Adaptive neural control in mobile robotics: experimentation for a wheeled cart

Patrick Henaff, M. Milgram, J. Rabit

Adaptive Neural Control In Mobile Robotics: Experimentation for a Wheeled Cart

P. HENAFF, M. MILGRAM, J. RABIT

Laboratoire de Robotique de Paris-URA 1778
10-12, Avenue de l'Europe
78140 Vélizy. FRANCE
e-mail: henaff@robot.uvsq.fr

Abstract **This paper presents experimental results of an original approach to the Neural Network learning architecture for the control and the adaptive control of mobile robots. The basic idea is to use non-recurrent multi-layer-network and the backpropagation algorithm without desired outputs, but with a quadratic criterion which spezify the control objective. To illustrate this method, we consider an experimental problem that is to control cartesian position and orientation of an non-holonomic wheeled cart.**
The results establish that the neural net learns on-line the kinematic constraints of the robot. After several on-line learning lessons the net is able to control the robot at any configurations in a limited cartesian space.

I. Introduction

It is interesting to use multilayered networks in control of robotic systems because of their following properties:

- their strong generalization capabilities of multilayered networks (a neurocontroller can be designed even no explicit form of a control law is known),
- their reduced computation time,
- their inherent robustness to real noisy data.

A standard neural control scheme is build on the academic feedback control approach. The neural net replace the classical corrector (see Fig.1). The net input is a combination between the current state of the system and a desired state. The net outputs are the control parameters. Multilayered networks are usually trained with the well known backpropagation learning algorithm (direct backpropagation). The goal of backpropagation is to minimize the quadratic error E (addition of quadratic errors E_p for every input pattern p) between the net and a desired ouput (see Fig. 2). Consequently, it requires an a priori knowledge of the desired outputs (e.g. the desired controls) corresponding to the net input.

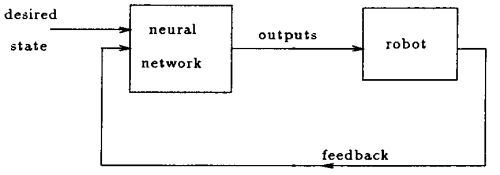

Fig. 1. Neural feedback control

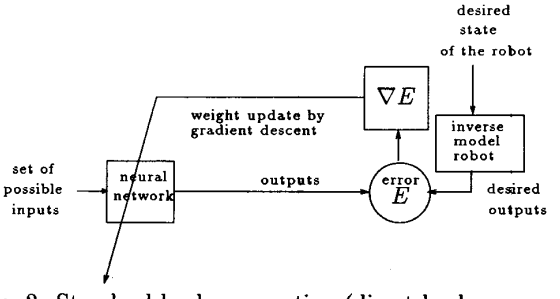

Fig. 2. Standard backpropagation (direct backpropagation)

Determination of desired controls is the main problem of using standard backpropagation in learning control of robotic applications. Two solutions are generally used to calculate it:

- the first requires a reference model: a PID or fuzzy controller [4] or a human teacher [5]. This solution does not exploit the learning and identification capabilities of neural nets (except for the human teacher).
- The second inverse a model (geometric, kinematic or dynamic) of the robot. But in control of mechanical complex systems (arm or legged robots), the

0-7803-2129-4/94 $3.00

inverse model can not, in general, be obtained analytically and it is not often uniquely determinable (kinematics redonduncy). So, the main academic methods used criterion optimization technics to find a solution of this inversion. This increase the on-line learning complexity.
In the case of non holonomic mobile robots (independant driving wheeled robots) the inverse model can not easily be obtained because of the non-integrability of equations.

Then, neural learning with standard backpropagation has limitations in control of robotic systems. To make allowance for these remarks and since backpropagation is an optimization algorithm, we propose a learning method that doesn't need desired output, but only a criterion specifying directly the robotics task.
This method, named undirect backpropagation, has been used for a neural dynamic reflex control problem in which the network learns to control the dynamic equilibrium of a simulated planar biped (see [6] for more details).

In this paper, we presents experimental results on the on-line undirect backpropagation applied to the control of the cartesian position and orientation of a two independant driving wheeled mobile robot. The next section describes first the undirect backpropagation.

II . Learning with a criterion

The basic idea is to mimimize the arbitrary criterion J_p (for every input pattern p) as a function of the net ouput, instead of the error E_p between the net and the desired output. The advantage of learning with criterion is that we need not any desired output, but only a cost function which spezify the control objective and its constraints. The neural network determines itself the way to achieve this objective.
In order to minimize this criterion, we use also a gradient descent method :

$$\frac{\partial J_p}{\partial w_{ji}} = \frac{\partial J_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} = \frac{\partial J_p}{\partial net_{pj}} o_{pi}$$

where

- w_{ji} is the weight from neuron i to neuron j,
- o_p is the net ouput vector,
- $net_{pj} = \sum_i w_{ji} o_{pi}$ denotes the activation of neuron j and f_j its function (sigmoïde).

We set $\delta_{pj} = -\frac{\partial J_p}{\partial net_{pj}}$ and obtain for a hidden neuron :

$$\delta_{pj}^{hid} = f_j'(net_{pj}) \sum_k \delta_{pk} w_{kj}$$

and for an output one :

$$\delta_{pj}^{out} = -\frac{\partial J_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}}$$

$$= -\frac{\partial J_p}{\partial o_{pj}} f_j'(net_{pj}) = \nabla J \, f_j'(net_{pj})$$

Instead of the error E_p, we backpropagate the gradient of the criterion $\nabla J = -\frac{\partial J_p}{\partial o_{pj}}$ related to the corresponding output. Therefore, the partial derivates of the criterion must be analytically calculable with respect to the control parameters. The computation of δ_{pj}^{hid} remains unchanged.

Figure 3 illustrates the new learning algorithm (compares with figure 2), which is using to train off-line the network with a model of the robot.

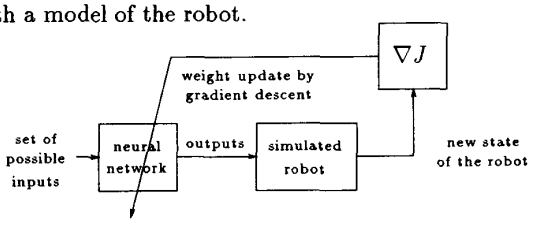

Fig. 3. Learning off-line with criterion (Off-line undirect backpropagation)

Figure 4 plottes the on-line learning control structure where the learning loop refines the weights to the real robot at periodic times.

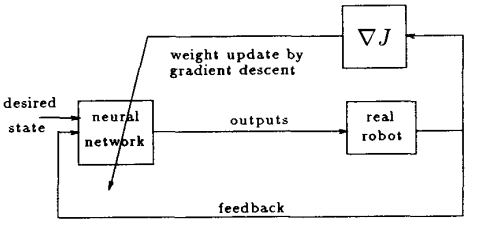

Fig. 4. Learning on-line with criterion (On-line undirect backpropagation)

The backpropagation of the criterion gradient permits the network to learn with the real system. This is the difference between our method and the one proposed by D. H. Nguyen and B. Widrow in [7]. They use a network to emulate the system. Then, they can backpropagate a criterion through the emulator net and trained the controller without a gradient. Nethertheless, the real robot cannot be used in their method because the criterion cannot be propagated throught it. Consequently, they cannot refine their neural controller to the real robot.

Experimental results presented in the next section establish that on-line learning scheme adjust the neural controller in order to solve the control objective. So it is an adaptive neural control scheme.

III . On-line learning control of an experimental mobile robot

A . *Control problem*

The problem is to control cartesian position and orientation (see fig 5) of a two independant driving wheeled autonomous mobile robot developed at the L.R.P [1].

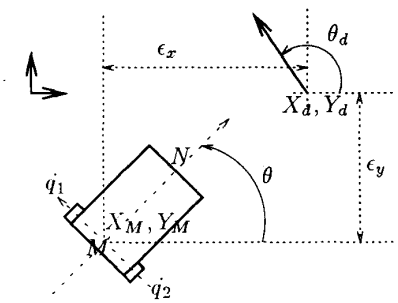

Fig. 5. Control problem

The kinematic equations are:

$$\begin{cases} \dot{X}_M &= \frac{r}{2}(\dot{q}_1 + \dot{q}_2)cos\theta \\ \dot{Y}_M &= \frac{r}{2}(\dot{q}_1 + \dot{q}_2)sin\theta \\ \dot{\theta} &= \frac{r}{2R}(\dot{q}_1 - \dot{q}_2) \end{cases} \quad (1)$$

where:

- r is the radius wheels.
- R is the half length of the axis wheels.

The main caracteristic of this robot is its non-holonomy (there is not stabilisant continous state feedback [2]).
The neural controler has to determine the 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.
Then, net outputs are the two wheel velocities and net inputs are cartesian and orientation errors.
The desired configuration is set as $X_d = Y_d = \theta_d = 0$, the control structure is described on figure 6.

B . *Control criterion and learning gradient*

The objective is to minimize the distance between the robot configuration and the desired one.

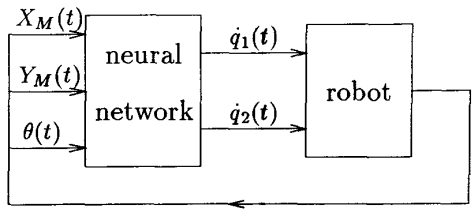

Fig. 6. Neural control structure

Then the criterion is :

$$J = \alpha_1 X_M^2(t+1) + \alpha_2 Y_M^2(t+1) + \alpha_3(\theta(t+1) - \theta_s)^2 \quad (2)$$

where

- α_1, α_2, α_3 are normalization coeficients
- $\theta_s = arctg(2Y_M(t))$ is a strategy constraint that helps the network on the singular configuration $X_M = 0$ and $Y_M \neq 0$.

By considering the time step Δt and the kinematic equations (1), the criterion becomes:

$$\begin{aligned} J &= \alpha_1(X(t) + \Delta X(t+1))^2 \\ &+ \alpha_2(Y(t) + \Delta Y(t+1))^2 \\ &+ \alpha_3(\theta(t) + \Delta\theta(t+1) - \theta_s)^2 \end{aligned}$$

where

$$\begin{cases} \Delta X(t+1) &= \Delta U(t+1)cos(\theta(t) + \frac{\Delta\theta(t+1)}{2}) \\ \Delta Y(t+1) &= \Delta U(t+1)sin(\theta(t) + \frac{\Delta\theta(t+1)}{2}) \\ \Delta\theta(t+1) &= \frac{r}{2R}(\dot{q}_1(t) - \dot{q}_2(t))\Delta t \\ \Delta U(t+1) &= \frac{r}{2}(\dot{q}_1(t) + \dot{q}_2(t))\Delta t \end{cases}$$

The gradient that trains the network is calculated with respect to the net outputs (i.e wheel velocitities) :

$$\frac{\partial J}{\partial \dot{q}_i(t)} = 2\alpha_1 X(t+1)\frac{\partial \Delta X(t+1)}{\partial \dot{q}_i(t)} \quad (3)$$

$$+ \ 2\alpha_2 Y(t+1)\frac{\partial \Delta Y(t+1)}{\partial \dot{q}_i(t)} \quad (4)$$

$$+ \ 2\alpha_3(\theta(t+1) - \theta_s)\frac{\partial \Delta\theta(t+1)}{\partial \dot{q}_i(t)} \quad (5)$$

where:

$$\begin{cases} \frac{\partial \Delta X(t+1)}{\partial \dot{q}_i(t)} &= \Delta t\frac{r}{2}cos(\theta(t) + \frac{\Delta\theta(t+1)}{2}) \\ &- \frac{1}{2}\frac{\partial \Delta\theta(t+1)}{\partial \dot{q}_i(t)}\Delta U(t+1) \\ &sin(\theta(t) + \frac{\Delta\theta(t+1)}{2}) \\ \frac{\partial \Delta Y(t+1)}{\partial \dot{q}_i(t)} &= \Delta t\frac{r}{2}sin(\theta(t) + \frac{\Delta\theta(t+1)}{2}) \\ &+ \frac{1}{2}\frac{\partial \Delta\theta(t+1)}{\partial \dot{q}_i(t)}\Delta U(t+1) \\ &cos(\theta(t) + \frac{\Delta\theta(t+1)}{2}) \\ \frac{\partial \Delta\theta(t+1)}{\partial \dot{q}_i(t)} &= \frac{r}{2R}\Delta t \ (i=1) \ or \ -\frac{r}{2R}\Delta t \ (i=2) \end{cases}$$

Note that gradient includes kinematic constraints of the robot.

C . Learning control structure

We see in section II . that our learning method trains the network off-line and on-line. The off-line learning method as been tested with good results. Nevertheless, we solely presents in this paper the results of the on-line learning phase because they are more interresting. Indeed, we argued that on-line learning scheme made feasible adaptive control. As an illustration of that, we proposed to use in the on-line structure (fig. 7) an untrained neural network which is updating after each time control step.

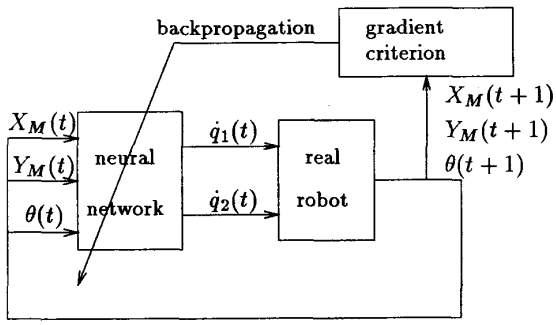

Fig. 7. On-line learning scheme to solve our problem

D . Experimentals results

We used a two hidden layers network (one layer has 6 neurons, and the other 4 neurons). The robot velocity is limited at 10 cm/s. At the begining of each learning lesson, the robot is at an initial configuration far from the desired one (between 3 and 4 meters). We stop each training when the robot completly stops.
We choose 2 initial robot configurations :

- the first : $X_M = 3m$, $Y_M = 2m$, $\theta = 0°$, is use for two successives lessons.
- the second : $X_M = 3m$, $Y_M = 0m$, $\theta = 180°$, is use for one lesson.

First training

Weights of neural net are randomly initialized. Figure 8 plottes training parameters and the M point cartesian trajectory. The learning duration is 10 minutes. The robot trajectory is hasardous during the first 3 minutes but we see clearly that the robot is attracted to the desired configuration. Figures 9 and 10 plottes errors configuration and wheel velocities (velocity of ± 0.15 rad/sec corresponds to a net output value of ± 1). Final errors are reasonably good in regards with the robot size (1.6 m $\times$ 0.7 m).

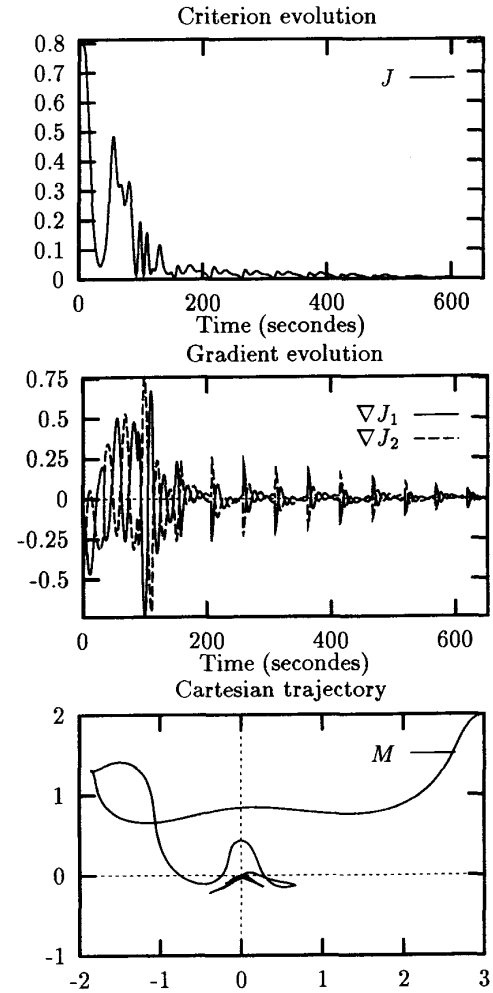

Fig. 8. Criterion during the first training lesson

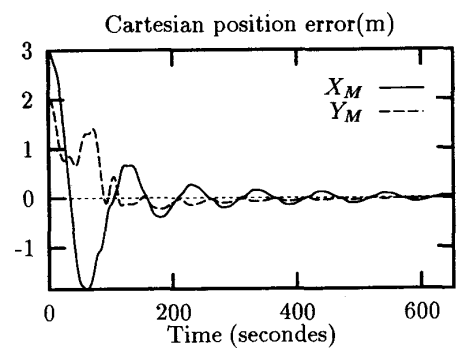

Fig. 9. Cartesian errors during the first training lesson

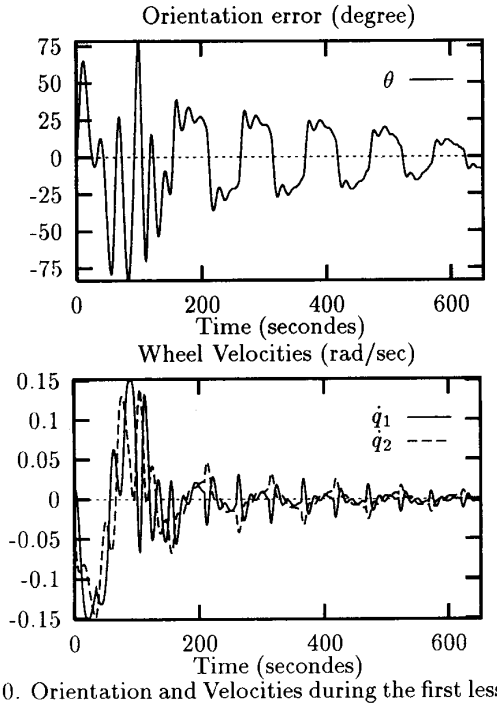

Fig. 10. Orientation and Velocities during the first lesson

Second training

There, the network has been trained in the first lesson and it is the same initial robot configuration. Figures 11, 12 and 13 show that the learning is five times faster than the first one. The trajectory is shorter, the average velocity greater and final errors smaller. These results shows that there is an adaption of the network to the robot kinematic constraints. Finally, we can argue that there is an knowledge acquisition of the robot kinematics. The following 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 2 minutes.

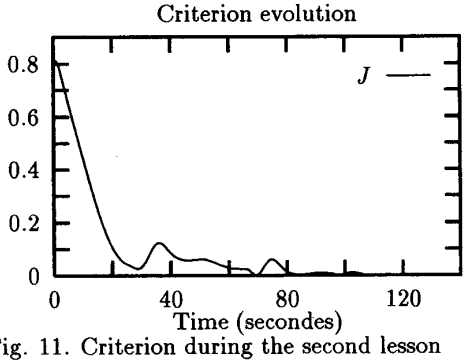

Fig. 11. Criterion during the second lesson

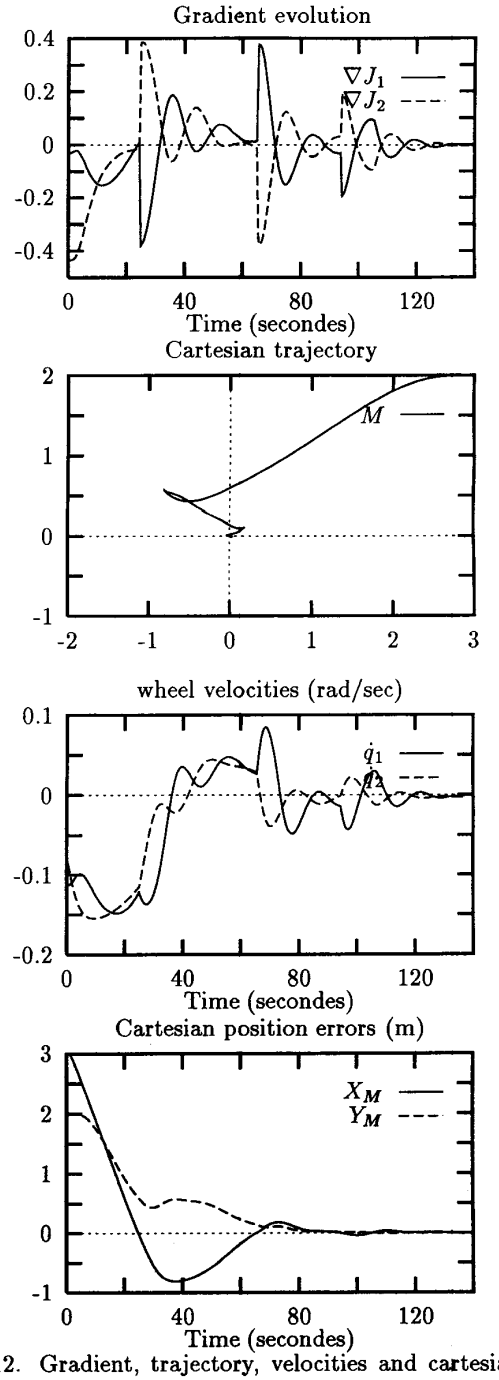

Fig. 12. Gradient, trajectory, velocities and cartesian errors during the second lesson

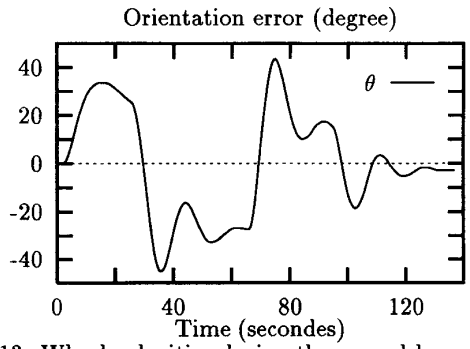

Fig. 13. Wheel velocities during the second lesson

Control after learning

The trained network controls now the robot from an initial configuration that has never used in the learning: $X_M = 2m$, $Y_M = -1m$, $\theta = -90^o$. Figures 14 and 15 show the net ability to control the robot with small final errors. Others tests establish that the neural controls the robot at any configuration while $-3\ m \leq X_M \leq 3\ m$ and $-2\ m \leq X_M \leq 2$.

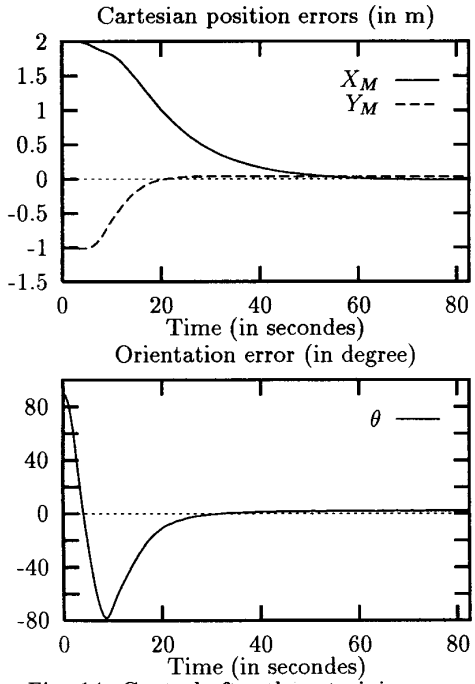

Fig. 14. Control after three trainings

IV . Conclusion

Our goal was to use backpropagation algorithm in order to learn control objective without desired output. The principle of our approach is to express the control objective as a criterion. The gradient of the criterion is back-

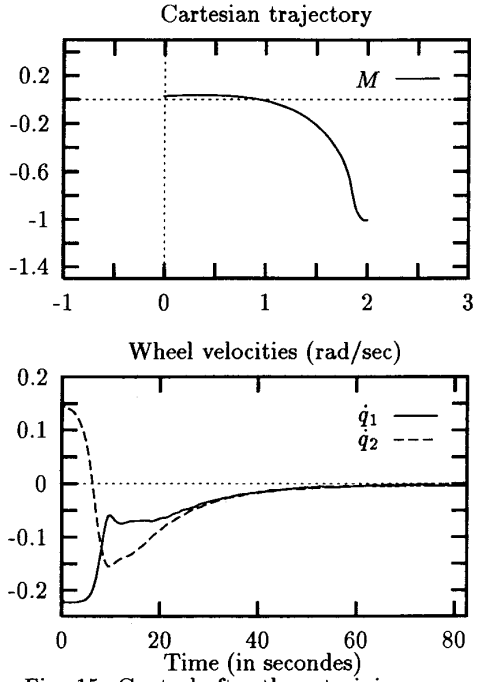

Fig. 15. Control after three trainings

propagated through the network instead of the classical quadratic error. Experimental validation is realised by the position and the orientation control of a two wheeled non-holonomic mobile robot. We show the feasability of the method, particularly the network adaptation in front of the robot kinematics constraints.

REFERENCES

[1] 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

[2] C. K. Ait-Abderrahim, "Commande de Robots Mobiles", *Thèse de l'Ecole des Mines de Paris*, January 1993.

[3] D.E Rummelhart, J.L. McClelland, "Parallel Distributed Processing", *The MIT Press*, Vol 1, pp. 318-362, 1986

[4] D.M.A. Lee, W.H. ElMaraghy, "A Neural Network Solution for Biped Gait Synthesis", *Int. Joint Conference on Neural Networks*, Boston, Vol II, pp. 763-767,1992

[5] A. Guez, J. Selinsky, *A neuromorphic controller with a human teacher*, Proceedings of IEEE International Conference on Neural Networks, pp II.595-II.602, 1988

[6] P. Henaff, H. Schwenk,M. Milgram " A Neural Network Approach of the Control of Dynamic Biped Equilibrium", *International Symposium on Measurement and Control in Robotics*,1993

[7] D. H. Nguyen, B. Widrow "Neural Networks for Self-Learning Control Systems", IEEE Workshop on Industrial Applications of Neural Networks, pp 18-23, Ascona Ticino,1991

[8] P. Henaff, "Mises en Œuvre de Commandes Neuronales par Rétropropagation Indirecte. Application à la Robotique Mobile", *thèse de l'Université Paris 6 (Pierre et Marie Curie)*, France, june 1994