Fuzzy and neural control for mobile robotics experimentations

G. Bourdon, Patrick Henaff

Fuzzy and Neural Control for Mobile Robotics Experimentations

G. Bourdon, P. Hénaff
Laboratoire de Robotique de Paris
10-12, avenue de l'Europe
78140 VELIZY - FRANCE
bourdon@robot.uvsq.fr
henaff@robot.uvsq.fr

Abstract :
In this paper we present some mobile robotics experimentations we are leading in our laboratory.
The first example we illustrated has for context mobile robot cooperation. We propose a reactive method to join and accost a dynamic vehicle in the view to perform collaboration tasks. We will show how the simple form of Takagi-Sugeno fuzzy controllers permits to define the desired accosting strategy according to the relative configuration of both robots.
The second experimentation concerns the using of neural techniques to control the cartesian position and orientation of a mobile robot. The aim is to illustrate how a neural controller could be used to implement a reactive control.

1 Fuzzy Accosting Method

1.1 Problem presentation

The first experimentation of this paper concerns the problem of mobile robots collaboration. Our aim is to provide collaboration tasks between two autonomous platforms [1], each one equipped with an arm. What we are presenting here is the first step : the definition of a strategy which permits to one robot, the *Follower*, to join a second, the *Target*, and to follow it (Figure 1).

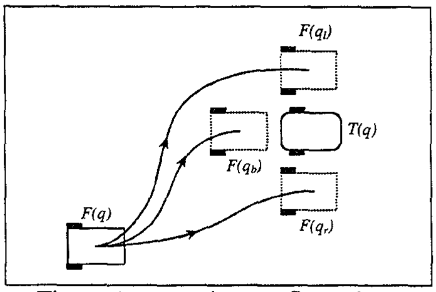

Figure 1 : accosting configurations.

Figure 1 shows the three accosting configurations *F(q)*. The current target configuration is noticed *T(q)*. The follower control point is noticed *f(q)* and the accosting point is *t(q)*.This tracking problem has been treated by several authors [2],[3],[4]. But in these cases the follower is already in the ideal tracking position, there is no accosting maneuver. In our study there is two problems to solve :

- The initial problem is to develop a tracking control law allowing to make *f(q)* join the accosting point *t(q)*. This control law has to respect the final accosting condition :

$$\begin{cases} x_f = x_t \\ y_f = y_t \, , \\ \vec{V}_F = \vec{V}_T \end{cases} \quad (1)$$

  with (x_f, y_f) and (x_t, y_t) respectively the position of points *f* and *t*, $\vec{V}_F$ and $\vec{V}_T$ are the velocity vectors of both mobiles.
- The definition of such a control law is not sufficient because of the collision risk between both vehicles. The configuration space of the target has to be on line rebuilt in order to modify the follower trajectory. At each sampling time this condition can be resumed by $F(q) \cap T(q) = \varnothing$.

Experimental conditions

Two mobile robots are necessary to realize the experimentation. The target is a ROBUTER II (ROBOSOFT) with a maximum velocity about 1m/s. It can be pilot by an operator with a joystick. So its trajectory is a priori totally unknown. The follower robot, ROMAIN, is an autonomous robot we built in our laboratory [5], its maximum velocity is about 4m/s. Both robots have approximately the same size (1m long per 0.8m large). The Robuter is equipped with a High Frequency Emitter. The target can send its position (x_t, y_t) with sample time of 10ms. For the reception of this information, ROMAIN, is equipped with a HF Receiver.

1.2 Tracking with Rendez-Vous Point prediction

Our tracking method is based on prediction guidance law [6]. At each sampling time t_i a pursuit vector $\vec{V}_P$ determines the rendez-vous point $P(t_i)$ as the intersection with target velocity direction. We obtain a succession of points *P* which constitute the set of predicted Rendez-Vous points (Figure 2).

0-7803-3755-7/97/$5.00 ©1997 IEEE

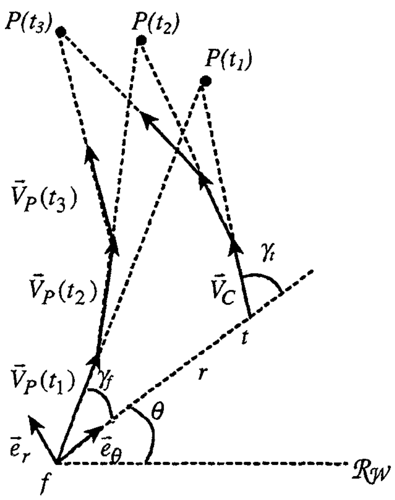

Figure 2 : Rendez Vous point prediction.

$\vec{V}_P$ has for components (γ_P, V_P). The idea is to compute each sampling time these components which can then give the instantaneous orientation and velocity of the robot. Let $\vec{L}$ the Line Of Sight (LOS) vector, the tracking law results from the relation $\vec{V}_P = \vec{V}_C + \vec{L}$ (Figure 3).

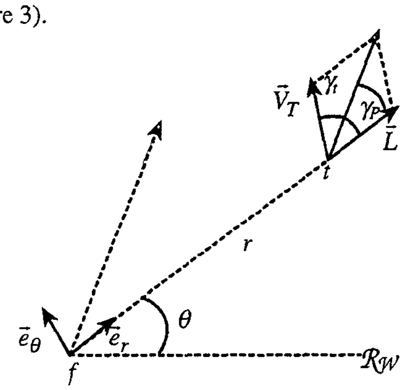

Figure 3 : tracking vector computation.

If r the distance between f and t the final conditions described in (1) implies for the pursuit vector : $\lim_{r\to 0} \vec{V}_P = \vec{V}_C$. This last Relation implies then $\lim_{r\to 0} \vec{V}_C + \vec{L} = \vec{V}_C$, that means $\lim_{r\to 0} L = 0$.

Considering figure 3, the target velocity can be expressed in the mobile frame as $\vec{V}_C = V_{C_r}\vec{e}_r + V_{C\theta}\vec{e}_\theta$.

$\vec{V}_P$ components can be written :

$$\gamma_P = arctg\left(\frac{V_{C\theta}}{V_{C_r} + L}\right),\ V_P = \sqrt{\left(L + V_{C_r}\right)^2 + V_{C\theta}^2}\ .$$

The projection V_{C_r} and $V_{C\theta}$ are computed each sampling time according to the received target position. γ_P and V_P are depending on the L parameter. A LOS controller gives according to the distance r the value of this parameter. This is a Takagi-Sugeno fuzzy controller with two rules :

if r is *NUL* then $L=0$
if r is *BIG* then $L=L_{max}$.

The fuzzy subsets of the input variable are : $\mu_N(r) = L(r,0,r_{\max})$ and $\mu_G(r) = \Gamma(r,0,r_{\max})$.

The rules expression of this controller allows to ensure the condition $\lim_{r\to 0} L = 0$.

The analytical model of this LOS controller is :

- if $r \geq r_{\max}$ $\quad L = L_{\max}$
- if $r \leq r_{\max}$ $\quad L = \mu_{BIG}(r) \cdot L_{\max}$

The tracking vector components become :

$$V_P = \sqrt{\left(\mu_{BIG}(r).L_{\max} + V_{C_r}\right)^2 + V_{C\theta}^2}$$

and $\gamma_P = arctg\left(V_{C\theta} / \left(V_{C_r} + \mu_{BIG}(r).L_{\max}\right)\right)$.

The experimentation implies to respect the saturation condition $V_P \leq V_{\max}$. So, from the V_P expression we deduce the value of L_{max}, function of the maximal linear follower velocity and the target velocity components : $L_{\max} = -V_{C_r} + \sqrt{V_{\max}^2 + V_{C\theta}^2}$.

There is so an on line adaptation of the universe of discourse of L. Figure 4 resumes the principle of this point tracking method.

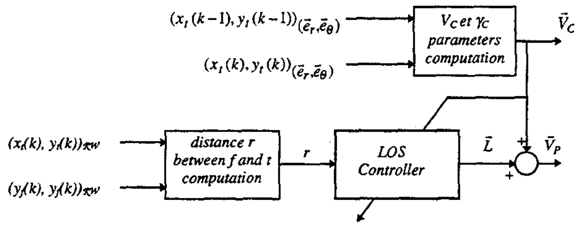

Figure 4 : tracking scheme.

Experimentation

The following figure present an experimentation in a ideal relative configuration. That means that both robot configurations and trajectories are such that there is no collision risk. The defined tracking method is in this case sufficient and has been implemented on ROMAIN. Figure 5 illustrates the evolution of the robots consecutive instants for a rear accosting.

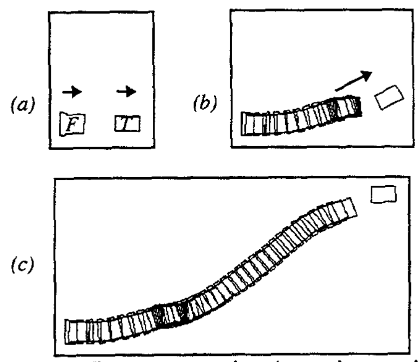

Figure 5 : rear accosting (experimentation)

Figure 6 shows the linear velocities during the motion. We can see the stop of the target corresponding to Figure 5.b, and the follower behavior.

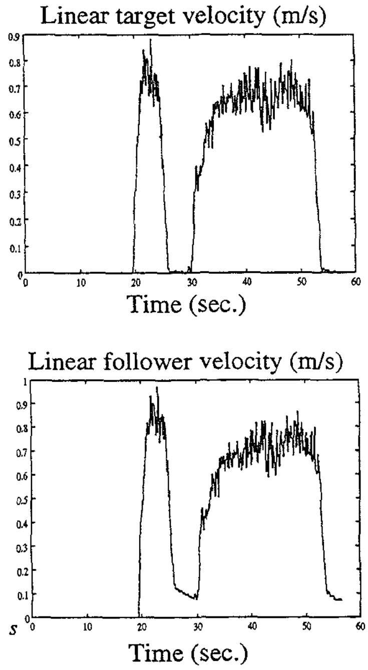

Figure 6 : linear velocity of T and F.

1.3 Non ideal relative configurations

As we said before, the tracking method we developed can be applied only in specific cases where there is no collision risk. But most situations are such that the intersection of spaces configuration are not empty : $F(q) \cap T(q) \neq \emptyset$. The configuration $T(q)$ have to be taken into account. Furthermore, the accosting side have also to be consider, and can indicate how avoid the collision with the target without losing the goal.

Fuzzy Modelisation Field

We introduce the concept of Fuzzy Modelisation Field (FMF) issued from the artificial potential field [7]. Our aim is, at each time of the process control to compute an envelope representing the geometrical configuration of the target. This envelope built on the repulsive potentiel concept, has to be function of the side to be accosted. Its specific influence zone will be then disymetric, if relative position of both robots is considered (Figure 7).

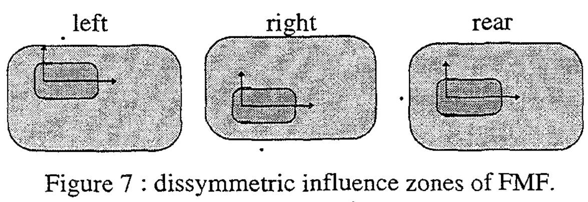

Figure 7 : dissymmetric influence zones of FMF. (funcion of accosting side)

Such a repartition of the target field is obtained by a fuzzy controller. This FMF Controller gives on line the magnitude M of the field at the position of the follower control point f. This magnitude is normalized between 0 and 1 wich correspond to the dicret values of the Sugeno's rules (Figure 8).

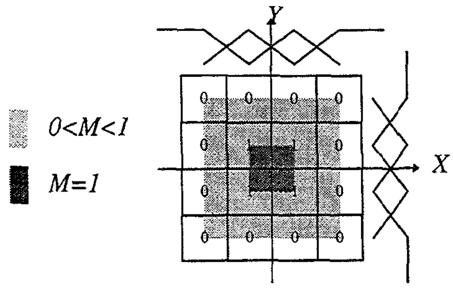

Figure 8 : rules of the FMF Controller.

The influence zone is modify according to the accosting side with the adaptation of the central values of the inputs membership functions. The following example (figure 9) is the result of the FMF computation for the entire specific influence zone for a right accosting.

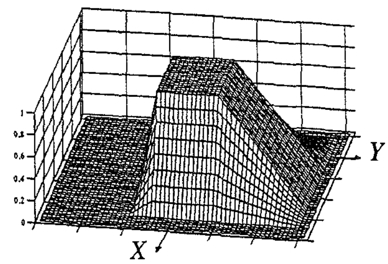

Figure 9 : FMF for a right accosting.

Fuzzy Round Forces

We can now detect when the follower is entering the FMF. We must then impose, considering point t and the relative configurations, the direction allowing to avoid the collision. This directions are resumed in figure 10 for each side. We see here that such tangential forces can't be determined by a classical potentiel field method. Furthermore a same line of field must be composed of forces with adverse directions.

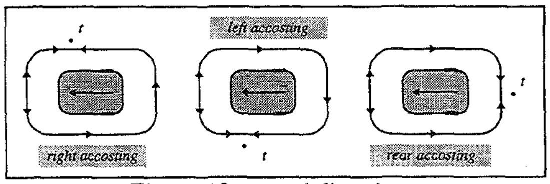

Figure 10 : round directions.

We introduce the Fuzzy Round Forces wich are a priori defined by a Takagi-Sugeno fuzzy controller. Memberships of the input universe of discourse are the same as for the FMF. For a security question, we complete the set of forces by fuzzy repulsive forces in the domaine where M=1. That means when collision risk is high the follower behavior must be like a flight. Figure 11 shows the rule table of the fuzzy round forces for a right accosting.

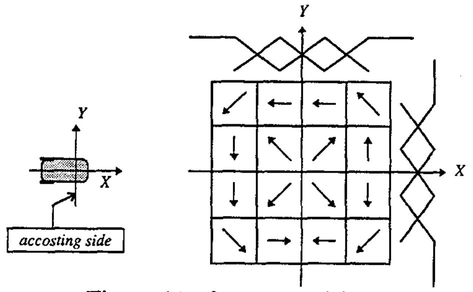

Figure 11 : fuzzy round forces.

The result of the inference procedure gives the force field Figure 12.

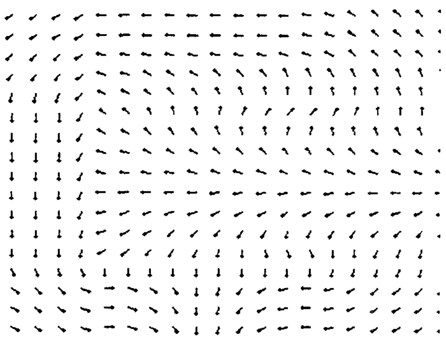

Figure 12 : force field.

1.4 Accosting Force

We have now two kinds of vectors computed each sampling time. $\vec{V}_P$ is the tracking vector, result of the predicted point tracking law. It is only referred to a point and doesn't take care about collision between robots. The second kind of vector is the round force, noticed $\vec{F}_r$. Its norm is given by the Fuzzy Modelisation Field $\|\vec{F}(q)\| = M(q)$.

The accosting force, noticed $\vec{F}_a$, is the resultant of both vectors. It will permits to track the predicted Rendez Vous Point taking into account the collision risk. Its expression is the following :

$$\vec{F}_a(q) = \vec{F}_r(q) + \left(1 - \|\vec{F}_r(q)\|\right) \cdot \frac{\vec{V}_P(q)}{\|\vec{V}_P(q)\|}.$$

Experimentation

We present an experimentation of a left accosting. The follower is place on the right rear of the target. The relative configuration is non ideal because the position are such that a collision can occur during the accosting phase. Figure 13 illustrates the vehicles motion at different times (a/b/c.) until the target stabilization (d).

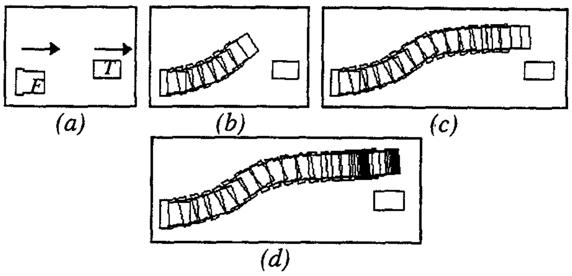

Figure 13 : non ideal left accosting (experimentation)

Following figures show the robots velocities.

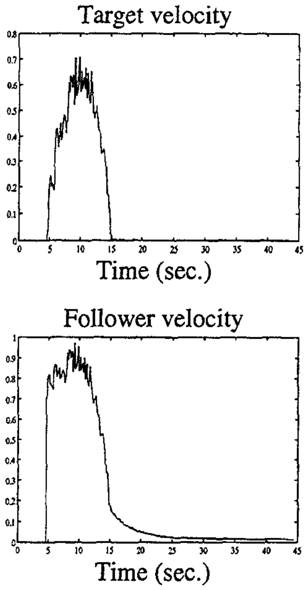

Figure 14 : linear velocity of T and F

Last figure gives the evolution of FMF magnitude. The peak corresponds to the follower entry into the specific influence zone.

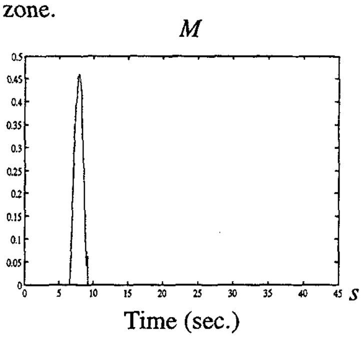

Figure 15 : FMF magnitude along the motion.

1.5 Conclusion

We presented a fuzzy approach based on Takagi-Sugeno controllers to solve the problem of maneuvering target accosting. These solutions are validated by efficient experimentations. We have so define a reactive strategy allowing to perform cooperation between mobile vehicles. Now we are trying to optimize this accosting method, using for this learning through gradient algorithm.

2 Neural adaptive control of a mobile robot

2.1 Interest of neural technique to control mobile robots

We know the generalization and the learning capabilities of Artificial Neural Networks (ANN) due to the efficiency of the learning backpropagation algorithm. This algorithm can train a neural-controller to control a robot. This technique becomes very interesting if the controller has to determine a reactive control. Indeed, when backpropagation is applied on-line (during the control), it permits to adapt and train the controller at any situation.
The goal of the experimentation presented in this section is to show that on-line backpropagation permits a neural network to identify itself a control law. We propose to experiment on-line training in the most unfavorable case. We train an initial random network to control the cartesian configuration (position and orientation) of a mobile robot. The ANN is updating after each control time step.
Our aim is to show that neural technique are able to implement reactive controller in mobile robotics (adaptivity, obstacle avoidance for wheeled robots, control of equilibrium for legged robots..). In this paper, we experiment the control of the robot position without avoiding obstacles. The final objective is to prove that the neural-controller learn to control the robot in a unknown environment. So, in a next experimentation, backpropagation will be used to train the neural-controller both to control the robot and to avoid obstacles.

2.2 Robot position control problem

The problem is to control the cartesian position and the orientation of a two independent driving wheeled industrial mobile robot (figure 16). This problem is not trivial because, the main characteristic of this robot is its non-holonomy (see [14][10]). Kinematic equations are done by equation (1).

$$\left\{ \begin{array}{lcl} \dot{X}_M & = & \frac{r}{2}(\dot{q}_1 + \dot{q}_2)cos\theta \\ \dot{Y}_M & = & \frac{r}{2}(\dot{q}_1 + \dot{q}_2)sin\theta \\ \dot{\theta} & = & \frac{r}{2R}(\dot{q}_2 - \dot{q}_1) \\ r \text{ is} & & \text{radius of wheels} \\ R \text{ is} & & \text{half length of axis wheels} \end{array} \right. \quad (1)$$

The neural controller has to determine instantaneous wheel velocities $\dot{q}_1$ and $\dot{q}_2$ that drive the robot at the desired configuration (X_d, Y_d, θ_d) where (X_d, Y_d) are the absolute desired cartesian coordinates of the middle point M of wheel axis ($X_d =$

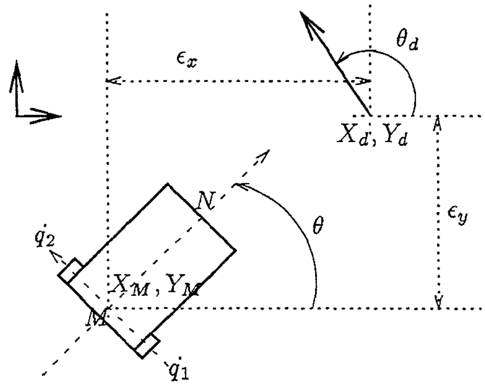

Figure 16: Control of the mobile robot

$Y_d = \theta_d = 0$). So, net inputs are cartesian and orientation errors of the robot, and net outputs are its wheel velocities $\dot{q}_2$ and $\dot{q}_1$. Inputs are normalized between -1 et $+1$ with respect to their maximum value (4 meters for the cartesian errors and π for the orientation).

2.3 Approach for the training

The goal of backpropagation is to minimize, on a large set of examples, a cost function that depending on the net output. This minimization is obtain by the gradient descent technique (see [15] for more precisions). The gradient is calculated with respect to the net outputs. Usually, the cost function is a quadratic error calculated between the net output and a desired one. In the cartesian control of non-holonomic mobile robots, it is often difficult to determine the control law. Then, it is difficult to compute a large set of desired outputs (desired wheeled velocities).
Nevertheless, M.I. Jordan in [12] and [13] propose to include in the quadratic error, some constraint functions. He introduce the difficulties of using a criterion for the learning particularly in the determination of the gradient when these constraints are unknown.
In the case of the control of robots, it is often possible to explain the goal of the control as a function of the control parameters. If we determine the gradient of this objective function, calculated with respect to the control parameters (or a approximation of it) the neural controller can minimize this function on a large set of possible inputs. Then backpropagation permits to identify a control law that satisfy the objective. Learning with a criterion is an interresting method because, like reenforcement techniques, we can train a neural controller on-line, that is during the control.

The experimentation described in this section is based on a learning on-line architecture where the neural controller try to minimize a control criterion when it controls the real robot.
The objective is to determine, at each time step k, the wheel velocities ($\dot{q2}_k$ and $\dot{q1}_k$) that minimize the distance between the robot configuration and the desired one. Then the criterion is (X_k, Y_k are coordinates of the M point at time step k) :

$$J = \alpha_1 X_{k+1}^2 + \alpha_2 Y_{k+1}^2 + \alpha_3(\theta_{k+1} - \theta_s)^2 \qquad (2)$$

where α_1, α_2, α_3 are normalization coefficients and $\theta_s = arctg(2\frac{Y_k)}{D})$ is a strategy constraint that helps the network on the singular configuration $X = 0$
By considering the time step Δt and the kinematic equations (1), the criterion becomes:

$$\begin{aligned} J_{k+1} &= \alpha_1(X_k + \Delta X_{k+1})^2 + \alpha_2(Y_k + \Delta Y_{k+1})^2 \\ &+ \alpha_3(\theta_k + \Delta\theta_{k+1} - \theta_s)^2 \end{aligned}$$

$$\text{where} \begin{cases} \Delta X_{k+1} = \Delta U_{k+1} cos(\theta_k + \frac{\Delta\theta_{k+1}}{2}) \\ \Delta Y_{k+1} = \Delta U_{k+1} sin(\theta_k + \frac{\Delta\theta_{k+1}}{2}) \\ \Delta\theta_{k+1} = \frac{r}{2R}(\dot{q}_{1_k} - \dot{q}_{2_k})\Delta_t \\ \Delta U_{k+1} = \frac{r}{2}(\dot{q}_{1_k} + \dot{q}_{2_k})\Delta_t \end{cases}$$

The gradient that trains the network is calculated with respect to the net outputs :

$$\begin{aligned} \frac{\partial J_{k+1}}{\partial \dot{q}_{i_k}} = 2(\alpha_1 X_{k+1}\frac{\partial \Delta X_{k+1}}{\partial \dot{q}_{i_k}} + \alpha_2 Y_{k+1}\frac{\partial \Delta Y_{k+1}}{\partial \dot{q}_{i_k}} \\ + \alpha_3(\theta_{k+1} - \theta_s)\frac{\partial \Delta\theta_{k+1}}{\partial \dot{q}_{i_k}}) \end{aligned}$$

where:

$$\begin{cases} \frac{\partial \Delta X_{k+1}}{\partial \dot{q}_{i_k}} = \Delta_k \frac{r}{2} cos(\theta_k + \frac{\Delta\theta_{k+1}}{2}) \\ \quad - \frac{1}{2}\frac{\partial \Delta\theta_{k+1}}{\partial \dot{q}_{i_k}} \Delta U_{k+1} sin(\theta_k + \frac{\Delta\theta_{k+1}}{2}) \\ \frac{\partial \Delta Y_{k+1}}{\partial \dot{q}_{i_k}} = \Delta_k \frac{r}{2} sin(\theta_k + \frac{\Delta\theta_{k+1}}{2}) \\ \quad + \frac{1}{2}\frac{\partial \Delta\theta_{k+1}}{\partial \dot{q}_{i_k}} \Delta U_{k+1} cos(\theta_k + \frac{\Delta\theta_{k+1}}{2}) \\ \frac{\partial \Delta\theta_{k+1}}{\partial \dot{q}_{i_k}} = \frac{r}{2R}\Delta t \ (i=1) \ or \ -\frac{r}{2R}\Delta t \ (i=2) \end{cases}$$

2.4 Experimentation of line training

The learning structure that trains the neural controller is represented on Figure 17. This structure has a control loop to control the robot and a learning loop to update the ANN.
This made feasible adaptive control. As an illustration of that affirmation, we proposed to experiment on-line training with an **untrained network** which is updating after each sampling time.

Protocol of experimentation

The ROBUTER II robot is the one used in the dynamic target tracking experimentation based on fuzzy techniques, presented in the first section.

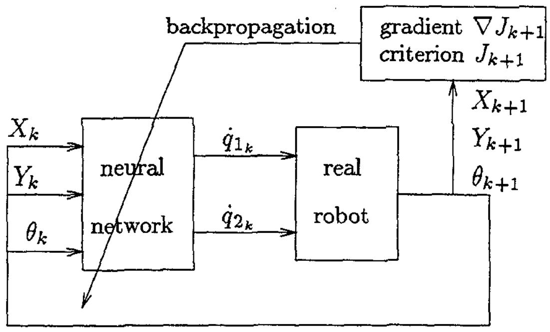

Figure 17: On-line training scheme

We have used a one hidden layer (of 9 neurons) network. The robot velocity is limited at 0.8 m/s during training. Only three learning lessons have been necessary to train the network. At the beginning of each lesson, the robot is at an initial configuration far from the desired one (about 4 meters). The protocol is as follows:

- **Lesson 1:** The network is untrained (all the weights are randomly initialized). Initial robot configuration is $X_M = 3m$, $Y_M = 2m$, $\theta = 0^o$.
- **Lesson 2:** Network has been trained by the first lesson. Initial robot configuration is still $X_M = 3m$, $Y_M = 2m$, $\theta = 0^o$.
- **Lesson 3:** Network has been trained by the firsts lessons. Initial robot configuration is $X_M = 3m$, $Y_M = 0m$, $\theta = 180^o$.

We stop each training when $\dot{q2}_k = \dot{q1}_k = 0$.

First and second training

Figure 18 shows evolution of the training criterion. The learning duration is three times faster for the second training. Figure 19 shows cartesian trajectory, orientation of the robot, error and velocities of the wheels (i.e net outputs converted in m/s).

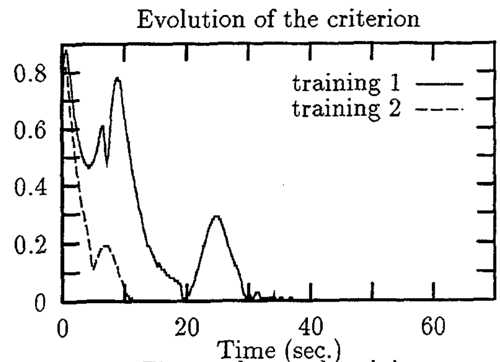

Figure 18: First and second trainings

For the first lesson, the robot trajectory is hazardous during ten minutes but we see clearly that the robot is attracted to the desired configuration.The trajectory is more direct and shorter in lesson 2 than in lesson 1. Final errors are reasonably good (about 1cm and 5 degrees) in regards with the robot size (1.025 m × 0.68 m) and are smaller after lesson 2.

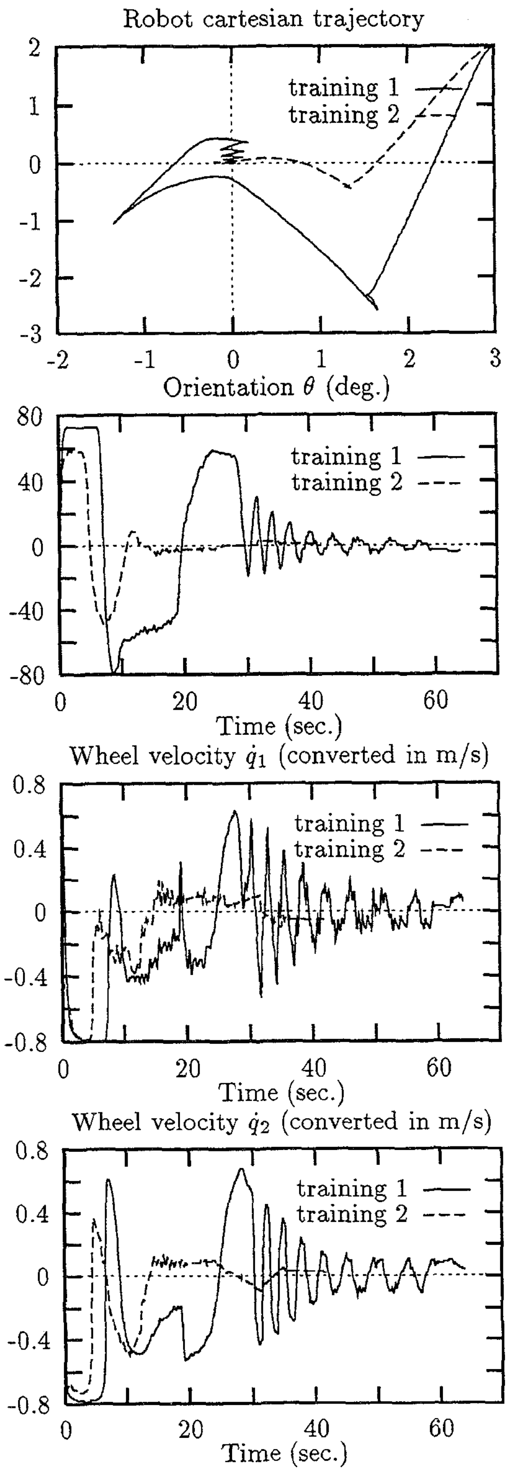

Figure 19: Trajectory, orientation and wheel velocitie of the robot during first and second trainings

Wheel velocities are more smooth in lesson 2 and the average velocity greater. The third lesson used a different initial configuration ($X_M = 3m$, $Y_M = 0m$, $\theta = 180°$). The results are similar to the previous and the robot reaches the desired configuration in 40 seconds. minutes (results are not plotted) . These results shows that indirect backpropagation allows the neural-controller to identify the real behavior of the robot and determine a control law to drive the robot to the desired configuration and increase the quality of the control. Finally, we can argue that there is an knowledge acquisition of the robot kinematics and an adaption of the control.

Control of the robot after on-line training

The trained network controls now the robot without learning from an initial configuration which has never been observed during the learning: $X_M = 4m$, $Y_M = -2m$, $\theta = -90^o$. The maximum velocity of the robot is 1.25 m/s. Figures 20 and 21 show the net ability to control the robot with small final errors.

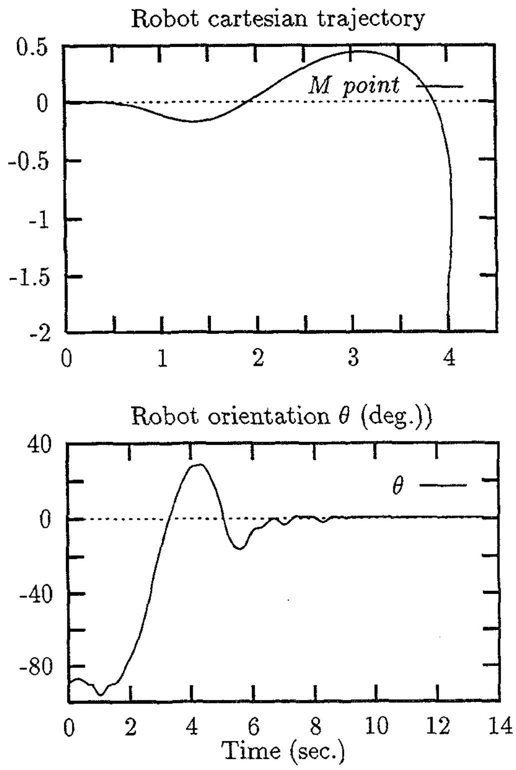

Figure 20: Control of the robot after three trainings

Others tests establish that the neural net controls the robot at any configuration while $X, Y \leq \pm 3\ m$ These results show clearly that indirect backpropagation allows neural networks to learn control laws by especially specifying the control objective.

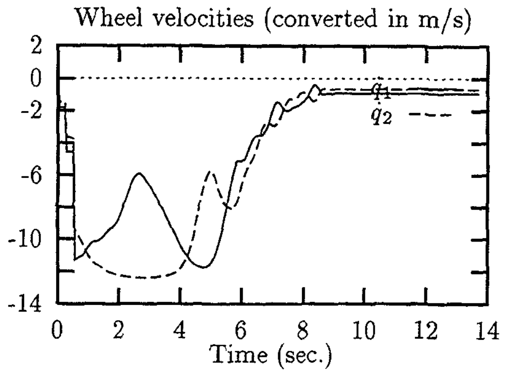

Figure 21: Control of the robot after three trainings

2.5 Conclusion

In this paper, we have presented experimentation results of the neural adaptive control of a mobile robot. Our approach consists to express the control objective as a criterion. The gradient of the criterion is backpropagating through the network instead of the classical quadratic error. The technique permits the neuro controller to learn on-line. Experimental validation is realized by the position and the orientation control of a fast industrial mobile robot. To show the feasibility of the method, we trained a random neural network on the real robot with. Results show clearly the very good and very fast adaption of the neural-network in front of the kinematics constraints and the dynamics effects of the robot.

We are now working on training a neural controller to control the robot and to avoid obstacles. The principle of the approach is to insert in the objective function (2), the distance informations issued from 6 ultra-sonic sensors. The criterion (2) becomes:

$$
\begin{aligned}
J_{k+1} &= \alpha_1 X_{k+1}^2 + \alpha_2 Y_{k+1}^2 + \alpha_3(\theta_{k+1} - \theta_s)^2 \\
&+ \alpha_4 \sum_{j=0}^{6} \beta_j e^{-d_j^2}
\end{aligned}
$$

$$
\text{where} \left\{ \begin{array}{l} d_j\text{: measure issued from the sensor } j \\ \beta_i \text{ is a penality : } \beta_j = 0 \;\; if \;\; d_j > d_s, \\ \beta_j = 1 \;\; if \;\; d_j < d_s, \\ d_s \text{ : security distance} \end{array} \right.
$$

The gradient $\frac{\partial d_j}{\partial \dot{q}_i}$ is determine by the distances d_i depending on $\dot{q}_1$ and $\dot{q}_2$.

3 General Conclusion

We have presented in this paper solutions using Artificial Intelligent Techniques (Fuzzy Logic, Neural Networks) to solve complex mobile robotics problems. These approaches are validated on real robots. The corresponding experimentations are taking place on a large study on the using of non academic methods for reactive control of mobile vehicles. Our final objective is to confer to an autonomous plateform a reactive behavior and an adaption capability to perform a wide class of tasks.

References

[1] G. Bourdon, *Stratégie Réactive d'Accostage entre Robots Mobiles Autonomes en Milieu Contraint. Approche par Techniques Floues*, PHD thesis of Université Pierre et Marie Curie, Paris, dec. 1996

[2] J. Amat, J.Aranda, A. Csals, *A Tracking System for Dynamic Control of Convoys*, Robotics and Autonmos Systems,11, pp 269-277, 1993

[3] S. H. Ji & Al., *Tracking Navigation Using Fuzzy Inference and Sonar-Base Obstacle Avoidane*, Int. Conf. on Intelligent Robots and Systems, 1993

[4] P. Daviet and M. Parent, *Platooning for Small Public Urban Vehicles*, Fourth Int. Symp. on Experimental Robotics, California, pp. 213-218, 1995

[5] G. Bourdon, P. Ruaux, S. Delaplace, *C.A.N. for Autonomous Mobile Robot*, Int. CAN Conference, Paris, pp 9-17, 1996

[6] C. D. Yang, F. B. Yeh, *Generalized Guidance Law for Homing Missiles*, IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-25, N°2, pp. 197-212, march 1989

[7] O. Khatib, *Real-time Obstacle Avoidance for Manipulators and Mobile Robots*, Int. Journal of Robotics Research, Vol. 5, N°1, pp. 90-98, 1986

[8] B. d'Andrea Novel *" Sur la Commande des systèmes mécaniques sous actionnés et des systèmes non holonomes"*, rapport d'Habilitation à Diriger les Recherches, Ecole des Mines de Paris,1 mars 1995,

[9] D. Nguyen, B. Widrow *"Neural Networks for Self-Learning Control Systems"*,IEEE Work. on Industrial Applications of Neural Networks, pp 18-23,1991

[10] S. Delaplace, *"Navigation à Coût Minimal d'un Robot Mobile dans un Environnement Partiellement Connu"'* thèse de l'Université Paris 6 (Pierre et Marie Curie), France, November 1991

[11] Andrew G. Barto *"Connectionnist Learning for Control"*, Neural Network for Control, pp 5-58, 1990

[12] M.I. Jordan and D.E Rummelhart, *"Forwars Models: Supervised Learning with a Distal Teacher"*, Cognitive Science, Vol 16, pp. 307-354, 1992

[13] M.I. Jordan, *"Constrained Supervised Learning"*, Journal of Mathematical Psychologie , Vol 36, pp. 396-425, 1992

[14] C. Samson , C.K. Ait-Abderrahim, *"Feedback Control of a Non Holonomic Wheeled cart in cartesian space"*, Proc. of IEEE Int. Conf. of Robotics and Automation,pp 1136-1141 April 91, U.S.A.

[15] Y. Le Cun, *Modèles connexionnistes de l'apprentissage*, thèse de l'Université Pierre et Marie Curie, juin 1987.

[16] P. Hénaff, *"Mises en Œuvre de Commandes Neuronales par Rétropropagation Indirecte: Application à la Robotique Mobile"*, PHD thesis of Université (Pierre et Marie Curie), Paris, june 1994