

Educational hands-on testbed using Lego robot for learning guidance, navigation, and control^{*}

Seungkeun Kim, Hyondong Oh^{*} Amir Kolaman^{**}
Antonios Tsourdos, Brian White^{*} Hugo Guterman^{***}

^{*} *Autonomous Systems Group, Dept of Informatics and Systems Engineering, Cranfield University, Defence Academy of the UK, Swindon, SN6 8LA UK (e-mail: s.kim@cranfield.ac.uk, h.oh@cranfield.ac.uk, a.tsourdos@cranfield.ac.uk, b.a.white@cranfield.ac.uk)*

^{**} *Department of Electro Optical Engineering, Ben Gurion University of the NEGEV, P.O.B 653 84105 Beer Sheva, Israel, (e-mail: kolaman@bgu.ac.il)*

^{***} *Department of Electric Engineering, Ben Gurion University of the NEGEV, P.O.B 653 84105 Beer Sheva, Israel, (e-mail: hugo@ee.bgu.ac.il)*

Abstract: The aim of this paper is to propose an educational hands-on testbed using inexpensive systems composed of a Lego Mindstorms NXT robot and a webcam and easy-to-deal-with tools especially for learning and testing guidance, navigation, and control as well as search and obstacle mapping, however the extendibility and applicability of the proposed approach is not limited to only the educational purpose. In order to provide navigation information of the Lego robot in an indoor environment, an vision navigation system is proposed based on a colour marker detection robust to brightness change and an Extended Kalman filter. Furthermore, a spiral-like search, a command-to-line-of-sight guidance, a motor control, and two-dimensional Splinegon approximation are applied to sensing and mapping of a complex-shaped obstacle. The experimental result shows that the proposed testbed can be viewed as an efficient tool for the education of image processing and estimation as well as guidance, navigation, and control with a minimum burden of time and cost.

Keywords: Education, guidance, navigation, and control, obstacle mapping, colour marker detection, Splinegon, Lego Mindstorms NXT.

1. INTRODUCTION

Recently Lego Mindstorms NXT which is one of the commercial robot kits has been used for testing the robotics applications [Worrall and Mcgookin (2006); Oliveira et al. (2009)] and more actively implemented as the educational tool for teaching robotics and control engineering: to name a few, Gawthrop and McGookin (2004); Carusi et al. (2004); A. Behrens (2008); Grega and Piat (2008). Worrall and Mcgookin (2006) derived the dynamics and kinematics of the differential-drive type Lego robot and compared the simulation result with the real experiment case for waypoint navigation. Gawthrop and McGookin (2004) demonstrated the educational laboratory experiments using Lego systems in the field of continuous-time control systems, and Carusi et al. (2004) proposed the distance learning facility for robotics and automation by which students can remotely control the Lego mobile robots. A. Behrens (2008) presented a practically motivated course development based on controlling the Lego robots under MATLAB environment. More recently Oliveira et al. (2009) presented a SLAM(simultaneous localisation and mapping) of an indoor environment using the Lego robot with a ultrasonic sensor and a compass sensor.

The aim of this paper is to propose an educational hands-on testbed using Lego Mindstorms NXT robot using inexpensive systems (a Lego robot and a webcam) and easy-to-deal-with tools especially for learning and testing GNC(Guidance, Navigation, and Control) as well as search and obstacle mapping, however the extendibility and applicability of the proposed approach is not limited to only the educational purpose. In order to provide navigation information of the Lego robot in an indoor environment, an vision navigation system using a low-cost webcam is implemented. To make the colour marker detection algorithm more robust to brightness change, a novel modified thresholding technique are proposed using the geometrical relationship of RGB(Red Green Blue) space in this study. The position and attitude of the robot can be estimated by detecting colour markers composed of red and blue balls attached on the robot, and the EKF(Extended Kalman filter) enhances its estimation accuracy. Furthermore, a spiral-like search, a CLOS(Command to Line Of Sight) guidance, a motor control, and two-dimensional Splinegon approximation are applied to sensing and mapping of a complex-shaped obstacle. To be easily handled with by even beginners, RWTH Mindstorms NXT Toolbox for MATLAB [A. Behrens (2008); RWTH-Aachen-Univ (2010)] was selected as a programming language for the Lego robot control. The experimental result shows that the proposed testbed can be viewed as an efficient tool for the education of image processing and estimation as well as guidance, navigation, and control with a minimum burden of time and cost.

^{*} A part of this study was successfully used as a student hands-on project for Working Group 2, 'Mapping of Unknown Obstacle by Lego Mindstorms NXT', at the NATO Advanced Study Institute: Advanced All-Terrain Autonomous Systems held in Cesme, Turkey on 15-24 Aug 2010.

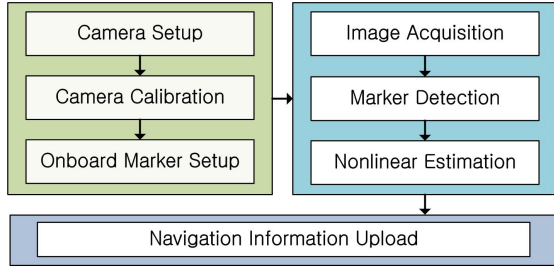


Fig. 1. A block diagram of the indoor vision navigation system

The overall structure of this paper is organised as: Section 2 describes an indoor navigation using colour marker detection and pose estimation filter design based on the EKF. Section 3 deals with the search, guidance, and control strategies for sensing and mapping of obstacle and also gives the obstacle approximation algorithm using the Splinegon. Section 4 contains an experimental setup and looks at the results and discussions on the experiment to evaluate the performance of the proposed approach. Lastly, conclusions and future works are given in Section 5.

2. NAVIGATION VIA COLOUR MARKER DETECTION

In the GPS-denied environments such as an indoor area, it is difficult to obtain accurate navigation information on a mobile robot. Although onboard sensors like a INS(Inertial Navigation System) can be used as an active sensor of the robot, it is costly and the error might grow fast and sometimes diverge during the integration of noisy accelerations. An alternative solution can be an indoor vision navigation system to be able to provide position, velocity, heading angle and its rate of the mobile robot. The accuracy of the vision system depends largely on camera performance, and generally to get the navigation information of target object the stereo images using multiple cameras are required for triangulation [Hartly and A. (2003)]. However, in order to reduce the cost and complexity of the system, only a low-cost webcam and two colour markers attached on the robot are used by applying a nonlinear estimation method, EKF, as a compensation for the need of multiple cameras. The EKF makes it possible to get the reasonably accurate navigation information even with a single camera by optimally tuning the information between the uncertain vehicle dynamics and the camera measurements. If required, accuracy and robustness of this vision system could be significantly improved by increasing the number of cameras or using a high quality camera. [Oh et al. (2011); Valenti et al. (2007)]. Figure 1 shows an operation procedure of indoor vision navigation system proposed in this paper, and the details of the colour marker detection and nonlinear estimation algorithm will be successively explained in the subsequent sections.

2.1 Colour marker detection robust to brightness

The precise estimation of the navigation states of the robot highly depends on the effective detection of the colour markers. It can be divided into two main stages: the colour detection and its segmentation. The specific colour is detected by using threshold parameters in a colour space and they can be represented as a colour detection map. This is then fed into the segmentation stage providing the coordinates of the center point of the marker having the largest shape detected in the map.

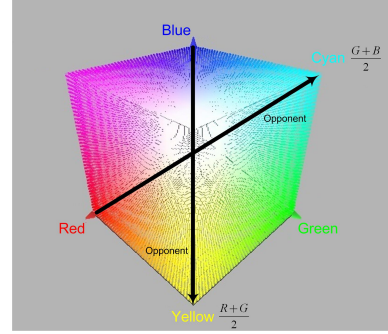


Fig. 2. 3D colour space and graphical description of opponent colour

The previous works detected the colour by thresholding in a RGB colour space [Oh et al. (2011)] or in a normalized RGB colour space [Zhang et al. (2009)]. They might have difficulties in coping with change of illumination and light, such as in Sampe et al. (2009) and Shah et al. (2010). To overcome the limitation of the aforementioned methods, this study proposes a different set of thresholding parameters which can be robust to most of brightness situations such as daylight, florescent light, tungsten light and shadows. The proposed robust colour detection is based on an adapted opponent colour space. A human visual system is known to be best at distinguishing colours under any type of brightness by comparing opponent colours. Using this hint, this study adopts two opponent colours: cyan versus red and yellow versus blue for detecting the red and blue colour markers. As shown in Fig. 2, each colour pixel can be expressed as a vector composed of three colour values: red(R), green(G) and blue(B) and then can be viewed as a point in a 3D euclidean space having R, G, and B axes. In other words, each colour pixel of the image captured by the webcam can be mapped into the following scalar functions representing the strength of blue and red colours.

$$f_B(\vec{v}_i) = B_i - \left(\frac{R_i + G_i}{2} \right) \quad (1a)$$

$$f_R(\vec{v}_i) = R_i - \left(\frac{G_i + B_i}{2} \right) \quad (1b)$$

where $\{\vec{v}_i = (R_i, G_i, B_i) \in \mathfrak{R}^3\}$, and $R_i, G_i,$ and B_i are the RGB values of the i -th pixel. The large values of f_B and f_R of the pixels in the RGB space represent they have the higher probability to be the desired colour. Eq. (1) is used to build the detection map for the blue and red colour markers. The colour detection maps are created by selecting a group of pixels having large values of f_B and f_R as:

$$C_{colour} = \{\vec{v}_i \mid (\text{the number of } j \text{ satisfying } f_{colour}(\vec{v}_i) < f_{colour}(\vec{v}_j) \text{ where } i \neq j) < \alpha_{colour}\}. \quad (2)$$

where $colour$ represents red or blue, and this study applied $\alpha_{colour} = 50$. In other words, the closest α number of pixels to the desired colour are selected. Since in this way Eq. (1) remains the same regardless of illumination and light changes, robustness of the colour marker detection can be significantly improved as shown in Fig. 3. The generated colour detection map is then fed into the segmentation process, which then finds the coordinate of the colour marker after finding the center of the largest shape in the map. A noise is removed by smoothing with a blur kernel followed by morphology. To reduce both processing time and vulnerability to adverse environment, a recursive target tracking method is introduced. Once the marker

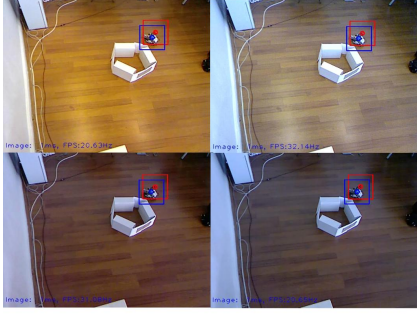


Fig. 3. Robust colour detection results with different brightness is detected, the search is performed within the region of interest, which is marked as blue and red squares around the marker center point as shown in Fig. 3.

2.2 Nonlinear pose estimation

Process model To describe the dynamics of the robot, let us consider a simple three degrees-of-freedom nonlinear equation. Its state variables are composed of as:

$$\mathbf{X} = [x_v \ y_v \ u \ v \ \psi_r \ r]^T. \quad (3)$$

where (x_v, y_v) , (u, v) , (ψ) , and (r) are a position, a velocity, a heading angle, and a body angular rate, respectively. In this study, the uncertainty from external forces and driving torques to the robot is considered as a zero-mean white Gaussian noise w . Then, the dynamic model of the vehicle for pose estimation can be simplified as:

$$\dot{\mathbf{X}} = f(\mathbf{X}, w) = \begin{pmatrix} u \cos(\psi_r) - v \sin(\psi_r) \\ u \sin(\psi_r) + v \cos(\psi_r) \\ rv + w \\ -ru + w \\ r \\ w \end{pmatrix}. \quad (4)$$

Measurement model The coordinates of the vision navigation system and the robot is represented in Fig. 4. This study considers the basic pinhole camera model used for a CCD (Charge Coupled Device) to describe a mapping between the 3D world and the 2D image. The basic pinhole camera model is expressed as [Hartly and A. (2003)]:

$$\bar{\mathbf{x}}_{image} = P\bar{\mathbf{X}}_{world} \quad (5)$$

where $\bar{\mathbf{X}}_{world}$ is the 3D world point represented by a homogeneous four-element vector $(X, Y, Z, W)^T$, $\bar{\mathbf{x}}_{image}$ is the image point represented by a homogeneous vector $(x, y, w)^T$. W and w are the scale factors which represent depth information, and P is the 3 by 4 camera projection matrix. In the calibration procedure, the origin of the inertial frame for the 3D world point is set to be a certain interested point as shown in Fig. 4. The camera measurements are the 2D visual information in the image coordinates as:

$$\mathbf{z} = [z_1 \ z_2] \quad (6)$$

$$z_i = [x_i^{cam} \ y_i^{cam}]^T, \ i \in (1, 2) : marker.$$

The measurement model can then be expressed as the nonlinear equation using the rotation transformation matrix and the camera projection matrix. First of all, the positions of two colour markers with respect to the inertial frame are determined by using the position (x_v, y_v) and the heading angle (ψ_r) of the vehicle and the predefined relative positions of the markers as:

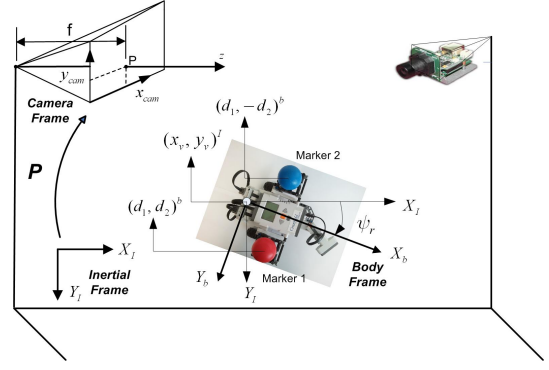


Fig. 4. Geometry among robot, camera, and environment

$$\bar{\mathbf{X}}_{mk,i}^I = [x_v \ y_v]^T + R_b^I \bar{\mathbf{X}}_{mk,i}^b \quad i \in (1, 2) : marker \quad (7)$$

$$\bar{\mathbf{X}}_{mk,i}^b = [d_1 \ \pm d_2]^T \quad (8)$$

where $\bar{\mathbf{X}}_{mk,i}$ represents the 2D position of i -th marker and I denotes an inertial frame, b denotes a body frame, and R_b^I is rotation transformation matrix from body to inertial frame. d_1 and d_2 represent the distance between the markers' position and the centre of the vehicle. Then the positions of two markers in the reference frame can be transformed into 2D visual information in the image coordinates by using the camera projection matrix as:

$$\mathbf{z} = [z_1 \ z_2] \quad (9)$$

$$z_i = \begin{bmatrix} P_1 \bar{\mathbf{X}}_{mk,i} & P_2 \bar{\mathbf{X}}_{mk,i} \\ P_3 \bar{\mathbf{X}}_{mk,i} & P_3 \bar{\mathbf{X}}_{mk,i} \end{bmatrix}^T, \quad i \in (1, 2) : marker$$

where P_j is the j -th row of the camera projection matrix P , and $\bar{\mathbf{X}}_{mk,i} = [\bar{\mathbf{X}}_{mk,i}^I \ 0 \ 1]^T$. The final measurement equation is obtained by incorporating the measurement noise into Eq. (9) as:

$$\mathbf{z}_k = h_k(\mathbf{X}_k) + v_k, \quad v_k \sim N(0, R_k). \quad (10)$$

Although the measurement noise can be incurred by various sources such as a calibration error, a CCD noise, a marker detection error, and a time delay, in this study it is assumed as a zero-mean white Gaussian noise with the covariance matrix R [Oh et al. (2011)].

Extended Kalman filter The extended Kalman filter (EKF) is used to estimate the state variables of the robot. The EKF is a widely-used filtering method in tracking and control problems, which linearises the nonlinear process and measurement models and applies the traditional Kalman filter. The details of the EKF algorithm can be found in [Simon (2006)]. By using the EKF, the navigation information of the Lego robot can be estimated even with a single camera. Furthermore, with a slight modification of Eqs. (6)-(9), the proposed navigation system can be easily extended to a 3D pose estimation and can also adopt multiple cameras to improve performance and robustness.

3. SENSING AND MAPPING OF OBSTACLE

3.1 Obstacle sensing: search, approach, and surveillance

Before going forward, the position information (x_v, y_v) of the vehicle obtained from estimation algorithm is converted to (x_r, y_r) with respect to the rotation centre of the ultrasonic sensor as shown in Fig. 5:

$$x_r = x_v + l_{vu} \cos \psi_r \quad (11)$$

$$y_r = y_v + l_{vu} \sin \psi_r \quad (12)$$

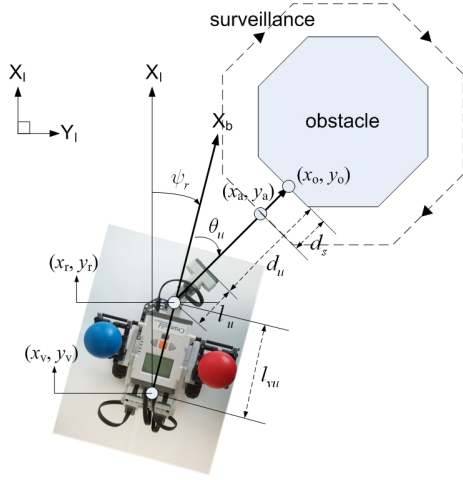


Fig. 5. Geometry among robot, obstacle, and environment

Search To find an obstacle inside a certain area interested, an effective search algorithm is required. This study selects a heuristic method: spiral-like search in which from the boundary of area interested, the robot searches spirally into its centre. This is effective in terms of search time since the Lego robot is equipped with the ultrasonic sensor rotating at the front side with -90 to 90 deg variation with respect to the body axis X_b . To apply the spiral-like search, the sensing range l_s shown in Fig. 6(a) must be carefully decided by considering the characteristics of ultrasonic sensor. The ultrasonic sensor module provided by the Lego Mindstorms NXT kit measures the TOF (Time Of Flight) to the obstacle surface and then converts it to a distance using the relation $d = ct/2$ where c and t are a ultrasonic speed and the TOF, respectively. However, this TOF measurement has angular ambiguity since the ultrasonic reflections are returned from any point within an area of emission cone. Especially, when the angle the ultrasonic sensor is facing towards the surface is askew, the max detection range becomes not more than 50cm rather than 255cm which is a default sensing limit. For this reason 40cm is selected as a sensing range of the ultrasonic sensor for the spiral-like search. As another challenge for students, circle packing [Lazarus (2009); Guo and Qu (2004)] can be also used for the search algorithm, which involves finding minimum number of circles to fully cover the area interested.

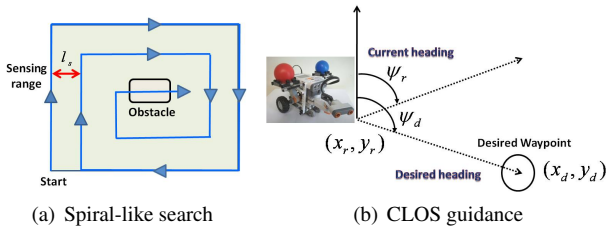


Fig. 6. Conceptual diagrams of sensing strategy

Measuring mapping points on obstacle surface Once the obstacle surface is detected by the ultrasonic sensor, the position vector of obstacles, (x_o, y_o) , can be decided as shown in Fig. 5:

$$x_o = x_r + (d_u + l_u) \cos(\psi_r + \theta_u) \quad (13)$$

$$y_o = y_r + (d_u + l_u) \sin(\psi_r + \theta_u) \quad (14)$$

where d_u is a distance to the obstacle sensed by the ultrasonic sensor, l_u is a length of rotating arm for the ultrasonic sensor, θ_u is a rotation angle of the ultrasonic sensor, and (x_r, y_r) and ψ_r are a position and a heading angle of Lego robot, respectively. Now the robot approaches to (x_a, y_a) having a distance d_s from (x_o, y_o) as shown in Fig. 5.

Guidance for waypoint following For searching for and approaching to an obstacle, a guidance loop for waypoint following is required. This study adopts a CLOS (Command to Line of Sight) guidance in which the command to the controller is generated using the difference between the desired heading and the current heading. The desired heading ψ_d could be simply computed using the current position of the Lego robot (x_r, y_r) and the desired target waypoint location (x_d, y_d) as:

$$\psi_d = \tan^{-1} \frac{x_d - x_r}{y_d - y_r} \quad (15)$$

The difference between the desired and the current heading of the Lego robot $e_\psi = \psi_d - \psi_r$ as shown in Fig. 6(b) can be fed back to the motor controller as a guidance command as will be described later in this section. As another choice, students can challenge other guidance techniques such as pure pursuit, cross-track following, or exact path design and following. The detailed classification on the path guidance can be found in Shima and Rasmussen (2010).

Guidance for surveillance around obstacle After succeeding the search and approach to (x_a, y_a) , the robot carries out the surveillance around the obstacle as shown in Fig. 5 in order to catch its mapping points. Unlike the waypoint following, the difference between the desired distance d_s and current ultrasonic distance to the obstacle, $e_d = d_s - d_u$ is fed back to the motor controller. During the surveillance around the mapping points on the obstacle surface can be periodically cumulated using Eqs. (13)-(14) as:

$$\begin{pmatrix} X_o \\ Y_o \end{pmatrix} = \begin{pmatrix} x_o^1 & x_o^2 & \dots & x_o^n \\ y_o^1 & y_o^2 & \dots & y_o^n \end{pmatrix} \quad (16)$$

where n is the total number of mapping points measured.

Motor control using PID control Now let us move on to the motor control for waypoint following and obstacle surveillance. A theoretical dynamics of differential-drive Lego robot is represented as:

$$\dot{x}_r = \frac{v_l + v_r}{2} \cos \psi_r \quad (17)$$

$$\dot{y}_r = \frac{v_l + v_r}{2} \sin \psi_r \quad (18)$$

$$\dot{\psi}_r = \frac{v_r - v_l}{2} \quad (19)$$

where (v_l, v_r) is the linear speeds of left and right wheels. Actually, a system identification is required for accurate control because the above model has unmodelled dynamics and nonlinearities due to frictions depending on the surface characteristics and so on. As a practical way, the PID controller with respect to a trim velocity v_0 can be used for the motor control. The left and right wheel speeds (v_l, v_r) in Eqs. (17)-(19) can be designed as:

$$v_l = v_0 + \delta \quad (20)$$

$$v_r = v_0 - \delta \quad (21)$$

$$\delta = k_P e + k_I \int e + k_D \dot{e} \quad (22)$$

where (k_P, k_I, k_D) are proportional, integral, and differential gains, respectively and e is a feedback error term. The gain scheduling has to be done depending on the mode change: $e = e_\psi$ for the waypoint following and $e = e_d$ for the obstacle surveillance. Students should be cautious of selecting the trim velocity v_0 and limiting on δ considering the characteristics of the motors and the friction between the wheel and floor. As an additional challenge for students, parameter identification, feedback linearisation, and nonlinear control could be tested to the motor control of differential-drive robot.

3.2 Obstacle mapping: 2D Splinegon

As an easy mapping method, the measurement points defined in Eqs. (13), (14), and (16) can be regarded as vertices of a polygon with straight line segments. However, this is unsuitable for representing a complex or curved shape of obstacle. To tackle this, Splinegon approximation has been recently applied to mapping of complex-shaped objects [Sinha et al. (in press)], which produces a set of vertices to be connected by line segments having a constant curvature. It means the line segments have a C^2 contact at the vertices to share a common vertex and the same tangents at the vertices. It is impossible to meet the abovementioned constraints at the vertices with a single arc since it has only single degree of freedom: the arc curvature. One way to solve this is to use an intermediate vertex between the original vertices which results in two arc segments of different curvatures to be designed. Figure 7 shows an example of Splinegon arc segment with C^2 contact intermediate vertex. A vector sum from the start vertex v_s to the end vertex v_f must yield:

$$c\mathbf{t}_c = b_s\mathbf{t}_s + (b_s + b_f)\mathbf{t}_I + b_f\mathbf{t}_f \quad (23)$$

where \mathbf{t}_I is the tangent vector of the intermediate vertex, and (\mathbf{t}_c, c) , (\mathbf{t}_s, b_s) , and (\mathbf{t}_f, b_f) are the tangent vectors and the lengths of the chord, the start vertex and the finish vertex, respectively. The detailed construction principle can be found at Chap.5 of [Lazarus (2009)] including solutions for \mathbf{t}_I , its bounds, Splinegon shape parameterisation, vertex selection and placement.

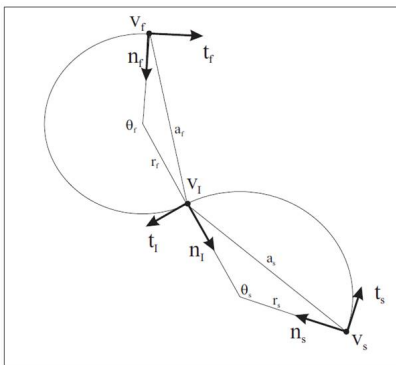


Fig. 7. Splinegon arc segment with C^2 contact intermediate vertex [Lazarus (2009)]

4. EXPERIMENTAL EVALUATION

4.1 Experimental setup

Figure 8 shows the overall block diagram of the experimental setup considered in this study, which requires only a Lego Mindstorms NXT kit, a webcam, and two laptops. The webcam (Logitech C500) gives the vision information for the Lego robot motion to a vision laptop via USB communication, in which the colour marker detection and EKF algorithms compute and deliver the navigation information of the Lego robot to a control laptop via TCP/IP communication. At the same time, the Lego robot collects the ultrasonic distance to the obstacle in the area interested and sends it to the control laptop. Then the control laptop decides the control command for the left and right motors of the Lego robot using the navigation information of the Lego robot and the measurements of the ultrasonic sensor and delivers it to the Lego robot via Bluetooth communication. The Lego robot is designed using the five NXT components shown in Fig. 9: a NXT brick, an ultrasonic sensor, and three interactive servo motor for rotating the left/right wheels and the ultrasonic sensor. An arbitrary obstacle is simply made using papers and boxes inside the area interested.

As a brain of the Lego robot, the NXT brick shown in Fig. 9(a) is based on 32-bit ARM7 microcontroller with 256K FLASH and 64K RAM and 8-bit AVR microcontroller with 4K FLASH and 512 Byte RAM. It is supported by USB 2.0 or Bluetooth communication and operated by a rechargeable lithium-polymer battery. Although there are many options for embedded programming of the NXT brick such as NXT-G, NXC, NBC, etc., RWTH - Mindstorms NXT Toolbox for MATLAB [RWTH-Aachen-Univ (2010)] is selected as an interactive and time-saving tool for students. By adopting this, a host computer can control the Lego robot via a direct communication between the NXT robot and the MATLAB main engine with the direct use of the MATLAB math functions and other toolboxes. Since debugging the code is easy, and low-level functions and direct commands are encapsulated, the selection of this toolbox is useful for the course development for beginner students.

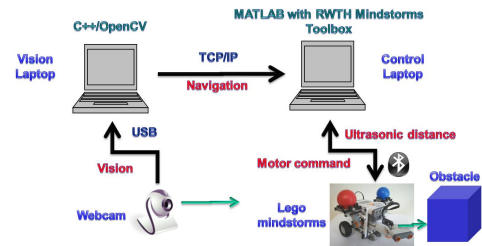


Fig. 8. A overall block diagram of experimental setup



Fig. 9. NXT components used in the testbed (thanks to <http://mindstorms.lego.com/>)

4.2 Results and discussion

The experiment was done in the small office room whose the size of the area interested is 0.9m-by-1.7m. At first, the Lego

robot starts the spiral-like search from the left-bottom corner of the area and successively moves to the left-top and right-top corners which are marked as red dots in Fig. 10(a). The CLOS guidance and motor control for the waypoint following works well, and the colour marker detection algorithm with the EKF gives the accurate navigation information of the Lego robot as displayed in the left-bottom corner of Fig. 10. At

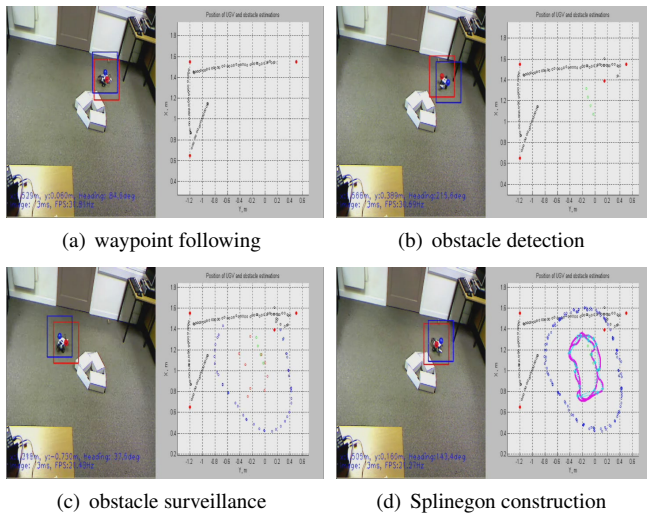


Fig. 10. Experimental result of testbed using Lego robot

the right-top corner the Lego robot detects the obstacle by rotating its ultrasonic sensor, and the measurement points on the obstacle are marked as green circles in Fig. 10(b). After detecting the obstacle, the Lego robot approaches to another red waypoint near the obstacle as shown in Fig. 10(b). Once arriving that red waypoint, the Lego robot starts the surveillance clockwise around the obstacle. Then it periodically detects and saves the detection points using the measurement of ultrasonic sensor as shown in Fig. 10(c) in which blue and red circles represent the positions of the Lego robot and the detection points, respectively. The motor controller keeping a certain distance from the obstacle works appropriately for the obstacle surveillance. After circling around the obstacle by 360 degrees, the Splinegon is drawn as a cyan line using the detection points stored during the obstacle surveillance as shown in Fig. 10(d). Although the resulting shapes of the Splinegons drawn during two times of circling is not exactly identical to that of the original obstacle, the overall tendency of the obstacle surface shape is reasonably reflected. Through this experiment using a webcam and a Lego Mindstorms NXT robot, students can learn a concept and design process for guidance, navigation, and control as well as mapping and vision processing.

5. CONCLUSIONS AND FUTURE WORKS

This study proposed an effective educational hands-on testbed so that students can learn a concept and design process for guidance, navigation, and control as well as mapping and vision processing. The proposed testbed is also fascinating in view of cost and simplicity since it is composed of a webcam, a Lego Mindstorms NXT robot, and two laptops only. Its extensibility is another merit since it can include more sophisticated guidance, navigation, and control techniques. As a future work, the error analysis and correction of the ultrasonic sensor should be done, and this testbed will be used as a course work and

extended to more complex system using multiple Lego robots for verifying cooperative control and estimation strategies.

REFERENCES

- A. Behrens, L. Atorf, R.S.J.B.T.H.A.T. (2008). First steps into practical engineering for freshman students using matlab and lego mindstorms robots. *Acta Polytechnica*, 48(3), 44–49.
- Carusi, F., Casini, M., Prattichizzo, D., and Vicino, A. (2004). Distance learning in robotics and automation by remote control of lego mobile robots. *IEEE International Conference on Robotics & Automation, New Orleans, LA*.
- Gawthrop, P. and McGookin, E. (2004). A lego-based control experiment. *IEEE Control Systems Magazine*, 43–56.
- Grega, W. and Piat, A. (2008). Real-time control teaching using lego mindstorms nxt robot. *International Multiconference on Computer Science and Information Technology, Wisla, Poland*.
- Guo, Y. and Qu, Z. (2004). Coverage control for a mobile robot patrolling a dynamic and uncertain environment. *The 5th world congress on intelligent control and automation, Hangzhou, China*.
- Hartly, R. and A., Z. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- Lazarus, S. (2009). Navigation and mapping using multiple autonomous vehicles. *Phd dissertation, Dept of Informatics and Sensors, Cranfield University*.
- Oh, H., Won, D., Huh, S., Shim, D., Tahk, M., and Tsourdos, A. (2011). Indoor UAV Control Using Multi-camera Visual Feedback. *Journal of Intelligent and Robotic Systems*, 61(1-4), 57–84.
- Oliveira, G., Silva, R., Lira, T., and Reis, L. (2009). Environment mapping using the lego mindstorms nxt and lejos nxj. *Fourteenth Portuguese Conference on Artificial Intelligence, Aveiro, Portugal*.
- RWTH-Aachen-Univ (2010). Rwth - mindstorms nxt toolbox for matlab. URL <http://www.mindstorms.rwth-aachen.de/>.
- Sampe, I., Amar Vijai, N., Tati Latifah, R., and Apriantono, T. (2009). A study on the effects of lightning and marker color variation to marker detection and tracking accuracy in gait analysis system. *ICICI-BME, Bandung, Indonesia*.
- Shah, S., Wu, A., and Johnson, E. (2010). Visual marker detection in the presence of colored noise for unmanned aerial vehicles. *AIAA Infotech, Atlanta, Georgia*.
- Shima, T. and Rasmussen, S. (2010). *UAV Cooperative Decision and Control: Challenges and Practical Approaches*. SIAM, Philadelphia.
- Simon, D. (2006). *Optimal State Estimation; H-inf and Non-linear Approaches*. John Wiley & Sons, Inc.
- Sinha, A., Tsourdos, A., and White, B. (in press). Multi uav coordination for tracking the dispersion of a contaminant cloud in an urban region. *European Journal of Control*.
- Valenti, M., Bethke, B., Dale, D., Frank, A., McGrew, J., Ahrens, S., How, J.P., and Vian, J. (2007). The mit indoor multi-vehicle flight testbed. *IEEE International Conference on Robotics and Automation, Rome, Italy*.
- Worrall, K. and McGookin, E. (2006). A mathematical model of a lego differential drive robot. *International Control Conference, Glasgow, Scotland*.
- Zhang, H., Yang, F., Wu, Y., and Painsavoine, M. (2009). Robust color circle-marker detection algorithm based on color information and Hough transformation. *Optical Engineering*, 48(10), 107202.