Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT

Miguel Pinto, *Member, IEEE*, António Paulo Moreira, *Member, IEEE*, and Aníbal Matos, *Member, IEEE*

***Abstract*—The inspiration for this paper comes from a successful experiment conducted with students in the "Mobile Robots" course in the fifth year of the integrated Master's program in the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto (FEUP), Porto, Portugal. One of the topics in this Mobile Robots course is "Localization of Mobile Robots using the Extended Kalman Filter in a LEGO NXT," which gives the students the opportunity to study the concepts of localization. This experiment comes within the framework of teaching localization concepts in mobile robotics and focuses primarily on explaining the Kalman filter concept. It involves a specific tool developed by the authors and based on LEGO NXT technology. The work presented here could be a helpful guide for teaching concepts related to localization in mobile robotics to ensure adequate understanding of the concept and of the use of the extended Kalman filter (EKF). The LegoFeup robot described here was built using a LEGO Mindstorms NXT and tested both in simulation and in real scenarios. Based on the results obtained, the authors concluded that the developed tool is effective in motivating students. The implementation of the tool, the structure of the Mobile Robots course, and the criteria for student assessment are described in this paper.**

***Index Terms*—Robots, robot kinematics, robot programming, robot sensing systems, state estimation.**

I. Introduction

THE EXPERIMENT described here was carried out by undergraduate students from the Department of Electrical and Computers Engineering, Faculty of Engineering, University of Porto, Porto, Portugal, in the "Mobile Robots" course. The textbooks for this course are [1] and [2].

The Department of Electrical and Computer Engineering enrolls around 200 students per year. In the first two years of the Electrical and Computer Engineering program, the students learn the fundamental theoretical concepts of mathematics, physics, programming languages, and electronics. After the second year, students have to choose between three branches of the curriculum: automation, energy, or telecommunications. Within the automation branch, the students then have to choose a specialization. Students choosing to specialize in robotics will take a course on mobile robots, in which they will learn about the navigation and localization of mobile robots.

This paper is organized as follows. Section II describes the state of the art. Section III presents the experiment. Section IV describes the teaching method, while Section V describes the tool developed, LegoFeup loop, and its extended Kalman filter (EKF) algorithm. Section VI describes the communication between the algorithm and the simulation or real scenario. Section VII describes the user interface. Section VIII presents the organization of the theoretical and practical lessons. Finally, Section IX shows the results of the implemented method. Some conclusions are drawn in Section X.

A. Mobile Robots Course

The Mobile Robots course covers a wide range of subjects in the area of localization and navigation, such as relative and global localization, kinematics, and sensor fusion. The issues of probabilistic methods, as the particle and Kalman filter, are discussed in the lessons on sensor fusion.

The experiment described in this paper focuses mainly on the Kalman filter. This could be very useful for teachers of similar courses wanting to develop the theme of localization of mobile robots using an extended Kalman filter. The experiment can also be easily adapted; the same experiment could be performed using other localization methods such as the particle filter.

B. Student Learning

In terms of student learning, the main goals of the experiment are that the undergraduate students should: 1) realize that sensor fusion is important to relating environmental information to odometry; 2) understand the necessity of a probabilistic method such as the Kalman filter instead of using direct fusion between the environmental information and odometry; 3) realize how the Kalman filter can be applied to a practical problem within the mobile robots' area of operation; and 4) fully understand the operation of a Kalman filter, its cycle (prediction and update steps), and the importance of the covariance matrices and the respective tuning for the correct operation of the EKF.

II. Background

LEGO Mindstorms is widely used in education to teach programming languages, robotics, and embedded systems. Its use in teaching is a good way to motivate engineering students, which is fundamental to successful teaching.

For over 40 years, the most common approach in education was to teach pure mathematics and science without immediate application of that knowledge. This could be characterized as teaching based on descriptions of engineering

Manuscript received November 13, 2010; revised February 22, 2011; accepted May 02, 2011. Date of publication May 31, 2011; date of current version February 03, 2012. This work was supported in part by INESC Porto—Institute for Systems and Computer Engineering of Porto, Porto, Portugal.

The authors are with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Porto 4200-465, Portugal (e-mail: miguelp@fe.up.pt; amoreira@fe.up.pt; anibal@fe.up.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2011.2155066

applications instead of teaching based on practical engineering. Director *et al.* [3] believe that only when an engineering course begins to be seen as a whole, and not as a "constant republishing of aging courses," will it have a real impact. Therefore, it is important to ensure that freshman engineers have the ability to face real and complex problems.

A set of laboratory courses—Robotic Manipulation, Computer Vision, Artificial Intelligence, Mechatronics—and their respective laboratory experiments and exercises were presented in a 1996 seminal paper on robotics education [4]. These experiments do not cover mobile robotics.

In the "Statistical Techniques in Robotics" course given at the Robotics Institute of Carnegie Mellon University (CMU), Pittsburgh, PA, probabilistic techniques are taught as a fundamental element of a robot. In other courses such as "Introduction to Robotics," "Mobile Robot Programming Laboratory," or "Kinematics, Dynamic Systems and Control," undergraduate students learn kinematics, vision, motion planning, and collision avoidance. However, it is during the "Robotic Motion Planning" and "Introduction to Mobile Robots" courses that localization is discussed and sensor-based probabilistic techniques such as the Kalman filter are implemented [5].

At the Massachusetts Institute of Technology (MIT), Cambridge, the electrical engineering and computer science curriculum offers a "Robotics: Science and Systems I" course, where motion planning, kinematics, tracking, and state estimation are just some of the topics covered. These topics are then put into practice during the "Robotics: Science and Systems II" course [6].

At the School of Electrical Computer Engineering, Cornell University, Ithaca, NY, an "Introduction to Probability and Random Signals" course introduces the issue of state estimation. At the same school, in an "Autonomous Mobile Robots" course, topics such as sensing and localization are covered during laboratory lessons [7].

At European universities, robotics is also an area of interest. At Imperial College London, London, U.K., the Department of Computing offers a robotics course in the third year of the curriculum, in which students learn probabilistic methods, such as the Monte Carlo method and the Kalman filter, for localization in a known map when using odometers and sonar. The topics covered are then used in practical lessons, and the students work with LEGO Mindstorms NXT kits [8].

At the Swiss Federal Institute of Technology, ETH Zurich, Zurich, Switzerland, the Master's program in "Robotics, Systems and Control" offers courses such as "Stochastic Systems," where the Kalman filter and its application in finance and engineering are taught in theory lessons. In courses such as "Theory of Robotics & Mechatronics," "Autonomous Mobile Robots," and "Unmanned Aircraft Design, Modeling and Control," kinematics, sensor fusion, pose estimation, and localization based on probabilistic methods, such as Monte Carlo localization and the Kalman filter, are taught with application in real scenarios [9].

Since October 2010, the authors of this paper have been conducting an experiment in the context of mobile robotics education at the Faculty of Engineering of the University of Porto. The experiment described here can easily be adapted and used in the courses discussed above. Therefore, this paper can be used as a guide to complete a pedagogical experiment involving the concepts of sensor fusion, navigation, and probabilistic localization using an extended Kalman filter. The goal is to improve undergraduate teaching and help students put theory into practice.

A. Work With LEGO Mindstorms

As demonstrated in [10], social robots like LEGO Mindstorms can be used to increase young students' interest in robotics and in other areas such as electrical systems. In [11], Ruiz-del-Solar describes outreach activities using LEGO Mindstorms. He outlines some of the reasons why students lack interest in engineering education programs: 1) students do not understand "what an engineer really is and does"; 2) there is an absence of motivation to learn basic science, such as mathematics. Therefore, it is necessary to present them with realistic and motivating problems, which might include using robot platforms such as LEGO Mindstorms. In [12], LEGO Mindstorms was chosen to perform robotic experiments in an introductory "Control Engineering" course at the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil.

It is easier for students to learn when the subject is made real, since active learning has a better impact on students' understanding. To ensure this active learning, LEGO Mindstorms is introduced in other practical courses, such as [13].

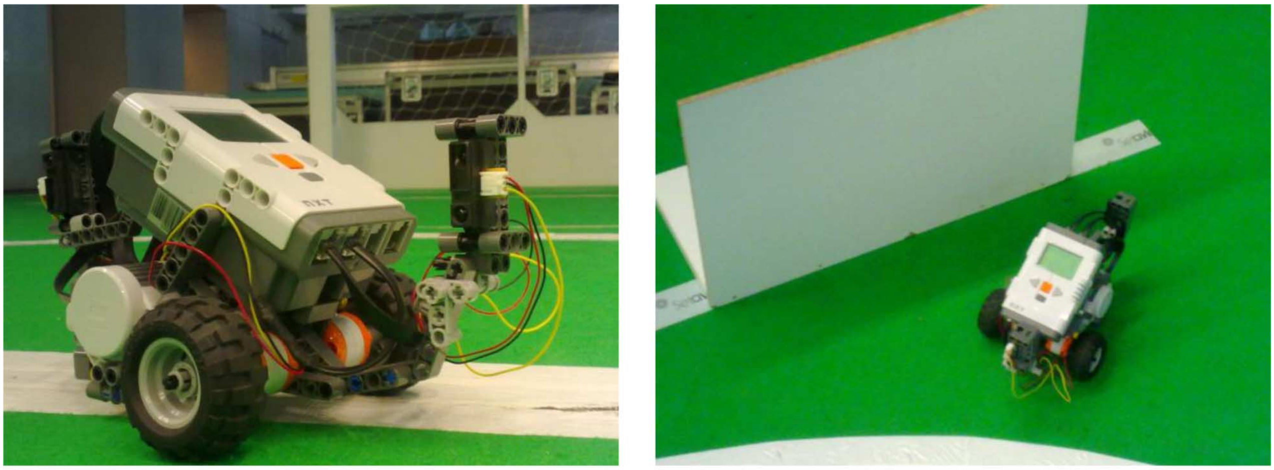
Behrens *et al.* [14] believe that no matter what students are taught and in how much detail they are taught it, they will only learn when the lessons actually come home to them, as they are put in the engineer's role.

LEGO Mindstorms makes it possible to build embedded systems without any prerequisite knowledge and to solve real problems with constraints such as sensor limitations, low computational power, and memory. LEGO Mindstorms' most important component is the LEGO Brick, which has dedicated software, inputs, and output and has both limitations and advantages. Kim and Jeon [15] describe how freshman engineers learn embedded systems using the LEGO Mindstorms as a tool.

Sharad [16] describes and analyzes the architecture of the LEGO Mindstorm NXT embedded system and its effectiveness when used to teach embedded systems and engineering concepts. Azlan *et al.* [17] explain the importance of using the LEGO Robot to understand the fuzzy logic theory.

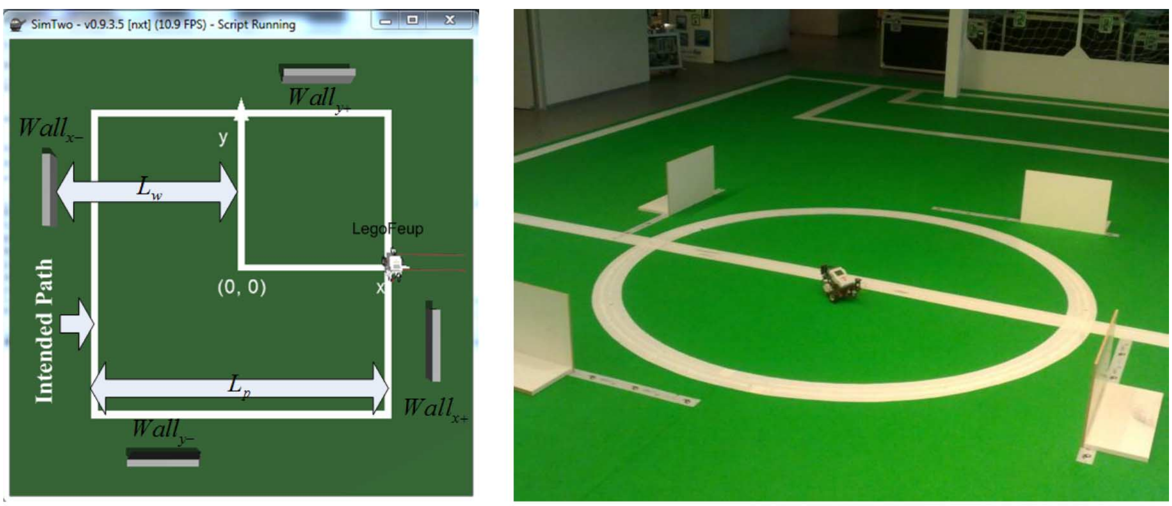
While other studies also describe the use of LEGO Mindstorms in teaching programming, artificial intelligence, and embedded systems to first-year engineering students [18]–[25], none of these covers localization-based on an EKF, the proposal and experiment described in this paper.

Extensive online communities, repositories, blogs, and competitions, such as [26]–[31], support LEGO Mindstorms. LEGO Mindstorms is widely used in education and outreach activities as a way of teaching programming, embedded systems, artificial intelligence, and simultaneously motivating students. However, it tends to be used only during the first years of the engineering curriculum. The authors propose an experiment that could teach more complex topics, such as localization and estimation based on EKF, to older undergraduate students.



(a) (b)

Fig. 1. Real scenario and LegoFeup. (a) LEGO NXT: LegoFeup. (b) Real LEGO and features.



(a) (b)

Fig. 2. Scenario. (a) Simulation Scenario, (b) Real Scenario.

III. Description of the Experiment

The LEGO NXT kit has three servomotors and a sensor each for touch, sound, ultrasound, and light. Furthermore, the kit has a fundamental component: the NXT brick that allows for the development of the robot's software using, for instance, RoboLab, Lejos JAVA for LEGO Mindstorms [32]. In complex and longer programming tasks, such as in the implementation of a Kalman filter, an object-oriented language such as JAVA for LEGO Mindstorms is preferred. The LabView RoboLab is not a satisfactory solution because the NXT memory becomes full with just a few programming blocks.

A. LegoFeup Description

The LegoFeup, shown in Fig. 1, is a differential vehicle with two traction wheels and one free wheel. Each servomotor has an encoder capable of measuring the number of turns of the correspondent wheel. It has two infrared sensors as well as IR Sharps [33], which are able to measure distances relative to objects in the sensor's range. If the LegoFeup navigates inside a certain area, describing a random or a predefined trajectory (for instance a square, rectangle, or circle), predicting its position based on odometry, and using encoders on each wheel, the odometry leads to the an unlimited increase of pose estimation error.

The LegoFeup vehicle has the ability to measure its distance and orientation with respect to landmarks (for instance white walls) every time that the landmark comes within the range of the infrared sensors. If white walls are placed at known positions inside the area where the robot is navigating, its position can be corrected: 1) directly, with 100% confidence ignoring the sensor's noise; or 2) by implementing the EKF as a probabilistic method.

B. Scenario

A particular scenario was used during the experiment described here (see Fig. 2). Four walls are placed at known positions forming a square. The wall_{x+} and wall_{x-} are placed parallel to the x-axis, with L_w and $-L_\mathrm{w}, x$ coordinates, respectively, while the wall_{y+} and wall_{y-} are placed parallel to the

y-axis, with $L_{\rm w}$ and $-L_{\rm w}, y$ coordinates, respectively. The trajectory that the LegoFeup should follow is a square-shaped path of length $L_{\rm p}$. The white walls are placed at the first meter of each square side. The student experiment includes a set of tasks, described in Section VIII.

The experiment was performed in a simulation scenario and in a real scenario (using the LEGO NXT). The simulator SimTwo [34] used in the experiment takes into account collisions between rigid bodies and considers additive noise in sensor measurements, thus making it very realistic. The test in the simulation scenario is very important as a first validation. Furthermore, the simulation does not require space, a physical scenario, or a real robot. This means that a large number of student groups can perform the experiment at the same time. The simulation scenario is shown in Fig. 2(a), while the real scenario is presented in Fig. 2(b).

IV. Teaching Method

The difference between the successful and nonsuccessful teaching of any topic can be seen by comparing practical results and theoretical knowledge. When the theoretical concepts are put into practice, the knowledge becomes much more solid.

When taking into account all of the objectives discussed in Section I-B, the goal of the Mobile Robots course is not to explain the entire process of the implementation of the algorithm that runs the localization based in the EKF. This would be a difficult and time-consuming task, which may not yield immediate results. The fundamental goal is to lead students to understand the EKF and its application in the localization problem, without worrying them about implementation issues, such as communication between applications, program cycle, synchronism between sensor readings, and the like. Therefore, the authors built and implemented a "LegoFeup tool," which included: 1) the simulation (SimTwo) and real scenario, presented in Section III; 2) the LegoFeup loop using the Java programming language, described in Section V; 3) communication with the simulation scenario or the real scenario (UDP or Bluetooth) using Java, explained in Section VI; and 4) the GUI interface, described in Section VII, that allows the students to perform a set of tasks as described in Section VIII-A.

The great advantage of the LegoFeup tool in undergraduate teaching is that it makes it possible to change the parameters of the estimation and control, as described in Section VII, and obtain immediate results, which helps in understanding how the EKF works.

V. LegoFeup Loop

The LegoFeup Loop can be divided into four different courses: 1) the vehicle path "Control Module"; 2) the "Feature Association Module"; 3) the "Observation Module," provided by the IR sensors; and 4) the "Estimation Module," using the EKF. The entire LegoFeup loop is shown in Fig. 3.

The loop inputs are the value of the IR sensors and the number of pulses between two consecutive time-steps, Odo_1 and Odo_2. The sensor values are converted into distances, $d_{\rm s1}$ and $d_{\rm s2}$, using the characteristic curve of the IR Sharp sensors, discussed in Section V-C.

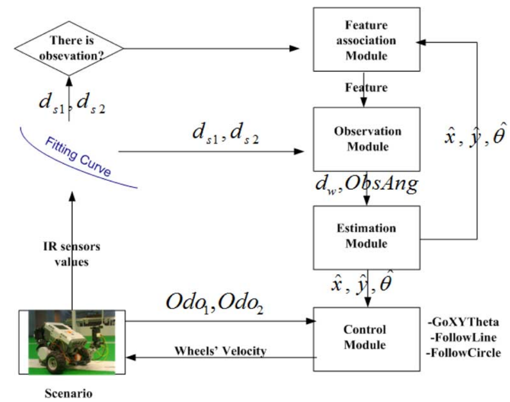


Fig. 3. LegoFeup loop.

If the distance value in both sensors is less than 30 cm, then it is considered that an observation was made. In that case, the observation is assigned to the correct feature (wall) using the actual estimated state ("Association Module"). Therefore, the observation $Z(d_{\rm w}, \mathrm{ObsAng})$ is generated by the "Observation Module" and passed to the "Estimation Module," which is responsible for predicting and updating the vehicle's state estimation, the 2-D coordinates $(\hat{x}, \hat{y})$, and the orientation relative to the x-axis $(\hat{\theta})$. Finally, the estimated vehicle state will be used in the "Control Module," which controls the robot's intended velocities, V_1 and V_2, for each wheel to follow the intended path.

A. Control Module

The control module has the estimated state variables $\hat{x}, \hat{y}$, and $\hat{\theta}$ as inputs given by the estimation module. The module's objective is to make the vehicle follow an intended path using the knowledge of the estimated state variables and a set of predefined routines: 1) "GoXYTheta" and 2) "FollowLine."

In addition to the state variables, these navigation routines receive the linear reference and angular velocities $V_{\rm ref}$ and $W_{\rm ref}$ as inputs.

The "GoXYTheta" routine should allow the vehicle to reach a certain point in space, located at position XY, finishing with Theta of the orientation. Other routine inputs are the intended point XY and the desired final orientation.

The aim of the "FollowLine" is that the LEGO NXT follows a line trajectory. In the case of the "FollowLine" routine, the other inputs are the line that is to be followed and the direction of, or two points on, the line.

Finally, the control module implements a "FollowPath" routine, where the basic routines described above are used to perform a path. An example of a path is that shown in Fig. 2(a). The path is square, and its sides are lines in the positive and negative directions x and y. All of these routines have the linear and rotation velocities (V and w) as outputs, which are afterwards converted into velocities of each wheel using the following expressions:

$$V_1 = V + \frac{b \cdot W}{2} \quad V_2 = V - \frac{b \cdot W}{2} \tag{1}$$

where b is the distance between wheels and V_1, V_2 are the velocities for each wheel.

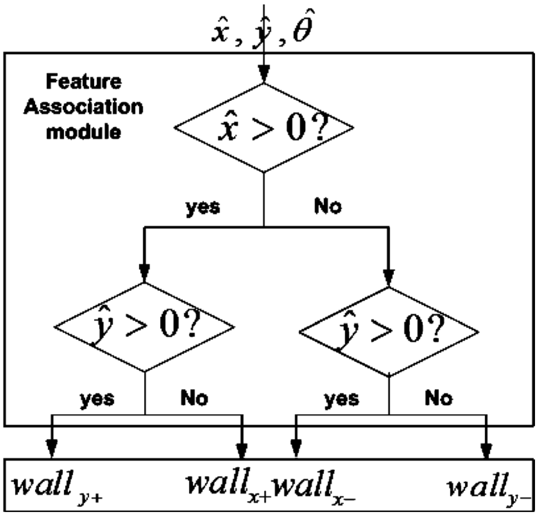


Fig. 4. Feature association module.

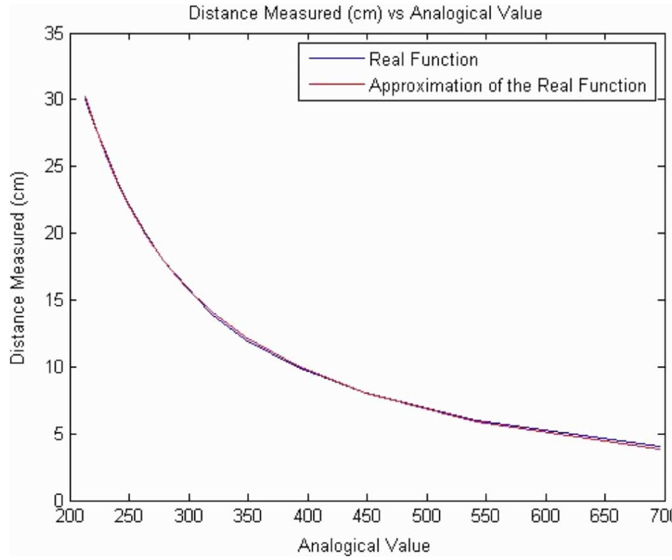


Fig. 5. Real and fitted curve.

B. Association Module

If the LegoFeup passes close enough to a wall that the infrared sensors are able to measure the distances, a new observation can be generated. Before the observation is generated, it is necessary to assign the observation to the correct wall. In the case of the experiment described here, the feature association module could be summarized by the diagram shown in Fig. 4.

C. Observation Module

An example of an IR sensor is the Sharp IR Range Finder, which offers high accuracy without the influence of sunlight and is easy to use. It has low power consumption, a small thin beamwidth, and good range (to meters) and is available at a low price [33].

The IR sensor used in this experiment measures distances between 4 and 30 cm. The characteristic curve of the Sharp IR sensors is represented by the empirical graph shown in Fig. 5. It can be seen that the distance value (in centimeters) is nonlinearly dependent on the sensor's analog value. The best function that fits the sensor curve is given by the following expression:

$$D = a \cdot \left(\frac{C_M}{A + C_R} - b \right) \tag{2}$$

where A is the NXT read value and D is the distance. C_M is the constant of multiplication and C_R is the constant of linearization. a and b are the linear function constants.

Fig. 5 shows the real acquired values and the results of the best relation that makes it possible to compute the distance using the analog Sharp's value. The best relation was obtained by optimizing, on MATLAB, the sum of the quadratic error between the real acquired values and the fitting curve.

Using two IR sensors, it is possible to measure the distance d_{w} and the orientation ObsAng of the robot related to the feature wall. The observation module is given by the following equations (deduced from Fig. 6):

$$\mathrm{obsAng} = \mathrm{atan}\left(\frac{d_{\mathrm{s1}} - d_{\mathrm{s2}}}{L} \right)$$

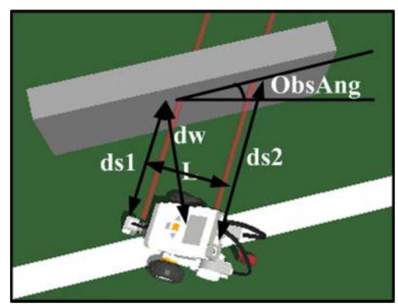


Fig. 6. Angle observed (ObsAng) and distance to the wall (d_{w}).

$$d_w = \frac{d_{\mathrm{s1}} + d_{\mathrm{s2}}}{2} \cdot c(\mathrm{ObsAng}). \tag{3}$$

Due to the maximum range of the infrared sensors, every time the distance measured in both the IR sensors is less than or equal to 30 cm, it is considered that an observation has been made.

At each wall, only two states could be observed with the measurement of the distance from the wall. Therefore, the observation model, $Z = [x \text{ or } y\ \theta]^T$, will be given by the following equation:

$$Z(\mathrm{wall}_{x+}) = \begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} L_{\mathrm{w}} - d_{\mathrm{w}} \\ \frac{\pi}{2} - \mathrm{ObsAng} \end{bmatrix}$$
$$Z(\mathrm{wall}_{x-}) = \begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} -(L_{\mathrm{w}} - d_{\mathrm{w}}) \\ -\frac{\pi}{2} + \mathrm{ObsAng} \end{bmatrix} \tag{4}$$
$$Z(\mathrm{wall}_{y+}) = \begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} L_{\mathrm{w}} - d_{\mathrm{w}} \\ \pi + \mathrm{ObsAng} \end{bmatrix}$$
$$Z(\mathrm{wall}_{y-}) = \begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} -(L_{\mathrm{w}} - d_{\mathrm{w}}) \\ \mathrm{ObsAng} \end{bmatrix}. \tag{5}$$

The observation error r is approximated by Gaussian noise (with zero mean and covariance R). The covariance R is an important parameter for the updating step of the EKF and is described in the following section. The observation noise vector is equal to the following: $r = [\varepsilon_{ds1}\ \varepsilon_{ds2}]^T$.

D. Estimation Module: Extended Kalman Filter

The EKF is a Kalman Filter estimator in which the nonlinear kinematic and observation models are transformed into linear models using the Taylor expansion to update the covariance (see chapter "Nonlinear Estimation" in [1]).

The EKF, as shown in Algorithm 1, consists of two steps that work in a cycle. First, the vehicle state and covariance are predicted (EKF prediction). Then, if there is a new observation, this is assigned to the correct feature and the vehicle state and covariance, are corrected (EKF update).

Algorithm 1: Extended Kalman Filter

$P_0 \leftarrow$ Initial Covariance
$X_0 \leftarrow$ Initial State
for $i \leftarrow 1$ to Max Steps do
EKF Prediction
  $[X_k \quad P_k] \leftarrow$ Predict by Odometry
New Observations?
  $[Z_k] \leftarrow$ Measures
Data Association: Associate to the correct Feature—Wall
EKF Update
  $[X_k \quad P_k] \leftarrow$ Update (Kalman Gain, Inovation)
end for

1) Prediction Step: The kinematic of the vehicle can be defined by the following equation:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V \cdot c(\theta) \\ V \cdot s(\theta) \\ \omega \end{bmatrix} \tag{6}$$

where $x, y,$ and θ are the state variables (X), 2-D position, and orientation, respectively. The vehicle speed as it moves forward is represented by V, and the rotation speed by ω.

$c(\cdot)$ and $s(\cdot)$ represent $\cos(\cdot)$ and $\sin(\cdot)$. This nomenclature will be used throughout this paper. In discrete time, the dynamics can be described by the kinematic that is based on the centered differences

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} (k+1) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} (k) + \begin{bmatrix} d \cdot c\left(\theta + \frac{\Delta\theta}{2}\right) \\ d \cdot s\left(\theta + \frac{\Delta\theta}{2}\right) \\ \omega + \Delta\theta \end{bmatrix} \tag{7}$$

where d is the distance covered by the robot and $\Delta\theta$ is the angle variation between the two steps

$$d = \frac{d_1 + d_2}{2} \quad \Delta\theta = \frac{d_1 - d_2}{b} \tag{8}$$

where d_1 and d_2 are the distances traveled by the first and second wheel, respectively, and b is the distance between the two wheels.

The encoder gives the number of pulses for each wheel between two continuous steps. This input, $U = [\mathrm{Odo}_1 \quad \mathrm{Odo}_2]$, makes it possible to estimate the distance traveled by each wheel using the following expressions:

$$d_1 = \frac{\mathrm{Odo}_1}{\mathrm{Codo}_1} \quad d_2 = \frac{\mathrm{Odo}_2}{\mathrm{Codo}_2} \tag{9}$$

where Codo_1 and Codo_2 are the constants of odometry in each wheel, i.e., the number of encoder pulses per meter.

The error in the odometry appears in the measurement of d and $\Delta\theta$, resulting in an error of $d_1, d_2,$ and b. Therefore, the real kinematic of the vehicle (taking the measurement errors into account) could be written as: $X(k+1) = f(X(k), U, q)$, where the nonlinear function f is given by

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} (k+1) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} (k) + \begin{bmatrix} (d+\varepsilon_d) \cdot c\left(\theta + \frac{\Delta\theta + \varepsilon_{\Delta\theta}}{2}\right) \\ (d+\varepsilon_d) \cdot s\left(\theta + \frac{\Delta\theta + \varepsilon_{\Delta\theta}}{2}\right) \\ \omega + \Delta\theta + \varepsilon_{\Delta\theta} \end{bmatrix} \tag{10}$$

where $q = [\varepsilon_d \quad \varepsilon_{\Delta\theta}]$, taking into account q Gaussian noise (with mean zero and covariance matrix equal to Q). Performing the Taylor expansion, using only the first-order term, the prediction step equations are given by

$$\hat{X}(k+1) = f(\hat{X}(k), U, 0). \tag{11}$$

The estimated covariance is equal to (see chapter "Nonlinear Estimation" in [1])

$$P(k+1) = \nabla f_X P(k) \nabla f_X^T + \nabla f_q Q \nabla f_q^T \tag{12}$$

where ∇f_X is the dynamics gradient used in order to estimate state $\hat{X}$, equal to $(\partial f)/(\partial X)$. While $\nabla f_q = (\partial f)/(\partial q)$ is the gradient used in order to the Gaussian noise q, taking the mean noise into account, which is by definition equal to zero

$$\nabla f_X = \begin{bmatrix} 1 & 0 & -d \cdot s\left(\hat{\theta}(k) + \frac{\Delta\theta}{2}\right) \\ 0 & 1 & d \cdot c\left(\hat{\theta}(k) + \frac{\Delta\theta}{2}\right) \\ 0 & 0 & 1 \end{bmatrix}$$
$$\nabla f_q = \begin{bmatrix} c\left(\hat{\theta}(k) + \frac{\Delta\theta}{2}\right) & -\frac{1}{2} d \cdot s\left(\hat{\theta}(k) + \frac{\Delta\theta}{2}\right) \\ s\left(\hat{\theta}(k) + \frac{\Delta\theta}{2}\right) & \frac{1}{2} d \cdot c\left(\hat{\theta}(k) + \frac{\Delta\theta}{2}\right) \\ 0 & 1 \end{bmatrix}. \tag{13}$$

The predicted covariance depends on the dynamics and previous covariance: $\nabla f_X P(k) \nabla f_X^T$. The term $\nabla f_q Q \nabla f_q^T$ increases the covariance and the uncertainty until an observation arrives.

2) Update Step: The state update is carried out during every cycle when an observation is obtained with the following equation:

$$\hat{X}(k+1) = \hat{X}(k) + W(k) \cdot V(k) \tag{14}$$

where W is the Kalman gain and the innovation. The innovation in the $\mathrm{wall}_{x+/-}$ and $\mathrm{wall}_{y+/-}$, respectively, is equal to [see (4) and (5)]

$$V(k) = \begin{bmatrix} \hat{x} \\ \hat{\theta} \end{bmatrix} - z \quad V(k) = \begin{bmatrix} \hat{y} \\ \hat{\theta} \end{bmatrix} - z. \tag{15}$$

The extended Kalman filter gain is equal to the following (see chapter "Nonlinear Estimation" in [1]):

$$W(k) = P(k) \nabla h_X^T \left[\nabla h_X P(k) \nabla h_X^T + \nabla h_r R \nabla h_r^T\right]^{-1} \tag{16}$$

Defining the observation equal to $z = h(X, U, r)$, the gradient $(\partial h)/(\partial X) = \nabla h_X$, in order to the state, is equal to

$$\nabla h_X = \begin{bmatrix} C_1 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

when the walls that are seen are $\text{wall}_{x+/-}$, then $C_1 = 1, C_2 = 0$. In the case of $\text{wall}_{y+/-}$, then $C_1 = 0, C_2 = 1$. By defining

$$\frac{\text{dObsAng}}{dd_{s1}} = \frac{L}{L^2 + (d_{s1} - d_{s2})^2} \tag{18}$$

$$\begin{aligned} a &= \frac{1}{2} c(\text{ObsAng}) \\ b &= -\frac{d_{s1} + d_{s2}}{2} s(\text{ObsAng}) \frac{\text{dObsAng}}{dd_{s1}} \end{aligned} \tag{19}$$

the gradient of the observation for the noise $r, (\partial h)/(\partial r) = \nabla h_r$, is equal to:

$$\begin{aligned} \nabla h_r &= \begin{bmatrix} \dfrac{\text{signal} \cdot \frac{dd}{dd_{s1}}}{\frac{\text{dObsAng}}{dd_{s1}}} & \dfrac{\text{signal} \cdot \frac{dd}{dd_{s2}}}{-\frac{\text{dObsAng}}{dd_{s1}}} \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{\text{signal} \cdot (a + b)}{\frac{\text{dObsAng}}{dd_{s1}}} & \dfrac{\text{signal} \cdot (a - b)}{-\frac{\text{dObsAng}}{dd_{s1}}} \end{bmatrix} \end{aligned} \tag{20}$$

where signal is equal to -1 when the observed walls are wall_{x-} or wall_{y-}. On the contrary, when the observed walls are wall_{x-} or wall_{y-}, signal is equal to 1.

The covariance update is performed taking the following equation into account (see chapter "Nonlinear Estimation" in [1]):

$$P(k) = [I - W(k)\nabla h_X]\nabla h_X^T P(k). \tag{21}$$

VI. LegoFeup Communication

Both scenarios, real and simulation, are able to exchange information with a host computer on which the LegoFeup Loop algorithm is running. The information exchanged is: 1) the vehicle's sensor measurements that go from the scenario to the LegoFeup Loop algorithm (the encoder values of each wheel—odometry and the infrared distances); 2) in the opposite direction, from the LegoFeup Loop to the scenario, the velocity of each wheel calculated by the EKF algorithm is sent, V_1 and V_2.

Fig. 7 shows the structure of communication between the EKF algorithm, the interface, and the scenario. Fig. 7 also shows the change between the real (Bluetooth) and simulation (UDP) scenarios.

VII. User Interface

The user interface, as shown in Figs. 8 and 9, makes it possible to "start" and "stop" the algorithm or control the LegoFeup in a remote way using keyboard keys.

This interface also makes it possible to "set" the EKF algorithm configuration: 1) the covariance matrix Q (distance variance Q_{11}, the angle variance Q_{22}); 2) the sensor covariance R (R_{11} and R_{22}); 3) the initial covariance matrix; 4) the initial vehicle location; 5) the trajectory controller parameters; and 6) the vehicle reference velocities V_{ref} and ω_{ref}.

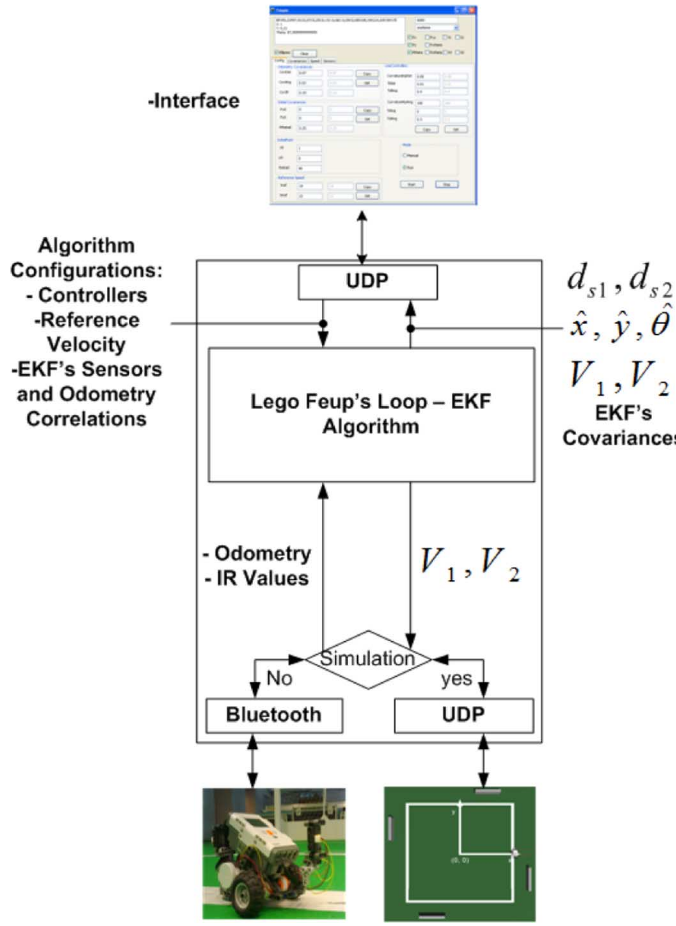


Fig. 7. Algorithm, communication, and interface.

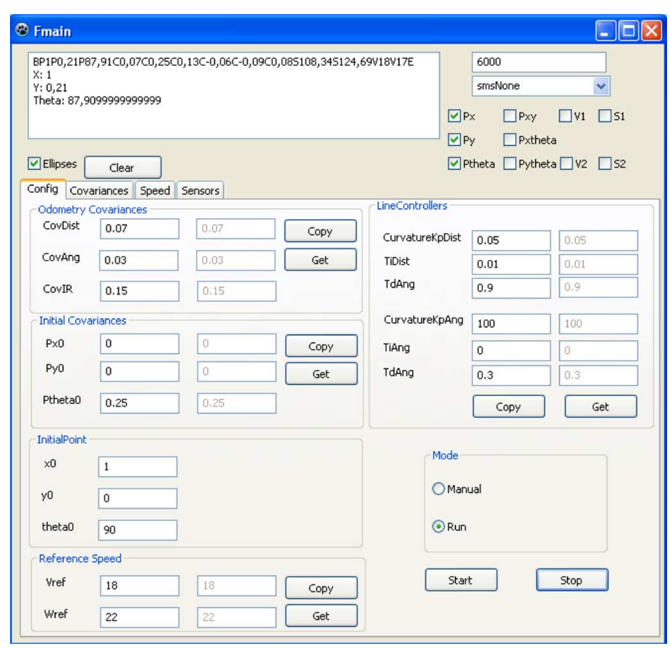
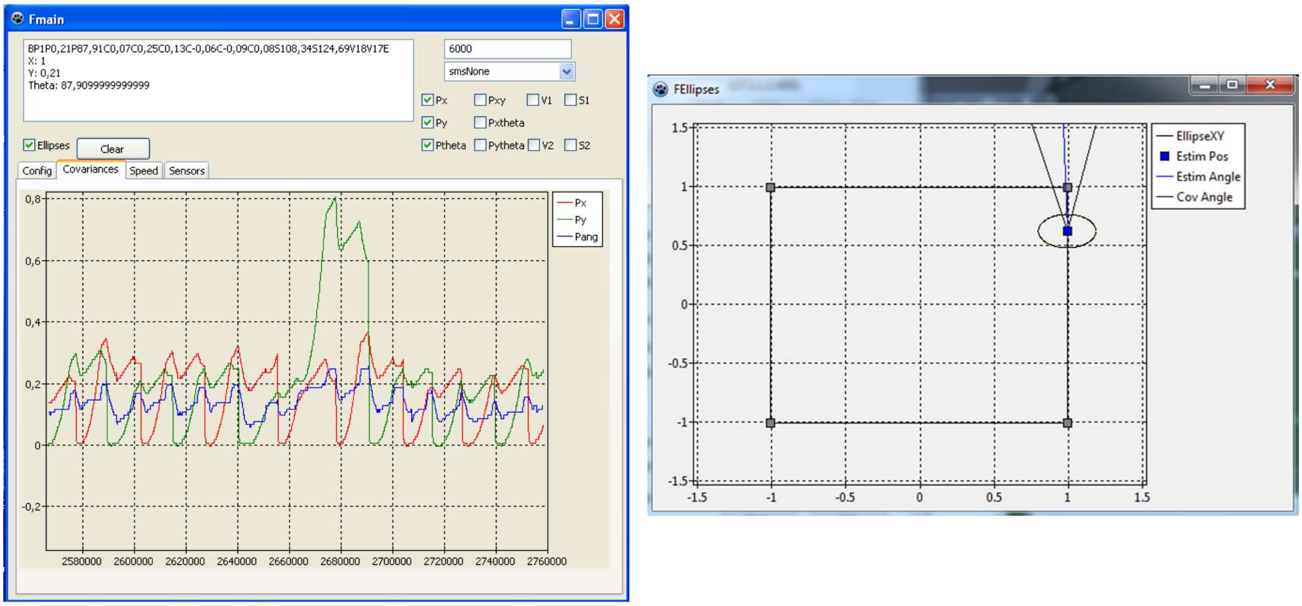


Fig. 8. Main screen: configuration tab.

To help undergraduate students understand the EKF concepts more clearly, the interface was developed to show the extended Kalman filter variables: 1) the covariance matrix; 2) the covariance ellipse; 3) the vehicle's estimated position. The interface



(a) (b)

Fig. 9. User interface. (a) Main screen where the covariance P_x, P_y, and P_θ are plotted. (b) Main screen where the covariance P_x, P_y, and P_θ are plotted.

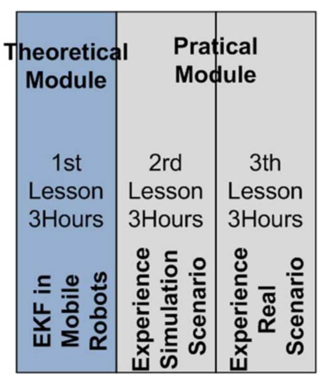


Fig. 10. Lessons Organizations.

also shows the velocities of the vehicle's wheels V_1 and V_2 and the values of the infrared sensors, d_{s1} and d_{s2}. This interface communicates with the LegoFeup Loop algorithm via UDP.

The form, called "FEllipses," shown in Fig. 9(b), shows the intended path, the vehicle's estimated position, and the vehicle's estimated orientation. Consequently, it is also possible to see the covariance ellipse with a confidence of 68.2% (one times the standard deviation). The user interface lets students tune the filter and see the immediate effect on the extended Kalman filter's behavior. Therefore, this module provides students with the opportunity to learn about EKF operation, the importance of the covariance matrices of the odometry and sensor noise, and how these affect the operation of the filter.

VIII. Lesson Organization

The topic "Localization of Mobile Robots using an Extended Kalman Filter in a LEGO NXT" was taught in three 3-h lessons, shown in Fig. 10. During the first lesson, the theoretical content was presented including the theory behind the Kalman filter, plus its mathematical formulation. All of the modules of the LegoFeup loop explained in Section V were presented in detail during the lesson.

The second and third lessons consisted of teaching the practical module, with pairs of student performing the experiment. During the first practical lesson, the students familiarized themselves with the LegoFeup tool and with the simulation scenario. After gaining an understanding of the algorithm/tool, an experiment in the simulation scenario was performed that was based on the guide presented in Section VIII-A. The third and last lesson finishes the test by configuring the EKF as performed in the simulation scenario, but this time in the real scenario. The test was performed one group at a time.

A. Experiment Guide

In the simulation scenario, Gaussian noise with zero mean and standard deviation equal to $\sigma = 0.01$ m is introduced in the Sharp IR sensor readings. In the simulation, it is possible to select "on" or "off" Gaussian noise introduced in the Sharp IR sensors. A Gaussian noise with zero mean and standard deviation of $\sigma = 0.01$ pulses is also introduced in the encoders' readings, simulating the error in the odometry that appears with the wheels' radius changes and caused by the variation of the distance between wheels. These values are measured experimentally by the authors and proved to be close to the real behavior of the vehicle and Sharp IR sensors.

The experiment consists of the following steps in the simulation scenario.

1) Executing a square trajectory based only on the odometry.
2) Executing the same trajectory with direct localization, sensors' covariance matrix R with very low values, based on the Sharp IR measurement without noise (turn "off" the noise).
3) Adding Gaussian noise in the Sharp IR sensors' readings (turn "on" the noise) and repeating the previous step.

4) Putting the EKF's sensors' covariance matrix $R = \sigma^2$ equal to the Gaussian noise introduced in the simulation scenario. Giving the odometry covariance matrix Q a very low value and completing the same trajectory.
5) Changing the value of the covariance matrix, Q, to a very high value and repeating the previous step.
6) Using the results of the previous steps to tune the filter to the best solution for Q and R. This last test and respective filter tuning should be kept and implemented in the real scenario.

The students were given instructions on how to perform each step. Each group delivered a report at the end of the lesson explaining their vehicle's behavior at each step and justifying their conclusions.

The students should conclude at Step 1 that the vehicle covariance should increase without bounds due to the absence of observations. The vehicle does not update or correct its position with the walls, and it will then rapidly become lost. At Step 2, the user has a high level of confidence in the infrared sensor readings at the EKF filter, and the actualization is done directly with 100% confidence. At this stage, the algorithm localizes the vehicle with success, but only because the scenario is ideal, i.e., the Sharp IR measurements are obtained without noise. At Step 3, confidence in the infrared sensor readings at the EKF filter is high, and the actualization is done directly with 100% confidence. However, in this case the algorithm does not localize the vehicle correctly because the scenario in this step is not ideal, i.e., the infrared sensors have noise. At Step 4, students should conclude that the filter should not work since the sensors' measurements are rejected because confidence in the odometry is high. Then, the localization is performed based only on the odometry. At Step 5, the simulated situation is similar to that of Step 2. Now, instead of having high confidence in the sensors, the user has less confidence in the odometry than in the infrared sensors. This situation leads the filter to update the vehicle localization directly, with a confidence near to 100% when a wall is observed.

B. Course Web Page

Videos and theoretical content on the implementation of the experiment described here can be found in [35]. Furthermore, an application of the EKF algorithm can be downloaded with some tests that could be performed during the simulation scenario. The theory taught during the theoretical lesson and the contents of the Experiment Guide are also available.

IX. Mobile Robots Course Results

The administration of a student questionnaire indicated the impact the Mobile Robots course had on the students' understanding. The large majority of them had heard about Kalman filters, but had never implemented them and did not understand how they operated or what their purpose was. By the end of the course, the large majority had completed the fundamental objective; they understood the extended Kalman filter and the localization based on the EKF as a probabilistic method of making the fusion between the odometry and the Sharp IR observations.

On the anonymous student survey, questions such as "Are you now more interested in mobile robotics than before the lessons?", "Do you think that the experiment and the tutorial will be helpful for future studies on mobile robotics topic?", or "Do you now understand, and feel capable of describing, the method of navigation and localization implemented?" received mostly positive answers.

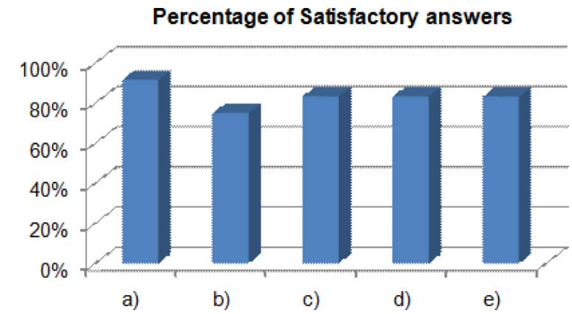


Fig. 11. Percentage of correct answers plotted.

The nonanonymous reports delivered by each group at the end of the practical lessons were evaluated by the authors, and they lead to the conclusion that the experiment was a success. The undergraduate students understood theoretical concepts based on a practical tool. According to the experimental guide presented in Section VIII-A, the reports showed a percentage of satisfactory answers in each step of the experiment, as shown in Fig. 11.

At the end of the semester, the students sat a final exam for the course. The exam had three questions relating to the experiment and the extended Kalman filter applied in the robot localization. By comparison to previous years' exams (where this tool was not used), the percentage of correct answers for this topic increased by about 20% with the introduction of the proposed tool in the Mobile Robots lessons. Furthermore, analyzing the correct answers highlights two aspects: consistency of reasoning and a correct visualization of the problem. These results prove that the method presented improves the students' understanding.

X. Conclusion

The intention of this work was to give undergraduate students a form of tutorial to perform experiments related to localization concepts, namely odometry and the EKF. The experiment described could be carried out in mobile robotics courses. LEGO NXT is widely used in an educational environment, but only to teach simple and basic concepts. The authors' proposal can be applied to more complex subjects such as the EKF estimator. Undergraduate students should learn the importance of using a probabilistic method, such as the EKF, to estimate a vehicle's position and to filter the environmental information. They should also understand basic EKF concepts: the EKF's cycle (predict and update) and the importance of odometry and sensor covariance, Q and R, for the correct operation of the EKF.

Due to the fact that the experiment can be performed not only in simulation, but also in a real scenario, students are able to apply the theoretical concepts to practical problems, thus motivating them. An improvement in the students' understanding of localization concepts using EKF can therefore be expected in undergraduate courses on robotics.

References

[1] A. Gelb, J. F. Kasper, Jr., R. A. Nash, Jr., C. F. Price, and A. A. Sutherland, Jr., *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 2001.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, Sep. 2005.

[3] S. W. Director, P. K. Khosla, R. A. Rohrer, and R. A. Rutenbar, "Reengineering the curriculum: Design and analysis of a new undergraduate electrical and computer engineering degree at Carnegie Mellon university," *Proc. IEEE*, vol. 83, no. 9, pp. 1246–1269, Sep. 1995.

[4] E. Krotkov, "Robotics laboratory exercises," *IEEE Trans. Educ.*, vol. 39, no. 1, pp. 38, 94–45, 97, Feb. 1996.

[5] "Robotic courses," The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2011 [Online]. Available: http://www.ri.cmu.edu/ri_static_content.html?menu_id=276

[6] "Electrical engineering and computer science," MIT, Cambridge, MA, May 2011 [Online]. Available: http://student.mit.edu/catalog/m6a.html

[7] "School of Electrical and Computer Engineering," Cornell University, Ithaca, NY, May 2011 [Online]. Available: http://www.ece.cornell.edu/

[8] "Department of Computing, Robotics course," Imperial College London, London, U.K., May 2011 [Online]. Available: http://www.doc.ic.ac.uk/teaching/coursedetails/333

[9] "Master in robotics, systems and control," ETH Zurich, Zurich, Switzerland, May 2011 [Online]. Available: http://www.master-robotics.ethz.ch

[10] J. Ruiz-del-Solar, "Robotics-centered outreach activities: An integrated approach," *IEEE Trans. Educ.*, vol. 53, no. 1, pp. 38–45, Feb. 2010.

[11] J. Ruiz-del-Solar and R. Avilés, "Robotics courses for children as a motivation tool: The Chilean experience," *IEEE Trans. Educ.*, vol. 47, no. 4, pp. 474–480, Nov. 2004.

[12] M. B. R. Vallim, J.-M. Farines, and J. E. R. Cury, "Practicing engineering in a freshman introductory course," *IEEE Trans. Educ.*, vol. 49, no. 1, pp. 74–79, Feb. 2006.

[13] A. B. Williams, "The quality impact of using Lego Mindstorms robots to teach computer engineering," *IEEE Trans. Educ.*, vol. 49, no. 1, p. 206, Feb. 2006.

[14] A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Ballé, T. Herold, A. Telle, T. G. Noll, K. Hameyer, and T. Aach, "MATLAB meets LEGO mindstorms—A freshman introduction course into practical engineering," *IEEE Trans. Educ.*, vol. 53, no. 2, pp. 306–317, May 2010.

[15] S. H. Kim and J. W. Jeon, "Introduction to embedded systems using Lego Mindstorms," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 99–108, Feb. 2009.

[16] S. Sharad, "Introducing embedded design concepts to freshmen and sophomore engineering students with Lego Mindstorms NXT," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ.*, San Diego, CA, 2007, pp. 119–120.

[17] N. Z. Azlan, F. Zainudin, H. M. Yusuf, S. F. Toha, S. Z. S. Yusoff, and N. H. Osman, "Fuzzy logic controlled miniature Lego robot for undergraduate training system," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl.*, Harbin, China, May 2007, pp. 2184–2188.

[18] D. Ye, I. Brutman, G. W. Georgi, and L. M. Folan, "Freshman project: Autonomous underwater vehicle (AUV)," presented at the ASAEE Annu. Conf. Expo., Honolulu, HI, Jun. 2007.

[19] S. H. Kim and J. W. Jeon, "Educating C language using Lego Mindstorms robotic invention system 2.0," in *Proc. IEEE Int. Conf. Robot. Automat.*, Orlando, FL, May 2006, pp. 715–720.

[20] D. C. Cliburn, "Experiences with the Lego Mindstorms throughout the undergraduate computer science curriculum," in *Proc. 36th ASEE/IEEE Frontiers Educ. Conf.*, San Diego, CA, Oct. 2006, pp. T2F-1–T2F-6.

[21] M. I. Garduno Mota, "Work In progress—Using Lego Mindstorms and Robolab as a mean to lowering dropout and failure rate in programming course," in *Proc. 37th ASEE/IEEE Frontiers Educ. Conf.*, Milwaukee, WI, Oct. 2007, pp. F4A-1–F4A-2.

[22] M. A. Garcia and H. P. Mc-Neill, "Learning how to develop software using the toy Lego Mindstorms," presented at the 7th Annu. Conf. Innov. Technol. Comput. Sci. Educ., Aarhus, Denmark, Jun. 2002.

[23] A. Nagchaudhuri, G. Singh, M. Kaur, and S. George, "Lego robotics products boost student creativity in precollege programs at UMES," in *Proc. 32nd ASEE/IEEE Frontiers Educ. Conf.*, Boston, MA, Nov. 2002, vol. 3, pp. S4D-1–S4D-6.

[24] F. Klassner and S. D. Anderson, "Lego Mindstorms: Not just for K-12 anymore," *IEEE Robot. Automat. Mag.*, vol. 10, no. 2, pp. 12–18, Jun. 2002.

[25] C. Mc Goldrick and M. Huggard, "Peer learning with Lego Mindstorms," in *Proc. 34th ASEE/IEEE Frontiers Educ. Conf.*, Oct. 2004, vol. 3, pp. S2F-24–S2F-29.

[26] "LEGO: Mindstorms," LEGO, Billund, Denmark, May 2011 [Online]. Available: http://www.lego.com

[27] "LEGO Mindstorms," LEGO, Billund, Denmark, May 2011 [Online]. Available: http://www.legomindstorms.com

[28] "The NXT step—LEGO Mindstorms NST blog," May 2011 [Online]. Available: http://www.thenxtstep.blogspot.com/

[29] "Lugnet," May 2011 [Online]. Available: http://www.lugnet.com/

[30] "RoboCup Junior," May 2011 [Online]. Available: http://rcj.robocup.org/

[31] "RoboCup 2010 at Singapore," May 2011 [Online]. Available: http://www.robocup2010.org/

[32] "Lejos, Java for LEGO mindstorms," May 2011 [Online]. Available: http://lejos.sourceforge.net/

[33] "IR Range Finder. Society of Robots," May 2011 [Online]. Available: http://www.societyofrobots.com/sensors_sharpirrange.shtml

[34] P. Costa, "Paco-Wiki: SimTwo," May 2011 [Online]. Available: http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo

[35] "Lego NXT—EKF Web page," University of Porto, Porto, Portugal, May 2011 [Online]. Available: http://paginas.fe.up.pt/~miguelp/LegoNxt

Miguel Pinto (M'09) was born in Caracas, Venezuela, in 1986. He received the M.Sc. degree in electrical engineering from the University of Porto, Porto, Portugal, in 2009, and is currently pursuing the Ph.D. degree in electrical and computer engineering at the Robotic and Intelligent Systems Unit, Institute for Systems and Computer Engineering of Porto (INESC Porto), Porto, Portugal.

After completing his M.Sc. studies, he was a Guest Assistant with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto. His main research areas are in process control and robotics, navigation, and localization of autonomous vehicles, including terrain and marine robotics.

António Paulo Moreira (M'09) received the degree in electrical engineering, M.Sc. degree in electrical engineering—systems, and Ph.D. degree in electrical engineering from the University of Porto, Porto, Portugal, in 1986, 1991, and 1998, respectively.

From 1986 to 1998, he also worked as an Assistant Lecturer with the Electrical Engineering Department, University of Porto. He is currently a Lecturer in electrical engineering, developing his research within the Robotic and Intelligent Systems Unit, Institute for Systems and Computer Engineering of Porto (INESC Porto), Porto, Portugal. His main research areas are process control and robotics.

Aníbal Matos (M'08) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering from the University of Porto, Porto, Portugal, in 1991, 1994, and 2001, respectively.

He is currently an Assistant Lecturer with the Electrical and Computer Engineering Department, University of Porto, and is also a Researcher with the Robotics and Intelligent Systems Unit, Institute for Systems and Computer Engineering of Porto (INESC Porto), Porto, Portugal. His research areas include modeling, navigation and control of autonomous vehicles, nonlinear control systems, and marine robotics.