Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Design and Implementation of Kalman Filters applied to Lego NXT based Robots

Ángel Valera, Marina Vallés, Leonardo Marín, Pedro Albertos

Department of Systems Engineering and Control
Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de València, Valencia, Spain
P.O.Box 22012 E-46071. Tel. 34-96-3879570 Fax 34-96-3879579
e-mail: {giuprog, mvalles}@isa.upv.es, leomarpa@posgrado.upv.es, pedro@aii.upv.es

Abstract: As it is well known, the Kalman filter (KF) provides an efficient computational tool to estimate the state of a process, minimizing the mean squared error. By allowing the estimation of past, present and future states, even dealing with some uncertainty in the process model, the KF presents a very good solution in state estimation. This work will deal with the design and the implementation of this powerful and important tool for the control engineering students in an easy to set-up laboratory environment. For that purpose the LEGO NTX robot is proposed. With this motivating tool the students can implement different estimators, like a velocity and acceleration observer of the robot wheels, or an observer of the position and orientation of the robot, and analyze different alternatives and solutions.

1. INTRODUCTION

The Kalman filter (KF) is a powerful solution for the case of controlling systems that could be time varying and have several inputs and outputs, minimizing a quadratic function of the states and the control signals and where the states as well as the measurements have Gaussian disturbances. In this case, the KF acts as an optimal state estimator that minimizes the variance of the state estimation error. The principles about its design are broadly described in different books and scientific papers, like (Bozic, 1994) (Brown and Hwang, 1996), (Julier and Uhlmann, 1997), (Grewal and Andrews, 2001), (Simon, 2006), (Welch and Bishop, 2007).

However, despite the KF is a very effective and useful estimator for a large class of problems, sometimes it is difficult for the students to understand and apply it and to know how to design it.

In general, in control engineering, practical work is of high importance to understand the complex coherences between system-structure, controller types and technical realization problems as filtering, discretization and delays (Valera et al, 2005). But, in this particular case, the practical work results very illustrative.

In order to help the students in this goal, there are a lot of complete laboratory experiments of all thinkable types sold by companies or built themselves by many universities. However these experiments use to be quite expensive and, so, the number of students per experiment ratio uses to be high.

This paper proposes the use of LEGO Mindstorms as a platform for doing practical work with KF. In this sense some applications are presented where the use of the KF is very suitable for the proposed problems.

A brief description of the KF theory is presented in the next section. In section 3, the KF equations are algorithmically described for being implemented in a mobile robot. In section 4, the implementation of different KF for improving the robot measurements are shown. Finally, some conclusions are drafted in the last section.

2. THE KALMAN FILTER THEORY

As it is well known, the KF provides an efficient computational tool to estimate the state of a process, in a way that minimizes the mean squared state estimation error. There are several variations of the filter (see, for instance http://www.cs.unc.edu/~welch/kalman/), but this work is focused in the application of two of them: first the KF in its standard formulation for a linear system, and the Extended Kalman Filter (EKF) which is the extension of the KF to nonlinear processes.

Consider the *linear, time invariant,* discrete time system with noise:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \quad (1)$$

where $x \in \Re^n$ is the state vector, $u \in \Re^u$ is the input vector and $z \in \Re^m$ is the measurement vector. w_k, v_k represent process and measurement noises at time k. They are considered to have an independent, white, normal probability distribution with zero mean. The exact values of w_k and v_k at time k are normally unknown but it is assumed the knowledge of the covariance matrices, Q_k and R_k respectively (assumed to be constant).

The KF goal is to estimate the state x_k of (1) based on the knowledge of the system dynamics (linear model), the disturbance characteristics (covariance matrices) and the availability of the noisy measurements z_k. This filter will minimize the effects of w_k and v_k in the state estimation. The operation of the KF is shown in figure 1. It is defined $\hat{x}_k^- \in \Re^n$ as the *a priori* state estimate at time k (which is obtained from the knowledge of the process prior to time k),

Copyright by the
International Federation of Automatic Control (IFAC)

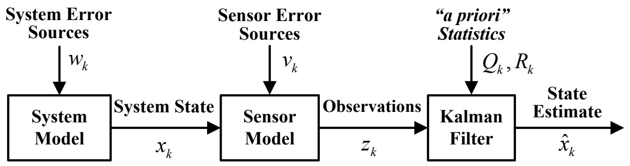


Fig. 1. Kalman filter operation (Grewal 2001).

and $\hat{x}_k \in \Re^n$ as the *a posteriori* estate estimate (at time k using the measurement z_k). The a priori and a posteriori estimate errors are defined as:

$$e_k^- = x_k - \hat{x}_k^-; \qquad e_k = x_k - \hat{x}_k \tag{2}$$

The a priori and a posteriori estimates of the error covariance are expressed in (4) (where the operator E represents the expected, or average, value).

$$P_k^- = E\left[e_k^- e_k^{-T}\right] \qquad P_k = E\left[e_k e_k^T\right] \tag{3}$$

The a posteriori state estimate $\hat{x}_k$ is expressed as a linear combination of the a priori estimate $\hat{x}_k^-$ and the estimated error, which is the difference between the actual measurement z_k and the measurement prediction $H\hat{x}_k^-$:

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H \cdot \hat{x}_k^-\right) \tag{4}$$

The term $\hat{z}_k - H \cdot \hat{x}_k^-$ is called the innovation or residual which reflects the discrepancy between measurements and predicted measurements to update the a priori estimated state. Here the gain matrix K is chosen to minimize the a posteriori error covariance P_k and it is determined by a probabilistic approach using error and noise covariances. Just in order to better understand the KF application, an outline of the Kalman gain derivation is sketched (a complete development can be seen, for instance, in Simon 2006 and Bozic 1994). The minimization is done by substituting (4) in e_k (2), and then the result is substituted in P_k (3). As already mentioned, minimizing (4) with respect to K the optimal filter gain K is obtained as:

$$K_k = P_k^- H^T \left(H\, P_k^- H^T + R\right)^{-1} \tag{5}$$

This equation needs the value for the *a priori covariance* P_k^-, this value is obtained from the definition in (3), (which requires the knowledge of the system variable x_k). By simple manipulations, the following results:

$$P_k^- = A P_{k-1} A^T + Q \tag{6}$$

Again the value of the a posteriori covariance, P_k, is needed, this is obtained also from its definition in (4):

$$P_k = \left(I - K_k H\right) P_k^- \tag{7}$$

With the equations (5) to (8) the KF is implemented in an algorithm . This can be classified in two stages (Welch 2007):

- *Time update* (or Predictor): It calculates the next *a priori* estimate of the state based on the previous estimate of the state and the current value of the input (normally using an initial guess to start the calculation); this is used to calculate the a priori covariance.

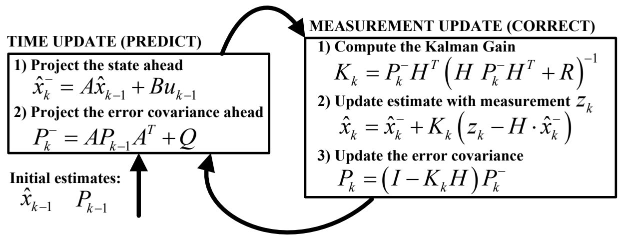


Fig. 2. Kalman filter algorithm (Welch 2007).

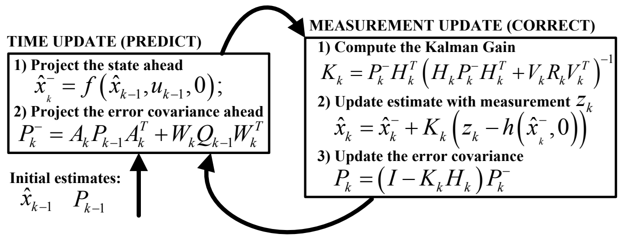


Fig. 3. Extended Kalman filter algorithm (Welch 2007).

- *Measurements update* (or Corrector): It uses the current measurement to refine the result given by the previous stage to obtaining an improved a posteriori estimate.

The algorithm is shown in the figure 2.

Now, the procedure to derive the EKF is also summarized. When dealing with non-linear processes instead of (2) the process model is expressed by:

$$x_k = f\left(x_{k-1}, u_{k-1}, w_{k-1}\right); \qquad z_k = h\left(x_k, v_k\right) \tag{8}$$

with w_k and v_k unknown. The state and the measurements can be approximated by:

$$\hat{x}_k^- = f\left(\hat{x}_{k-1}, u_{k-1}, 0\right); \qquad \hat{z}_k^- = h\left(\hat{x}_k^-, 0\right) \tag{9}$$

With $\hat{x}_{k-1}$ being the *a posteriori* state estimate from a previous time step *k-1*. In order to use the filter equations already developed for the linear KF, the model (8) is linearized around the operating point, such that:

$$\begin{aligned} x_k &\approx \hat{x}_k^- + A\left(x_{k-1} - \hat{x}_{k-1}\right) + W w_{k-1} \\ z_k &\approx \hat{z}_k^- + H\left(x_k - \hat{x}_k^-\right) + V v_k \end{aligned} \tag{10}$$

with $\hat{x}_k^-$, $\hat{z}_k^-$ being the estimates from (9). The system matrices of (10) are calculated every time k by:

$$\begin{aligned} A_k &= \frac{\partial f}{\partial x}\Big|\hat{x}_{k-1}, u_{k-1}, 0 \qquad & W_k &= \frac{\partial f}{\partial w}\Big|\hat{x}_{k-1}, u_{k-1}, 0 \\ H_k &= \frac{\partial h}{\partial x}\Big|\hat{x}_k^-, 0 & V_k &= \frac{\partial h}{\partial v}\Big|\hat{x}_k^-, 0 \end{aligned} \tag{11}$$

The EKF gain is calculated in a similar way as the KF (a complete development is shown in Simon 2006 and Bozic 1994), and also it can be written in algorithm form as shown in the figure 3.

3. APPLYING THE KF TO THE LEGO NTX

3.1 Introduction

In the last years, for high school and university courses, robotics is becoming successful due to a combination of factors (Weinberg and Yu, 2003). The students are very motivated with robotics because they can physically experiment its work.

Despite that usually the educational material for process control is very expensive, it is possible to buy different economical robot platforms (ActivMedia's Pionner robot, MIT's HandyBoard and Cricket controller cards, The LEGO Group's LEGO Mindstorms, etc.). These platforms usually consist of controllers, electronic sensors, low cost mechanical systems and/or small robots. These ones don't allow as much precision as an industrial robot but they are enough for the educational processes. Moreover, these platforms are very interesting for promoting the team work due to its multidisciplinary nature. So, each member of the team can be in charge of the development of a specific part of the work, as the physical development of the robot, programming its different subsystems, developing the strategies and response planning, etc.

In this work, the LEGO Mindstorms NXT platform will be used. It is the evolution of the first version developed in 1998 in collaboration with LEGO and the MIT (Resnick *et al.*, 1996), (Papert, 2000). The LEGO platform is popular in docent and research works. As well as the extension for the hardware and software of this system in different ways (Baum, 2000), (Baum *et al.*, 2000), different works based on this platform have been published in special editions of journals such as IEEE Robotics and Automation Magazine (Weinberg and Yu, 2003), (Klassner and Anderson, 2003), (Greenwald and Kopena, 2003) or IEEE Control Systems Magazine (Gawthrop and McGookin, 2004).

On January of 2006 the next generation, LEGO Mindstorms NXT, was introduced on the International Consumer Electronics Show. Besides other minor changes on the electronic sensors and the construction pieces, the new version incorporates a new control unit: the NXT. It is based on a powerful 32-bits microcontroller: ARM7, with 256 Kbytes FLASH and 64 Kbytes of RAM. For programming and communications, NXT has an USB 2.0 port and a wireless Bluetooth class II, V2.0 device. The new control unit has 4 inputs (one of them provides an IEC 61158 Type 4/EN 50 170 expansion for future use) and 3 analog outputs.

The new LEGO version offers 4 electronic sensor types: touch, light, sound and distance. In addition of these basic sensors, nowadays a great variety of optional sensors can be bought, as for example vision cameras, magnetic compass, accelerometers, gyroscopes, infrareds searchers, etc. (http://www.hitechnic.com/, http://www.mindsensors.com/).

The last most significant LEGO components are the actuators. In this case, they consist on dc motors that incorporate encoders sensors integrated. These sensors have a 1-degree resolution, which improves the previous version encoders' resolution. A more detailed description about LEGO Mindstorms NXT motors can be found in http://www.philohome.com/motors/motorcomp.htm).

For this work a mobile robot with differential wheels configuration has been developed using the basic set of the LEGO NXT, 2 accelerometers and one gyroscope according to the experiment. Its Kinematic model represents the evolution of the robot's velocities in a fixed inertial frame. The robot's position is defined by the position of $P_0 = (x,y)$ and the heading angle θ in the global reference frame (see figure 4).

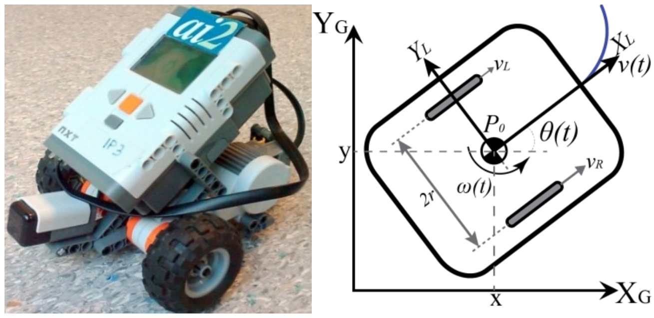


Fig. 4. LEGO based mobile robot used in this work.

By knowing the local linear and angular velocities (v and ω) the global position is defined by:

$$\begin{bmatrix} \cdot \\ \cdot \\ \dot{\iota} \end{bmatrix} \quad \begin{bmatrix} & \theta) & 0 \\ & \theta) & 0 \\ & & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{12}$$

From this equation the position and orientation of the robot can be estimated using the odometry equation:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} v_{k-1} T_s \cos\left(\theta_{k-1} + \omega_{k-1} \cdot 0,5T_s\right) \\ v_{k-1} T_s \sin\left(\theta_{k-1} + \omega_{k-1} \cdot 0,5T_s\right) \\ T_s \omega_{k-1} \end{bmatrix} \tag{13}$$

where T_s is the sampling time (50ms in all the experiments). The kinematic relation between v and ω and the linear velocities in the wheels v_L and v_R is:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ -1/(2r) & 1/(2r) \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix}, \quad \begin{bmatrix} v_L \\ v_R \end{bmatrix} = \frac{1}{T_s} \begin{bmatrix} \Delta s_L \\ \Delta s_R \end{bmatrix} \tag{14}$$

where $\Delta s_L = EncL_k - EncL_{k-1}$ and $\Delta s_R = EncR_k - EncR_{k-1}$ are the change in the encoders values of the wheels left and right. Using these models 3 experiments are proposed. For the LEGO encoder sensors there is a lot of noise in the values of the wheels velocity measurements. As the velocity values of left and right wheel are needed to be known to estimate the robot position in the global plane using the odometry equation (13), the error in the velocity measurements (14) will affect the position estimation; this will cause an unwanted behaviour when controlling the robot's position and movement. To avoid this problem, the Kalman filter will be used to estimate the wheel velocity using the information of the encoder measurements and a *linear* model of the wheels velocity as the first experiment. Using two accelerometers and the encoders, the KF will be used to make an estimation of the velocities of change in the position and orientation of the robot (v and ω) as the second experiment. Finally the third experiment will be implemented using the extended Kalman filter (EKF) with the *nonlinear* kinematic model (13), two accelerometers and a gyroscope to improve the position estimation (this filter also allow fusing other global sensor information, for example a camera or a GPS to improve the estimation of the real position of the robot). The performance of these experiments in estimating the global position and heading of the robot is compared with the measurements of a zenithal camera for a square trajectory.

3.2 Kalman Filter to improve encoder measurements

In this case the Kalman filter will be applied to each wheel encoder (position, $\theta_{L,R}$) to obtain a good estimation of each of the wheels velocities ($\omega_{L,R}$). These velocities will be used to estimate the global position an orientation of the robot using the odometry equation (14). Using $\omega(t)=\dot{\theta}(t)$, consider for each wheel the next state definition:

$$x_1(t)=\theta(t) \qquad (15)$$
$$x_2(t)=\omega(t)$$

The linear model of the wheel velocity will be:

$$\dot{x}(t)=Ax(t)+w(t), \qquad A=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} \qquad (16)$$

Discretizing (16) the following equation is obtained:

$$x_k=e^{A\cdot T_s}x_{k-1}+w_{k-1} \qquad (17)$$

Considering that only the encoder measurements of the wheel are accessible, next equation is obtained:

$$z_k=Cx_{k-1}+v_{k-1}, \qquad c=\begin{bmatrix}1 & 0\end{bmatrix} \qquad (18)$$

The encoder measurements have to be pre-processed to be used in the estimation; this is done by measuring the wheels diameter in an accurate way. An experimental procedure to do this is proposed by (Michel and Rohrer, 2009) and it is used to pre-process the encoders before using the KF.

To implement the Kalman filter algorithm the process Q and measurements R covariance matrix are needed. The matrix Q defines the accuracy of the model to describe the real process, the diagonal terms are the variance of each state and the off diagonal terms are the covariance between states. A diagonal matrix is chosen (no correlation between states) as an initial guess and its terms are modified as needed to improve the estimation. A higher diagonal value indicates that the model is less taken into account to do the estimation of the corresponding state.

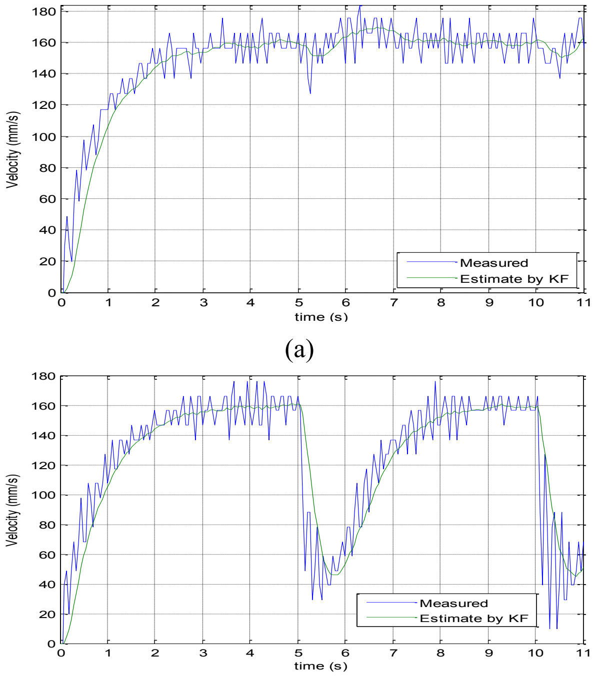
The matrix R is obtained by measuring the sensor covariance. This is obtained from an initial test with the robot, i.e. when the robot moves a short distance and stops, measuring the encoder values for some time after it stops. These values are used to calculate de covariance of the encoder when it measures the same position for a several time. A similar test can be used to obtain the covariance of any sensors as needed by recording their values when the robot moves in a straight line (measuring the same sensor value for a several time). The covariance of the sensors can also be manually adjusted to improve the estimation of the state. Lower values indicate that the sensor has a high resolution and the filter will base the estimation mostly in this measurement (the model is less taken into account). A big value makes the filter trust more the model or other sensors (with lower values) than the measurement of the sensor with the big covariance value.

Finally the filter needs the initial state estimation $\hat{x}_{k-1,ini}$ and the initial error covariance $P_{k-1,ini}$. The initial state is normally known because the initial encoder values and the wheel velocity are cero in most cases (robot starts moving from repose). The initial error covariance is defined as a diagonal matrix with its terms greater than cero. These terms indicates the accuracy of the initial state and should be carefully chosen to make the filter converge. For the LEGO robot the values are low since the initial position and velocity of the wheels are known.

These initial assumptions should be carefully done to make the filter converge. The values used in the filter are:

$$R_{\text{right\_wheel}}=2.5; \quad R_{\text{left\_wheel}}=1.5$$
$$Q=\begin{bmatrix}0.1 & 0\\ 0 & 10\end{bmatrix}; \quad x_{k\text{-}1,ini}=\begin{bmatrix}0\\ 0\end{bmatrix}; \quad P_{k\text{-}1,ini}=10\cdot \mathrm{I}_{2x2} \qquad (19)$$

where **I** is de identity matrix. Using (15) to (19) in the Kalman filter equations (figure 2) the wheels velocity estimation is done by implementing the algorithm in the on-board microcontroller of the LEGO NTX. This is shown in figure 5 as an example for the right and left wheel.



(a) (b)

Fig. 5. Right (a) and left (b) wheel velocity estimation improvement using KF

3.3 Kalman Filter to improve linear and angular velocities

In this case the Kalman filter will be applied to estimate the linear and angular velocity of the robot (v, ω) using the equation (14) as measurements. These velocities will be used in some of the trajectory control algorithms and in (13) to obtain the global position of the robot. Consider the follow state definition:

$$\underline{x}(t) = \begin{bmatrix} v(t) & \omega(t) \end{bmatrix}^T \tag{20}$$

Using the linear and angular accelerations (a, α) obtained by differentiating (14) and using two Hitechnic accelerometers to measure the wheels linear accelerations, is defined the model that predicts the velocities in (21). The accelerometer data is pre-processed to remove its bias and change its units to S.I.

$$\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix} + \begin{bmatrix} T_s & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} a_{k-1} \\ \alpha_{k-1} \end{bmatrix} + w_{k-1} \tag{21}$$

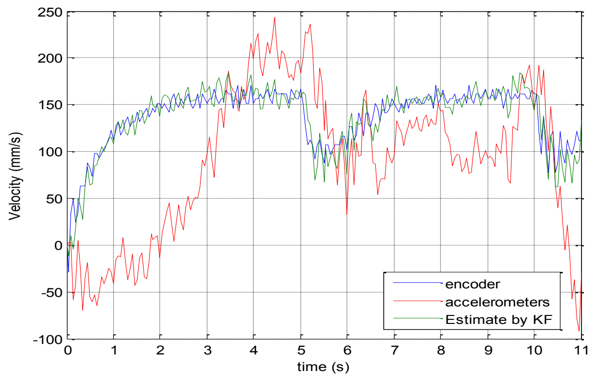
Using (21) with (14) (v, ω from the encoders) allows the estimation of v and ω using two different types of sensors the encoders for the measurement equation (22) and accelerometers as inputs for the model (21).

$$z_k = Cx_{k-1} + v_{k-1}, \qquad c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{22}$$

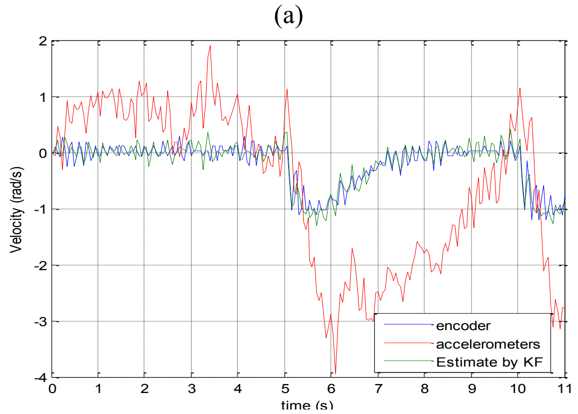
Following the same criterion used in the first experiment, the variables needed by the KF are defined:

$$R = \begin{bmatrix} 5.1 & 0 \\ 0 & 1.938 \end{bmatrix};\ Q = \begin{bmatrix} 2.2 & 0 \\ 0 & 2.2 \end{bmatrix};\ x_{k\text{-}1,ini} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};\ P_{k\text{-}1,ini} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{23}$$

Using (20) to (23) in the Kalman filter equations (figure 2) the robot velocity estimation can be improved by implementing the algorithm in the on-board microcontroller of the LEGO NTX. This is shown in figure 6(a) for the linear velocity and in the figure 6(b) for the angular velocity.



(a)



(b)

Fig. 6. Robot's linear (a) and angular (b) velocity estimation using KF with two accelerometers.

3.4 Extended Kalman Filter to improve robot's position and orientation

In this case, the Extended Kalman filter will be applied to improve the estimation of the position and heading of the robot in the global reference frame. The estimation model is obtained using (13) and adding (21) to use as inputs the accelerations as the previous case. The model is:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ v_{k-1} \\ \omega_{k-1} \end{bmatrix} + \begin{bmatrix} v_{k-1}T_s \cos(\theta_{k-1} + \omega_{k-1} \cdot 0{,}5T_s) \\ v_{k-1}T_s \sin(\theta_{k-1} + \omega_{k-1} \cdot 0{,}5T_s) \\ T_s\omega_{k-1} \\ T_s a_{k-1} \\ T_s \alpha_{k-1} \end{bmatrix} + w_{k-1} \tag{24}$$

The measurements are obtained from the encoders using (14) to obtain v but integrating ω to measure θ. Also a Hitechnic gyroscope is used to measure ω and integrate it to obtain θ (its data is also pre-processed to remove the bias and change the units to S.I.). The measurement equation (25) is used to fuse both sensors using their respective covariance values (26) and improve the estimation of the heading angle.

$$z_k = Cx_{k-1} + v_{k-1}, \qquad c = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{25}$$

Using (24) and (25) the position and heading estimation is done using three different types of sensors. Following the same criterion used in the first and second experiment, the variables needed by the KF are defined:

$$R = \begin{bmatrix} 5.1 & 0 & 0 \\ 0 & 0.938 & 0 \\ 0 & 0 & 0.332 \end{bmatrix};\ Q = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix};\ x_{k\text{-}1,ini} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix};\ P_{k\text{-}1,ini} = 0.05 \cdot \mathrm{I}_{5x5} \tag{26}$$

Using (24) to (26) in the Extended Kalman filter equations (figure 3) the robot position estimation can be improved by implementing the algorithm in the on-board microcontroller of the LEGO NTX. This is shown in figure 7 also with the results from the previous cases and the raw encoder data.

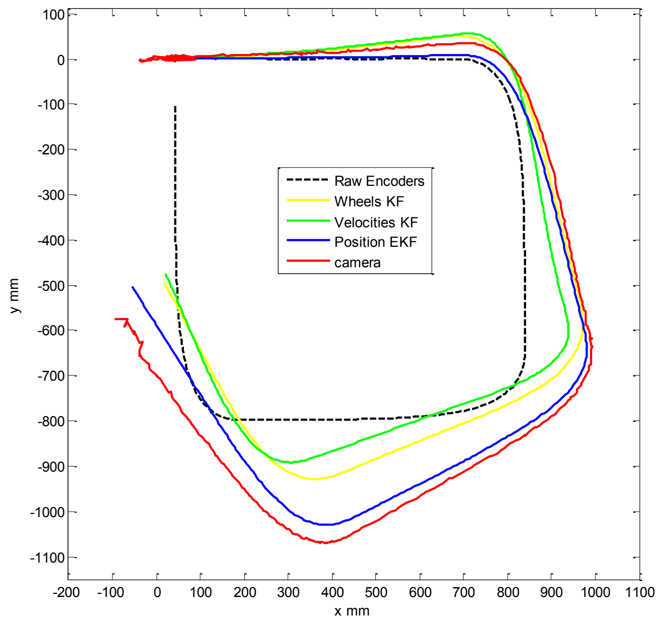


Fig. 7. Robot's positions of all the experiments.

5. CONCLUSIONS

The suitability of LEGO Mindstorms robots as a platform for designing and implementing Kalman filters have been showed through different activities where the measurements (encoders, position, orientation) have been improved using them.

Different kinds of Kalman filters have been proposed (KF and EKF) and for different measurements obtaining different performance results. The results from the experiments were compared with the real position obtained with a zenithal camera. From figure 7 is shown that any of the filters implemented improves the estimation of the position compared to the raw encoder data. According the filters use more information available from the different sensors they improves more the position estimation. A more advanced activity could be to include the robot position given by the camera into the EKM for improving the estimation so it would be more close to the real position.

If only few LEGO robots are available for the students, the sensors data can be exported from the robot to a text file and then be processed offline in a computer with Matlab or similar software. The camera data can be also provided and processed to give the real trajectory of the robot and used to compare the performance of the implemented filters. In this case the implementation task is simpler than the online case because the matrix operations needed by the filter are done in a single line as described in algorithms form figures 2 and 3. In the online case the matrix operations are done element by element and could be harder to implement by the students.

Further filter implementations can be used as needed to instruct the students in other subjects like highly nonlinear systems using the "unscented" Kalman Filter or sensor data fusing using the extended an unscented Kalman filter along with measurements from different kind of sensors like a digital compass, accelerometers, 3D gyroscope and a zenithal camera all in one filter implementation.

A web-based repository with the Kalman filters implementations, videos and photographs can be found in: http://wks.gii.upv.es/sidireli/node/8

ACKNOWLEDGEMENTS

This work has been partially funded by FEDER-CICYT projects with references DPI2008-06737-C02-01 and DPI2010-20814-C02-02 financed by Ministerio de Ciencia e Innovación (Spain). Also, the financial support from the University of Costa Rica is greatly appreciated.

REFERENCES

ActivMedia Robotics Web site [Online]. Available: http://www.activrobots.com.

Baum, D., (2000). Definitive Guide to LEGO Mindstorms. Berkley, Ca: Apress.

Baum, D., M. Gasperi, R.Hempel, and L. Villa (2000). Extreme Mindstorms. Berkley, CA: Apress.

Bozic, S. M., (1994). Digital and Kalman filtering: an introduction to discrete-time filtering and optimum linear estimation. Halsted Press.

Brown, R. G., and P. Y. C. Hwang (1996). Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions (Third ed.): Wiley & Sons, Inc.

Gawthrop, P., and E. McGookin (2004). A LEGO-Based Control Experiment. IEEE Contr. Syst. Mag., 24, 5, pp. 43-56.

Grewal, M. S., and A.P. Andrews (2001). Kalman Filtering: Theory and Practice Using Matlab. John Wiley & Sons

Hitechnic home page [Online]. Available: http://www.hitechnic.com/

Julier, S.J., J.K. Uhlmann (1997). A New Extension of the Kalman Filter to nonlinear Systems. In: *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*. SPIE

Klassner, F., and S.D. Anderson, "LEGO mindstorms; Not just K-12 anymore", IEEE Robot. Automat. Mag., vol. 10, no. 2, pp. 12-18, 2003.

LEGO Mindstorms home page [Online]. Available: http://mindstorms.lego.com/

LEGO Motors compared characteristics [Online]. Available: http://www.philohome.com/motors/motorcomp.htm

Mindsensors home page [Online]. Available: http://www.mindsensors.com/

Michel, O., F. Rohrer, N. Heiniger, and wikibooks contributors, Cyberbotics'Robot Curriculum, 2009. http://en.wikibooks.org/wiki/Cyberbotics\' Robot Curriculum

Papert, S., (2000). What's the big idea? Towards a pedagogy of idea power, *IBM Syst. J.*, **39**, **3&4**, pp. 720-729.

Resnick, M., F. Martin, R. Sargent, and B. Silverman, (1996). Programmable bricks: Toys to think with, *IBM Syst. J.*, **35**, **3&4**, pp. 443-452.

Simon, D. (2006). Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. John Wiley & Sons

The Handy Board home page [Online]. Available:

http://handyboard.com/

Valera, A., J.L. Díez, M. Vallés, and P. Albertos (2005). Virtual and Remote Control Laboratory Development, *IEEE Contr. Syst. Mag*., **25, 1**, pp. 35-39.

Weinberg, J.B., and X. Yu (2003). Robotics in Education: Low-Cost Platforms for Teaching Integrated Systems. *IEEE Robot. Automat. Mag.* **10, 2**, pp. 4-5.

Welch, G., and G. Bishop (2007). *An Introduction to the Kalman Filter*. UNC-Chapel Hill. [Online]: Available: http://www.cs.unc.edu/~welch/kalman/