Fuzzy Logic Based Nonlinear Kalman Filter Applied to Mobile Robots Modelling

Rodrigo Carrasco Sch.
Department of Electrical Engineering
Pontificia Universidad Católica de Chile, CHILE
E-mail: rax@ing.puc.cl

Aldo Cipriano Z.
Department of Electrical Engineering
Pontificia Universidad Católica de Chile, CHILE
E-mail: aciprian@ing.puc.cl

***Abstract* – In order to reduce the false alarms in fault detection systems for mobile robots, accurate state estimation is needed. Through this work, a new method for localization of a mobile robot is presented. First, a Takagi – Sugeno fuzzy model of a mobile robot is determined, which is optimized using genetic algorithms, creating a precise representation of the kinematic equations of the robot. Then, the fuzzy model is used to design a new extension of the Kalman filter, based on several linear Kalman filters. Finally, the fuzzy filter is compared to the conventional extended Kalman filter, showing an improvement over the estimation made. The fuzzy filter also presents advantages in implementation, due to the fact that the covariance matrices needed are easier to estimate, increasing the estimation frequency.**

I. INTRODUCTION

Over the last few years new commercially available robots have appeared in the market, giving a technological solution to problems that range from soldering in production lines and space exploration to more mundane issues such as vacuum cleaning or grass cutting. As more complex tasks are being addressed, more sophisticated control, navigation and monitoring systems are needed to accomplish these tasks efficiently. This is especially true for mobile robots, where the design of every subsystem is a challenge, as it must work with a dynamic environment, using limited resources.

One basic requirement of mobile robotics is to accurately pinpoint the position and direction of the robot, also known as its posture. The simplest way to achieve this is to use a technique called dead-reckoning which integrates odometric measures over time to estimate the relative posture. The problem with this method, as with other relative position systems, is that the errors are cumulative, resulting in a divergence of the estimation. These errors can range from simple measuring discordances in the model or sensors, to wheel slippage. Tests such as the UMBmark designed by Borestein [1], can be applied to determine the extent of the systematic errors in the model, compensate the estimation and thus reduce the error, but these cannot compensate for external errors such as wheel slippage.

The use of absolute position systems such as GPS is a common practice to improve the estimation, reducing the cumulative error, but adding an additional complexity to the robot. These measurements are generally fused together with the relative position measurements, improving the accuracy of the estimation.

Since its appearance in 1960, the Kalman filter (KF) has become a very useful tool to reduce the effect of Gaussian white noise and fuse together measurements within a linear system [2]. Several works have extended the use of this filter to non-linear models, such as mobile robots, improving the results of the estimations [3]. The extended Kalman filter (EKF) is applied using a linearization of the system on an operating point. The main difficulty is that the covariance matrices needed, must be precisely determined to get optimum results, but this is seldom achieved in non-linear systems.

As fuzzy logic is an excellent tool for working with nonlinearities, it has been used to determine dynamically the covariance matrices, depending on the system state, thus reducing the divergence [4], [5].

This work first presents a fuzzy model of the kinematic equations that describe a mobile robot, based on the Takagi – Sugeno fuzzy structure, which is then optimized using genetic algorithms. Finally, a new extension of the Kalman filter is implemented using fuzzy logic and the optimized model is then compared to the EKF.

II. ROBOT MODEL

For simplification purposes the body of the robot is considered as a circular disc of radius $\boldsymbol{b}$, with two independent wheels of radius $\mathbf{r}$ each. As figure 1 shows, the angular velocity of the right wheel is considered to be ω_1, whereas ω_2 describes the velocity of the left one.

The posture of the robot is defined as a three element vector $\boldsymbol{X}=[x\ y\ \varphi]^{\mathrm{T}}$, which contains the absolute position of the robot in some reference plane, and the direction angle.

Using simple geometric relations, the linear and angular velocities of the robot can be related to the angular velocity of the wheels as equations 1 and 2 describe:

$$V_k = \frac{r}{4}\left[\left(\omega_{1_{k-1}} + \omega_{2_{k-1}}\right) + \left(\omega_{1_k} + \omega_{2_k}\right)\right] \qquad (1)$$

$$\dot{\varphi}_k = \frac{r}{4b}\left[\left(\omega_{1_{k-1}} - \omega_{2_{k-1}}\right) + \left(\omega_{1_k} - \omega_{2_k}\right)\right] \qquad (2)$$

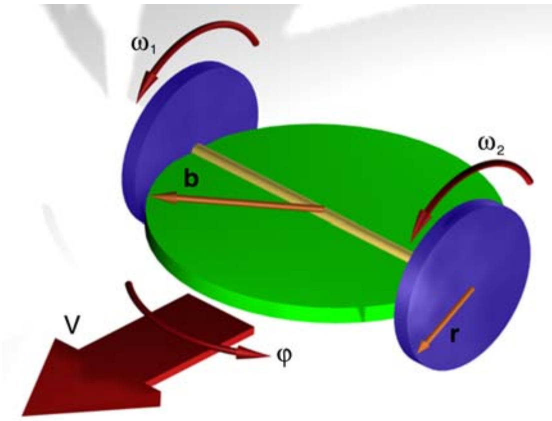

Fig. 1. Mobile robot model and basic parameters.

As a means of improving the accuracy of the estimation, considering that discrete time equations are used, the velocity for time k is calculated as the average between the values at times k-1 and k. Integrating these two values over time, the kinematic relations for this robot can be obtained [6]. These will determine the complete posture vector using the angular velocity measured on each wheel, as equations 3, 4, and 5 show:

$$x_k = x_{k-1} + V_k \Delta T \cos(\varphi_k) \tag{3}$$

$$y_k = y_{k-1} + V_k \Delta T \sin(\varphi_k) \tag{4}$$

$$\varphi_k = \varphi_{k-1} + \dot{\varphi}_k \Delta T \tag{5}$$

To deal with the localization issue, the test mobile robot will be equipped with only two different sensors. Encoders are used on each wheel to determine its speed, allowing relative localization and posture determination. A magnetic digital compass is also used, to obtain an absolute reading of the direction angle, as a way of reducing the cumulative error of relative positioning. Both sensors are considered as ideal sensors, with a white Gaussian noise added, of standard deviation σ_e for the encoders and σ_c for the compass.

III. TAKAGI – SUGENO FUZZY MODEL

The main objective of this work is to create a more accurate modelling scheme that can allow a reliable estimation of the posture of the robot.

The first step is to create an ideal model that does not consider the effect of noise and external errors. Although this can be done by using equations 3, 4, and 5, these are nonlinear equations, limiting the usefulness of statistical optimization tools such as the Kalman filter.

The Takagi – Sugeno Fuzzy Structure (TSFS) allows the use of multiple linear models of a system, and fuses them together using fuzzy logic [7], [8]. As for mobile robots there is no single operating point, the nonlinear equations must be linearized in several different ones, covering the whole spectrum of possibilities. For this work, the equations are linearized in four different angle values: 0, π/2, -π/2, and π. Each linear model, L_n, has the following structure, where φ_0 is the linearization angle:

$$x_{Ln_k} = x_{Ln_{k-1}} + V_k \Delta T \left[\cos(\varphi_0) - \sin(\varphi_0)(\varphi - \varphi_0)\right] \tag{6}$$

$$y_{Ln_k} = y_{Ln_{k-1}} + V_k \Delta T \left[\sin(\varphi_0) + \cos(\varphi_0)(\varphi - \varphi_0)\right] \tag{7}$$

Due to the fact that φ can span from π to -π, there is a discontinuity in φ when the robot rotates. This raises an important issue, because the linearization about π will only be valid for values near it, whereas it will be a very bad approximation for φ=-π, although π and -π are actually the same angle of direction. To solve this, a new linearization about -π is added, which is used for negative angle values, whereas the linearization about π is used for positive ones.

As two variables need to be modelled, two independent TSFS are used, one for x and one for y, using the same variable, φ, as input for fuzzification. The output of the fuzzy model is given by the linear combinations of the outputs of each linear model. Equations 8 and 9 describe the output of each fuzzy model, where x_{Ln} and y_{Ln} are the values estimated for x and y by linear model n, whereas μ_n and ν_n are coefficients determined by the membership function (MF) associated to the lineal model n.

$$x_k = \sum_{n=1}^{5} \mu_n x_{Ln_k} \tag{8}$$

$$y_k = \sum_{n=1}^{5} \nu_n y_{Ln_k} \tag{9}$$

Once the structure is determined, the next step is to design the membership functions associated with each linear model, and that will determine the values for μ_n and ν_n. Because each linearization represents exactly the behaviour of the system at the linearization point, the output of the associated MF at that value must be 1, whereas the rest of the membership functions must have a value of 0. Also, it is reasonable to conclude that if the input value is between two linearization points, the output will consider only the two associated linear models, which implies that the rest of the membership functions must have a value of 0.

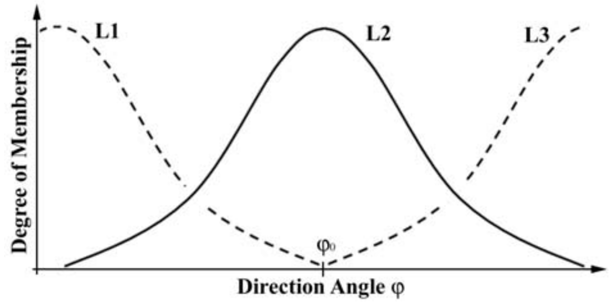

Fig. 2. Membership function example.

Figure 2 shows an approximate description of what each MF should be. A critical variable in this model is the form of each MF, because this will give the nonlinearity in the resulting fuzzy model, and thus it will combine the linear models correctly.

IV. Queen-Bee Genetic Optimization

A. Method Description

In order to improve the output of the fuzzy model each MF must be optimized. One method that has proved to return good results is genetic optimization, which is able to modify the morphology of each MF and with it, increase the efficiency of the model [9].

Genetic optimization is based on the same principles used in evolution theories, where a group of possible solutions or population is combined to create a new child population, whose solutions may be better. For doing this each solution must be first coded. The cycle of testing and combining solutions is repeated for a certain number of generations, hoping to find a better solution at the end. More information about the methods for coding and combining can be found on [7].

The combination is dependant on how effective a solution is, also known as its fitness. A new method of doing this is based on bee societies, in which only one individual (the queen) is able to reproduce and create the new generation, reducing the computation needed for the combination step [10]. This means that the best solution is used to generate the new set, simplifying the process.

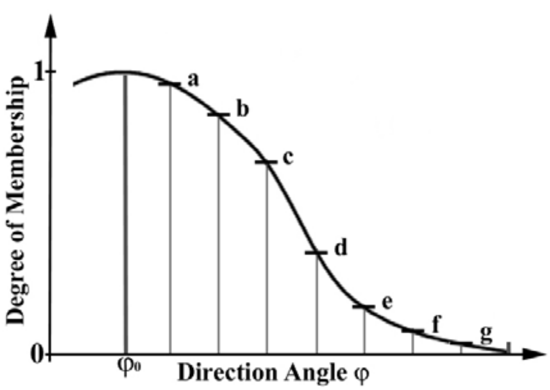

Fig. 3. Membership function coding.

B. Genetic Optimization Implementation

To reduce the number of free variables in the optimization scheme, each MF is designed symmetrical, with a maximum value of 1 at the associated linearization point and with a value of 0 at the adjacent linearization points.

With these parameters, each MF is coded as a 7 element vector: $MF_n=[a_n\ b_n\ c_n\ d_n\ e_n\ f_n\ g_n]^T$, containing the height of the membership function at constant intervals as shown in figure 3. Each element of the vector has values between 0 and 1. The values in each interval are determined as Sinc interpolation of both interval limits.

As five linearizations are used to design the fuzzy model, there are five different membership functions, each with its coding vector, creating a 5x7 matrix. Two of these matrices describe every individual, one for the *X* position fusion, and one for the *Y* position.

The first step of the algorithm is to test each individual to determine its fitness. This is done by calculating the absolute difference between the path described by the robot equations and the path described by the fuzzy model, as equation 10 shows. This is done for several random trajectories and then averaged, to eliminate the possibility that the solution is only valid for a single trajectory. The solution with the smallest fitness is considered to be the best solution.

$$fitness = \sum_{k=1}^{n} \left| Pos_k - FuzzyPos_k \right| \tag{10}$$

Once the best solution is found, the combination is done by averaging the matrices of each member of the population, with the matrix that describes the queen, using a certain combination probability, P_C. Elitism is used to keep the best solution in the population, thus assuring that after the generations pass the best solution encountered is equal to or better than the original best solution.

Finally, a mutation step is used to reduce the possibility that the algorithm converges to a local minimum. This is done by changing some elements of an individual randomly, with a certain mutation probability, P_M.

C. Results

The optimization was done using an initial population of 30 individuals, P_C=95% and P_M=5%. Several different trajectories were used to test the fitness of the solutions.

The resulting membership functions, after simulating for 100 generations, are shown in figure 4 for the *X* position and in figure 5 for the *Y* position estimates. The simulation of the fuzzy model and the robot model for several different trajectories show, that the estimation of the fuzzy model is very accurate.

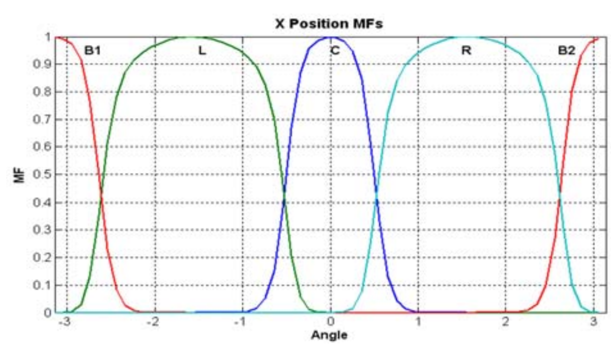

Fig. 4. X position membership functions.

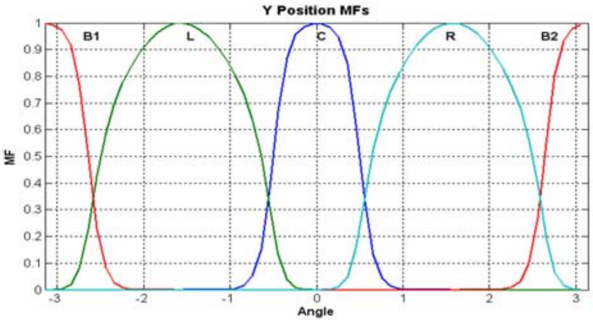

Fig. 5. Y position membership functions.

For both variables the five membership functions are: Centre (C), associated to φ=0; Right (R), associated to φ=-π/2; Left (L), associated to φ=π/2; Back 1 and 2 (B1 and B2), associated to -π and π respectively.

V. Kalman Filter Design

If noise is added to the sensor readings in the previous model, the distance between the estimation of the fuzzy system and the real position increases over time.

There are several ways to improve the estimation. The use of sensor fusion to take advantage of information redundancy or the EKF to reduce noise on the estimation, are commonly used tools for dealing with the problem. The EKF has shown very good results in reducing the estimation error, but several problems appear during implementation, such as the determination of the covariance matrices that represent the noise in the system.

The KF is an optimal filter that minimizes the mean error of the estimation of a linear system, such as the one on equation 11, whereas the EKF is applied to nonlinear systems, such as the one described on equation 12.

$$\begin{aligned} X_k &= AX_{k-1} + BU_{k-1} + \varepsilon_1 \\ Z_k &= CX_k + \varepsilon_2 \end{aligned} \tag{11}$$

$$\begin{aligned} X_k &= f\left(X_{k-1}, U_{k-1}\right) + \varepsilon_1 \\ Z_k &= h\left(X_k\right) + \varepsilon_2 \end{aligned} \tag{12}$$

In equations 11 and 12 X_k represents the system state at time k, Z_k the observation, and U_k the applied input. The system and measurement Gaussian white noise are represented by ε_1 and ε_2 respectively.

The KF algorithm first estimates the state of the system at time k using the previous information and the system equations. Along with this, the covariance matrix P_k is also estimated using the system error covariance matrix Q:

$$X_k^- = A\hat{X}_{k-1} + BU_{k-1} \tag{13}$$

$$Z_k^- = CX_k^- \tag{14}$$

$$P_k^- = A\hat{P}_{k-1}A^T + Q \tag{15}$$

Using the measurement error covariance matrix R, the Kalman gain is then obtained. This gain indicates how reliable the measurement is, with respect to the estimation given by the system equations:

$$K_k = P_k^- C^T \left(CP_k^- C^T + R\right)^{-1} \tag{16}$$

Finally, the estimations are corrected using the Kalman gain and the difference between the estimation and the measurement.

$$\hat{X}_k = X_k^- + K_k\left(Z_k - Z_k^-\right) \tag{17}$$

$$\hat{P}_k = \left(I - K_k C\right)P_k^- \tag{18}$$

The equations for the EKF are very similar, with the difference that the system matrices A, B, and C must be calculated every time step by a linearization of the system equations shown in equation 12 [11]:

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial X_{k-1[j]}}\left(\hat{X}_{k-1}, U_{k-1}\right) \tag{19}$$

$$B_{[i,j]} = \frac{\partial f_{[i]}}{\partial U_{k-1[j]}}\left(\hat{X}_{k-1}, U_{k-1}\right) \tag{20}$$

$$C_{[i,j]} = \frac{\partial h_{[i]}}{\partial X_{k[j]}}\left(X_k^-\right) \tag{21}$$

VI. Fuzzy Extended Kalman Filter

The fuzzy model implemented before, is composed of several linear models, which allow the use of a linear Kalman filter on each of them to improve the estimation, creating a fuzzy observer [12]. The first advantage of doing this is that the estimation results are optimal in reducing the noise error for each model. The determination of the covariance matrices can also be an advantage. Generally it is easier to determine the covariance matrices of the KF than the ones of the EKF, especially because in the EKF the system equations are nonlinear, resulting in a poor estimation of their value.

As every linear model has its own set of equations, there are different covariance matrices Q, depending on the direction of movement. This fact gives an advantage over the EKF, because each model has its own set of parameters, returning an optimal estimation in that sense.

The R covariance matrix is the same for all models, as the only measured variable is the absolute angle using the magnetic compass:

$$R = \left[\sigma_C^2\right] \tag{22}$$

In the same way as it was done with the linear models of the robot equations, the different linear KF can be fused together using the Takagi – Sugeno Fuzzy Structure as figure 6 shows.

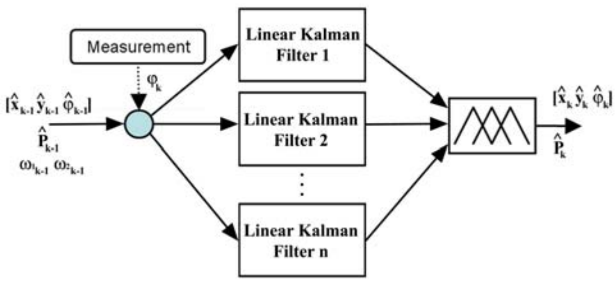

Fig. 6. Fuzzy extended Kalman filter structure.

For every time step k, each of the tuned linear Kalman filters give an estimation of the posture of the robot. The new estimation is based on the previous knowledge on the posture of the robot, the velocity of each wheel (which is an input of the model), and the measurements made. Finally, using the same membership functions calculated for the fuzzy model of the robot, the output of each filter is fuzzified into the TSFS. The output of the whole scheme is used as the estimation for the time step k, and fed back to the linear models for the next step estimation.

As it is done for the fuzzy model of the robot, one set of membership functions is used for obtaining the crisp value of X, whereas another set is used for Y.

It is important to note that the different system covariance matrices, given by the linear models must also be fused together to determine the covariance matrix used for the following step. This can be done by using a linear combination of the covariance matrices obtained from each KF. When the direction angle is one of the operating points chosen, the covariance matrix used is the same as the one estimated by the KF associated to the respective linearization point. At any other direction angle, only two different linear models are active, and due to the operating points used, one linear KF is associated with the X position variable, whereas the other is associated to Y. In this case, the estimated covariance matrix for the fuzzy nonlinear model is obtained by combining the estimated covariance matrices of both KF. The covariance matrix given by the KF associated to the X position variable is multiplied using the coefficient obtained from the X variable membership functions, whereas the one obtained from a KF associated to Y, is multiplied by the coefficient from the Y variable membership functions. This implies that the covariance matrix of the fuzzy model uses the relative weights between the X related linearization and the Y related linearization as shown on equation 23:

$$P_{Fuzzy} = \frac{\mu_n P_{X_n} + \nu_m P_{Y_m}}{\mu_n + \nu_m} \tag{23}$$

Apart from the advantages previously described, this structure also helps to reduce the number of computations needed compared to a conventional EKF. In the EKF, for each time k, all system matrices must be calculated using the related Jacobian and evaluating it on the operating point, as shown on equations 18 to 20. On the other hand, in the Fuzzy Extended Kalman Filter (Fuzzy EKF), although more matrices are needed, all of them have constant values, limiting the calculations to the fuzzification of each KF.

VII. Results

To check the estimation given by the Fuzzy EKF, several different trajectories were tested on the mobile robot, and the estimation error compared to the one given by a conventional EKF. The EKF was designed using equations 19 to 21, on the nonlinear system of equations that describes the kinematics of the mobile robot. For the EKF, the R covariance matrix is kept the same as for the Fuzzy EKF, because in both cases the only measured variable is the angle of direction. The values of the system error covariance matrix Q were determined to obtain the best performance possible for all tests.

Figures 7 and 8 show the estimation, using both methods, for the localization of the robot. The standard deviation of the noise used was a 20% of the maximum value. It can be observed that while the robot is moving on a straight line both methods give a very accurate estimation, whereas the main differences occur when the robot rotates. In this case, the estimation given by the Fuzzy EKF is better than that given by the conventional EKF.

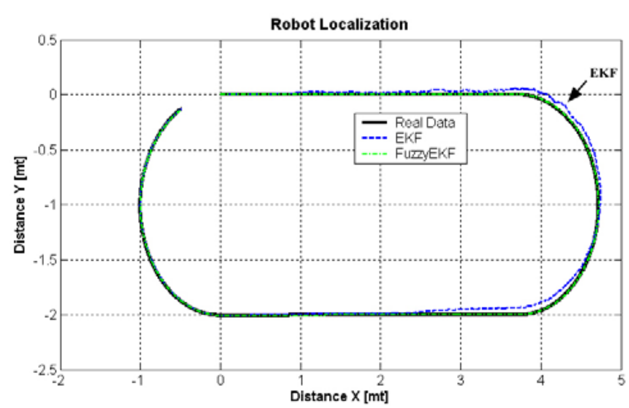

Fig. 7. Localization estimation using the EKF and the Fuzzy EKF.

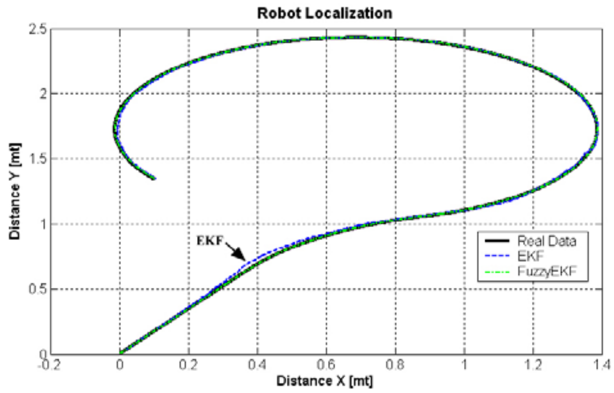

Fig. 8. Localization estimation using the EKF and the Fuzzy EKF.

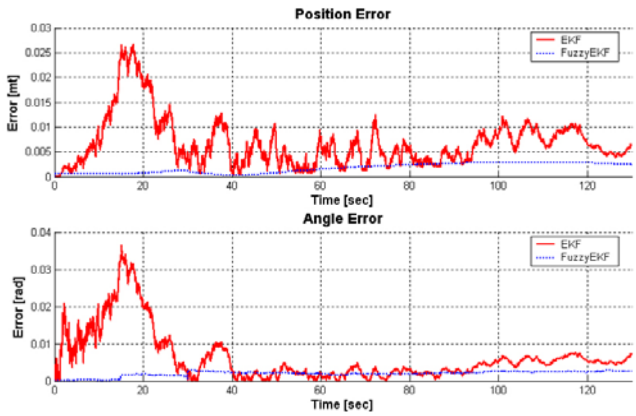

Fig. 9. Comparison of the error using an EKF and a FuzzyEKF.

Figure 9 shows the evolution over time of the estimation error for both methods, for the second example. Thanks to the fuzzification mechanism, the output of the Fuzzy EKF is smoother than the one given by the normal EKF, and the estimation error is less.

The difference between both estimations can be explained by the fact that the Fuzzy EKF contains optimal filters especially tuned for each operating point. This means that the covariance matrices used for describing the system and measurement noise can be estimated better for each linear model in the Fuzzy EKF, than for the nonlinear model in the conventional EKF.

It was also noted that for small noise standard deviations, both methods gave similar results, whereas if the value is increased, the Fuzzy EKF returns a better estimation. For the example on figures 7 and 8 the estimation error for the conventional EKF was of 6.7×10^{-3} [mt] RMS, whereas for the Fuzzy EKF the error was reduced to 1.7×10^{-3} [mt] RMS.

VIII. Conclusions

Through this work, a new way of extending the Kalman Filter is presented. Based on the membership functions obtained from a fuzzy model of the robot, a Fuzzy Extended Kalman Filter is implemented, resulting in a better posture estimation for mobile robots, in comparison with the conventional Extended Kalman Filter, thus reducing the estimation error.

The presented Fuzzy EKF shows other interesting advantages over the conventional EKF. Due to the fact that the Fuzzy EKF is based on linear models, the estimation of the covariance matrices needed is easier and results in an improvement over the posture estimation. This makes the implementation simpler and the output smoother, also reducing the computation complexity.

Based on these results, less noise contaminated residuals can be obtained to determine the measured behaviour of the robot with respect to the expected one. These residuals, which are a comparison between a model output and the sensors measurements, are used to detect and diagnose faulty elements such as sensors or actuators in the mobile robot, without the need for extra sensors.

Acknowledgment

This work was supported by FONDECYT project n^{o} 1020741, "Fault Detection and Diagnosis in Nonlinear Time Variant Systems".

References

[1] J. Borestein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 869-879, Dec. 1996.

[2] R.E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering*, vol. 86, pp. 35-45, March 1960.

[3] T. D. Larsen, K. L. Hansen, N. A. Andersen, and O. Ravn, "Design of Kalman filters for mobile robots: evaluation of the kinematic and odometric approach," *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala Coast, USA, 1999, vol. 2, pp. 1021-1026.

[4] J. Z. Sasiadek and P. Hartana, "Adaptive fuzzy logic system for sensor fusion in dead-reckoning mobile robot navigation," *Proceedings of the IFAC 15th Triennial World Congress*, Barcelona, Spain, 2002.

[5] H. Wang and C. T. Goh, "Fuzzy logic Kalman filter estimation for 2-wheel steerable vehicles," *Proceedings of the 1999 International Conference on Intelligent Robots and Systems*, Kyongju, South Korea, 1999, vol. 1, pp. 88-93.

[6] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Springer, 1997.

[7] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, 1996.

[8] R. Palm, D. Driankov, and H. Hellendoor, *Model Based Fuzzy Control*, Springer, 1996.

[9] M. Maniadakis and H. Surmann, "A genetic algorithm for structural and parametrical tuning of fuzzy systems," *Proceedings of the 1999 European Symposium on Intelligent Techniques*, Crete, Greece, 1999.

[10] S. H. Jung, "Queen-bee evolution for genetic algorithms," *IEEE Electronic Letters*, vol. 39, pp. 575-76, March 2003.

[11] S. I. Roumeliotis and G. A. Bekey, "An extended Kalman filter for frequent local and infrequent global sensor data fusion," *Proceedings of the 1997 SPIE, Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, Pittsburgh, USA, 1997, pp. 11-22.

[12] X. J. Ma, Z. Q. Sun, and Y. Y. He, "Analysis and design of fuzzy controller and fuzzy observer," *IEEE Trans. on Fuzzy Systems*, vol. 6, pp. 41-51, Feb. 1998.