Localization of Mobile Robots using an Extended Kalman Filter in a Lego NXT

Miguel Pinto, *IEEE Member (M'09)*, A. P. Moreira, *Member (M'05)*, A. Matos, *Member (M'08)*,
Department of Electrical Engineering and Computers, Faculty of Engineering from University of Porto.
INESC Porto - Institute for Systems and Computer Engineering of Porto
Porto, Portugal
(miguelp,amoreira,anibal)@fe.up.pt

***Abstract*— The Lego Mindstorms is widely used in the education as a way to teach programming languages, robotic and embedded systems, in a more fun and exciting way, aiming to motivate engineering students.**

This paper was inspired by the success that the experience presented had during the Mobile Robots course in the Faculty of Engineering of the University of Porto, Portugal, where navigation and localization of mobile robotic was taught.

The module/work presented in this paper could be used in education to teach mobile robotic concepts, namely when the fundamental objective is to teach navigation and localization using the Extended Kalman Filter (EKF).

The Lego Mindstorms NXT is used to build a differential robot. The test is performed in simulation and real scenarios. The Lego robot, also called LegoFeup by the authors, is equipped with tachometers and infrared sensors.

The Electrical and Computers engineering curriculum from the Faculty of Engineering of the University of Porto, Portugal, is constituted, among others, by the automation major with a specialization in robotics. The undergrad students of the robotic minor, at the 5th year of the master graduation, have the opportunity to learn navigation and localization concepts during the Mobile Robots course, where the topic "Localization of Mobile Robots using an Extended Kalman Filter in a Lego NXT" is taught.

In the Mobile Robots course a tool is available to the students, aiming their learning about the EKF applied in the localization of mobile robots in a particular scenario. The implementation of such tool is presented here. Also, the Mobile Robots course theoretic and practical lessons, scheduled, exposed material and student's evaluation are described here.

I. INTRODUCTION

The paper was motivated by the success that the experience described here had during the Mobile Robots course, carried out by undergraduate students from the Electrical and Computers Engineering curriculum from the Faculty of Engineering of the University of Porto, Portugal. The course base books are: [1] and [2].

The Electrical and Computers Engineering curriculum from the Faculty of Engineering of the University of Porto receive all years about two hundred students. During the first and second years, essential theoretic concepts are taught, such as: mathematics, physics, programming courses and electronics. After the second year the undergrad students have to choose a branch: automation, energy or telecommunications. In the automation branch, the students still have to choose a specialization. Among others, the freshman engineers could follow the robotic specialization, where courses about industrial robotics and automation, artificial intelligence, artificial vision, acquisition and data possessing and topics about estimation and identification are taught. Still, the robotic specialization offers to the freshman engineers the possibility to learn about navigation and localization of mobile robots during the course of Mobile Robots.

A. Mobile Robots Course

The Mobile Robot course covers a wide area in the localization and navigation topic: relative and global localization, kinematics and sensor fusion. Inside the sensor fusion topic, probabilistic methods, as the Particle and Kalman Filter, are taught. This document particularly focuses the attention in the Kalman Filter. In that way, the intention with this paper is to describe the experience realized by the freshmen engineers of Mobile Robots and evaluated by their teachers, with Lego Mindstorms, that could be helpful to teach the same topic (Localization of Mobile Robots using an Extended Kalman Filter in a Lego NXT), during similar courses. However, the experience can be easily transformed and performed using other localization method, as is example the Particle Filter.

B. *Contribute in the students learning*

The main objectives and contribution in the students learning with the experience presented here are: 1) the undergrad students should realize the importance of the sensor fusion to relate the environment information with the odometry; 2) understand the necessity of a probabilistic method, as the Kalman Filter, instead the direct fusion between the environment information and the odometry; 3) realize how the Kalman filter could be applied to a practical problem inside the localization of mobile robots; 4) learn deeply the operation of a Kalman filter, its cycle, the prevision and update steps, as the importance of the covariance matrices and the respective tuning in the correct operation of the Extended Kalman Filter (EKF).

II. BACKGROUND

The teaching of unmotivated mathematics and sciences without immediate application, followed by the description of engineering applications was the most common approach for more than four decades, instead of an education based on practice engineering. Director *et al.* [8], believe that only when the engineer curriculum begins to be seen as a whole, and not as a "constant republishing of aging courses", it will have a real impact. Therefore, it is important to give freshman engineers the ability to face real and complex problems (from mathematics, through computer science to reach applied physics, like in mobile robotics).

A set of laboratory courses, such as Robotic Manipulation, Computer Vision, Artificial Intelligence and Mechatronics, and the respective laboratory experiments and exercises are presented in a classical paper on the robotic and education topic, published in the IEEE Transactions on Education, in February 1996, was [9]. The paper describes a set of good and actual experiments, but they do not cover the mobile robotic area, at least the proposal work described here.

The Carnegie Mellon University (CMU), at the robotic institute, in the course of "Statistical Techniques in Robotics", probabilistic techniques are taught as a fundamental part of a robot. In other courses as: "Introduction to Robotics", "Mobile Robot Programming Laboratory", "Kinematics, Dynamic Systems and Control" the undergrad students have the opportunity to learn kinematics, vision, motion planning and collision avoidance. But, it is during the courses of "Robotic Motion planning" and "Introduction to Mobile Robots" that localization is talked and sensor-based probabilistic techniques as Kalman filter are implemented, [3].

The Massachusetts Institute of technology (MIT), the Electrical Engineering and Computer Science curriculum offers the course of "Robotics: Science and Systems I", where motion planning, kinematics, tracking and finally state estimation are some of the taught topics. These topics are applied during the course of "Robotics: Science and Systems II", [4].

In the school of Electrical Computer Engineering, in the Cornell University, the course of "Introduction to Probability and Random Signals" introduces the state estimation. In the other hand, in the course of "Autonomous Mobile Robots" topics as sensing and localization are applied during laboratory lessons, [5].

In the Europe, the Robotics is also a area of interest in the Universities. In the Imperial College University, in the Imperial's Department of Computing during the 3rd year the "Robotics" course is taught. Probabilistic methods as Monte Carlo and Kalman Filter algorithm for localization in a known map using odometry and sonar are taught and experimented at the practical lessons. At this course the undergrad students work with Mindstorms Systems NXT kits, [6].

At the ETH Zurich, Swiss Federal Institute of Technology Zurich, the Master in "Robotics, Systems and Control" offers courses as "Stochastic Systems", where the Kalman Filter and its application in the finance and engineering are theoretical taught. In courses as "Theory of Robotics & Mechatronics", "Autonomous Mobile Robots" and "Unmanned Aircraft Design, Modeling and Control", kinematics, sensor fusion, pose estimation, localization based on probabilistic methods, as Monte Carlo localization and Kalman Filter is taught with application in real scenarios, [7].

Therefore the paper described here, could be easily adapted and used during any of these courses, as a guide to realize a pedagogical experiences, where concepts of sensor fusion, navigation and probabilistic localization using Extended Kalman Filter is used, aiming to improve the undergrad learning, allowing consolidate the theoretical content.

A. Related Work

Javier Ruiz-del-Solar says in [10], social robots, like Lego Mindstorms could be used to increase the interest of young students in robotics and other areas, such as electrical systems. Javier Ruiz-del-Solar in [11] describes outreach activities using Lego Mindstorms.

Among other reasons given by the author that explain the lack of students' disinterest in engineer education programs, from which we highlight the following: 1) Students do not understand "What the engineer really is and does", 2) Absence of motivation to learn basic science as mathematics. Then, it is necessary put them facing with realistic and motivating problems using robot platforms as the Lego Mindstorms.

In the work published in the IEEE Transaction on Education, [12], the Lego Mindstorm is chosen to perform robotic experiences in an introductory course called Control Engineering, at the Federal University of Santa Catarina (UFSC), Brazil, for freshman engineers.

It is easier for students to learn when the subjects are taught in their way. In other words, learning actively leads to a higher quality impact on the students' knowledge. To ensure this quality impact, the Lego Mindstorms robots are introduced in some laboratory courses, such as [13].

A. Behrens et al. [14] believe that no matter what you're taught and with how much detail you're taught, the students learn only when the given lessons touch them, only when they are put in the engineer's role.

Also, the Lego Mindstorms makes it possible to build embedded systems without any knowledge, solve real problems, with constraints, sensor limitations, low computational power and memory, [15]. The use of Lego is, therefore, an excellent way to teach embedded systems, since it is an embedded system where the most important component, the Lego Brick, has dedicated software, inputs and output, and limitations or advantages, such as small size and low weight, low power cost and safety. Kim and Jeon [15] describe the learning of embedded systems by freshmen engineers using the Lego Mindstorm as tool.

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

The work [18] talks about a freshmen project based on Lego Mindstorms, aiming to build autonomous underwater vehicles (AUV) to perform tasks, such as surface, underwater and bottom navigation. It is an example of a non land-based robotic freshman project that uses Lego Mindstorms kits.

Other works and papers that also describe the use of Lego Mindstorms to teach programming, artificial intelligence and embedded systems to freshmen engineering students are: [19] to [25]. But any one of them covers the topic of navigation and localization based on the Extended Kalman Filter (EKF), which is the proposal and experience described in this paper.

There are extensive on-line communities, repositories and blogs that support Lego Mindstorms. Some of them are [26], [29] to [31].

Also, there are some important competitions related with the Lego Mindstorm robots, such as the RoboCup Junior Soccer (LEGO League) [32] and the Junior Dance competition during the RoboCup, [33].

The Lego Mindstorm is widely used in education and outreach activities as a motivating way to teach programming, embedded systems and artificial intelligence. However, they are only taught during the first years of engineering curricula. The authors' proposal is an experience that should teach more complex subjects, such as navigation, localization and estimation based on EKF to older undergraduate students.

III. Experience Description

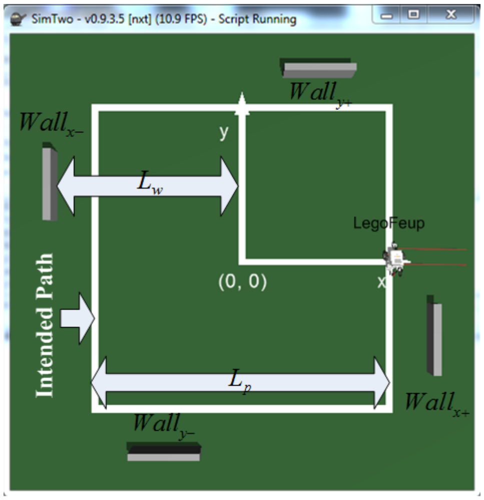
Imagine that the LegoFeup navigates inside a certain area, describing a random or a predefined trajectory (for instance square, rectangle or circle), predicting its position based on odometry, using tachometers at each wheel. The odometry leads to an error increase in the prediction without bounds.

The vehicle LegoFeup has the ability to measure the distance and orientation relatively to landmarks (for instance white walls), every time that the landmark is inside the infrared sensor range.

If white walls are placed at known positions, inside the area where the robot navigates, its position can be corrected: 1) directly, with 100% confidence ignoring the sensors noise or 2) Implementing the Extended Kalman Filter (EKF) as probabilistic method.

A. Scenario

A particular scenario was used during the experience described here, Figure 1 Scenario.. In this scenario, four walls are placed at positions that are known, forming a square. The $wall_{x+}$ and $wall_{x-}$ are placed parallel to the x axis, with L_w and $-L_w$, x coordinates, respectively. While, the $wall_{y+}$ and $wall_{y-}$ are placed parallel to the y axis, with L_w and $-L_w$, y coordinates, respectively. The trajectory that the LegoFeup is intended to follow is a square shape path with L_p of length. The white walls are placed at the first meter of each square side. The students experience includes a set of tasks, described at section V, that were performed at simulation and real (using the Lego NXT) scenarios. The simulator SimTwo [35], used in the experience, takes into account collisions between rigid bodies and considers additive noise in sensor measurements which makes it truly realistic. The test in the simulation scenario is truly important as first validation. Besides that, the simulation does not require space or a physical scenario and a real robot. This becomes possible a large number of groups performing the experience at the same time. The simulation scenario is shown at Figure 1 a) while the real scenario is presented at Figure 1 b).



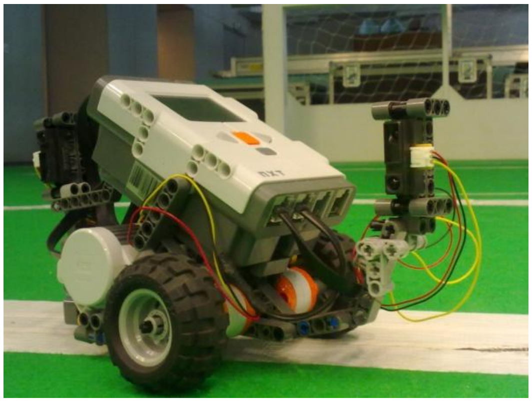
a) Simulation Scenario

b)Real Scenario

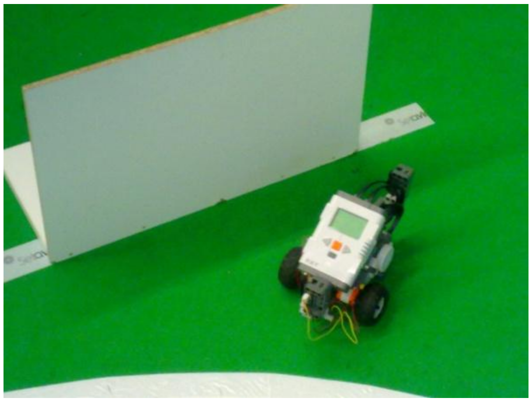
Figure 1 Scenario.

B. *LegoFeup Robot*

The Lego NXT kit is set up by three servo motors, touch, sound, ultrasound and finally, light sensor. Besides that, the kit is set up by its fundamental component, the NXT brick that allows the robot's software development, using, for instance, Labview RoboLab, Lejos JAVA for Lego Mindstorms. For more complex and longer programming tasks, as is example the implementation of a Kalman Filter, the LabView RoboLab it is not a good solution, since with few blocks the NXT memory stays full occupied. In the other hand, to implement complex algorithms, an object language oriented is preferred, as is example JAVA for Lego Mindstorms.



a) LEGO NXT: LegoFeup.



b) Real Lego and features.

Figure 2 Real scenario and LegoFeup.

The LegoFeup, shown in Figure 2, is a differential vehicle, with two traction wheels and one free wheel. Each servo motor has a tachometer, capable of measuring the number of turns of each wheel. The LegoFeup robot has two infrared sensors as well, IR Sharps [36]. The two sensors offer the system the ability to measure, besides the distance, the orientation relatively to the feature (wall).

IV. LegoFeup Loop

The LegoFeup Loop could be divided into four different modules: 1) the vehicle path Control Module, 2) the Feature Association Module, 3) the Observation Module, provided by the IR sensors, and finally, 4) the Estimation Module, possible using the Extend Kalman Filter (EKF). The entire LegoFeup loop is shown in Figure 3.

The loop inputs are the IR sensors value and the number of pulses between two consecutive time steps, Odo_1 and Odo_2. The sensor values are converted into distances, d_{s1} and d_{s2}, through the characteristic curve of the IR Sharp sensors (talked at "Observation Module").

If the distance value in both sensors is less than 30 centimetres, then the existence of an observation is considered. In the affirmative case, the observation is assigned to the correct feature (wall), using the actual estimated state, Feature Association Module. Therefore, the observation $Z(d_w, ObsAng)$ is generated by the Observation Module and passed to the Estimation Module, responsible to predict and update the vehicle state $(\hat{x}, \hat{y}, \hat{\theta})$.

Finally, the estimated vehicle's state will be used in the Control Module, which imposes the intended robot 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 perform an intended path, with knowledge on the estimated state variables, using a set of predefined routines: 1) "GoXYTheta", 2) "FollowLine", 3) "Follow Circle".

Besides the state variables, these navigation routines receive the reference linear and angular velocities V_{ref} and W_{ref} as inputs.

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

The aim of the "FollowLine" and "FollowCircle" is that the Lego NXT follows trajectories as a line or a circle (with any radius) at any direction.

In the case of the "FollowLine" routine, the other inputs are: the line that is supposed to be followed and the direction or two points of the line, where the direction will be given by the first to the second point.

The "FollowCircle" routine also has others inputs: the circle centre, the radius and the direction (clockwise or counter clockwise).

Finally, the control module implements a "FollowPath" routine, where the basic routines described above are used to perform a path. An example of path is the one intended in this work, as shown in Figure 1 a). The path has a square shape and its sides are lines in the positive and negative x and y. Therefore, the "FollowPath", in this case, only uses the "FollowLine" routine, and changes the line followed every time that a side of the square is completed.

All these routines have as outputs the linear and rotation velocity (V and w), which are afterwards converted into the velocity of each wheel using the expressions:

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

where b is the distance between wheels and V_1, V_2 the velocity of each wheel. A low level controller has the functionality of imposing the velocity at each wheel.

B. *Association Module*

If the LegoFeup passes close enough to a wall, so that the infrared sensors are available to measure the distances, a new observation could be generated. Before the observation generation, it is necessary to associate the observation to the correct wall. In the case of the experience described here, the module of the Feature Association could be summarized by the diagram shown in Figure 4.

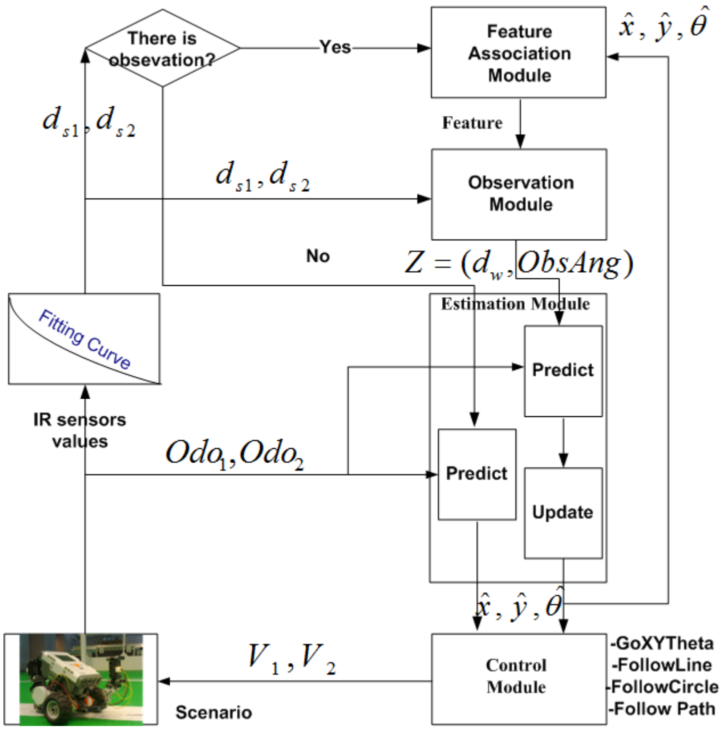


Figure 3 LegoFeup Loop.

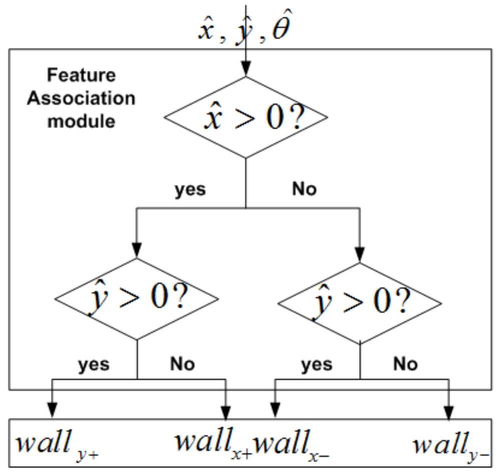


Figure 4 Feature association module.

C. Observation Module

The infrared sensors emit infrared light to the obstacle and measure the angle of the light that is reflected. It is the angle measured that makes it possible to obtain the distance of the obstacle. Some IR limitations are the influence of sun light and the obvious difficulty to operate in outdoor environments.

An example of IR sensors is the Sharp IR Range Finder, which offers a high accuracy without the influence of sunlight, ease of use, low power consumption, small, thin beam width, good range (to meters) and also, an important feature, a lower price, [36].

The IR sensor used in this work measures distances between 4 and 30 centimetres. The characteristic curve of the IR Sharp sensors is represented by the graphic shown in Figure 5, achieved empirically.

As it is possible to see, the distance value (in centimetres) depends on a non-linear way from the sensor analogical value. The best function to fit the sensor curve is given by the 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.

Figure 5 shows the real acquired values and the result of best relation that makes it possible to compute the distance, measuring the analogical Sharp's value. The best relation was obtained 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 $ObsAng$ relatively to the feature wall. The observation module is given by the following equations, deduced through Figure 6:

$$obsAng = atan\left(\frac{d_{s1} - d_{s2}}{L}\right), \qquad d_w = \frac{d_{s1} + d_{s2}}{2} \cdot c(ObsAng) \tag{3}$$

Due the maximum range of the infrared sensors, every time the distance measured in both the IR sensors is less than 30 centimetres, the existence of an observation is considered.

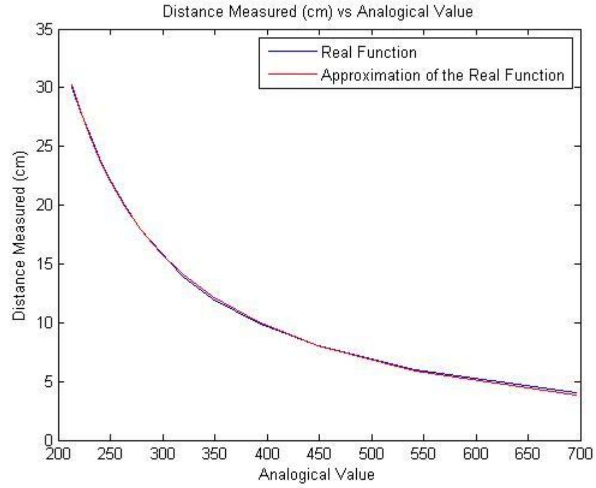


Figure 5 Real and Fitting curve.

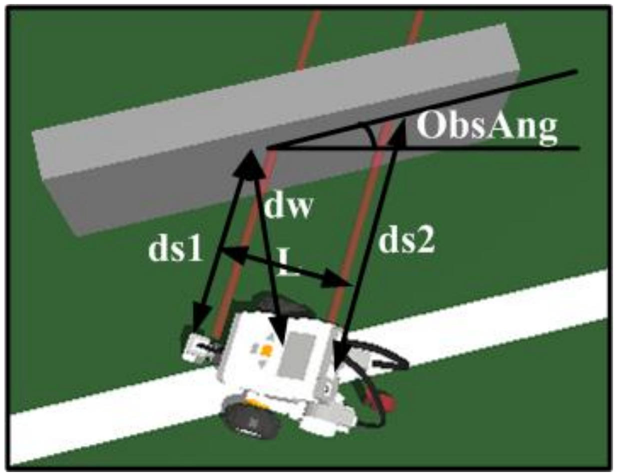


Figure 6 Angle observed ($ObsAng$) and distance to the wall (d_w).

At each wall only two states could be observed with the measurement of the wall distance. That way, the observation model, $Z = [x \; or \; y \quad \theta]^T$, will be given by:

$$Z(wall_{x+}) = \begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} L_w - d_w \\ \frac{\pi}{2} - ObsAng \end{bmatrix}, \qquad Z(wall_{x-}) = \begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} -(L_w - d_w) \\ -\frac{\pi}{2} + ObsAng \end{bmatrix} \tag{4}$$

$$Z\left(wall_{y+}\right) = \begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} L_w - d_w \\ \pi + ObsAng \end{bmatrix}, Z\left(wall_{y-}\right) = \begin{bmatrix} y \\ \theta \end{bmatrix} = \begin{bmatrix} -(L_w - d_w) \\ ObsAng \end{bmatrix} \tag{5}$$

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

D. Estimation Module: Extended Kalman Filter

The Extended Kalman Filter (EKF) is a Kalman Filter estimator, where the non-linear 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, is set up by two steps that work in a cycle. First, we predict the vehicle state and covariance (EKF prediction). Then, if there is a new observation, it is associated to the correct feature and the vehicle state and covariance are corrected (EKF update).The prediction and update step is done according to the equations described in the following subsections.

Algorithm 1 Extended Kalman Filter.

$P_0 \leftarrow Initial\ Covariance$
$X_0 \leftarrow Initial\ State$
$for\ i \leftarrow 1\ to\ MaxDteps\ do$
$EKF\ Prediction$
$[X_k \quad P_k] \leftarrow Predict\ By\ Odometry$
$New\ Observations?$
$[Z_k] \leftarrow Measures$
$Data\ Association: Assiciate\ to\ the\ correct\ Feature - Wall$
$EKF\ Update$
$[X_k \quad P_k] \leftarrow Update\ (Kalman\ Gain, Innovation)$
$endfor$

a. Prediction Step

The Kinematic of the vehicle could 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), 2D position and orientation, respectively. The vehicle forward speed is represented by V and the rotation speed by ω.

$c(.)$ and $s(.)$ represents $cos(.)$ and $sin(.)$. This nomenclature will be used in the entire paper. In discrete time, the dynamics could be described by the kinematic 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 two steps:

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

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

The tachometer gives, for each wheel, the number of pulses between two continuous steps. This input, $U = [Odo_1 \quad Odo_2]$, makes it possible to estimate the travelled distance of each wheel using the following expressions:

$$d_1 = \frac{Odo_1}{Codo_1}, \qquad d_2 = \frac{Odo_2}{Codo_2} \tag{9}$$

where $Codo_1$ and $Codo_2$ is the constant of odometry in each wheel, i.e. the number of encoder pulses per meter.

The error in the odometry appears in the measure of d and $\Delta\theta$, resulting in error of d_1, d_2 and b.

That way, the real kinematic of the vehicle (entering in account with the measurement errors), could be written as $X(k+1) = f(X(k), U, q)$, where the non-linear function f is given by:

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

where $q = [\varepsilon_d \quad \varepsilon_{\Delta\theta}]$. Considering 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:

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

The estimated covariance is equal to, see chapter "Nonlinear Estimation" on [1].

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

where $\nabla f_X$is the dynamics gradient in order to estimate state $\hat{X}$, equal to $\frac{\partial f}{\partial X}$. While ∇f_q is the gradient in order to the Gaussian noise q, considering the mean noise, by definition equal to zero, is $\frac{\partial f}{\partial q}$.

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

The predicted covariance depends on the dynamics and previous covariance $\nabla f_X P(k) \nabla f_X^T$. The other term increases the covariance and then the uncertainty until any observation arrives. This term, $\nabla f_q Q \nabla f_q^T$ is related with the odometry noise, approximated by Gaussian noise.

b. *Update Step*

The state update is carried out during every cycle that 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 V the innovation. The innovation in the $wall_{x+/-}$ and $wall_{y+/-}$, respectively, is equal to, see equations (4) and (5).

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

The Extended Kalman Filter Gain is equal to, see chapter "Nonlinear Estimation" on [1]:

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

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

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

when the walls seen are $wall_{x+/-}$, then $C_1 = 1$, $C_2 = 0$. In the case of $wall_{y+/-}$, then $C_1 = 0$, $C_2 = 1$. Defining:

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

$$a = \frac{1}{2} c(ObsAng), \qquad b = -\frac{d_{s1} + d_{s2}}{2} s(ObsAng) \frac{dObsAng}{dd_{s1}} \quad (19)$$

The gradient of the observation for the noise r, $\frac{\partial h}{\partial r} = \nabla h_r$, is equal to:

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

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

The covariance update is performed taking into account the following equation, see chapter "Nonlinear Estimation" on [1].

$$P(k) = [I - W(k)\nabla h_X]\nabla h_X^T \, P(k) \quad (21)$$

V. LegoFeup Communication

Both scenarios, real and simulation, are prepared to exchange information with a host computer where it is supposed the EKF algorithm run. The exchanged information are: 1) In the scenario to the EKF algorithm direction, the vehicle sensors measures (the tachometer values of each wheel - odometry and the infrared distances); 2) in the contrary direction, from the EKF algorithm to the scenario, the velocity of each wheel calculated by the EKF algorithm is sent, V_1 and V_2.

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

VI. AVAILABLE INTERFACE

The available interface, as shown in Figure 8 and Figure 9Figure 9 is a tool that 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; finally 6) the vehicle reference velocities V_{ref} and ω_{ref}.

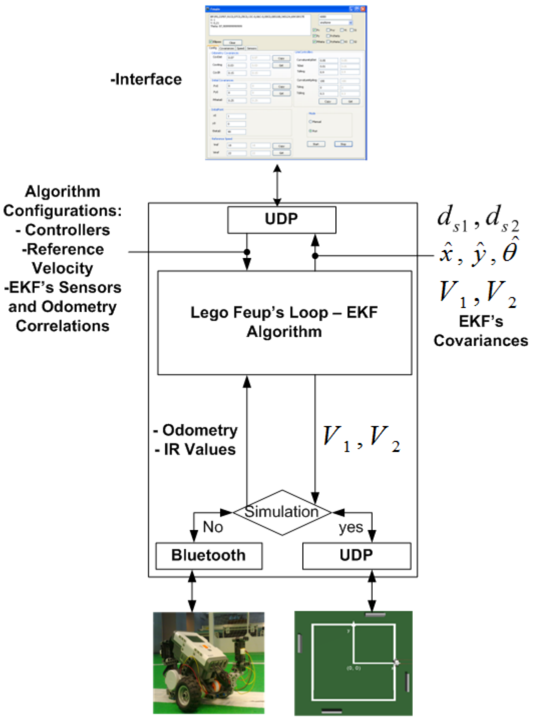


Figure 7 Algorithm, communication and interface.

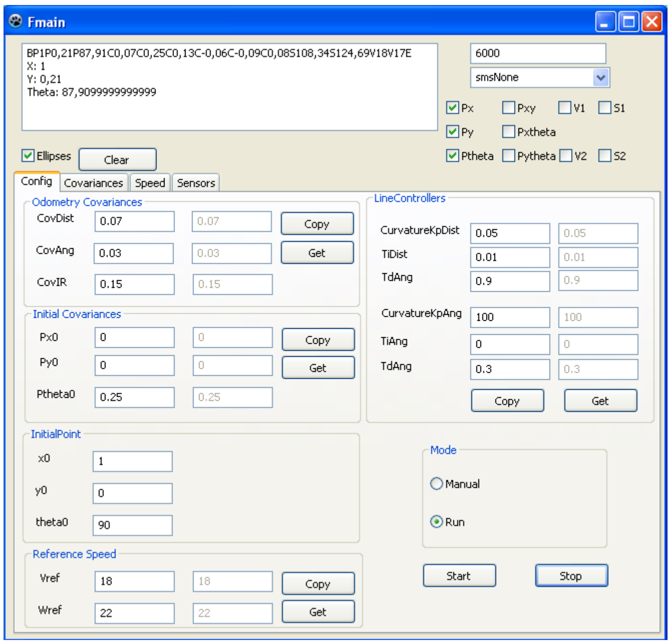
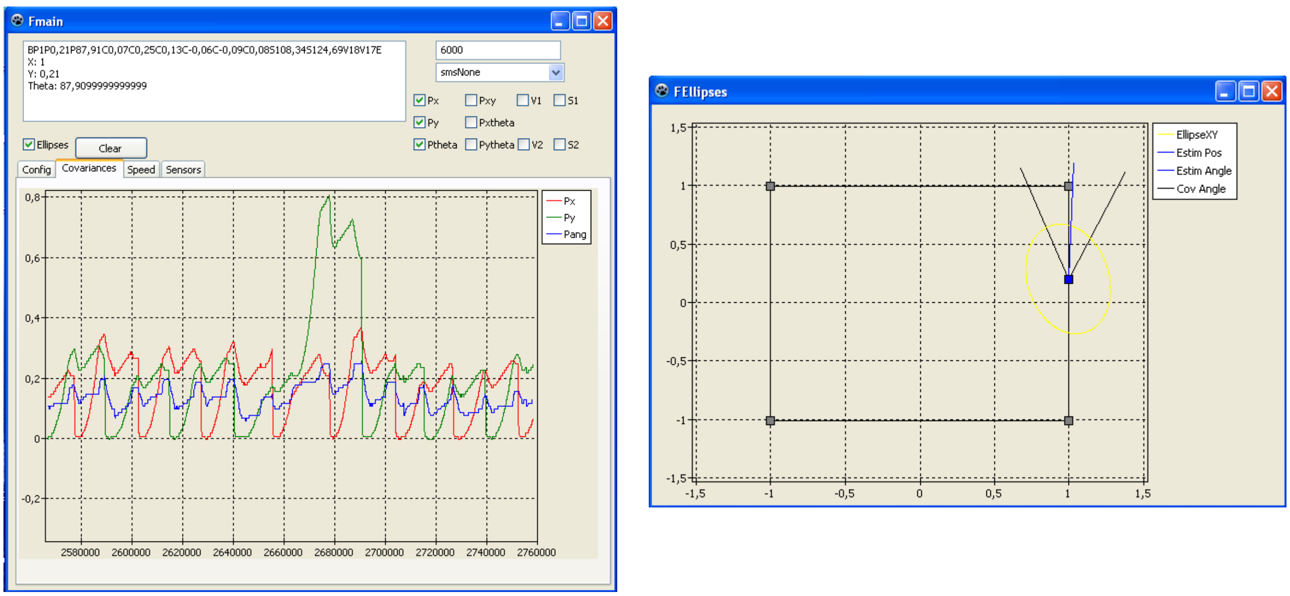


Figure 8 Main Form. Configuration Tab.

Aiming for a better understanding of EKF concepts for the undergraduate students, the interface was developed in order to show the Extended Kalman Filter variables: 1) covariance matrix, 2) covariance ellipse, 3) vehicle's estimated position. The interface also allows to see the vehicle's wheels velocities V_1 and V_2 and the infrared sensors values, d_{s1} and d_{s2}. This interface communicates with the LegoFeup loop algorithm via UDP.

The form, called "FEllipses" (Figure 9 b)) shows the intended path, the vehicle estimated position and finally the vehicle estimated orientation. In this form, it is also possible to see the covariance ellipse with a confidence of 68,2% (one time the standard deviation). The user interface allows the undergraduate students to tune the filter, seeing at the same time the effect on the Extended Kalman Filter covariance variables. Therefore, this module provides students the possibility to learn about the EKF operation and the importance of the covariance matrices of the odometry and sensors noise, in the quality operation of the filter.



a) Main Form. In the present main form, the covariance P_x, P_y and P_θ are plotted.

b) Main Form. In the present main form, the covariance P_x, P_y and P_θ are plotted.

Figure 9 Available Interface.

VII. TEACHING METHOD

The difference between a good and bad learning of any topic is the comparison between the practical results and theoretical knowledge. When the theoretical concepts work in the practice the knowledge becomes much more solid.

Enter in account with all the objectives talked above, section Contribute in the students learning, the objective of the Mobile Robots course is not all the process of implementation of the algorithm that runs the localization based in the EKF, since it could be an hard and delay task, which does not returns with the immediate and intended learning.

Therefore, the whole LegoFeup tool was built and implemented by the authors: 1) the LegoFeupLoop using Java Programming Language, 2) the communication with the simulation scenario or the real scenario, (UDP or Bluetooth) using Java Programming Language as well, 3) the simulation (SimTwo) and real scenario and finally 5) the interface GUI that allows the students perform a set of tasks as described below, section Experience Guide.

The great advantage of the LegoFeup tool in the learning of the undergrad students is: allow change the estimation and control modules parameters, as described in section Available Interface, and obtain immediate results, which should be helpful to truly understand the EKF applied to a mobile robot problem.

A. Lessons Organization

The topic "Localization of Mobile Robots using an Extended Kalman Filter in a Lego NXT" was taught in five lessons each of them with three hours. Both the first and second lessons have had a theoretical content. These two lessons compose the theoretical module of the topic presented: 1) During the first theoretical lesson, the theory behind the Kalman Filter, such as its mathematical formulation was presented. 2) In the second theoretical lesson the experience applied to the LegoFeup was explained. All the modules of the LegoFeup loop, section IV, were presented in detail.

The 3th, 4th and 5th lessons compose the practical module. The practical lessons were performed by two different classes in different days. One class is composed by twenty three undergrad students, while the other is composed by twenty six. The experience is performed by groups of two students. The practical module is divided as follows: 1) during the first practical lesson the objective is the student familiarization with the available tool, LegoFeup tool, and with the simulation scenario. 2) At the second practical lesson, after a deeply knowledge about the available algorithm\tool, an experience in the simulation scenario was performed, based on the guide presented at section: Experience Guide. 3) the last and 5th lesson has the finality of test the configuration of the EKF done in the simulation scenario, lesson 2 of the practical module, but now in the real scenario. The test was performed a group at a time.

B. Experience Guide

The experience is constituted by the following steps: a) Execute a square trajectory, based only on the odometry. b) Execute the same trajectory with direct localization, sensors covariance matrix R with very low values, based in the IR Sharp Measures without noise. c) Add Gaussian noise with variance equal to 0.01 in the IR Sharp sensors readings and repeat the previous step. d) Put the EKF's sensors covariance matrix R equal to the injected in the simulation scenario. Give a very low value to the odometry covariance matrix Q. Realize the same trajectory. e) Change now the value of the covariance matrix Q to a very high value and repeat the previous step. f) Using the results of the previous steps tune the filter to the best solution Q and R.

Instructions about how to perform each step were given to the undergrad students. Each group delivered a report, at the end of the lesson, explaining the vehicle behaviour in each step. Besides this explanation it is requested a conclusion and justification.

Some conclusions and solutions should be achieved by the students at each step: a) the vehicle covariance should increase without bounds due the absence of observation. The vehicle never updates and corrects its position with the walls, and then it will be lost rapidly. b) The algorithm as a lot of confidence in the infrared sensors, then the actualization is done directly with 100% of confidence. At this case the algorithm localizes the vehicle with success but only because the scenario is ideal, i.e. the IR Sharps measures are obtained without noise. c) The algorithm as a lot of confidence in the infrared sensors, then the actualization is done directly with 100% of confidence. But in this case the algorithm does not localize the vehicle correctly, due the fact that in this step the scenario is not ideal, i.e. the infrared sensors have noise. d) In this case, the filter should not work, since the sensors measures are rejected due the confidence in the odometry is high. Then the localization is done based only in the odometry. e) In this step the simulated situation is similar to the step b). Now, instead a high confidence in the sensors, we have a low confidence in the odometry compared with the confidence that we have in the infrared sensors. This situation leads the filter to update the vehicle localization directly, with a confidence near of 100% when a wall is observed.

C. Course Webpage

Videos and theoretical content about the implementation of the experience described here can be found in [37]. Also an application of the EKF algorithm could be downloaded, which some tests could be done in the simulation scenario. The theoretical and the Experience Guide content taught during the theoretical lesson is available.

VIII. MOBILE ROBOTS COURSE RESULTS

After a questionnaire it was possible conclude the importance and success that the Mobile Robot Course was in the students learning. The large majority of them heard about Kalman Filters, but never implemented and did not understand the Kalman Filter operation or finality. Before the course, also the large majority reached the final objective, understanding the Extended Kalman Filter and the localization based on the EKF as a probabilistic method to make the fusion between the odometry and the IR Sharps observations.

Using a student's survey, questions as: "Do you are now more interested in the mobile robotics topic than before the course?", "Do you think that the experience realized and the tutorial given will be helpful in future works inside the mobile robotics topic?" or finally "Do you understand and feel now capable to describe the method of navigation and localization implemented?", had in the majority of the cases really positives answers (near 90%).

Theory Module		Pratical Module		
1st Lesson 3Hours	2nd Lesson 3Hours	3rd Lesson 3Hours	4th Lesson 3Hours	5th Lesson 3Hours
EKF	EKF in Mobile Robotic	LegoFeup Tool	Experience Simulation Scenario	Experience Real Scenario

Figure 10 Lessons Organizations

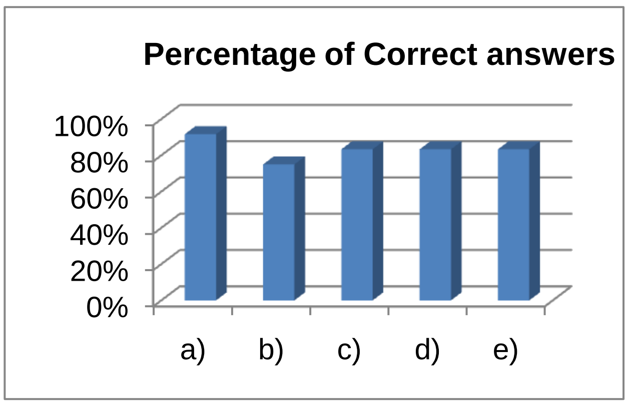


Figure 11 Percentage of correct answers plotted.

The report delivered by each group (49 students) at the end of the practical lessons was evaluated by the authors. The results allowed conclude the success that the experience have had during the practical lessons. According the experience guide presented at section VII, subsection B, the reports have had a percentage of satisfactory answers, in each experience step a), b), c), d) and e), as shown in Figure 11.

At the end of the semester, the undergrad students realized the course final exam. The exam had three questions about the experience realized and the extended Kalman filter applied in the robot localization. The average of correct answers in the three questions was superior to 79%. The topics evaluated in the others questions are taught without the teaching method described here. The results in these questions had a significant descent. The average of correct answers was 61%, which is the proof that the method presented improves the students learning.

IX. Conclusions

The intention with this paper was to give undergraduate students a kind of tutorial to perform experiences related with navigation and localization concepts, namely odometry and the Extended Kalman Filter (EKF). The experience described could be carried out on mobile robots courses.

The Lego NXT has a large use in the educational environment, but only to teach basic and easier concepts. The authors' proposal aims to be applied in more complex subjects such as the EKF estimator.

The undergraduate students should learn the importance of using a probabilistic method to estimate the vehicle position and filter the environment information, such as the EKF. They should also achieve basic EKF concepts: the EKF's cycle (predict and update) and the importance of odometry and sensor covariance, Q and R, in the correct operation of the EKF.

So, an improved learning of navigation and localization concepts using EKF is expected during undergraduate courses of robotics, as it was during the Mobile Robots class.

REFERENCES

[1] A. Gelb, J. F. Kasper Jr., R. A. Nash Jr., C.F. Price and A.A. Sutherland Jr., " Apllied Optimal Estimation", The MIT Press, Massachssets & London, 2001.

[2] Sebastian Thrun, Wolfram Burgard and Dieter Fox, "Probabilistic Robotics", The MIT Press, September 2005.

[3] "CMU: robotic courses", available at: " http://www.ri.cmu.edu/ri_static_content.html?menu_id=276", February, 2011.

[4] "MIT: Electrical Engineering and Computer Science" available at: " http://student.mit.edu/catalog/m6a.html", February, 2011.

[5] "Cornell University: In the school of Electrical Computer Engineering" available at: "http://www.ece.cornell.edu/", February, 2011.

[6] "Imperial College University, Department of Computing, Robotics course", available at: "http://www.doc.ic.ac.uk/teaching/coursedetails/333", February, 2011.

[7] " ETH Zurich : Master in Robotics, Systems and Control " ", available at: "http://www.master-robotics.ethz.ch", February, 2011.

[8] 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", Proceedings of IEEE, vol.83, no. 9, September, 1995.

[9] Eric Krotkov, "Robotics Laboratory Exercises", IEEE Transactions on education, vol.39, no. 1, February, 1996.

[10] Javier Ruiz-del-Solar, "Robotics-Centered Outreach Activities: An Integrated Approach", IEEE Transactions On Education,vol.53,no.1,February 2010.

[11] Javier Ruiz-del-Solar and Roberto Avilés, "Robotics Courses for Children as a Motivation Tool: The Chilean Experience", IEEE Transactions On Education, vol.47, no.4, November 2004.

[12] Marcos B. R. Vallim, Jean-Marie Farines and José E. R. Cury, "Practicing Engineering in a Freshman Introductory Course", IEEE Transactions On Education, vol.49, no.1, February 2006.

[13] Andrew B. Williams, "The quality Impact of Using Lego Mindstorms Robots to Teach Computer Engineering", IEEE Transactions On Education, vol.49, no.1, February 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 Transactions On Education, vol.53, no.2, May 2010.

[15] Seung Han Kim and Jae Wook Jeon, "Introduction to Embedded Systems Using Lego Mindstorms", IEEE Transactions On Education, vol.52, no.1, February 2009.

[16] Shekhar Sharad, "Introducing Embedded Design Concepts to Freshmen and Sophomore Engineering Students with LEGO MINDSTORMS NXT", 2007 IEEE International Conference on Microelectronic Systems Education,(MSE'07), 2007.

[17] N. Z. Azlan, F.Zainudin,H. M. Yusuf,S. F. Toha,S. Z. S. Yusoff, N. H. Osman, "Fuzzy Logic Controlled Miniature LEGO Robot for Undergraduate Training System", 2007 Second IEEE Conference on Industrial Electronics and Applications, 2007.

[18] David Ye, Ilya Brutman, Gunter W. Georgi and Lorcan M. Folan, "Freshman Project: Autonomous Underwater Vehicle (AUV)", American Society for Engineering Education, 2007.

[19] Seung Han Kim and Jae Wook Jeon, "Educating C Language Using LEGO Mindstorms Robotic Invention System 2.0", Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida - May 2006.

[20] Daniel C. Cliburn, "Experiences with the LEGO Mindstorms throughout the Undergraduate Computer Science Curriculum", 36th ASEE/IEEE Frontiers in Education Conference.

[21] Martha Isela Garduno Mota, "Work In Progress - Using Lego Mindstorms and Robolab as A Mean To Lowering Dropout and Failure Rate In Programming Course", 37th ASEE/IEEE Frontiers in Education Conference, October 10 – 13, 2007, Milwaukee.

[22] Mario A. Garcia and Holly Patterson Mc-Neill, "Learning How To Develop Software Using The Toy LEGO Mindstorms", ITiCSE'02, June 24-26, 2002, Aarhus, Denmark.

[23] Abhijit Nagchaudhuri, Gurbax Singh, Manpreet Kaur, Sudhakar George, "Lego Robotics Products Boost student Creativity in Pre-College Programes at Umes", 32nd ASEE/IEEE Frontiers in Education Conference,November 6 - 9, 2002, Boston.

[24] Frank Klassner and Scott D. Anderson, "Lego Mindstorms: Not Just for K-12 Anymore", IEEE Robotics & Automation Magazine, June 2003.

[25] Ciarán Mc Goldrick, Meriel Huggard, "Peer Learning with Lego Mindstorms",34th ASEE/IEEE Frontiers in Education Conference, October 20 – 23, 2004, Savannah.

[26] "Lego: Mindstorms", available at: "http://www.lego.com", July, 2010.

[27] Holly Patterson-McNeill and Carol L. Binkerd, "Resources for using Lego Mindstorms", 2001 by the Consortium for Computing in Small Colleges, CCSC: South Central Conference, 2001.

[28] "Mindstorms NXT Faqs", available at: "http://www.active-robots.com/products/lego/mindstorms/MINDSTORMS NXT FAQs.pdf", July 2010.

[29] "Lego Mindstorms", available at: "http://www.legomindstorms.com", July, 2010.

[30] "The NXT STEP - LEGO MINDSTORMS NXT Blog", available at: "http://www.thenxtstep.blogspot.com/", July 2010.

[31] "Lugnet", available at: "http://www.lugnet.com/", July 2010.

[32] "RoboCup Junior", available at: "http://rcj.robocup.org/", July 2010.

[33] "RoboCup 2010 at Singapore", available at: "http://www.robocup2010.org/", July 2010.

[34] "Lejos, Java for Lego Mindstorms", available at: "http://lejos.sourceforge.net/", July, 2010.

[35] Paulo Costa, "Paco-Wiki: SimTwo", available at: "http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo", July, 2010.

[36] "IR Range Finder. Society of Robots", available at: "www.societyofrobots.com/sensors_sharpirrange.shtml", June, 2010.
[37] "Lego NXT— EKF Webpage",available at: "http://paginas.fe.up.pt/~miguelp/LegoNxt", June, 2010.

BIOGRAPHY

Miguel Pinto (IEEE member M'09) was born in Caracas, Venezuela in 1986. He has Portuguese nationality. Graduated with MSc degree in Electrical Engineering at the University of Porto, Portugal in 2009. After receiving the MSc, he worked as an invited assistant at the Department of Electrical Engineering and Computers in the Faculty of Engineering of University of Porto. Since 2009, he has been a Ph.D. student at the Electrical Engineering in the University of Porto, developing his research within the Robotic and Intelligent Systems department of INESC (National Institute of Engineering and Computer Science) of Porto. His main research interest is in Process Control and Robotics, navigation and localization of autonomous vehicles, including terrain and marine robotics.

A. Paulo Moreira (IEEE member M'09) graduated with a degree in Electrical Engineering at the University of Oporto, in 1986. Then, he pursued graduate studies at the Oporto University, obtaining a MSc degree in Electrical Engineering - Systems in 1991, and a Ph.D. degree in Electrical Engineering in 1998. From 1986 to 1998 he also worked as an assistant lecturer in the Electrical Engineering Department of the Oporto University. Presently he is a Lecturer in Electrical Engineering, developing his research within the Robotic and Intelligent Systems department of INESC (National Institute of Engineering and Computer Science) of Porto. His main research interest is Process Control and Robotics.

Aníbal Matos (IEEE member M'08) received the Licenciatura, M.Sc. and Ph.D. in Electrical and Computer Engineering from Porto University in 1991, 1994, and 2001, respectively. Currently, he is an assistant professor at the Electrical and Computer Engineering Department of Porto University, and also a researcher at the Robotics and Intelligent Systems Group of Institute for Systems and Computer Engineering of Porto (INESC Porto). His research interests include modeling, navigation and control of autonomous vehicles, nonlinear control systems, and marine robotics.