Architecture of a Generic Vehicle for Evaluating Localization Algorithms

A. Zakzouk, A. Lambert, E. Seignez, S. Bouaziz, T. Maurin, R. Reynaud
IEF, UMR CNRS 8622
Université de Paris-Sud,Box 220
91405 Orsay, France
name@ief.u-psud.fr

Abstract

In these last few years, a number of localization methods have been proposed for vehicles equipped with proprioceptive and exteroceptive sensors. Unfortunately, a systematic study of these methods has never been done. In order to easily make such a study, Minicar (a small teleoperated mobile vehicle) has been built. A Client/Server architecture with a wireless communication enables to deport computing to a distant computer. Consequently, the development of the localization algorithms is simplified and the embedded electronics of the vehicle is low cost. Furthermore, a light embedded hardware offers more autonomy to the vehicle.

1. Introduction

Development of a powerful localization system has suggested number of works. Most of the published works [1] use the same kinds of proprioceptive and exteroceptive sensors. Localization is always done in a two phases process: prediction and estimation. Prediction (which integrates the odometric data) is quite similar whatever the method is. However, the estimation could be done in very different ways. Thus, we have seen the emerging of several technics using probabilistic [5], interval analysis [7] and Markovian [3] methods.
Depending on the chosen method, the localization can be performed in complex, dynamic or badly mapped environments. Some methods are well adapted to light electronic architectures [4] whereas others [3], [6] are more robust but need a lot of computing power. So as to compare and test some localization methods (which were never been used in real environment), we decided to design a generic experimental vehicle. This vehicle must be able:

- to be controlled by a distant operator
- to give an estimation of the localization on an environment map
- to provide a visual return

Finally, so as to enable easy experimentation, our vehicle architecture has to be light and must have a small size. As some localization algorithms require a lot of computing power (not easily implementable in a light embedded electronics), we chose a client-server architecture with a powerful distant workstation. The communication is ensured by a wireless connection (Wifi) and an high level protocol that we defined. All the actuators and sensors of our small vehicle are accessible through the Wifi network. The mobile is a data server which can accept requests and commands from one or more distant workstations. The choice of a rough electronics seems particularly judicious because it makes possible to increase the autonomy of the mobile.
Keeping in mind this constraint, we designed a single card (low consumption) where various sensors and actuators are connected. The starting point of this project was the hardware architecture of the PICAR project [2] dedicated to controlling a full size electric vehicle.
In the following section we present the vehicle architecture. Next, a localization method based on Kalman filtering is presented. Finally, some experimental results show the soundness of our architecture.

2. Description of the Experimental Vehicle

2.1. Hardware description

We use a mechanical base of a vehicle model reduced to 1/8 dimensions. Its dimensions are 1m length by 40cm width (figure 1). The propulsion is ensured by an electric motor (D.C. motor) assembled on a reducer and a differential bridge of the aft wheels. We instrumented this vehicle (figure 7) while adding to it:

- an embedded computer
- a data transmission module (WIFI)
- sensors (odometer, magnetic course,ultrasonic sensor, image sensor)
- actuators to order the propulsion and the direction of the vehicle

The engine is ordered (through a power module) by a PWM signal generated by the embedded computer. Another PWM signal makes it possible to control a steering servo-motor. The ordering of steering is carried out *a priori* in open loop but a pair of odometric sensors (gone up on the two aft wheels) allows us to implement a correction on the steering. Consequently, the odometric sensors also allow us to establish a control of position or speed on the vehicle.

Fig. 1. Minicar

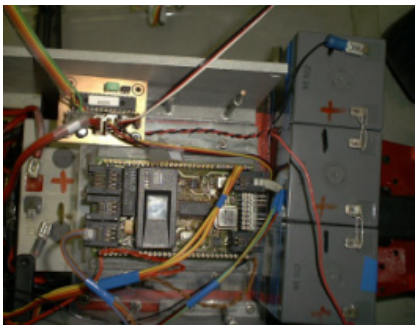

Fig. 2. Embedded computer

2.2. Material architecture

Hardware architecture is based on a centralized computer using a 16bits microcontroller (INTEL186 family) (figure 2). The sensors are accessible through an I2C bus and RS232 serial data transfers. The actuators are ordered by PWM signals. A programmable circuit, interfaced to the microcontroller, measures the covered distance from the odometers by accumulating incremental coders pulses. Such a choice discharges the computer and ensures a resolution of movement of 1 point per μs, which is largely enough. Figure 3 gives a synoptic presenting the various modules embedded on the mobile. The following paragraphs will detail the most relevant parts of the material architecture.

2.2.1. The embedded computer

The embedded computer is based on an intel80186 microcontroller which have a 16 bits core clocked at 20Mhz. It is an on-chip computer designed by BECK company (see HTTP: \\www.bcl.de). It integrates 512Ko RAM, 512Ko Flash ROM and several peripherals (TIMER, DMA, 2 RS232 UART, I2C bus, many I/O pins and an Ethernet 802.3 10BaseT link) in one single DIL32-housing. BECK circuit comes with a preinstalled real time operating system including TCP/IP stack, Web-server, FTP-server, Telnet-server. It also embeds HAL API (Hardware Abstraction Layer API). It is possible to carry out up to 64 tasks in

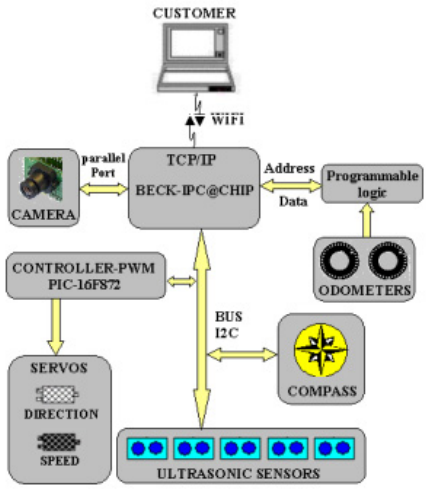

Fig. 3. Flowchart of the hardware

time sharing with preemption mechanisms.

We use a normalized PHYTEC format board to implement our embedded computer. We especially added CAN Controller and CPLD logic programmable chip (Complex Programmable Logic Device) for general purpose.

The I2C bus was prefered to the CAN bus because of the sensors space dispersion on the experimental vehicle. Indeed, the vehicle is 1 meter long and ultrasonic sensors modules are fixed like in the figure 7. These car dimensions do not justify the use of a more complex bus protocol like CAN bus used on PICAR vehicle [2].

The operating system of the BECK computer is strongly inspired by the DOS system (with Real Time specificity) and is accessible by Ethernet network. This very useful solution allows us to download the application program and to easily carry out a debugging (DEBUG). It is also possible to implement high level software applications like telnet, ftp or WEB-server. We use standard development tools such as Borland C++.

2.2.2. Proprioceptive and exteroceptive sensors

Motions of the mobile are measured by two incremental coders assembled on the rear wheel unit. These sensors, with a 200 points per turn angular resolution, give a millimeter-length precision which is considered as enough given the mechanical precision.

The mobile localization within its environment is assured by ultrasonic sensors (SFR08 of Devantech) fixed around the vehicle. These sensors are also used by the anti-collision system in order to stop the vehicle in the presence of too close obstacles. The ultrasonic sensors are connected to the computer by using the I2C bus. It is possible to launch a simultaneous measurement on all the sensors by using general I2C call.

Currently, seven ultrasonic modules are connected and our hardware is able to accept up to sixteen sensors. The software part deals with the hot connection and disconnection of the sensors. This brings robustness to the

system. Each ultrasonic module uses transmitter resonator and receiver resonator both calibrated at 40KHz frequency. Taking into account the principle of measurement based on shooting windows, the sensor is blind below 3 cm and its maximum sensing distance is 6m.

A light intensity sensor, not used at the present time, is placed in the center of the ultrasonic module for future uses coupled with image sensors.

We also embedded an electronic compass in order to have a magnetic course with an accuracy of 2 degrees. This sensor is connected to the I2C bus.

Lastly, a color image sensor (80 lines * 143 columns) is connected to a parallel interface. This sensor embeds a microcontroller allowing a video acquisition and a low level image processing. The goal of this sensor is to check the good course of the experiment (thus it is not taken into account for automatic course tracking).

2.2.3. Vehicle control

A servo-motor interprets the orders given through PWM signal (Pulse Width Modulation) to steer the vehicle. A second PWM signal ensures the gear control. We connected on the I2C bus a module which generate PWM signals according to the instructions of the embedded decision system. However, the initiative of a movement is not realized by the embedded computer but by another delocalized system which sends its orders through WIFI connection according to a specific protocol.

2.3. Communication protocole

The client/server communication was designed in order to make it possible that the mobile accepts several connections. By doing this, several workstations will be able to collect the sensors data. The mobile being a server, its IP (Internet Protocol) address is known by potential clients. This choice makes it possible to carry out the teleoperation of the mobile from any computer. In a traditional operation, the client sends a request to the mobile which answers him. These requests can be specific requests for sensor data or orders. The client can also ask for receiving periodic sensors data. At the beginning of the communication sequence, the client request the properties of the experimental vehicle. The response, made up of dimensions of the mobile, the number of sensors and their location, makes it possible to automatically adapt the client software to the evolutionary characteristics of the mobile. Consequently, the client program is independent of the characteristics of the experimental vehicle.

3. Localization

Once the experimental vehicle was achieved, we decided to implement a first algorithm of localization using telemeters and odometers. We chose a widely used algorithm : the Kalman filter.

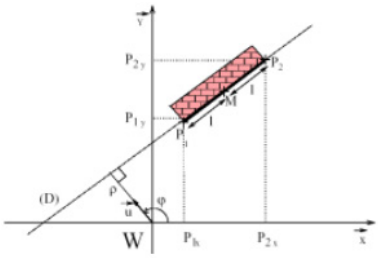

Fig. 4. Representation of a wall

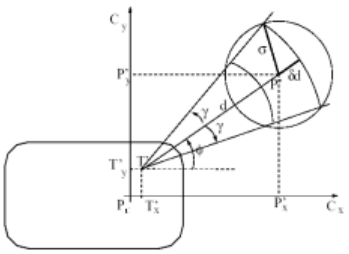

Fig. 5. Telemeter's uncertainty

3.1. Environment model

We use an ideal 2D world map of the environment. Walls and obstacles are represented as polygonal lines and each line segment is parametrized either by $(M,\, l \geq 0,\, \rho \geq 0,\, \varphi \in [-\pi, \pi[)$ or by $(P_1,\, P_2)$ as shown in figure 4.

3.2. Vehicle model

The vehicle configuration is denoted $\mathbf{x} = (x_c, y_c, \theta)^T$ where $(x_c, y_c) = P_c$ are the coordinates of a characteristic point P_c which is located midway between the two rear wheels and θ is the vehicule's orientation. All variables are defined with respect to the global frame. The control inputs are composed by both longitudinal and rotational speeds or equivalently by the incremental longitudinal (rotational) motion Δs ($\Delta\theta$) during one time period (Δt). The robot model is an approximation of the continuous case when the integration is done using finite differences. The evolution of the robot state is thus written as follows :

$$\begin{aligned} \widehat{\mathbf{x}}\,(t+\Delta t) &= f\,(\mathbf{x}\,(t)\,, \Delta s, \Delta\theta) \\ &= \begin{pmatrix} x(t) + \Delta s \cdot \cos(\theta(t) + \Delta\theta/2) \\ y(t) + \Delta s \cdot \sin(\theta(t) + \Delta\theta/2) \\ \theta(t) + \Delta\theta \end{pmatrix} \end{aligned} \quad (1)$$

We compute Δs and $\Delta\theta$ from the rotational movement of the two rear wheels :

$$\Delta s = \frac{\Delta ard + \Delta arg}{2}$$

$$\Delta\theta = \frac{\Delta ard - \Delta arg}{2e}$$

Where Δarg represents the left and Δard the right wheel displacement; $2e$ is the distance between the two wheels.

3.3. Telemetric sensor model

Our experimental vehicle is equipped with a belt of seven telemetric sensors. The output of each sensor is time t, which represents the time elapsed between the emission and the reception of a sound signal. Knowing the sound speed and the location T' of the sensor on the robot, one deduces the distance d (d cannot be beyond the scope of the sensor) to the detected obstacle and next the rough position $P' = (P'_x, P'_y)$ of the obstacle detected point. In fact, we are not sure of the obstacle location, we only know that it lies in a cone whose aperture is γ (in our case γ= 25°) and at distance $d - \delta d \leq d' \leq d + \delta d$ (figure 5).

Thus, the position of the impact point P', described by a cone sector, can be included by an uncertainty ellipsoid [11] of large axis $\sigma_\gamma = 2d.\sin\frac{\gamma}{2}$ and of small axis $\sigma_d = \delta_d$

$$\mathbf{x}_{P'}^T \begin{bmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\gamma^2 \end{bmatrix}^{-1} \mathbf{x}_{P'} = 1 \tag{2}$$

We simplify the data interpretation process by approximating [5] the covariance by a circular variance σ which includes the whole arc shaped region (figure 5) which is larger than any error:

$$\sigma = \sqrt{2d\,(d+\delta d)\,(1-\cos\gamma) + (\delta d)^2} \tag{3}$$

3.4. The Extended Kalman Filter

3.4.1. Configuration uncertainties

$\widehat{\mathbf{x}}_k$ represents the $\mathbf{x}$ configuration estimation at time k and $\mathbf{P}_k$ is its associated covariance matrix:

$$\mathbf{P}_k = E\left((\mathbf{x}_k - \widehat{\mathbf{x}}_k)\,(\mathbf{x}_k - \widehat{\mathbf{x}}_k)^T\right) \tag{4}$$

Given a $\mathbf{P}_k$ covariance matrix, we represent the uncertainty with an ellipsoid whose center is $\widehat{\mathbf{x}}_k$:

$$(\mathbf{x}_k - \widehat{\mathbf{x}}_k)^T\,\mathbf{P}_k^{-1}\,(\mathbf{x}_k - \widehat{\mathbf{x}}_k) \leq \chi^2 \tag{5}$$

Where $\chi^2 = -2\log(1-P_r)$ [11] enlarges the ellipsoid in order to be sure with a probability P_r that every point is located inside the ellipsoid. The ellipse which represents the distribution of errors is a two dimensional section of the ellipsoid ($\widehat{\mathbf{x}}_k$ is reduced to its x and y components). Then the orientation uncertainty is given by:

$$\Theta_k = \left\{\theta_k \in]-\pi,\pi]\ mod(2\pi), \left|\theta_k - \widehat{\theta}_k\right| \leq \sqrt{(\mathbf{P}_{3,3})_k \cdot \chi^2}\right\} \tag{6}$$

3.4.2. Prediction using proprioceptive sensor

The prediction equation which uses proprioceptive (odometric) measurement is given by the robot model (eq. 1). We need to compute the new uncertainty matrix $\mathbf{P}_k$ with the help of $\mathbf{P}_{k-1}$ and $\mathbf{Q}_{k-1}$. Assuming that $\widehat{\mathbf{x}}_{k-1}$, Δs and $\Delta\theta$ are not correlated, and due to the non-linearity of f, $\mathbf{P}_k$ can be calculated by a first order Taylor expansion:

$$\mathbf{P}_{k/k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1/k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{7}$$

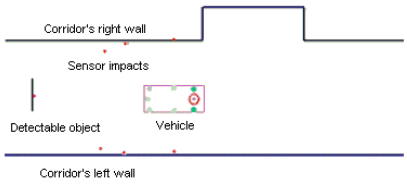

Fig. 6. Minicar environment

with:

$$\begin{aligned} \mathbf{F}_{k-1} &= \left(\frac{\partial f}{\partial \mathbf{x}}\right)_{\mathbf{x}=\widehat{\mathbf{x}}_{k-1}} \\ &= \begin{bmatrix} 1 & 0 & -\Delta s \cdot \sin\left(\widehat{\theta}_{k-1} + \Delta\theta/2\right) \\ 0 & 1 & +\Delta s \cdot \cos\left(\widehat{\theta}_{k-1} + \Delta\theta/2\right) \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{8}$$

3.4.3. Estimation using exteroceptive sensor

Each exteroceptive measurement is associated to a segment by a classical matching process. Firstly, the obstacles are enlarged with both measurement uncertainty and robot position uncertainty. Secondly, the measurement is matched with the segment if it is included in the enlarged segment. Such a process is not detailed in this paper (for an explanation see [5][10]). We use each ultrasonic sensor independently and we iterate the Kalman process [4] once for each sensor used so that the $\mathbf{K}_k$ computation is simplified to a scalar inversion. The updated estimation of $\mathbf{x}_k$ and its associated covariance matrix is given by:

$$\begin{aligned} \widehat{\mathbf{x}}_{k/k} &= \widehat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k\,(\mathbf{y}_k - \widehat{\mathbf{y}}_{k-1}) \\ \mathbf{P}_{k/k-1} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\,\mathbf{P}_{k/k-1} \end{aligned} \tag{9}$$

where the Kalman gain, $\mathbf{K}_k$ provides the confidence that we give to the measurement:

$$\mathbf{K}_k = \mathbf{P}_{k/k-1}\mathbf{H}_k\left(\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^T + \mathbf{S}_k\right)^{-1} \tag{10}$$

with the observation matrix:

$$\mathbf{H}_k = \frac{d\left(h_k\left(\widehat{\mathbf{x}}_{k/k-1}\right)\right)}{d(\mathbf{x}_k)} \tag{11}$$

the non linear equation measurement $y_k = h_k(\mathbf{x}_k) + \mathbf{w}_k$ and the covariance matrix of the noise $\mathbf{S}_k$.

4. Experimental Results

We now are going to present our first experiments, aiming at jointly testing the experimental vehicle and the localization algorithm described in the preceding sections.

4.1. User interface

The graphical user interface (figure 8) was developed in the C language using OpenGL.

The main window shows (figure 6):

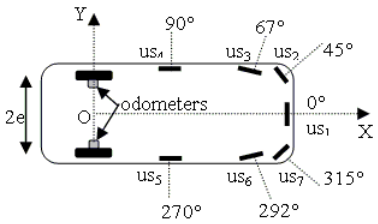

Fig. 7. Ultrasound sensors positions

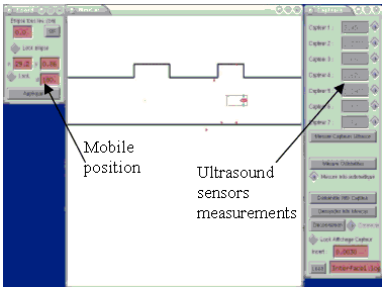

Fig. 8. User environment

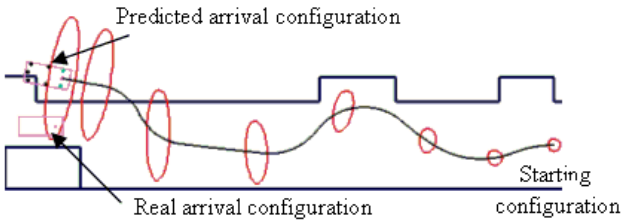

Fig. 9. Mobile position estimated by the prediction part of the EKF

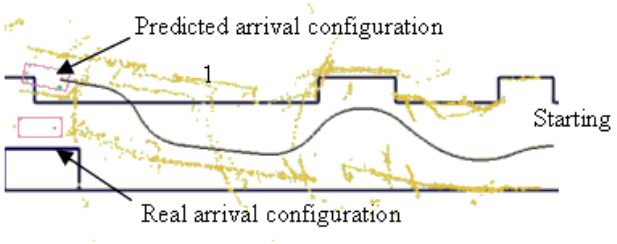

Fig. 10. Telemetric measurements obtained during the displacement

- the robot, symbolized by a rectangle with a point drawn at the middle between the two rear wheels.
- the sensors impacts.

In a first stage, we checked the correct operation of the ultrasound sensors in the charted environment represented in the figure 6.

4.2. Using of the EKF predictive part

In a second stage, we proceeded to ultrasound measurements while the vehicle moved within the corridor at a speed of 10cm/s. Localization was realized according to odometers: ultrasound sensors measurements was only taken for evaluation purpose. The ultrasonic sensors were positioned in accordance with figure 7. The mobile was initially placed on the right of the corridor, as it is showed in the figure 9. The curve represents the trajectory of the mobile calculated thanks to the odometric data treated by the predicted part of the EKF. The uncertainty ellipses along the trajectory represent the ignorance of the real position of the mobile.

We can notice that this curve, deferred in the global reference, deviates and crosses a wall. Such a localization error is due to the wheels slipping on the ground and to the imprecision made on the measurements of the distance between the two rear wheels and the wheels diameters.

Thus, on a fifteen meters course, the estimated ending point is located at a little more than one meter far from the real mesured ending point of the experimental vehicle course (figure 12).

Measurements of the ultrasonic sensors, which were not used in this phase, were however taken each second. They are represented on figure 10 and positioned according to the predicted position. We can thus discover a form that coarsely corresponds to the corridor and follows the moving vehicle. Unfortunately, we can note that a great part of these points corresponds to erroneous data and does not represent any charted element.

Among these erroneous points, the presence of two lines of parallel sensor impacts, (zone 1 of figure 10) can be explained by the sensor aperture angle (see §3.3). Indeed, the supposed impact points are placed in the sensors' axe whereas the real position of the impact is in the uncertainty zone. The presence of aberrant points is not only due to the uncertainty of the results returned by the sensors but also to the presence of foreign elements not represented in the chart. Thus, the developed matching function should only consider a relevant part of telemetric measurements, and eliminate the others considered as wrong.

4.3. Joint use of prediction and estimation

By integrating exteroceptive measurements within the previous process of localization (§ 4.2), we hope to correct a part of the errors on the mobile position. Contrary to the last figure, the trajectory of figure 11 does not meet any wall, which makes it realistic.

The points that were used for localization are represented by a darker color with a larger diameter than those considered as erroneous. We can notice that a great part of the points have not been used for the localization of the mobile: they were considered as aberrant. These points represent 30 to 40% of the obtained measurements.

The rectangle in the top of figure 12 (predicted position) makes it possible to precisely visualize the final position of the mobile and its uncertainty ellipse when we only use the odometers. The rectangle in dotted lines corresponds to the real position of the mobile measured manually at the end of the handling. The last rectangle corresponds to the estimated mobile position when we use both odometric and telemetric measurements. We can notice

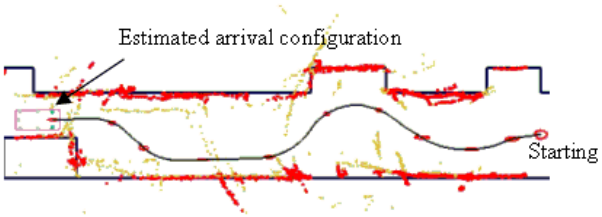

Fig. 11. Integration of sensors measurements in the process of localisation

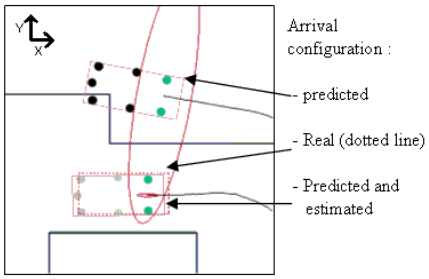

Fig. 12. Final configurations comparison

that the two uncertainty ellipses are realistic because they both integrate the real mobile final position.

Table II gives us the differences between the predicted position and the real one and between estimated and real position. These variations are calculated starting from the points of table I. When we use the exteroceptive measurements, we can notice a great improvement on the precision of the final position compared to the single use of the odometers.
Let us note that the most significant improvement was carried out according to the Y axis. This is due to the number of samples that have been put in correspondence with the horizontal segments of the chart. The uncertainty ellipse on the final position is very flattened and corroborates these results. The significant correction on the angle may be explained by the same reasons. On the other hand, along the X axis, the improvement is less perceptible. This is due to the fact that the number of samples having been used to readjust the mobile along this axis remains weak because of the corridor geometry and the trajectory that the mobile followed.

5. Conclusion and prospects

In this paper, a description was made of the localization experimental vehicle. As the tests showed it, the mobile is operational and a first localization method had been implemented. In a few time, three additional methods (error-bounded method [8], probability grid and particulate filtering) will be tested and compared in real environment. Futhermore, low level actions reflexes are going to be implemented within the embarqued electronic and will allow the mobile to avoid collision by following another path.

Position	Real	Predicted	Estimated
X(m)	19.46	19.65	19.52
Y(m)	1.18	2.15	1.16
θ(m)	180	169.6	180.4

TABLE I
FINAL VEHICLE CONFIGURATION COORDONATES

Position	Predicted	Estimated
ΔX (cm)	19	-6
ΔY (cm)	97	-2
$\Delta\theta$ (cm)	11.4	0.4

TABLE II
POSITION VARIATION

References

[1] Borenstein J., Everett B., Feng L., ”Navigating Mobile Robots: Systems and Techniques”, A. K. Peters, Ltd., Wellesley, MA, (1996).
[2] Bouaziz S., Fan M., Lambert A., Maurin T., Reynaud R., ”PICAR: experimental Platform for road tracking Applications”, IEEE Intelligence Vehicles Symposium, (2003).
[3] Burgard W., Fox D.,Thrun S., ”Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids”, In Proc. of the 21st German Conference on Artificial Intelligence, Germany, (1997).
[4] Chui C.K., Chen G.: ”Kalman filtering with real-time applications”. Springer Verlag, (1990).
[5] Crowley. J. L., ”World modeling and position estimation for a mobile robot using ultrasonic ranging”, IEEE International Conference on Robotics and Automation, pp. 674-680, Scottsdale, (1989).
[6] Dellaert F., Fox D., Burgard W.,Thrun S., ”Monte Carlo localisation for mobile robots”, in Proc. IEEE Internationnal Conference on Robotics and Automation (ICRA-99), Detroit, MI (1999).
[7] Kieffer M., Jaulin L., Walter E., Meizel D., ”Robust Autonomous robot localization using Interval Analysis”, Reliable Computing, 6, 337-362, (2000).
[8] Kieffer M., Seignez E., Lambert A., Maurin T., Walter E., “Vehicle Tracking Based on Robust Bounded-Error Nonlinear State Estimation using Interval Analysis”, International Conference on Information & Communication Technologies: from Theory to Application, 2004.
[9] Lambert A., Le Fort-Piat N., ”Safe Task Planning Integrating Uncertainties and Local Maps Federations”, International Journal of Robotics Research, 19(6), (2000).
[10] Leonard J.J; Durant-Whyte H.F., ”Mobile robot localization by tracking geometric beacons”, IEEE Trans. On Robotics and Automation, pp. 376-382, 7(3), (1991).
[11] R.C. Smith and P. Cheeseman. *On the representation and estimation of spatial uncertainty*, International Journal of Robotics Research, pp.56-68, 5(4), winter, 1986.