Fusing odometric and vision data with an EKF to estimate the absolute position of an autonomous mobile robot

M. Marrón J.C. García M.A. Sotelo E. López M. Mazo
Dept. of Electronics
Universidad de Alcalá
CAMPUS UNIVERSITARIO Escuela Politécnica. 28871 Alcalá de Henares (MADRID)
SPAIN

Abstract – **This paper presents the development of a probabilistic algorithm based on an Extended Kalman Filter (EKF), used to estimate the absolute position of an indoor autonomous robot. With EKF it is possible to fuse relative and absolute positioning data, including some kind of uncertainty related to sensory systems. To reach this objective it is necessary to do an important model analysis to enable the on-line adaptation of the estimation algorithm. The development presented in this paper has been designed for an autonomous wheelchair, whose real-time and reliability constraints have to be taken into account in the algorithm.**

I. INTRODUCTION

Robot's position is one of the most important data processed by a navigator in an automated mobile to achieve its movement objective. Robot's functionality depends on the precision of this measurement and so it does the success of its task.

The most direct way to find robot's position is to use odometric sensors in a dead-reckoning positioning model. Nevertheless an integration process is needed to calculate the absolute position from this kind of sensor and model, so if measurements are corrupted with noise the integrated information will get worse with time. To achieve more reliable position estimation odometric data are usually completed with some other given by external sensors. Even so, the external information is always complementary to that of the relative sensors, because the processing time related to absolute positioning systems is usually too high to uniquely use these data in real time positioning.

Because of that, a fusion algorithm is generally needed to obtain robot's position from different sensory systems [1], [2].

The application presented in this communication is an autonomous wheelchair that moves autonomously in partially structured environments [3]. It uses a vision system and artificial landmarks to localize itself inside buildings. So that, the vision process will be the external absolute positioning system in the fusion development presented in this paper.

The optimal position estimator used in this work is an Extended Kalman Filter (EKF). The main drawback of this estimator is that a complete model of both positioning systems, and their related noises are needed to develop the fusion process. Providing that these models are known, this method is the most appropriated in this case because it allows to calculate the optimal position estimation, in a very short processing time, in comparison with some other probabilistic algorithms [4][5].

This algorithm is mainly used in robotic applications related to sensory filtering and position data fusion. For tasks related to map building, some other probabilistic techniques that take into account a whole history of sensed measurements, are mostly used.

In Fig. 1 it is shown a block diagram of the positioning systems and the EKF estimator developed for this robot.

II. THE EKF AS A FUSION ALGORITHM

The EKF is an optimal recursive estimator that can be used to fuse two measurements of the state vector of a system, if its non-linear model and the covariance matrix of noises related to the two measurements are known.

As the algorithm is recursive, the state vector estimation gets better as time goes by and more sensory information is added, even though if the system model is not perfectly know. This feature makes the EKF technique more reliable in the estimation process than the rest of methods based on the conditional likelihood.

The complete a-priori information needed to develop this algorithm is:

- A discrete linear or non-linear model from the system whose state vector ($\vec{x}_k$) is going to be estimated in the fusion process. The model includes also the noise vectors related to the state ($\vec{w}$) and the output ($\vec{v}$) measurements:

$$\begin{aligned} \vec{x}_{k+1} &= f(\vec{x}_k, \vec{u}_k, \vec{w}_k) \\ \vec{z}_k &= h(\vec{x}_k, \vec{v}_k) \end{aligned} \qquad (1)$$

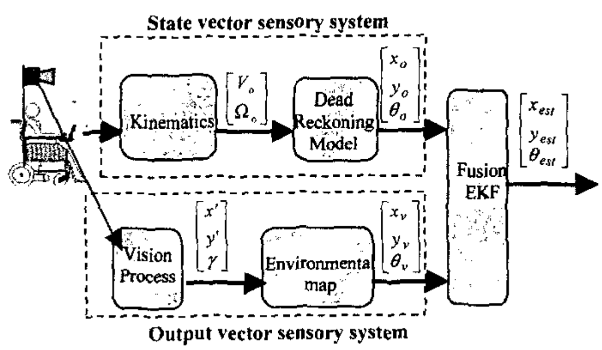

Fig. 1. The positioning system in the autonomous wheelchair

0-7803-7937-3/03/$17.00 ©2003 IEEE

- The statistic model of the noises related to the measurements (denoted by their mean μ_w, μ_v and covariance matrix Q, R):

$$\begin{aligned} \vec{w} &\Rightarrow N(\mu_w, Q) \\ \vec{v} &\Rightarrow N(\mu_v, R) \end{aligned} \quad (2)$$

With these data, the EKF development is mainly divided in two phases: the prediction and the correction phase. Their functionality is as follows:

1. The prediction phase begins obtaining the estimated state vector ($\hat{\vec{x}}_{k+1/k}$) with the system model, zeroing the noise related to the state vector ($\vec{w}=0$). After that, the innovation value of the estimation error covariance matrix ($P_{k+1/k}$) is also calculated, using the noise covariance matrix (Q) related to the state vector measurements.

2. At the correction phase the Kalman matrix (which is used to fuse position data) is firstly obtained from the innovation value of the estimation error covariance matrix ($P_{k+1/k}$) and the output vector measurements covariance matrix (R). With the Kalman gain (K), the prediction of the state vector (calculated in the first phase of the algorithm) is corrected including in it the new output vector measurement ($\vec{z}_{k+1}$). At the end, the estimation error covariance matrix is also updated ($P_{k+1/k+1}$).

Kalman filter can only be used if noises coupled to all measurements are normal, white and uncorrelated among them [6].

Any sensor normally used in robotics usually fulfils the specifications. If it is not the case, whiteness can be easily obtained with an adequate calibration, and correlation among sensed data can be easily removed, decoupling sensors on board the robot.

II. THE EKF AS A FUSION ALGORITHM

At this point it is necessary to link the positioning process in Fig. 1 with the system model in (1). According to this relation, the system state vector ($\vec{x}_k$) is going to be obtained with the odometric measurements. As the dead-reckoning model is non-linear, the Extended version of the Kalman Filter has to be used.

On the other hand, data generated by the vision process [3] will be used as the output vector ($\vec{z}_k$) in the model (1) doing some simple but also non-linear trigonometric relations.

With that, at the first phase of the EKF, the state vector of the positioning model will be estimated with the odometric measurements. This estimation will be corrected using vision measurements at the second phase of the fusion algorithm.

The global model needed to develop the estimation with the EKF algorithm, will be completed with the stochastic models that characterize the noises related to each of the measurements.

In the following paragraphs, the proposed model will be described.

A. The state vector model

The standard dead-reckoning equation of a differential kinematics robot used to obtain this first model. The noise related to the state vector ($\vec{w}$ in (1)) has been removed in this case:

$$\vec{x}_{o,k+1/k} = f(\vec{x}_{o,k}, \vec{u}_k, 0), \quad (3)$$

$$\begin{aligned} x_{o,k+1/k} &= x_{o,k} + T_s \cdot V_k \cdot \cos\theta_{o,k} \\ y_{o,k+1/k} &= y_{o,k} + T_s \cdot V_k \cdot \sin\theta_{o,k} \\ \theta_{o,k+1/k} &= \theta_{o,k} + T_s \cdot \Omega_k \end{aligned} \quad (4)$$

Sub-index 'o' included in position variables informs about their odometric origin.

The set point ($\vec{u}_k = [V_k \quad \Omega_k]^T$) can be obtained directly from the odometric measurements ($\vec{\varpi}_k = [\omega_R \quad \omega_L]^T$) using the robot inverse kinematics:

$$\begin{aligned} V_k &= \frac{R}{2}(\omega_R + \omega_L) \\ \Omega_k &= \frac{R}{D}(\omega_R - \omega_L) \end{aligned} \quad (5)$$

B. The model of noise related to the state vector

To obtain this noise model, the linear matrix that relates the odometric measurements with the state vector has to be found. This relation can be achieved applying the Taylor law to the (4), including this time the related noise $\vec{w}_k$ in the non-linear model:

$$\begin{aligned} \vec{x}_{on,k+1} &= f(\vec{x}_{on,k}, \vec{u}_k, \vec{w}_k) = \\ f(\vec{x}_{on,0}) &+ A \cdot (\vec{x}_{on,k} - \vec{x}_{on,0}) + \ldots + W_k \cdot \vec{w}_k \end{aligned} \quad (6)$$

Where $\vec{w}_k = [w_R \quad w_L]^T$ is the noise vector related to odometric measurements, and A is the transference matrix in the linear model state equation. This last matrix will be needed to calculate the estimation error covariance matrix ($P_{k+1/k+1}$), when processing the EKF correction phase, and its value is obtained as follows:

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\vec{x}_0, \vec{u}_0) = \begin{bmatrix} 0 & 0 & -V_0 \sin(\theta_0) \\ 0 & 0 & V_0 \cos(\theta_0) \\ 0 & 0 & 0 \end{bmatrix} \quad (7)$$

From equations (4), (5) and (6), the searched model will be obtained as the relation between $\vec{x}_{on,k+1}$ and $\vec{w}_k$ vectors, as follows:

$$W_k = \begin{bmatrix} R/2 \cdot T_s \cdot \cos(\theta_{o,k}) & R/2 \cdot T_s \cdot \cos(\theta_{o,k}) \\ R/2 \cdot T_s \cdot \sin(\theta_{o,k}) & R/2 \cdot T_s \cdot \sin(\theta_{o,k}) \\ R/D \cdot T_s & -R/D \cdot T_s \end{bmatrix} \quad (8)$$

Where the noise vector $\vec{w}_k$ will be characterized by its statistics (2): zero mean ($\mu_w = 0$) and covariance matrix,

$$Q_k = \begin{bmatrix} \sigma_{w_R}^2 & 0 \\ 0 & \sigma_{w_L}^2 \end{bmatrix} \quad (9)$$

To remove the mean value related to this noise vector a previous calibration process was done to the sensory system. Borenstein and Feng [7] [8] developed the UMBmark method to quantify this systematic error and calculate two correction constants that, included in the kinematical equations, remove it.

On the other hand, non-systematic errors can be modeled with another test (Extended UMBmark) designed also by Borenstein and Feng. Nevertheless, this method is only useful to compare different robots behavior in the same environmental conditions.

In fact, there is no way to model these kinds of noises with a static value, so that, the covariance matrix (9) has to be calculated on-line periodically with a data set coming from the actual odometric measurements.

All the positioning experiments presented in this work, have been done with non-sliding encoders. These kinds of sensors are mounted in passive wheels placed in parallel with driving ones [9]. Passive wheels will not slide when the robot does, so that the most important source of non-systematic noises related to the state vector will be removed using non-sliding encoders in the odometric measurements.

C. The output vector model

The output vector is determined directly from the vision system. This process is based on the detection and post processing of an artificial landmark specially designed and located in the robot environment [3]. In Fig. 2 a simplified diagram of the vision process and landmark can be seen.

As it can be notice in the figure, the landmark has three remarkable patterns: a vertical Barker code (thick lines) used to find landmarks on images, 4 black dots located in the corners used to obtain the relative position from the mark to the camera, and another vertical pattern (bar code of thin lines) used to identify each landmark in the environment.

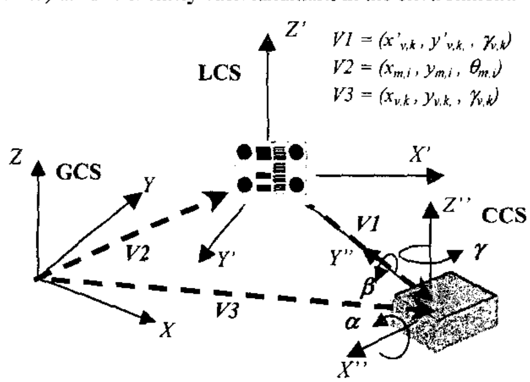

Fig. 2. Relation used by the vision process to obtain the output vector
GCS [X,Y,Z]: Global Coordinates System
LCS [X',Y',Z']: Landmark Coordinates System
CCS [X'',Y'',Z'',α,β,γ]: Camera Coordinates System

The objective of the vision algorithm is to obtain vector V3 in Fig. 2, which will be the robot's absolute position in global coordinates.

Once a mark is localized in the image, thanks to the Barker code, vision algorithms obtain the robot's relative position to the mark (in coordinates LCS). This information is shown in Fig. 2 as vector V1.

To find the positioning vision measurement (V3 $= \vec{z}_{v,k}$), the vision algorithm identifies the bar code included in the mark, which will be used as a pointer in an environmental map. This map contains the absolute position of each landmark (in coordinates system GCS), which is vector V2 in Fig. 2.

With all this information, the output vector model can be calculated as a coordinates transformation (from LCS to GCS):

$$\begin{aligned} \vec{z}_{v,k} &= h(\vec{x}_{v,k},0) = \vec{x}_{v,k} \\ \vec{z}_v &= V3 = V1 + V2 \\ x_{v,k} &= x_{m,i} + x'_{v,k} \cdot \cos(\theta_{m,i}) - y'_{v,k} \cdot \sin(\theta_{m,i}) \\ y_{v,k} &= y_{m,i} + x'_{v,k} \cdot \sin(\theta_{m,i}) + y'_{v,k} \cdot \cos(\theta_{m,i}) \\ \theta_{v,k} &= \theta_{m,i} + \theta'_{v,k} = \theta_{m,i} + \gamma_k \end{aligned} \quad (10)$$

In this case, sub-index 'v' included in position variables informs about their vision origin.

The first expression of (10) has been simplified taking into account that the output vector is equivalent to the state vector, and removing the noise from (1).

D. The model of noise related to the output vector

In [3], a detailed analysis of the noises associated to the output vector is also developed. As a result of this study, a value called 'pixel error variance' is found comparing the final mark location obtained by the complete algorithm with the firstly sensed one.

In the study it is used a different pattern for the output vector, and for the noise vector related to it:

$$\begin{aligned} \vec{z}''_{v,k} &= \begin{bmatrix} r_k & \sin(\gamma_k) & \cos(\gamma_k) \end{bmatrix}^T \\ \vec{v}''_k &= \begin{bmatrix} v_r & v_{\sin\gamma} & v_{\sin\gamma} \end{bmatrix}^T \end{aligned} \quad (11)$$

Where r_k is the mean value of the distance between the robot and the mark, and $\gamma_{v,k}$ is the camera orientation relative to the mark (see Fig. 2).

To find the model of the noise associated to the output vector (10), it is necessary to find a linear matrix that relates the position measurements obtained by the vision system and the noise vector used (11). The linear equation desired is the following:

$$\vec{z}_{vn,k} = h(\vec{x}_{vn,k}, \vec{v}_k) = \vec{x}_{vn,k} + V_k \cdot \vec{v}_k \quad (12)$$

Notice that V_k is not the same than the set point in the dead-reckoning model (5).

The relation (12) can be obtained from Fig. 2, taking into account some simplifications due to environmental conditions [3]. The required transformation is:

$$x_{vn,k} = M1 - (r_k + v_r)\cdot(\sin(\gamma_k) + v_{\sin\gamma})\cdot M2 - \\ - (r_k + v_r)\cdot(\cos(\gamma_k) + v_{\cos\gamma})\cdot M3 \\ y_{vn,k} = M4 + (r_k + v_r)\cdot(\sin(\gamma_k) + v_{\sin\gamma})\cdot M3 + \\ + (r_k + v_r)\cdot(\cos(\gamma_k) + v_{\cos\gamma})\cdot M2 \\ \theta_{vn,k} = M5 + \gamma_k \tag{13}$$

Where the constants Mj are: $M2 = \cos(\theta_{m,i})$, $V2 = \begin{bmatrix} x_{m,i} & y_{m,i} & \theta_{m,i} \end{bmatrix}^T = \begin{bmatrix} M1 & M4 & M5 \end{bmatrix}^T$ and $M3 = \sin(\theta_{m,i})$, and the notation 'i' identifies the concrete landmark that the vision system has detected in the image.

From equations (12) and (13), the model of the noise related to the output vector is as follows:

$$V_k = \begin{bmatrix} -\sin(\gamma_k)\cdot M2 & \sin(\gamma_k)\cdot M3 & 0 \\ -r_k\cdot M2 & r_k\cdot M2 & 0 \\ -r_k\cdot M3 & r_k\cdot M2 & 0 \end{bmatrix} \tag{14}$$

Where the noise vector $\vec{v}_k$ is defined by its statistics: zero mean ($\mu_v = 0$ thanks to a previous calibration process) and a covariance matrix,

$$R_k = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_{\sin\gamma}^2 & 0 \\ 0 & 0 & \sigma_{\cos\gamma}^2 \end{bmatrix} \tag{15}$$

The way to find the dynamic value of this last covariance matrix is also shown in [3], where different variables must to be taken into account: the full orientation vector from the camera to the mark, the camera parameters and the 'pixel error variance'.

Once the model analysis has been completely defined it can be easily noticed that covariance matrixes of both noise sources are dynamic. This fact makes the estimation algorithm slower, but on the other hand it is more reliable and it is easier for the EKF to converge.

IV. APPLICATION AND RESULTS OF THE EKF

Once defined the models, the will give the real time absolute position of the robot, from odometric and vision data. Fig. 3 shows the processing flowchart developed to implement the estimator; anyway, there are still some important considerations to take into account. The first one is related with the different executing period of each one of the sensory systems, due to their different computing load (as commented in section I):

- T_{so}, will be the sampling period for the odometric system.
- T_{sv}, will be the sampling period for the vision system, always larger that T_{so} and related with it by the multiplicity expression: $T_{sv} = n\cdot T_{so}$, where n is integer.

The estimator execution period will match the faster one, which is the odometric system one (T_{so}). Even though, just the EKF prediction phase is developed each T_{so}, while the EKF correction phase will only be developed each T_{sv}, because it is then when there are new vision data to fuse with the predicted state vector. This means that the algorithm will generate the complete estimation of the state vector only each T_{sv}.

The most important drawback of this fact is that the Kalman gain recursive evolution can diverge if the multiplicity factor n is too big. At the same time, noise related to the state vector gets more important as n grows, because corrections of the state vector get more distant in time. Different tests have been done to analyse this effect, concluding that if the multiplicity relation gets higher that 3 or 4 (depending on the noise contents of relative measurements), the algorithm do not converge and the estimation is not correct.

This last conclusion limits the vision system processing time. If robot's kinematics fixes T_{so} to 50ms, the EKF convergence will limit the T_{sv} to 200ms, which can be

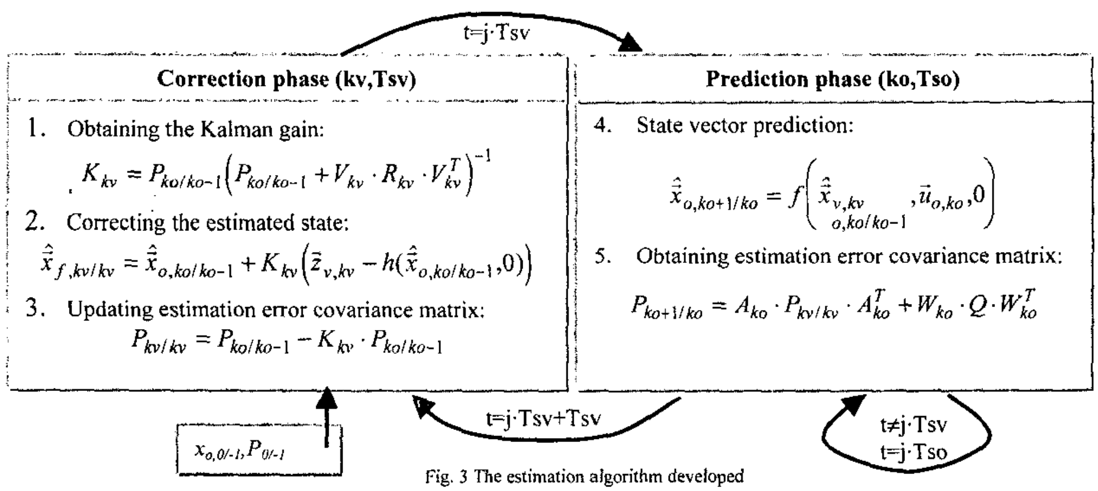

Fig. 3 The estimation algorithm developed

accomplished using an appropriate processor for running vision algorithms [3].

These timing constraints have been applied to the EKF final design, which is presented in Fig. 3.

In this figure signals sub-indexes show not only the origin of sensed data used to calculate the estimator variables, but also the execution period used to update them ('kv' if they are updated each T_{sv} and 'ko' if they are updated each T_{so}).

V. RESULTS

The final algorithm shown in previous Fig. 3 has been developed in 'C' and compiled for MATLAB, with the objective of inserting it afterwards in a complete simulation model. Using Real Time Workshop MATLAB tool, this simulation model can be downloaded and monitored from MATLAB in almost any embedded processing system.

Fig. 4 shows the simulation model of a trajectory generator and its associated position controller for the autonomous wheelchair. In this system, the EKF estimator has been inserted in the feedback loop of the position controller, so that an optimal estimation of robot's position can be used in the tracking loop.

The results obtained from the simulation of the model in Fig. 4. are shown in Figs. 5,6 and 7. For this test, noises related to both sensory systems have been modelled by static covariance matrixes, with a value according to the worst noise conditions found in real experiments (0.01 pixels^2 of 'pixel error variance' and 0.2 rad^2/s^2 of odometric variance). On the other hand, T_{so} has been set to 50ms and T_{sv} to 100ms.

Fig. 5 shows the evolution of speed variables: the ones generated by the controller (the thick and light line) and the ones sensed by the odometric system (the thin and dark line).

Fig. 6 shows the evolution of position variables: the set points obtained by the trajectory generator (the thick and light line), the ones calculated by the artificial vision system (the thin and dark line) and the ones estimated by the EKF fusion algorithm (the dark line).

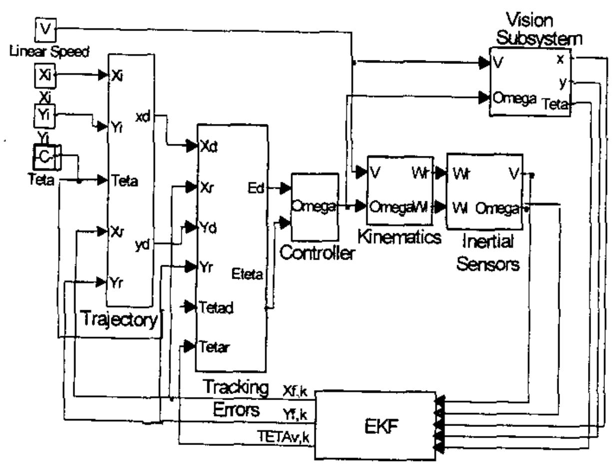

Fig 4. Trajectory generator used to test the EKF

In this figure, the different plots for variables X and Y cannot be seen properly, because they are mostly equal (this is due to the high precision of the fusion algorithm). Only variable *Theta* shows the difference between the estimation, the sensed and the set point values.

Finally, Fig. 7 shows the trajectory followed by the robot during the experiment: the set point from the trajectory generator (the thick and light line) and the one developed by the robot (the dark line).

As it can be noticed in these plots, the fusion process developed by the EKF algorithm minimizes the effect of noises associated to the different sensory systems, to obtain an optimal estimation of robot's position.

V. CONCLUSIONS

It has been developed a robust estimation algorithm based on an EKF that obtains a reliable state vector informing about robot's position. The algorithm can be used to develop control loops and trajectory tracking in autonomous indoor navigation with high reliability.

The estimation is done fusing data from odometric and vision sensory systems, whose different sampling periods and accuracy are taken into account in the prediction process.

Estimation tests made to the navigation system of an automated wheelchair have shown that the results of the position estimation are robust enough to achieve an autonomous indoor navigation of a mobile robot with high security constraints.

Different works have already been developed with similar and more complex objectives. In [10] a simultaneous localization and map generator is build with the same fusion idea. The application and the processing results (in terms of time and accuracy) are the main contribution of the development presented in this paper in comparison with some other related works.

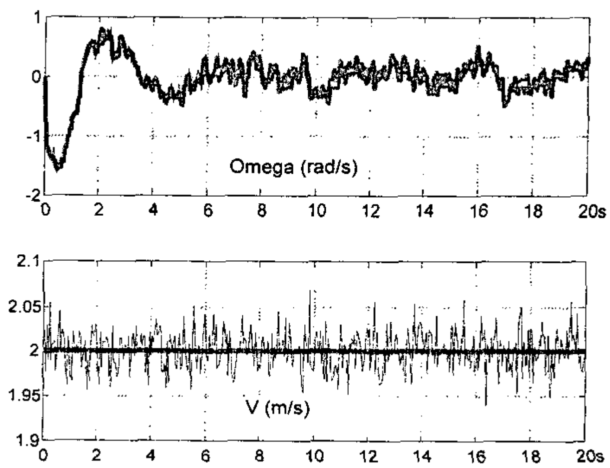

Fig. 5. Speed variables evolution

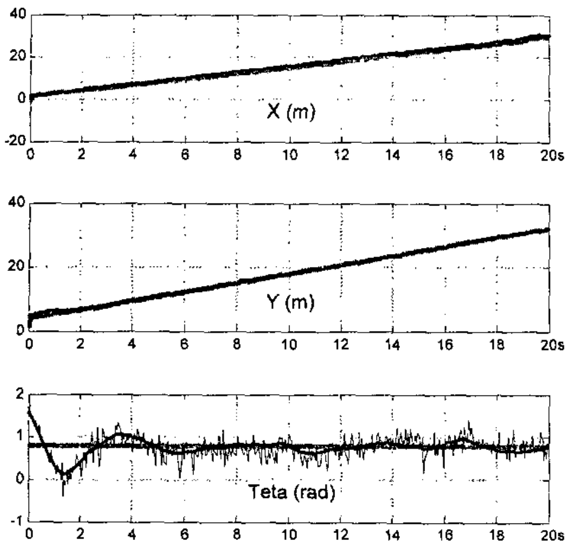

Fig. 6. State vector evolution

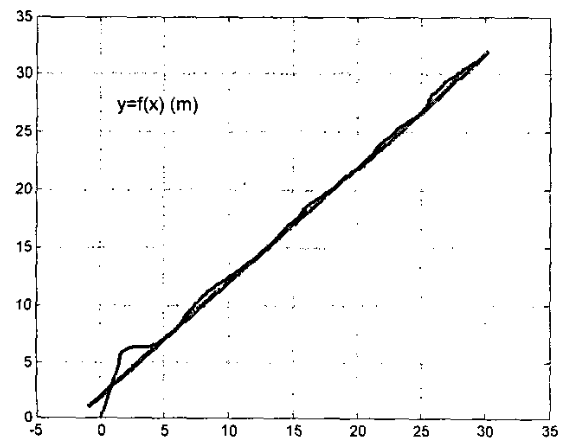

Fig. 7. Final trajectory

Real time operation results have not been obtained yet, but the extreme conditions tested in simulations assure that on line results will be precise and robust enough for the autonomous indoor navigation objective.

V. REFERENCES

[1] Ph. Bonnifait, G. Garcia, "Design and Experimental Validation of an Odometric and Goniometric Localization System for Outdoor Robot Vehicles", *IEEE Transactions of Robotics and Automation*, vol. 14, no. 4, August 1998, pp. 541 - 548.

[2] E. Fabrizi, G. Oriolo, S. Panzieri, G. Ulivi, "Mobile Robot Localization via Sensor Fusion of Ultrasonic and Inertial Sensor Data", *in Proceedings of the 8th International Symposium on Robotics Applications,* Maui, 2000.

[3] J. C. García, M. Marrón, M. Mazo, J. Ureña, "Positioning and Localization System for Autonomous Wheelchairs", *in Proceedings of the 28th Annual Conference of IEEE Industrial Electronics Society,* Sevilla, 2001.

[4] D. Fox, W. Burgard, F. Dellaert, S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots", *in Proceedings of the Sixteenth National Conference on Artificial Intelligence,* Orlando, 1999.

[5] D. Maksarov, H. Durrant-Whyte, "Mobile Vehicle Navigation in Unknown Environments: A Multiple Hypothesis Approach", *IEEE Proc. in Control Theory Applications*, vol. 142, no. 4, July 1995, pp. 385-400.

[6] G. Whelch, G. Bishop, "An Introduction to the Kalman Filter", *in Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques,* Los Angeles, 2001.

[7] J. Borenstein, L. Feng, "UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots", *in Proceedings of the 1995 SPIE Conference on Mobile Robots,* Philadelphia.

[8] J. Borenstein, L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots", *IEEE Transactions of Robotics and Automation,* vol. 12, no. 5, October 1998, pp. 869-880.

[9] K S Chong and L Kleeman, "Accurate odometry and error modeling for a mobile robot", *in Proceedings of the 1997 IEEE International Conference on Robotics and Automation,* Albuquerque USA, pp. 2783-2788.

[10] M. Gamini, P. Newman, S. Clark, H. Durrant-Whyte, M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem", *IEEE Transactions On Robotics And Automation,* vol. 17, no. 3, June 2001, pp. 229-241.