Using Hands-on Project with Lego Mindstorms in a Graduate Course

SEUNGKEUN KIM¹, HYONDONG OH², JIYOUNG CHOI³, and ANTONIOS TSOURDOS⁴
¹Department of Aerospace Engineering, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon 305-764, Korea. E-mail: skim78@cnu.ac.kr
²Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK. E-mail: h.oh@surrey.ac.uk
(Corresponding author)
³KSLV-II R&D Program Executive Office, Korea Aerospace Research Institute, 169-84 Gwahak-ro, Yuseong-gu, Daejeon 305-806, Korea. E-mail: jchoi@kari.re.kr
⁴Division of Engineering Science, Cranfield University, Cranfield, MK43 0AP, UK. E-mail: a.tsourdos@cranfield.ac.uk

Abstract
The aim of this paper is to propose an educational hands-on project using inexpensive subsystems for learning guidance and control. A Lego Mindstorms NXT, a low-cost web camera, and tractable tools are used for searching for and mapping of an obstacle in an indoor environment. In order to provide the Lego robot with navigation information indoors, visual tracking is implemented by using color marker detection and an extended Kalman filter. Furthermore, spiral-like search, command-to-line-of-sight guidance, and motor control are applied to sensing and mapping of an unknown obstacle. The experimental results from a short-term group project verify that the proposed hands-on work is an efficient educational tool for learning vision processing and estimation as well as guidance and control with a low-level burden of time and cost.

Keywords Educational hands-on project, guidance, control, visual tracking, obstacle mapping, color marker detection, Lego Mindstorms.

1 Introduction
Lego Mindstorms NXT is a commercial robotic toolkit for testing the robotics applications and has been widely used as an educational tool for teaching robotics and control engineering. In control engineering domain, Worral and McGookin [1] derived the dynamics and kinematics of the differential-drive type Lego robot and compared the simulation result with the real experimental case for waypoint tracking. Recently Oliveira et al. [2] presented a SLAM(simultaneous localization and mapping) of an indoor environment using the Lego robot with an ultrasonic and a compass sensor. For educational applications, Williams [3] firstly discussed the qualitative impact of using Lego Mindstorms robots to teach computer engineering. Gawthrop and McGookin [4] demonstrated the educational laboratory experiments using Lego systems in the field of continuous-time control systems, and Carusi et al. [5] proposed the distance learning facility for robotics and automation by which students can remotely control the Lego mobile robots. Leva [6] presented the experimental activity of undergraduate automatic control courses where several different control problems could be tested by means of a single apparatus. Vallim et al. [7] discussed a teaching experience obtained in the undergraduate course of Control Engineering at the Federal University of Santa Catarina (UFSC), Brazil to introduce the Lego Mindstorms NXT for the students to develop an understanding of basic engineering concepts and foster professional skills. Behrens et al. [8] presented a practically motivated course development based on controlling the Lego robots under MATLAB environment. Grega and Pilat [9] analyzed the possibility of real-time control teaching using the Lego Mindstorms NXT. Kim and Jeon [10] presented the course to introduce embedded systems using LEGO Mindstorms to freshmen under an ANSI-C programming environment. Stormont and Chen [11] discussed approaches to mechatronics education at Utah State University using inexpensive mobile robots including a Lego Mindstorm. Cejka et al. [12] adopted robotics for teaching math, science, technology and engineering in Kindergarten classroom using LEGO Mindstorms. Bishop et al. [13] proposed low-cost robotic laboratory exercises and projects by partially using the LEGO RCX. Valera et al. [14] used Lego Mindstorms for educating the control of mobile robots with mobile technologies. Recently, Behrens et al. [15] proposed an improved-version freshman introduction course established within the Bachelor of Science curriculum of Electrical Engineering and Information Technology of RWTH Aachen University, Germany. The course verified the merits of using the MATLAB environment for controlling the Lego Mindstorms NXT through an eight-day, full-time block laboratory for over 300 freshman students.

The aim of this paper is to propose a MSc (Master of Science)-level educational hands-on project using
inexpensive systems (a Lego Mindstorms NXT and a web camera). During the project, students can learn to design the basic guidance and control algorithms for obstacle mapping. In order to provide navigation information of the Lego in an indoor environment, a vision tracking system using a low-cost web camera is implemented. In this system, the position and heading of the robot are estimated by detecting color markers composed of red and blue balls attached on the robot using the geometrical relationship of RGB (Red Green Blue) space, and then an EKF (Extended Kalman filter) improves the accuracy of them using a robot kinematics. Furthermore, a spiral-like search, a CLOS (Command to Line Of Sight) guidance, and a motor control are applied to sensing and mapping of a complex-shaped obstacle. For even beginners to easily handle embedded programming, RWTH Mindstorms NXT Toolbox for MATLAB [8, 16] is selected as a programming language for the Lego robot control. The experimental result and the one-week hands-on project by graduate students at the NATO Advanced Study Institute [17, 18] showed that the proposed testbed could be a MSc-level project efficient enough to learn vision processing and estimation as well as guidance and control with a low burden of time and cost.

Since the previous attempts using the Lego Mindstorms in the literature have focused on devising an undergraduate level of learning, their target systems have been limited to simple or single-task applications. On the other hand, the proposed hands-on project highly motivates MSc-level students to apply guidance and control theories to a real system and can be further used for a group project of a semester MSc course because it has challenging missions such as vision processing, extended Kalman filtering, waypoint guidance, motor control, and search. Also, since this hands-on project is constructed with only a few equipments such as a Lego Mindstorms NXT, a web camera, and two laptops, the size and budget of the project group can be flexibly set up. Furthermore, the proposed approach is not limited to only the educational purpose, but extendible to robotic and control engineering applications by replacing the current algorithms by feature tracking, nonlinear filtering, and nonlinear guidance & control.

The overall structure of this paper is organized as follows. Section 2 defines an experimental setup, intended learning outcomes, and a workflow of the proposed project. Section 3.1 describes the first subgroup hands-on activity: visual tracking with a web camera by using color marker detection and EKF-based state estimation. Section 3.2 presents the second subgroup hands-on activity: sensing/mapping of an obstacle, including guidance and control strategies for surveillance against an obstacle. Section 4.1-4.2 introduce a case study and evaluation for applying the proposed hands-on project to NATO Advanced Study Institute in 2010. Then, Section 4.3 discusses the recommendations about how to use the proposed hands-on project for a semester graduate course and its assessment plan. Lastly, conclusions and future work are given in Section 5.

2 Learning targets and project structure
2.1 The testbed for hands-on project

Figure 1 shows the overall block diagram of the experimental setup considered in this study, which requires only a Lego Mindstorms NXT kit, a web camera, and two laptops. The web camera (Logitech C500) gives the vision information for the Lego robot motion to a vision laptop via USB communication, in which the color marker detection and EKF algorithm 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 measures the ultrasonic distance to the obstacle in the area of interest and sends the distance 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 in this study is composed of five NXT components shown in Fig. 1: a NXT brick, an ultrasonic sensor, and three interactive servo motors for spinning the left/right wheels and the ultrasonic sensor. An arbitrary obstacle is simply created with papers and boxes inside the area of interest.
As a brain of the Lego robot, the NXT brick shown in Fig. 1 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 [16] 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.

2.2 Intended learning outcomes
The ILOs (Intended Learning Outcomes) of this hands-on project are described: after the project, a MSc-level student will be able to

1) transfer mathematical methods for guidance and control to practical engineering tasks and understand their performances and limits in a deeper practical perspective;
2) use MATLAB and C++ languages for the implementation of guidance and control, visual tracking, and Kalman filtering on a computer and a Lego Mindstorms NXT;
3) create MATLAB and C++ programs for practical engineering applications beyond simulations in future challenges.
Figure 2 represents an overall structure of the proposed hands-on project. In the first phase of the project, teachers give a guidance presentation and take a Q&A session with students. Afterwards, every five or six students build a team, and then each team will be divided into two groups. After both the groups collaborate to assemble their own Lego Minstorms NXT robot, Group 1 and 2 work for vision tracking and robot control, respectively. Group 1 develops the algorithms for marker detection, tracking, and Kalman filtering using C++ programming, while Group 2 carries out motor control, waypoint guidance, and search/approach using MATLAB programming. Next to finishing the individual works, two groups work together for integration and experiments against an unknown obstacle.

It is preferable that a master-level student has the prerequisites of MATLAB/C++ programming, classical control, and estimation theory. However, the contents of the hands-on project can be adjusted depending on the background knowledge of target students. For instance, the vision tracking information of the Lego robot can be just provided to students if they do not have enough background to deal with vision processing and filtering. As an alternative, the theoretical fundamentals to be applied within the project can be taught by short lectures. Also, the proposed hands-on project would be accompanied by a semester MSc course about guidance and control, automatic control, or robotics, which will be discussed as a sample schedule in Section IV.

To accelerate the project progress, a two-hour MATLAB and C++ introductory exercise can be augmented with the project, which covers basic syntax, the manipulation of vectors and matrices, and communications through serial ports and Ethernet. Table 1 represents an example of timeline for the proposed seven-day hands-on project. The target timeline can be adjusted without affecting the intended learning outcomes depending on the circumstances of students and teachers.

Table 1. Project timeline

<table>
<thead>
<tr>
<th>Day</th>
<th>AM/PM</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>AM</td>
<td>Teachers give a presentation with Q&amp;A.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A block diagram of hands-on project structure (software/hardware)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i) Pinhole camera model, camera calibration, and color marker detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) Robot kinematics model and extend Kalman filtering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) Line-of-sight guidance, differential-drive robot dynamics, and PID control</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Students build teams and plan the project.</td>
</tr>
<tr>
<td>Day 2</td>
<td>AM</td>
<td>Students assemble their own Lego Mindstorms NXT robot.</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Two groups of students start to work on vision processing (Group 1) / Kalman filtering (Group 2).</td>
</tr>
<tr>
<td>Day 3</td>
<td>AM</td>
<td>Both the groups continue vision processing and Kalman filtering.</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Group 2 finishes the Kalman filtering.</td>
</tr>
<tr>
<td>Day 4</td>
<td>AM</td>
<td>Group 1 finishes vision processing and integrates the extended Kalman filtering from Group 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 2 starts the motor control for obstacle circling</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Group 2 finishes the motor control for obstacle circling.</td>
</tr>
<tr>
<td>Day 5</td>
<td>AM</td>
<td>Group 1 finishes integrating the extended Kalman filtering.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 2 starts waypoint guidance and control for the search for and the approach to an obstacle.</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Group 2 finishes the waypoint guidance and control.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 1 integrate the vision tracking information with the guidance and control.</td>
</tr>
<tr>
<td>Day 6</td>
<td>AM</td>
<td>Group 1/2 create an obstacle mapping algorithm.</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Group 1/2 finish overall integration.</td>
</tr>
<tr>
<td>Day 7</td>
<td>AM</td>
<td>Students perform experiments and algorithm modifications.</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>Students wrap up the experiment and give a short presentation.</td>
</tr>
</tbody>
</table>

3 Practical hands-on activities
3.1 Subgroup activity 1: Visual tracking with a web camera
3.1.1 Intended learning outcome of subgroup activity

In a GPS-denied environment such as an indoor area, it is difficult to obtain accurate navigation information on a mobile robot. Although onboard sensors like an INS (Inertial Navigation System) can be used as an active sensor for the robot movement, it is costly and the error might grow fast or even diverge during the integration of noisy
accelerations. An alternative solution can be an indoor visual tracking/positioning system which can provide motion information of the mobile robot. Through this subgroup activity, students can understand the necessity of indoor navigation for robots and apply color detection, triangulation, and estimation theories for visual tracking.

3.1.2 Introduction to visual tracking with a single camera

The accuracy of the visual tracking depends on an individual camera performance as well as the number of cameras, and stereo images from multiple cameras are typically used to obtain the navigation information of a target object by using triangulation [19, 20]. However, in order to reduce cost and complexity, this hands-on project proposes a visual tracking system using only a single low-cost web camera with implementing an EKF (Extended Kalman Filter). The EKF allows to obtain reasonably accurate navigation information with a single camera by optimally tuning the information between robot dynamics and camera measurements.

![Figure 3. A block diagram of the indoor visual tracking system](image)

The environment setup and operation for the visual tracking system is illustrated in Fig. 3. Before running the tracking algorithm, camera hardware and software setup is performed initially with camera calibration. Once visual tracking of onboard markers begins, the image captured by the camera is periodically transmitted into a vision computer. Core visual tracking algorithm consists of two steps: color marker detection and nonlinear estimation. Color markers in the image are detected and tracked by image processing algorithm, and then image coordinates of color markers centers are computed. Since the onboard color markers are attached at predefined positions on the robot, robot motion is estimated by nonlinear filtering using robot kinematics and geometric relations among an inertial frame, a robot body frame, and a camera frame. The details of the color marker detection and nonlinear estimation algorithm will be explained in the subsequent sections.

3.1.3 Color marker detection

The detection of the markers is done by extracting distinct colors in given images captured by the web camera using the RGB color-based algorithm. Firstly, the original image is decomposed into the RGB color space, and each pixel of the image is expressed as three color channels: red(R), green(G) and blue(B). The desired red and blue color of markers are then detected by using threshold parameters while generating a color detection map. The detection threshold is determined by testing various viewpoints and illumination conditions since the sensing of colors are varied depending on the lighting condition, shadow, and noise. Since there can be many noise blobs from objects having similar color as the desired color balls, noise removal by smoothing and morphology process is performed in the generated detection map. By selecting the largest blobs in the color map, color marker detection is completed, and the center positions of the selected blobs in the image coordinates are handed over to the nonlinear state estimation process as measurements for the EKF. Note that the entire detection process can be easily done by using openCV library, an open source library for image processing written in C++ [21, 22].

3.1.4 Nonlinear state estimation

Students can learn how to use Kalman filter for a tracking problem with only one web camera and how system dynamics is used for estimation through this hands-on activity.
3.1.4.1 Process model

To describe the dynamics of the robot, let us consider a simple three degree-of-freedom nonlinear equation. The state variables are represented as:

\[
X = [x_v \ y_v \ u \ v \ \psi_r \ \omega_r]^T
\]  

(1)

where \((x_v, y_v)\) and \((u, v)\) are position and velocity vectors in Cartesian coordinates, and \(\psi_r\) and \(\omega_r\) are a heading angle and a body angular rate, respectively. In this study, the uncertainty from external forces and driving torques to the robot is assumed to be a zero-mean white Gaussian noise \(w\) with a covariance matrix \(Q\). Then, the dynamic model of the robot for state estimation can be simplified as:

\[
\dot{X} = f(X, w) = \begin{pmatrix}
ucos(\psi_r) - vsin(\psi_r) \\
usin(\psi_r) + vcos(\psi_r)
\end{pmatrix}
\begin{pmatrix}
rv + w \\
r - ru + w
\end{pmatrix}
\]  

(2)

3.1.4.2 Measurement model

The geometry of the visual tracking 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 from 3D world to 2D image. The basic pinhole camera model is expressed as:

\[
\mathbf{x}_{image} = P\mathbf{x}_{world}
\]  

(3)

where \(\mathbf{x}_{world}\) is the 3D world point represented by a homogeneous four-element vector \((X, Y, Z, W_d)^T\), \(\mathbf{x}_{image}\) is the image point represented by a homogeneous vector \((x, y, w)^T\). \(W_d\) and \(w_d\) are the scale factors which represent depth information. \(P\) is the 3 by 4 homogeneous camera projection matrix with 11-degree of freedom [19]. The camera projection matrix is easily obtained using the camera calibration toolbox for MATLAB [23], and in this calibration procedure, the origin of the inertial frame for the 3D world point is set to be a certain point as shown in Fig. 4. The measurement model can then be expressed as the nonlinear equation by using the position \((x_v, y_v)\) and the heading angle \(\psi_r\) of the robot, the predefined relative positions of the markers, and the camera projection matrix as:

\[
z_k = h_k(P, X_k) + v_k, v_k \sim N(0, R_k)
\]  

(4)

where

\[
z_i = [\mathbf{z}_1 \ \mathbf{z}_2]^T, i \in \{1,2\}: \text{marker.}
\]  

Figure 4: Geometry among robot, camera, and environment
Although the measurement noise $v_k$ is incurred by various sources such as a calibration error, a CCD noise, a marker detection error, and a time delay, this study assumed it as a zero-mean white Gaussian noise with the covariance matrix $R_k$.

3.1.4.3 Extended Kalman filter (EKF)

As the system/measurement models are nonlinear, the EKF is used to estimate the state variables of the robot. The EKF is a widely-used filtering method in tracking and control problems. It linearizes the nonlinear process and measurement models and applies the same routine as a linear Kalman filter. The EKF algorithm consists of the prediction and update stages [24]. At the prediction stage, robot states are first predicted by a dynamic model with zero input and noise, and then it is updated by the measurement equation with measurements (i.e. color marker positions in the image coordinates) from the camera. By using the EKF, the position and heading of the Lego robot for guidance and control can be estimated even with a single web camera.

Even though other inertial navigation information (such as odometry, wheel speed, or accelerometer) can be fused with visual measurements to provide more accurate estimation, this visual tracking system is developed as a stand-alone indoor positioning system (like GPS for an indoor application) without dynamic connection with the robot. The fact that this system does not use any inertial measurements can reduce the cost as well as system complexity and implementation difficulties caused by communication delay or synchronization of measurements between a robot and a camera. Moreover, the developed vision system can provide the consistent estimation performance regardless of robot size or shape since it uses only the predefined static location of color balls in the body frame of the robot. Note that, with a slight modification, the proposed visual tracking system can be easily extended to a three-dimensional estimation and applied to multiple cameras with high resolution to improve performance and robustness as can be seen in [25].

3.2 Subgroup activity 2: Sensing and mapping of obstacle

3.2.1 Intended learning outcome of subgroup activity

The previous subgroup activity 1 gives a Lego robot position and heading information in a GPS-denied indoor environment. Now students are ready to guide the Lego robot move to a waypoint, search for an obstacle using an ultrasonic sensor, and map the obstacle. Through this subgroup activity, students can understand the necessity of guidance for waypoint tracking and motor control for steering and apply line-of-sight guidance, search strategies, and classical control.

3.2.2 Obstacle sensing: search, approach, and surveillance

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

$$x_r = x_v + l_{uv} \cos \psi_r$$
$$y_r = y_v + l_{uv} \sin \psi_r$$

Figure 5. Geometrical relation among robot, obstacle, and environment
3.2.2.1 Search
To find an obstacle inside a certain area of interest, an effective search algorithm is required. This study selects a heuristic method: spiral-like search in which from the boundary of area of interest, the robot searches spirally into its center. 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) needs to be carefully decided by considering the characteristics of ultrasonic sensor. As another challenge for students, circle packing [26, 27] can be also used for the search algorithm, which involves finding minimum number of circles to fully cover the area of interest.

3.2.2.2 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:

$$x_o = x_r + (d_u + l_u)\cos(\psi_r + \theta_u)$$  \hspace{1cm} (8)
$$y_o = y_r + (d_u + l_u)\sin(\psi_r + \theta_u)$$  \hspace{1cm} (9)

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, $\theta_u$ is a rotation angle of the ultrasonic sensor, and $(x_r, y_r)$ and $\psi_r$ are a position and a heading angle of the Lego robot, respectively. Now the robot approaches to $(x_i, y_i)$ having a distance $d_s$ from $(x_o, y_o)$ as shown in Fig. 5. Note that as this formulation gives just an example, students can modify it depending on their own selections of the location and rotation mechanism of the ultrasonic sensor on the robot.

3.2.2.3 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 $\psi_d$ 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:

$$\psi_d = \tan^{-1}(y_d-y_r)/(x_d-x_r).$$  \hspace{1cm} (10)

The difference between the desired and the current heading of the Lego robot $\epsilon_\psi = \psi_d - \psi_r$, 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 [27].

3.2.2.4 Guidance for surveillance around obstacle
After succeeding to approach to $(x_o, y_o)$, 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. This surveillance acquires a set of mapping points periodically sensed from the obstacle surface by the ultrasonic sensor using Eqs. (8)-(9). Students can adjust the frequency of ultrasonic sensing during the obstacle surveillance for changing a resolution of mapping.
3.2.2.5 Motor controller design 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:

\[
\begin{align*}
\dot{x}_r &= \frac{v_l + v_r}{2} \cos \psi_r \\
\dot{y}_r &= \frac{v_l + v_r}{2} \sin \psi_r \\
\dot{\psi}_r &= \frac{v_r - v_l}{2}
\end{align*}
\]

(11) (12) (13)

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 unmodeled 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. (11)-(13) can be designed as:

\[
\begin{align*}
v_l &= v_0 + \delta \\
v_r &= v_0 - \delta \\
\delta &= k_p e + k_i \int e + k_d \dot{e}
\end{align*}
\]

(14) (15) (16)

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: \(e = e_g\) for the waypoint following and \(e = e_d\) for the obstacle surveillance. Students can tune the magnitude of control gains to feel what characteristics of a system response is affected by proportional, integral, and differential gains and can also select the trim velocity \(v_0\) and limiting on \(\delta\) by taking into account the motor characteristics and the friction between the wheel and floor.

3.2.3 Obstacle mapping

As an easy mapping method, the measurement points defined in Eqs. (8)-(9) can be regarded as vertices of a polygon with straight line segments. However, as this might be unsuitable for representing a complex or curved shape of obstacle. To tackle this, students can apply Splinegon approximation for mapping of complex-shaped objects [29], which produces a set of vertices to be connected by line segments having a constant curvature.

4 A case study and evaluation

4.1 A case study applying the proposed hands-on project

A part of this study was successfully used as a student hands-on project at the NATO Advanced Study Institute: Advanced All-Terrain Autonomous Systems held in Cesme, Turkey on 15-24, Aug. 2010 [17]. This NATO Advanced Study Institute (ASI) was an international conference focused on the next generation of aerial (UAV), ground (UGV) and underwater (UUV) autonomous systems. Technical tracks featured applications for civilian and military use including but not limited to search-and-rescue, disaster mitigation, and human aid. In addition to technical sessions, all participants of the NATO ASI were expected to join in one of five working groups during the conference. Those working groups completed hands-on projects in topical areas related to Advanced All-Terrain Autonomous Systems throughout the ASI. They also presented their work in the last day of the conference. Therefore, participants were expected to attend the entire ASI in order to make a significant contribution to their group.

The proposed hands-on project was tested for Working Group 2: Mapping of Unknown Obstacle by Lego Mindstorms NXT. Totally six master-level students from the US, Italy, Turkey, Israel, and UK took part in the hands-on project for a week. They had different backgrounds from aerospace, mechanical, electrical, and computer engineering. Due to their limited knowledge of the required techniques: vision processing, control, estimation or guidance algorithm either only theoretically or experimentally, two teams were made by taking the students backgrounds into account, and an instructor and a research assistant gave an introductory presentation on the first day and continued to support students for the remaining days by following the recommended timeline discussed in Section II.B. Figure 7 shows a test environment and the Lego robots which students built for the project for themselves.
The experiment was done in the small area of a conference room whose size is about 2m-by-2m. Figure 8 illustrates a sample data of one of two teams: the position/heading information of the Lego robot captured from the web camera and the MATLAB figure. 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 dots in Fig. 8(a). The CLOS guidance and motor control work properly for waypoint tracking, and the color marker detection from visual tracking gives the accurate navigation information of the Lego robot as displayed in the left-bottom corner of Figs. 8(a)-(c).
At the left-top corner the Lego robot detects the obstacle by rotating its ultrasonic sensor, and the detected points on the obstacle are marked as circles in Fig. 8(a). After detecting the obstacle, the Lego robot approaches to another waypoint near the obstacle as shown in Fig. 8(a). Once arriving the waypoint, the Lego robot starts surveillance circling clockwise around the obstacle. Then it periodically detects and saves the detected points on the obstacle using the ultrasonic sensor as shown in Fig. 8(b) in which outer and inner circles represent the positions of the Lego robot and the detected points of the obstacle, respectively. The motor controller keeps the Lego robot to maintain a certain distance (30 cm) from the obstacle. After circling around the obstacle by 360 degrees, the obstacle map is drawn as a blue line by linking all the detected points as shown in Fig. 8(c). Although the resulting mapping is not exactly identical to that of the original obstacle, the overall accuracy of the obstacle surface shape is reasonable taking into account a low resolution of an inexpensive ultrasonic sensor. Through this experiment using a web camera and a Lego Mindstorms NXT robot, students could learn a concept and design process for guidance and control as well as visual tracking.

4.2 Assessment and lessons
4.2.1 Assessment and lessons from student feedback

All students responded favorably to the hands-on project using a Lego Mindstorms robot and a web camera to learn guidance and control as well as visual tracking. During the project, several professors and instructors in robotics and control domains who attended the ASI were also interested in the proposed hands-on project for applying it to their courses. Although some students could not entirely understand and implement all the techniques for a week, using part of materials given by an instructor, they could learn individual technical algorithms and perform search and obstacle mapping mission successfully. They experienced figuring out what kind of problems can occur in a real robotic and control system. Besides, they could contribute to project by integrating their own algorithms (e.g. search or vision process) to improve the performance. In fact, one student was able to succeed to improve vision tracking performance significantly against brightness change. Even though the number of students taking the hands-on project is small, and there was not an official questionnaire at the end of the project, six MSc-level students provided the instructor with valuable feedback. Examples of the lessons learned and comments from the students are as follows.

Six students agreed:
• As a whole it was a nice hands-on project to learn guidance and control as well as visual tracking in terms of conciseness and expense.
• Starting the project with assembling Lego Robots can motivate an active participation of students.
• The Lego robot helps to understand the components and characteristics of robotic system (hardware) and control theory (software) easily and efficiently with the help of RWTH Mindstorms NXT toolbox for MATLAB [12].

Some of students raised the following issues:
• Learning vision processing in a short time was difficult to achieve. But, by experiencing some of image processing procedures such as camera calibration, image acquisition or color detection using MATLAB toolbox [23] and openCV library [21, 22], student found it interesting and feasible to do if they had more time.
• The EKF (estimation theory) was difficult to understand and apply to a hands-on project in a few days without previous knowledge. However, after implementing the EKF with given dynamic and camera models, they were able to understand at least how it works and why it is required. In this project, the EKF was required to obtain a continuous heading angle of a Lego robot from color marker position measurements.
• Basic programming skills for MATLAB and C++ language or equivalent to these are essential.
• It is better to provide extra time to students to implement their own ideas at the end of the project to maximize the effect of learning. Only following the designed schedule given by a course director was not enough for high-level graduate students.

This hands-on project involves various technical areas such as robotics, programming, image processing, Kalman filtering, and guidance & control as addressed in Section III. Even if it is an important benefit of this hands-on project that students can experience an integration of those various technical algorithms, it may distract students from carrying out an original target for learning guidance & control. To positively resolve this issue, the following section will discuss how to use the proposed hands-on project for lab sessions of a semester graduate course related
to robotics and control.

4.3 Recommendations for a semester MSc course

Taking the issues raised by students in the case study into account, it is recommendable to apply the proposed hands-on project for a semester graduate course about guidance and control within an extended time schedule as shown in Table 2. This recommended course assumes that target MSc or early-stage PhD students do not have enough prerequisite knowledge on computer vision, Kalman filtering, and guidance & control of a differential-drive robot, and the length of a semester is 15 weeks even if it is variable depending on universities and terms. Six weeks, three weeks, and five weeks are devoted to learning visual tracking, Kalman filtering, and basic guidance & control techniques, respectively. In the last week, students give technical presentations, demonstrations and exhibitions. With this time frame, students in robotics and control domains will be able to learn basic visual tracking, estimation, and guidance/control techniques and to apply them to a real robotic problem.

The evaluation of the proposed hands-on project will be able to be done via an anonymous student online evaluation (A: excellent, B: good, C: average, D: below average, E: inadequate) with the following check points.

1) Previous C++/MATLAB programming skills before participating in the hands-on project
2) Previous vision processing skills before participating in the hands-on project
3) Previous guidance & control skills before participating in the hands-on project
4) Improved C++/MATLAB programming skills after the hands-on project
5) Improved vision processing skills and level of understanding after the hands-on project
6) Improved guidance & control skills and level of understanding after the hands-on project
7) Motivation level to apply theories learned from lectures to the hands-on project
8) Motivation level to team work
9) Overall rating of the hands-on project

As this evaluation gives pre/post assessment data, instructors can further develop or modify the proposed hands-on project for maximizing an educational effectiveness.

Table 2: An example timeline of a semester graduate course integrated with the proposed hands-on project

<table>
<thead>
<tr>
<th>Week</th>
<th>Lectures</th>
<th>Lab sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to visual tracking</td>
<td>A presentation and Q&amp;A about the hands-on testbed</td>
</tr>
<tr>
<td>2</td>
<td>Pinhole camera model</td>
<td>Team building and Lego robot assembly</td>
</tr>
<tr>
<td>3</td>
<td>Camera calibration</td>
<td>Experiencing a MATLAB calibration toolbox</td>
</tr>
<tr>
<td>4</td>
<td>Color marker detection 1: color detection</td>
<td>Introduction to basic C++ syntax and OpenCV library</td>
</tr>
<tr>
<td>5</td>
<td>Color marker detection 2: segmentation</td>
<td>Testing the examples of using OpenCV library</td>
</tr>
<tr>
<td>6</td>
<td>Color marker detection 3: ROI tracking</td>
<td>Using OpenCV library for color marker detection</td>
</tr>
<tr>
<td>7</td>
<td>Kalman filter 1: Robot/camera kinematics</td>
<td>Finishing color marker detection</td>
</tr>
<tr>
<td>8</td>
<td>Kalman filter 2: Observer &amp; Kalman filter Testing a MATLAB example of Kalman filtering</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Kalman filter 3: Extended Kalman filter</td>
<td>Developing a MATLAB routine of extended Kalman filtering</td>
</tr>
<tr>
<td>10</td>
<td>G&amp;C 1: Intro. to differential drive robots</td>
<td>Implementation of extended Kalman filtering with C++</td>
</tr>
<tr>
<td>11</td>
<td>G&amp;C 2: PID control</td>
<td>Intro. to RWTH Mindstorms NXT toolbox for MATLAB</td>
</tr>
<tr>
<td>12</td>
<td>G&amp;C 3: Ultrasonic sensing</td>
<td>Implementing motor control &amp; obstacle sensing routine</td>
</tr>
<tr>
<td>13</td>
<td>G&amp;C 4: Guidance &amp; path planning</td>
<td>Implementing a waypoint guidance routine</td>
</tr>
<tr>
<td>14</td>
<td>G&amp;C 5: Mapping techniques</td>
<td>Implementing a mapping routine and overall integration</td>
</tr>
<tr>
<td>15</td>
<td>Summary and student presentations</td>
<td>Demonstrations and exhibitions</td>
</tr>
</tbody>
</table>

5 Conclusions and future work

This study proposed an educational hands-on project so that students can learn a concept and design process for robotics, guidance and control as well as visual tracking. The proposed project is effective in view of cost and simplicity since it uses a web camera, a Lego Mindstorms NXT robot, and two laptops only. Its extendibility is another merit since it can include more sophisticated guidance and control techniques. The proposed study was successfully used as a student hands-on project at the NATO Advanced Study Institute: Advanced All-Terrain
Autonomous Systems in 2010, and the feedback from six MSc-level students agreed that as a whole it was a useful hands-on project to learn guidance and control as well as visual tracking in a single work. Based on these feedback and student presentations after the group work, the intended learning outcomes were satisfied: students could transfer mathematical methods for guidance and control to practical engineering tasks and understand their performances and limits, could learn and further use MATLAB and C++ languages for the implementation of guidance and control, visual tracking, and Kalman filtering on a computer and a Lego Mindstorms NXT, learned potential techniques in order to create MATLAB and C++ programs for practical engineering applications beyond simulations in future challenges. Taking the issues raised by students in the case study into account, a semester graduate course about guidance and control and its assessment plan were recommended to apply the proposed hands-on project. Even if the evaluation of this work is currently based on the experience by limited number of students in a group work in a NATO technical symposium, as a future work, this hands-on project will be used as lab sessions for a semester graduate course as discussed in Section 4.3 and extended to more complex system using multiple Lego robots for verifying cooperative control and estimation strategies.

Acknowledgments
This work was supported by research fund of Chungnam National University.

References


Seungkeun Kim received BSc and PhD degrees in Mechanical & Aerospace Engineering from Seoul Natl’ University, Seoul, Korea, in 2002 and 2008, respectively. He is currently an assistant professor at Dept of Aerospace Engineering, Chungnam Natl’ University, Daejeon, Korea. Previously he was a lecturer in Cooperative Control and Estimation in Cranfield University, UK between Dec 2010 and Feb 2012 and also worked as a Research Fellow at Cranfield University between 2008 and 2010. He has actively published book chapters, journal and conference papers related to unmanned systems. His research interests cover nonlinear guidance and control, estimation, sensor and information fusion, fault diagnosis, fault tolerant control, and decision making for unmanned systems.

Hyondong Oh received the BSc. and MSc. degrees in Aerospace Engineering from KAIST, South Korea, in 2004 and 2010, respectively. And then he acquired a PhD on Autonomous Surveillance and Target Tracking Guidance Using Multiple UAVs from Cranfield University, UK in 2013. He is currently a postdoctoral research fellow at Nature Inspired Computing and Engineering (NICE) Group, Department of Computing, University of Surrey, UK. His research interests include biologically-inspired swarming robotics, cooperative control and path planning of autonomous systems, target surveillance and tracking guidance, and information/sensor fusion.

Jiyoung Choi received a BSc. and MSc. degrees in Mechanical and Aerospace Engineering from Seoul Natl’ University, Seoul, Korea, in 2004 and 2006, respectively. And then she acquired a PhD from Cranfield University in 2012. She is currently a senior researcher at Korea Aerospace Research Institute, Daejeon, Korea. Previously she was a post-doc. researcher in Dept of Aerospace Engineering in KAIST, Korea between Mar 2012 and Dec 2012 and also worked as a researcher at Korea Aerospace Research Institute between Jul 2006 and Mar 2009. Her research interests cover fault diagnosis, fault tolerant control, verification of autonomous system and flight control system for launch vehicle.

Antonios Tsourdos obtained a MEng on Electronic, Control and Systems Engineering from the University of Sheffield (1995), an MSc on Systems Engineering from Cardiff University (1996) and a PhD on Nonlinear Robust Missile Autopilot Design and Analysis from Cranfield University (1999). He is a Professor of Control Engineering with Cranfield University. Appointed as the Head of the Centre for Cyber-Physical Systems in 2013. Professor Tsourdos was member of the Team Stellar, the winning team for the UK MoD Grand Challenge (2008) and the IET Innovation Award (Category Team, 2009).