Real-time localization of an omnidirectional mobile robot resorting to odometry and global vision data fusion: an EKF approach

José Gonçalves
Polytechnic Institute of Bragança,
Department of Electrical Engineering,
Bragança, Portugal
Email: goncalves@ipb.pt

José Lima
Polytechnic Institute of Bragança,
Department of Electrical Engineering,
Bragança, Portugal
Email: jllima@ipb.pt

Paulo Costa
Faculty of Engineering
of University of Porto, DEEC
Porto, Portugal
Email: paco@fe.up.pt

***Abstract*— This paper describes a robust localization system, similar to the used by the teams participating in the Robocup Small size league (SLL). The system, developed in Object Pascal, allows real-time localization and control of an autonomous omnidirectional mobile robot. The localization algorithm is done resorting to odometry and global vision data fusion, applying an extended Kalman filter.**

I. Introduction

Soccer was the original motivation for Robocup. Besides being a very popular sport worldwide, soccer brings up a set of challenges for researchers while attracting people to the event, promoting robotics among students, researchers and general public. RoboCup chose to use soccer game as a central topic of research, aiming at innovations to be applied for socially significant problems and industries [1].

As robotics soccer is a challenge in an highly dynamic environment, the robot and ball position information must be accessible as fast and accurate as possible [2]. As an example if the ball has a velocity of 2 ms^{-1} and if the lag time is 100 ms, the ball will travel a distance of 20 cm between two sampling instants, compromising the controller performance. The presented localization algorithm is updated 25 times per second, fulfilling the proposed real-time requisites.

Robots maintain a set of hypotheses with regard to their position and the position of different objects around them. The input for updating these beliefs come from poses belief and various sensors. An optimal estimation can be applied in order to update their beliefs as accurately as possible. After one action the pose belief is updated based on data collected up to that point in time, by a process called filtering. Kalman filtering is a standard approach for reducing the error in a least squares sense, using measurements from different sources [3][4].

This paper describes a robust real-time localization system based on odometry and global vision data fusion applying an extended Kalman filter. Similar approaches were applied successfully in other domains, such as Unmaned Air Vehicles (UAVs) localization. Unmanned aerial vehicles are increasingly used in military and scientific research. Some miniaturized UAVs rely entirely on the global positioning system (GPS) for navigation. GPS is vulnerable to accidental or deliberate interference that can cause it to fail. For UAVs relying solely on GPS for navigation such an event can be catastrophic [5] [6].

II. Relative position estimation

Omnidirectional vehicles are widely used in robotics soccer, allowing movements in every direction, where the extra mobility is an important advantage. The fact that the robot is able to move from one place to another with independent linear and angular velocities contributes to minimize the time to react, the number of maneuvers is reduced and consequently the game strategy can be simplified [7] [8] [9]. The omnidirectional robots use special wheels, that allow movements in every direction. The movement of these robots does not have the restraints of the differential robots [10], presenting the disadvantage of a more complex control. It is possible to conclude from the geometry of a three wheel omnidirectional robot, presented in Figure 1, that the velocities V_x, V_y and w vary with the linear velocities V_1, V_2 and V_3, as shown in equations system (1) [11].

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} -sin(\theta) & cos(\theta) & L \\ -sin(\frac{\pi}{3}-\theta) & -cos(\frac{\pi}{3}-\theta) & L \\ sin(\frac{\pi}{3}+\theta) & -cos(\frac{\pi}{3}+\theta) & L \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} \quad (1)$$

The robot relative position estimation is based on the odometry calculation. The odometry calculation uses each wheel velocity in order to estimate the robot position, the disadvantage is that the position estimate error is cumulative and increases over time.

The robot kinematic equations can be represented by the equations system (2), in alternative to the equations system (1).

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & L \\ -sin(\frac{\pi}{3}) & -cos(\frac{\pi}{3}) & L \\ sin(\frac{\pi}{3}) & -cos(\frac{\pi}{3}) & L \end{pmatrix} \begin{pmatrix} V \\ V_n \\ w \end{pmatrix} \quad (2)$$

978-1-4244-1666-0/08/$25.00 ©2008 IEEE

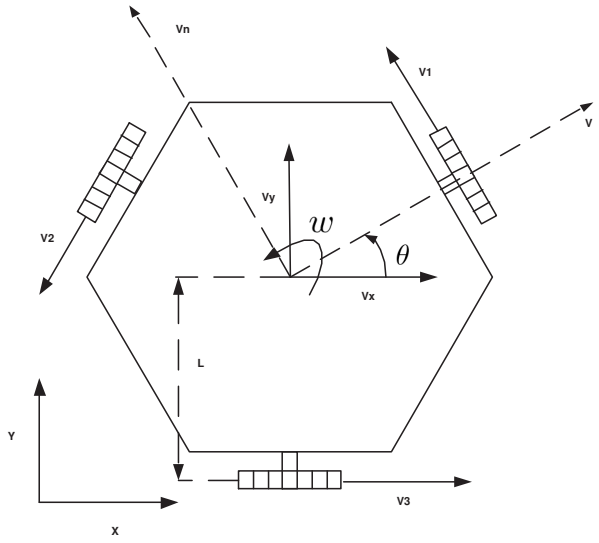

Fig. 1. Geometry of a three wheel omnidirectional robot

The linear and angular velocities V, V_n and w can be obtained rewriting equations system (2) as equations system (3),

$$\begin{pmatrix} V \\ V_n \\ w \end{pmatrix} = G \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} \tag{3}$$

where G is :

$$\begin{pmatrix} 0 & \frac{-1}{2\sin(\frac{\pi}{3})} & \frac{1}{2\sin(\frac{\pi}{3})} \\ \frac{1}{1+\cos(\frac{\pi}{3})} & \frac{-1}{2(1+\cos(\frac{\pi}{3}))} & \frac{-1}{2(1+\cos(\frac{\pi}{3}))} \\ \frac{\cos\frac{\pi}{3}}{L(1+\cos(\frac{\pi}{3}))} & \frac{1}{2L(1+\cos(\frac{\pi}{3}))} & \frac{1}{2L(1+\cos(\frac{\pi}{3}))} \end{pmatrix} \tag{4}$$

By this way θ can be found, applying an first order approximation, as shown in equation (5),

$$\theta(K) = \theta(K-1) + wT \tag{5}$$

where T is the sampling time.

After θ calculation an rotation matrix, presented in matrix (6), is applied in order to obtain V_x and V_y, as shown in equations system (7),

$$B = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$\begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} = BG \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} \tag{7}$$

where:

$$BG = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \tag{8}$$

$$M_{11} = \frac{-\sin(\theta)}{1+\cos(\frac{\pi}{3})} \tag{9}$$

$$M_{12} = \frac{-\cos(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\sin(\theta)}{2(1+\cos(\frac{\pi}{3}))} \tag{10}$$

$$M_{13} = \frac{\cos(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\sin(\theta)}{2(1+\cos(\frac{\pi}{3}))} \tag{11}$$

$$M_{21} = \frac{\cos(\theta)}{1+\sin(\frac{\pi}{3})} \tag{12}$$

$$M_{22} = \frac{-\sin(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\cos(\theta)}{2(1+\cos(\frac{\pi}{3}))} \tag{13}$$

$$M_{23} = \frac{\sin(\theta)}{2\sin(\frac{\pi}{3})} + \frac{-\cos(\theta)}{2(1+\cos(\frac{\pi}{3}))} \tag{14}$$

$$M_{31} = \frac{\cos\frac{\pi}{3}}{L(1+\cos(\frac{\pi}{3}))} \tag{15}$$

$$M_{32} = \frac{1}{2L(1+\cos(\frac{\pi}{3}))} \tag{16}$$

$$M_{33} = \frac{1}{2L(1+\cos(\frac{\pi}{3}))} \tag{17}$$

x and y estimate is calculated applying an first order approximation, as shown in equations (18) and (19),

$$x(K) = x(K-1) + V_x T \tag{18}$$

$$y(K) = y(K-1) + V_y T \tag{19}$$

where T is the Sampling Time.

III. Absolute position estimation

A global vision system was used in order to obtain the robot absolute position estimation, it is presented in Figure 2. The global vision system is required to detect and track a mobile robot in an area supervised by one camera. The camera is placed perpendicular to the ground, fixed to an metallic structure, allowing a maximal height of 3 meters, although in the presented case is placed only at 2 meters height. Placing the camera higher reduces the parallax error, reduces problems such as ball occlusion and the vision field increases, although the image quality decreases and the error due to the barrel distortion effect increases. All this items are discussed bellow.

- **Image quality**:

The image quality concept, in this case study, is related with the number of pixels that are available at each frame to identify and localize an robot marker. The markers, placed on the robot top, have the goal to provide information about the robot localization. Their geometric shape is a circle, all with the same dimensions and with different colors. The number of observed pixels for each marker depends on the illumination conditions, color calibration and camera height. If the camera is placed higher the vision field is bigger, consequently the

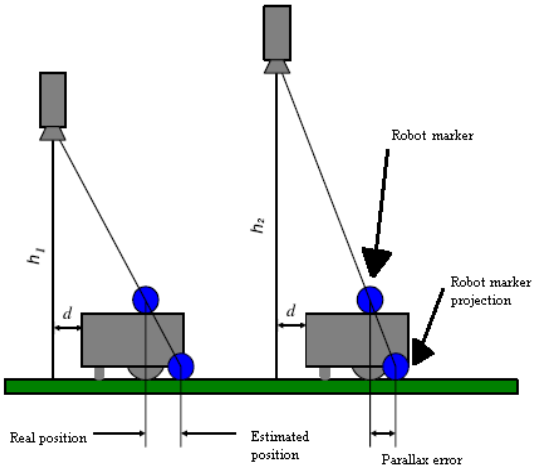

Fig. 2. Global vision system

maximum number of observed pixels for each marker will be reduced.

- **Parallax**:

The parallax error is minimized by placing the camera higher. In Figure 2 it can be observed that for a height $h_2 > h_1$ the parallax error is reduced considerabilly [12].

- **Barrel distortion**:

Barrel distortion consists in a lens aberration or defect that causes straight lines to bow outward away from the center of the image. As the vision field is only 1 m^2 and the camera is only placed at 2 meters height the barrel distortion error is not considerable, by this way its associated localization error is negligible.

A. Markers Localization

Knowing at first hand that are necessary to localize the robot two different markers, one to identify the center and another to provide information for the angle calculation. Being the field green, and the ball orange, the colors for the robot markers should be the most distant as possible in the RGB cube. The chosen colors were blue for the robot center and yellow for the angle, being the official Robocup colors to distinguish two teams in the SSL [1].The used robot is illustrated in Figure 3. The ball localization is achieved the same way as the robot center, the only difference is that is a marker placed at a different height and associated to a different color.

Once detected a calibrated pixel with the color of the object to localize, its coordinate is registered in *x* and *y* of the image. This process is repeated for all the active image area, being the coordinates accumulated. At the end of searching this area the average of all coordinates is calculated, which corresponds to the calculation of the marker center.

B. Colors Calibration

An essential component of a colored vision system is the color classification and detection algorithms for each pixel.

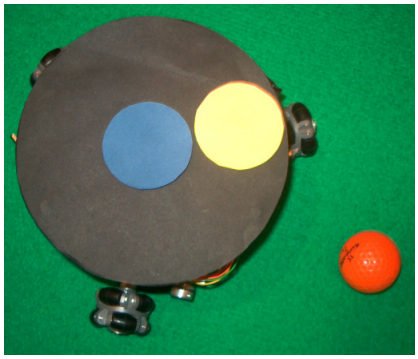

Fig. 3. Omnidirectional robot prototype

Considering the 16777216 colors $(256 * 256 * 256)$ that is possible to represent with 8 bit for each component R, G and B, it is possible to build a colored cube defined from $(0, 0, 0)$ to $(2^8; 2^8; 2^8)$ with RGB components in each vertex.

A cube edge represents 256 discrete and different color points and there are 16777216 different colored points inside the cube. It is necessary to teach the system the colors of the 3 used markers. In practice, several cube points belong to one marker color. In short, calibration fits marker colors and cube points (RGB combinations). It is necessary to calibrate the marker colors in the localization system setup and after a lightning change. This calibration can be saved in a file that keeps all RGB combinations for each color.

C. Image-World mapping

In order to extract the observed object localization, using the image, it is necessary to construct the following function:

$$m : Nx \times Ny \longrightarrow \Re^2(u, v) \longrightarrow (x, y) \tag{20}$$

Function (20) maps 2d image coordinates in to world coordinates. This function provides the *x* and *y*, assuming that the component *z* is zero. As the objects are not all placed at the same height there will be error due to parallax. This error can be compensated having in account the objects and camera height. This compensation can only be done after the marker localization.

D. Parallax correction

With the acquired image, it is only possible to extract information about the object in the *xy* plane. Knowing *z* (Marker height) and *h* (Camera height) it is possible to correct the values $\tilde{x}$, $\tilde{y}$ to obtain *x*, *y* [12].

The function F that implements this compensation is described by:

$$F : \Re^3 \longrightarrow \Re^2(\tilde{x}, \tilde{y}, z) \longrightarrow (x, y) \tag{21}$$

where:

$$x = F_x(\tilde{x}, \tilde{y}, z) = \tilde{x} - zh \tag{22}$$

$$y = F_y(\tilde{x}, \tilde{y}, z) = \tilde{y} - zh \tag{23}$$

E. Robot angle calculation

As referred previously in subsection III-A, the blue and yellow markers are used for the robot detection and localization. The blue marker allows x and y calculation, and the yellow marker allows the angle calculation. The extracted information from each marker is each one center position (*x,y*). Considering $P_r(x, y)$ the position of the blue marker and $P_a(x, y)$ the position the yellow marker, it is possible to calculate the vector that connects P_r to P_a, having as parameters a and b, as shown in Figure 4.

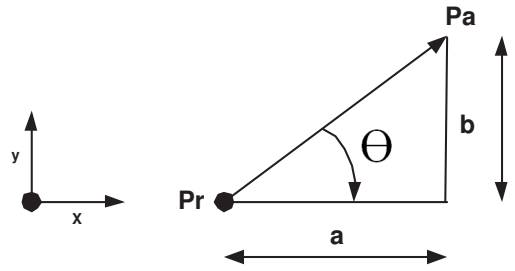

Fig. 4. Robot angle

By this way, the vector components in x and y are given by a and b.

$$a = P_a(x) - P_r(x) \tag{24}$$

$$b = P_a(y) - P_r(y) \tag{25}$$

By this way is possible to know the robot angle, resorting to an trigonometric operation:

$$\theta = arctg\frac{b}{a} \tag{26}$$

This operation can be achieved using the function arctan2(b,a), which receives 2 parameters, solving the problem related with division by zero.

F. Global vision error study

It was made for the global vision localization system an analysis of the error probability distributions [3][4]. The position error probability distributions were approximated to Gaussian distributions [13][14], being the results presented in SI units.

The number of obtained pixels for the blue marker ($Q1$), affects the error variance in x and y, as shown in the next table:

$Q1$	x	y
5-10	1,5E-05	1,9E-05
10-20	9,25E-06	7,36E-06
20-30	4,84E-06	4,86E-06
30-40	4,15E-06	3,80E-06
≥ 40	1,96E-06	2,21E-06

On the other hand the variance of the angle error probability distribution is affected by the number of pixels obtained for both makers, for the blue ($Q1$) and for the yellow ($Q2$), as shown in the next table:

$Q1$	5-10	10-20	20-30	30-40	≥ 40
5-10	0,14	8E-02	1,2E-02	1E-02	6,2E-03
10-20	1,6E-02	9,9E-03	1,3E-02	6,6E-03	4,6E-03
20-30	1,5E-02	9,9E-03	7,2E-03	4,9E-03	3,9E-03
30-40	1,4E-02	9,5E-03	5,9E-03	4,4E-03	2,9E-03
≥ 40	1,4E-02	7,2E-03	5,77E-03	3E-03	3E-03
$Q2$					

The variance for less than 5 pixels is not presented, because it is considered that the thrust in the sensor data is null.

As an example an angle error probability distribution is shown in Figure 5. The probability distribution was approximated to a Gaussian distribution being the variable of interest to extract the standard deviation which allows the variance calculation.

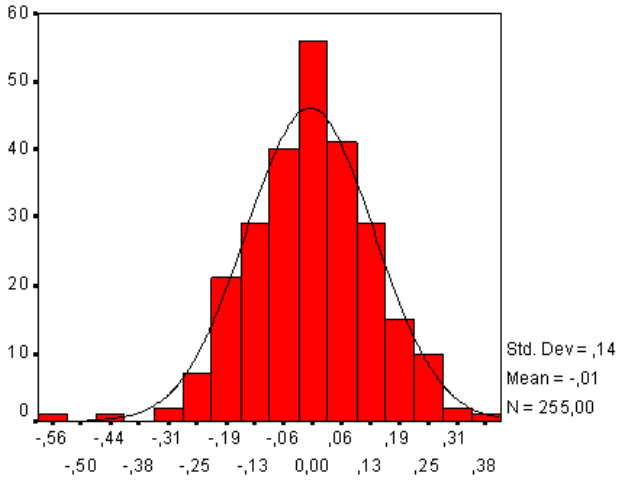

Fig. 5. Probability distribution for Q1 and Q2 between 5 and 10 pixels

IV. Odometry and Global Vision Data Fusion

Odometry and global vision data fusion was achieved applying an extended Kalman filter. This method was chosen because the robot motion equations are nonlinear and also because the measurements error probability distributions can be approximated to Gaussian distributions.

A. Extended Kalman filter algorithm

With the dynamic model given by equations system (7) and considering that control signals change only at sampling instants, the state equation is:

$$\frac{dX(t)}{dt} = f(X(t), u(t_k), t), t\epsilon[t_k, t_{k+1}] \tag{27}$$

Where $u(t) = [V_1 V_2 V_3]^T$, that is, the odometry measurements are used as kinematic model inputs. This state should

be linearized over $t = t_k$, $X(t) = X(t_k)$ and $u(t) = u(t_k)$, resulting in:

$$A^*k = \begin{pmatrix} 0 & 0 & \frac{-\sin(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\cos(\theta)}{2(1+\cos(\frac{\pi}{3}))} \\ 0 & 0 & \frac{\cos(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\sin(\theta)}{2(1+\cos(\frac{\pi}{3}))} \\ 0 & 0 & 0 \end{pmatrix} \quad (28)$$

with state transition matrix:

$$\phi^*(k) = exp(A^*(k)(t_k - t_{k-1})) \quad (29)$$

Resulting in:

$$\phi^*k = \begin{pmatrix} 1 & 0 & (\frac{-\sin(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\cos(\theta)}{2(1+\cos(\frac{\pi}{3}))})T \\ 0 & 1 & (\frac{\cos(\theta)}{2\sin(\frac{\pi}{3})} + \frac{\sin(\theta)}{2(1+\cos(\frac{\pi}{3}))})T \\ 0 & 0 & 1 \end{pmatrix} \quad (30)$$

Where T is the sampling time $(t_k - t_{k-1})$.

Thus the observations are obtained directly, H^* is the identity matrix.

The extended Kalman filter algorithm steps are as follows [15]:

1) State estimation at time $t = t_k, X(k^-)$, knowing the previous estimate at $t = t_{k-1}$, $X(k-1)$ and control $u(t_k)$, calculated by numerical integration as shown in equations (5), (18) and (19).
2) Propagation of the state covariance

$$P(k^-) = \phi^*(k)P(k-1)\phi^*(k)^T + Q(k) \quad (31)$$

Where $Q(k)$ is the noise covariance (27) and also relates to the model accuracy. In order to achieve a more realistic model of the odometry error probability distribution it is necessary to have in account that for abrupt acceleration an deceleration the wheels can slip, consequently there is an significant position estimate error increase, mainly in the angle. By this way the used odometry variance error model for x and y is:

$$Var_{xy} = K_1 + K_2 la(k-1)^2 \quad (32)$$

Where K_1 is the variance when the robot is moving in steady state and K_2 is a constant that relates the variance with the previous sample time acceleration $la(k-1)$. The previous sample time acceleration is used instead of the acceleration obtained for the present sample time because it is more representative to evaluate the odometry error noise, because transitions are updated from the previous sample time up to the present sample time.

The angle variance is modeled in a similar way, but using different constants (K_3 and K_4):

$$Var_{Angle} = K_3 + K_4 la(k-1)^2 \quad (33)$$

As there is a measure, the follow also apply:

3) Kalman gain calculation

$$K(k) = P(k^-)H^*(k)^T(H^*(k)P(k^-)H^*(k)^T + R(k))^{-1} \quad (34)$$

Where $R(k)$ is the covariance matrix of the measurements.

4) State covariation update

$$P(k) = (I - K(k)H^*(k))P(k^-) \quad (35)$$

5) State update

$$X(k) = X(k^-) + K(k)(z(k) - h(X(k^-, 0))) \quad (36)$$

Where z(k) is the measurement vector and $h(X(k^-, 0))$ is $X(k^-)$.

B. Kalman filter performance

With the objective of evaluating the Kalman filter performance a robot race was made, as shown in the flowchart presented in Figure 6. It is possible to observe that the robot moves across several locations, executing the trajectory presented in Figure 7.

The goal of the controller is to move the robot to a target position with controlled velocity. As input parameters we have as goal the robot displacement to a target position. Initially a position vector pointing to the target position is calculated, the position vector is normalized converting it into a velocity vector, becoming this the objective to accomplish. The system (1) is used to calculate the velocity that each wheel must have in order to accomplish the objective. At each sampling time the estimated position changes, consequently the position vector changes, the velocity vector changes and the reference speed of each motor changes. The controller has also as objective to follow the trajectory with an angle near to zero. One important fact that needs to be enhanced from the graphics presented in Figures 7 and 8, is that when it is expected the robot to pass by the position $x = 20\ cm$ and $y = -20\ cm$, the robot starts to move to the next target position. This happens because the objective of reaching one position is accomplished if the error in x and in y is less than $2\ cm$, making the state machine evolve to the next state, changing the objective to $x = 20\ cm$ and $y = 20\ cm$.

The image quality of the robot markers for the presented robot race are presented in Figure 9 and the variance of the estimated robot position is presented in Figure 10. Whenever the image quality decreases of position estimate variance increases, compromising the controller performance. On the other hand whenever the image quality increases the error variance is reduced and when the state update is done the position estimate error is reduced.

V. Conclusions

It is very important for the robot to have an accurate knowledge of its position in order to better accomplish its mission requisites. If the robot does not know where it is, it can't decide what to do next.

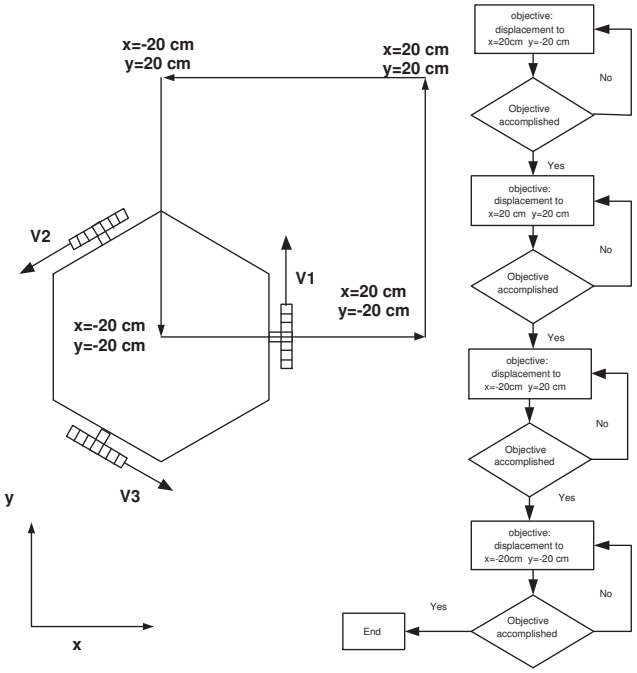

Fig. 6. Flowchart of the robot race

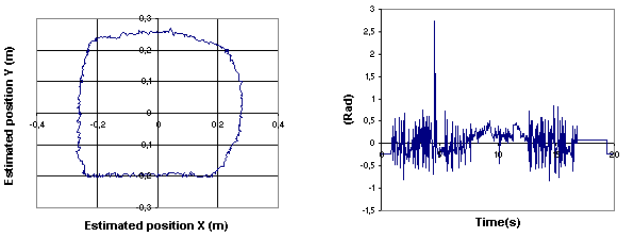

Fig. 7. a) Robot trajectory, and (b) Estimated Angle

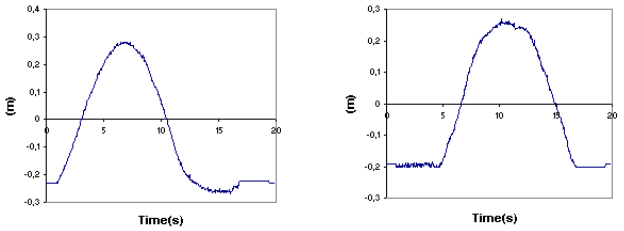

Fig. 8. a)Estimated x position, and (b) Estimated y position

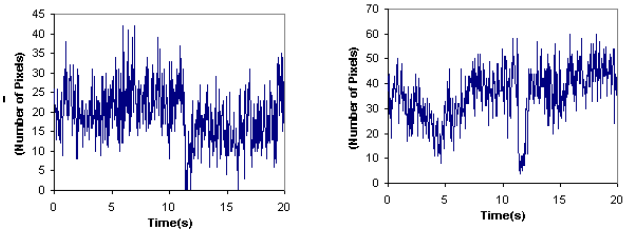

Fig. 9. a) Image quality of the center marker Q1, b)Image quality of the angle marker Q2

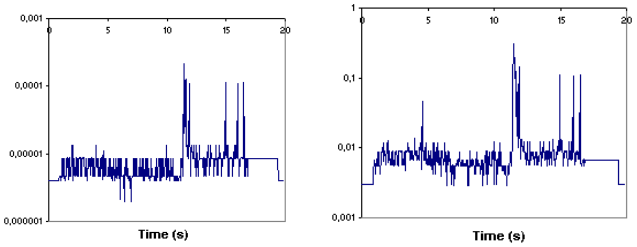

Fig. 10. a) x and y variance, and (b) Angle variance

Odometry and global vision real-time data fusion was achieved applying an extended Kalman filter. This method was chosen because the robot motion equations are nonlinear and also because the measurements error probability distributions can be approximated to Gaussian distributions.

Omnidirectional vehicles have many advantages in robotics soccer applications. The fact that the robot is able to move from one place to another with independent linear and angular velocities contributes to minimize the time to react, the number of maneuvers is reduced and consequently the game strategy can be simplified.

REFERENCES

[1] "Robocup" http://www.robocup.org/, 2008.

[2] A. Sousa, *Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão,* PHD Thesis, Faculty of Engineering of the University of Porto, 2003.

[3] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics,* MIT Press, 2005.

[4] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, *Principles of Robot Motion : Theory, Algorithms, and Implementations,* MIT Press, 2005.

[5] G. Mao, S. Drake, and B. Anderson, *Design of a Extended Kalman Filter for UAV Localization,* Information, Decision and Control, 2007.

[6] Heng C. Lin, *Kalman Filtering of FDOA/TDOA Missile Tracin System,* Master thesis, Naval Postgraduate School Monterey CA, 2001.

[7] F. Ribeiro, I. Moutinho, P. Silva, C. Fraga and N. Pereira, *Controlling Omni-Directional Wheels of a Robocup MSL Autonomous Mobile robot,* Scientific Meeting of the Robotics Portuguese Open, 2004.

[8] L. Huang, Y. Lim, D. Lee and C. Teoh, *Design and analysis of a four-wheel omnidirectional mobile robot,* Procedings of the 2nd International Conference of Autonomous Robots and Agents, Palmerston North, New Zealand 2004.

[9] J. Gonçalves, P. Costa and A. Moreira, *Controlo e estimação do posicionamento absoluto de um robot omnidireccional de três rodas,* Revista Robótica, Nr 60, pp 18-24, 2005.

[10] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics,* Cambridge University Press, 2000.

[11] T. Kalmár-Nagy, R. D'Andrea and P. Ganguly *Near-Optimal Dynamic Trajectory Generation and Control of an Omnidirectional Vehicle,* Sibley School of Mechanical and Aerospace Engineering, 2002.

[12] P. Costa, *Localização em tempo real de múltiplos robots num ambiente dinâmico,* PHD Thesis, Faculty of Engineering of the University of Porto, 1999.

[13] M. I. Ribeiro, *Gaussian Probability Density Functions: Properties and Error Characterization,* Technical Report, IST, 2004.

[14] R. Negenborn, *Robot Localization and Kalman Filters - On finding your position in a noisy world,* Master Thesis, Utrecht University, 2003.

[15] G. Welch and G. Bishop, *An introduction to the Kalman filter,* Technical Report, University of North Carolina at Chapel Hill, 2001.