersecting joint axes (i.e. Cartesian or polar configurations) might be more appropriate for the counter balancing approach.

13.4.2 Joint Actuators

One of the fundamental advances of this research has been the development of a control strategy and pointer mechanism that ensures the mechanism cannot drive against (and harm) the operator. With this in mind, it is clear to see that if a clutch/brake system could be developed that would allow the application of a known braking force, the safety of the operator could be further enhanced by completely removing the motors from the system. A detailed study of braking mechanisms should be undertaken to effect this approach.

13.4.3. Joint Mechanism

During the development of the mathematical model for the joint mechanism several critical assumptions have been made. Firstly, it was assumed that the three gears could be modelled as blocks sliding against each other and that the contact between the gears is uniform. The second assumption was made that the worm-wheel never loses contact with either of the two worms. However, in reality contact between the worm and worm-wheel is not uniform and acceleration of the worm with respect to the worm-wheel can be quite large causing loss of contact between the gears. The effect of this on the overall system needs to be investigated.

13.4.4. Control Strategy

The control strategy for the 2-DOF system is flawed in the region where the measured joint torque changes sign. Work to increase system stiffness and to develop more robust strategies than those presented in this work should be continued.

References

- [1] Surgery of the knee, 2nd Ed , J N Insall, R E Windsor, W N Scott, M A Kelly, P. Aglietti (Eds), Churchill Livingstone, 1993
- [2] Surgical Technique for use with P F C.[®] Sigma Knee Systems, Johnson & Johnson Orthopaedic, 1998
- [3] Marcacci, M., Dario, P., Fadda, M., Marcenaro, G., and Martelli, S., Computer-Assisted Knee Arthroplasty, In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), *Computer-Integrated Surgery*, MIT Press, 1996, pp717-423
- [4] Safety in the use of industrial robots, *Occupational Safety and Health Series*, No 60, 1989, International Labour Office, Geneva, pp49-65
- [5] Taylor, R. H., Paul, H. A., Kazanzides, P., Mittlestadt, B. D., Hanson, W., Zuhars, J., Williamson, W., Musits, B., Glassman, E., and Bargar, W. L., Taming the Bull: Safety in a Precise Surgical Robot, *Proceedings of the 1991 International Conference on Advanced Robotics*, Pisa, Italy, June.
- [6] Cain, P., Kazanzides, P., Zuhars, J., Mittlestadt, B., and Paul, H., Safety Considerations in a Surgical Robot, *Biomedical Sciences Instrumentation*, Vol 29, 1993, pp291-294
- [7] B. L. Davies, A Discussion of Safety Issues for Medical Robots, In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), *Computer-Integrated Surgery*, MIT Press, 1996, pp287-296.

- [8] **B. L. Davies**, Safety of medical robots, *Safety Critical Systems*, Chapman Hall Press, Part 4, Chapter 15, pp193-201, 1993
- [9] **R. G. Epstein**, The Case of the Killer Robot, (John Wiley & Sons), 1997.
- [10] The Medical Devices Directive, *Directive # 93/42/EEC*
- [11] **Mohsen, A. M. M. A., Serman, K. P., Cain, T. J., Karpinski, M. R. K., Howell, F. R., Phillips, R., Viant, W. J., Giffiths, J. G., and Dyer, K. D. F.**, End User Issues for Computer Assisted Surgical Systems, *Transactions of the Institution of Measurement and Control*, **17**(5), 1995, pp265-271
- [12] **Cinquin, P., et al**, Computer Assisted Medical Interventions, *IEEE Engineering in Medicine and Biology*, **14**(3), May/June, 1995.
- [13] **DiGioia, A. M., Jaramaz, B., and Colgan, B. D.**, Computer Assisted Orthopaedic Surgery – Image Guided and Robotic Assistive Technologies, *Clinical Orthopaedics and Related Research*, No 354, September, 1998, pp8-16
- [14] **Davies, B. L.**, A review of robotics in surgery, *Proceedings of the Institution of Mechanical Engineers Part H Journal of Engineering in Medicine*, Vol 214, No. H1, 2000.
- [15] **Simon, D. A., Jaramaz, B., Blackwell, M., Morgan, F., DiGioia, A. M., Kischell, E., Colgan, B., and Kanade, T.**, Development and Validation of a Navigational Guidance System for Acetabular Implant Placement, *Lecture Notes in Computer Science*, 1997, **1205**, pp583-592

- [16] Delp, S. L., Stulberg, S. D., Davies, B. L., Picard, F., and Leitner, F., Computer Assisted Knee Replacement, *Clinical Orthopaedics and Related Research*, No 354, September, 1998, pp49-56
- [17] Leitner, F., Picard, F., Minfelde, R., Shultz, H-J., Cinquin, P., and Saragaglia, D., Computer-Assisted Knee Surgical Total Replacement, *Lecture Notes in Computer Science*, 1997, **1205**, pp629-638.
- [18] Tonetti, J., Carrat, L., Lavallée, S., Pittet, L., Merloz, P., and Chirossel, J., Percutaneous Iliosacral Screw Placement Using Image Guided Techniques, *Clinical Orthopaedics and Related Research*, No 354, September, 1998, pp103-110
- [19] Langlotz, F., Bächler, R., Berlemann, U., Nolte, L., and Ganz, R., Computer Assistance for Pelvic Osteotomies, *Clinical Orthopaedics and Related Research*, No 354, September, 1998, pp92-102
- [20] Blackwell, M., Morgan, F., and DiGioia III, M., Augmented Reality and Its Future in Orthopaedics, *Clinical Orthopaedics and Related Research*, No. 354, September, 1998, pp111-122
- [21] Troccaz, J., and Lavallée, S., An alternative to actuated robots and passive arms in medical robotics, *Engineering in Medicine and Biology 15th Annual Conference*, San Diego, 1993.
- [22] Torccaz, J., Lavallée, S., and Hellion, E., A Passive Arm with Dynamic Constraints A Solution to Safety Problems in Medical Robots?, *IEEE International Conference on Systems, Man, and Cybernetics*, 1993, pp166-171

- [23] Lavallée, S., and Troccaz, J., Robot de guidage ge gestes et procédé de commande, *European Patent, Pat # EP00574330A1*, 15 December 1993
- [24] Lavallée, S , and Troccaz, J., Robot for guiding movements and control method thereof, *United States Patent, Pat # US005399951A*, 21 March 1995
- [25] Troccaz, J., Roue libre débrayable, *European Patent, Pat # EP00658697A1*, 21 June 1995.
- [26] Troccaz, J., Disengageable free wheel, *United States Patent, Pat # US005529159A*, 25 June 1996.
- [27] Delnondedieu, Y., and Troccaz, J., PADyC A Passive Arm with Dynamic Constraints. A prototype with two degrees of freedom, *Proceedings of the Second Annual International Symposium on Medical Robotics and Computer Assisted Surgery*, Baltimore, Maryland, USA, November 4-7, 1995
- [28] Troccaz, J., and Delnondedieu, Y., Semi-Active Guiding Systems in Surgery A Two DOF Prototype of the Passive Arm with Dynamic Constraints (PADyC), *Mechatronics*, Vol. 6, No. 4, pp399-421, 1996
- [29] Schneider, O., Troccaz, J., Chavanon, O., and Blin, D., Synergistic Robotic Assistance to Cardiac Puntures, *Computer Assisted Radiology and Surgery 13th International Symposium and Exhibition (CARS'99)*, Paris, France, June 23-26, 1999.
- [30] Schneider, O., Troccaz, J., Chavanon, O., and Blin, D., PADyC A Synergistic Robot for Cardiac Puncturing, *Proceedings of the IEE International Conference on Robotics and Automation*, 3, 2000, pp2883-2888.

- [31] Phillips, R., Viant W. J., Mohsen, A. M. M. A., Griffiths, J. G., Bell, M. A., Bell, T. J., Cain, T. J., Sherman, K. P., and Karpinski, M. R., Image guided orthopaedic surgery design and analysis, *Transactions of the Institute of Measurement and Control*, Vol. 17, No 5, pp251-264, 1995
- [32] Viant, W. J., Phillips, R., Griffiths, J. G., Ozanian, T. O., Mohsen, A. M. M. A., Cain, T. J., Karpinski, M. R. K., and Sherman, K. P., A computer assisted orthopaedic surgical system for distal locking of intramedullary nails, *Proceedings of the Institute of Mechanical Engineering*, Part H, Vol. 211, No. H4, 1997
- [33] Taylor, R. H., Paul, H. A., Mittlestadt, B. D., Hanson, W., Kazanzides, P., Zuhars, J., Glassman, E., Musits, B. L., Williamson, W., and Bargar, W. L., An Image-Directed Robotic System for Precise Orthopaedic Surgery, *Images of the twenty-first Century – Proceeding of the 12th Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, Philadelphia, Pennsylvania, November 1-4, 1990, Vol. 12, No 5, pp1933-1935
- [34] Mittlestadt, B., Paul, H. A., Kazanzides, P., Zuhars, J., Williamson, W., Pettitt, R., Cain, P., Kloth, D., Rose, L., and Musits, B. L., Development of a surgical robot for cementless total hip replacement, *Robotica*, Vol 11, Pt 6, Nov./Dec., 1993, pp553-560
- [35] Kazanzides, P., Mittlestadt, B. D., Musits, B. L., Bargar, W. L., Zuhars, J. F., Williamson, W., Cain, P. W., and Carbone, E. J., An Integrated System for Cementless Hip Replacement, *IEEE Engineering in Medicine and Biology*, Vol. 14, No 3, May/June, 1995, pp307-313.

- [36] Taylor, R. H., Mittlestadt, B. D., Paul, H. A., Hanson, W., Kazanzides, P., Zuhars, J. F., Williamson, W., Musits, B. L., Glassman, E., and Bargar, W. L., An Image-Directed Robotic System for Precise Orthopaedic Surgery, *In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), Computer-Integrated Surgery*, MIT Press, 1996, pp379-396
- [37] Mittlestadt, B. D., Kazanzides, P., Zuhars, J. F., Williamson, W., Cain, P., Smith, F., and Bargar, W. L., The Evolution of a Surgical Robot from Prototype to Human Clinical Use, *In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), Computer-Integrated Surgery*, MIT Press, 1996, pp397-407.
- [38] Bargar, W. L., Bauer, A., and Börner, M., Primary and Revision Total Hip Replacement Using the Robodoc[®] System, *Clinical Orthopaedics and Related Research*, No 354, September, 1998, pp82-91
- [39] Davies, B. L., Ng, W. S., and Hibberd, R. D., Prostatic resection: an example of safe robotic surgery, *Robotica*, Vol 11, 1993, Cambridge University Press, pp561-566
- [40] Davies, B. L., Hibberd, R. D., Timoney, A. G., and Wickham, J. E. A., A Clinically Applied Robot for Prostatectomies, *In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), Computer-Integrated Surgery*, MIT Press, 1995, pp593-601
- [41] Wickham, J. E. A., Davies, B. L., Hibberd, R. D., and Timoney, A. G., Frame for medical implement, *European Patent, Pat # EP00416863A1*, 13 March 1991
- [42] CASPAR. Computer-Assisted Planning and Robotic System for Orthopaedic Surgery, <http://www.orthomaquet.de/eseiten/index.htm> Accessed 3rd October 2001

- [43] First results of CASPAR (Computer Assisted Surgical Planning and Robotics) TKR, http://www.orthomaquet.de/eseiten/first_results_of_CASPAR_TKR.htm
- [44] Davies, B. L., Fan, K. L., Hibberd, R. D., Jakopec, M., and Harris, S. J., A Mechatronic Based System for Knee Surgery, *International Conference on Intelligent Information Systems*, Grand Bahama Island, The Bahamas, December 1997, pp48-52
- [45] Ho, S.C., Hibberd, R. D., and Davies, B. L., Robot Assisted Knee Surgery – Establishing a force control strategy incorporating active motion constraint, *IEEE Engineering in Medicine and Biology*, Vol 14, No 3, May/June, 1995, pp292-300
- [46] Davies, B. L., Harris, S. J., Lin, W. J., Hibberd, R. D., Middleton, R. Middleton, and Cobb, J. C., Active compliance in robotic surgery – the use of force control as a dynamic constraint, *Proceedings of the Institute of Mechanical Engineers*, Part H: Journal of Engineering in Medicine, 1997, 211(4), pp285-292.
- [47] Harris, S. J., Lin, W. J., Fan, K. L., Hibberd, R. D., Cobb, J., Middleton, R., and Davies, B. L., Experiences with Robotic Systems for Knee Surgery, *Lecture Notes in Computer Science*, 1997, 1205, pp757-766
- [48] Kienzle III, T. C., Stulberg, S. D., Peshkin, M. A., Quaid, A., Lea, J., Goswami, A., and Wu, C. H., Total Knee Replacement, *IEEE Engineering in Biology and Medicine*, Vol. 14, No 3, May/June, 1995
- [49] Kienzle III, T. C. , Stulberg, S. D., Peshkin, M., Quaid, A., Lea, J., Goswami, A., and C. H. Wu, A Computer-Assisted Total Knee Replacement Surgical System Using a Calibrated Robot, *In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), Computer-Integrated Surgery*, MIT Press, 1996, pp409-416

- [50] Burckhardt, C. W. , Flury, P., and Glauser, D., Stereotactic Brain Surgery, *IEEE Engineering in Medicine and Biology*, Vol. 14, No. 3, 1995, pp314-317.
- [51] Glauser, D., Frankhauser, H., Epitoux, M., Hefti, J. L., and Jaccottet, A., Neurosurgical Robot Minerva. First Results and Current Developments, *Proceedings of the Second Annual International Symposium on Medical Robotics and Computer Assisted Surgery*, Baltimore, Maryland, USA, November 4-7, 1995, pp24-30
- [52] Intuitive Surgical Systems website, <http://www.robodoc.com/>, Accessed 3rd October 2001
- [53] Brant, G., Radermacher, K., Lavallée, S., Staudt, H. W., and Rau, G., A Compact Robot For Image Guided Orthopaedic Surgery: Concept and Preliminary Results, *Lecture Notes in Computer Science*, 1997, 1205, pp767-776.
- [54] Brett, P. N., Fraser, C. A., Hennigan, M., Griffiths, M. V., and Kamel, Y., Automatic Surgical Tools for Penetrating Flexible Tissues, *IEEE Engineering in Medicine and Biology*, Vol 14, No 3, May/June, 1995
- [55] Brett, P. N., Baker, D. A., Reyes, L., and Blanshard, J., An automatic technique for micro-drilling a stapedotomy in the flexible stapes footplate, *Proceedings of the Institute of Mechanical Engineers Part H Journal of Engineering in Medicine*, Vol. 209, No H4, 1995, pp255-262.
- [56] Baker, D. A., Brett, P. N., and Griffiths, M. V., Application Considerations for the Implementation of an Automated Stapedotomy Drill, *Proceedings of the 29th International Symposium on Robotics*, Birmingham, UK 27th April – 1st May 1998.

- [57] Baker, D. A., Brett, P. N., Griffiths, M. V., and Reyes, L., A Mechatronic Drilling Tool for Ear Surgery. A Case Study of Some Design Characteristics, *Mechatronics*, Vol. 6, No. 4, pp461-477, 1996
- [58] Dobson, R., 'Intelligent' Drill Stops at Points of Danger, *The Sunday Times*, 24 September, 1995
- [59] Soianovici, D., Whitcomb, L. L., Anderson, J. H., Taylor, R. H., and Kavoussi, L. R., A Modular Surgical Robotic System for image Guided Percutaneous Procedures, *Lecture Notes in Computer Science*, 1496, pp404-410
- [60] Cleary, K., Stoianovici, D., Glossop, N., Gary, K., Onda, S., Cody, R., Lindisch, D., Stanimir, A., Mazilu, D., Patriciu, A., Watson, V., and Levy, E., CT_Directed Robotic Biopsy Testbed: Motivation and Concept, *Proceedings of the SPIE - International Society for Optical Engineering*, 4319, 2001, pp231-236.
- [61] Stoianovici, D., Cadeddu, J. A., Demaree, R. D., Basile, S. A., Taylor, R. H., Whitcomb, L. L., and Kavoussi, L. R., A Novel Mechanical Transmission Applied to Percutaneous Renal Access, *Proceedings of the ASME Dynamics and Control Division*, 61, 1997, pp401-406.
- [62] Bouazza-Marouf, K., Browbank, I., and Hewitt, J. R., Robtic-assisted internal fixation of femoral fractures, *Proceedings of the Institute of Mechanical Engineers Part H Journal of Engineering in Medicine*, Vol 209, No. H1, 1995, pp51-58
- [63] Bouazza-Marouf, K., Browbank, I., and Hewitt, J. R., Robot-Assisted Invasive Orthopaedic Surgery, *Mechatronics*, Vol 6, No 4, pp381-397, 1996

- [64] **Finlay, P. A.**, ORTHOSISTA An Active Surgical Localiser for Assisting Orthopaedic Fracture Fixation, *Proceedings of the Second Annual International Symposium on Medical Robotics and Computer Assisted Surgery (MRCAS'95)*, Baltimore, Maryland, 4-7 November, 1995, pp203-207
- [65] **Phillips, R., Hafez, M. A., Mohsen, A. M. M. A., Sherman, K., Hewitt, J. R., Browbank, I., and Bouazza-Marouf, K.**, Computer and Robotic Assisted Osteotomy around the Knee, *The 8th Annual "Medicine Meets Virtual Reality" Conference, MMVR 2000*, 27-30 January, 2000, Newport Beach, CA.
- [66] **Lueth, T. C., Hein, A., Albrecht, Demirtas, M., Zachow, S., Heissler, E., Klein, M., Menneking, H., Hommel, G., and Bier, J.**, A Surgical Robot System for Maxillofacial Surgery, *IEEE International Conference on Industrial Electronics, Control, and Instrumentation (IECON)*, Aachen, Germany, 1998, pp2470-2475.
- [67] **Sackier, J. M., and Wang, Y.**, Robotically Assisted Laparoscopic Surgery From Concept to Development, *In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), Computer-Integrated Surgery*, MIT Press, 1995, pp577-580.
- [68] **Taylor, R. H., Funda, J., Eldridge, B, Larose, D., Gomory, S., Gruben, K., Talamini, M., Kavoussi, L., and Anderson, J.**, A Telerobotic Assistant for Laparoscopic Surgery, *In R H Taylor, S Lavallée, G C Burdea & R S Mosges (Eds), Computer-Integrated Surgery*, MIT Press, 1995, pp581-592.
- [69] **Armstrong HealthCare**, <http://www.armstrongbg.co.uk/>, Accessed 8th October 2001.
- [70] **Endoxirob**, <http://www.endoxirob.com/>, Accessed 10th October 2001.

- [71] **ZEUS Robotic Surgical System**, <http://www.computermotion.com/zeus.html>. Accessed 3rd October 2001
- [72] **J. K. Sainsbury, Jr.**, The heart of microsurgery, *Mechanical Engineering*, December, 1998
- [73] **Roderick, D.**, Doctor's little helper, *Time Magazine*, June 4, 2001.
- [74] **Crompton, S.**, In a Safe Pair of Metal Hands, *The Times (Times 2)*, 24 April 2001, pp12-13.
- [75] **Smith, N., Betemps, M., Jutard, A., San Felix, A., and Dittmar, A.**, Penetrating Keratoplasty: A Robotised Cut of the Cornea, *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, **2**, Kyongju, Korea, 1999, pp740-745.
- [76] **HIPPOCRATE Robot**, <http://www.lirmm.fr/~duchemin/Hippo.htm>, Accessed 10th October 2001.
- [77] **Armstrong HealthCare**, PathFinder Precision Localiser for Neurosurgery, <http://www.armstrong.co.uk/>, Accessed 8th October 2001
- [78] **Wilks, N.**, Robots on the Brain, *Professional Engineering*, **14**(17), Professional Engineering Publishing, 19 September 2001
- [79] **Book, W., Charles, R., Davis, R., and Gomes, M.**, The Concept and Implementation of a Passive Trajectory Enhancing Robot, *Proceedings of the ASME Dynamic Systems and Control Division*, Vol 58, 1996, pp633-638.

- [80] Love, L. J., and Book, W., Design and Control of a Multi-Degree-of-Freedom Haptic Interface, *Proceedings of the 1994 International Mechanical Engineering Congress and Exposition*, 2, 6-11 November, Chicago, IL, pp851-856
- [81] Tognetti, L. J., Actuator Design for a Passive Haptic Display, *MS Thesis Georgia Institute of Technology*, 1999
- [82] Colgate, J. E., Peshkin, M. A., and Wannasuphoprasit, W., Nonholonomic Haptic Display, *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Vol 1, pp539-544.
- [83] Colgate, J. E., Wannasuphoprasit, W., and Peshkin, M. A., Cobots; Robots for Collaboration with Human Operators, *Proceedings of the International Mechanical Engineering Congress and Exhibition*, Atlanta, GA, DSC-Vol 58, pp433-439, 1996
- [84] Akella, P., Peshkin, M., Colgate, E., Wannasuphoprasit, W., Nagesh, N., Wells, J., Holland, S., Pearson, T., and Peacock, B., Cobots for the automobile assembly line, *IEEE International Conference on Robotics and Automation*, 1999.
- [85] Moore, C. A., Continuously Variable Transmission for Serial Link Cobot Architectures, *M S Thesis Northwestern University*, 1997.
- [86] Moore, C. A., Peshkin, M. A., and Colgate, J. E., A Three Revolute Cobot Using CVTs in Parallel, *Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE)*, Nov. 14-19, Nashville, TN, 1999.
- [87] Book, W., Charles, R., Davis H., and Gomes, M., The Concept and Implementation of a Passive Trajectory Enhancing Robot, *Proceedings of the ASME Dynamics and Controls Division*, 58, 1996, pp633-638.

- [88] Davis, H., and Book, W., Torque Control of a Redundantly Actuated Passive Manipulator, *Proceedings of the American Control Conference*, Albuquerque, New Mexico, June 1997, pp959-963
- [89] Akella, P., Peshkin M., and Colgate, J. E., Cobots. A Novel Material Handling Technology, *Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE)*, Nov 15-20, Anaheim, CA, 1998
- [90] Peshkin, M., Colgate, J. E., Moore, C., Passive Robots and Haptic Displays Based on Nonholonomic Elements, *Proceedings of the IEEE International Conference on Robotics and Automation*, Philadelphia, 1996
- [91] Meriam, J. L., and Kraig, L.G., *Engineering Mechanics, Volume 1 Statics*, 2nd Ed , John Wiley and Sons, New York, 1987, pp318-319.
- [92] Patton, W. J., *Mechanical Power Transmission*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980
- [93] MINIMITOR Catalogue, MiniMotor SA, 6980 Croglio, Switzerland, <http://www.minimotor.ch/>.
- [94] Yager, T., The QNX Operating System, *Byte*, 15(8), August 1990, pp281-283.
- [95] Agilent Technologies, General Purpose Motion Control ICs: HCTL-1100 Series, <http://www.semiconductor.agilent.com/cgi-bin/morpheus/home/home.jsp?pSection=Motion+Control>, Accessed 9th October 2001.
- [96] Bo, L. C., and Pavelescu, D., The Friction-Speed Relationship and its Influence on the Critical Velocity of Stick-Slip Motion, *Wear*, 82, 1982, pp277-289.

- [97] **Armstrong-Hélouvry, B., Dupont, P., and Canudas De Wit, C.**, A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction, *Automatica*, **30**(7), 1994, pp1083-1138.
- [98] **Craig, J. J., Hsu, P., and Sastry, S. S.**, *Adaptive Control of Mechanical Manipulators*, International Journal of Robotics Research, **6**(2), 1987, pp16-28.
- [99] **Psaltis, D., Sideris, A., and Yamamura, A.**, Multilayered Neural Netowrk Controller, *IEEE Control Systems Magazine*, April 1988, **8**(2), pp17-21
- [100] **Ozaki, T., Furuhashi, T., Okuma, S., and Uchikawa, Y.**, Trajectory Control of Robotic Manipulators Using Neural Networks, *IEEE Trans on Ind Elec* , June 1991, **38**(3), pp195-202.
- [101] **Safaric, R., and Jezernik, K.**, Trajectory Tracking Neural Network Controller for a Robotic Mechanism and Lyapunov Theory of Stability, *Proc Intelligent Robots and Systems Conf IROS'94*, Munich, 1994.
- [102] **Kwan, C., Lewis, F. L., and Dawson, D. M.**, Robust Neural-Network Control of Rigid-Link Electrically Driven Robots, *IEEE Trans on Neural Networks*, July 1998, **9**(4), pp581-588.
- [103] **Slotine, J. J., and Li, W.**, Adaptive Robot Control: A New Perspective, *Proceedings of the IEEE 28th International Conference on Decision and Control*, Los Angeles, CA, December 1987, pp192-198.
- [104] **Lewis, F. L., Abdallah, C. T., and Dawson, D. M.**, *Control of Robot Manipulators*, MacMillan Publishing Company, New York, 1993

- [105] Li, W., and Slotine, J. J., Parameter Estimation Strategies for Robotic Applications, *ASME Winter Annual Meeting – Dynamic Systems and Control Division*, 1987, pp213-218
- [106] Wang, H., Liu, G. P., Harris, C. J., and Brown, M., *Advanced Adaptive Control*, Pergammon Press, 1995, pp24.
- [107] Schilling, R. J., *Fundamentals of Robotics Analysis and Control*, Prentice Hall, Englewood Cliffs, New Jersey, 1990
- [108] Chen, C-T., *Analogue and Digital control system design Transfer function, state-space and algebraic methods*, Saunders College Publishing, 1993, pp70-81.

Single Axis Motion Control Interface Card

for real-time data capture and motion control

Table of Contents

| | | |
|------------------|--|------------|
| CHAPTER 1 | INTRODUCTION | 192 |
| | GENERAL DESCRIPTION | 192 |
| | DETAILS OF THIS DOCUMENT | 192 |
| CHAPTER 2 | HARDWARE | 193 |
| | ELECTRICAL SPECIFICATIONS | 193 |
| | HARDWARE DESCRIPTION | 193 |
| | 44 WAY D TYPE CONNECTOR | 195 |
| | SETTING THE BASE ADDRESS | 195 |
| | JUMPER SETTINGS | 196 |
| | HCTL-1100 REGISTER MAP | 196 |
| | <i>Using the TEST EXE Program (MS-DOS Version)</i> | 197 |
| | <i>Using the TEST EXE Program (Windows 9x Version)</i> | 198 |
| CHAPTER 3 | SOFTWARE | 200 |
| | C LIBRARY | 200 |
| | <i>Initialisation Functions</i> | 200 |
| | <i>Digital Input/Output Functions</i> | 200 |
| | <i>Analogue Input/Output Functions</i> | 201 |
| | <i>Low Level HCTL-1100 Control Functions</i> | 202 |
| | <i>High Level HCTL-1100 Control Functions</i> | 203 |
| | <i>HCTL-1100 Control Mode Functions</i> | 206 |
| | C++ CLASS LIBRARY | 207 |
| | <i>Constructor</i> | 208 |
| | <i>Initialisation Functions</i> | 208 |
| | <i>Digital Input/Output Functions</i> | 209 |
| | <i>Analogue Input/Output Functions</i> | 210 |
| | <i>Low Level HCTL-1100 Control Functions</i> | 211 |
| | <i>High Level HCTL-1100 Control Functions</i> | 211 |
| | <i>HCTL-1100 Control Mode Functions</i> | 219 |
| | <i>C++ Code Sample</i> | 221 |
| | PCB PRODUCTION ARTWORK | 222 |
| | BUILDING AND TESTING THE INTERFACE CARD | 222 |
| | <i>Address Decoding Circuit</i> | 222 |
| | <i>Digital I/O Circuit</i> | 224 |
| | <i>Digital-To-Analogue Converter Circuit</i> | 225 |
| | <i>Analogue-to-Digital Converter Circuit</i> | 226 |
| | <i>HCTL-1100 Circuit</i> | 227 |
| | FULL COMPONENT LIST | 227 |
| | COMPONENT LAYOUT DIAGRAM | 228 |

Chapter

1

Introduction

General Description

This single-axis motion control EISA interface card provides 8 channel, 12-bit analogue inputs, 1 channel, 12-bit analogue output, 8 digital inputs, 8 digital outputs, and an HCTL-1100 interface for the control of brushed DC motors and 3 or 4 phase stepper/brush-less DC motors. The board can be used on any IBM compatible machine with an EISA bus.

Details of this document

PCB artwork and details of how to construct the single-axis controller (including a component list and layout diagram) are included as part of the document. The document also details the software that has been written to provide fast prototyping of control systems. The software consists of a C library, a C++ class library and example source code. The code has been developed in C++ for both the Windows 9x and QNX 4 xx operating systems. Example system diagrams and application notes have been included to aid fast prototyping of control systems.

Chapter
2

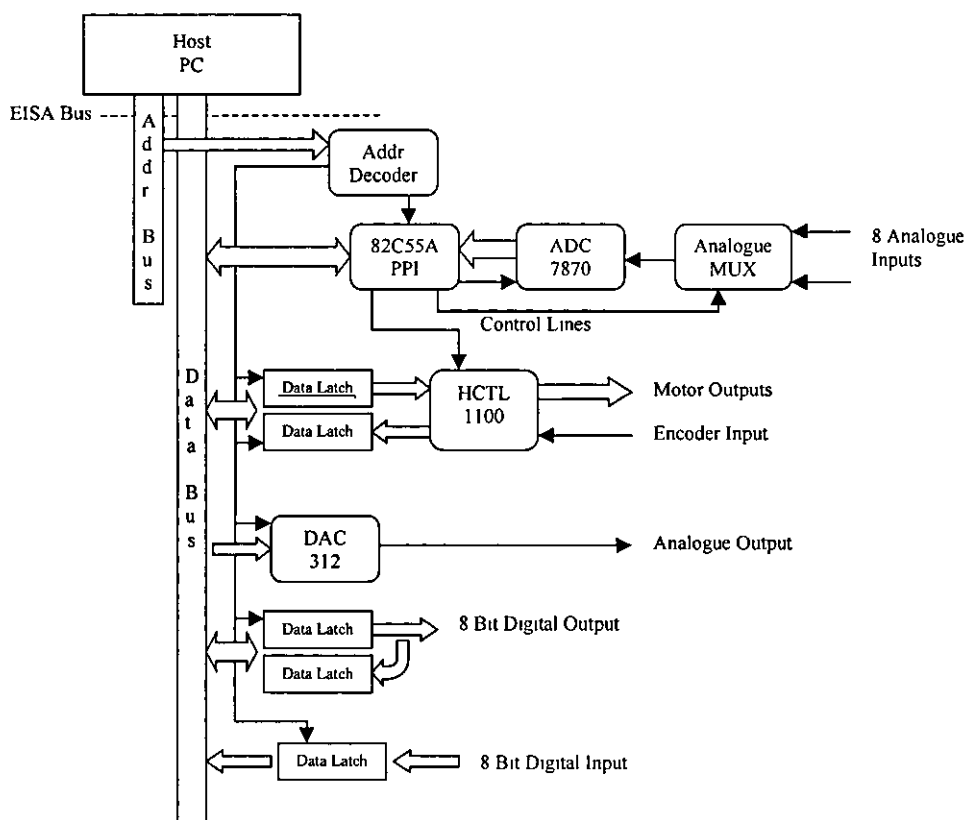
Hardware

Electrical Specifications

| Signal Description | Signal Name | Electrical Specification |
|------------------------------------|----------------------|--------------------------|
| Digital Output | DOUT0-DOUT7 | TTL Logic Levels |
| Digital Input | DIN0-DIN7 | TTL Logic Levels |
| Analogue Output | DAC1, HCTL DAC | ± 10v |
| Analogue Input | AIN0-AIN7 | ± 3v |
| Encoder Inputs | CHA, CHB, /INDEX | TTL Logic Levels |
| HCTL PWM and Stepper Motor Outputs | PHA-PHD, PULSE, SIGN | TTL Logic Levels |
| HCTL Control Inputs | /STOP, /LIMIT | TTL Logic Levels |

Hardware Description

The diagram of Figure 1 below best describes the system. The 82C55A Programmable Interface controls the timing of read and write operations from both the AD7870 analogue-to-digital converter and the HCTL-1100 motion control interface. The digital inputs and outputs are latched and can be accessed by the basic read/write operations of the host processor. A more detailed circuit diagram can be seen in Appendix A. The memory map of the motion control interface card is shown in Table 1. Functions for the control of the timing are included in the software library that supports the interface card.

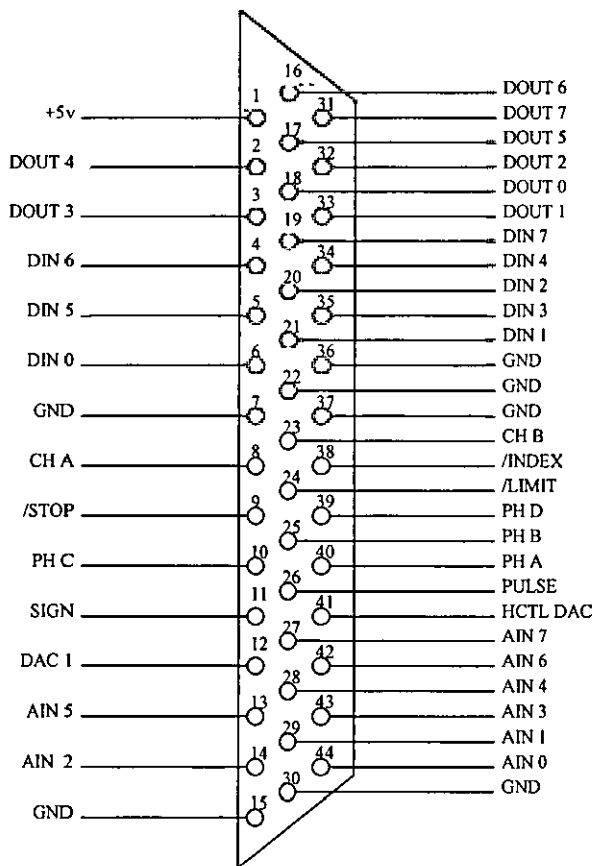


• Figure 1 – Interface card block diagram

| I/O Name | Access | Address |
|-----------------------------|-----------|-------------|
| 8255 PPI 1 – Port A Access | Read | BASE + 0x00 |
| 8255 PPI 1 – Port B Access | Read | BASE + 0x01 |
| 8255 PPI 1 – Port C Access | Wnte | BASE + 0x02 |
| 8255 PPI 1 – Control Byte | Wnte | BASE + 0x03 |
| HCTL 1100 Data | Read/Wnte | BASE + 0x08 |
| ADC 7870 – Start Conversion | N/A | BASE + 0x09 |
| DAC 312 – 12 Bit DAC | Wnte | BASE + 0x0A |
| Digital Input Byte | Read | BASE + 0x0B |
| Digital Output Byte | Read/Wnte | BASE + 0x0C |

• Table 1 - Interface card memory map

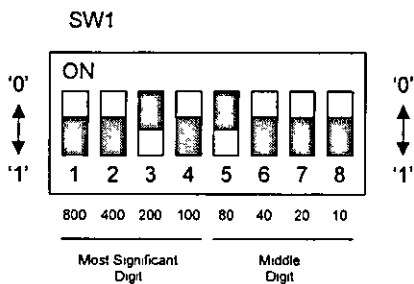
44 way D type connector



• Figure 2 – 44 Way D Type Connector

Setting the Base Address

It is possible to set the base address of each motion control interface card anywhere in the range 000(HEX) to FE0(HEX). Care should be taken to ensure that two cards do not have the same address. The diagram of Figure 3 below shows a base address of 280 (HEX)



• Figure 3 - Base Address Switch 1

Jumper Settings

There is only one jumper on the interface board. This provides for the special synchronisation function of the HCTL-1100. With this jumper it is possible to enable synchronisation by one of two methods:

- 1) Each board has its jumper connected to its own digital output 0
- 2) Each board is tied to the digital output of only one board

The first option does not offer anything in the way of useful functionality. However, the second option allows the user to synchronise motion to a common event, i.e. all axes will start moving at the same time.

HCTL-1100 Register Map

| Register Address | | Register Function |
|------------------|-----|------------------------------|
| Dec | Hex | |
| 00 | 00 | Flag Register |
| 05 | 05 | Program Counter |
| 07 | 07 | Status Register |
| 08 | 08 | 8 bit Motor Command Port |
| 09 | 09 | PWM Motor Command Port |
| 12 | 0C | Command Position (MSB) |
| 13 | 0D | Command Position |
| 14 | 0E | Command Position (LSB) |
| 15 | 0F | Sample Timer |
| 18 | 12 | Read Actual Position (MSB) |
| 19 | 13 | Read Actual Position |
| 20 | 14 | Read Actual Position (LSB) |
| 21 | 15 | Preset Actual Position (MSB) |
| 22 | 16 | Preset Actual Position |
| 23 | 17 | Preset Actual Position (LSB) |
| 24 | 18 | Commutator Ring |
| 25 | 19 | Commutator Velocity Timer |
| 26 | 1A | X |
| 27 | 1B | Y Phase Overlap |
| 28 | 1C | Offset |
| 31 | 1F | Maximum Phase Advance |
| 32 | 20 | Filter Zero, A |
| 33 | 21 | Filter Pole, B |
| 34 | 22 | Gain, K |
| 35 | 23 | Command Velocity (LSB) |
| 36 | 24 | Command Velocity (MSB) |
| 38 | 26 | Acceleration (LSB) |
| 39 | 27 | Acceleration (MSB) |
| 40 | 28 | Maximum Velocity |
| 41 | 29 | Final Position (LSB) |
| 42 | 2A | Final Position |
| 43 | 2B | Final Position (MSB) |
| 52 | 34 | Actual Velocity (LSB) |
| 53 | 35 | Actual Velocity (MSB) |
| 60 | 3C | Command Velocity |

• Table 2 - HCTL-1100 register map

Using the **TEST.EXE** Program (MS-DOS Version)

This test program is for use under the MS-DOS operating system. The simplest way to test the board is by the following method:

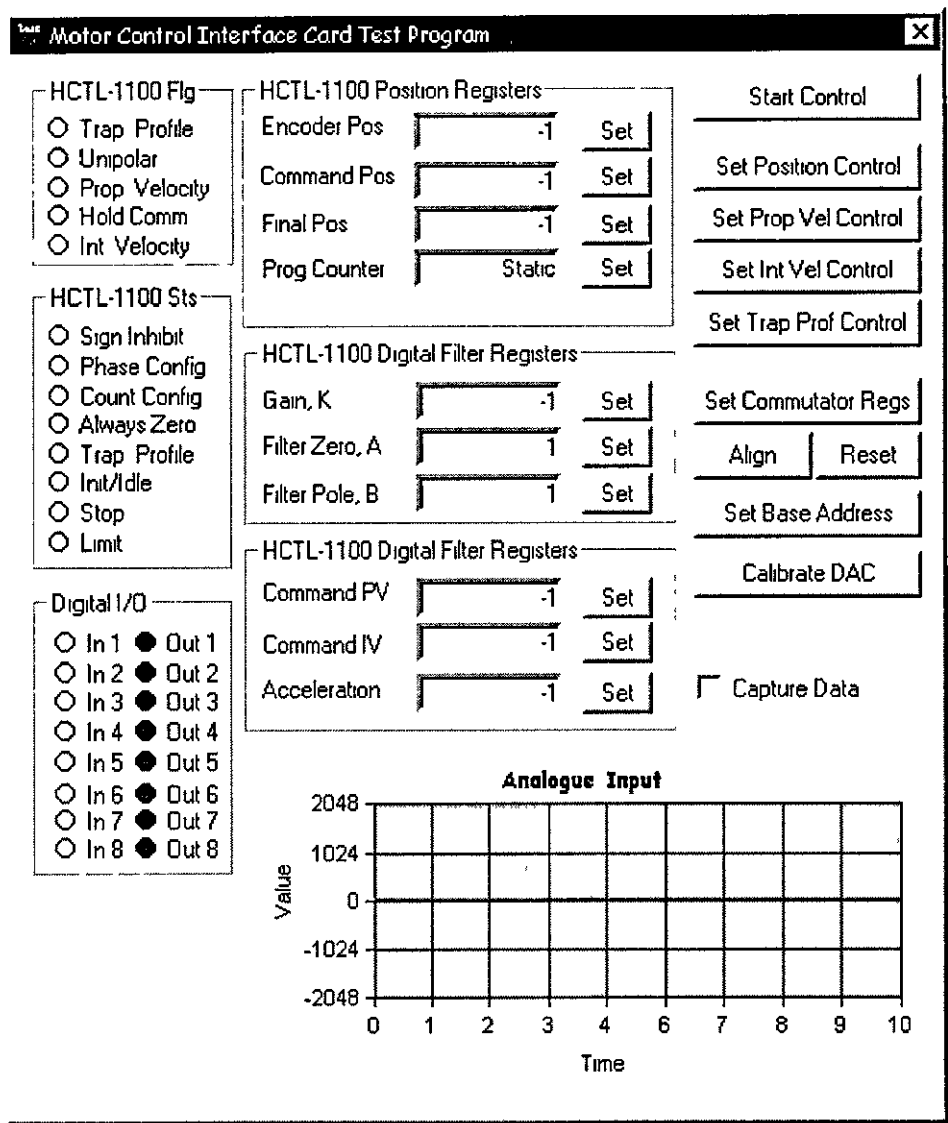
- 1) Before running **TEST.EXE** check that the board is getting the correct power supply. Once this has been successfully established, run **TEST EXE** and use an oscilloscope to probe.
- 2) Install the digital input and output components (if required) and test using **TEST.EXE**. It is easy to see if the digital output is working because of the ability of the interface card to read the values that have been set at the output. However, the input can be tested by applying either a logic '0' (0v) or a logic '1' (+5v) to the input.
- 3) Install and test the DAC312 circuit (if required). The DAC can be tested in several ways. Firstly, connect either an oscilloscope or a digital multi-meter to pin 12 of the 44 way 'D' connector and Ground (i.e. pin 7). Execute **TEST.EXE** and press 't' to test the DAC. Set the output to the minimum value (i.e. the most negative) and adjust the gain, PR1, until a suitable voltage for the application is achieved (MIN – 15v, MAX +15v). This particular amplifier circuit does not include zero offset adjustment, but can be included externally through further post-amplification. It is possible to test the DAC further by using the "Output Ramp Waveform" option on the "Test DAC" menu. Using an oscilloscope it is possible to see the ramp waveform.
- 4) Install the Analogue data capture interface (if required) and 82C55A interface. Run **TEST.EXE** and apply an input to each analogue channel. For test purposes it is sufficient to attach a potentiometer to +3v and –3v supply with the wiper connected to the input being tested. Turning the potentiometer in either direction will vary the input.
- 5) The final step is to install and test the HCTL-1100 circuit (if required) and the 82C55A interface (if it is not already installed).
 - (i) The first test is to see if data can be read from/written to the HCTL-1100. This is best accomplished by connecting an optical encoder to the HCTL's input. Again, run the **TEST.EXE** program and rotate the shaft of the encoder. The value showing the encoder position should change when the shaft is rotated.
 - (ii) Next, the write process shall be tested. Whilst still running **TEST EXE** press the 'd' key and type a new position (in encoder counts) to be entered in to the HCTL and press RETURN. The value is then written in to the HCTL, read back out and displayed in the same box. If the value is not the same as that typed there is a problem.

- (iii) Assuming that all is well, the HCTL has a new destination value and one other test can be performed in order to check that operation is correct. Switch the HCTL-1100 into position control mode by pressing the 's' key and selecting Position control from the following menu. Connect the input of an oscilloscope between pin 41 of the 44 way 'D' connector and ground. The 'scope should be reading either a positive or negative value (it doesn't matter which). Rotate the shaft of the encoder until the reading is close to the new destination value. Rotate the shaft very slowly through and past the desired position shown in the program window, the 'scope input should have changed direction. If the 'scope is always reading zero volts check that the gain PR2 is not set too low.

Using the **TEST.EXE** Program (Windows 9x Version)

This test program can be run under the Windows 95 and 98 operating systems. Currently the test program uses two ActiveX controls: GraphWindowCtrl and IndicatorCtrl. These two controls **MUST** be registered on the computer where the program is to be run. A screen shot of the Windows test program is shown in Figure 4. The program allows the user to view the status of the Digital I/O and the HCTL-1100. The state of digital output bit *n* can be toggled by clicking with left mouse button on the button labelled *outn*. The program also allows the user to change the values of the HCTL-1100 registers. The buttons on the right hand side mostly control the functionality of the HCTL-1100.

- 1 Start Control - Sets the HCTL-1100 in to the control mode
- 2 Position Control - Sets the HCTL-1100 Control mode to position
- 3 Set Prop Vel Control - Sets the control mode to proportional velocity control
- 4 Set Int Vel Control - Sets the control mode to proportional + integral velocity control
- 5 Set Trap Prof Control - Set the control mode to trapezoidal profile control mode
- 6 Set Commutator Regs - Allows the user to set the registers for the commutator for 3 and 4 pahse stepper/Brushless DC motors
- 7 Align - Forces the HCTL-1100 to ALIGN the stepper/Brushless DC motor (see HCTL-1100 data sheet)
- 8 Set Base Address - Forces the software to use the motion control interface card given by the specified base address
- 9 Calibrate DAC - Allows the user to test DAC output
- 10 Capture Data - Starts recording data from the ADC channels 1-8. By right-clicking on the graph a menu can be used to control which channels are to be displayed and allows the user to edit the graph's properties



• Figure 4 - Windows 9X test program

Chapter

3

Software

C Library

This library is for use with development systems like Microsoft DOS with Microsoft Quick C. This library will allow the user to perform motion control on more than one single-axis controller. The following paragraphs detail each instruction that has been provided in the library giving comprehensive examples of their use.

Initialisation Functions

void InitAddr(unsigned int base_addr)

This function must be called before any read/write function is used. *InitAddr* initialises the card at the given address. The initialisation involves setting-up the ports of the 82C55A Interface, resetting the HCTL-1100 and zeroing the output of the digital-to-analogue converters. The following example shows what must be done to initialise a system with two motion control interface cards before control can begin.

```
#include "motor.h"

#define BASE_ADDR1 0x280
#define BASE_ADDR2 0x320

void main (void)
{
    /* First initialise the two boards */
    InitAddr(BASE_ADDR1);
    InitAddr(BASE_ADDR2);
}
```

Digital Input/Output Functions

void WriteDigitalOut(unsigned int base_addr, char c)

Writes the byte *c* to the digital output port of the card specified by *base_addr*

unsigned char ReadDigitalOut(unsigned int base_addr)

Reads the bytes that has been previously set with *WnteDigitalOut*. The function returns the digital output port value of the card specified by *base_addr*

void DigitalOutBitSet(unsigned int base_addr, unsigned char bit, unsigned char set)

Sets or resets a specific bit on the digital output port of the card specified by *base_addr* *bit* denotes the index of the bit to change. If *set* is '0' the bit is reset (i.e. '0') and if *set* is '1' the bit is set (i.e. '1')

unsigned char ReadDigitalIn(unsigned int base_addr)

Samples the digital input port of the board specified by *base_addr*

Analogue Input/Output Functions

int ReadADC(unsigned int base_addr, int chan)

Performs an analogue-to-digital conversion of analogue input [*chan*] and returns the integer value from the conversion. The conversion is performed based on the address specified by *base_addr* where $0 \leq \text{chan} \leq 7$. The example below shows how to create an array of integer values and store an analogue-to-digital conversion from the respective channel in each.

```
#include "motor.h"

#define BASE_ADDR 0x280

void main(void)
{
    /* Declare an array of 8 ints and a count var */
    int ain[8], i,

    /* Initialise interface card */
    InitAddr(BASE_ADDR),

    /* Loop thru each of the 8 channels */
    for(i=0; i<8; i++)
    {
        /* Store the ADC value */
        ain[i] = ReadADC(BASE_ADDR, i),
    }
}
```

void OutDAC(unsigned int base_addr, signed int val)

Performs a digital-to-analogue conversion of *val* on the DAC312 and card specified by *base_addr*. The analogue output appears on pin 12 of the 44 way 'D' type connector. *val* is limited to the range $-2048 \leq \text{val} \leq +2047$.

Low Level HCTL-1100 Control Functions

void ResetHCTL(unsigned int base_addr)

This function forces a hardware reset of the HCTL-1100 motion control interface on the card specified by *base_addr*. Please refer to the HCTL-1100 data sheet for a detailed description of the status of the HCTL-1100 after reset.

void SoftResetHCTL(unsigned int base_addr)

This function forces a software reset of the HCTL-1100 motion control interface on the card specified by *base_addr*. Please refer to the HCTL-1100 data sheet for a detailed description of the status of the HCTL-1100 after a software reset.

unsigned char ReadHCTL(unsigned int base_addr, unsigned char reg_addr)

Reads data from the HCTL-1100 register specified by *reg_addr* on the board specified by *base_addr*. For more details on register access and register assignment see the section entitled "HCTL-1100 Register Map" on page 196. This example shows how to initialise the HCTL-1100 and read the status register. The pre-processor directives and declaration of the *main* function have been omitted for brevity.

```
/* Declare variable to store value of status register */
unsigned char sts,

/* Initialise the card */

InitAddr(BASE_ADDR);

/* Reset the HCTL */
ResetHCTL(BASE_ADDR);

/* Read the value from the status register */
sts = ReadHCTL(BASE_ADDR, 0x07);
```

void WriteHCTL(unsigned int base_addr, unsigned char reg_addr, unsigned char byte)

WriteHCTL puts the value ***byte*** into the register specified by ***reg_addr*** ***base_addr*** specifies the card on which the operation will occur. Again, in this example the pre-processor commands and declaration of the ***main*** function have been excluded

```
/* Initialise the card */
InitAddr(BASE_ADDR);

/* Reset the HCTL */
ResetHCTL(BASE_ADDR);

/* Write 255 to the sample timer register */
WriteHCTL(BASE_ADDR, 0x0f, 0xff);
```

High Level HCTL-1100 Control Functions

unsigned char GetHCTLStatus(unsigned int base_addr)

Returns the contents of the status register on the board specified by ***base_addr***

void SetHCTLStatus(unsigned int base_addr, unsigned char sts)

Sets the contents of the status register to ***sts*** on the board specified by ***base_addr***. Only the lower four bits are set.

unsigned char GetHCTLFlags(unsigned int base_addr)

Returns the contents of the flags register on the board specified by ***base_addr***

void SetHCTLFlags(unsigned int base_addr, unsigned char flg)

Sets the contents of the flags register to ***flg*** on the board specified by ***base_addr***

signed long int Get HCTLPosition(unsigned int base_addr)

Retrieves the *signed* current encoder position for the card specified by *base_addr*

void Set HCTLPosition(unsigned int base_addr, signed long int pos)

Presets the encoder position to *pos* counts. This is only allowed when the HCTL-1100 is in idle mode

signed long int Get HCTLDesiredPosition(unsigned int base_addr)

Retrieves the current desired position in encoder counts from the HCTL-1100's registers

void Set HCTLDesiredPosition(unsigned int base_addr, signed long int pos)

Sets the desired position to *pos* encoder counts

signed long int Get HCTLFinalPosition(unsigned int base_addr)

Retrieves the final position in encoder counts from the HCTL-1100's registers

void Set HCTLFinalPosition(unsigned int base_addr, signed long int pos)

Sets the final position to *pos* encoder counts

signed int Get HCTLVelocity(unsigned int base_addr)

Retrieves the current velocity from the HCTL-1100's registers in quadrature counts per sample time

void Set HCTLProportionalVelocity(unsigned int base_addr, signed int vel)

Sets the registers that contain the command velocity used during Proportional Velocity Control to *vel* quadrature counts per sample time

void Set HCTLIntegralVelocity(unsigned int base_addr, unsigned char vel)

Set the register that contains the command velocity used during Integral Velocity Control to *vel* quadrature counts per sample time

void Set HCTLAcceleration(unsigned int base_addr, unsigned int acc)

Sets the acceleration registers used in Trapezoidal Profile and Integral Velocity Control modes to *acc* quadrature counts per sample time per sample time *acc* is a positive scalar number between 0(HEX) and 7FFFF(HEX)

void Set HCTLMaximumVelocity(unsigned int base_addr, unsigned char vel)

Set the maximum allowable velocity to *vel* encoder counts per sample time *vel* is a positive scalar number between 0(HEX) and 7F(HEX)

```
void Set HCTLDigitalFilter(unsigned int base_addr, unsigned char k, unsigned char a,  
                           unsigned char b)
```

Sets the digital filter registers gain to *k*, zero to *a*, and pole to *b*. Refer to HCTL-1100 documentation to see how to calculate the values of *k*, *a*, and *b*.

```
void SetHCTLCommutatorRegs(unsigned int base_addr, unsigned char no_phases,  
                           unsigned char lsQuadCounts, unsigned char rng, unsigned char  
                           timer, unsigned char x, unsigned char y, signed char offset, unsigned  
                           char max_adv)
```

Sets up the commutator registers for control of 2, 3, and 4-phase stepper and brushless dc motors. Please refer to the HCTL-1100 data sheet for more information on programming the commutator state-machine.

no_phases specifies the number of phases of the motor.

lsQuadCounts indicates whether the system should use full or quadrature decoding. 1 specifies quadrature decoding, any other value specifies full decoding.

rng is measured in encoder counts per torque cycle and is stored in the *rng* register.

timer specifies the value to be loaded in the velocity timer register.

x gives the number of encoder counts that only one phase is active.

y denotes the number of encoder counts the two consecutive phases are active. $x+y = rng/no_phases$.

offset determines the relative start of the commutation cycle with respect to the index pulse.

timer sets the value in the Velocity timer register.

max_adv sets the value in the maximum advance register.

HCTL-1100 Control Mode Functions

```
void StartHCTLControl(unsigned int base_addr)
```

Sets the HCTL-1100 on the board specified by *base_addr* to control mode. In position control mode the HCTL-1100 will take desired position values and move to that position.

void StopHCTLControl(unsigned int base_addr)

Sets the HCTL-1100 in idle/init mode. In position control mode any change in desired position will not cause the axis to move.

void SetHCTLPositionControl(unsigned int base_addr)

Sets the HCTL-1100 in to position control mode. Use *GetHCTLFlags*, *GetHCTLPosition*, and *SetHCTLDesiredPosition* to monitor and control the axis.

void SetHCTLPropVelocityControl(unsigned int base_addr)

Sets the HCTL-1100 in to proportional velocity control mode. Use *GetHCTLFlags*, *GetHCTLPosition*, *GetHCTLVelocity*, and *SetHCTLProportionalVelocity* to monitor and control the axis.

void SetHCTLIntVelocityControl(unsigned int base_addr)

Sets the HCTL-1100 in to integral velocity control mode. Use *GetHCTLFlags*, *GetHCTLPosition*, *SetHCTLAcceleration*, and *SetHCTLIntegralVelocity* to monitor and control the axis.

void SetHCTLTrapProfControl(unsigned int base_addr)

Sets the HCTL-1100 in to trapezoidal profile control mode. Use *GetHCTLFlags*, *GetHCTLStatus*, *GetHCTLPosition*, *SetHCTLMaximumVelocity*, *SetHCTLFinalPosition*, *SetHCTLAcceleration*, and *SetHCTLIntegralVelocity* to monitor and control the axis.

C++ Class Library

This library was developed using Microsoft Visual C++ 5.0/6.0 and can be used under Windows 9x to access the hardware directly. It is not possible to use this library directly under Windows NT because of the way in the operating system prohibits read/write access to hardware. However, it is possible to use these classes to write virtual device drivers for both Windows9x and Windows NT. The C++ library is valid for use in the QNX 4.25 real-time operating system, but does not compile using the Watcom C++ compiler using either the 16-bit or 32-bit compiler/linker.

The *CSingleAxisControlCard* class can be used in the following way

Constructor

CSingleAxisControlCard .*CsingleAxisControlCard*

CsingleAxisControlCard(unsigned int *addr*),

CsingleAxisControlCard(),

Parameters.

addr The base address of the instances interface card

Remarks

Call either form of the constructor to create an instance of this class One form of the constructor creates an instance with a predefined address, while the second form simply creates and instance Problems will occur if the base address has not been set correctly

See Also : *GetBaseAddress* and *SetBaseAddress*

Initialisation Functions

CSingleAxisControlCard::SetBaseAddress

void *SetBaseAddress*(unsigned int *addr*),

Parameters:

addr The base address of the instances interface card

Remarks :

This function is used to change the base address that this instance refers to

CSingleAxisControlCard::GetBaseAddress

unsigned int *GetBaseAddress*(),

Remarks:

GetBaseAddress does not take any parameters This function returns the address of the board that this instance currently refers to

Digital Input/Output Functions***CSingleAxisControlCard::SetDigitalOutput***

void SetDigitalOutput(unsigned char ch),

Parameters:

ch byte to set at output port.

Remarks:

Set the digital output port of the card to the value specified by *ch*

CSingleAxisControlCard::GetDigitalOutput

unsigned char GetDigitalOutput(),

Remarks:

This function sample the digital output port and returns the value that currently appears on this port.

CSingleAxisControlCard::BitSetDigitalOutput

void BitSetDigitalOutput(unsigned char bit, unsigned char set),

Parameters

bit index of bit to be set or reset (0-7)

set '1' sets bit[*bit*] to '1', '0' resets bit[*bit*]

Remarks

BitSetDigitalOutput sets or resets a specific bit of the digital output port

CSingleAxisControlCard::GetDigitalInput

unsigned char GetDigitalInput()

Remarks:

This function samples the digital input port and return the byte that represent the data on that port.

Analogue Input/Output Functions***CSingleAxisControlCard::ReadADC***

long ReadADC(unsigned char chan),

Parameters:

chan Channel to sample data (0-7)

Remarks:

Samples data on channel *chan* and returns the data

CsingleAxisControlCard::OutDAC

void OutDAC(long data),

Parameters:

data Value to be sent out from DAC312

Remarks:

Outputs the given value from the DAC312

Low Level HCTL-1100 Control Functions

CSingleAxisControlCard::ReadHCTL

unsigned char ReadHCTL(unsigned char reg),

Parameters:

reg address of HCTL-1100 register to retrieve data from

Remarks:

Returns the contents of the specified register. See HCTL-1100 register map in **Error! Reference source not found.**

CSingleAxisControlCard::WriteHCTL

void WriteHCTL(unsigned char reg, unsigned char ch),

Parameters:

reg address of HCTL-1100 register in which to write data

ch byte of data to write

Remarks:

Writes a byte of data to the register specified by *reg*

High Level HCTL-1100 Control Functions

CSingleAxisControlCard::ResetHCTL

void ResetHCTL(void),

Remarks:

Force s a hardware reset of the HCTL-1100 Control Interface. See HCTL-1100 data sheet for information on status after a hardware reset.

CSingleAxisControlCard::SoftResetHCTL

void SoftResetHCTL(void),

Remarks:

Performs a software reset of the HCTL-1100 Control Interface. See HCTL-1100 data sheet for information on status after a software reset.

CSingleAxisControlCard::GetHCTLStatus

unsigned char GetHCTLStatus(void),

Remarks:

Returns the status byte from the HCTL-1100

CSingleAxisControlCard::SetHCTLStatus

void SetHCTLStatus(unsigned char ch),

Parameters:

ch status byte to be set (lower 4 bits used only)

Remarks:

Sets the lower 4 bits of the status register to the values sets in *sts*

CSingleAxisControlCard::GetHCTLFlags

unsigned char GetHCTLFlags(void),

Remarks

Returns the byte from the flags register of the HCTL-1100

CSingleAxisControlCard::SetHCTLStatus

void SetHCTLFlags(unsigned char bit, unsigned char set),

Parameters:

bit flag to be set. 0, 2, 3, 4 or 5

set set (1) or clear (0)

Remarks:

Sets or clears the specified flag

CSingleAxisControlCard::GetHCTLPosition

long GetHCTLPosition(void),

Remarks:

Returns the current encoder position from the HCTL-1100

CSingleAxisControlCard::SetHCTLPosition

void SetHCTLPosition(long pos),

Parameters:

pos new encoder position

Remarks:

Pre-sets the actual encoders position register to the specified value

CSingleAxisControlCard::GetHCTLCommandPosition

long GetHCTLCommandPosition(void),

Remarks.

Returns the contents of the command position register

CSingleAxisControlCard::SetHCTLCommandPosition

void SetHCTLCommandPosition(long pos),

Parameters

pos new command position

Remarks

Pre-sets the command position register to the specified value

CSingleAxisControlCard::SetHCTLCommutatorRegs

void SetHCTLCommutatorRegs(unsigned char Is4Phase, unsigned char IsFullCounts, unsigned char nng, unsigned char timer, unsigned char x, unsigned char y, signed char offset, unsigned char max_adv)

Parameters:

| | |
|---------------------|--|
| <i>Is4Phase</i> | Specifies 4 phase motor (1) or 3 phase motor (0) |
| <i>IsFullCounts</i> | Specifies full count decoding (1) or quadrature decoding (0) |
| <i>nng</i> | Contents of the nng register |
| <i>timer</i> | Determines phase advance at given velocity |
| <i>x</i> | Specifies interval during which only one phase is active |
| <i>y</i> | Specifies interval during which two pahses are active |
| <i>Offset</i> | Determines relative start of commutation cycle wrt index pulse |
| <i>max_adv</i> | Maximum allowable phase advance |

Remarks

Sets the data required to operate the commutator Refers to HCTL-1100 data sheet for more information on determining the values for these registers

CSingleAxisControlCard GetHCTLFinalPosition

long GetHCTLFinalPosition(void),

Remarks:

Returns the contents of the final position register

CSingleAxisControlCard::SetHCTLFinalPosition

void SetHCTLFinalPosition(long pos),

Parameters:

pos new final position

Remarks:

Sets the final position register to the specified value

CSingleAxisControlCard::GetHCTLFilterGain

signed char GetHCTLFilterGain(void),

Remarks:

Returns the contents of the digital filter gain register

CSingleAxisControlCard::SetHCTLFilterGain

void SetHCTLFilterGain(signed char ch),

Parameters:

ch new filter gain

Remarks:

Sets the digital filter gain register to the specified value

CSingleAxisControlCard::GetHCTLFilterPole

signed char GetHCTLFilterPole(void),

Remarks:

Returns the contents of the digital filter pole register

CSingleAxisControlCard::SetHCTLFilterPole

void SetHCTLFilterPole(signed char ch),

Parameters:

ch new filter pole

Remarks:

Sets the digital filter register to the specified value

CSingleAxisControlCard::GetHCTLFilterZero

signed char GetHCTLFilterZero(void),

Remarks:

Returns the contents of the digital filter zero register

CSingleAxisControlCard::SetHCTLFilterZero

void SetHCTLFilterZero(signed char ch),

Parameters:

ch new filter zero

Remarks:

Sets the digital filter zero register to the specified value

CSingleAxisControlCard::GetHCTLProportionalVelocity

signed int GetHCTLProportionalVelocity(void),

Remarks:

Returns the contents of the command velocity register. Used in proportional velocity control

CSingleAxisControlCard::SetHCTLProportionalVelocity

void SetHCTLSetProportionalVelocity(signed int vel),

Parameters:

vel new command velocity

Remarks:

Sets the contents of the command velocity register to the specified value. Used in proportional velocity control.

CSingleAxisControlCard::GetHCTLIntegralVelocity

signed char GetHCTLIntegralVelocity(void),

Remarks:

Returns the contents of the command velocity register. Used in integral velocity control.

CSingleAxisControlCard::SetHCTLFilterZero

void SetHCTLFilterZero(signed char vel),

Parameters:

vel new command velocity

Remarks:

Sets the command velocity register to the specified value. Used in integral velocity control.

CSingleAxisControlCard GetHCTLAcceleration*signed int GetHCTLAcceleration(void),***Remarks:**

Returns the contents of the acceleration register. Used in integral velocity control and trapezoidal profile mode

CSingleAxisControlCard..SetHCTLProportionalVelocity*void SetHCTLSetProportionalVelocity(signed int acc),***Parameters:**

acc new command velocity

Remarks:

Sets the contents of the acceleration register to the specified value. Used in integral velocity control and trapezoidal profile mode

HCTL-1100 Control Mode Functions***CSingleAxisControlCard SetHCTLPositionControl****void SetHCTLPositionControl(void),***Remarks:**

Sets the flags and status registers of the HCTL-1100 to position control mode. Use *StartHCTLControl* to enable this control mode.

CSingleAxisControlCard SetHCTLProportionalVelocityControl*void SetHCTLProportionalVelocityControl(void),***Remarks**

Sets the flags and status registers of the HCTL-1100 to Proportional Velocity control mode. Use *StartHCTLControl* to enable this control mode.

CSingleAxisControlCard::SetHCTLIntegralVelocityControl

void SetHCTLIntegralVelocityControl(void),

Remarks:

Sets the flags and status registers of the HCTL-1100 to Integral Velocity control mode Use *StartHCTLControl* to enable this control mode

CSingleAxisControlCard::SetHCTLProportionalVelocityControl

void SetHCTLProportionalVelocityControl(void),

Remarks:

Sets the flags and status registers of the HCTL-1100 to Proportional Velocity control mode Use *StartHCTLControl* to enable this control mode

CSingleAxisControlCard::SetHCTLTrapProfileControl

void SetHCTLTrapProfileControl(void),

Remarks:

Sets the flags and status registers of the HCTL-1100 to Trapezoidal Profile control mode Use *StartHCTLControl* to enable this control mode

CSingleAxisControlCard::StartHCTLControl

void StartHCTLControl(void),

Remarks.

Enables the control loop in the current mode

CSingleAxisControlCard StopHCTLControl

void StopHCTLControl(void),

Remarks.

Disables the control loop

C++ Code Sample

This simple example creates two instances of the *CSingleAxisControlCard* class and assigns each of them a base address. The first card is then used to perform a single movement from the current position to a new destination. Both instances are then deleted.

```
void main(void)
{
    CSingleAxisControlCard *pCard1, pCard2;

    // Create 2 instances
    pCard1 = new CSingleAxisControlCard(0x280),
    pCard2 = new CSingleAxisControlCard(),

    // Set addr of card2 (card1 addr already set)
    pCard2->SetBaseAddress(0x320);

    // Reset the HCTL chips
    pCard1->ResetHCTL();
    pCard2->ResetHCTL();

    // Set card 1 to position control
    pCard1->SetHCTLPositionControl();
    // Set desired position to 15000 encoder counts
    pCard1->SetHCTLCommandPosition(15000);

    // Start the control
    pCard1->StartHCTLControl();

    // We might wait for motor to reach dest position
    do
    {
        }while(pCard1->GetHCTLPosition()<14995 ||
              pCard1->GetHCTLPosition()>15005),

    // Stop the control loop
    pCard1->StopHCTLControl(),

    // clean-up and exit
    delete pCard1;
    delete pCard2,
}
```

Appendix

A

PCB Production

PCB Production Artwork

The design of the single-axis controller interface card requires a double sided, plated through-hole PCB. The artwork for each side of the PCB, solder and component, can be found in Appendix A along with a component list and layout diagram.

It is possible to populate the board to implement only certain functions. That is if only digital input/output is required there is no need to use the relatively expensive HCTL 1100, 82C55A, DAC312 and AD7870 components that would not be required in such a situation. However, it must be noted that if either the HCTL1100 OR the AD7870 are to be used the 82C55A must be included for the successful operation of these circuits. The address decoding logic will always be implemented.

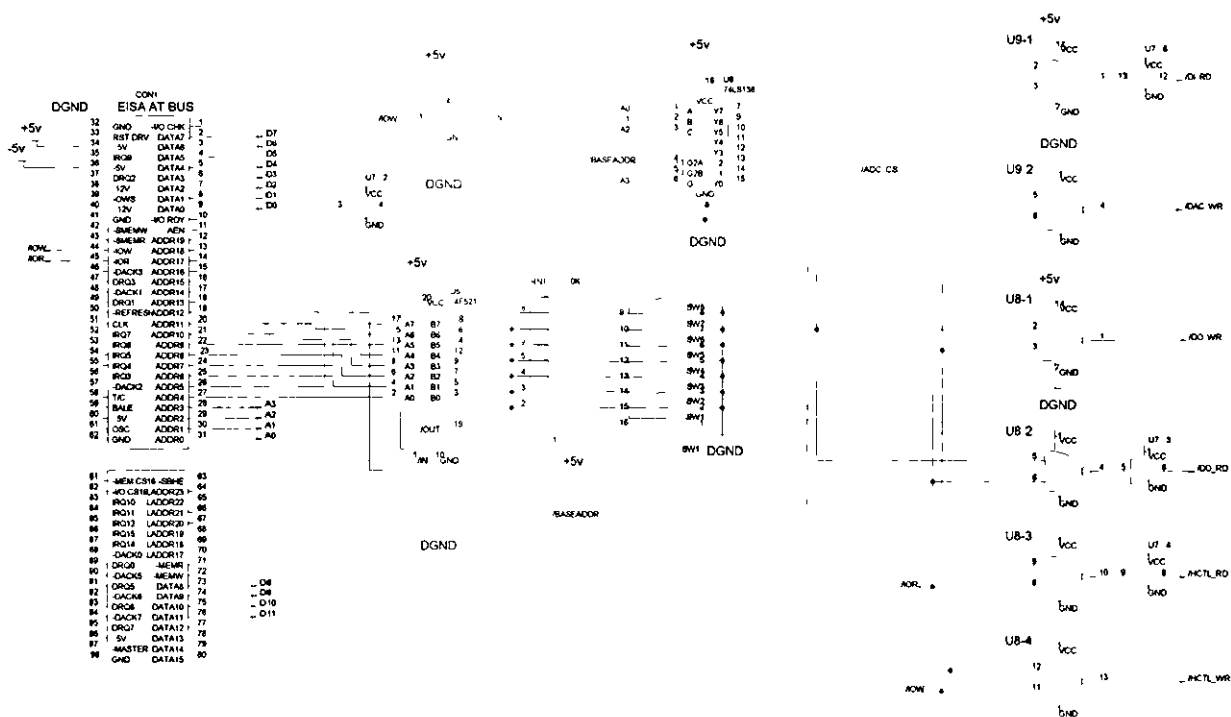
Building and Testing the Interface Card

A test program is provided as part of the package that enables quick and easy testing of the motion control interface card. The program contains help information on which keys perform functions. The source code for the program can be found in the examples directory.

Address Decoding Circuit

Before installing any of the input/output circuits it is imperative to test the address decoding logic. Install the address decoding logic components and plug the board in to the computer. Table 3 below lists the components required to install the address decoding circuit. A component layout diagram is included in Figure 11. It is recommended that capacitors C1-C14 be installed regardless of how many circuits are to be used.

Figure 5 shows the circuit diagram of the address decoding logic. A memory map showing how the logic decodes the address signal is given in Table 1 on page 194.



• Figure 5 - Address decoding circuit

| Qty | Component | Identity |
|-----|--------------------------------------|----------|
| 2 | 74LS02 – Quad Dual input OR gates | U8, U9 |
| 1 | 74LS27 – Triple three input OR Gates | U10 |
| 1 | 74LS04 – Hex Inverter | U7 |
| 1 | 74LS138 – 3-to-8 Decoder | U6 |
| 1 | 74LS521 – 8 bit identity comparator | U5 |
| 1 | 10K 8 resistor 9 pin network | RN1 |
| 1 | 8 way DIL switch | SW1 |

• Table 3 - Address decoding circuit components

Prior to installing the card in a computer it is necessary to check that no damage will occur to the power supply. Using a digital multi-meter check that there is no continuity between pins B1, B3, or B5 of the EISA connector, which represent ground, +5v and – 5v respectively. Set the base address to a suitable value checking that it will not conflict with other ISA devices such as sound cards, network adapters, etc. See “Setting the Base Address” on page 195 for more details of this process.

Once the interface card has been installed, run the program *TEST EXE* (it does not matter if you use the MS-DOS or Windows 9x version of the program). Using an oscilloscope it is then possible to check the operation of the address decoding circuitry.

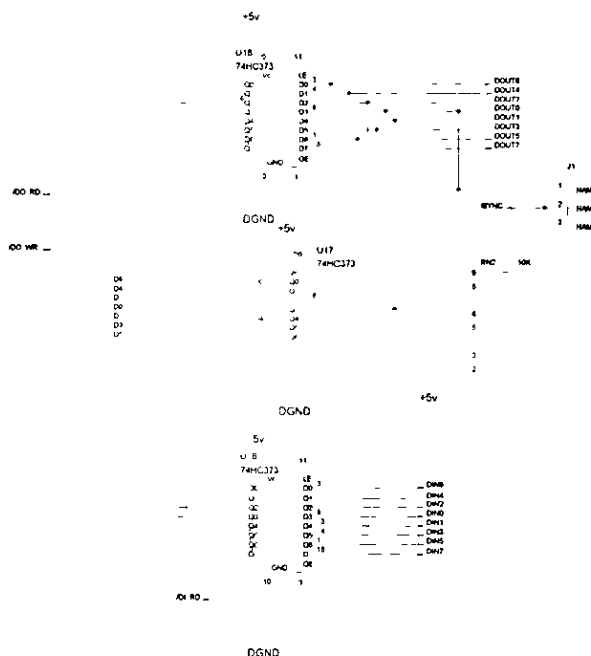
Firstly, check that the IC's U5-U10 are receiving the correct power supply. If it is not, make sure that the interface card has been installed properly.

With the test program running, connect the oscilloscope probe to pin 19 of U5. The signal on this pin should be normally high (logic 1), periodically going low (logic 0). If this is not the case, check that pins 2, 3, 6, 8, 11, 13, 15, and 17 of U5 are receiving data from the address bus. If the address signals are appearing at these pins, check that the base address selected on the card and the base address used by the program are the same. Otherwise, check that the card has been installed properly and that the power supply is stable. Probe pins 3, 5, 7, 9, 12, 14, 16, and 18 to see if the resistor network RN1 and the switch SW1 are working correctly.

The next step is to verify the operation of the 74LS138 3-to-8 decoder, U6. Using the oscilloscope probe, check that pin 16 and pin 8 of U6 have the correct supply voltages and that pin 4 of U6 has the same signal as pin 19 of U5. Pin 5 of U6 should always be 0V. Pins 1, 2, 3, and 6 should be connected to A0, A1, A2, and A3 of the address bus. The test program periodically reads the digital input in order to update the display. Pin 12 of U6 represents the selection of the digital output interface as the current device, normally high, periodically going low. This signal is ORed with /IOR from the EISA connector and is used to select the digital input device when a read operation is performed on this particular address. Pin 1 of U9 should be normally low, periodically going high, and pin 12 of U7 should be the inverse of this. If this is not the case, check the power supply to U9 and U7. It is also possible to check the operation of the other device selection signals; refer to Figure 5.

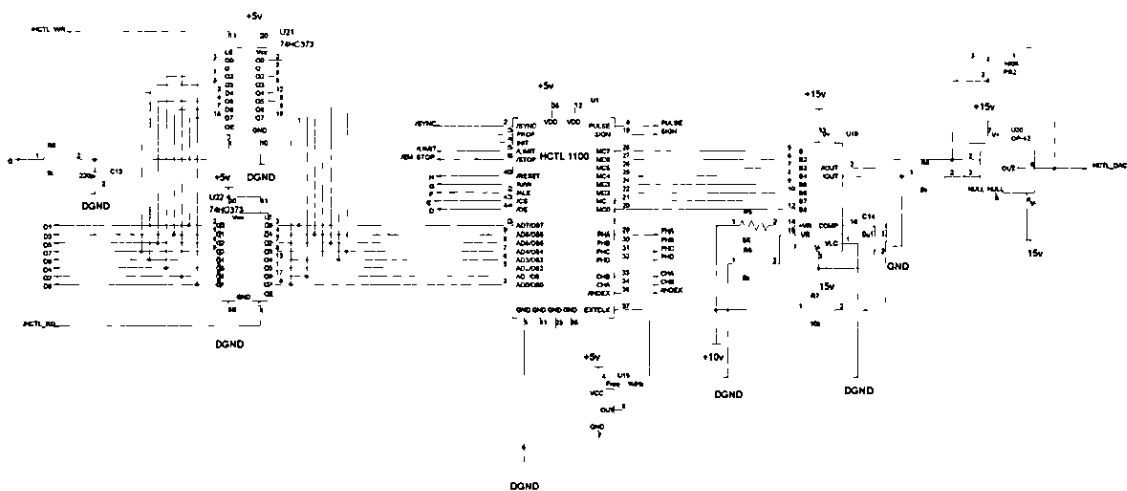
Digital I/O Circuit

Figure 6 shows the circuit diagram of the digital I/O interface and Table 4 lists the components required to implement the circuit.



• Figure 6 - Digital I/O Circuit

HCTL-1100 Circuit



• Figure 10 - HCTL-1100 Control Interface

| Qty | Component | Identity |
|-----|---|------------|
| 1 | HCTL-1100 40 pin PLCC | U1 |
| 1 | 1 MHz Xtal Oscillator 14 pin DIL TTL output | U15 |
| 1 | DAC08 | U19 |
| 1 | OP42 or TL071 | U20 |
| 1 | 74LS373 – Octal Data Latch | U21, U22 |
| 1 | 5K Resistor (0.25W) | R5, R6, R8 |
| 1 | 10K Resistor (0.25W) | R7 |
| 1 | 1K Resistor (0.25W) | R9 |
| 1 | 100K 24 Turn Cermet Trimmer | PR2 |
| 1 | 220 pF Capacitor | C13 |
| 1 | 0.1 uF Capacitor | C14 |

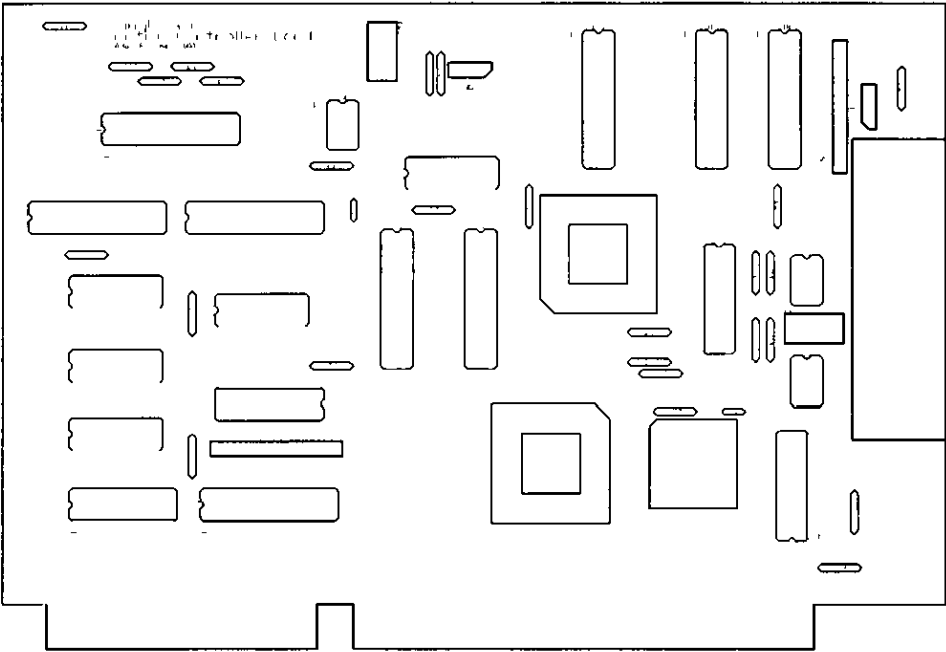
Full Component List

The components that are required to build one single-axis controller interface card

| Qty | Component | Identity |
|-----|---|-------------------|
| 7 | 74LS373 – Octal Data Latch | U11, U12, U15-18, |
| 2 | 74LS02 – Quad Dual input OR gates | U8, U9 |
| 1 | 74LS27 – Triple three input OR Gates | U10 |
| 1 | 74LS04 – Hex Inverter | U7 |
| 1 | 74LS138 – 3-to-8 Decoder | U6 |
| 1 | 74LS521 – 8 bit identity comparator | U5 |
| 1 | DAC312 – 12 bit DAC | U13 |
| 1 | DAC08 – 8 bit DAC | U19 |
| 1 | 82C55A (PLCC package) – Programmable Peripheral Interface | U4 |
| 1 | HCTL1100 (PLCC package) – General Purpose Motion Control IC | U1 |
| 1 | ADC7870 (PLCC package) – 12 bit ADC | U3 |
| 1 | ADG408 – Single 8 channel/Differential 4 channel analogue MUX | U2 |
| 1 | 741 Operational amplifier | U23 |
| 2 | TL071 Low noise Operational Amplifier | U14, U20 |
| 1 | 1 MHz Xtal Oscillator 14 pin DIL TTL output | U15 |
| 14 | 0.1 uF Multi-layer ceramic Capacitor 50v DC | C1-12, C14, C15 |

| | | |
|---|---|--------------|
| 1 | 220pF Multi-layer ceramic capacitor 200v DC | C13 |
| 3 | 10K Resistor 0.25W | R1, R3, R7 |
| 1 | 9K1 Resistor 0.25W | R10 |
| 4 | 5K1 Resistor 0.25W | R2, R4-6, R8 |
| 1 | 3K9 Resistor 0.25W | R11 |
| 1 | 1K Resistor 0.25W | R9 |
| 2 | 10K 8 resistor 9 pin network | RN1, RN2 |
| 2 | 100K 24 Turn Cermet Trimmer | PR1, PR2 |
| 1 | 8 way DIL switch | SW1 |
| 1 | 44 Way 90 deg, PCB Mount, High Density D type connector | CON3 |
| 1 | 3 way Square pin KK header (with friction lock) | CON2 |
| 1 | 3 pin 0.1" pitch jumper | J1 |

Component Layout Diagram



• Figure 11 - Component Layout Diagram

Appendix B **Power Electronics Interface Connection**
Diagrams

The diagram in Figure 85 shows the layout of the back plate from the power electronics interface unit. Each column of connectors represent the inputs and outputs of one single axis. There are five multi-pole connectors and 3 uni-pole connectors for each axis. The multi-pole connectors are labelled (from top to bottom) Control Card Input, Encoder Input, Motor Output, Digital I/O, and Analogue Input. These five connectors are discussed in more detail below. The uni-pole connectors are labelled (from left to right) Motor Voltage +, Motor Ground, and Motor Voltage - and are for connecting the power supply to power the motor.

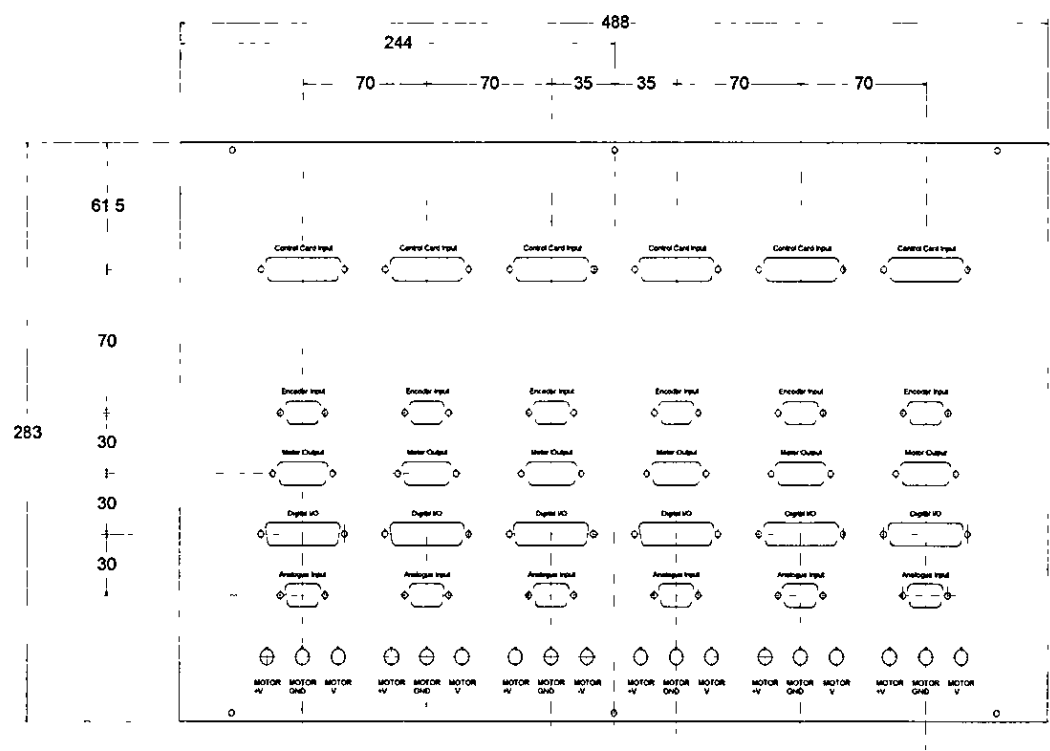


Figure 85 - Back plate dimensions

B.1. Input and Output Connector Diagrams

B.1.1 Control Card Input

It should be noted that this connector does not exactly match the output of the Single Axis Control Interface. Care should be taken when wiring an interface cable. The +12V and -12V supply are connected to the 3 pin Molex connector on the Single Axis Control Card.

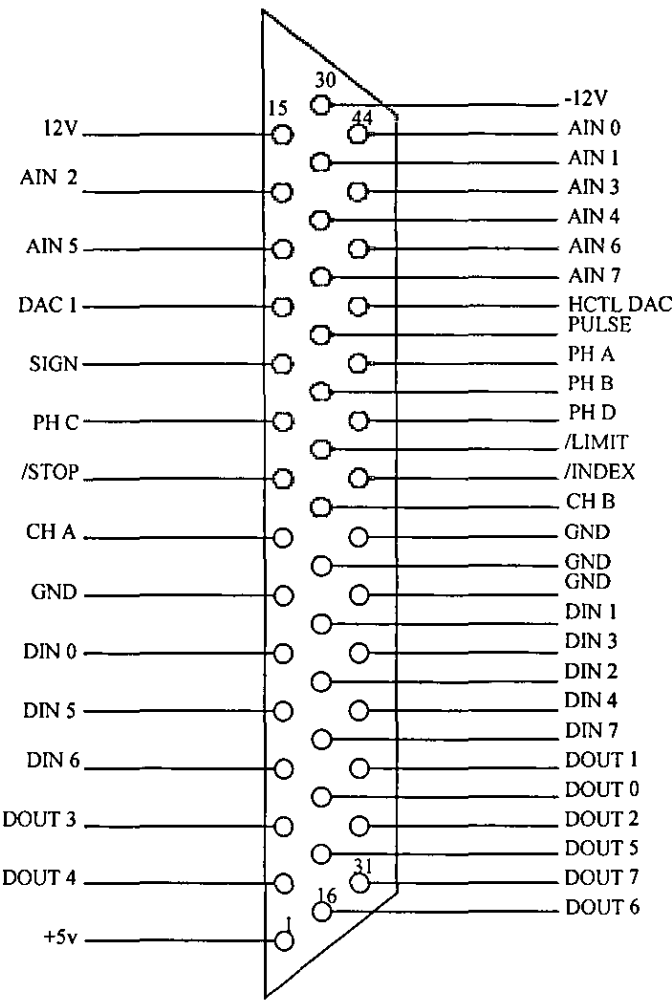


Figure 86 - 44 Way D-Type Control Card Connector

B.1.2 Encoder Input Connector

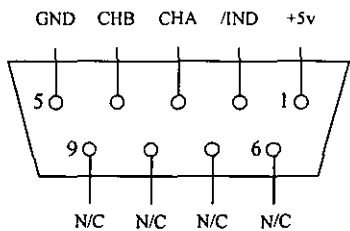


Figure 87 - 9 Way D-Type Encoder Connector

B.1.3 Motor Output Connector

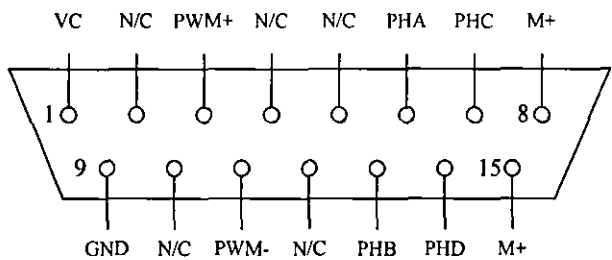


Figure 88 - 15 Way D-Type Motor Connector

B.1.4 Digital I/O Connector

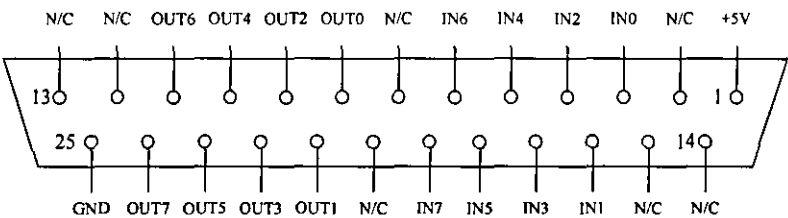


Figure 89 - 25 Way Digital IO Connector

B.1.5 Analogue Input Connector

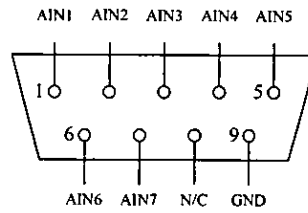


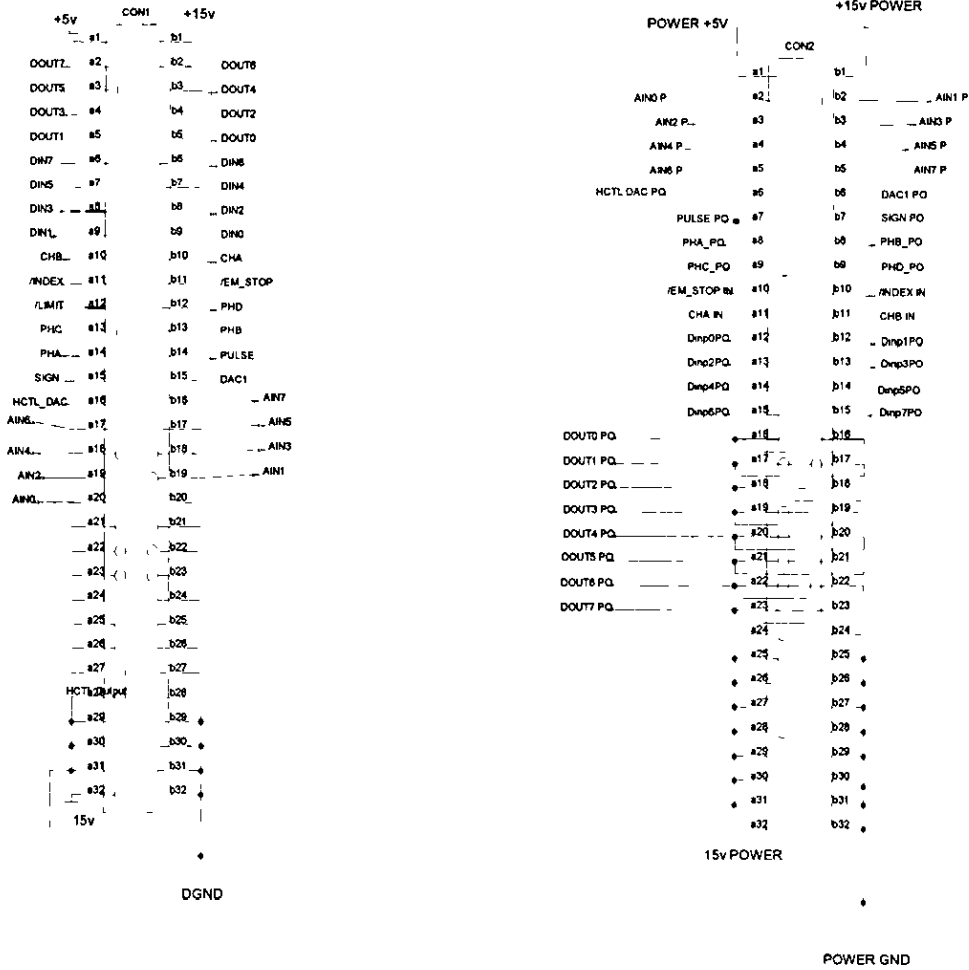
Figure 90 - 9 Way Analogue Input Connector

B.2. Internal Connector Diagrams

So that new cards can be designed for the system, the three internal connectors for each axis are described here

B 2.1 Isolator Connectors

The uppermost of the two connectors for the isolator board represents the signals to/from the interface card. The lower connector connects the power amplifier board and digital IO. The symbols in the diagrams have the same meaning as in the Interface Control Card Manual, with the addition to some of the symbols indicating that it is a high-power output (PO).



Upper Isolator
Connector

Lower Isolator
Connector

Figure 91 - Isolator Card Connectors

B.2 2 Motor Amplifier Connector

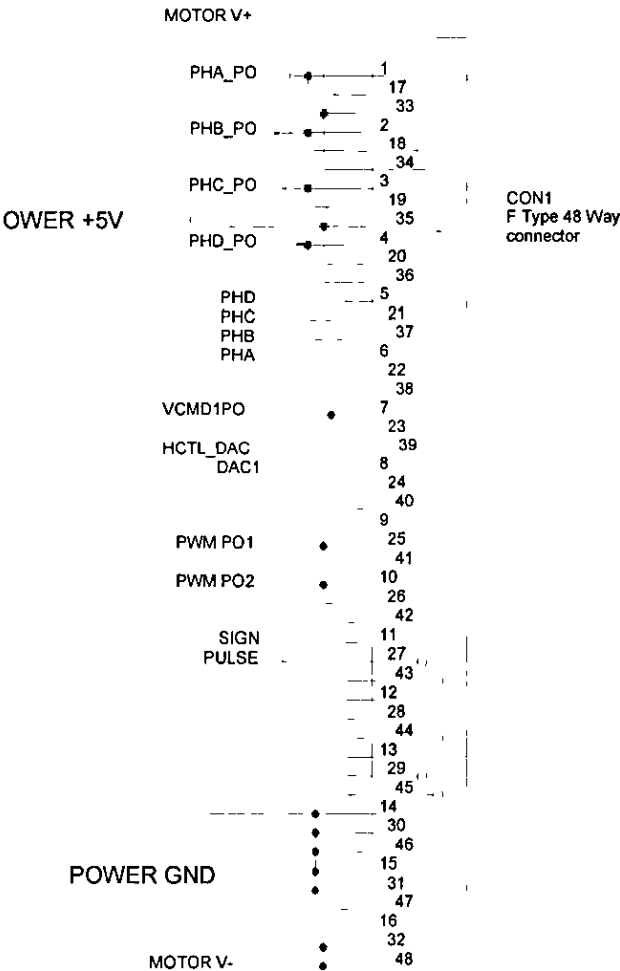


Figure 92 - Motor Amplifier Connector

Appendix C **Control Development Studio**

Programming/User Guide

C.1. Getting Started

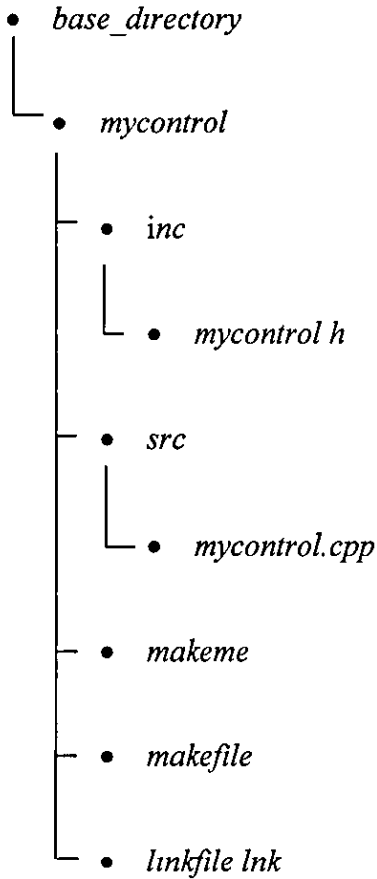
Turn the motion control PC on. When the Photon Login appears, login as *root* (no password is required). Once Photon has started, select Control Tools from the Task Bar and click on the Icon for the CDS.

The sequence that a user needs to follow in order to run a control program is as follows:

- 1 Using the CDS the user selects the desired control program from a file-list
2. The CDS spawns the selected program (starts it running) as a separate process.
3. The control program immediately starts communication with the CDS in order to relay information about data logging and control variables.
4. Once this operation is complete the user can customise the control by changing how logged data is to be displayed in the graph and altering the parameters from the control variable list
5. In order to start/stop the actual control loop being performed (i.e. enabling the desired interrupt source) the user left-clicks with the mouse on the "Start Control" button (while the control is running the label on the button is changed to "Stop Control")
- 6 The control loop is performed every time an interrupt occurs and until the user left-clicks the "Stop Button"

C.2. Generating a new control program

By selecting "New Control" from the "Control" menu a file selection dialog is displayed. Select the desired base directory where the new control program is to be generated. Also type the name for the program (e.g. *mycontrol*) in the dialog box and click "Done". A number of files are then generated in a directory that has the following structure:



The files *mycontrol.h* and *mycontrol.cpp* contain the base C++ code that is required to fully generate a control. The example included in the code, uses the 20ms internal clock interrupt to generate and log a sine wave. The CDS also compiles, links and automatically loads this control (called *./base_control/mycontrol/mycontrol*)

C.3. Building a control program

The control program requires that there are at least the following four functions available somewhere in the code files

- (i) InitControl()
- (ii) StartControl()
- (iii) DoControl()
- (iv) StopControl()

These four functions are described in more detail below. The code provided by these functions is linked with the base control library (*basecontrol lib*) which provides functions for communicating with CDS. In order to build the control program select "Make" from the "Control" menu. The control program is then compiled and linked and any errors are reported. The control program is not automatically reloaded.

C.4. Running a control program

To select the control that needs to be run, click on the 'Open' command from the 'File' menu. Select the desired program from the file selection dialog.

C.5. Saving Logged Data

Once the control has been run, the user is able to save the data in a tab-delimited text format. This file can be copied to a Windows machine and opened using Excel. Select 'Save' from the 'File' menu using the mouse.

C.6. Editing a control program

In order to be able to generate the various types of control that may be required, it will be necessary for the programmer to develop and insert code in to the four main functions described above. To enable quick and effective editing, the CDS uses a generic text editor (*vedit*). To start the editor select "Edit" from the "Control" menu. The editor is automatically started and the programmer will need to load and edit the C++, header, make and link files. For more help on editing make and link files the programmer is referred to the online QNX and Watcom C/C++ help.

C 6 1 int InitControl(void)

In order to register and pre-initialise the log and control variables the control program calls this function. Calls to *AddLogVariable()* and *AddControlVariable()* should be made here. The *InitControl()* function also serves to inform the control program which interrupt is to be used in the control algorithm by returning the interrupt number. By returning the number -1, the default 20ms system timer is used as the interrupt source.

C 6 2 void StartControl(void)

This function is called when the user clicks the start button. This function allows the programmer to initialise control and log variables before the interrupt (specified by the *InitControl* function) is enabled. If for example, the interrupt is to be generated from a programmable timer, the programmer might set the interrupt frequency using this function. The programmer should also initialise any electronic outputs here.

C.6 3 void DoControl(long count)

The function *DoControl* is called every time an interrupt is received. The variable *count* contains the number of times the interrupt has been triggered. The programmer should be aware that the control program does not check to see if time taken to perform the interrupt takes longer than the time between interrupts. If data needs to be recorded it should be sampled at the end of *DoControl* using the *LogVariables* function. The function *DisplayUserMessage* can also be called here to display information to the user or for debugging purposes.

C 6 4 void StopControl(void)

This is called once the user clicks on the 'Stop' button in the CDS. The programmer should put all hardware into a safe state in this routine, i.e. make sure all motors are

turned off etc. It is also possible to call the function *UpdateControlVariable* in order to force the CDS to update any control variables that have been changed during

*C.6.5 int AddLogVariable(char *buffer, double *pfVar, long clr)*

The *AddLogVariable* function is used to inform the control program and the CDS that the variable (of type *double*) specified by the memory address *pfVar* is to be recorded every time the function *LogVariables* is called. The text given in the argument *buffer* is only used in the CDS as a label for the button in the log variable window. The variable *clr* is used to give the colour that should be used to draw the plot of the variable *pfVar*. The colour that is to be used for the plot, *clr*, is specified using a standard 32-bit hexadecimal colour code of the form 0x00rrggbb. If the operation was successful the function returns the total number of log variables that have been declared otherwise it returns -1.

C.6.6 void LogVariables(void)

Forces the control program to send a list of the current values of all registered log variables to the CDS. The CDS then stores these values for plotting.

*C.6.7 int AddControlVariable(char *buffer, double *pfVar)*

This function is used to inform CDS that the user can change the variable given by the pointer *pfVar* from the CDS. The text given by *buffer* is only used by the CDS in the control variable window. By clicking on the respective entry in the control variable window the user can edit the desired variable. This is particularly useful when tuning of control gains, as the control does not need recompiling to change only variable. Again, if the function was successful it returns the total number of control variables that have been declared otherwise it returns -1.

*C.6.8 void UpdateControlVariable(double *pfVar)*

The *UpdateControlVariable* function allows the programmer to force the CDS to update the value of the variable in the control variable window. The CDS is informed

using the message passing connection and it is not recommended that this function be used in the function *DoControl*.

*C 6 9 void DisplayUserMessage(char *pstr)*

This function is used to display the NULL terminated text string given by *pstr* in the log messages window. This function is useful for displaying status messages to the user or debugging information for the programmer

C.7. Adding Library and C++ Source files

In order to add code from third party sources or to help organise the code in a more efficient manner it may be necessary to add many library and C++ source files to the

C.7.1 Library Files

In order to add a library file (**lib*) to the project modifications to the linker file are required. Open the file */base_directory/linkfile lnk* and add full name (including the path) of the library file in the line

```
'file ~fullpath/libraryname.lib
```

at the end of the list of files. An example is given below. The control program can now be rebuilt using the *make* command from the control menu. The new library file will be linked into the executable

C.7.2 C++ Source Files

To add C++ source files (**cpp*) to the list changes to both the make and link files are required. Firstly, open the file */base_directory/makefile*. The make file needs to be told how to deal with the new file. Copy the C++ source file to the */base_directory/src/* directory (for tidiness) and at the end of the make file add the line:

```
srcfile.o: <tab> ./src/srcfile.cpp [ + list of dependencies]
```

where *srcfile.o* is the name of the object file that will be created from the source file *srcfile.cpp*. One other change also needs to be made to this file. This command informs the compiler of the dependency of the files on the main program. The line:

```
main: ./src/myCT.o
```

should be modified to include the name of the new object file. The addition is shown in below in bold letters.

```
main: ./src/myCT.o ./src/srcfile.o
```

The list of dependencies is a list of files (including the pathnames) that the source file depends on. See the Watcom C++ Help on make files for more information.

C 7.3 An example make file

The following example is the make file for the 1-DOF control program. In this program a file called *filter.cpp* was added to the project. The bold letters show the changes that were made in the code sample below.

```
# Make file for control project myCT
# Generated - Fri Jan 5 10:13:39
.SUFFIXES: .c .cpp .o
CPP = wpp386

INC = /usr/watcom/10.6/usr/include
CPPOPTS = -zu -s

.cpp.o:
    $(CPP) -I$(INC) $(CPPOPTS) $<
.c.o:
    wcc -zu "-Wc,-s" -Tl $<

main: ./src/myCT.o ./src/filter.o

    wlink @linkfile.lnk

myCT.o:    ./inc/myCT.h

filter.o:    ./inc/filter.h ./src/filter.cpp
```

C.7 4 An example link file

Along with the source file that was included in the *myCT* control program a library file for the Single Axis Interface Card was also included. Again the bold lettering in the code sample below show the changes that were required in order to successfully build and compile the program.

```
file myCT
file filter
file /home/mcavcr/ControlDevStudio/Control/basecontrol.lib
file /home/mcavcr/SAMC/lib/samccl.lib

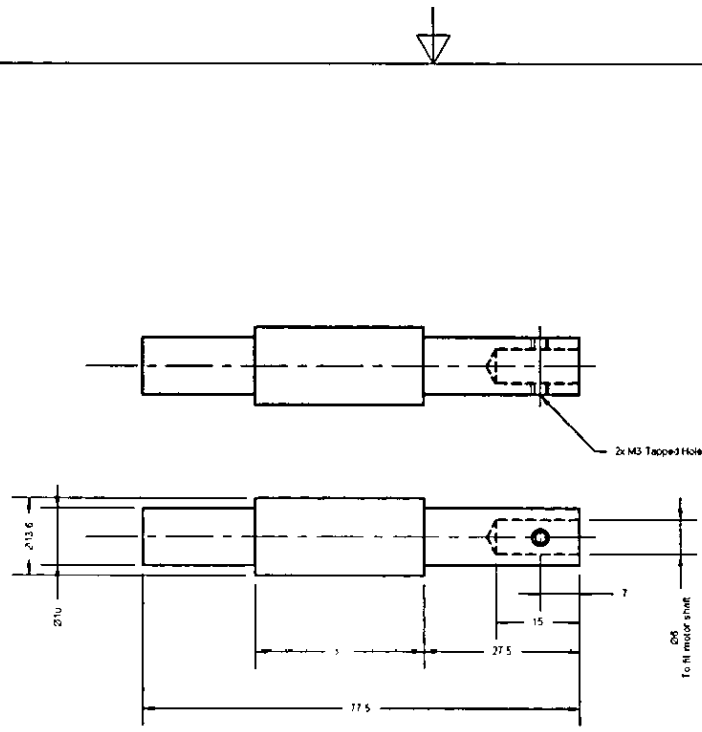
libpath /usr/watcom/10.6/usr/lib
op map=myCT.map
op priv=1

format qnx
name myCT.ct1
```

Appendix D Engineering Drawings

D.1. Worm Drive Units

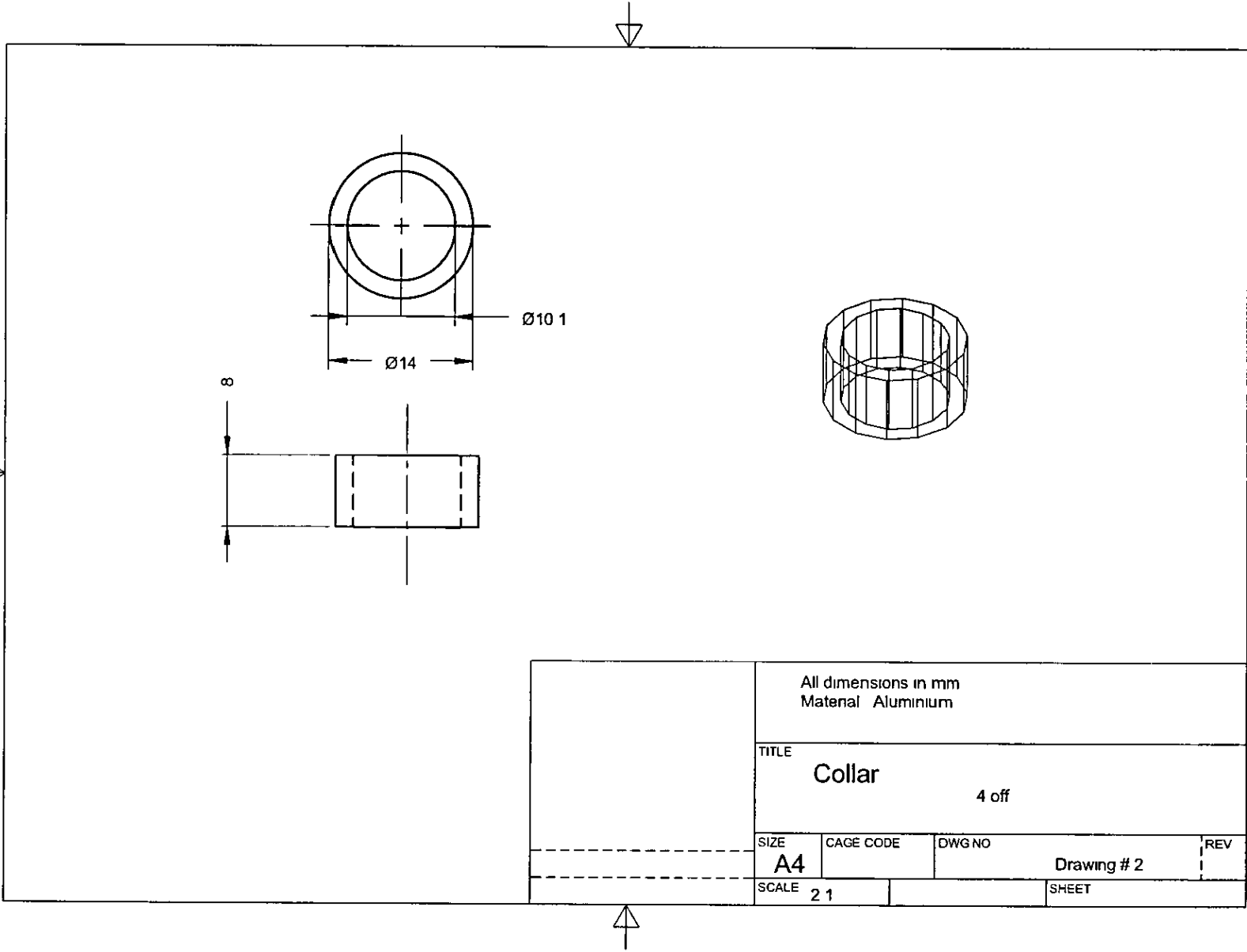
| Drawing # | Title | Qty for 1 unit |
|-----------|-------------------------------------|----------------|
| 1 | Worm modifications | 2 |
| 2 | Worm Collar | 4 |
| 3 | Alpha Worm Shaft Hanger (motor) | 1 |
| 4 | Alpha Worm Shaft Hanger (non-motor) | 1 |
| 5 | Beta Worm Shaft Hanger (motor) | 1 |
| 6 | Beta Worm Shaft Hanger (non-motor) | 1 |
| 7 | Motor Mount | 2 |
| 8 | Alpha Drive Assembly | 1 |
| 9 | Beta Drive Assembly | 1 |

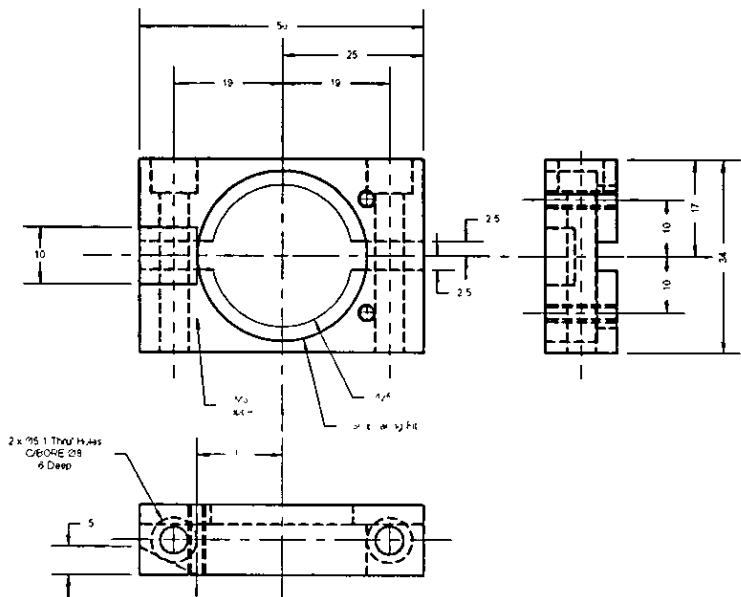


Bought in component
Modified as per drawing

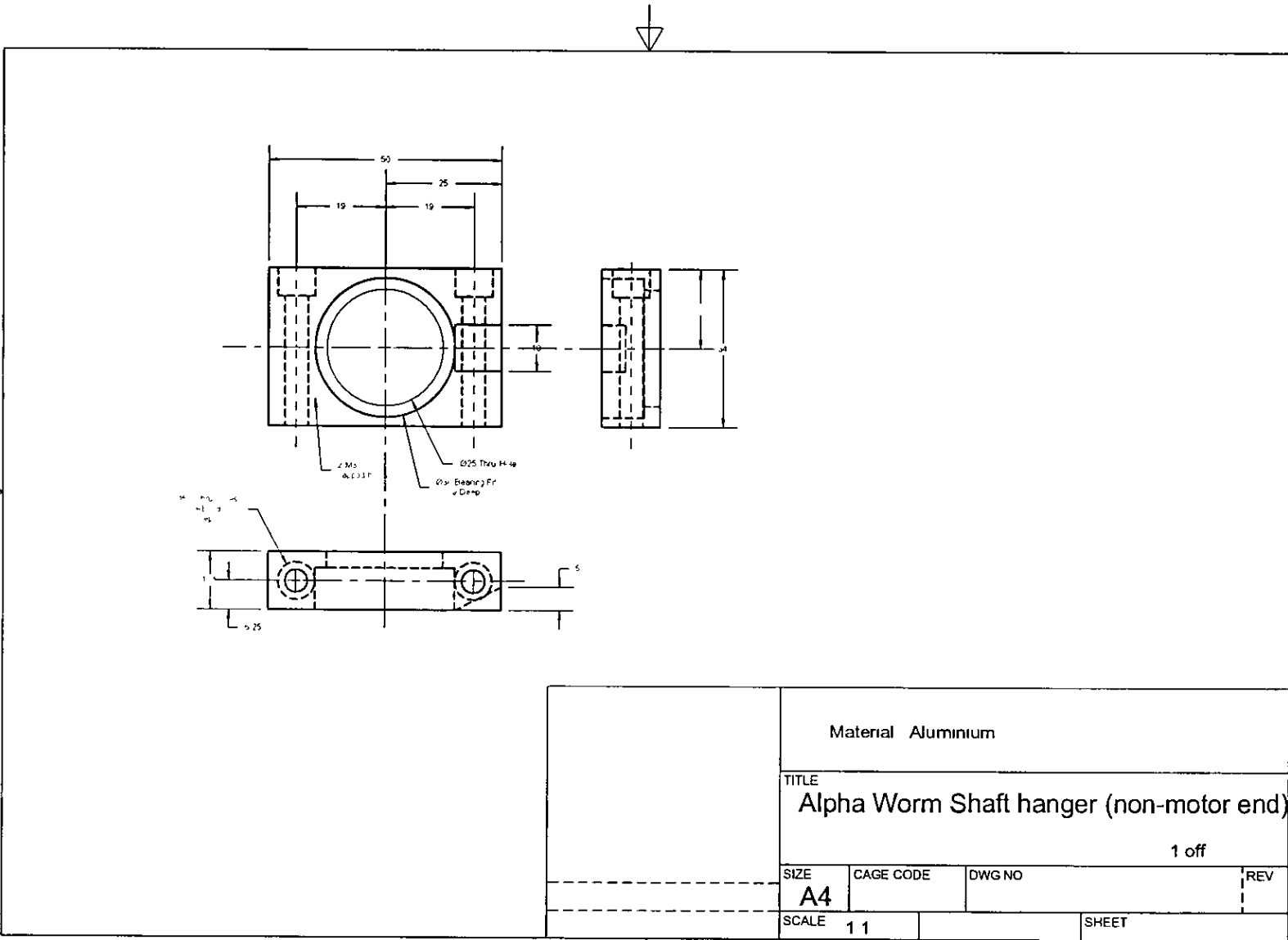
HPC Gears Order Code SW0 8-1
Single Start 0.8MOD Worm

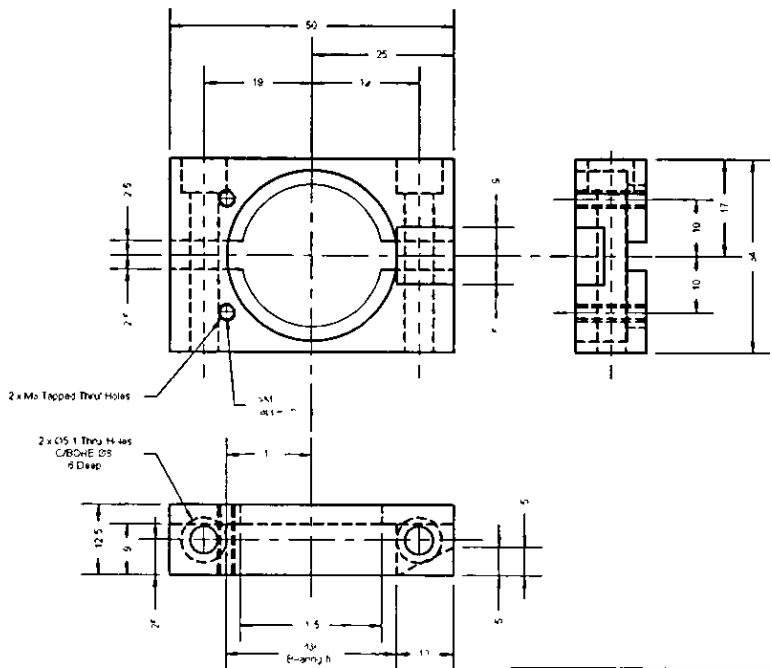
| | | | |
|--------------------------|-----------|-------------|-----|
| Material Steel | | | |
| TITLE | | | |
| Worm Modifications 2 off | | | |
| SIZE | CAGE CODE | DWG NO | REV |
| A4 | | Drawing # 1 | |
| SCALE | 2 1 | SHEET | |



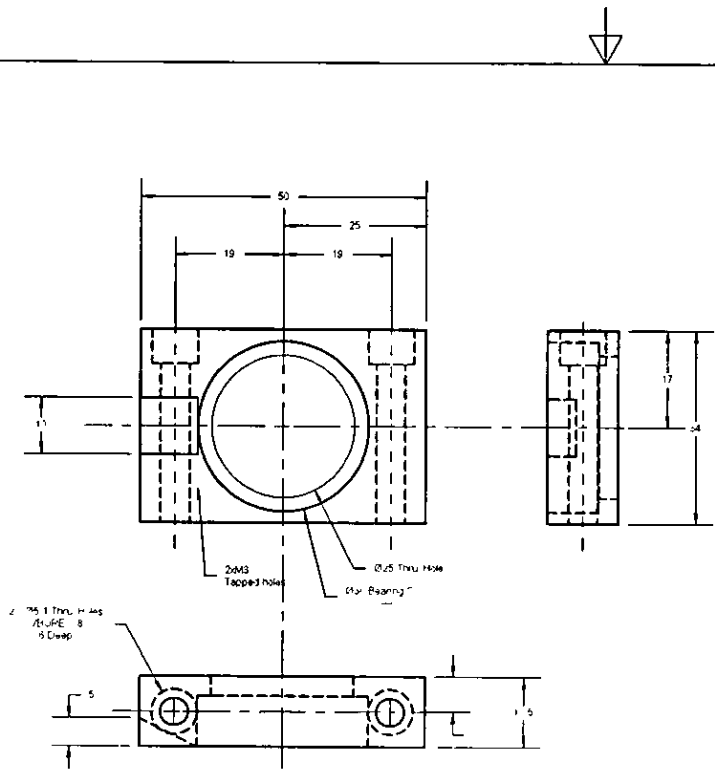


| | | | |
|---|-----------|-----------------------|-----|
| Material Aluminium | | | |
| TITLE Alpha Worm Shaft Hanger (motor end) 1 off | | | |
| SIZE A4 | CAGE CODE | DWG NO Drawing # 3 | REV |
| SCALE 1 1 | SHEET | | |



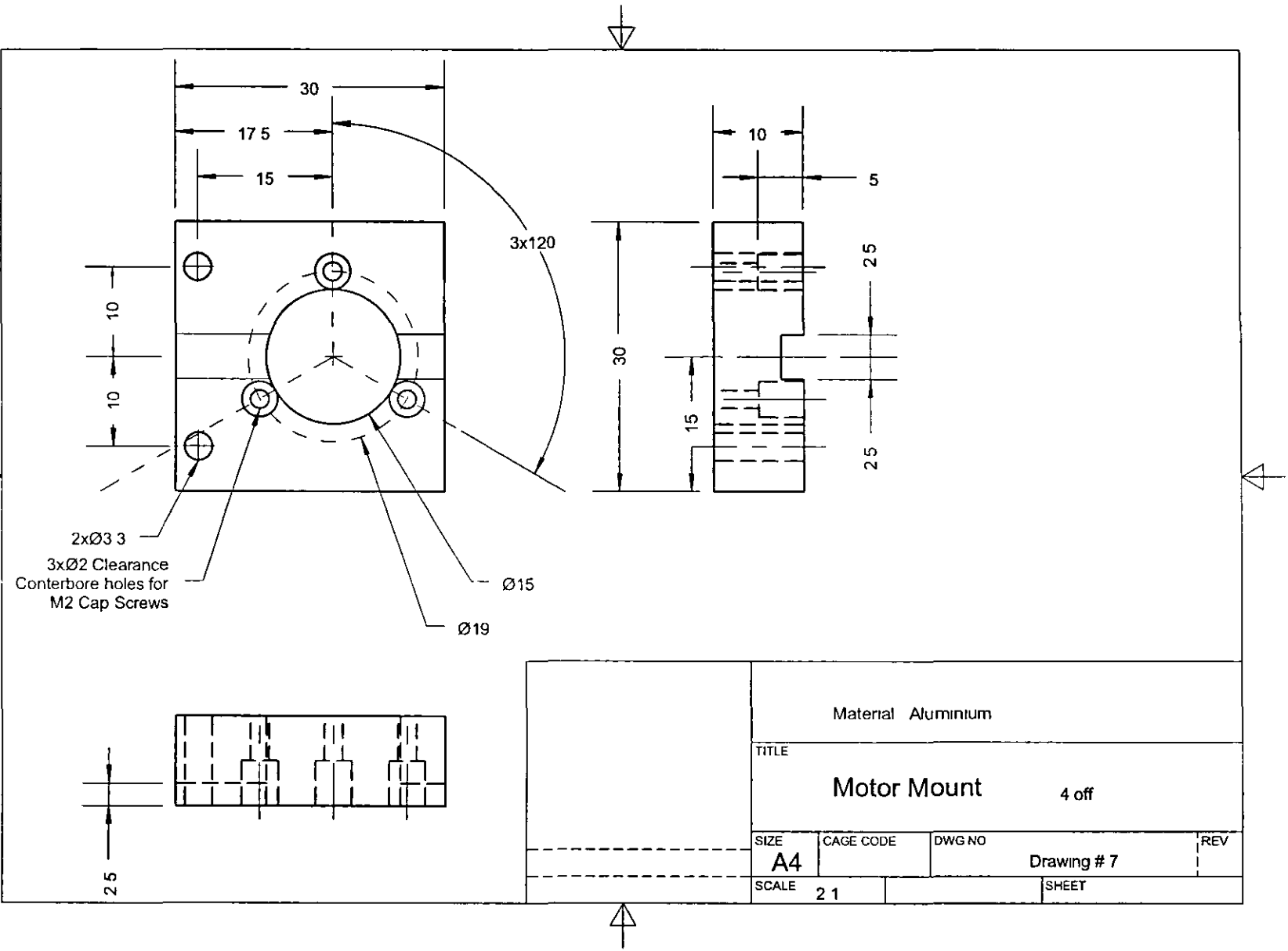


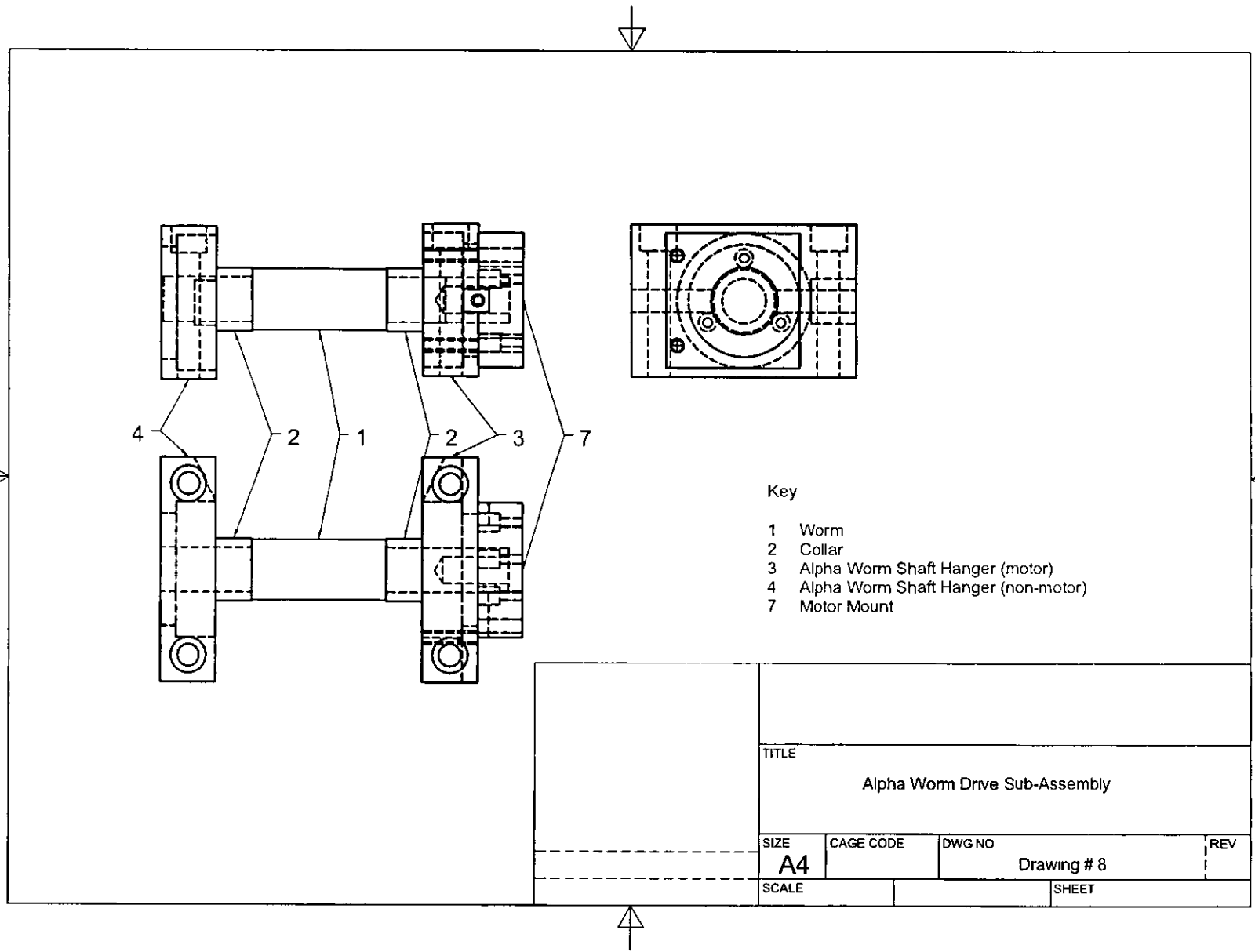
| | | | |
|------------------------------------|-----------|-------------|-----|
| Material Aluminium | | | |
| TITLE | | | |
| Beta Worm Shaft hanger (motor end) | | | |
| 1 off | | | |
| SIZE | CAGE CODE | DWG NO | REV |
| A4 | | Drawing # 5 | |
| SCALE | 1 1 | SHEET | |

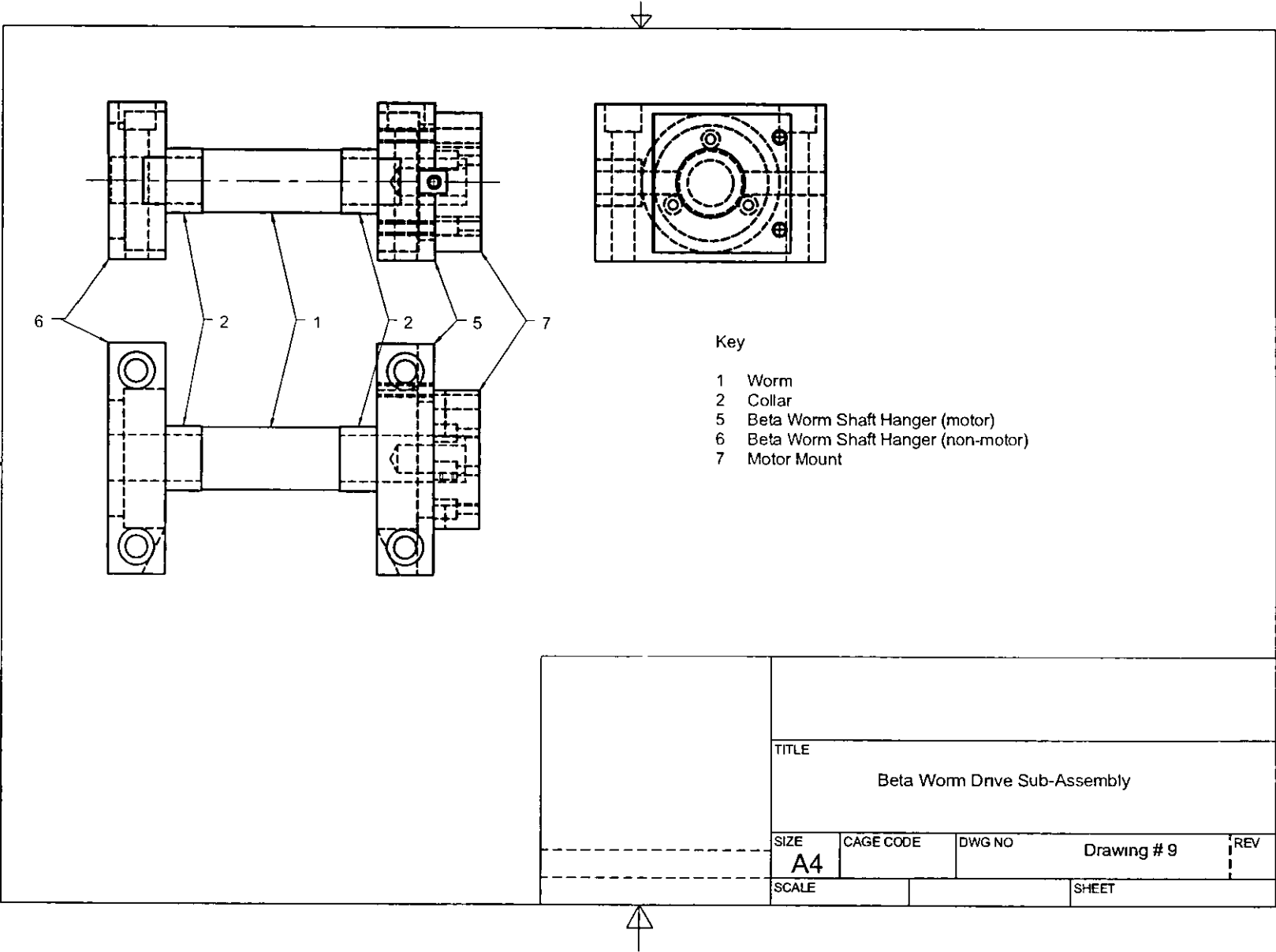


| | | | |
|--|-----------|-----------------------|-----|
| Material Aluminium | | | |
| TITLE Beta Worm Shaft hanger (non-motor end) 1 off | | | |
| SIZE A4 | CAGE CODE | DWG NO Drawing # 6 | REV |
| SCALE 1 1 | SHEET | | |

250

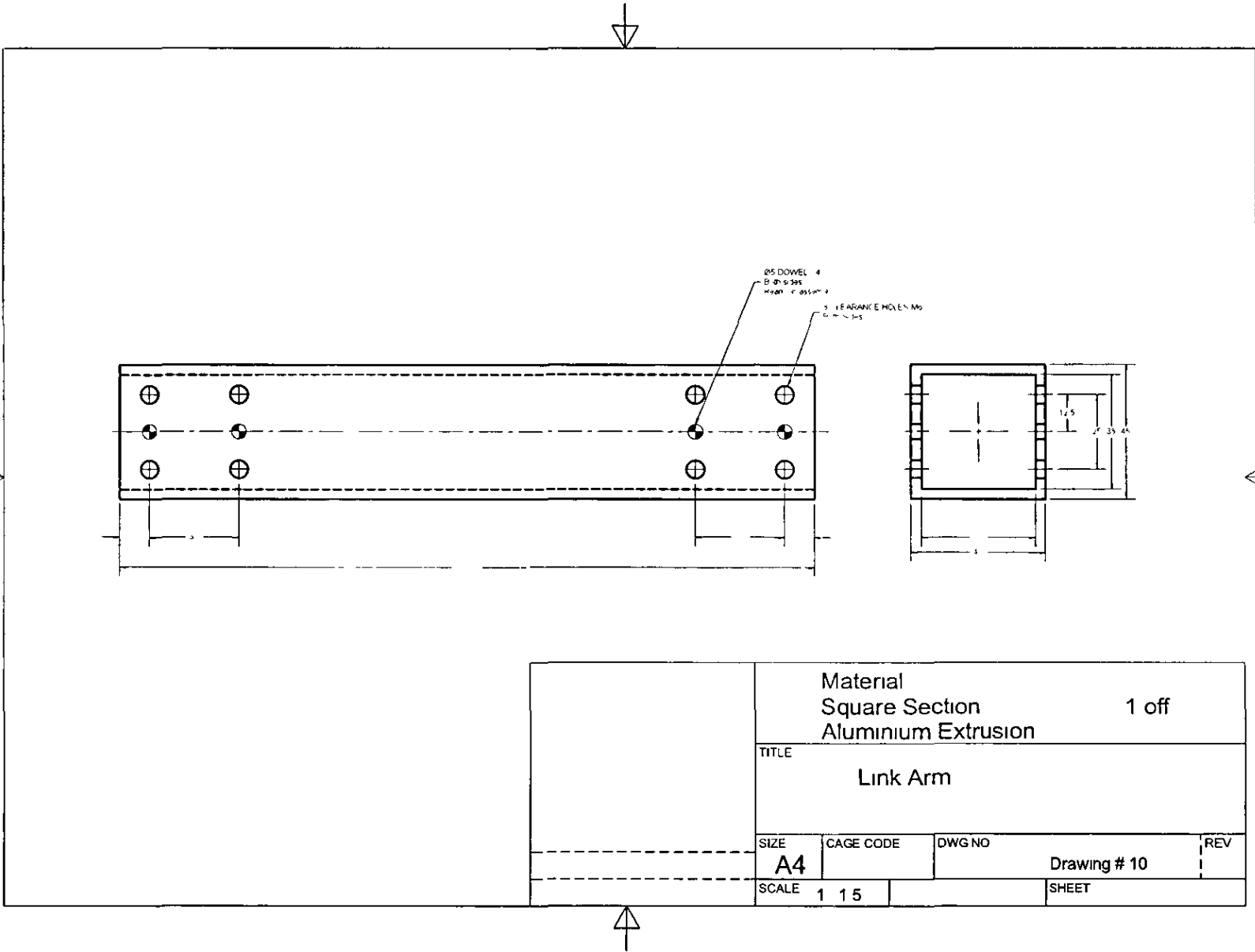




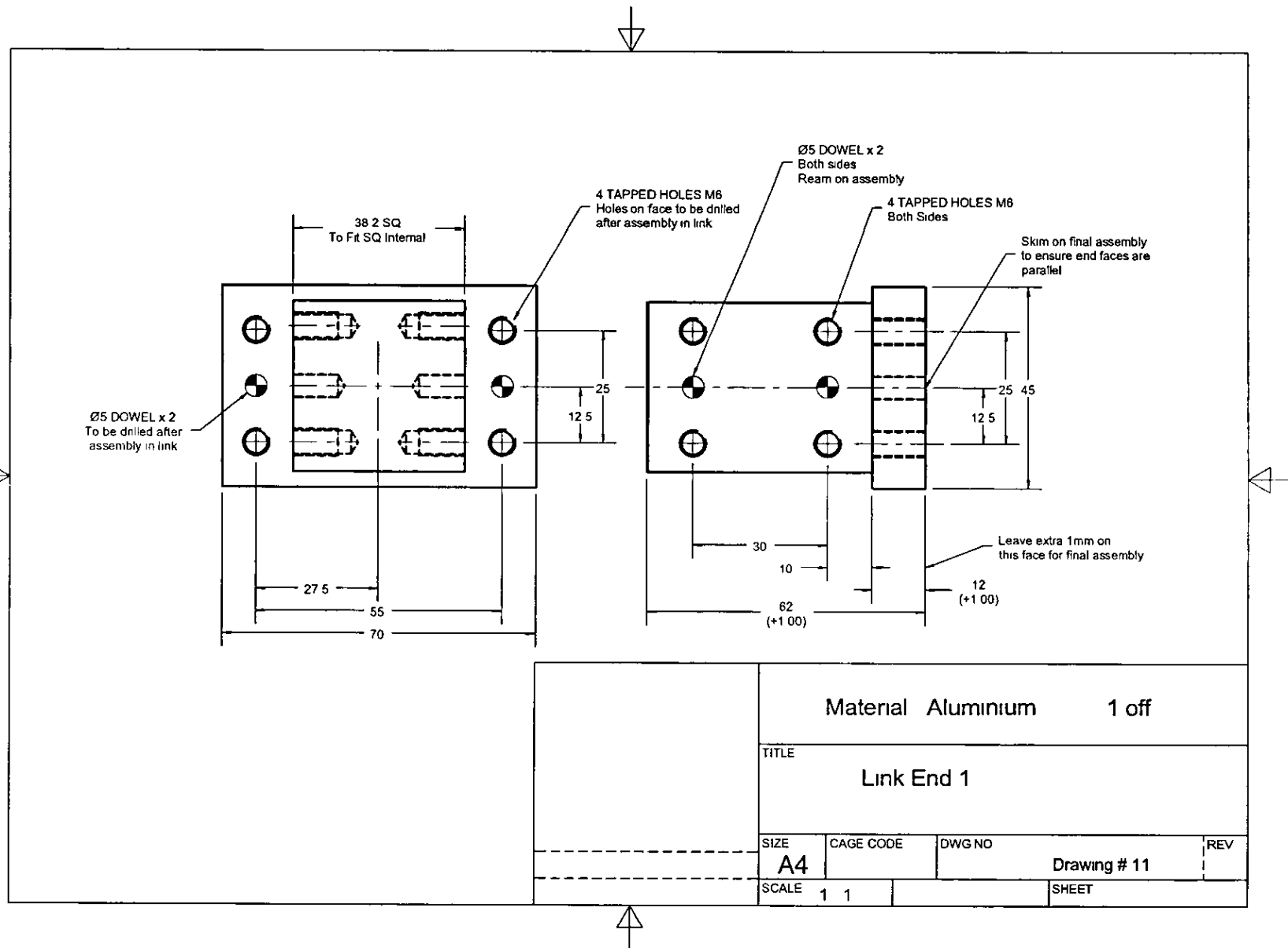


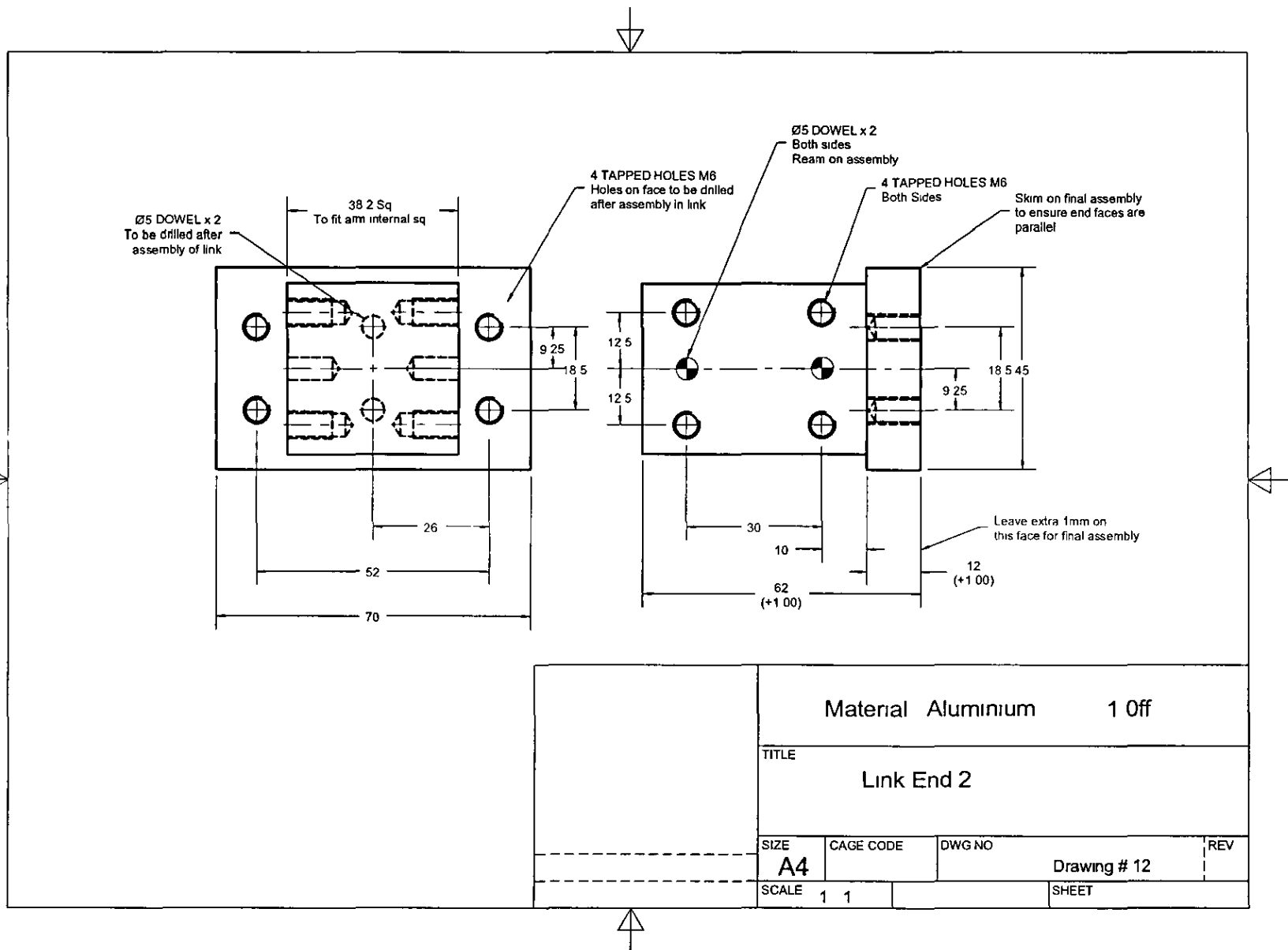
D.2. Link Assembly

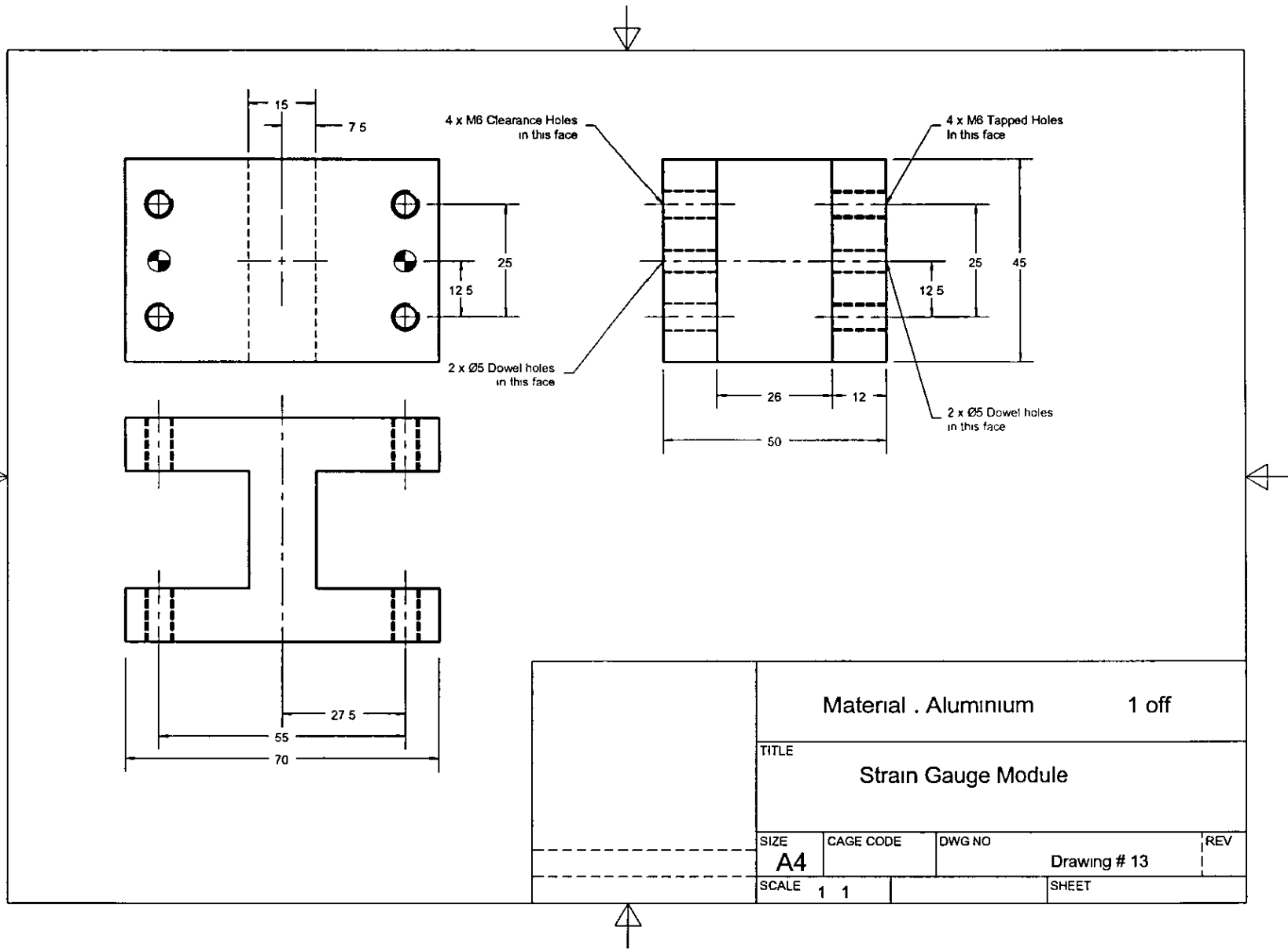
| Drawing # | Title | Qty for 1 unit |
|-----------|---------------------|----------------|
| 10 | Link Arm | 1 |
| 11 | Link End 1 | 1 |
| 12 | Link End 2 | 1 |
| 13 | Strain Gauge Module | 1 |
| 14 | Link Assembly | 1 |

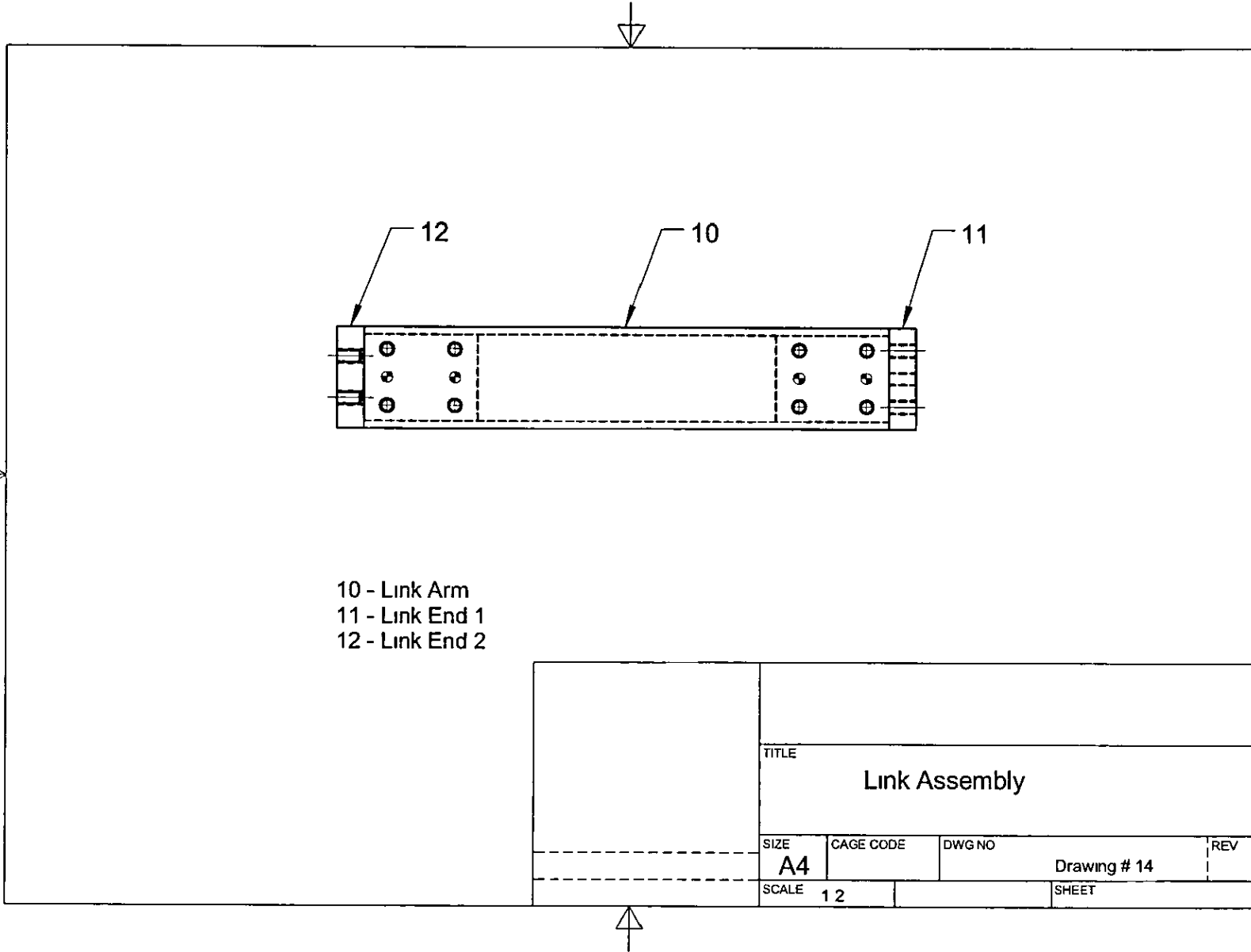


254



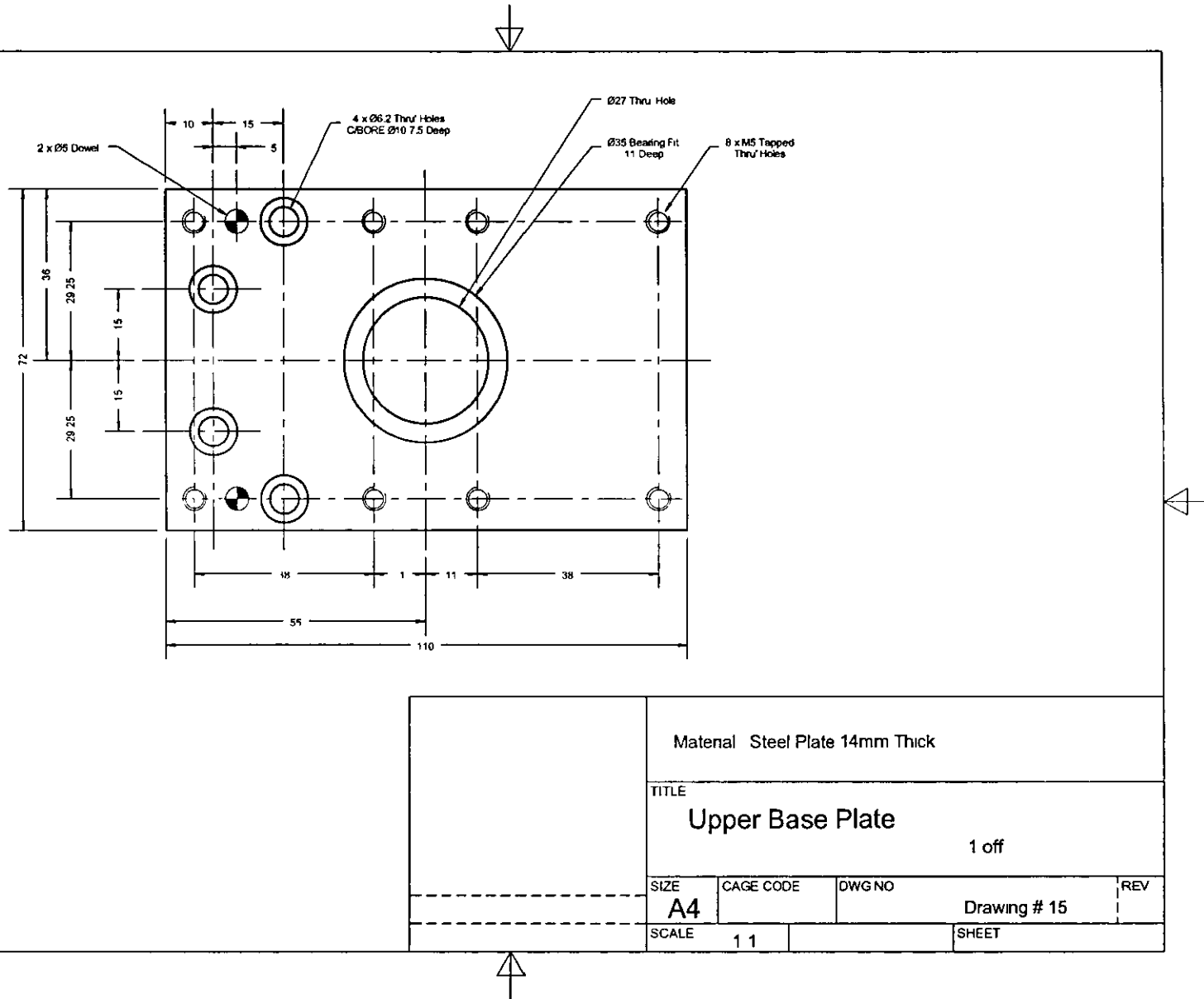


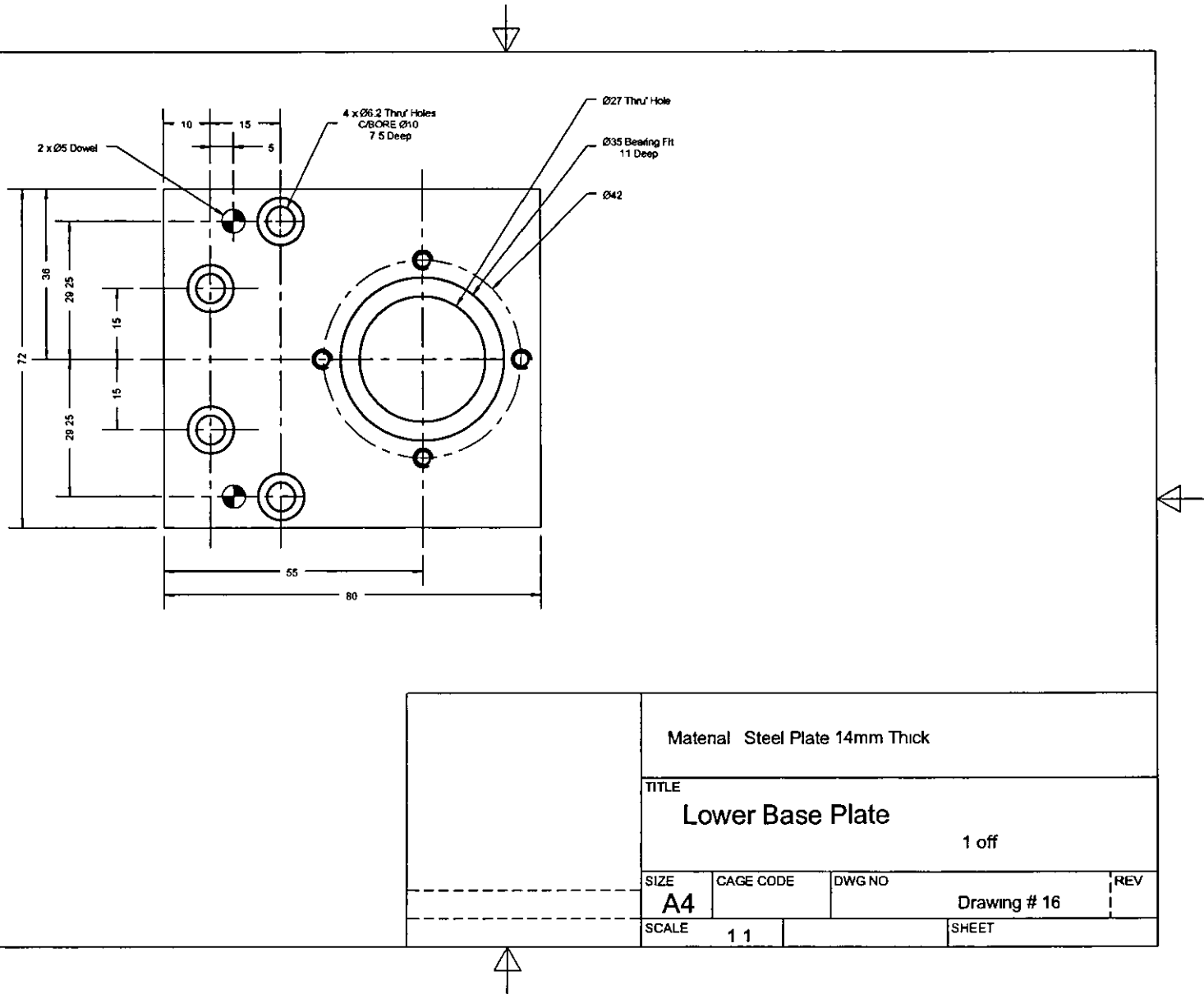


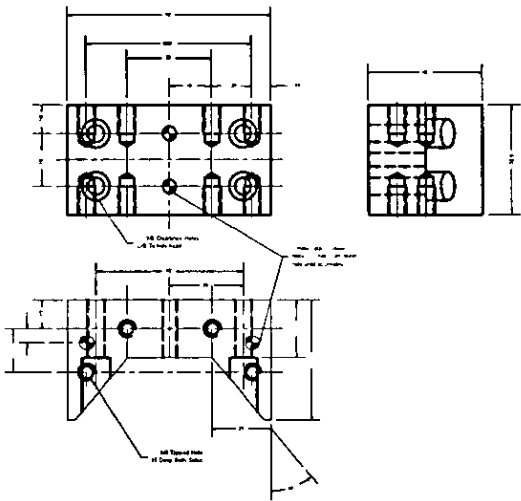


D.3. Joint Assembly

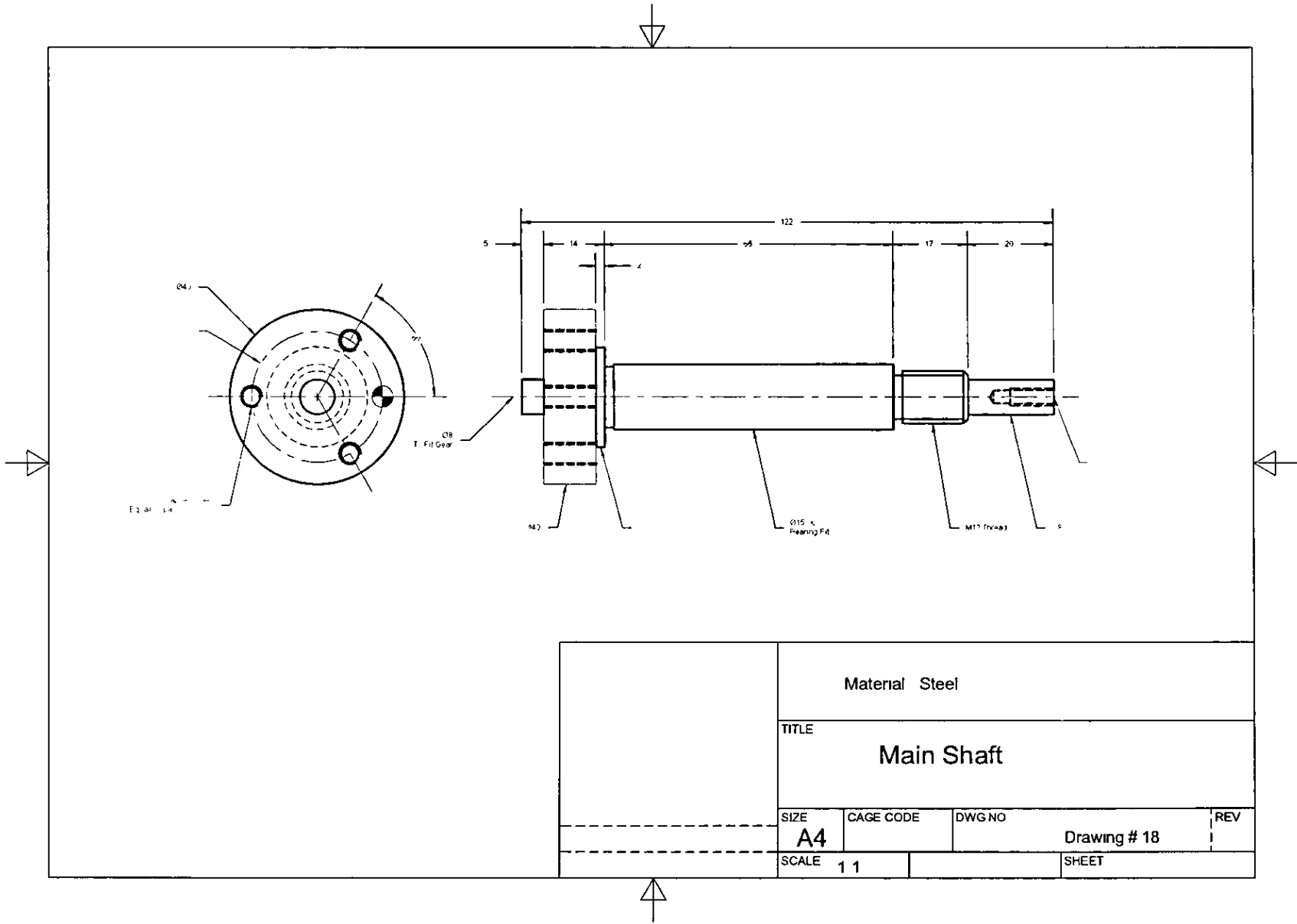
| Drawing # | Title | Qty for 1 unit |
|-----------|-------------------------|----------------|
| 15 | Upper Base Plate | 1 |
| 16 | Lower Base Plate | 1 |
| 17 | Joint Base Unit | 1 |
| 18 | Main Shaft | 1 |
| 19 | Stock | 1 |
| 20 | Link Mounting plate | 1 |
| 21 | Encoder Anchor Pin | 1 |
| 22 | Encoder Bush | 1 |
| 23 | Worm Wheel Modification | 1 |
| 24 | Joint Assembly | 1 |

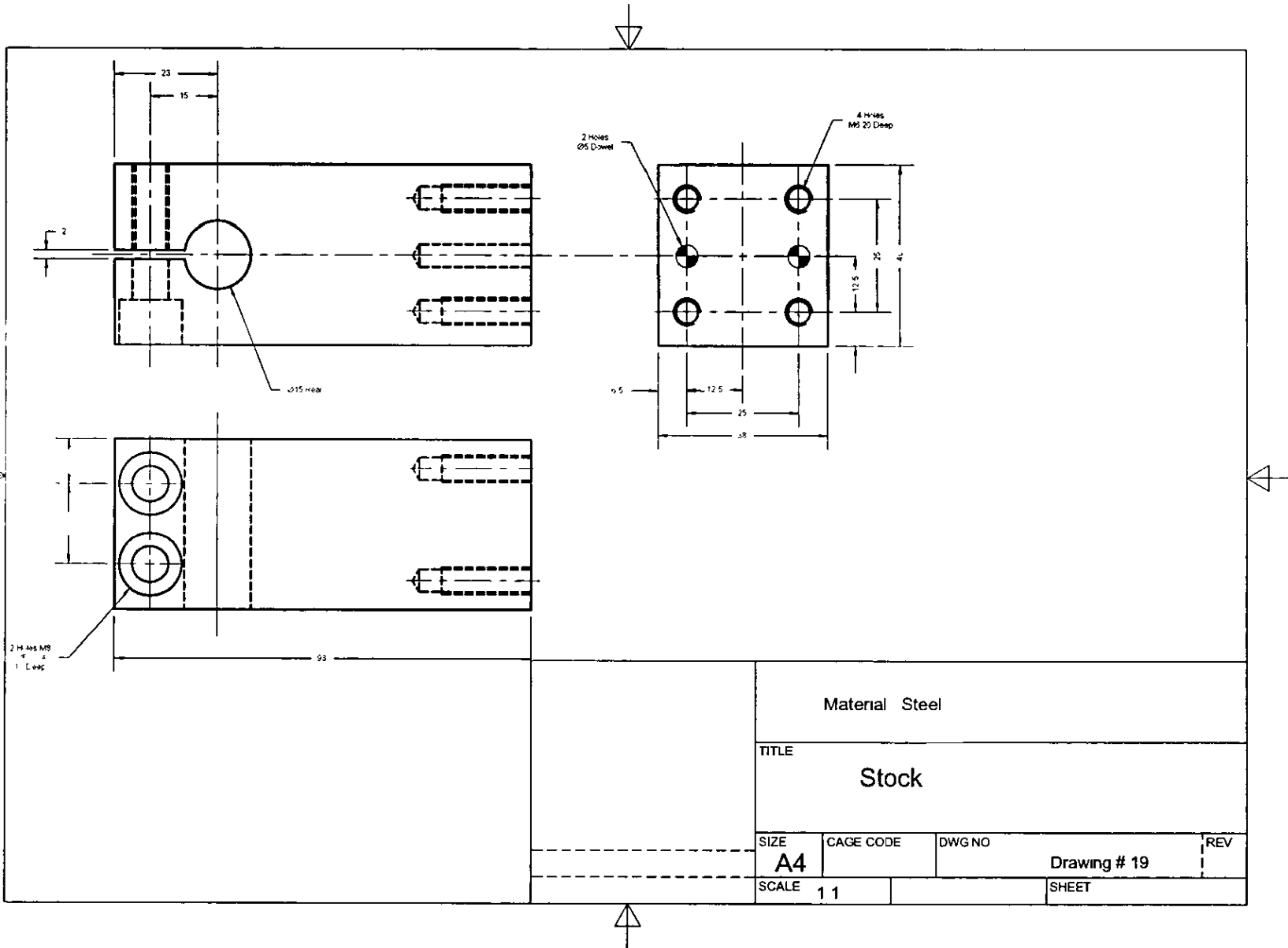


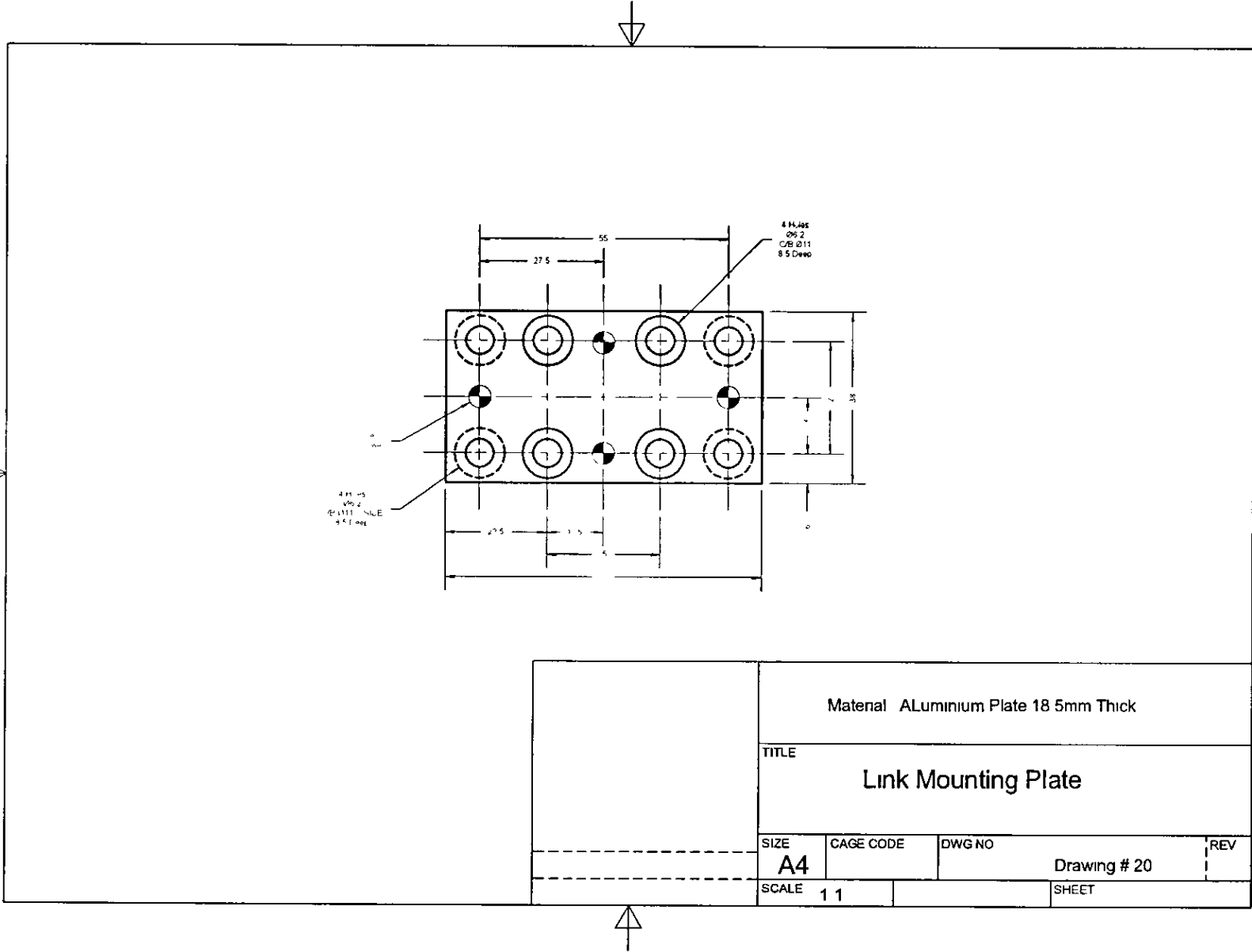


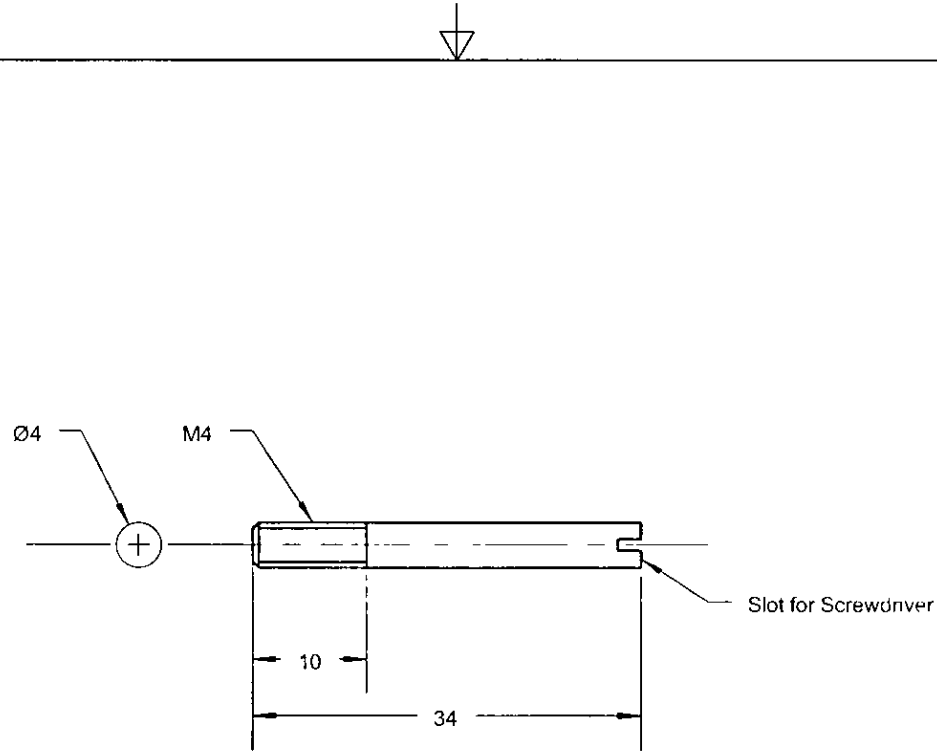


| | | | |
|-----------------|-----------|--------------|-------|
| Material Steel | | | |
| TITLE | | | |
| Joint Base Unit | | | 1 off |
| SIZE | CAGE CODE | DWG NO | REV |
| A4 | | Drawing # 17 | |
| SCALE | 1 2 | SHEET | |

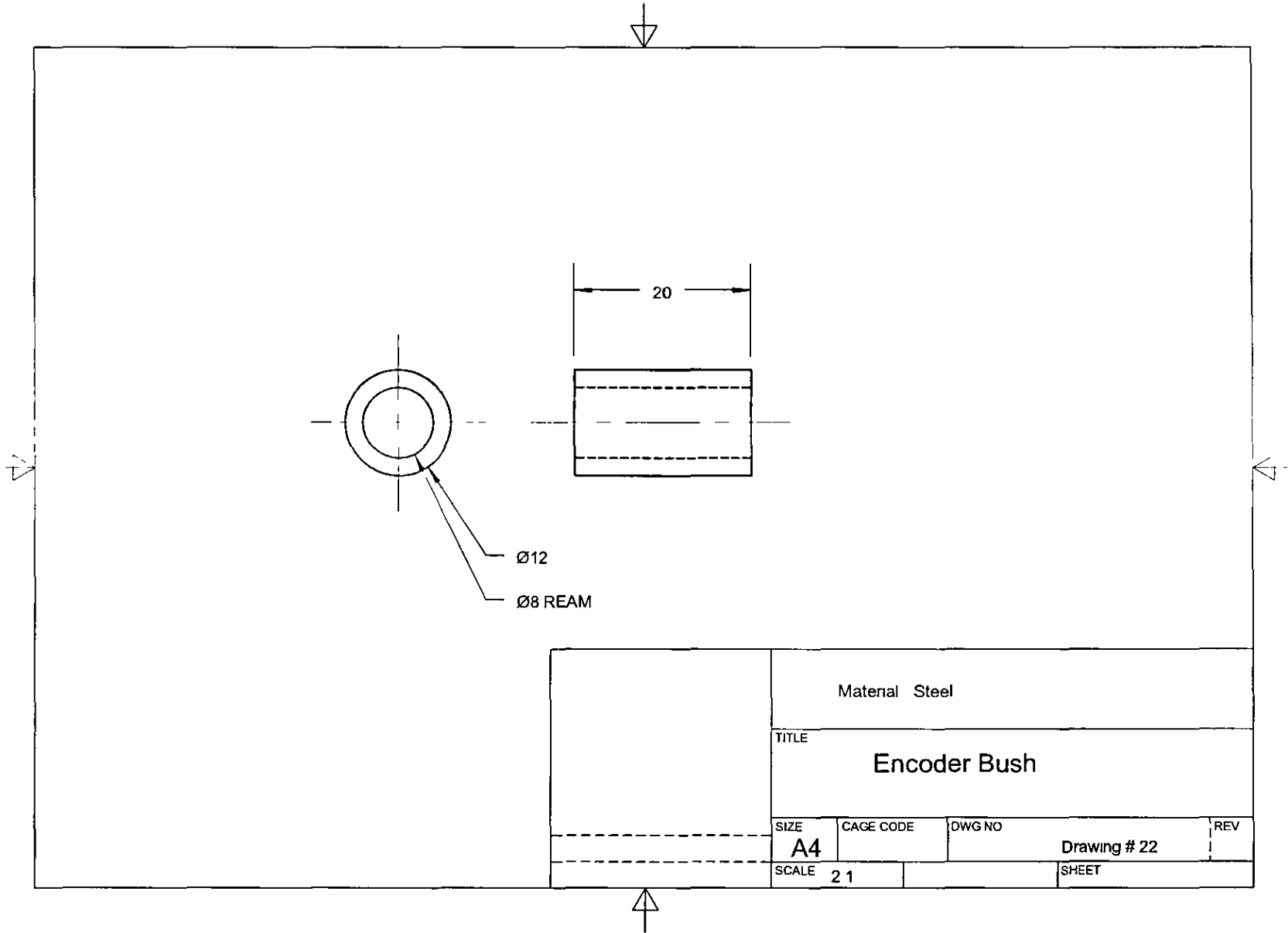


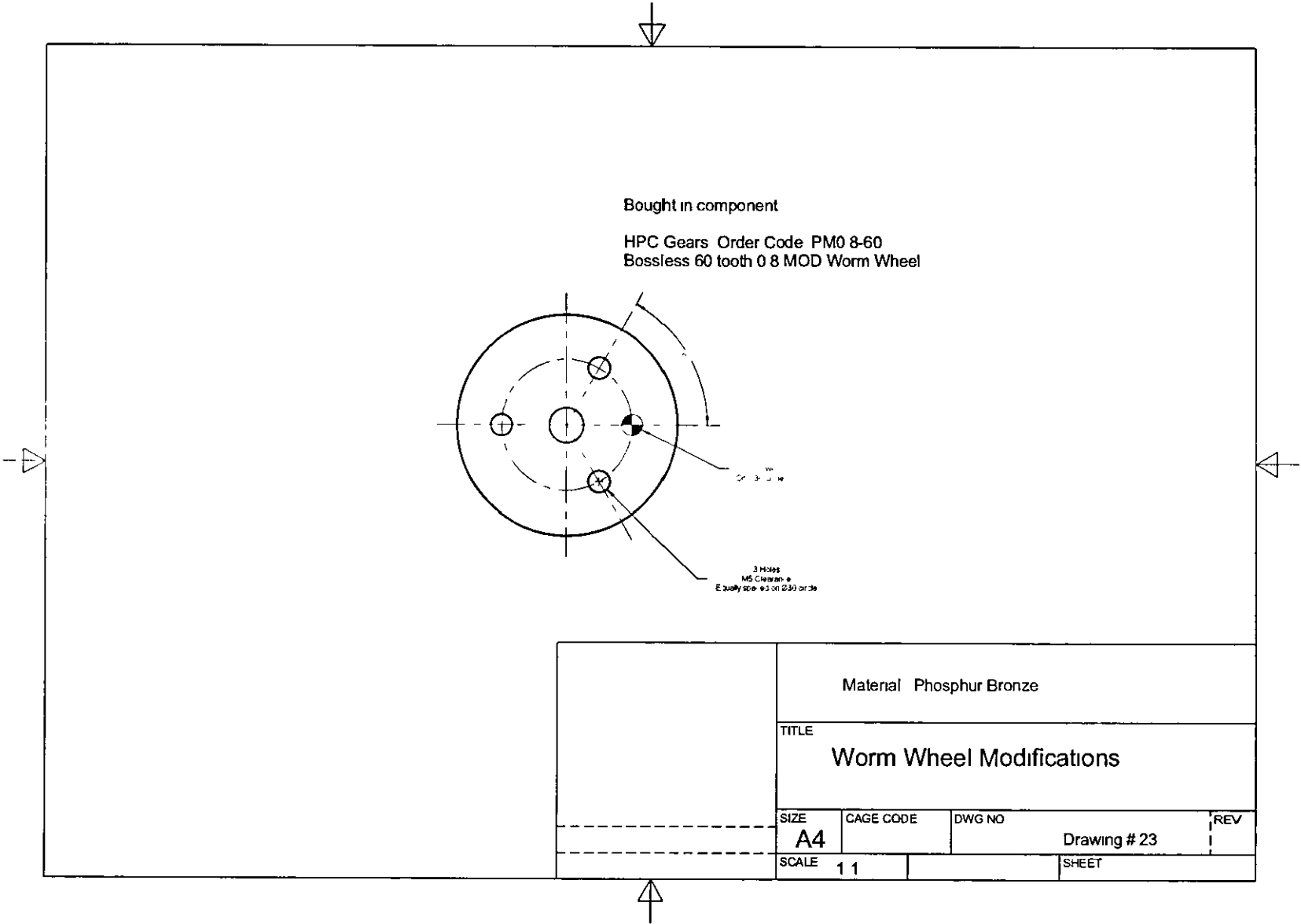


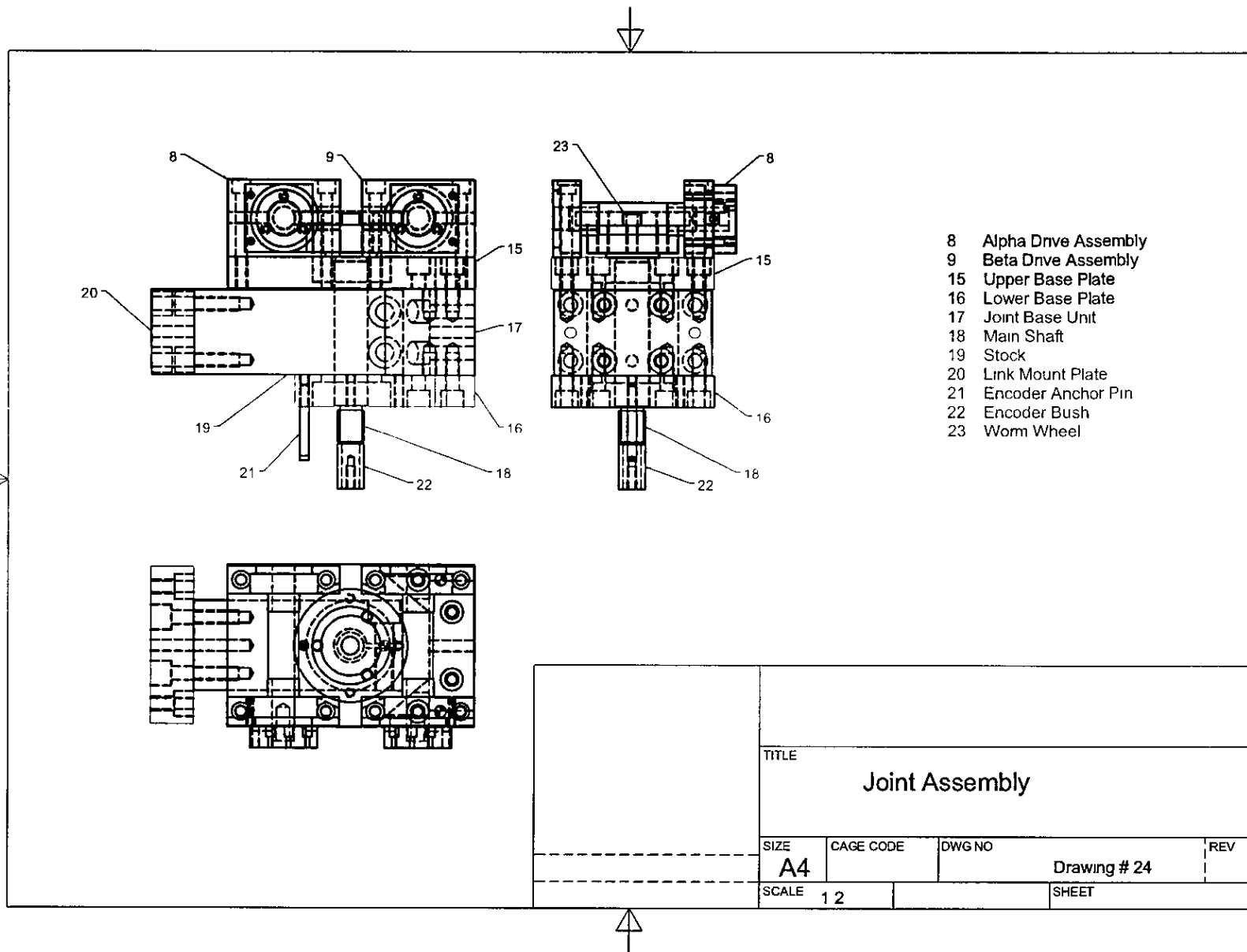




| | | | |
|-----------------------------|-----------|------------------------|-----|
| Material Steel | | | |
| TITLE Encoder Anchor pin | | | |
| SIZE A4 | CAGE CODE | DWG NO Drawing # 21 | REV |
| SCALE 2 1 | SHEET | | |

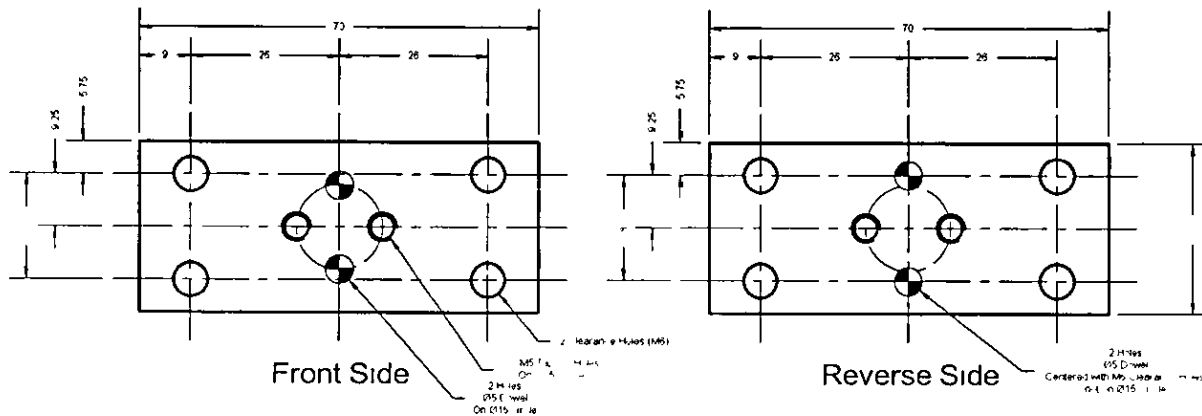




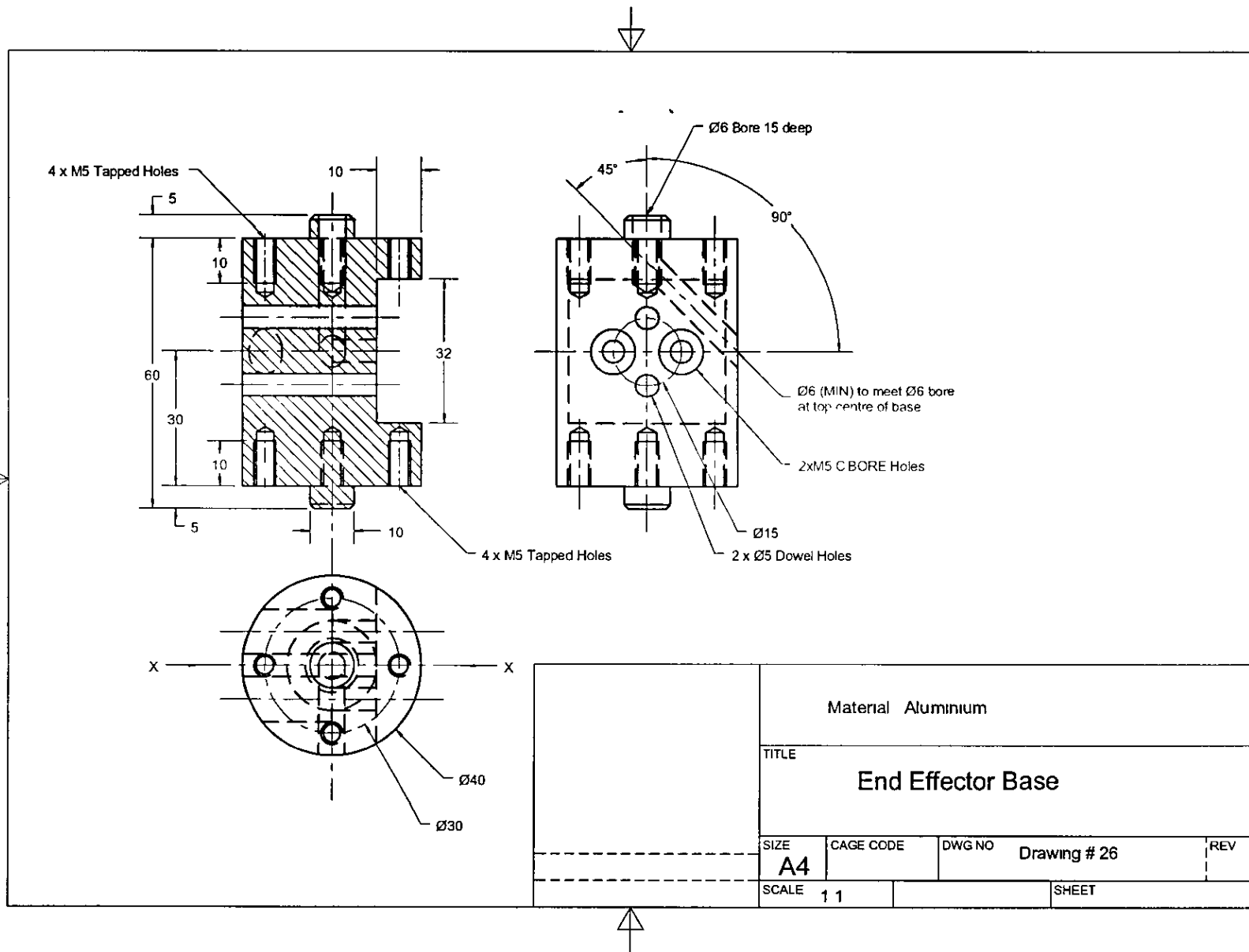


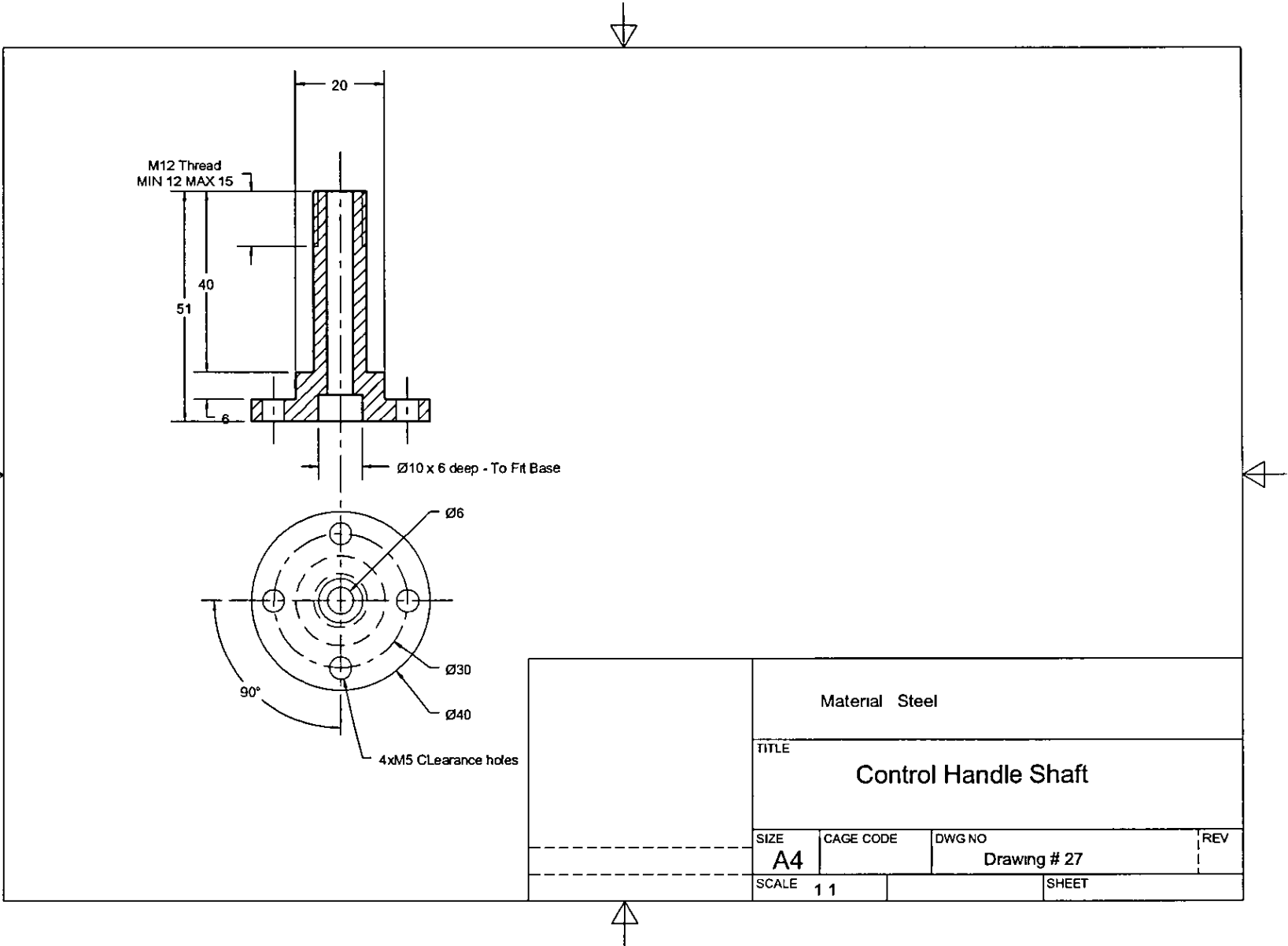
D.4. Control Handle Assembly

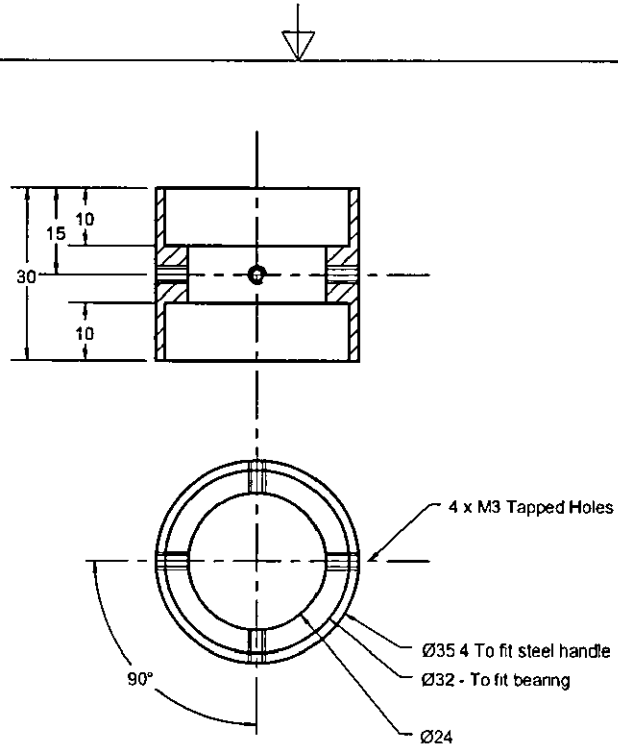
| Drawing # | Title | Qty for 1 unit |
|-----------|------------------------------------|----------------|
| 25 | End Effector Mounting Plate | 1 |
| 26 | End Effector Base | 1 |
| 27 | Control Handle Shaft | 1 |
| 28 | Control Handle Bearing Housing | 1 |
| 29 | Control Handle Motor Housing | 1 |
| 30 | Control Handle Motor Mount | 1 |
| 31 | Pointer Rotor | 1 |
| 32 | Control Handle Sleeve | 1 |
| 33 | Control Handle Optoswitch Assembly | 1 |
| 34 | Control Handle Assembly | 1 |



| | | | |
|-----------------------------|-----------|--------------|-----|
| Material Aluminum | | | |
| TITLE | | | |
| End Effector Mounting Plate | | | |
| SIZE | CAGE CODE | DWG NO | REV |
| A4 | | Drawing # 25 | |
| SCALE | Scale | SHEET | |

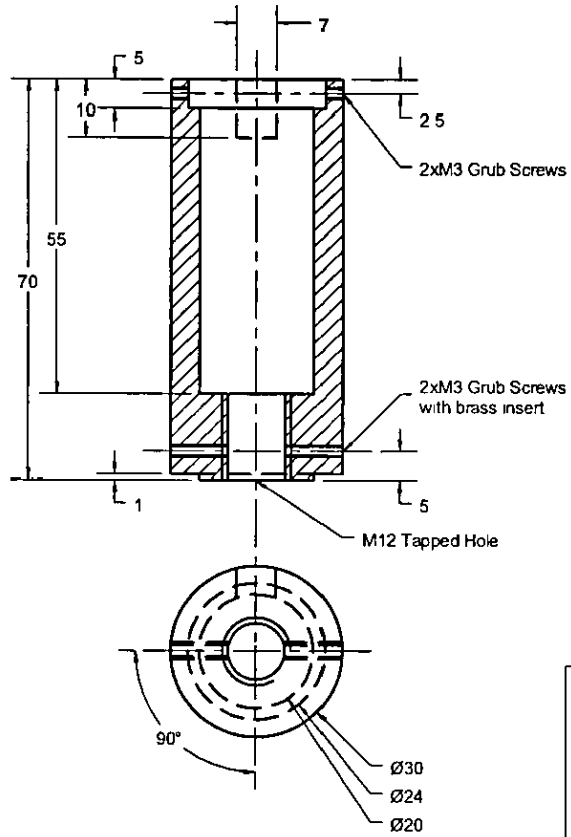






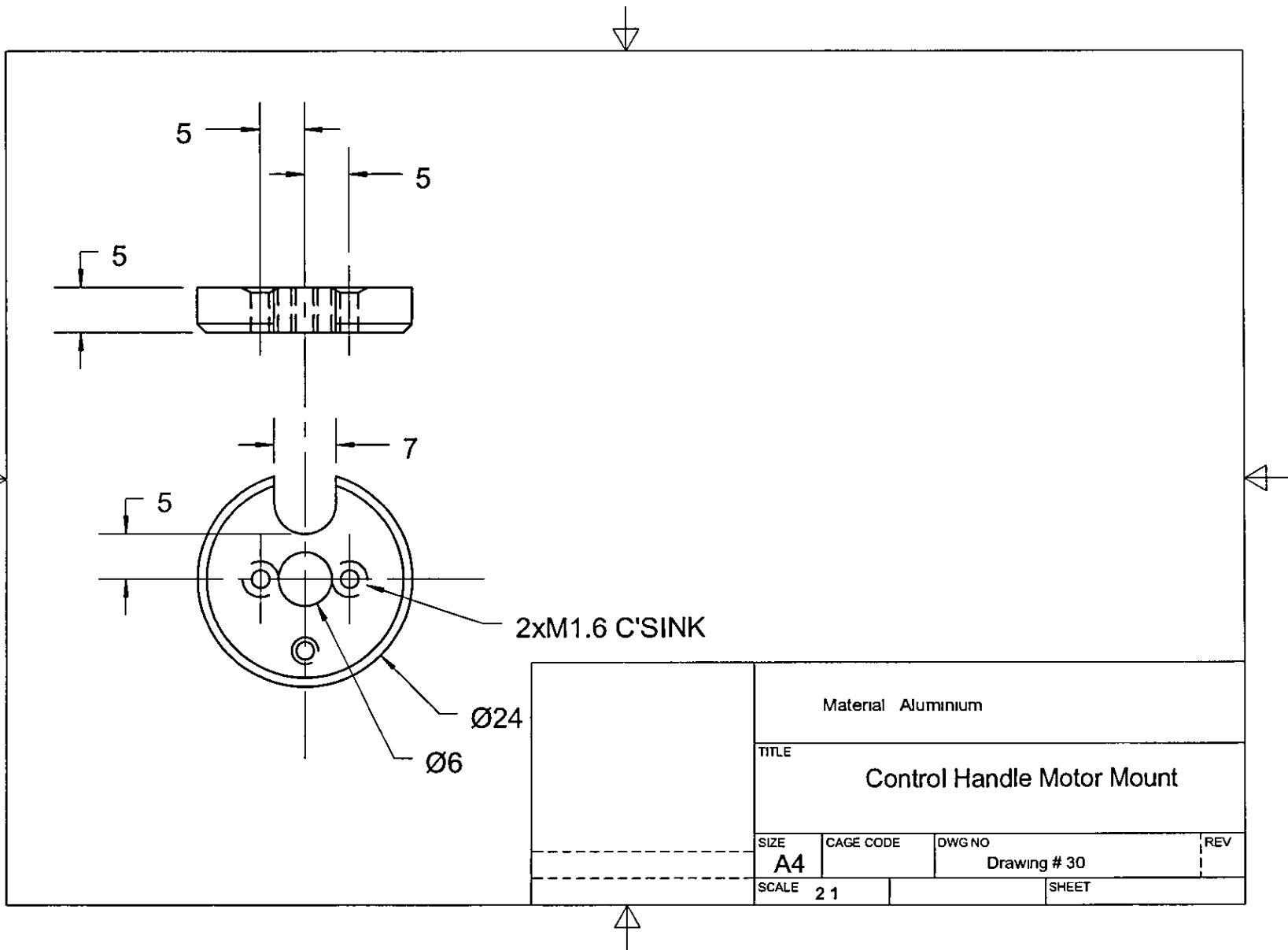
| | | | |
|---|-----------|------------------------|-----|
| Material Aluminium | | | |
| TITLE Control Handle Bearing Housing | | | |
| SIZE A4 | CAGE CODE | DWG NO Drawing # 28 | REV |
| SCALE 1 1 | SHEET | | |

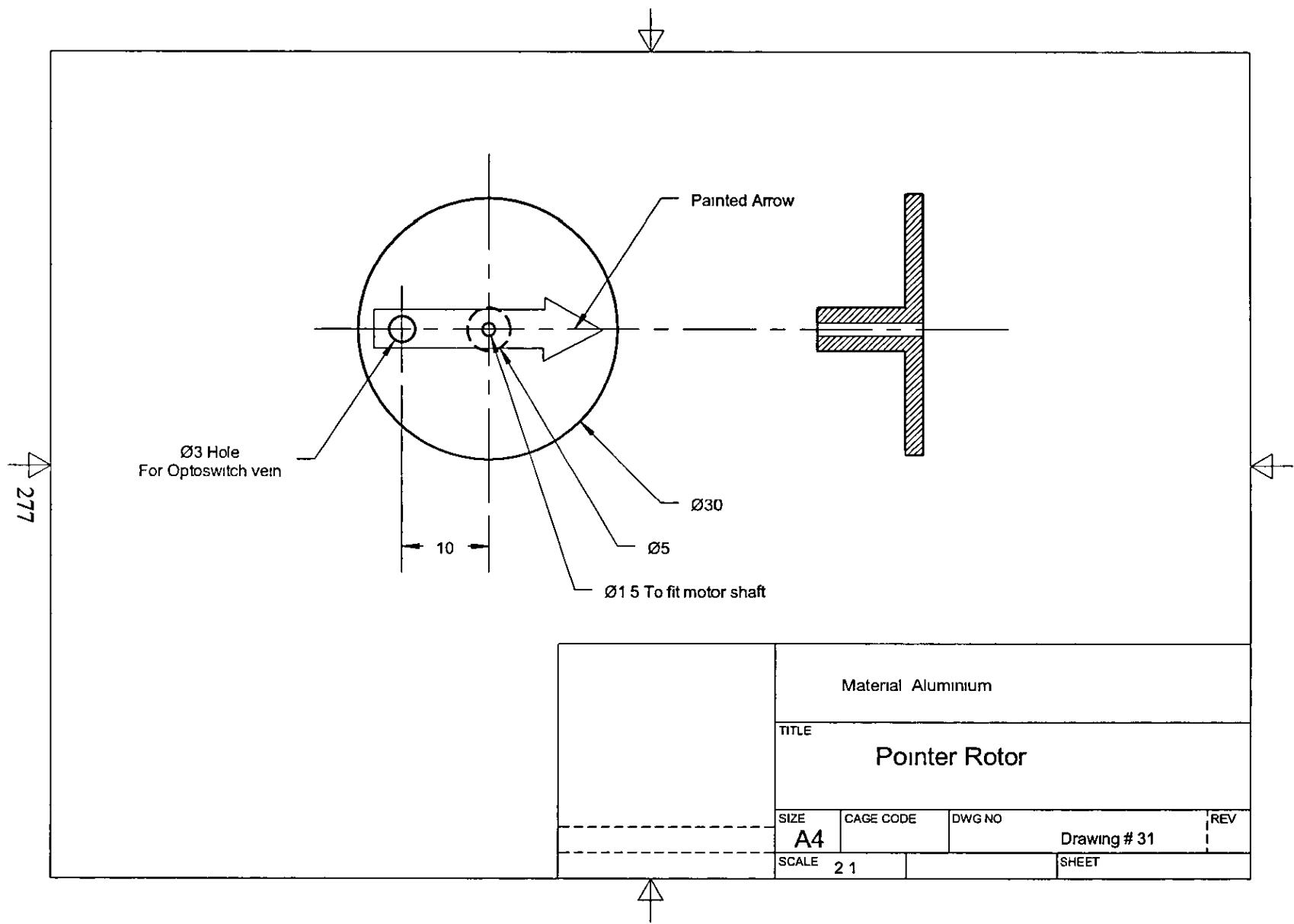
275



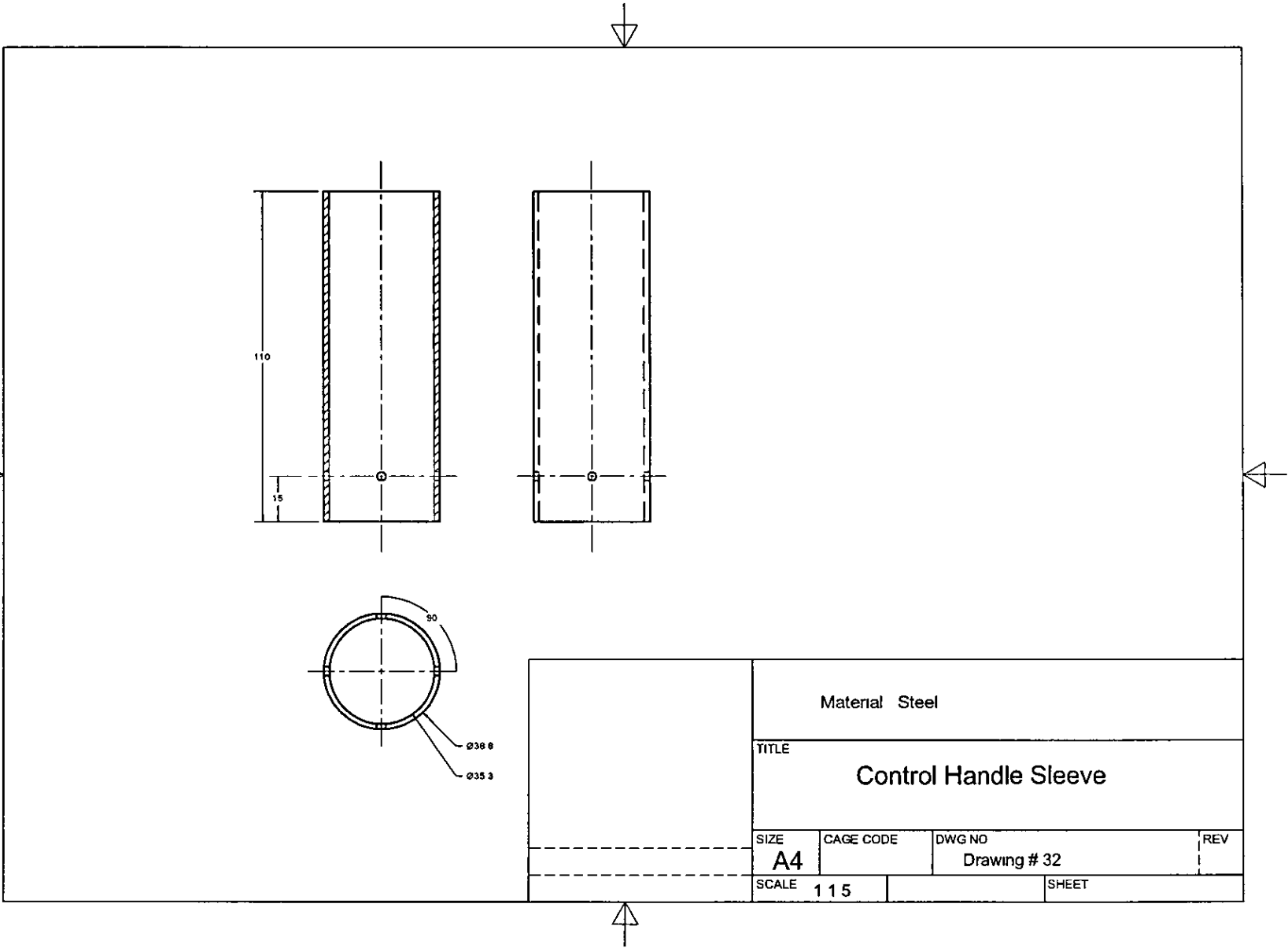
| | | | |
|---------------------------------------|-----------|------------------------|-----|
| Material Aluminium | | | |
| TITLE Control Handle Motor Housing | | | |
| SIZE A4 | CAGE CODE | DWG NO Drawing # 29 | REV |
| SCALE 1 1 | SHEET | | |

276

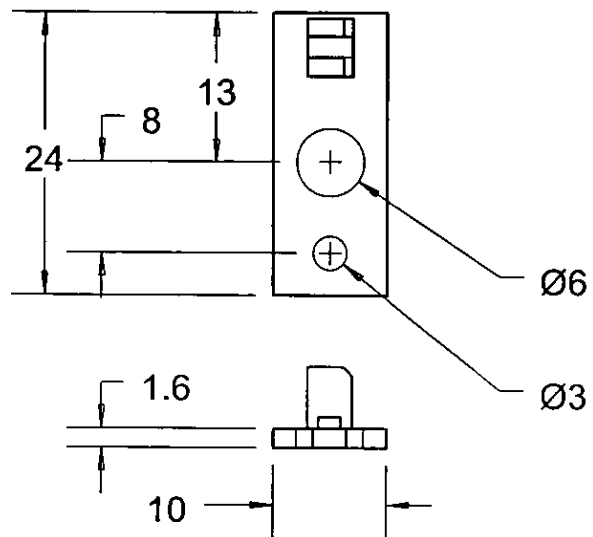




| | | | | |
|------------------------|-----------|--------|--------------|-----|
| Material Aluminium | | | | |
| TITLE Pointer Rotor | | | | |
| SIZE A4 | CAGE CODE | DWG NO | Drawing # 31 | REV |
| SCALE 2 1 | | | SHEET | |



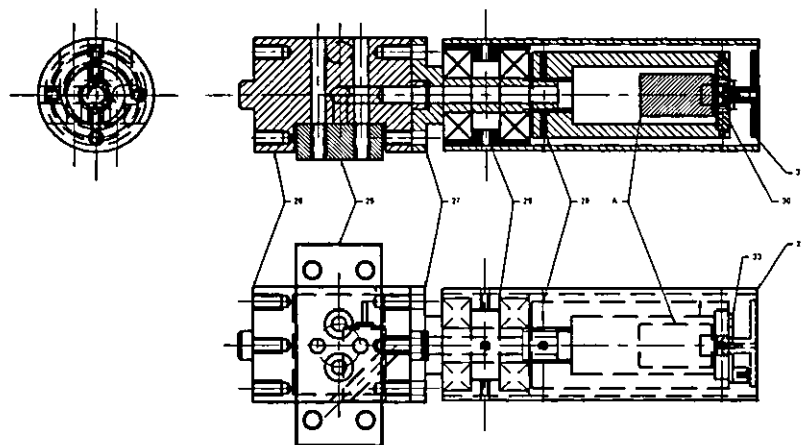
279



The slotted opto-switch is Farnell Part # 491354
The switch is mounted on PCB Fibre glass

| | | | |
|---------------------|-----------|--------------|-------|
| 1 off | | | |
| TITLE | | | |
| Optoswitch Assembly | | | |
| SIZE | CAGE CODE | DWG NO | REV |
| A4 | | Drawing # 33 | |
| SCALE | | | SHEET |

280



- 25 End effector mounting plate
- 26 End effector base
- 27 Control Handle Shaft
- 28 Control Handle Bearing Housing
- 29 Control Handle Motor Housing
- 30 Control Handle Motor Mount
- 31 Pointer Rotor
- 32 Control Handle Sleeve
- 33 Optoswitch Assembly
- A 1524T012S Minimotor plus IE2-512 Encoder

| | | | | | |
|-----------|--|-----------|-------|-------------------------|-----|
| | | | | Material Aluminium | |
| | | | | TITLE | |
| | | | | Control Handle Assembly | |
| SIZE | | CAGE CODE | | DWG NO | REV |
| A4 | | | | Drawing # 34 | |
| SCALE 1 2 | | | SHEET | | |

Appendix E Derivation of Mathematical Model

E.1. Case I

The system can be represented as a pair of horizontal sliding wedges with a block sliding in-between, corresponding to the two worm gears and worm wheel respectively. The free-body diagram of the simplified system is characterised by Figure 93, which shows that the upper wedge represents the α -worm, the worm wheel is shown as the centre wedge and the lower wedge represents the β -worm.

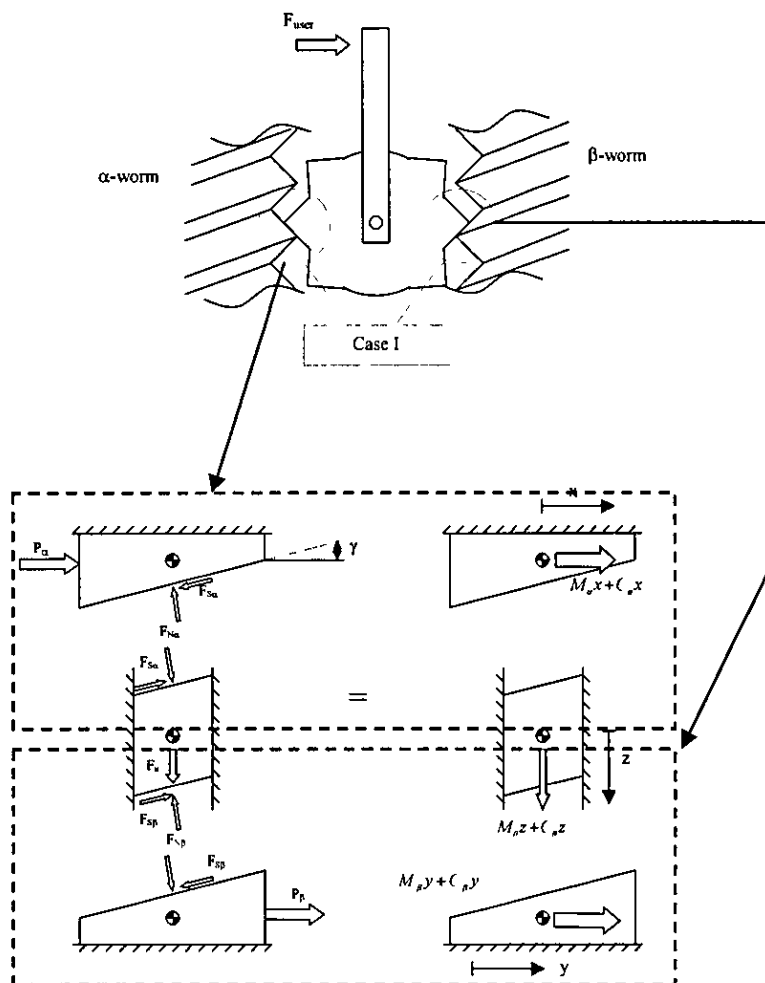


Figure 93 – Free-body diagram: Case I.

The equations of motion (EOM) for the wedges and block are given by

$$M_a \ddot{x} + C_a \dot{x} = P_a - F_{\backslash a} \sin \gamma - F_{\backslash a} \cos \gamma \quad (\text{E.1})$$

$$M_o \ddot{z} + C_o \dot{z} = F_s - F_{\backslash \theta} + F_{\backslash a} \cos \gamma + F_{\backslash \beta} \sin \gamma - F_{\backslash \beta} \cos \gamma - F_{\backslash a} \sin \gamma \quad (\text{E.2})$$

$$M_\beta \ddot{y} + C_\beta \dot{y} = P_\beta + F_{\backslash \beta} \sin \gamma - F_{\backslash \beta} \cos \gamma \quad (\text{E.3})$$

Solving (E.1) and (E.3) for $F_{\backslash a}$ and $F_{\backslash \beta}$ respectively yields

$$F_{\backslash a} = \frac{P_a - M_a \ddot{x} - C_a \dot{x} - F_{\backslash a} \cos \gamma}{\sin \gamma} \quad (\text{E.4})$$

and

$$F_{\backslash \beta} = \frac{-P_\beta + M_\beta \ddot{y} + C_\beta \dot{y} + F_{\backslash \beta} \cos \gamma}{\sin \gamma} \quad (\text{E.5})$$

The frictional forces within the system have been modelled using an exponential stick-slip friction model [96]. During motion, i.e. when $\dot{x} \neq 0$, $\dot{y} \neq 0$ or $\dot{z} \neq 0$, the frictional forces that oppose motion, $F_{\backslash a}$ and $F_{\backslash \beta}$, are described by

$$F_{\backslash a} = \left(\mu_{da} + (\mu_{sa} - \mu_{da}) e^{-\frac{|\dot{x}|}{\epsilon \sin \gamma}} \right) F_{\backslash a} \operatorname{sgn}(\dot{x}) \quad (\text{E.6})$$

$$\text{and} \quad F_{\backslash \beta} = \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{-\frac{|\dot{y}|}{\epsilon \sin \gamma}} \right) F_{\backslash \beta} \operatorname{sgn}(\dot{y}) \quad (\text{E.7})$$

It is also assumed that no backlash is present in this system. Therefore, from Figure 93, it is easy to show that x , y , and z are related by

$$x = y = \frac{z}{\tan \gamma} \quad (\text{E.8})$$

Substituting (E.4), (E.5), (E.6), and (E.7) into (E.2) and applying (E.8) yields the system dynamic equation

$$M_o \ddot{z} + C_o \dot{z} = F_s + \frac{P_a}{\Delta_{a1}} - \left(\frac{M_a \ddot{z} + C_a \dot{z}}{\Delta_{a1} \tan \gamma} \right) + \frac{P_\beta}{\Delta_{\beta 1}} - \left(\frac{M_\beta \ddot{z} + C_\beta \dot{z}}{\Delta_{\beta 1} \tan \gamma} \right) \quad (\text{E.9})$$

where

$$\Delta_{\alpha 1} = \frac{\sin \gamma + \left(\mu_{da} + (\mu_{\alpha} - \mu_{da}) e^{-\frac{|z|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(z) \cos \gamma}{\cos \gamma - \left(\mu_{da} + (\mu_{\alpha} - \mu_{da}) e^{-\frac{|z|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(z) \sin \gamma} \quad (\text{E.10})$$

and

$$\Delta_{\beta 1} = \frac{\sin \gamma - \left(\mu_{d\beta} + (\mu_{\beta} - \mu_{d\beta}) e^{-\frac{|z|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(z) \cos \gamma}{\cos \gamma + \left(\mu_{d\beta} + (\mu_{\beta} - \mu_{d\beta}) e^{-\frac{|z|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(z) \sin \gamma} \quad (\text{E.11})$$

Converting from the linear co-ordinate system, z , to the rotational system, θ , substitution of J_{θ}/r_{θ}^2 , J_{α}/r_{α}^2 , J_{β}/r_{β}^2 , $C_{j\theta}/r_{\theta}^2$, $C_{f\alpha}/r_{\alpha}^2$, $C_{f\beta}/r_{\beta}^2$, τ_u/r_{θ} , $\tau_{m\alpha}/r_{\alpha}$, $\tau_{m\beta}/r_{\beta}$, and θ r_{θ} in place of M_{θ} , M_{α} , M_{β} , C_{θ} , C_{α} , C_{β} , F_u , P_{α} , P_{β} , and z respectively, yields equation (E.12)

$$\begin{aligned} & J_{\theta} \ddot{\theta} + J_{\alpha} \frac{\ddot{\theta} r_{\theta}^2}{\Delta_{\alpha 1} r_{\alpha}^2 \tan \gamma} + J_{\beta} \frac{\ddot{\theta} r_{\theta}^2}{\Delta_{\beta 1} r_{\beta}^2 \tan \gamma} + \\ & C_{j\theta} \dot{\theta} + C_{f\alpha} \frac{\dot{\theta} r_{\theta}^2}{\Delta_{\alpha 1} r_{\alpha}^2 \tan \gamma} + C_{f\beta} \frac{\dot{\theta} r_{\theta}^2}{\Delta_{\beta 1} r_{\beta}^2 \tan \gamma} = \tau_u + \frac{\tau_{\alpha} r_{\theta}}{r_{\alpha} \Delta_{\alpha 1}} + \frac{\tau_{\beta} r_{\theta}}{r_{\beta} \Delta_{\beta 1}}. \end{aligned}$$

$$\begin{aligned} \Delta_{\alpha 1} &= \frac{\sin \gamma + \left(\mu_{da} + (\mu_{\alpha} - \mu_{da}) e^{-\frac{r_{\theta} |\theta|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(\theta) \cos \gamma}{\cos \gamma - \left(\mu_{da} + (\mu_{\alpha} - \mu_{da}) e^{-\frac{r_{\theta} |\theta|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(\theta) \sin \gamma} \\ \Delta_{\beta 1} &= \frac{\sin \gamma - \left(\mu_{d\beta} + (\mu_{\beta} - \mu_{d\beta}) e^{-\frac{r_{\theta} |\theta|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(\theta) \cos \gamma}{\cos \gamma + \left(\mu_{d\beta} + (\mu_{\beta} - \mu_{d\beta}) e^{-\frac{r_{\theta} |\theta|}{\epsilon \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(\theta) \sin \gamma} \end{aligned} \quad (\text{E.12})$$

The system of equation (E.12) describes the dynamic equations of motion. However, before motion can occur there are a set of conditions on the input torques that need to be satisfied. Figure 94 shows a free-body diagram of the two unwinding

conditions which enables the derivation of conditions given in (E.13) and (E.14). The condition that $P_a < 0$ and $P_\beta > 0$ in (E.13) or $P_\beta < 0$ and $P_a > 0$ in (E.14) represent the need to model the non-backdriveability of the system in this case

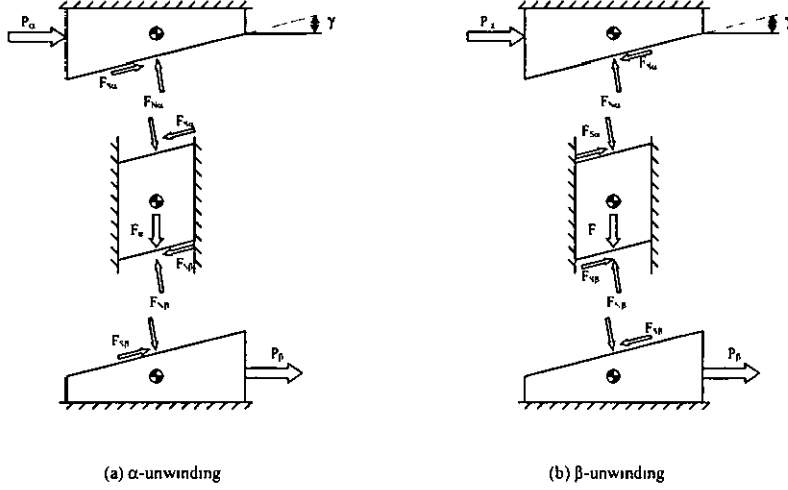


Figure 94 – Free-body diagram of static conditions: Case I

$$P_a < -F_u \frac{(\sin \gamma - \mu_{\alpha} \cos \gamma)}{(\cos \gamma + \mu_{\alpha} \sin \gamma)} - P_\beta \frac{(\sin \gamma - \mu_{\alpha} \cos \gamma)}{(\cos \gamma + \mu_{\alpha} \sin \gamma)} \frac{(\cos \gamma - \mu_{\beta} \sin \gamma)}{(\sin \gamma + \mu_{\beta} \cos \gamma)} < 0 \quad P_\beta < 0 \quad (\text{E.13})$$

$$P_\beta > -F_u \frac{(\sin \gamma - \mu_{\beta} \cos \gamma)}{(\cos \gamma + \mu_{\beta} \sin \gamma)} - P_a \frac{(\sin \gamma - \mu_{\beta} \cos \gamma)}{(\cos \gamma + \mu_{\beta} \sin \gamma)} \frac{(\cos \gamma - \mu_{\alpha} \sin \gamma)}{(\sin \gamma + \mu_{\alpha} \cos \gamma)} > 0 \quad P_a > 0 \quad (\text{E.14})$$

E.2. Case II

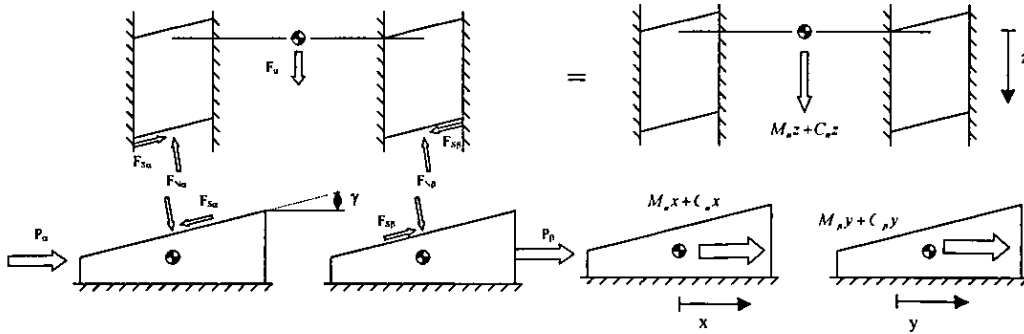


Figure 95 - Free-body diagram: Case II

The equivalent form of equations (E.1), (E.2), and (E.3) can be written for this case as

$$M_\alpha x + C_\alpha x = P_\alpha + F_{N\alpha} \sin \gamma - F_{V\alpha} \cos \gamma \quad (\text{E.15})$$

$$M_\beta z + C_\beta z = F_\alpha - F_{V\beta} + F_{V\beta} \sin \gamma - F_{V\alpha} \cos \gamma - F_{V\beta} \cos \gamma - F_{V\alpha} \sin \gamma \quad (\text{E.16})$$

$$M_\beta y + C_\beta y = P_\beta + F_{V\beta} \sin \gamma + F_{V\beta} \cos \gamma \quad (\text{E.17})$$

Substituting the relationships of equations (E.6) and (E.7) into (E.15) and (E.17) yields

$$F_{V\alpha} = \frac{-P_\alpha + M_\alpha x + C_\alpha x}{\sin \gamma - \left(\mu_{da} - (\mu_{sa} - \mu_{da}) e^{\frac{-|x|}{\epsilon \sin \gamma}} \right) F_{V\alpha} \operatorname{sgn}(z) \cos \gamma} \quad (\text{E.18})$$

$$F_{V\beta} = \frac{-P_\beta + M_\beta y + C_\beta y}{\sin \gamma + \left(\mu_{db} - (\mu_{sb} - \mu_{db}) e^{\frac{-|y|}{\epsilon \sin \gamma}} \right) \operatorname{sgn}(z) \cos \gamma} \quad (\text{E.19})$$

Substituting equations (E.18) and (E.19) into equation (E.16) and transforming the system for equivalent translation to rotation, as in equation (E.12), yields

$$\begin{aligned}
 & J_\theta \ddot{\theta} + J_a \frac{\partial r_\theta^2}{\Delta_{a2} r_a^2 \tan \gamma} + J_\beta \frac{\partial r_\theta^2}{\Delta_{\beta 1} r_\beta^2 \tan \gamma} + \\
 & C_{f\theta} \dot{\theta} + C_{fa} \frac{\partial r_\theta^2}{\Delta_{a2} r_a^2 \tan \gamma} + C_{f\beta} \frac{\partial r_\theta^2}{\Delta_{\beta 1} r_\beta^2 \tan \gamma} = \tau_u + \frac{\tau_a r_\theta}{r_a \Delta_{a2}} + \frac{\tau_\beta r_\theta}{r_\beta \Delta_{\beta 1}}
 \end{aligned} \tag{E.20}$$

where $\Delta_{\beta 1}$ is defined in equation (E 12) and

$$\Delta_{a2} = \frac{\sin \gamma - \left(\mu_{da} - (\mu_{da} - \mu_{sa}) e^{-\frac{|v|}{c \sin \gamma}} \right) \operatorname{sgn}(z) \cos \gamma}{\cos \gamma + \left(\mu_{da} - (\mu_{sa} - \mu_{da}) e^{-\frac{|z|}{c \sin \gamma \tan \gamma}} \right) \operatorname{sgn}(z) \sin \gamma} \tag{E.21}$$

E.3. Case III

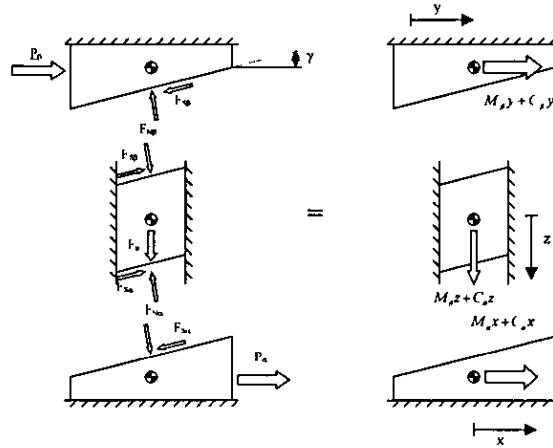


Figure 96 – Free-body diagram: Case III.

Similarly, for Case III the equations of motion for each of the wedges can be written

$$M_a \dot{x} + C_a x = P_a + F_{Na} \sin \gamma - F_{\gamma a} \cos \gamma \tag{E.22}$$

$$M_\theta z + C_\theta z = F_u - F_{\gamma \theta} + F_{\beta \theta} \sin \gamma - F_{\gamma a} \cos \gamma + F_{\beta \theta} \cos \gamma - F_{\gamma a} \sin \gamma \tag{E.23}$$

$$M_{\beta}y + C_{\beta}y = P_{\beta} - F_{\beta} \sin \gamma - F_{\beta} \cos \gamma \quad (\text{E.24})$$

Substituting (E 6) and (E 7) into (E 22) and (E 24) respectively leads to

$$F_{\beta} = \frac{-P_{\alpha} + M_{\alpha}x + C_{\alpha}x}{\sin \gamma - \left(\mu_{d\alpha} - (\mu_{va} - \mu_{da})e^{-\frac{|v|}{\epsilon \sin \gamma}} \right) F_{\alpha} \operatorname{sgn}(z) \cos \gamma} \quad (\text{E.25})$$

$$F_{\beta} = \frac{P_{\beta} - M_{\beta}y - C_{\beta}y}{\sin \gamma + \left(\mu_{d\beta} - (\mu_{v\beta} - \mu_{d\beta})e^{-\frac{|v|}{\epsilon \sin \gamma}} \right) F_{\alpha} \operatorname{sgn}(z) \cos \gamma} \quad (\text{E.26})$$

Hence, the overall equation of motion can be found to be

$$\begin{aligned} J_{\theta} \ddot{\theta} + J_{\alpha} \frac{\ddot{\theta} r_{\theta}^2}{\Delta_{\alpha 2} r_{\alpha}^2 \tan \gamma} + J_{\beta} \frac{\ddot{\theta} r_{\theta}^2}{\Delta_{\beta 2} r_{\beta}^2 \tan \gamma} + \\ C_{f\theta} \dot{\theta} + C_{f\alpha} \frac{\dot{\theta} r_{\theta}^2}{\Delta_{\alpha 2} r_{\alpha}^2 \tan \gamma} + C_{f\beta} \frac{\dot{\theta} r_{\theta}^2}{\Delta_{\beta 2} r_{\beta}^2 \tan \gamma} = \tau_v + \frac{\tau_{\alpha} r_{\theta}}{r_{\alpha} \Delta_{\alpha 1}} + \frac{\tau_{\beta} r_{\theta}}{r_{\beta} \Delta_{\beta 2}} \end{aligned} \quad (\text{E.27})$$

where $\Delta_{\alpha 2}$ is defined in equation (E.21) and

$$\Delta_{\beta 2} = \frac{\sin \gamma + \left(\mu_{d\beta} - (\mu_{v\beta} - \mu_{d\beta})e^{-\frac{|v|}{\epsilon \sin \gamma}} \right) \operatorname{sgn}(z) \cos \gamma}{\cos \gamma - \left(\mu_{d\beta} - (\mu_{v\beta} - \mu_{d\beta})e^{-\frac{|v|}{\epsilon \sin \gamma}} \right) \operatorname{sign}(z) \sin \gamma} \quad (\text{E.28})$$

E.4. Case IV

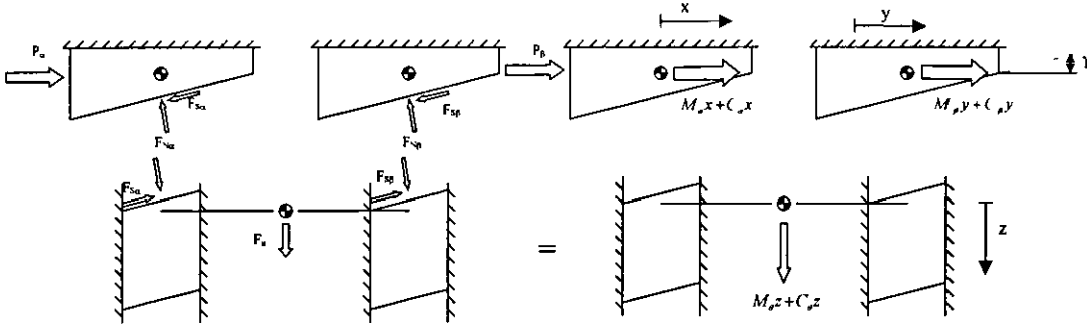


Figure 97 – Free-body diagram: Case IV

$$M_{\alpha}x + C_{\alpha}x = P_a - F_{\alpha a} \sin \gamma - F_{\alpha a} \cos \gamma \quad (\text{E.29})$$

$$M_{\theta}z + C_{\theta}z = F_s - F_{\beta \theta} + F_{\alpha a} \cos \gamma + F_{\beta \beta} \cos \gamma + F_{\beta \theta} \sin \gamma - F_{\alpha a} \sin \gamma \quad (\text{E.30})$$

$$M_{\beta}y + C_{\beta}y = P_b - F_{\beta \beta} \sin \gamma - F_{\beta \beta} \cos \gamma \quad (\text{E.31})$$

Following the same methods as in cases I, II, and III above leads to

$$F_{\alpha a} = \frac{P_a - M_{\alpha}x - C_{\alpha}x}{\sin \gamma + \left(\mu_{d\beta} - (\mu_{d\beta} - \mu_{\beta}) e^{\frac{|\nu|}{4 \sin \gamma}} \right) \text{sgn}(z) \cos \gamma} \quad (\text{E.32})$$

$$F_{\beta \beta} = \frac{P_b - M_{\beta}y - C_{\beta}y}{\sin \gamma + \left(\mu_{d\beta} - (\mu_{d\beta} - \mu_{\beta}) e^{\frac{|\nu|}{4 \sin \gamma}} \right) \text{sgn}(z) \cos \gamma} \quad (\text{E.33})$$

Yielding the system equation of motion

$$J_\theta \ddot{\theta} + J_\alpha \frac{\dot{\theta}^2}{\Delta_{\alpha 1} r_\alpha^2 \tan \gamma} + J_\beta \frac{\dot{\theta}^2}{\Delta_{\beta 2} r_\beta^2 \tan \gamma} + C_{f\theta} + C_{f\alpha} \frac{\dot{\theta}^2}{\Delta_{\alpha 1} r_\alpha^2 \tan \gamma} + C_{f\beta} \frac{\dot{\theta}^2}{\Delta_{\beta 2} r_\beta^2 \tan \gamma} = \tau_s + \frac{\tau_\alpha r_\theta}{r_\alpha \Delta_{\alpha 1}} + \frac{\tau_\beta r_\theta}{r_\beta \Delta_{\beta 2}}, \quad (\text{E.34})$$

where $\Delta_{\alpha 1}$ and $\Delta_{\beta 2}$ are defined as in equations (E.10) and (E.28) respectively.

Appendix F Common definitions in stability analysis

In order to detail the stability of the composite adaptive controller several standard theorems and definitions are required. These definitions are stated here for completeness.

F.1. Barbalat's lemma

If $f(t)$ is a uniformly continuous, real function and $\lim_{t \rightarrow \infty} f(t) = k < \infty$, then $\lim_{t \rightarrow \infty} f(t) = 0$ [Error! Bookmark not defined].

F.2. Rayleigh-Ritz theorem

A is a symmetric $n \times n$ positive-definite matrix. Let $\lambda_{\max}\{A\}$ and $\lambda_{\min}\{A\}$ be the maximum and minimum eigenvalues of A respectively. Then for any real x

$$\lambda_{\min}\{A\}\|x\|^2 \leq x^T A x \leq \lambda_{\max}\{A\}\|x\|^2$$

where $\|x\| = \sum_{i=1}^n |x_i|$

F.3. Uniform Continuity

Let $f() : [0, \infty) \rightarrow \mathfrak{R}$ be a uniformly continuous function. A function f is uniformly continuous if for any $\varepsilon > 0$, there is a $\delta(\varepsilon)$ such that

$$|t - t_0| < \delta(\varepsilon) \Rightarrow |f(t) - f(t_0)| < \varepsilon$$

Then f is said to belong to the Lebesgue space L_p if for $p \in [1, \infty)$,

$$\int_0^{\infty} |f(t)|^p dt < \infty$$

F.4. Properties of a stable transfer function

If $\frac{y(s)}{u(s)} = P(s)$ is a stable transfer function, then

- a) If $u(t)$ is bounded, so is $y(t)$,
- b) If $\lim_{t \rightarrow \infty} u(t) = 0$, then $\lim_{t \rightarrow \infty} y(t) = 0$,
- c) If $u(t) \in L_2$ then $\lim_{t \rightarrow \infty} y(t) = 0$.

Appendix G Motor/gearbox model

Although motor inertia is included in the simulation and experimental implementation of the controllers, the electrical dynamics and planetary gearbox have not been included. This Appendix presents the derivation of the adjusted model for the motor torque.

The model of the planetary gearbox may be shown as Figure 98

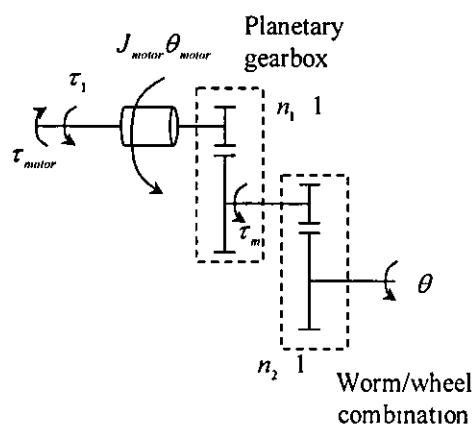


Figure 98 - Gearbox model

Therefore, the torque at the output of the gearbox, τ_m , may be written in terms of the motor torque as given in equation (G-1)

$$\tau_m = \eta_1 n_1 \tau_1 \tag{G-1}$$

where n_1 is the speed ratio of the planetary gearbox and η_1 represents the efficiency. Similarly the speed of the output shaft may be written as equation (G-2)

$$\theta_{motor} = n_1 n_2 \theta \quad (G-2)$$

where θ_{motor} is the speed of the motor shaft, θ is the speed of the joint, and n_2 is the speed ratio of the worm-wheel gearbox defined by equation .

$$n_2 = \frac{r_\theta}{r_a \tan \gamma} \quad (G-3)$$

Therefore it is possible to write

$$\tau_{motor} = J_{motor} \ddot{\theta}_{motor} + \tau_1 \quad (G-4)$$

Substitution of equations (G-2), and (G-4) into (G-1), yields

$$\tau_m = \eta_1 n_1 \tau_{motor} - J_{motor} n_1^2 n_2 \ddot{\theta} \quad (G-5)$$

The electrical model of an armature controlled DC motor may be written as equation (G-6) [108]

$$\tau_{motor} = k_t i_a = k_t \left(\frac{v_c - k_b s \theta_{motor}}{R_a + L_a s} \right) \quad (G-6)$$

In equation (G-6), k_t , k_b , R_a , L_a , and v_c are the torque constant, back EMF (electromotive force) constant, armature resistance, armature inductance, and input voltage. In terms of the output shaft velocity, equation (G-6) may also be written as

$$\tau_{motor} = k_t \left(\frac{v_c - k_b n_1 n_2 s \theta}{R_a + L_a s} \right) \quad (G-7)$$

Therefore, assuming that the armature inductance is negligible, equation (G-5) can be written as

$$\tau_m = \frac{\eta_1 n_1 k_t}{R_a} v_c - \frac{k_t k_b \eta_1 n_1^2 n_2}{R_a} \dot{\theta} - J_{motor} n_1^2 n_2 \ddot{\theta} \quad (G-8)$$

Appendix H **Publications**

This appendix includes the two papers that had been published prior to the submission of this thesis

H.1. Control of an Actively Constrained Robotic Joint for Passive Deployment Applications

Reedman. A. V. C., and Bouazza-Marouf, K., Control of an Actively Constrained Robotic Joint for Passive Deployment Applications, *Proceedings of the Institution of Mechanical Engineers, Part K Journal of Multibody Dynamics*, **215**, 2001, pp187-197.

Control of an actively constrained robotic joint for passive deployment applications

A V C Reedman and K Bouazza-Marouf*

Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Leicestershire, UK

Abstract: The design and control of an actively constrained revolute joint with backlash cancellation for passively deployed devices is presented in this paper. The drive mechanism consists of two motor-driven worms coupled to a single worm wheel. A mathematical model of the system is used in order to develop a backlash cancellation strategy and computed-torque motion control algorithm. Experimental results show that the position of the joint can be successfully controlled to track a trajectory generated from a user-input force command signal while cancelling backlash at the gear interface.

Keywords: backlash cancellation, computed-torque control, robotics

NOTATION

| | | | |
|-----------------------------------|---|---------------------------------|---|
| $C_{fx}, C_{f\beta}, C_{f\theta}$ | viscous friction coefficient of the α -worm system, the β -worm system and the worm-wheel system respectively | $r_\alpha, r_\beta, r_\theta$ | radius of contact of the α -worm with the worm wheel, of the β -worm with the worm wheel and of the α - and β -worms respectively |
| $C_\alpha, C_\beta, C_\theta$ | equivalent viscous friction coefficient of the α -worm system, the β -worm system and the worm-wheel system respectively | x, y, z | equivalent linear displacement of the α -worm, β -worm and worm wheel respectively |
| $F_{N\alpha}, F_{N\beta}$ | reaction force at the interface between the α -worm and the worm wheel and between the β -worm and the worm wheel respectively | γ | lead angle of the α - and β -worms |
| $F_{S\alpha}, F_{S\beta}$ | Coulomb friction force between the α -worm and the worm wheel and between the β -worm and the worm wheel respectively | ε | friction constant |
| F_u | applied user-input force | θ | link position |
| F_α, F_β | equivalent force generated by the torque from the motors | θ_d | desired link position |
| $J_\alpha, J_\beta, J_\theta$ | inertia of the α -worm and the motor, of the β -worm and the motor and of the worm wheel, shaft and robot arm respectively | $\mu_{d\alpha}, \mu_{d\beta}$ | coefficient of dynamic friction of the α - and β -worms |
| $M_\alpha, M_\beta, M_\theta$ | equivalent mass of J_α, J_β and J_θ respectively | $\mu_{s\alpha}, \mu_{s\beta}$ | coefficient of static friction of the α - and β -worms |
| | | $\tau_{m\alpha}, \tau_{m\beta}$ | torque generated by the driving motors |
| | | τ_u | equivalent torque generated by the applied user-input force |

1 INTRODUCTION

Development of robotic devices that interact closely with humans has been of particular interest to the surgical industry for the last 20 years. In an attempt to improve the accuracy and repeatability of surgical procedures, robotic tool-positioning and tool-guiding devices have been introduced into the operating theatre.

Most of the robotic devices that are currently used to aid surgeons in performing operational tasks have been based on modified industrial manipulators. The best known of these modified robots is ROBODOC [1, 2]. ROBODOC was designed for machining a cavity in bone

The MS was received on 13 June 2001 and was accepted after revision for publication on 24 September 2001

*Corresponding author: Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK

to receive a prosthetic component during cementless total hip replacement. The surgeon oversees the operation but has no control over the motion other than the ability to stop and start the robot using a control pendant. Although these large industrial systems are highly accurate, they have typically been designed to meet the requirements of high-speed and high-torque applications. The introduction of a large, powerful robot into an environment such as the operating theatre casts doubt on the safety of the patient, surgeon and other operating room staff [3]. It is for this reason that the makers of ROBODOC, and other similar devices such as the master-slave robot Da Vinci [4], have been required to implement many layers of software and hardware safety systems in order to ensure that unauthorized motion cannot be made. It seems that more attention has been focused on limiting rather than eliminating the ability to do harm.

Custom-built robotic devices, such as ACROBOT (active constraint ROBOT) [5] and PADyC (passive arm with dynamic constraints) [6], have been a more attractive alternative to the larger and more expensive industrial systems. ACROBOT is a four-degree-of-freedom (4-DOF) manipulator designed to help the surgeon perform total knee replacement surgery. The surgeon controls the motions of the manipulator by applying a force to the control handle in order to drive the joints of the robot. All of the joints are backdriveable, and a d.c. motor at each joint controls the resistive force that the surgeon feels using a force control strategy. The workspace of the robot is actively constrained to confine the end-effector to remain within a preplanned safe working region. However, while the ACROBOT system requires physical input from the surgeon in order to move the manipulator, the force-controlled servomotors in each joint are still powerful enough to provide motion against the user. This again raises many issues regarding surgeon and patient safety. Another proposed robot assistant device is PADyC, a robot incapable of motion of its own. At every joint there exists a pair of overrunning clutches, each running on a separate motor driven hub. By controlling the speed of each motor, one clutch limits the maximum allowable speed in the clockwise direction while the second clutch restricts speed in the anticlockwise direction. It was demonstrated that PADyC could be used to confine the motion of the surgeon within a predefined work area and, to a limited extent, follow a predefined path. With 20 N of force applied on the control handle, the 2-DOF prototype of PADyC exhibited up to 20 mm of error at the tool tip with link lengths of 0.25 m [7]. The error was attributed to joint flexibility and backlash in the clutch mechanism. Errors of this magnitude cannot be tolerated in most surgical applications.

In this paper, a robot joint architecture designed to cancel backlash is presented. A conceptual overview of the mechanism is given for a system that has been designed to be inherently fail safe, non-backdriveable, and suitable for use in applications where direct control

of the end-effector (surgical tool) by a human is required. A mathematical model of the mechanism is then presented and used to develop a computed-torque control algorithm for regulating position. A brief description of the experimental set-up and control system architecture is also included. Simulation and experimental results of the mechanism and control algorithm are given, followed by concluding remarks.

2 CONCEPTUAL OVERVIEW

In gear system design, a small amount of backlash is required in order to allow for thermal expansion, lubrication and lower frictional losses within a drivetrain [8]. In many robotic applications, even minute amounts of backlash can cause errors in position that are not acceptable in demanding applications such as surgery. For example, if a single link of a robot arm exhibits ± 0.0087 rad (or $\pm 0.5^\circ$) of backlash, there will be an error of ± 3.5 mm at the end of a 0.4 m link. In order to reduce the backlash to an acceptable level, gear train housings must be manufactured to high engineering tolerances. This is expensive in terms of both the machinery required and the time taken for manufacture. Another solution would be to use antibacklash gears, readily available from many gear manufacturers. However, these antibacklash gears are spring-loaded systems and still allow relative motion between input and output shafts under certain loading conditions. The proposed system, presented in this paper, is a robotic joint designed to cancel backlash.

The joint, shown schematically in Fig. 1, consists of

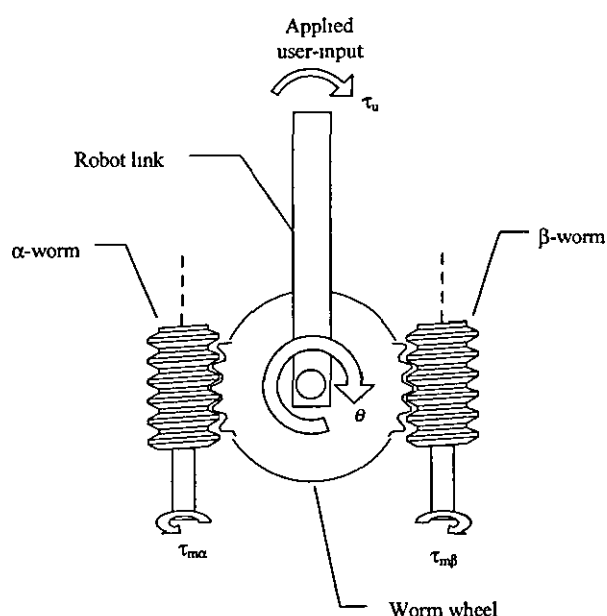


Fig. 1 Dual-worm drive joint

two worms, each driven by a low-power d.c. servomotor. The two worms follow a worm wheel that is fixed to the robot link. The user provides a force on the end-effector and motion is allowed by controlling the two worms simultaneously. The non-backdriveability of the worm mechanism means that, no matter how much force the operator applies, motion is not allowed until both motors are controlled to move in the same direction.

The parallel nature of this overactuated joint structure makes this active system safer than modified industrial robots. If there were a failure, both motors would have to drive in the same direction for any motion to occur. However, if this failure occurs, acceleration of the joints will be very small owing to the inertia of the links and low power of the motors, giving the operator plenty of time to react, i.e. to release the deadman handle. The dual-worm mechanism also has the ability to eliminate backlash using a control strategy that is discussed below. The control of the joint must be robust against the non-linear effects of inertia and frictional forces at the worm/wheel interface.

3 MATHEMATICAL MODEL

Analysis of the dual-worm mechanism identifies four

separate cases of contact between the two worms and the worm wheel. The first of these four cases (case I), shown in Fig. 2a, represents the worm wheel in contact with the leading edges of the thread of both the α - and β -worms. In this condition the two worms cancel backlash. Figure 2b shows the second case (case II) where the worm wheel is in contact with the trailing edge of the thread of the α -worm and the leading edge of the thread of the β -worm. This situation is highly undesirable in terms of positional accuracy, as backlash is not cancelled. The third case (case III), illustrated in Fig. 2c, shows the trailing edges of the thread of the two worms in contact with the worm wheel. Backlash is cancelled in this condition. Finally, in Fig. 2d (case IV), the leading edge of the thread of the α -worm and the trailing edge of the thread of the β -worm are shown in contact with the wheel. Similar to case II, backlash is not cancelled in this condition. Therefore, backlash is eliminated if either case I or case III is continuously maintained. In order to simplify the model, a control strategy is chosen that will maintain the conditions required for case I only.

Unwinding the thread of the three gears allows the mechanism to be modelled as a system of three wedges sliding against each other. A free-body diagram for case I is shown in Fig. 3, where the upper wedge represents the α -worm, the middle wedge represents the worm wheel and the bottom wedge represents the β -worm. Using Fig.

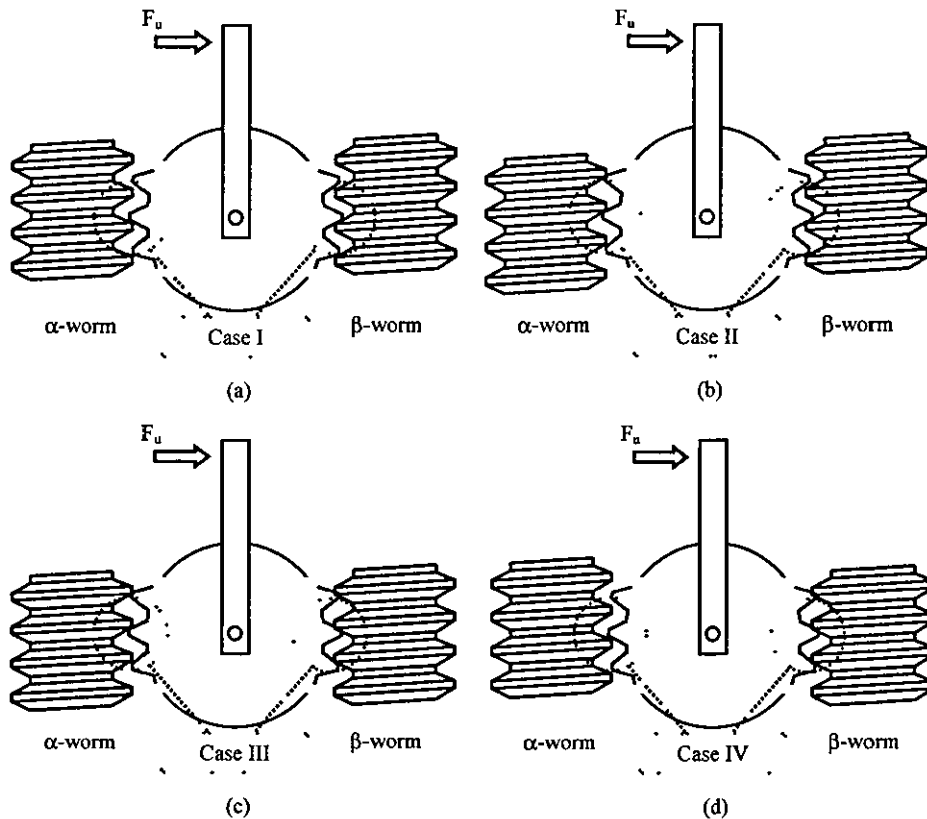


Fig. 2 Dual-worm joint modes of contact

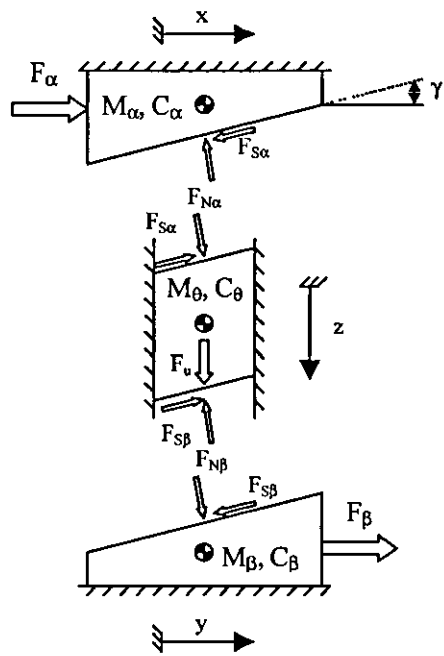


Fig. 3 Free-body diagram of the worm drive joint case I

3, it is possible to write the dynamic equations of each wedge as equations (1) to (3) respectively. The friction forces, $F_{S\alpha}$ and $F_{S\beta}$, between the two worms and the wheel are modelled using an exponential stick-slip friction model [9, 10] given in equations (4) and (5) respectively. In order to simplify the model, it is assumed that neither the α -worm nor the β -worm loses contact with the worm wheel, i.e. $F_{N\alpha} \neq 0$ and $F_{N\beta} \neq 0$.

$$M_\alpha \ddot{x} + C_\alpha \dot{x} = F_\alpha - F_{N\alpha} \sin \gamma - F_{S\alpha} \cos \gamma \tag{1}$$

$$M_\theta \ddot{z} + C_\theta \dot{z} = F_u + F_{N\alpha} \cos \gamma - F_{S\alpha} \sin \gamma - F_{N\beta} \cos \gamma + F_{S\beta} \sin \gamma \tag{2}$$

$$M_\beta \ddot{y} + C_\beta \dot{y} = F_\beta + F_{N\beta} \sin \gamma - F_{S\beta} \cos \gamma \tag{3}$$

$$F_{S\alpha} = \left(\mu_{d\alpha} + (\mu_{s\alpha} - \mu_{d\alpha}) e^{|x|/(\varepsilon \cos \gamma)} \right) \text{sgn}(x) F_{N\alpha} \tag{4}$$

$$F_{S\beta} = \left(\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{|y|/(\varepsilon \cos \gamma)} \right) \text{sgn}(y) F_{N\beta} \tag{5}$$

Substitution of (1), (3), (4) and (5) into (2) yields

$$M_\theta \ddot{z} + C_\theta \dot{z} = F_u + \frac{F_\alpha - M_\alpha \ddot{x} - C_\alpha \dot{x}}{\Delta_\alpha} + \frac{F_\beta - M_\beta \ddot{y} - C_\beta \dot{y}}{\Delta_\beta} \tag{6}$$

where

$$\Delta_\alpha = \frac{\sin \gamma + (\mu_{d\alpha} + (\mu_{s\alpha} - \mu_{d\alpha}) e^{|x|/(\varepsilon \cos \gamma)}) \text{sgn}(x) \cos \gamma}{\cos \gamma - (\mu_{d\alpha} + (\mu_{s\alpha} - \mu_{d\alpha}) e^{|x|/(\varepsilon \cos \gamma)}) \text{sgn}(x) \sin \gamma}$$
$$\Delta_\beta = \frac{\sin \gamma - (\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{|y|/(\varepsilon \cos \gamma)}) \text{sgn}(y) \cos \gamma}{\cos \gamma + (\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{|y|/(\varepsilon \cos \gamma)}) \text{sgn}(y) \sin \gamma}$$

Converting from the unwound threads to the rotational system by replacing $F_\alpha, F_\beta, F_u, M_\alpha, M_\beta, M_\theta, C_\alpha, C_\beta, C_\theta, z, x$ and y in equation (6) with $\tau_{m\alpha}/r_\alpha, \tau_{m\beta}/r_\beta, \tau_u/r_\theta, J_\alpha/r_\alpha^2, J_\beta/r_\beta^2, J_\theta/r_\theta^2, C_{f\alpha}/r_\alpha^2, C_{f\beta}/r_\beta^2, C_{f\theta}/r_\theta^2, \theta r_\theta, \theta r_\theta/\tan \gamma$ and $\theta r_\theta/\tan \gamma$ respectively yields

$$J_T \ddot{\theta} + C_T \dot{\theta} = \tau_u + \frac{\tau_{m\alpha} r_\theta}{\Delta_\alpha r_\alpha} + \frac{\tau_{m\beta} r_\theta}{\Delta_\beta r_\beta} \tag{7}$$

where

$$J_T = J_\theta + \frac{J_\alpha r_\theta^2}{\Delta_\alpha r_\alpha^2 \tan \gamma} + \frac{J_\beta r_\theta^2}{\Delta_\beta r_\beta^2 \tan \gamma}$$
$$C_T = C_{f\theta} + \frac{C_{f\alpha} r_\theta^2}{\Delta_\alpha r_\alpha^2 \tan \gamma} + \frac{C_{f\beta} r_\theta^2}{\Delta_\beta r_\beta^2 \tan \gamma}$$

For simulation purposes, it is necessary to ensure that there is no motion until the motor torque of either the α -worm or the β -worm exceeds the static friction forces. Figure 4a shows how the friction forces act on the three wedges during motion of the wheel in the anticlockwise direction, and Fig. 4b shows the orientation of the forces during motion of the wheel in the clockwise direction.

From Fig. 4 it is possible to express the equilibrium conditions in the anticlockwise direction [equation (8)] and clockwise direction [equation (9)]. In Fig. 4, $\mu_{S\alpha} F_{N\alpha}$ and $\mu_{S\beta} F_{N\beta}$ are substituted for $F_{S\alpha}$ and $F_{S\beta}$ respectively:

$$F_u + F_\alpha \left(\frac{\cos \gamma + \mu_{S\alpha} \sin \gamma}{\sin \gamma - \mu_{S\alpha} \cos \gamma} \right) + F_\beta \left(\frac{\cos \gamma - \mu_{S\beta} \sin \gamma}{\sin \gamma + \mu_{S\beta} \cos \gamma} \right) = 0 \tag{8}$$

$$F_u + F_\alpha \left(\frac{\cos \gamma - \mu_{S\alpha} \sin \gamma}{\sin \gamma + \mu_{S\alpha} \cos \gamma} \right) + F_\beta \left(\frac{\cos \gamma + \mu_{S\beta} \sin \gamma}{\sin \gamma - \mu_{S\beta} \cos \gamma} \right) = 0 \tag{9}$$

Using equation (8) for the system of unwound threads, motion of the wheel in the anticlockwise direction will occur if the condition given by equation (10) is satisfied. Equation (11) gives this condition for the actual

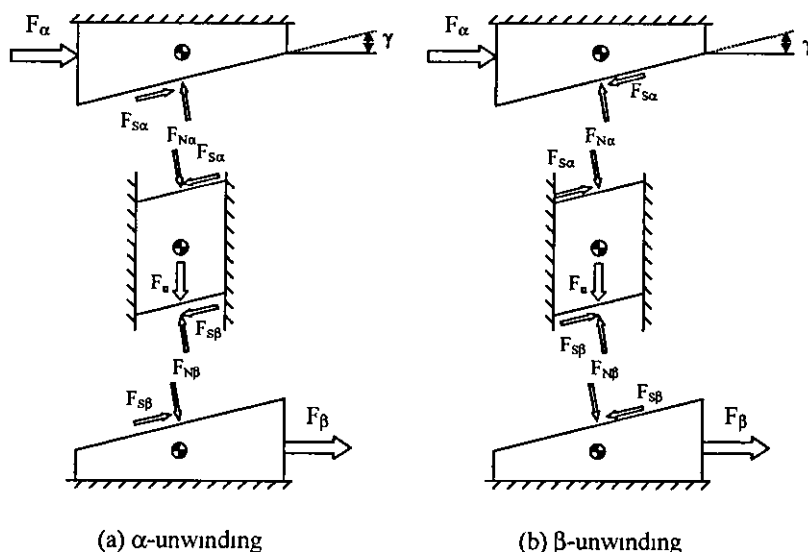


Fig. 4 Free-body diagram of static conditions

rotational system

$$F_\alpha < -F_u \left(\frac{\sin \gamma - \mu_{S\alpha} \cos \gamma}{\cos \gamma + \mu_{S\alpha} \sin \gamma} \right) - F_\beta \left(\frac{\sin \gamma - \mu_{S\alpha} \cos \gamma}{\cos \gamma + \mu_{S\alpha} \sin \gamma} \right) \times \left(\frac{\cos \gamma - \mu_{S\beta} \sin \gamma}{\sin \gamma + \mu_{S\beta} \cos \gamma} \right) < 0 \quad \text{and} \quad F_\beta < 0 \quad (10)$$

$$\frac{\tau_{m\alpha}}{r_\alpha} < -\frac{\tau_u}{r_\theta} \left(\frac{\sin \gamma - \mu_{S\alpha} \cos \gamma}{\cos \gamma + \mu_{S\alpha} \sin \gamma} \right) - \frac{\tau_{m\beta}}{r_\beta} \left(\frac{\sin \gamma - \mu_{S\alpha} \cos \gamma}{\cos \gamma + \mu_{S\alpha} \sin \gamma} \right) \times \left(\frac{\cos \gamma - \mu_{S\beta} \sin \gamma}{\sin \gamma + \mu_{S\beta} \cos \gamma} \right) < 0 \quad \text{and} \quad \tau_{m\beta} < 0 \quad (11)$$

Similarly, using equation (9) for the system of unwound threads, clockwise rotation of the wheel will be allowed if the condition given by equation (12) is satisfied. Equation (13) gives the same condition in terms of the actual rotational system

$$F_\beta > -F_u \left(\frac{\sin \gamma - \mu_{S\beta} \cos \gamma}{\cos \gamma + \mu_{S\beta} \sin \gamma} \right) - F_\alpha \left(\frac{\sin \gamma - \mu_{S\beta} \cos \gamma}{\cos \gamma + \mu_{S\beta} \sin \gamma} \right) \times \left(\frac{\cos \gamma - \mu_{S\alpha} \sin \gamma}{\sin \gamma + \mu_{S\alpha} \cos \gamma} \right) > 0 \quad \text{and} \quad F_\alpha > 0 \quad (12)$$

$$\frac{\tau_{m\beta}}{r_\beta} > -\frac{\tau_u}{r_\theta} \left(\frac{\sin \gamma - \mu_{S\beta} \cos \gamma}{\cos \gamma + \mu_{S\beta} \sin \gamma} \right) - \frac{\tau_{m\alpha}}{r_\alpha} \left(\frac{\sin \gamma - \mu_{S\beta} \cos \gamma}{\cos \gamma + \mu_{S\beta} \sin \gamma} \right) \times \left(\frac{\cos \gamma - \mu_{S\alpha} \sin \gamma}{\sin \gamma + \mu_{S\alpha} \cos \gamma} \right) > 0 \quad \text{and} \quad \tau_{m\alpha} > 0 \quad (13)$$

4 CONTROL DEVELOPMENT

The aim of the control algorithm is to make the manipulator track a desired position command while cancelling backlash at the worm interface. In order to ensure surgeon and patient safety, the joint and control method *must not* exhibit

- (a) any motion against the user,
- (b) any motion without direct control from the user or
- (c) any backlash at the worm/wheel interface

To this end, effective control of the dual-worm mechanism requires two algorithms. In the clockwise direction (i.e. $\tau_u > 0$) the motor command voltage for the α -worm is set to a constant value and the β -motor torque, $\tau_{m\beta}$, is controlled to unwind the β -worm to track the trajectory. In this condition the β -worm *leads* and the α -worm is used to *follow* the motion of the worm wheel without applying unnecessary frictional forces to the system. Control of motion in this manner shall be termed β -unwinding control. However, in the anticlockwise direction (i.e. $\tau_u < 0$) the β -worm motor command

voltage is set to a constant value and $\tau_{m\alpha}$ is used to control the unwinding of the α -worm to track the trajectory. This is termed α -unwinding control. In this instance, the α -worm *leads* and the β -worm *follows*. This dual-control strategy is shown to work well.

For the purpose of this preliminary work, the desired velocity is proportional to the user-input torque. That is

$$\dot{\theta}_d = K_u \tau_u \quad (14)$$

where K_u could be chosen as a function of position in order to restrict motion. For a manipulator with more than one DOF, this velocity-limiting algorithm would be used to constrain the user to remain along/within a predefined path/region within the robot workspace.

4.1 α -Unwinding control

The α -unwinding control law regulates the unwinding of the α -worm to follow a time-varying trajectory. The error between the desired position and the actual position of the system is defined as

$$e = \theta_d - \theta \quad (15)$$

From equations (7) and (15) it is possible to derive the tracking error dynamics given by equation (16):

$$e = \theta_d + \frac{1}{J_T} \left(C_T \theta - \tau_u - \frac{\tau_{m\alpha} r_\theta}{\Delta_\alpha r_\alpha} - \frac{\tau_{m\beta} r_\theta}{\Delta_\beta r_\beta} \right) \quad (16)$$

Using feedback linearization, $\tau_{m\alpha}$ can be chosen to cancel the dynamic effects of friction and the inertial forces; $\tau_{m\alpha}$ is obtained as given in equation (17), where $\hat{\Delta}_\alpha$, $\hat{\Delta}_\beta$, \hat{C}_T and \hat{J}_T are the estimates of Δ_α , Δ_β , C_T and J_T respectively

$$\tau_{m\alpha} = \frac{r_\alpha \hat{\Delta}_\alpha}{r_\theta} \left(\hat{J}_T u_\alpha + \hat{J}_T \theta_d + \hat{C}_T \theta - \tau_u - \frac{\tau_{m\beta} r_\theta}{r_\beta \hat{\Delta}_\beta} \right) \quad (17)$$

In equation (17) the control u_α is chosen to drive the error to zero and $\tau_{m\beta}$ is the measured β -motor torque. Convergence of the error to zero is achieved using a proportional + integral + derivative (PID) control law

$$u_\alpha = K_{v\alpha} e + K_{p\alpha} e + K_{a\alpha} \int e \, dt \quad (18)$$

It is important to note that, if the components of equation (17) are not known exactly, the system parameter estimates are erroneous, and without the integral term in equation (18) there may be a non-zero steady state error.

4.2 β -Unwinding control

A computed-torque controller for unwinding the β -worm can be found in a similar fashion to that described in Section 4.1. Equation (7) can be used to derive the β -motor torque as given in equation (19)

$$\tau_{m\beta} = \frac{r_\beta \hat{\Delta}_\beta}{r_\theta} \left(\hat{J}_T u_\beta + \hat{J}_T \theta_d + \hat{C}_T \theta - \tau_u - \frac{\tau_{m\alpha} r_\theta}{r_\alpha \hat{\Delta}_\alpha} \right) \quad (19)$$

However, in this case the control u_β is chosen to drive the error to zero and $\tau_{m\alpha}$ is the measured α -motor torque. Convergence of the error is achieved using the PID controller

$$u_\beta = K_{v\beta} e + K_{p\beta} e + K_{a\beta} \int e \, dt \quad (20)$$

5 EXPERIMENTAL SET-UP

Figure 5 shows the dual-worm driven joint mechanism. The joint is controlled using a Pentium 233 MHz personal computer running the QNX 4.25 real-time operating system. A schematic diagram of the control system is shown in Fig. 6.

The control algorithm requires measurements of motor torque, user-input force, and position. The user-input force is measured using four strain gauges, in a Wheatstone bridge configuration, mounted on a specially designed section of the link. The 3.27 W geared d.c. motors are capable of producing 0.4 Nm of torque at the output of the gearbox. Closed-loop regulation of the motor output torque is accomplished by measuring the armature current and implementing a digital proportional + integral (PI) controller. All analogue signals are measured and generated using 12 bit analogue-to-digital converters and 12 bit digital-to-analogue converters. The link rotation is recorded using an encoder and appropriate electronics to generate 20000 counts per revolution, giving a resolution of 0.0003142 rad/count (or 0.018°/count). The encoder position is read as a 24 bit number from an HCTL-1100 motion control interface. The frequency of the control loop is set at 600 Hz, and link rotational velocity measurement is obtained in software by using a backward difference algorithm.

6 RESULTS AND DISCUSSION

For the model, given by equation (7), and the controllers of equations (17) and (19), the estimated values of the system parameters are given in Table 1. The values of

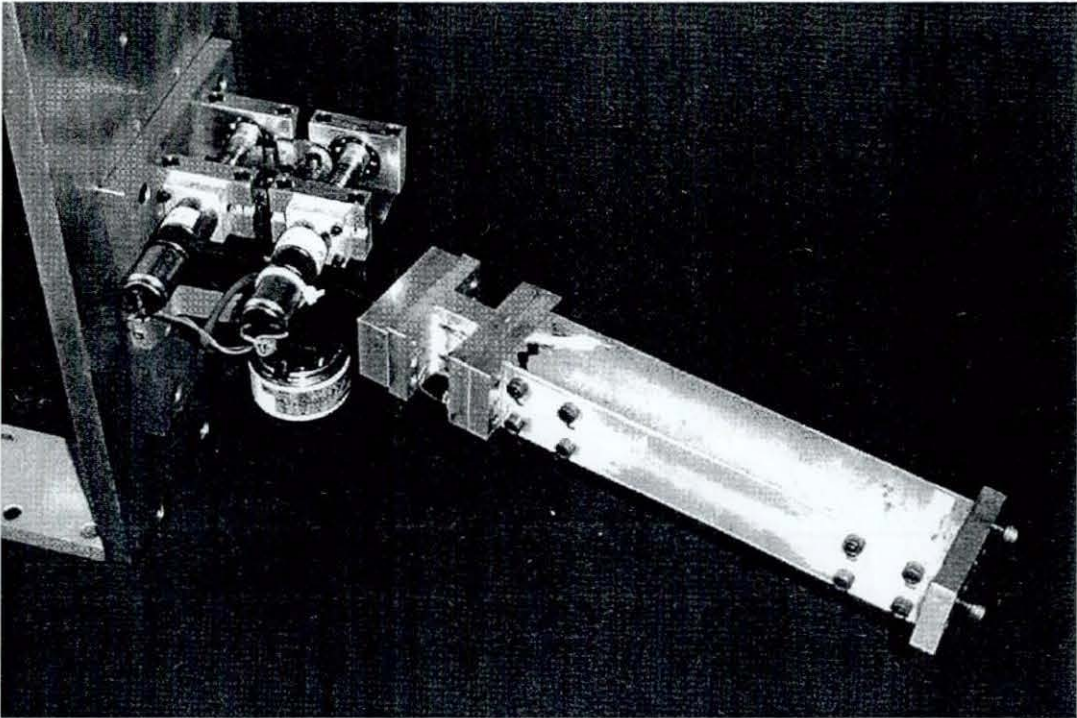


Fig. 5 Worm driven joint

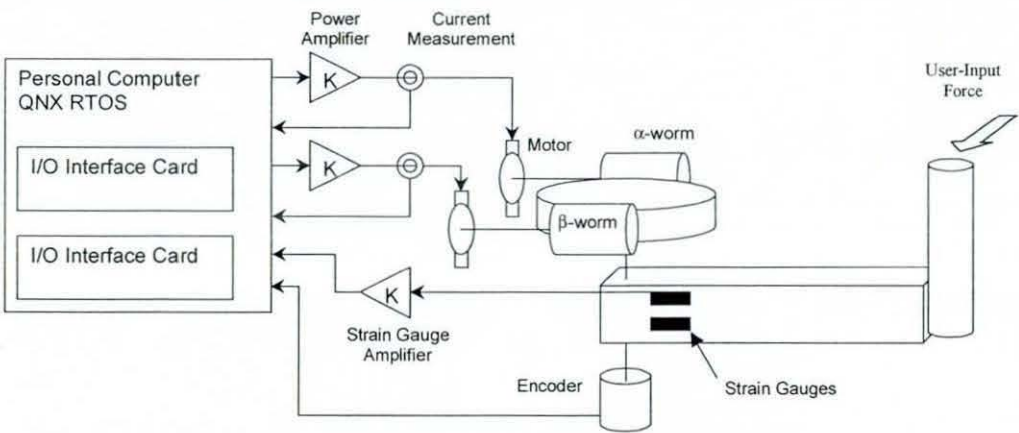


Fig. 6 Schematic overview of the control system

inertia were calculated from the joint geometry. The worm and worm wheel dimensions were assumed to be the pitch centre diameter of the respective gears, while

the friction coefficients were determined experimentally. It should be noted that there will be variations in the coefficients of friction owing to temperature and wear.

Table 1 System parameters

| | Worm wheel | α -Worm | β -Worm |
|--|------------------|------------------------|-----------------------|
| Inertia (kg m^2) | $J_0 = 0.062833$ | $J_\alpha = 0.002855$ | $J_\beta = 0.002855$ |
| Viscous damping (N m s/rad) | $C_{f0} = 0.005$ | $C_{f\alpha} = 0.001$ | $C_{f\beta} = 0.001$ |
| Gear geometry (m) | $r_0 = 0.025$ | $r_\alpha = 0.0065$ | $r_\beta = 0.0065$ |
| Static friction | | $\mu_{s\alpha} = 0.12$ | $\mu_{s\beta} = 0.16$ |
| Dynamic friction | | $\mu_{d\alpha} = 0.10$ | $\mu_{d\beta} = 0.14$ |
| Friction constant | | $\varepsilon = 0.001$ | $\varepsilon = 0.001$ |
| Worm lead angle (rad) | | $\gamma = 0.05236$ | $\gamma = 0.05236$ |

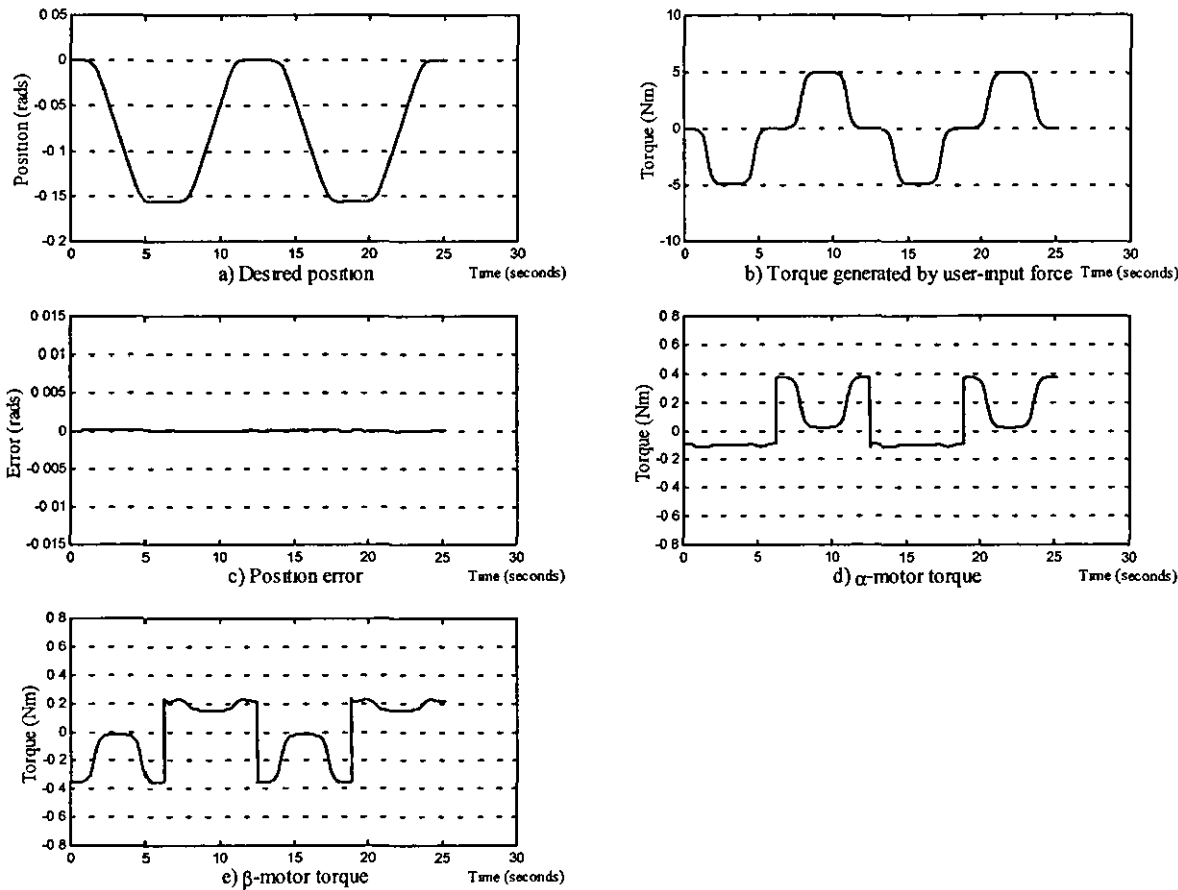


Fig. 7 Dual-worm joint simulation results

These variations may affect the implementation of this system in a multi-DOF robot. Therefore, the robot control system should be calibrated to update the estimates of the friction coefficients periodically, or whenever tracking performance is degraded. This could be done automatically.

6.1 Simulation results

In order to test the performance of the control strategy and algorithms, a simulation of the model, given by equation (7), was developed using SIMULINK. Parameters were set according to Table 1, and it was assumed that the control algorithms had exact knowledge of these parameters. Figure 7 shows the results obtained from the simulation. The user input that was applied to the system is given by equation (21) and is shown in Fig. 7b. Figure 7a shows the desired position generated from the speed demand, where $\dot{\theta}_d$ is given by equation (14) with $K_u = 0.01$. Sufficient tracking performance was achieved using the gains given in Table 2. The graph of position error given in Fig. 7c highlights this. Figures 7d and e show the applied α -motor and β -motor torque used to track the

desired trajectory.

$$\tau_u = 2.5 \times \text{sgn}[\sin(0.5t)] \{ \tanh[3 \cos(t)] - 1 \} \tag{21}$$

6.2 Experimental results

The experimental results for a joint with a single worm are compared with those of the dual-worm mechanism to illustrate backlash cancellation. In both experiments the trajectory is generated by manually applying a force to a control handle attached to the robot link. The velocity demand is generated on the basis of equation (14), where $K_u = 0.01$. The desired velocity is then digitally integrated and differentiated to give the desired position and desired acceleration respectively. Figure 8 presents the

Table 2 Simulation gains

| Control direction | Proportional | Integral | Derivative |
|---------------------|----------------------|----------------------|---------------------|
| α -Unwinding | $K_{p\alpha} = 5000$ | $K_{a\alpha} = 1000$ | $K_{d\alpha} = 100$ |
| β -Unwinding | $K_{p\beta} = 5000$ | $K_{a\beta} = 1000$ | $K_{d\beta} = 100$ |

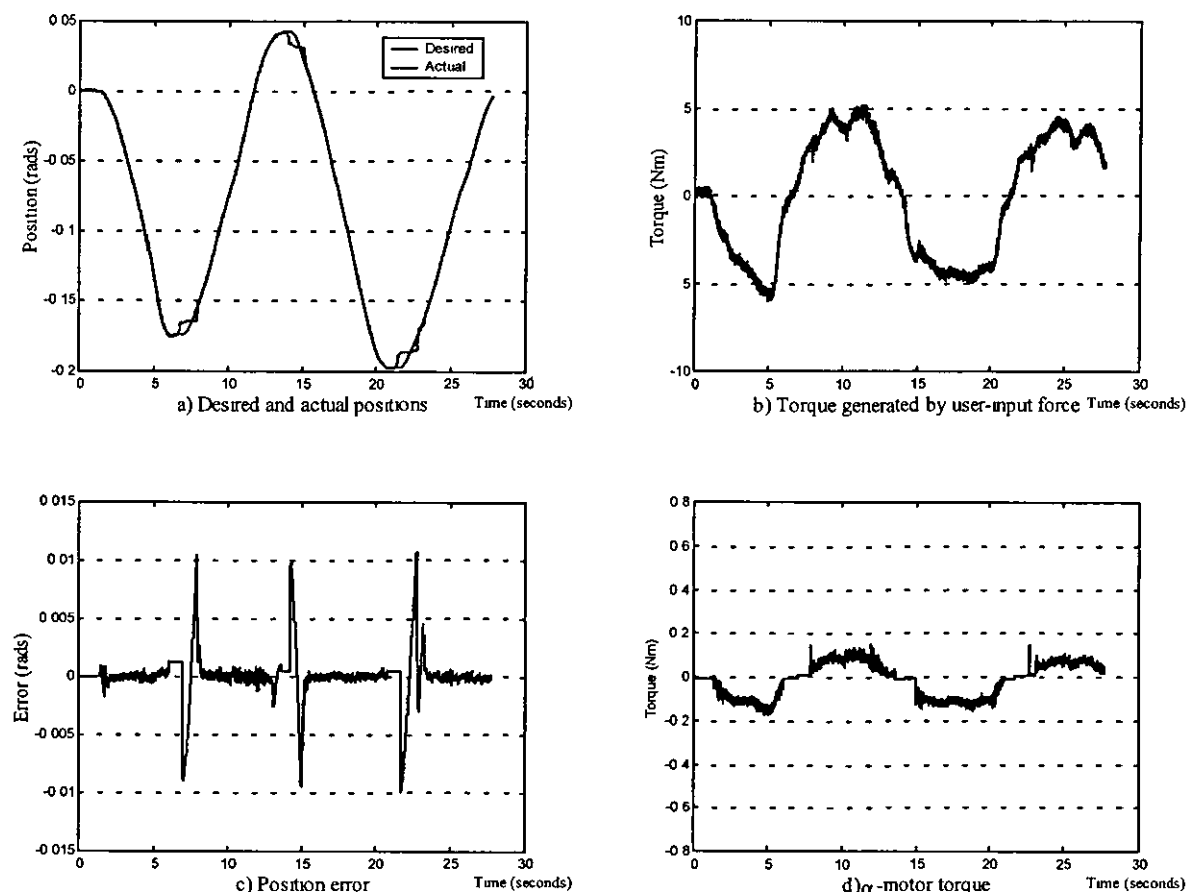


Fig. 8 Single-worm joint experimental results

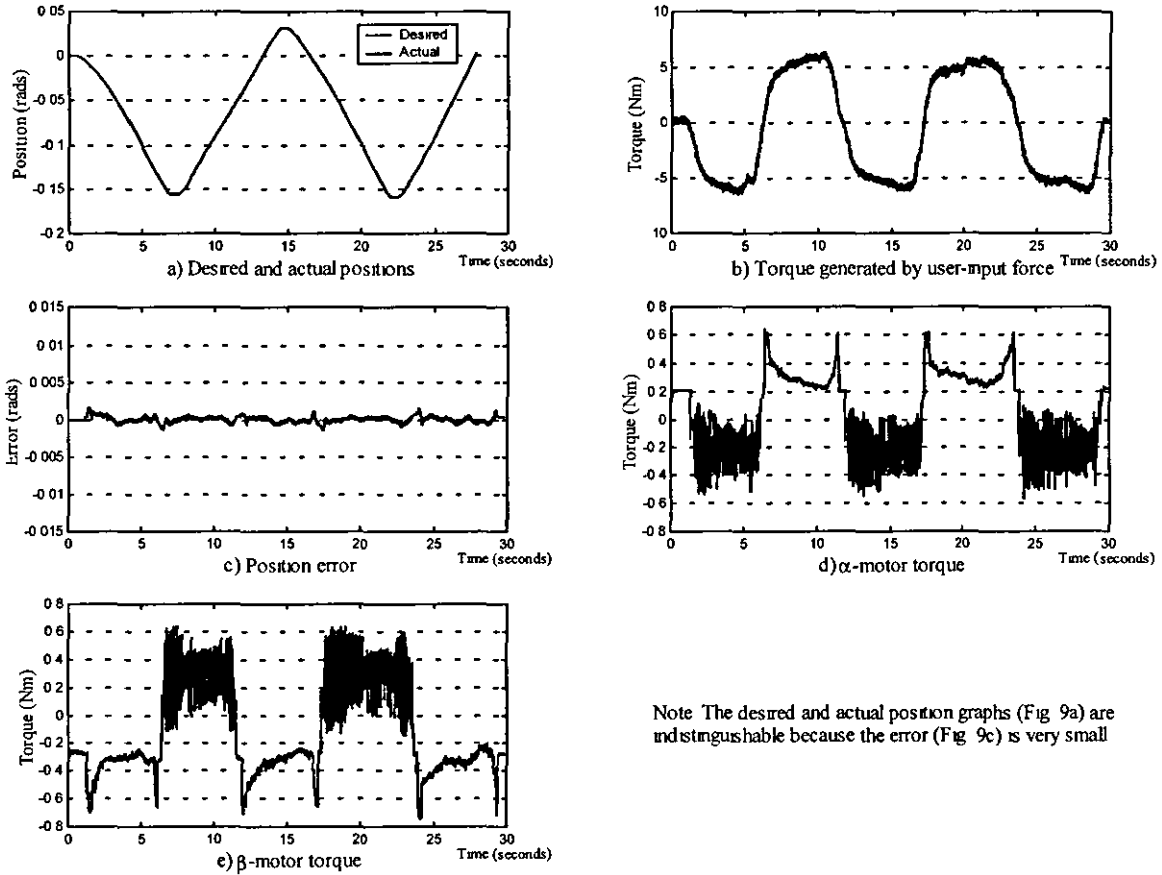
results for the single-worm mechanism. Figure 8a shows the position demand generated from the user-input force of Fig. 8b. Figure 8c displays the time history of the error during the movement, and Fig. 8d illustrates the motor torque used to drive the α -worm. It can be seen from Fig. 8c that the change in direction of the user-input force causes large errors in the order of ± 0.01 rad (or $\pm 0.57^\circ$). These errors are attributed to the backlash between the worm and worm wheel present in the single-worm system. Figure 9 presents the experimental results of a similar motion to that of the simulation (see Fig. 7) with the dual-worm mechanism. Figure 9a shows the desired and actual positions of the joint. The torque generated by the user-input force is shown in Fig. 9b, and the position error is illustrated in Fig. 9c. It should be noted that the graphs of actual and desired positions in Fig. 9a are indistinguishable because the error (shown in Fig. 9c) is very small. Figures 9d and e also show the motor output torque for the α - and β -motors respectively. It is possible to see from Fig. 9c that the backlash in the mechanism has been cancelled and the overall error has been reduced to less than ± 0.0015 rad (or $\pm 0.086^\circ$).

Tests have also been carried out to determine how the tool-tip position is affected by an external input from the user when the joint is stationary. It has been possible to

measure two different sources of error: flexibility of the link and shaft and also end-float of the worm owing to limited preloading of the worm angular contact bearings. Table 3 shows the deflection of the tool tip owing to the two kinds of error for user-input forces of 10 and 20 N applied at the tool tip. This error was measured at the extremity of the link, a distance 0.4 m from the centre of rotation. Although this single DOF robot does exhibit some flexibility, 0.4 mm of error per 10 N of user-input force is not considered excessive when compared with similar devices. However, increasing the stiffness of the system could further reduce the link and shaft flexibility.

7 CONCLUSIONS

In this paper a revolute joint that provides inherent safety features and cancels backlash has been presented. The non-backdriveable dual-worm mechanism is much safer than backdriveable systems in that no motor torque is required to keep the manipulator stationary for most forms of user-input force. However, high-frequency user input (i.e. shaking of the end-effector) can cause the worms to unwind, allowing the mechanism to exhibit



Note The desired and actual position graphs (Fig. 9a) are indistinguishable because the error (Fig. 9c) is very small

Fig. 9 Dual-worm joint experimental results

backlash. For this reason, when the joint is stationary, each motor applies a small amount of torque to keep the system backlash free.

Suitable control laws for regulating position have been derived on the basis of a mathematical model of the system. These computed-torque control algorithms have been shown to give good results while tracking a trajectory generated from the measured user-input torque. The control algorithm has been effective in cancelling backlash and compensating for the non-linear effects of friction generated at the interface of the two worms and worm wheel. The mechanical system can also be produced using basic manufacturing technologies and at a low cost when compared with industrial robotic systems.

Table 3 Static error measurements

| Applied force (N) | Deflection from zero (mm) | | |
|-------------------|---------------------------|---------------------------|-------------------------------|
| | Total | Due to end-float in worms | Due to link/shaft flexibility |
| 10 | 0.36 | 0.16 | 0.20 |
| 20 | 0.74 | 0.32 | 0.42 |

REFERENCES

- 1 Taylor, R. H., Mittelstadt, B. D. *et al.* An image-directed robotic system for precise orthopaedic surgery. *IEEE Trans Robotics and Automat.*, 1994, 10(3), 261–274.
- 2 Kazanzides, P., Mittelstadt, B. D. *et al.* An integrated system for cementless hip replacement. *IEEE Engng in Medicine and Biol.*, 1995, 14(3), 307–313.
- 3 Davies, B. L. A discussion of safety issues for medical robots. In *Computer-Integrated Surgery* (Eds R. H. Taylor, S. Lavallée, G. C. Burdea and R. S. Mosges), 1996, pp. 287–296 (MIT Press, Cambridge, Massachusetts).
- 4 Sainsbury Jr, J. K. The heart of microsurgery. *Mech. Engng.*, December 1998.
- 5 Harris, S. J., Lin, W. J., Fan, K. L., Hibberd, R. D., Cobb, J. and Davies, B. L. Experiences with robotic systems for knee surgery. *Lecture Notes in Computer Sci.*, 1997, 1205, 757–766.
- 6 Delnondedieu, Y. and Troccaz, J. PADyC: A Passive Arm with Dynamic Constraints. A prototype with two degrees of freedom. In *Proceedings of 2nd Annual International Symposium on Medical Robotics and Computer Assisted Surgery*, Baltimore, Maryland, 4–7 November 1995.
- 7 Troccaz, J. and Delnondedieu, Y. Semi-active guiding systems in surgery: A two-dof prototype of the Passive Arm with Dynamic Constraints (PADyC). *Mechatronics*, 1996, 6(4), 399–421.

- 8 Patton, W. J. *Mechanical Power Transmission*, 1980 (Prentice-Hall, Englewood Cliffs, New Jersey)
- 9 Bo, L. C. and Pavelescu, D. The friction-speed relationship and its influence on the critical velocity of stick-slip motion *Wear*, 1982, **82**, 277–289
- 10 Armstrong-Hélouvry, B., Dupont, P. and Canudas De Wit, C. A survey of models, analysis tools and compensation methods for the control of machines with friction *Automatica*, 1994, **30**(7), 1083–1138

H.2. Composite Adaptive Control

Reedman. A. V. C., and Bouazza-Marouf, K., Composite Adaptive Control of a Robotic Joint for Passive Deployment Applications, *Proceedings of the Institution of Mechanical Engineers, Part I Journal of Control and Systems Engineering*, **216**, 2002.

Composite adaptive control of a robotic joint for passive deployment applications

A V C Reedman and K Bouazza-Marouf*

Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Leicestershire, UK

Abstract: A composite adaptive control scheme for the control of an actively constrained revolute joint with backlash cancellation is presented in this paper. The drive mechanism consists of two motor-driven worms coupled to a single worm wheel. The mathematical model and control strategies are reviewed. This is followed by the derivation of the composite adaptive controllers. Simulation and experimental results show that the composite adaptive control scheme gives an equivalent performance to a computed-torque algorithm without compromising the mechanism's ability to cancel backlash.

Keywords: backlash cancellation, ^{adaptive}~~composite~~ control, robotics

NOTATION

| | |
|---------------------------|---|
| $C_{f\alpha}$ | viscous friction coefficient of the α worm system |
| $C_{f\beta}$ | viscous friction coefficient of the β worm system |
| $C_{f\theta}$ | viscous friction coefficient of the worm-wheel system |
| $C_{T\alpha}, C_{T\beta}$ | total system viscous friction with respect to the α and β motor torque |
| C_{α} | equivalent viscous friction coefficient of the α worm system |
| C_{β} | equivalent viscous friction coefficient of the β worm system |
| C_{θ} | equivalent viscous friction coefficient of the worm-wheel system |
| e | joint tracking error |
| $J_{T\alpha}, J_{T\beta}$ | total system inertia with respect to the α and β motor torque |
| J_{α} | inertia of the α worm and motor |
| J_{β} | inertia of the β worm and motor |
| J_{θ} | inertia of the worm-wheel, shaft and robot arm |
| $K_{v\alpha}, K_{v\beta}$ | controller gains for α and β unwinding |
| M_{α} | equivalent mass of J_{α} |
| M_{β} | Equivalent mass of J_{β} |
| M_{θ} | equivalent mass of J_{θ} |

| | |
|---------------------------------|--|
| P_{α}, P_{β} | adaptive update law gain matrix for α and β unwinding |
| r | filtered tracking error |
| r_{α} | radius of contact of the α worm with the worm-wheel |
| r_{β} | radius of contact of the β worm with the worm-wheel |
| r_{θ} | radius of contact of the worm-wheel with α and β worms |
| W_{α}, W_{β} | regression matrix for α and β unwinding control of the rotational system |
| $W_{f\alpha}, W_{f\beta}$ | filtered regression matrix |
| x | equivalent linear displacement of the α worm |
| y | equivalent linear displacement of the β worm |
| Y_{α}, Y_{β} | regression matrix for α and β unwinding control of the filtered tracking error system |
| z | equivalent linear displacement of the worm-wheel |
| γ | lead angle of the α and β worms |
| ε | friction constant |
| θ | link position |
| θ_d | desired link position |
| $\mu_{d\alpha}, \mu_{d\beta}$ | coefficient of dynamic friction of α and β worms |
| $\mu_{s\alpha}, \mu_{s\beta}$ | coefficient of static friction of α and β worms |
| $\tau_{f\alpha}, \tau_{f\beta}$ | filtered α and β motor torque |
| $\tau_{m\alpha}, \tau_{m\beta}$ | torque generated by driving α and β motors |

The MS was received on 14 September 2001 and was accepted after revision for publication on 26 February 2002

* Corresponding author Wolfson School of Mechanical and Manufacturing Engineering Loughborough University, Loughborough, Leicestershire LE11 3TU, UK.

| | |
|------------------------------|---|
| τ_u | equivalent torque generated by applied user-input force |
| Φ_a, Φ_b | vector of system parameters |
| $\hat{\Phi}_a, \hat{\Phi}_b$ | vector of system parameter estimates |

1 INTRODUCTION

Demand for robotic devices that interact closely with humans is increasing. For surgical intervention, robotic devices have been introduced into the operating theatre to help the surgeon perform surgical procedures with a higher degree of accuracy and reliability. However, many of the robotic devices currently in use are based on modified industrial manipulators and have typically been designed for high-speed, high-torque applications. The introduction of a large, powerful robot into an environment such as the operating theatre casts doubt on the safety of the patient, surgeon and other operating room staff [1].

To this end, research in the field of robot-assisted surgery has increasingly been directed at making the surgeon physically interact with custom-built robotic devices. By grasping a control handle the user is required to apply a force to the manipulator in order to make it move. These devices have been classified as either passive or active. A passive device is one that cannot provide a force/torque to the links of the robot that would cause motion. The classic example of a passive device is PADyC [2]. At every joint PADyC has a pair of over-running clutches, each running on a separate motor-driven hub. By controlling the speed of each motor one clutch limits the maximum allowable speed in the clockwise direction while the second clutch restricts speed in the anticlockwise direction. It was demonstrated that PADyC could be used to confine the motion of the surgeon within a predefined work area and, to a limited extent, follow a predefined path. With 20 N of force applied on the control handle, the two-degree-of-freedom (DOF) prototype of PADyC exhibited up to 20 mm of error at the tool tip with link lengths of 0.25 m. The error was attributed to joint flexibility and backlash in the clutch mechanism. Errors of this magnitude cannot be tolerated in most surgical applications.

Another passive device, the 3-DOF COBOT (collaborative robot) developed at Northwestern University (Illinois), uses a continuously variable transmission (CVT) for each joint [3]. Each CVT consists of a sphere, a pair of separate drive rollers coupled to output shafts and two steering wheels that control the relative velocities of the drive shafts. By coupling these CVTs to a common drive unit the robot is kinematically constrained to follow the desired path, i.e. forcing motion of one joint causes motion of the end-effector along the path. However, as the CVT relies on a limited amount of friction between the drive rollers and the sphere to

maintain position, the user can overpower the CVT, forcing it to deviate from the path.

Active devices, on the other hand, use motors directly coupled to the joints to provide a motive force. ACROBOT, developed at Imperial College (London), is a 4-DOF manipulator designed to help the surgeon perform total knee replacement surgery [4]. The surgeon controls the motion of the manipulator by applying a force to the control handle that is attached to the end-effector. The motorized joints are controlled to allow motion in the direction required by the surgeon. All of the joints are backdriveable and a d.c. motor at each joint controls the resistive force that the surgeon feels using a force control strategy. The workspace of the robot is actively constrained to confine the end-effector within a pre-planned safe working region. However, while the ACROBOT system requires physical input from the surgeon in order to move the manipulator, the force-controlled servomotors in each joint are still powerful enough to provide motion against the user. This again raises many issues regarding surgeon and patient safety. Although passive devices potentially have significant advantages over active mechanisms in terms of safety, thus far the positional accuracy of such devices while tracking a predefined path/trajectory has been absent.

In a previous paper, an actively constrained, revolute robotic joint was presented for use in applications where direct interaction with humans is required [5]. The non-backdriveable dual-worm mechanism was shown to be capable of cancelling the effects of backlash. A mathematical model of the mechanism was presented and was used to develop a computed-torque control algorithm for tracking a desired path. The algorithm guarantees exact tracking when the robot parameters are known exactly. However, non-linear robot parameters (including frictional and inertial terms) are notoriously difficult to estimate. In this paper, a composite adaptive control scheme is presented that is used to estimate system parameters such as inertia, damping coefficients and friction coefficients. A review of the concept of the dual-worm robot joint mechanism is given, followed by a statement of the mathematical model of the gear system. The strategy for control of the mechanism is presented along with a discussion on the selection of an appropriate adaptive control scheme. Simulation and experimental results are presented showing a comparison between the performance of the computed torque and composite adaptive controllers.

2 CONCEPTUAL OVERVIEW

In gear system design, a small amount of backlash is required in order to allow for thermal expansion, lubrication and lower frictional losses within a drive train [6]. In many robotic applications, even minute amounts

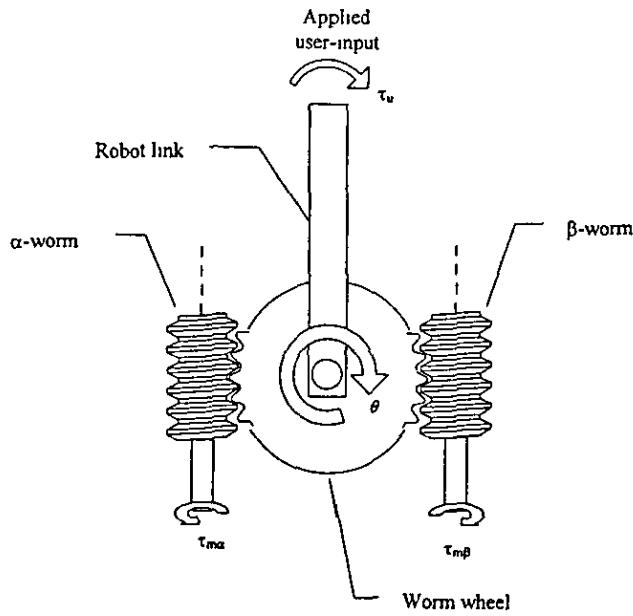


Fig. 1 Dual-worm drive joint

of backlash can cause errors in position that are not acceptable. For example, if a single link of a robot arm exhibits ± 0.0087 rad (or $\pm 0.5^\circ$) of backlash there will be an error of ± 3.5 mm at the end of a 0.4 m link.

The proposed system is a robotic joint designed to cancel backlash. The joint, shown schematically in Fig. 1, consists of two worms each driven by a low-power d.c. servomotor. The two worms follow a worm-wheel that is fixed to the robot link. The user provides a force on the end-effector and motion is allowed by controlling the two worms simultaneously. The non-backdriveability of the worm mechanism means that no matter how much force the operator applies, motion is not allowed until both motors are controlled to move in the same direction.

The parallel nature of this overactuated joint structure makes this active system safer than modified industrial robots. If there was a failure, both motors would have to drive in the same direction for any motion to occur. However, if this failure occurs, acceleration of the joints would be very small, due to the inertia of the links and low power of the motors, giving the operator plenty of time to react, i.e. to release the dead-man's handle. The dual-worm mechanism also has the ability to eliminate backlash using a control strategy which is discussed below. The proposed control of the joint is robust against the non-linear effects of inertia and frictional forces at the worm/wheel interface.

3 MATHEMATICAL MODEL

The mathematical model of the dual-worm driven mechanism is given by [8]

106701 © IMechE 2002

106701 © IMechE 2002

The Charlesworth Group, Huddersfield 01484 517077

$$J_T \ddot{\theta} + C_T \dot{\theta} = \tau_u + \frac{\tau_{ma} r_\theta}{\Delta_\alpha r_\alpha} + \frac{\tau_{mb} r_\theta}{\Delta_\beta r_\beta} \quad (1)$$

where

$$J_T = \left(J_\theta + \frac{J_\alpha r_\theta^2}{\Delta_\alpha r_\alpha^2 \tan \gamma} + \frac{J_\beta r_\theta^2}{\Delta_\beta r_\beta^2 \tan \gamma} \right)$$

$$C_T = \left(C_{r\theta} + \frac{C_{f\alpha} r_\theta^2}{\Delta_\alpha r_\alpha^2 \tan \gamma} + \frac{C_{f\beta} r_\theta^2}{\Delta_\beta r_\beta^2 \tan \gamma} \right)$$

$$\Delta_\alpha = \frac{\sin \gamma + [\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{-|\dot{\theta}|/(\epsilon \sin \gamma)}] \operatorname{sgn}(\dot{\theta}) \cos \gamma}{\cos \gamma - [\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{-|\dot{\theta}|/(\epsilon \sin \gamma)}] \operatorname{sgn}(\dot{\theta}) \sin \gamma}$$

$$\Delta_\beta = \frac{\sin \gamma - [\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{-|\dot{\theta}|/(\epsilon \sin \gamma)}] \operatorname{sgn}(\dot{\theta}) \cos \gamma}{\cos \gamma + [\mu_{d\beta} + (\mu_{s\beta} - \mu_{d\beta}) e^{-|\dot{\theta}|/(\epsilon \sin \gamma)}] \operatorname{sgn}(\dot{\theta}) \sin \gamma}$$

The friction forces between the two worms and the wheel are modelled using an exponential stick-slip friction model [7]

4 CONTROL STRATEGY

The aim of the control algorithm is to make the manipulator track a desired position command while cancelling backlash at the worm interface. In order to ensure surgeon and patient safety, and joint and control method must not exhibit

- (a) any motion against the user,
- (b) any motion without direct control from the user or
- (c) any backlash at the worm/wheel interface

To this end, effective control of the dual-worm mechanism requires two algorithms. In the clockwise direction (i.e. when the equivalent torque generated by the user-input force, $\tau_u > 0$) the motor command voltage for the α worm is set to a constant value and the β motor torque, τ_{mb} , is controlled to unwind the β worm to track the trajectory. In this condition the β worm *leads* and the α worm is used to *follow* the motion of the worm-wheel without applying unnecessary frictional forces to the system. Control of motion in this manner will be termed β unwinding control. However, in the anticlockwise direction (i.e. $\tau_u < 0$) the β worm motor command voltage is set to a constant value and τ_{ma} is used to control the unwinding of the α worm to track the trajectory. This is termed α unwinding control. In this instance, the α worm *leads* and the β worm *follows*. This dual-control strategy with the proposed control algorithm is shown to work well.

The velocity command is generated from the user-input force in the following way

Proc Instn Mech Engrs Vol 216 Part I J Systems and Control Engineering

developed in Realman and Bouazza-Marouf [5],

$$\theta_d = \begin{cases} 0 \\ \theta_{d \max} \operatorname{sgn}(\tau_u) \left[-\frac{2|\tau_u - \tau_{u \min}|^3}{(\tau_{u \max} - \tau_{u \min})^3} + \frac{3|\tau_u - \tau_{u \min}|^2}{(\tau_{u \max} - \tau_{u \min})^2} \right] \\ \theta_{d \max} \operatorname{sgn}(\tau_u) \end{cases}$$

where $\theta_{d \max}$, $\tau_{u \min}$ and $\tau_{u \max}$ are positive constants chosen to give a smooth motion from the mechanism. The velocity-limiting algorithm of equation (2), shown graphically in Fig 2, is different from that used in Reedman and Bouazza-Marouf [5]. This strategy has been employed to ensure sufficient smoothness of the desired position, velocity and acceleration command signals.

Craig *et al* [8] presented an adaptive control scheme for the control of robotic systems that ensures convergence of the parameter error to zero under certain conditions on the desired trajectory, known as persistency of excitation. The control method that has been proposed here involves switching between two controllers: one controller for controlling motion in the positive direction and another for the negative direction. Subsequently, it is not possible to generate a persistently exciting trajectory for each controller. The control method of Craig *et al* also requires inversion of the manipulator inertia matrix and measurement of acceleration. Although the parameter resetting technique used in reference [8] ensures the existence of the inverse of the inertia matrix, the poor acceleration measurement (derived by twice differentiating the position with respect to time) is also undesirable.

Due to the requirement of acceleration measurement and the inability to generate a persistently exciting trajectory of the proposed control strategy, a control method that relies on a less restrictive condition than persistency of excitation, known as the infinite integral condition, is used. Slotine and Li [9] proposed a composite adaptive controller that extracts information from both the tracking error and a prediction error in the filtered joint torque. This method consists of filtering the joint torque, making an estimate of this filtered quantity, designing a controller to track the desired trajectory and finally sel-

$$\forall |\tau_u| < \tau_{u \min}$$

$$\forall \tau_{u \min} \leq |\tau_u| \leq \tau_{u \max} \quad (2)$$

$$\forall |\tau_u| > \tau_{u \max}$$

ecting a parameter update rule that ensures convergence of both the tracking and parameter errors to zero. This torque filtering method also removes the need for acceleration measurements and inversion of the manipulator inertia matrix.

5 COMPOSITE ADAPTIVE CONTROLLER

In the development of the composite adaptive controller, the error between the desired position and the actual position of the link is defined by

$$e = \theta_d - \theta \quad (3)$$

and the quantity r , known as the filtered tracking error, is defined by

$$r = e + \lambda e \quad (4)$$

Both α unwinding and β unwinding control strategies are developed below, followed by a stability analysis of the controllers.

5.1 α unwinding control

The equation of motion for the system of gears, given in equation (1), can be used to solve for the α motor torque, $\tau_{m\alpha}$, by multiplying both sides by $\Delta_\alpha r_\alpha / r_\theta$, i.e.

$$\tau_{m\alpha} = J_{T\alpha} \ddot{\theta} + C_{T\alpha} \dot{\theta} - \tau_u \frac{\Delta_\alpha r_\alpha}{r_\theta} - \tau_{m\beta} \frac{\Delta_\alpha r_\alpha}{\Delta_\beta r_\beta} \quad (5)$$

where

$$J_{T\alpha} = J_T \frac{\Delta_\alpha r_\alpha}{r_\theta}, \quad C_{T\alpha} = C_T \frac{\Delta_\alpha r_\alpha}{r_\theta}$$

The assumption is made that the transition between static and dynamic friction is very fast (because of the $1/\sin \gamma$ term in both Δ_α and Δ_β), hence, once motion has started Δ_α and Δ_β are constant. Under this assumption, it is possible to write $\tau_{m\alpha}$ as a linear combination of constant unknown parameters, Φ_α , and known functions, W_α . Therefore, equation (5) can be written as

$$\tau_{m\alpha} = W_\alpha \Phi_\alpha \quad (6)$$

where

$$W_\alpha = [\ddot{\theta} \quad \dot{\theta} \quad \tau_u \quad \tau_{m\beta}]$$

$$\Phi_\alpha = \left[J_{T\alpha} \quad C_{T\alpha} \quad -\frac{\Delta_\alpha r_\alpha}{r_\theta} \quad -\frac{\Delta_\alpha r_\alpha}{\Delta_\beta r_\beta} \right]^T$$

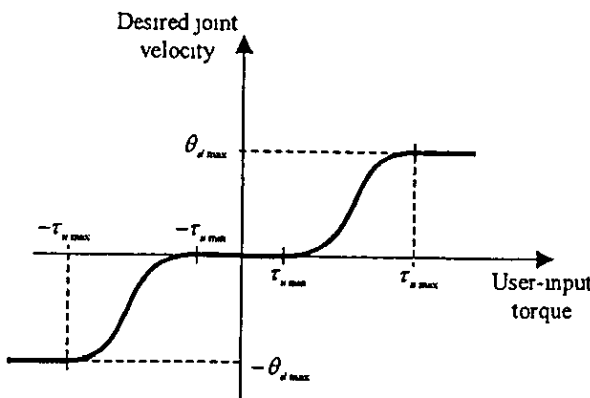


Fig. 2 User defined velocity demand

Equation (6) can also be written as

$$\tau_{ma} = h_a + g_a \quad (7)$$

where

$$h_a = \frac{\partial}{\partial t} (J_{Ta} \dot{\theta})$$

and

$$g_a = -J_{Ta} \ddot{\theta} + C_{Ta} \dot{\theta} - \tau_u \frac{\Delta_a r_a}{r_\theta} - \tau_{m\beta} \frac{\Delta_a r_a}{\Delta_\beta r_\beta}$$

Equation (5) has now been separated in a way that allows the acceleration term, $\ddot{\theta}$, to be filtered out. By filtering both sides of equation (7) it is possible to write the filtered torque equation given by

$$\tau_{fa} = f^* \tau_{ma} = f^* h_a + f^* g_a \quad (8)$$

where f is the impulse response of the linear stable, strictly proper filter, and $*$ is used to denote the convolution operation. By the property of convolution, it is also possible to write

$$f^* h_a = \int_0^t f(t-v) h_a(v) dv \quad (9)$$

and integrating by parts yields

$$f^* h_a = -f^* \dot{h}_a + f(0) h_a - f h_a(0) \quad (10)$$

Substitution of equation (10) into equation (8) and noting that J_{Ta} is constant and the filter and velocity are initialized to $f(0) = c$ and $\dot{\theta}(0) = 0$ respectively yields

$$\begin{aligned} \tau_{fa} &= -f^* (J_{Ta} \dot{\theta}) + c J_{Ta} \dot{\theta} \\ &\quad + f^* \left(C_{Ta} \dot{\theta} - \tau_u \frac{\Delta_a r_a}{r_\theta} - \tau_{m\beta} \frac{\Delta_a r_a}{\Delta_\beta r_\beta} \right) \\ &= W_{fa} \Phi_a \end{aligned} \quad (11)$$

where

$$W_{fa} = [-f^* \dot{\theta} + c \dot{\theta} \quad f^* \dot{\theta} \quad f^* \tau_u \quad f^* \tau_{m\beta}]$$

W_{fa} is referred to as the filtered regression matrix and Φ_a is the same as in equation (6). Assuming that τ_{ma} , $\tau_{m\beta}$, τ_u and θ are measurable, i.e. that the filtered regression matrix is known, it is possible to define the estimate of the filtered α motor torque, $\hat{\tau}_{fa}$, based on the estimate of the unknown parameters, $\hat{\Phi}_a$, such that

$$\hat{\tau}_{fa} = W_{fa} \hat{\Phi}_a \quad (12)$$

The error in the estimate of the filtered torque is defined as

$$\tilde{\tau}_{fa} = \tau_{fa} - \hat{\tau}_{fa} \quad (13)$$

In order to design a controller for the system, consider the Lyapunov-like function given by

$$V_a = \frac{1}{2} J_{Ta} r^2 + \frac{1}{2} \tilde{\Phi}_a^T P_a^{-1} \tilde{\Phi}_a \quad (14)$$

and its time derivative

$$\dot{V}_a = J_{Ta} r \dot{r} + \tilde{\Phi}_a^T P_a^{-1} \dot{\tilde{\Phi}}_a + \frac{1}{2} \dot{\tilde{\Phi}}_a^T P_a^{-1} \tilde{\Phi}_a \quad (15)$$

where $r = \dot{e} + \lambda e$ is the filtered tracking error, $e = \theta_d - \theta$ is the tracking error, $\tilde{\Phi}_a = \Phi_a - \hat{\Phi}_a$ is the error in the parameter estimates and P_a is the time-varying symmetric matrix. Making the following observation

$$J_{Ta} r = J_{Ta} (\theta_d + \lambda \dot{e}) - J_{Ta} \dot{\theta} \quad (16)$$

and replacing for $J_{Ta} \dot{\theta}$ from equation (5) yields

$$J_{Ta} r = J_{Ta} (\theta_d + \lambda \dot{e}) + C_{Ta} \dot{\theta} - \tau_{ma} - \tau_u \frac{\Delta_a r_a}{r_\theta} - \tau_{m\beta} \frac{\Delta_a r_a}{\Delta_\beta r_\beta} \quad (17)$$

which can be written as

$$J_{Ta} r = Y_a \Phi_a - \tau_{ma} \quad (18)$$

where $Y_a = [(\theta_d + \lambda \dot{e}) \quad \dot{\theta} \quad \tau_u \quad \tau_{m\beta}]$ and selecting the control input, τ_{ma} , as

$$\tau_{ma} = Y_a \hat{\Phi}_a + K_{va} r \quad (19)$$

equation (15) may be written as

$$\dot{V}_a = -K_{va} r^2 + \tilde{\Phi}_a^T (P_a^{-1} \dot{\tilde{\Phi}}_a + Y_a^T r) + \frac{1}{2} \dot{\tilde{\Phi}}_a^T P_a^{-1} \tilde{\Phi}_a \quad (20)$$

The least-squares adaptive update law proposed by Li and Slotine [10], given by the following equations, is used here:

$$\dot{\tilde{\Phi}}_a = \dot{\Phi}_a - \dot{\hat{\Phi}}_a = -P_a Y_a^T r - P_a W_{fa}^T \tilde{\tau}_{fa} \quad (21)$$

$$P_a^{-1} = W_{fa}^T W_{fa} \quad (22)$$

Using equation (21) and replacing for $\tilde{\tau}_{fa}$ from equations (12) and (13) and noting that Φ_a is a vector of constant parameters yields

$$\dot{\tilde{\Phi}}_a = P_a Y_a^T r + P_a W_{fa}^T (\tau_{fa} - W_{fa} \hat{\Phi}_a) \quad (23)$$

Using the matrix identity given by

$$\frac{dA^{-1}}{dt} = -A^{-1} \frac{dA}{dt} A^{-1} \quad (24)$$

equation (22) may be written more conveniently as

$$\dot{P}_a = -P_a W_{fa}^T W_{fa} P_a \quad (25)$$

Therefore, it follows that

$$\lim_{t \rightarrow \infty} \lambda_{\max} \{P_a\} = 0 \quad (26)$$

and

$$\lim_{t \rightarrow \infty} \lambda_{\min} \{P_a^{-1}\} = \infty \quad (27)$$

where $\lambda_{\min} \{A\}$ and $\lambda_{\max} \{A\}$ represent the minimum and maximum eigenvalues of the matrix A . Equation (27) is referred to as the infinite integral condition. Substitution of the parameter update law and least-squares estimator, given by equations (22) and (23) respectively, into equation (20) leads to

$$\dot{V}_a = -K_{va} r^2 - \frac{1}{2} \tilde{\Phi}_a^T W_{fa}^T W_{fa} \tilde{\Phi}_a \quad (28)$$

5.2 β unwinding control

Similarly, for β unwinding control, equation (1) can be used to solve for the β motor torque by multiplying both sides by $\Delta_\beta r_\beta / r_\theta$, as given by

$$\tau_{m\beta} = J_{T\beta} \ddot{\theta} + C_{T\beta} \dot{\theta} - \tau_u \frac{\Delta_\beta r_\beta}{r_\theta} - \tau_{m\alpha} \frac{\Delta_\beta r_\beta}{\Delta_\alpha r_\alpha} = W_\beta \Phi_\beta \quad (29)$$

where

$$W_\beta = [\ddot{\theta} \quad \dot{\theta} \quad \tau_u \quad \tau_{m\alpha}]$$

$$\Phi_\beta = \begin{bmatrix} J_{T\beta} & C_{T\beta} & -\frac{\Delta_\beta r_\beta}{r_\theta} & -\frac{\Delta_\beta r_\beta}{\Delta_\alpha r_\alpha} \end{bmatrix}^T$$

Following the method outlined above, the control input, $\tau_{m\beta}$, can be selected as

$$\tau_{m\beta} = Y_\beta \hat{\Phi}_\beta + K_{v\beta} r \quad (30)$$

The least-squares adaptive update law is given by

$$\dot{\hat{\Phi}}_\beta = -\dot{\Phi}_\beta = -P_\beta Y_\beta^T r - P_\beta W_{f\beta}^T \tilde{\tau}_{f\beta}$$

$$\dot{\hat{\Phi}}_\beta = P_\beta Y_\beta^T r + P_\beta W_{f\beta}^T (\tau_{f\beta} - W_{f\beta} \hat{\Phi}_\beta) \quad (31)$$

$$\dot{P}_\beta^{-1} = W_{f\beta}^T W_{f\beta}$$

$$\dot{P}_\beta = -P_\beta W_{f\beta}^T W_{f\beta} P_\beta \quad (32)$$

and the time derivative of the Lyapunov function is given by

$$\dot{V}_\beta = -K_{v\beta} r^2 - \frac{1}{2} \tilde{\Phi}_\beta^T W_{f\beta}^T W_{f\beta} \tilde{\Phi}_\beta \quad (33)$$

5.3 Stability analysis

The Lyapunov functions for α and β unwinding may be written as

$$V_i = \frac{1}{2} J_{Ti} r^2 + \frac{1}{2} \tilde{\Phi}_i^T P_i^{-1} \tilde{\Phi}_i \quad (34)$$

where the subscript i should be replaced by α for α unwinding or β for β unwinding. Similarly, the time derivative of the Lyapunov function, given by equation (34), can also be expressed as

$$\dot{V}_i = -K_{vi} r^2 - \frac{1}{2} \tilde{\Phi}_i^T W_{fi}^T W_{fi} \tilde{\Phi}_i \leq 0 \quad (35)$$

By showing that the second derivative of the Lyapunov function is bounded, so proving that \dot{V}_i is uniformly continuous, Barbalat's lemma can be invoked to show that the tracking error, r , and parameter error, $\tilde{\Phi}_i$, both converge to zero. Barbalat's lemma states that if $\dot{x}(t)$ is a uniformly continuous, real function and $\lim_{t \rightarrow \infty} x(t) = k < \infty$, then $\lim_{t \rightarrow \infty} \dot{x}(t) = 0$ [11]. Firstly, \dot{V}_i is obtained by differentiating equation (35) to give

$$\dot{V}_i = -2K_{vi} r \dot{r} - \tilde{\Phi}_i^T W_{fi}^T W_{fi} \dot{\tilde{\Phi}}_i - \tilde{\Phi}_i^T W_{fi}^T \dot{W}_{fi} \tilde{\Phi}_i \quad (36)$$

Therefore, in order to prove that \dot{V}_i is bounded and \dot{V}_i is uniformly continuous, it is necessary and sufficient to prove that r , \dot{r} , $\tilde{\Phi}_i$, $\dot{\tilde{\Phi}}_i$, W_{fi} and \dot{W}_{fi} are bounded. \dot{V}_i is at least negative semi-definite, implying that $V_i \leq V_i(0)$ and r and $\tilde{\Phi}_i$ are bounded. Subsequently, from the definition of r in equation (4), e and \dot{e} are also bounded. Considering Y_i as defined in equation (18), i.e. $Y_i = [\ddot{\theta}_d + \lambda \dot{e} \quad \dot{\theta} \quad \tau_u \quad \tau_{mi}]^T$, where i should be replaced by β for α unwinding and α for β unwinding, and assuming that the inputs, τ_u , τ_{mi} , θ_d , $\dot{\theta}_d$ and $\ddot{\theta}_d$, are bounded, then θ , $\dot{\theta}$ and Y_i must also be bounded. Substitution of the control law (19) or (30) into equation (18) leads to

$$J_{Ti} \dot{r} = Y_i \tilde{\Phi}_i - K_{vi} r \quad (37)$$

Noting that $J_{Ti} > 0$, then \dot{r} and $\ddot{\theta}$ are also bounded. Subsequently, from equation (5) or (29) and from the assumption that Φ_i is bounded, τ_{mi} , W_i and $\hat{\Phi}_i$ are bounded. From the definition of the filter, f , it can be seen that τ_{fi} , W_{fi} and \dot{W}_{fi} are also bounded. Finally, to prove that \dot{V}_i is bounded the parameter update law of equation (23) or (31) may be written as

$$\dot{\tilde{\Phi}}_i = -P_i Y_i^T r - P_i W_{fi}^T \tilde{\tau}_{fi} \quad (38)$$

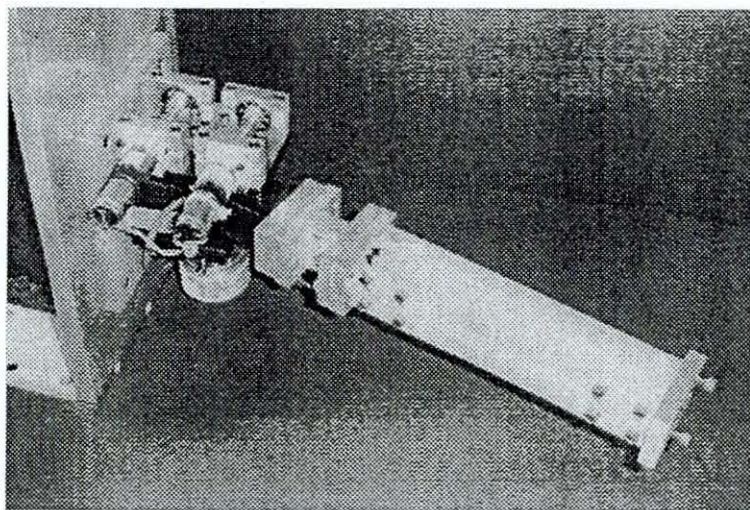


Fig. 3 Worm driven joint

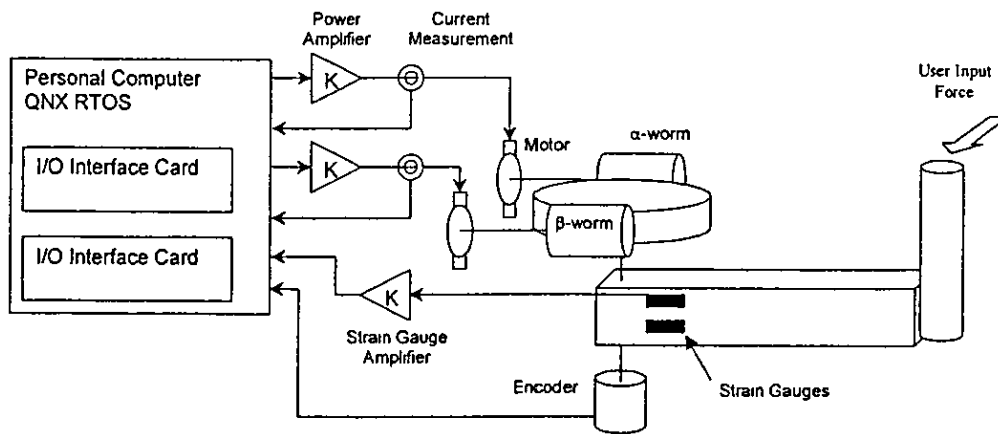


Fig. 4 Schematic overview of the control system

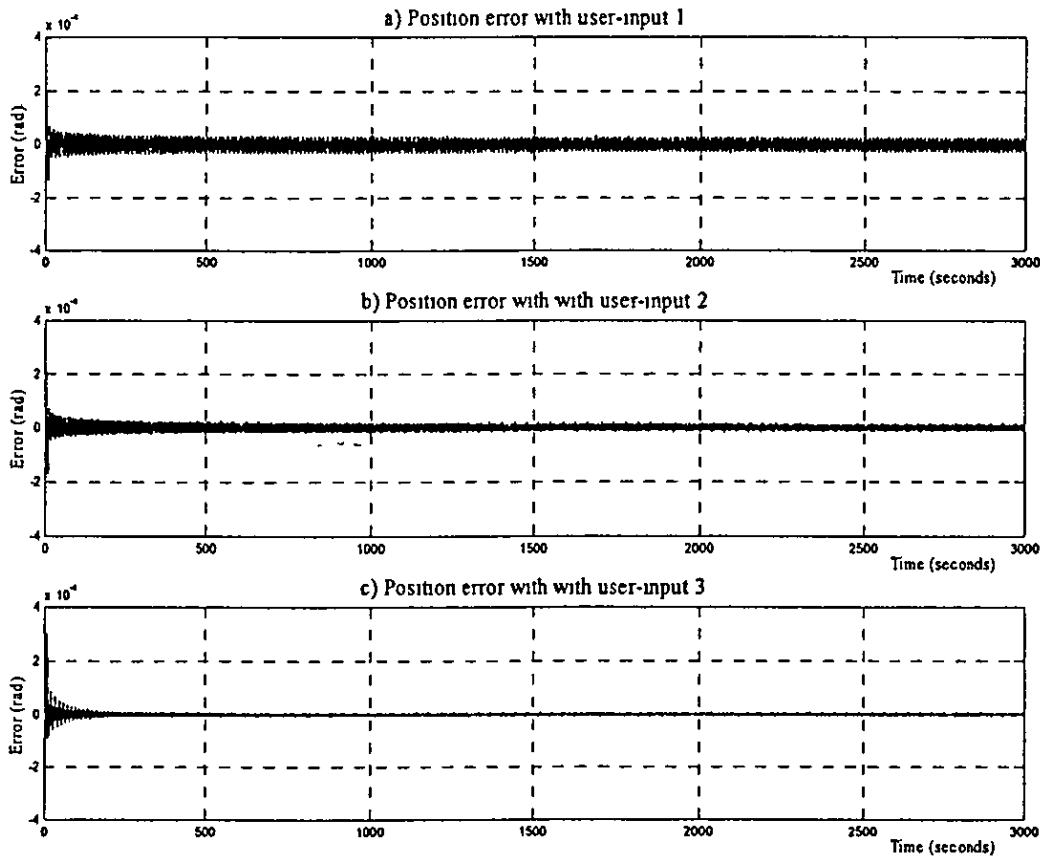


Fig. 5 Comparison of tracking errors during the simulation of the three user-input torque commands

From equations (22) or (23), P_t is bounded by $P_t(0)$ since P_t is negative semi-definite. From equations (12) and (13) and the fact that the filter, f , has a stable transfer function, it can be seen that $\dot{\Phi}_t$ is bounded. Application of Barbalat's lemma yields

$$\lim_{t \rightarrow \infty} V_t = 0 \quad (39)$$

and

$$\lim_{t \rightarrow \infty} r = 0 \quad (40)$$

Since the relationship between r and e , given in equation (4), may also be written in terms of the strictly proper, asymptotically stable transfer function, $H(s)$, such that

$$e(s) = H(s)r(s) \quad (41)$$

it can be concluded that

$$\lim_{t \rightarrow \infty} r = 0 \Rightarrow \lim_{t \rightarrow \infty} e = 0 \quad (42)$$

It is also possible to define the type of stability for the

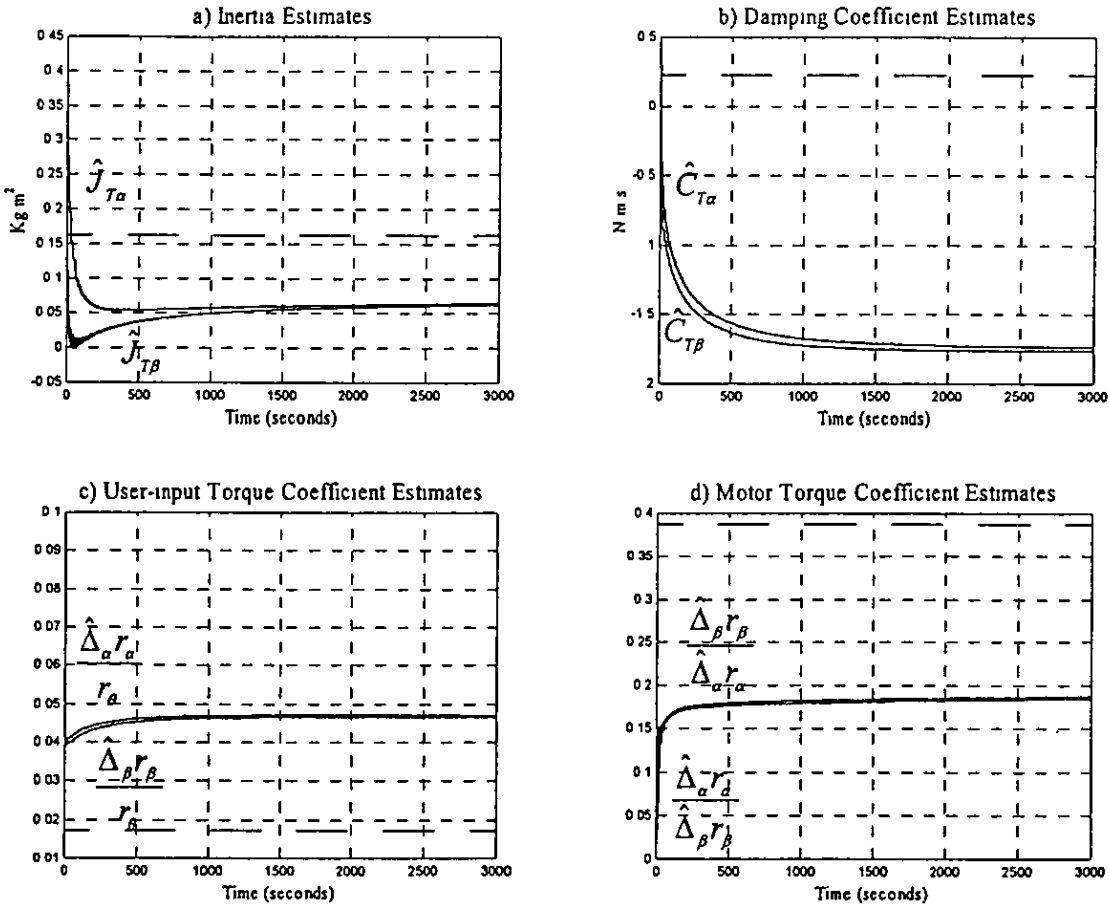


Fig 6 Simulation parameter estimates with user-input 1

parameter error Since V_i is bounded by $V_i(0)$, if the infinite integral condition given in equation (27) holds, it must be concluded that the parameter error tends to zero, i.e. $\lim_{t \rightarrow \infty} \tilde{\Phi}_i = 0$. The infinite integral condition is less restrictive than the persistency of excitation condition discussed above, owing to the fact that as long as there is input to the system and that the system is capable of motion it is easy to prove that equation (27) holds true. A persistently exciting trajectory/input, on the other hand, is not always easy to derive or generate in most robotic applications.

6 EXPERIMENTAL SET-UP

Figure 3 shows the dual-worm driven joint mechanism. The joint is controlled using a Pentium 233 MHz personal computer running the QNX 4.25 real-time operating system [12]. A schematic diagram of the control system is shown in Fig 4.

The control algorithm requires measurements of motor torque, user-input force and position. The user-input force is measured using four strain gauges, in a Wheatstone bridge configuration, mounted on a specially designed section of the link. Closed-loop regu-

lation of the motor output torque is accomplished by measuring the armature current and implementing a digital PI (proportional+integral) controller. All analogue signals are measured and generated using 12-bit analogue-to-digital converters and 12-bit digital-to-analogue converters. The link rotation is recorded using an encoder and appropriate electronics to generate 20000 counts per revolution, giving a resolution of 0.0003142 rad/count (or 0.018°/count). The encoder position is read as a 24-bit number from an HCTL-1100 motion control interface. The frequency of the control loop is set at 600 Hz and link rotational velocity measurement is obtained in software by using a backward difference algorithm.

7 RESULTS AND DISCUSSION

7.1 Simulation results

The mathematical model of the dual worm-driven joint was simulated using SIMULINK. The parameters of the model were set according to Table 1 and the control parameters set as follows: tracking error filter gain $\lambda = 50$ and controller gains $K_{va} = K_{vb} = 20$. The dynamic effects

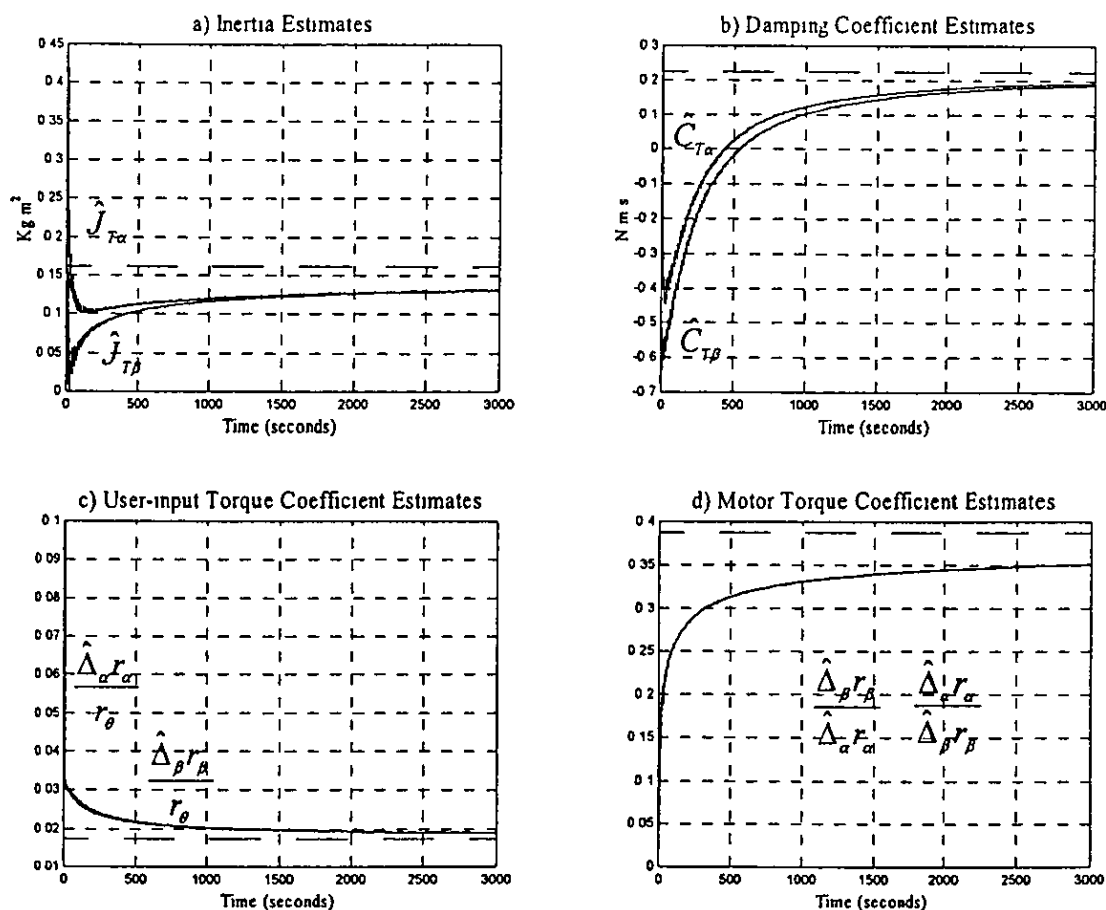


Fig. 7 Simulation parameter estimates with user-input 2

of the α and β motors are cancelled using a high-gain PI torque control feedback loop, assuming that motor output is measured without error. During the simulation the β motor is set to follow the worm with constant torque under α unwinding control. Similarly, during β unwinding the α motor is also set to follow the worm-wheel constant torque. The matrices P_α and P_β were chosen to give well-damped parameter estimates, and the parameter estimates themselves, $\hat{\Phi}_\alpha$ and $\hat{\Phi}_\beta$, were initialized with arbitrary values according to

$$P_\alpha(0) = P_\beta(0) = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

$$\Phi_\alpha(0) = \begin{bmatrix} 0.4 \\ 0.1 \\ 0.1 \\ 0.5 \end{bmatrix}, \quad \Phi_\beta(0) = \begin{bmatrix} 0.1 \\ 0.3 \\ 0.1 \\ 0.1 \end{bmatrix} \quad (43)$$

Torque filtering was performed using the linear filter given by the first-order transfer function of equation

(44). The same filter was used in both the α -unwinding and β -unwinding controllers

$$f(s) = \frac{1}{s+1} \quad (44)$$

Three user-input torque commands, τ_{u1} , τ_{u2} and τ_{u3} , given by the following equations respectively, were applied to the model

$$\tau_{u1} = 5 \tanh[\cos(4t) - 0.76152] \operatorname{sgn}[\sin(t/2)] \quad (45)$$

$$\tau_{u2} = 8 \tanh[\cos(4t) - 0.76152] \operatorname{sgn}[\sin(t/2)] \quad (46)$$

$$\tau_{u3} = 8[\cos(4t) + 1.5] \sin(t/2) \quad (47)$$

The desired velocity for the system is calculated using equation (2) with $\dot{\theta}_{d,\max} = 0.1$ rad/s, $\tau_{u,\max} = 10$ N m and $\tau_{u,\min} = 1$ N m. With the system parameters given by Table 1, the actual parameter vectors Φ_α and Φ_β can be calculated for $|\dot{\theta}| \neq 0$, as given by

$$\Phi_\alpha = \Phi_\beta = \begin{bmatrix} 0.162 \\ 0.223 \\ 0.017 \\ 0.387 \end{bmatrix} \quad (48)$$

Figure 5 shows the position error recorded during the

Table 1 Simulation model parameters

| | Worm-wheel | α and β worm |
|-------------------|--|--|
| Inertia | $J_{\theta} = 0.25 \text{ kg m}^2$ | $J_{\alpha} = J_{\beta} = 0.003698 \text{ kg m}^2$ |
| Viscous damping | $C_{\theta} = 0.005 \text{ N m s/rad}$ | $C_{f\alpha} = C_{f\beta} = 0.005 \text{ N m s/rad}$ |
| Gear geometry | $r_{\theta} = 0.025 \text{ m}$ | $r_{\alpha} = r_{\beta} = 0.0065 \text{ m}$ |
| Static friction | | $\mu_{s\alpha} = \mu_{s\beta} = 0.14$ |
| Dynamic friction | | $\mu_{d\alpha} = \mu_{d\beta} = 0.12$ |
| Friction constant | | $\varepsilon = 0.001$ |
| Worm lead angle | | $\gamma = 0.05236 \text{ rad}$ |

simulations for each user-input. The various rates at which the magnitude of the tracking error converges to zero can be clearly seen. Figures 6, 7 and 8 show the parameter estimates of the simulated system with the user-inputs defined by equations (45), (46) and (47) respectively. The expected values of the parameters are shown by dotted lines. Figure 6 illustrates that for the first user-input command [given by equation (45)] the estimated parameters do not converge on the expected values given by equation (48). However, with the two user-input commands given by equations (46) and (47),

the parameter error converges to zero. The input given by equation (46) yields slower convergence of the parameter estimates to the true values than the input of equation (47), as shown in Figs 7 and 8 respectively.

It should be noted that the magnitude of the user-input force affects the convergence of the parameters. The problem arises from the choice of θ_d in equation (2). With the user-input defined by equation (45) the desired velocity function never saturates, i.e. $\dot{\theta}_d < \dot{\theta}_{d \max}$. Thus the dependence of θ_d on τ_u is very strong and as inputs they appear to be very similar. Under these conditions the composite adaptive controller cannot extract enough information from these inputs to determine the correct parameter estimates. However, the parameter estimates and tracking error remain bounded at all times.

The user-inputs given by equations (45) and (46) differ only in magnitude. This difference significantly changes the appearance of $\dot{\theta}_d$ with respect to τ_u in the region where $|\tau_u| > \tau_{u, \max}$. The composite adaptive controller is able to extract more information about the system with the user-inputs defined by equations (46) and (47). These effects must be taken into account during experimental work.

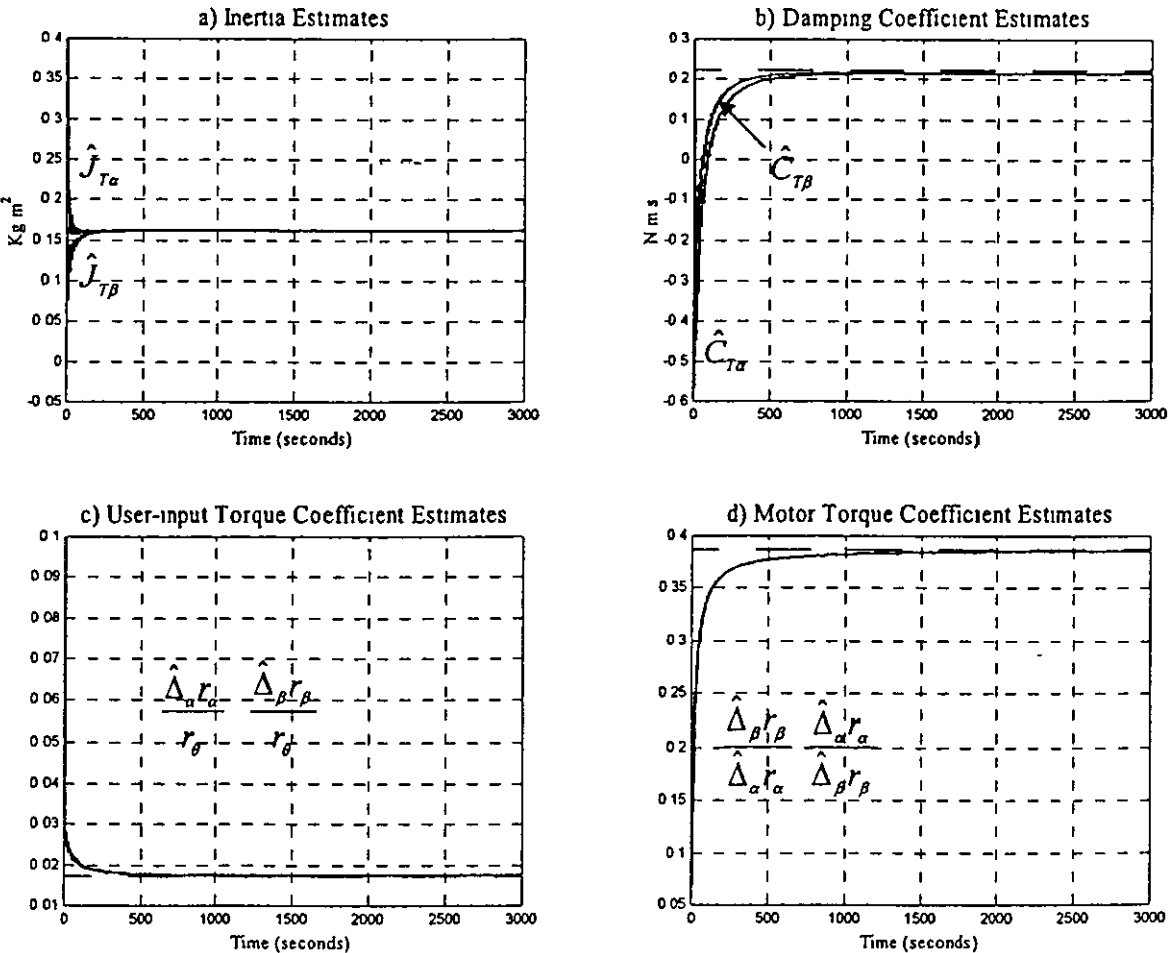


Fig. 8 Simulation parameter estimates with user-input 3

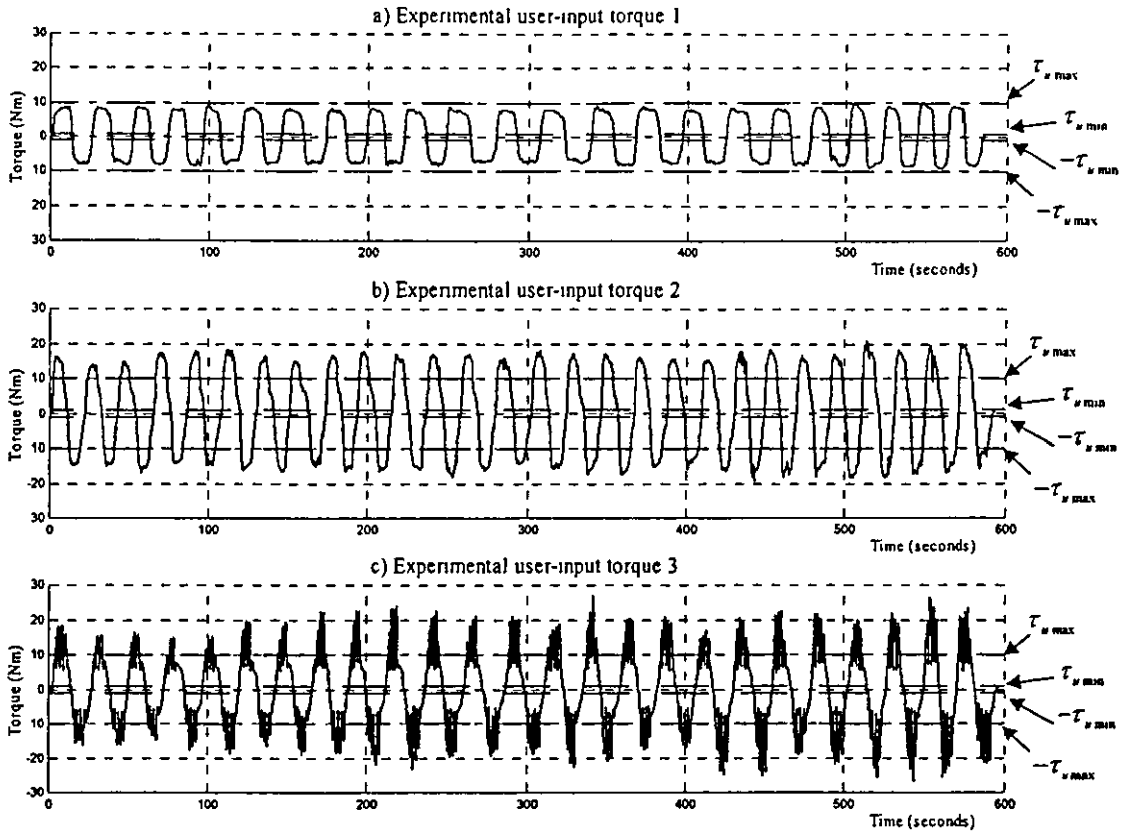


Fig. 9 Experimental user-input torque commands

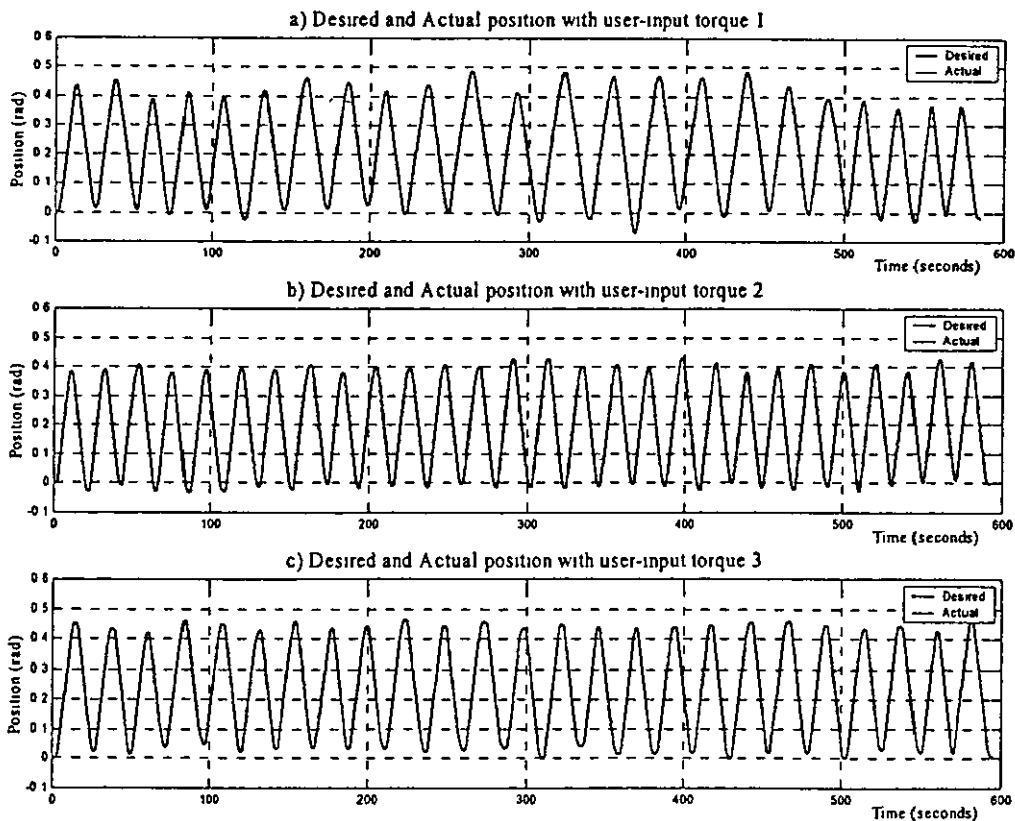


Fig. 10 Experimental desired and actual positions

(NB The desired and actual positions are indistinguishable because the error is very small as shown in fig 11)

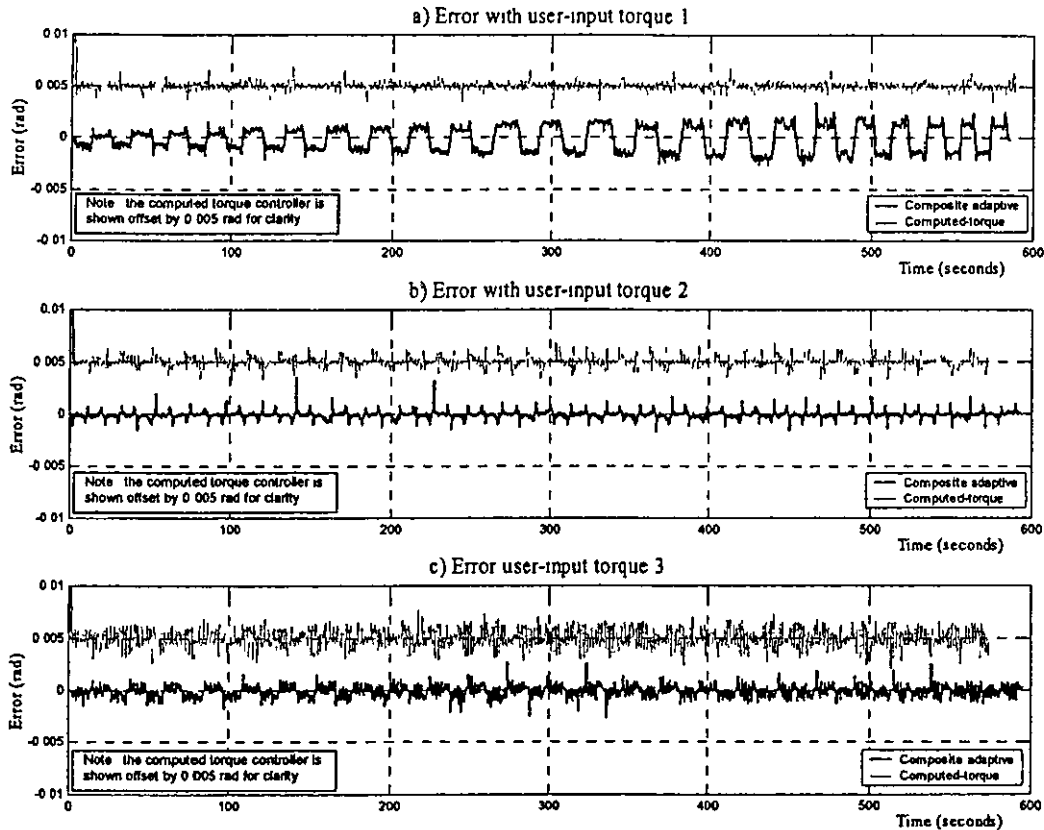


Fig. 11 Computed-torque control versus composite adaptive control

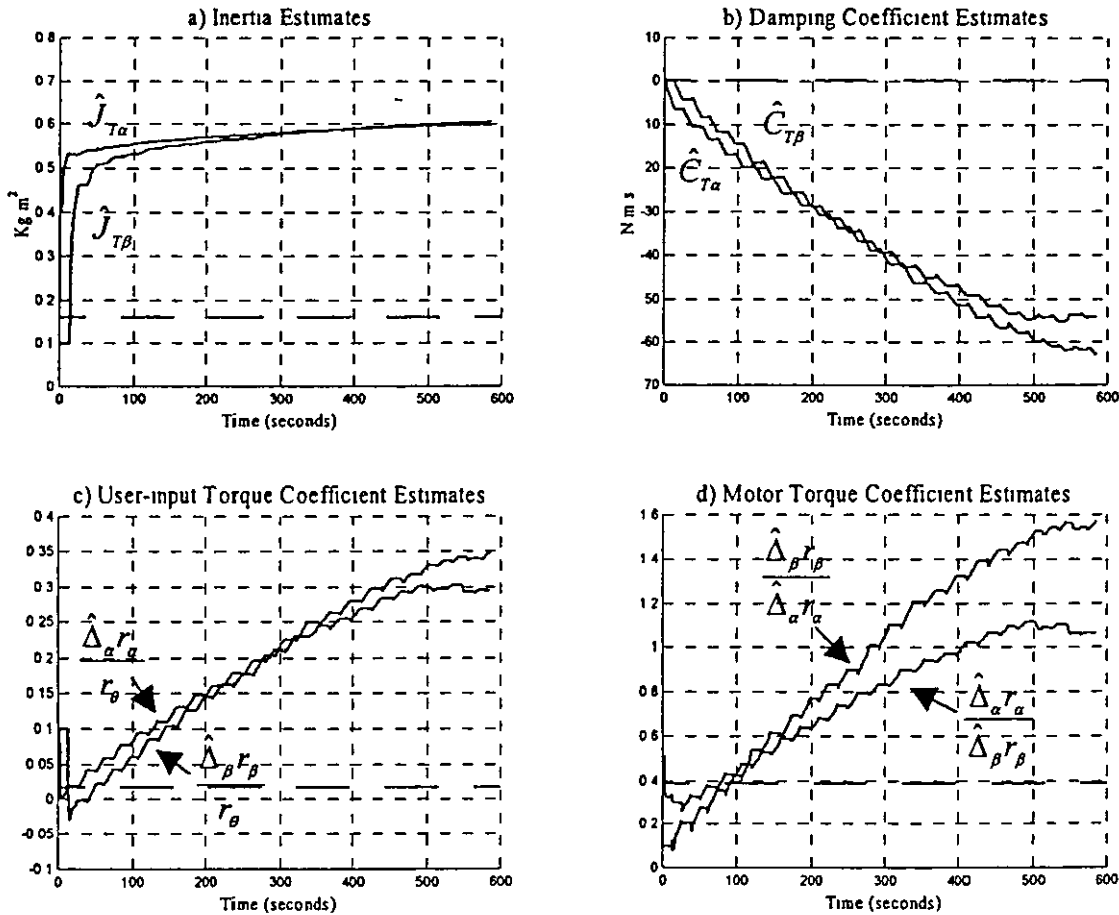


Fig. 12 Experimental parameter estimates with user-input 1

7.2 Experimental results and discussion

The experimental results of the computed-torque controller are compared to the results obtained from the composite adaptive controller. In both experiments the applied user-input force generates a command velocity based on equation (2), where $\theta_{d\max} = 0.05$ rad/s, $\tau_{u,\max} = 10$ N m, $\tau_{u,\min} = 1$ N m. The desired velocity is then digitally integrated and differentiated to give the desired position and desired acceleration respectively. The matrices P_α and P_β and the vectors $\hat{\phi}_\alpha$ and $\hat{\phi}_\beta$ were initialized as in equation (43) and the controller gains set as follows: tracking error filter gain $\lambda = 50$ and controller gains $K_{v\alpha} = K_{v\beta} = 35$. The torque filter that was used in the experiments for both α unwinding and β unwinding control is given by the first-order transfer function of

$$f(s) = \frac{10}{s + 10}$$

(49)

The filter of equation (49) used in these experiments differs from the filter used in the simulation tests [given in equation (44)] because it has been tuned to minimize

the effects of quantization and sensor noise that are not modelled in the simulation.

In a similar manner to the simulation tests, three user-input torque commands were manually applied to the joint mechanism, as shown by Fig. 9. Figures 9a, b and c show the user-input commands similar to those given in equations (45), (46) and (47) respectively. Figure 10 shows the desired and actual position, derived from the integration of equation (2), for the three user-input torque commands. The graphs of desired and actual positions in each are indistinguishable because the error is very small.

The system position error using the composite adaptive and computed-torque controllers is shown in Fig. 11. The error has been shown using a small scale, compared to Fig. 10, to enlarge the graph. Although it has not been possible to generate exactly the same trajectory, Fig. 11 shows a comparison between the composite adaptive control scheme developed in this paper and the computed-torque method developed previously [5]. Figure 11a shows the performance of the computed-torque controller compared to that of the composite adaptive controller (with the user-input command of

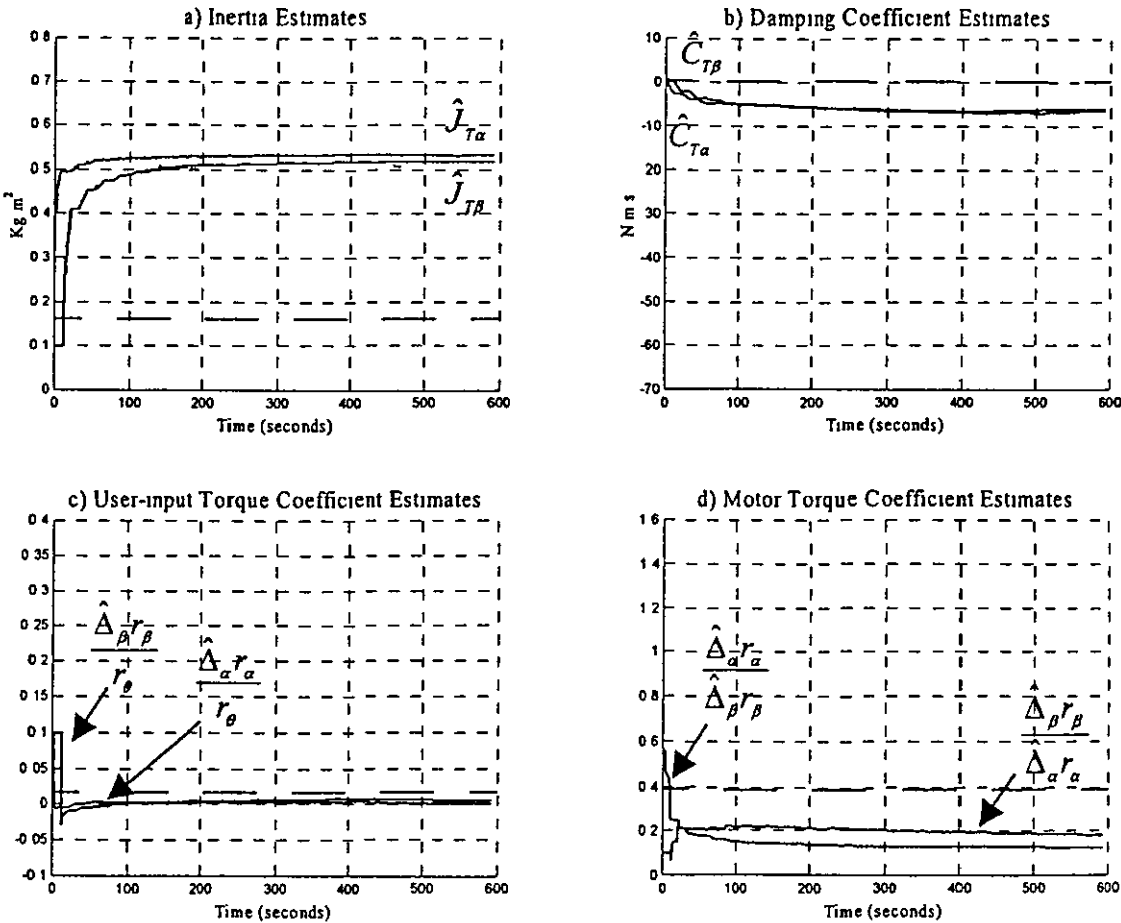


Fig. 13 Experimental parameter estimates with user-input 2

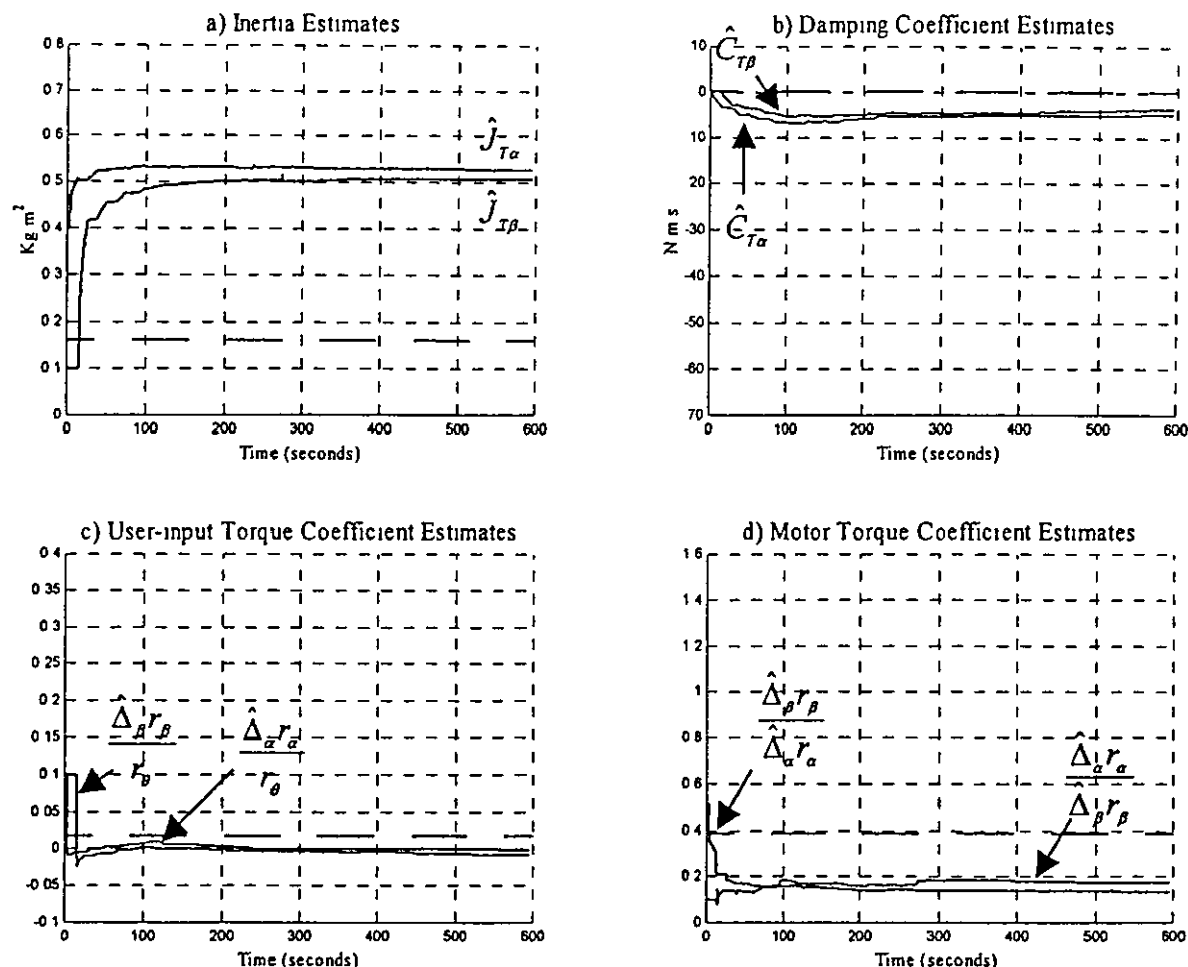


Fig 14 Experimental parameter estimates with user-input 3

Fig 9a). Similarly, Figs 11b and c illustrate the comparison for the user-inputs given by Figs 9b and c respectively. Computed-torque control was performed with user-inputs similar to those given in Fig. 9. The torque command could not be reproduced accurately owing to the input being applied manually. For clarity, the computed-torque controller error is shown offset by 0.005 rad. It can be seen that in the cases of Figs 11b and c there is not a significant difference in the performance of the composite adaptive controller compared to that of the computed-torque method.

The parameter estimates corresponding to the results of the user-input of Figs 9a, b and c are given in Figs 12, 13 and 14 respectively. Unlike the simulation, the parameter errors converge on different values to those calculated in equation (48) and shown by a dotted line in each figure. In the case of the low-magnitude user-input (Fig 12) the parameters are slow to reach a constant estimate. However, for the two larger magnitude user-input commands, Figs 13 and 14, the parameter errors are shown to reach steady state more rapidly.

The simulation and experimental results are different. It has not been the aim of this work to fully simulate the whole system accurately, including the digital

interface, but simply to use the simulation to develop the control strategy, which is then implemented practically. For example, the friction cannot be modelled precisely and the simulation model does not include flexibility in the joint mechanism and assumes perfect measurement of motor torque and speed. Despite these issues the composite adaptive controller is shown to work well and is robust against both modelling and measurement errors.

8 CONCLUSIONS

Both simulation and experimental results have shown that a composite adaptive control scheme can be used to track a desired path. It has been shown that the dependence of the desired velocity on the user-input force and error in the model has a detrimental effect on the convergence of the tracking and parameter errors to zero. The composite adaptive control method has been shown to give a performance equivalent to the computed-torque algorithm [5] under certain conditions on the user-input torque command. It has been shown experimentally that if the desired velocity and user-input are not sufficiently

independent of each other, the composite adaptive controller does not perform as well as the computed-torque algorithm. However, in both the simulation and experimental results, the tracking and parameter errors remain bounded under these conditions. The dual-worm mechanism is still able to cancel backlash using the composite adaptive control scheme.

Unlike the composite adaptive controller, the ability of the computed-torque control algorithm to track a path relies on fixed estimates of the joint parameters. Therefore, degradation of the computed-torque controller's performance due to wear of the gears or changes in load can be expected. This would require periodic re-tuning of the algorithm in order to ensure satisfactory performance, which can be a time consuming process. The composite adaptive control scheme has been shown to be robust in the face of large modelling and measurement errors, and as such can be used to compensate for changes in system parameters. Due to the nature of the proposed application, it may not be prudent to allow the controller to adjust parameters during operation. The composite adaptive controller should be used to train the manipulator off-line as part of a calibration procedure before it is used. Care must be taken to ensure that the user-input command consistently saturates the desired velocity function during this training period.

REFERENCES

- 1 Davies, B. L. A discussion of safety issues for medical robots. In *Computer-Integrated Surgery* (Eds R. H. Taylor, S. Lavallée, G. C. Burdea and R. S. Mösges), 1996, pp. 287–296 (MIT Press, Cambridge, Massachusetts).
- 2 Troccaz, J. and Delnondedieu, Y. Semi-active guiding systems in surgery. A two degree-of-freedom prototype of the passive arm with dynamic constraints (PADyC). *Mechatronics*, 1996, 6(4), 399–421.
- 3 Moore, C. A., Peshkin, M. A. and Colgate, J. E. A three revolute Cobot using CVTs in parallel. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE)*, Tennessee, 14–19 November 1999.
- 4 Harris, S. J., Lin, W. J., Fan, K. L., Hibberd, R. D., Cobb, J., Middleton, R. and Davies, B. L. Experiences with robotic systems for knee surgery. *Lecture Notes in Computer Sci.*, 1997, 1205, 757–766.
- 5 Reedman, A. V. C. and Bouazza-Marouf, K. Control of an actively constrained robotic joint for passive deployment applications. *Proc Instn Mech Engrs, Part K, Journal of Multi-body Dynamics*, 2001, 215(K4), 187–197.
- 6 Patton, W. J. *Mechanical Power Transmission*, 1980 (Prentice-Hall, Englewood Cliffs, New Jersey).
- 7 Bo, L. C. and Pavelescu, D. The friction-speed relationship and its influence on the critical velocity of stick-slip motion. *Wear*, 1982, 82, 277–289.
- 8 Craig, J. J., Hsu, P. and Sastry, S. S. Adaptive control of mechanical manipulators. *Int. J. Robotics Res.*, 1987, 6(2), 16–28.
- 9 Slotine, J. J. and Li, W. Adaptive robot control: a new perspective. In *Proceedings of the IEEE 28th International Conference on Decision and Control*, Los Angeles, California, December 1987, pp. 192–198.
- 10 Li, W. and Slotine, J. J. Parameter estimation strategies for robotic applications. In *ASME Winter Annual Meeting—Dynamic Systems and Control Division*, 1987, pp. 213–218.
- 11 Marino, R. and Tomei, P. *Nonlinear Control Design. Geometric, Adaptive and Robust*, 1995, 360 pp (Prentice-Hall, Englewood Cliffs, New Jersey).
- 12 Yager, T. The QNX operating system. *Byte*, August 1990, 15(8), 281–283.

