Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems

Multi-sensors Localization for Control of Autonomous Mobile Robot

H. Medromi, E. Zaafrani, E. Dekneuvel, M.C. Thomas
University of Nice - Sophia Antipolis
Laboratoire I3S-CNRS URA 1376
250 Rue Albert Einstein , Sophia Antipolis
06560 Valbonne, FRANCE
Phone : (33) 92.94.26.09 Fax : (33) 92 94 28 96
e-mail {hicham, elias, dekneuv and mct}@alto.unice.fr

Abstract

In this paper, we discuss an approach for designing and implementing a multisensory perception system in order to control an autonomous mobile robot when uncertainty and incomplete knowledge are the major characteristics of the situated environment. The approach relies on the combination of local and absolute localization methods. The former is lowly coupled to the environment and makes intensively use of odometric sensors. When high precision is required, telemetric sensors are used, being able to deal with error localization. Then, considering that these modules should cooperate in presence of constraints like limited resources, a knowledge-based system has to be used in order to approach the initially set goals.

1 Introduction

This paper present an approach that we have developed recently for the control of mobile robots in a barely known environment. Our approach is based on a multiple-sensored perception that allows to obtain information about the state of the robot and environment. In order to achieve this, we developed two methods of localization: local and absolute.

The local localization is based on the odometric sensors which have a good precision and an error accumulation proportional to the distance.

The absolute localization is based on telemetric sensors which allows to avoid error accumulations and hence results in a good precision concerning the robot environment.

The high-level control of the different processing modules is handled by a **propositional theorem prover** which is able to provide on-board reactive decisional capabilities. Execution control of predefined plans [1] leads to the development of generic methods called plan families. Such plans use a search space represented by an AND/OR graph, triggering primitive actions which are to be sent to the robot, and then, achieving given goals or reacting to particular events in the best way according to the current state of information.

This paper is organized as following: section 2 presents our perception system and the different localization methods; in section 3, we expose how the control problem is tackled; then, in section 4, some of the significant results we obtained from this approach are detailed; finally, in section 5, we conclude on the major benefits of our works and we talk about the extensions and problems we are currently working on.

2 Localization methods

The navigation defines [5] itself in a general manner like art to guide itself towards a final destination under certain constraints. For this, it is imperative to know at all times and as precisely as possible its localization and its road [12, 4]. It's why many techniques have been developed in using conjointly local localization and absolute localization.

Local localization consists of evaluating the position, the orientation, and eventually the speed through integration of information provided by diverse sensors since the starting position [17, 21].

[1] often referenced as *reactive planning*

0-7803-3700-X/96 $5.00

Absolute localization is the technique permitting it to find its way directly in the domain of the evolution of the mobile, be it the sea, the earth, or space [6, 21, 19, 18].

2.1 Mobile robot models

We reduce our study to robots having mobility constraints, that represent a large class.

Unicycle mobile robots : these are robots formed by a platform and two fixed independent motor coaxial wheels (figure 1).

We will consider that the wheels angular velocities $\dot{q_1}$ and $\dot{q_2}$ can be taken as control variables. By setting:

$$U = \begin{pmatrix} \dot{q_1} \\ \dot{q_2} \end{pmatrix}$$

Let us thus consider the following state vector:

$$X = \begin{pmatrix} x_c \\ y_c \\ \theta_c \end{pmatrix} \quad (1)$$

where x_c and y_c are the coordinates of the vector $\vec{OC}$ in the frame R_0 and θ_c is the cart's orientation angle.

We then obtain the following state equation :

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{pmatrix} = \begin{pmatrix} \frac{R}{2}cos\theta & \frac{R}{2}cos\theta \\ \frac{R}{2}sin\theta & \frac{R}{2}sin\theta \\ \frac{R}{2e} & \frac{-R}{2e} \end{pmatrix} \begin{pmatrix} \dot{q_1} \\ \dot{q_2} \end{pmatrix} \quad (2)$$

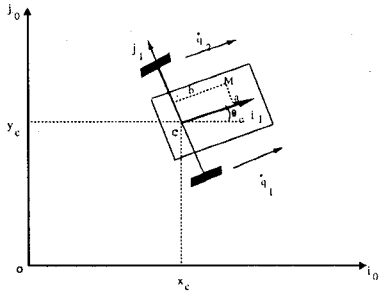

Figure 1: Configuration space of mobile robot

We could have a simpler representation than that given by (2) because of the fact that the trigonometric functions sin(.) and cos(.) will disappear from the new writing of the system.

Be it the mobile frame $R_1(i_1, j_1, k)$ attached to the robot and a point M other than the base, of coordinates (a,b) in the mobile frame like in (figure 1). Be they x, y the coordinates of the vector $\vec{MO}$ expressed in the mobile frame R_1. Let's take as command variables v and $\dot{\theta}$ [20, 15, 17]. Be it :

$$\vec{MO} = x\vec{i_1} + y\vec{j_1} \quad (3)$$

The representation of the state will be given in the following manner :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -1 & y+b \\ 0 & -(x+a) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \dot{\theta} \end{pmatrix} \quad (4)$$

The representation of state obtained is in the form $\dot{X} = B(X)U$. This representation is without terms of drift (A(X,t)) which appears through the introduction of a desired mobile situation.

2.2 Local localization

In this paragraph, we border upon the problems of localization of the mobile robot in a local context (local observability) or more precisely in a very limited neighborhood that contains the trajectories of the state space of the mobile robot.

This localization has a role to permanently supply an estimation of the real position of the robot, in a manner that allows it to be guided. The method proceeds through integration of elementary displacements, like inertial methods. But, for these last ones, we have preferred for cost reasons and as a means of experimentation, a method which use as measures, the rotation of two coaxial wheels of the mobile robot. These wheels have incremental optical codes that deliver successive measures T_{n+1} and T_n between two instants, the angular position of two wheels.

2.2.1 State equation of mobile robot

The choice of state vector is an important step in the placement of the estimator. We have chosen the state vector X that follows :

$$X = \begin{pmatrix} q_1 \\ x_c \\ y_c \\ \theta_c \end{pmatrix}$$

The estimator must apply itself to the trajectories primarily unknown. The model must therefore also be as general as possible but remaining in an acceptable complexity.

We have chosen a model of displacement at a constant linear speed for a range of steps. This model therefore authorizes all the trajectories to a constant curved ray

(this includes the rectilinear trajectories).

The state equation is in this case, a non-linear discreet equation :

$$\begin{cases} X_k = f_{k-1}(X_{k-1}) + g_{k-1}(X_{k-1})W_{k-1} \\ Y_k = h_k(X_k) + V_k \end{cases} \quad (5)$$

$\{X_k\}$, $\{Y_k\}$, $\{W_k\}$, $\{V_k\}$ taking their values in R^4, R^4, R^1, R^4, and

$$f_k, g_k, h_k : R^4 \longrightarrow R^4, R^{4\times 1}, R^4,$$

respectively. The functions f_k and h_k are derivable in relation to the variable X. $\{W_k\}$ and $\{V_k\}$ are the white noises (of respective covariants Q_k^w and Q_k^v) independent among them and independent from the initial condition X_0. The function of transition f_{k-1} is the following :

$$\begin{cases} q_{1k} = q_{1k-1} + \Delta q_{1k-1} \\ x_{ck} = x_{ck-1} + \Delta S_{k-1}\cos(\theta_{ck-1}) \\ y_{ck} = y_{ck-1} + \Delta S_{k-1}\sin(\theta_{ck-1}) \\ \theta_{ck} = \theta_{ck-1} + \Delta\theta_{ck-1} \end{cases} \quad (6)$$

and g_k is given by :

$$g_k(X_k) = \begin{pmatrix} \Delta q_{1k-1} \\ 0 \\ 0 \\ \Delta\theta_{k-1} \end{pmatrix} \quad (7)$$

thus Y_k is the observed vector,

$$\begin{pmatrix} q_{1k} \\ q_{2k} \\ \Delta q_{1k} \\ \Delta q_{2k} \end{pmatrix} \quad (8)$$

q_1 and q_2 are expressed in radians and result directly from the relations as $\Delta S = \frac{R(\Delta q_1 + \Delta q_2)}{2}$ and $\Delta\theta_c = \frac{R(\Delta q_1 - \Delta q_2)}{2e}$.
V_k is the noise of observation, white, Gaussien, and from the covariant ${Q_k}^w$.

For the system (5), the obtained property of observability is local. In particular, the solution process of (5) is not Gaussien, its moments cannot be calculated in a simple manner.

2.2.2 Local localization method

The system coefficients depend upon observations, therefore we can utilize the estimator of X_k : $\hat{X}_{k-1}$ for the prediction state and $\hat{X}_k^-$ for the estimation state.

For the prediction step (at instant k), we are using the linearization :

$$f_k(X) \simeq f_k(\hat{X}_{k-1}) + \nabla f_k(\hat{X}_{k-1})(X - \hat{X}_{k-1})) \quad (9)$$

for the estimation step (at instant k) we are using the following linearization :

$$g_k(X) \simeq g_k(\hat{X}_k^-) \quad (10)$$

$$h_k(X) \simeq h_k(\hat{X}_k^-) + \nabla h_k(\hat{X}_k^-)(X - \hat{X}_k^-)) \quad (11)$$

The stochastical local localization is realized in two steps :

1. step of prediction :

$$\begin{cases} \hat{X}_k^- = f_k(\hat{X}_{k-1}) \\ R_k^- = F_{k-1}R_{k-1}F_{k-1}^* + G_{k-1}Q_{k-1}^w G_{k-1}^* \end{cases} \quad (12)$$

with : $F_{k-1} = \nabla f_k(\hat{X}_{k-1})$, $G_{k-1} = g_{k-1}(\hat{X}_{k-1})$

2. step of estimation :

$$\begin{cases} K_k = R_k^- H_k^*[H_k R_k^- H_k^* + Q_k^v]^{-1} \\ X_k = \hat{X}_k^- + K_k[Y_k - h_k\hat{X}_k^-] \\ R_k = [I - K_k H_k]R_k^- \end{cases} \quad (13)$$

with : $H_k = \nabla h_k(\hat{X}_k^-)$

The estimator is initialized thanks to a first measure. An uncertainty Q_x is given for the initial state (initial x_c, y_c, θ_c). The first measure allowed to initialize as ΔS and $\Delta\theta_c$.

2.3 Absolute localization

In this case, the position of the robot is determined, neither from an earlier estimation, but rather in relation to fixed marks and known positions. We thus avoid a strong dependence on the measures of the mechanics of the mobile robot in avoiding thus the accumulation of successive errors.

Several systems, based on the marking of the environment by signposts, were experimented with to obtain this absolute localization. But they required a previous preparation of this environment and the risk remains of masking some signposts by obstacles. This

is why, we have developed methods utilizing the obstacles themselves as signposts (walls, corners, edges, machines) [2, 3, 14, 8]. The robot is therefore equipped with the telemetric sensors to give a structured environment as described in [16]. We present in this paragraph the approach which was developed in this context.

2.3.1 Observation model

We consider the telemetric sensors in a structured environment made up of the different geometries (walls, corners, edges). From the approximation of Hugen , we obtain the observation model (figure 2) that follows :

$$Y = h(X) = \frac{xsin\theta + ycos\theta}{cos\theta} \tag{14}$$

thus :

$$Y = d = h(X) = xtan\theta + y$$

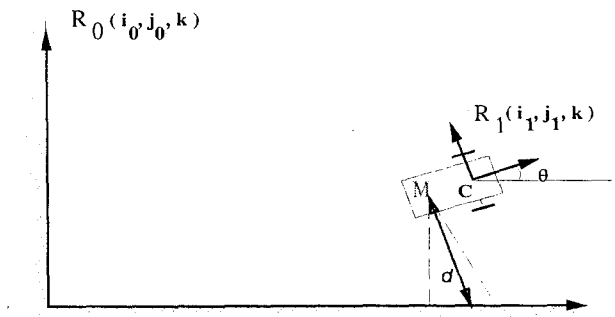

Figure 2: Observation model

with the model of mobile robot in the mobile frame (4). The observation vector is given by three successive measures, associated with instants t_k, t_{k-1} et t_{k-2}. We put therefore :

$$\begin{pmatrix} y(X_k) \\ y(X_{k-1}) \\ y(X_{k-2}) \end{pmatrix} \tag{15}$$

Therefore we can write the equation which link the measure and state at the same instant:

$$Y_k = \begin{pmatrix} h_k(X_k) \\ h_{k-1}(X_{k-1}) \\ h_{k-2}(X_{k-2}) \end{pmatrix} \tag{16}$$

1. reverse integration

We must now obtain the equation of measure function of the natural state, that is to say, of X_k. It's for this reason that we have effected a reverse integration. These allow us to evaluate the approximations of the preceding states, noted X^*_{k-1} et X^*_{k-2}:

state ${X_{k-1}}^*$:

$$\begin{cases} x^*_{k-1} = x^*_k - V^*_k[t_k - t_{k-1}]\cos(\theta^*_k) \\ y^*_{k-1} = y^*_k - V^*_k[t_k - t_{k-1}]\sin(\theta^*_k) \\ \theta^*_{k-1} = \theta^*_k - \dot{\theta}^*_k[t_k - t_{k-1}] \end{cases} \tag{17}$$

state ${X_{k-2}}^*$:

$$\begin{cases} x^*_{k-2} = x^*_{k-1} - V^*_{k-1}[t_{k-1} - t_{k-2}]\cos(\theta^*_{k-1}) \\ y^*_{k-2} = y^*_{k-1} - V^*_{k-1}[t_{k-1} - t_{k-2}]\sin(\theta^*_{k-1}) \\ \theta^*_{k-2} = \theta^*_{k-1} - \dot{\theta}^*_{k-1}[t_{k-1} - t_{k-2}] \end{cases} \tag{18}$$

2. observation model
We use this approximation in the measure equation (16), we can write :

$$Y_k \approx \begin{pmatrix} h_k(X_k) \\ h_{k-1}(X^*_{k-1}) \\ h_{k-2}(X^*_{k-2}) \end{pmatrix} \tag{19}$$

then, we use (17) and (18), we obtain the follow measure equation :

$$Y_k \approx \begin{pmatrix} h_k(X_k) \\ h^*_{k-1}(X_k) \\ h^*_{k-2}(X_k) \end{pmatrix} \tag{20}$$

We define the follow non-linear function :

$$\mathcal{H}(X) \begin{array}{c} R^3 \longrightarrow R^3 \\ X \longrightarrow \mathcal{H}(X) \end{array} \tag{21}$$

with :

$$\mathcal{H}(X) = \begin{pmatrix} h_k(X) \\ h^*_{k-1}(X) \\ h^*_{k-2}(X) \end{pmatrix} \tag{22}$$

and the measure equation have the following form:

$$Y_k = \mathcal{H}(X_k) + noise \tag{23}$$

The choice of state vector is an important step in the placement of the estimator. We have chosen the state vector X that follows :

$$X = \begin{pmatrix} x_c \\ y_c \\ \theta_c \end{pmatrix}$$

2.3.2 State equation of mobile robot

The estimator must apply itself to the trajectories primarily unknown. The model must therefore also be as general as possible but remaining in an acceptable complexity.
We have chosen a model of displacement at a constant linear speed for a range of steps. This model therefore authorizes all the trajectories to a constant curved ray (this includes the rectilinear trajectories).

The state equation $(x_{ck}, y_{ck}, \theta_{ck})$ is in this case a nonlinear discreet equation given by the system (6) and g_k is given by:

$$g_k(X_k) = \begin{pmatrix} \Delta x_{ck-1} \\ \Delta y_{ck-1} \\ 0 \end{pmatrix} \qquad (24)$$

thus Y_k is the observed vector given by (23).

For the system (5), the obtained property of observability is weakly. In particular, the solution process of (5) is not Gaussien, its moments cannot be calculated in a simple manner.

2.3.3 Absolute localization method

The system coefficients depend upon observations, therefore we can utilize the estimator of X_k : $\hat{X}_{k-1}$ for the prediction state and $\hat{X}_k^-$ for the estimation state. For the prediction step (at instant k), we are using the linearization given by (9,10,11). The stochastical absolute localization is realized in two steps : prediction (12) and estimation (13).

3 The control architecture

Our robot control system is based on the two localization methods described above. The architectural specification [13] of the control system lies in its ability to conciliate between the decision process and the reaction process .

The architecture of our robot control system is a hierarchical architecture. It is composed of two main levels (figure 3).

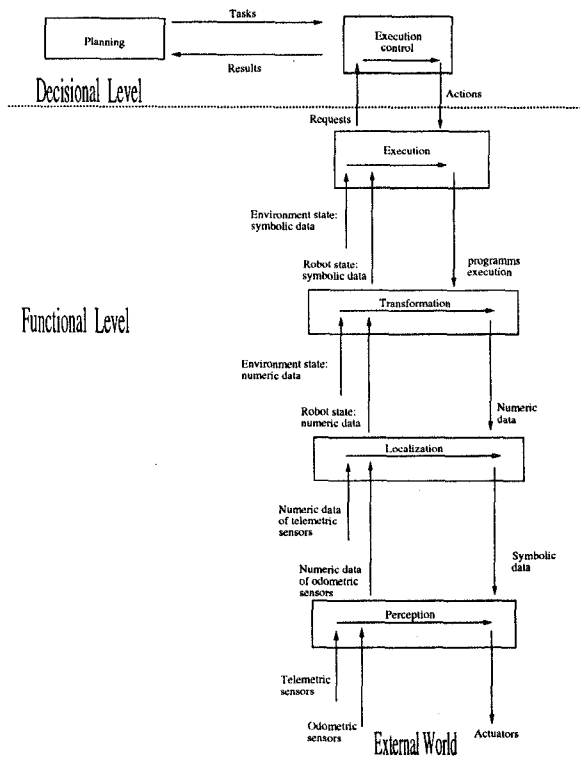

Figure 3: The robot control system

3.1 The decisional level

The robot mission is described as a set of tasks to be achieved. At this **mission planning** layer, the task decomposition is very rough and does not take into account the execution context and model of each elementary action. The planner orders tasks by a forecasting into the future; it does that without the knowledge of the real execution context, its projection is done only by the add/delete list which is the list of predicates changes. In addition, such a planner is able to deal with temporal reasoning. We will not tackle this level here ([11] for more detail).
So, between the predicted world and the real world, differences may occur and we need some flexibility to suit tasks to real world robot surrounding. This has to be done at the **execution control** layer [7, 9]. More precisely, the control is in charge of achieving each task in the best possible way decomposing it, taking into account the context and sensory feedback. It has to evaluate, in a flexible way, how successful an elementary action has been, and, depending on that evalua-

tion, pursue on the next action or adapt its plan for achieving the task to the current context. The execution control layer exchanges data with the other layers. In case the robot finds a problem, or one or more actions have not been performed properly, the controller suspends the execution of the task and asks the robot to go back to its initial state. The controller can decide to halt the system at any time during the execution of the task.
For each plan, there are some precise actions associated with it. When the planner sends a new plan to the controller, the latter chooses the corresponding actions and sends them to the functional level.

3.2 The functional level

The functionnal level, where the local and absolute localization are integrated, implements all the robot basic capabilities in sensing, acting and computing through the concept of **functional modules**. These functionalities are grouped according to data or ressource sharing, and integrated into layers. This level is composed of the execution layer, the transformation layer, the localization layer and the perception layer.

4 Experiences and results

We examine the example shown on (figure 4). The task of cleaning the supermarket consists of the robot moving from its starting position to its final destination position while passing through different well defined points. In this trip, the robot must ovoid any obstacles. It must also follow the path of the shelves whenever they exist in the environment (figure 4).

The controller Knowledge Base contains rules for deciding which actions should be taken in response to a request. The KB of the task "Go from 1 to 12" is shown below:

((Go1,2) ∧ (Go2,3) ∧ (Go3,4) ∧ (Go4,5) ∧ (Go5,6)
∧ (Go6,7) ∧ (Go7,8) ∧ (Go8,9) ∧ (Go9,10) ∧ (Go10,11)
∧ (Go11,12)) ∨ (Go_back) (exec) ⟶ (Go_from_1_to_12)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go1,2)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go2,3)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go3,4)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go4,5)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go5,6)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go6,7)
(using_local) (exec) ∨ (using_absolute) ⟶(Go7,8)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go8,9)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go9,10)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go10,11)
(using_local) (exec) ∨ (using_absolute) ⟶ (Go10,11)

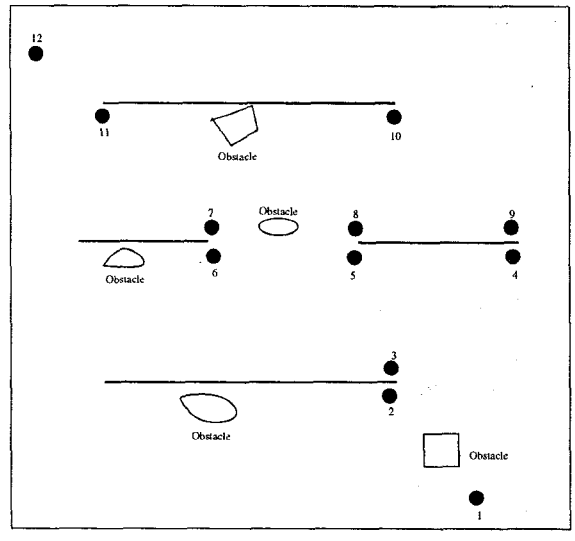

Figure 4: The robot path and the environment

(using_local) (exec) ∨ (using_absolute) ⟶ (Go11,12)
(presence_obstacle) ⟶ (using_absolute)
((follow_shelf) (exec)) ∨ ((follow_obst_right) (exec))
∨ ((follow_obst_left) (exec)) ⟶ (presence_obstacle)
(crossing_problem)? ⟶ (Go_back)
(using_local_localization)? ⟶ (using_local)
(shelf)? ⟶ (follow_shelf)
(right_obstacle)? ⟶ (follow_obst_right)
(left_obstacle)? ⟶ (follow_obst_left)

The control module receives requests (corresponding to the symbolic data) indicated below:
-(crossing_problem)
-(using_local_localization)
-(shelf)
-(right_obstacle)
-(left_obstacle)

The request (right_obstacle) means that an obstacle has been detected and that there is more free space to the right of the obstacle than to its left. According to this request, the controller chooses a sequence of actions and sends it to the execution layer. In this case, the action (follow_obst_right) is sent. This action informs the robot to avoid the obstacle by following the right side of this obstacle.

The actions are received by the execution layer which executes the corresponding program according to the programs KB. An example is shown below :
program name: path_computing
Corresponding action: Using_local
Calling syntax: local
input data: starting position and destination

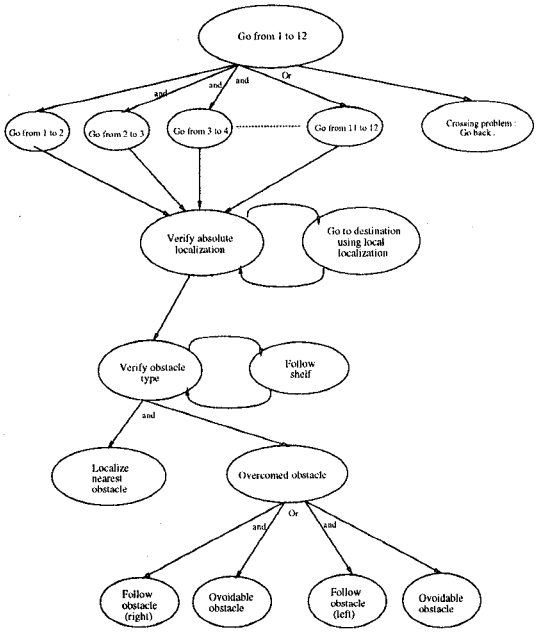

Figure 5: Graph of the task "Go from 1 to 12"

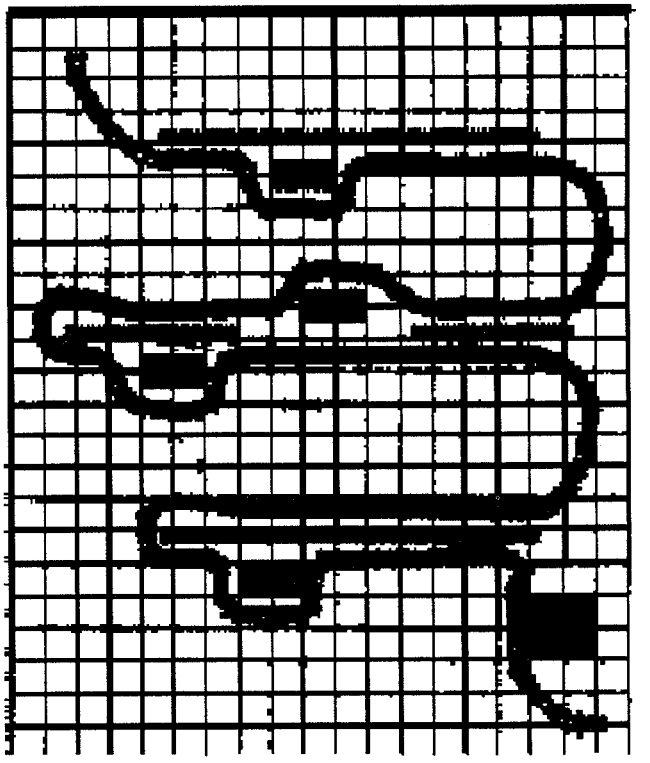

Figure 6: Experimental Results

output data: destination: Xll(x,y,θ)

The search space is represented as an AND/OR graph (figure 5). Walking the graph can be done using a program that allows to search a AND/OR decision tree [7, 10].

Our system of autonomous control based on localization was implemented on the Khepera mobile robot [1], with small wheels having a circular form about 5 cm in diameter. Its height varies according to the modules that were added to it, each one fixing itself on the card. The robot moved from 1 to 12. It avoided any obstacles by following the right or the left side of the obstacles and it followed the path of the shelves whenever they exist (figure 6).

5 Conclusion

In this paper, we presented a control system for autonomous mobile robot in a barely known environment. We tested our system on the Khepera robot; the experimental results were satisfactory. The principles used in our system are general and can be applied to a wide range of environment situation which results in the autonomy of the robot.

The local localization therefore presents a certain number of advantages, which are an independence from the method compared to the environment, and a precision independent from the situation of the robot. Moreover, the results of absolute localization are highly efficient. On the otherhand, at the time of the implementation of the extended Kalman filter, we have encountered measure asynchronism problems. To resolve this problem, we suggest a method that supposes the length between two measures Δt always synchronized and known, but we take $\Delta t = n\Delta T$ (with ΔT being the length between the two predictions). This could prove useful in the case of a function of non-linear evolution, we can refine the prediction of the measure in proceeding with many predictions of the intermediary state between two measures.

The same remarks are valid about the eventual dephasing at the origin. This solution represents, by its nature of coupling the prediction, which uses local localization, and of estimation, which uses absolute localization, from which we can speak of the method of fusion. This and another weakly coupled approach form the object of the next paper.

Currently, our system does not take into consideration the execution time of the robot actions.

Our future work includes the development of a robot control system for an unknown environment where the

robot can encounter mobile obstacles. In such situations, the robot should consider the time and hence the system must supervise complex programs in real time.

References

[1] A.Guignard, E. Franzi, and F. Mondada. Khepera a miniature mobile robot. Preprints of the third International Symposium on Experimental Robotics Kioto, Japan, oct. 93.

[2] B. Barshan and R. Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. In *In IEEE PAMI*, pages 560–569, 1990.

[3] O. Bozma and R. Kuc. Building a sonar map in a specular environment using a single mobile sensor. *Traitement du Signal*, 13(12):1260–1269, 1991.

[4] R. Chatila. Autonomie de déplacement : navigation et évitement d'obstacles. In Vers la Robotique de 3ème génération, Avril 1985.

[5] I.J. Cox. Blanche : position estimation for an autonomous robot vehicle. In Autonomous Robot Vehicles editor, editor, *In Spring Verlag*, pages 205–220, 1990.

[6] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In IEEE, editor, *Proceedings of the IEEE international conference on robotics and automation*, pages 674–680, 1989.

[7] B. Degallaix, J. Thibault, and M. Ghallab. Propositional logic applied to execution control. In *Procedings of the 1992 International conference on intelligent robots and systems, Raleigh*, pages 317–324, July 1992.

[8] C. Docarli, J.F. Le Corre, and O. Devise. Dynamic location of a mobile robot by extended kalman filter. In IROS'91, editor, *In IEEE International Symposium on Intelligent Robots and Systems*, Osaka, November 1991.

[9] E.Dekneuvel, M. Ghallab, and J.P. Thibault. Hypotheses management for scene interpretation in a multisensory perception machine. In *European Conference on Artificial Intelligence (ECA) Vienne*, January 1992.

[10] M. Ghallab and G. Escalada-Imaz. A linear control algorithm for a class of rule-based systems. *The journal of logic programming*, 1991.

[11] M. Ghallab and H. Laruelle. Representation and Control in IxTeT, a temporal planner. In *Second Artificial Intelligence Planning System (AIPS94)*, Chicago, June 1994.

[12] G. Giralt, R. Chatila, and M. Vaisset. An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots. In *Proceedings of the International Symposium on Robotics Research*, pages 191–214, Cambridge, Masachusetts, USA, 1984. MIT Press.

[13] T.C. Henderson and C. Hansen. The specification of distributed sensing and control. *J. Robotic Systems*, 2:387–396, 1985.

[14] L. Kleeman and R. Kuc. Mobile robot sonar for target localization and classification. *Robotics Research*, 14(4):295–318, August 1995.

[15] H. Medromi. Capteurs en robotique. Technical report, Cours-ESSI, Sophia-Antipolis, Sophia-Antipolis, Février 1994.

[16] H. Medromi. *Contribution à l'observation et à la localisation dynamique pour la commande en robotique mobile.* PhD thesis, Université de Nice - Sophia Antipolis. A publier prochainement, Septembre 1996.

[17] H. Medromi, J.Y. Tigli, and M.C. Thomas. Mobile robot localization by fusion of odometric and ultrasonic sensors. In *Euriscon'94 European Robotics and Intelligent Systems Conference*, Malaga, August 1994.

[18] H. Medromi, J.Y. Tigli, and M.C. Thomas. Posture estimation of mobile robots : Observers - sensors based. In *IEEE International Conference On Multisensor Fusion and Integration for Intelligent Systems*, Las Vegas, Nevada, October 1994.

[19] F. Pedrosa and M.I. Ribeiro. Mobile robot localization using ultrasonic sensors. In *Proceedings of the first Portuguese Meeting on Automatic Control'94*, Lisbon, Portugal, september 1994.

[20] C. Samson and K. Ait-Abderrahim. Mobile robot control part 1 : Feedback control of a nonholonomic wheeled cart in cartesian space. Technical Report RR 1288, INRIA Sophia-Antipolis, Sophia-Antipolis, Octobre 1990.

[21] J. Vaganay and M.J. Aldon. Sensor integration for mobile robot localization. In *ISIR'93*, pages 479–485, Tokyo, Japan, November 1993.