Stable Neural Network Adaptive Control of Redundant Robot Manipulators

A. Benallegue, B. Daâchi and N.K. M'Sirdi
Laboratoire de Robotique de Paris
10-12, avenue de l'Europe, 78140 Velizy, France.
e-mail: benalleg@robot.uvsq.fr

Abstract

We propose in this paper a neural network adaptive controller for redundant robot manipulators. The controller has been designed in cartesian space to avoid motion planning problem which is closely related to the inverse kinematics problem. The neural networks approximate separately the elements of the dynamical model of the robot manipulator written in catesian space. Adaptation laws are derived for each network from stability study of the closed loop system using Lyapunov approach with intrinsic properties of robot manipulators. Two strategies of control have been considered. In the first one the aim of the controller is to acheive good tracking of the end-effector regardless the robot configurations. In the second way, the controller has been improved using augmented space strategy to ensure minimum joint positions of the robot. Simulations results demonstrate that the proposed controller is effective.

1 Introduction

The assigned tasks to robot manipulators are always specified in Cartesian space coordinates called operational space, however, control signals are delivered at the robot joints. It is then natural to design the controller in joint space, but in this case the control synthesis problem is usually transformed into a problem of motion planning in joint space from trajectories described in Cartesian space. The transition form one space to another is achieved through geometric and kinematics transformations. These transformations are based on direct and inverse models describing geometry and kinematics of the robot. For parallel robots, the complexity is to compute direct models. By opposition, the problem in serial robot is to find inverse models particularly for redundant robots.

The use of kinematically redundant robots is particularly interesting because of their flexibility to circumvent the internal singular configurations and their ability to avoid obstacles. The controller is usually designed in joint space and the problem to be solved is to determine a solution to the velocity inverse kinematics. Much effort has been devoted in this area. Some solutions proposed in the literature are based on non-linear optimization methods which are complex to implement, may cause instability and have long time of convergence [2][3] [4][6][9]. Recently, new interest in the neural network research has been generated to reduce the computational complexity of motion planning and control for manipulators. Several neural networks methods have mainly been studied by researchers to model the forward and inverse kinematics mapping for manipulators [1] [7]. However, although of their faculty of inversion in the previous situations, these methods don't permit to give in real time the solution that it is necessary to execute a given task. In addition, they can cause instability of the system if they are used in the control loop.

In this paper our attention is focused on designing a stable adaptive neural controller with no inversion problem and without knowledge on the dynamic model of the robot. The proposed controller would be valid for all manipulator situations. Two strategies have been considered. Firstly the aim of the controller is to achieve good trajectory tracking of the end-effector regardless robot configurations. The drawback in this case is that the robot can be sometimes in a random configurations. To overcome this problem, we used optimization techniques to generate a new variable which ensures minimum joint positions. The control of this variable around zero permits the minimization of joint positions under the geometric constraint. Therefore the trajectory tracking in the

Cartesian space is assured with minimal and smooth movements of the robot.

The article is organized as follows: Section 2 describes the robot arm dynamics and its structural properties. In section 3 the neural network representation of the approximated model is given. Section 4 gives the first version of the proposed controller with stability analysis of the system in closed loop and simulation results on 3 DOF-planar-robot. Section 5 concern a modified controller which takes into account minimal joint positions and simulation results on the same robot. Section 6 is a conclusion.

2 Dynamic model and properties

We consider a n-degree-of-freedom robot manipulator with revolute joints. The dynamic model is expressed in the form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + H(q,\dot{q}) = \tau \qquad (1)$$

with q, $\dot{q}$ and $\ddot{q}$ (vectors of n dimension) are respectively, joint positions, joint velocities and joint accelerations, $M(q)$ the inertia matrix, $C(q,\dot{q})\,\dot{q}$ the Coriolis centripetal forces, $H(q,\dot{q})$ the vector of gravitational forces, frictions and the rest of dynamics, and τ the vector of the torques acting on the joints.

The end-effector velocity vector $\dot{x} \in R^m$ and the joint velocity vector $\dot{q} \in R^n$ are related through the Jacobian matrix $J(q) \in R^{m\times n}$ of the direct kinematic function as:

$$\dot{x} = J(q)\dot{q} \qquad \left(J(q) = \frac{\partial \phi(q)}{\partial q}\right) \qquad (2)$$

where $x = \phi(q)$ is the end-effector Cartesian position.

The end-effector acceleration vector $\ddot{x}$ is related to the joint acceleration vector $\ddot{q}$ as follows:

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \qquad (3)$$

In case of non-redundant robots $m = n$, and for no singular situations of the manipulator, the matrix J is always invertible. We can therefore use equations (2) and (3), to compute the joint velocities and accelerations respectively as:

$$\dot{q} = J^{-1}\dot{x} \quad \text{and} \quad \ddot{q} = J^{-1}\ddot{x} - J^{-1}\dot{J}J^{-1}\dot{x}$$

While replacing $\dot{q}$ and $\ddot{q}$ in the model (1), the model expressed in Cartesian space can be written as follows:

$$M^*(q)\ddot{x} + C^*(q,\dot{q})\dot{x} + H^*(q,\dot{q}) = \tau^* \qquad (4)$$

with:

$$\begin{cases} M^* = J^{-T}MJ^{-1} \\ C^* = J^{-T}(CJ^{-1} - MJ^{-1}\dot{J}J^{-1}) \\ H^* = J^{-T}H \\ \tau^* = J^{-T}\tau \Rightarrow \tau = J^T\tau^* \end{cases}$$

The robot dynamics (4) have physical properties that can be used for the control law synthesis:

Property 1: The matrix M^* is Symmetric Positive Definite (SPD).

Property 2: The matrix C^* can be chosen so that $\dot{M}^* - 2C^*$ is skew symmetric.

In case of redundant robots $m < n$, the inverse kinematics poses a challenging problem as for any given end-effector velocity, there exists an infinite number of solutions. Hence, the problem is to compute the matrices M^*, C^* and the vector H^* that include matrix inversion. To overcome this problem and in the same time to avoid any knowledge on the dynamic model of the system, we propose to design an adaptive control law in Cartesian space. Thus, the problem of inversion is transformed to a problem of estimation of matrices $M^*(q)$, $C^*(q,\dot{q})$ and the vector $H^*(q,\dot{q})$. The neural networks are the interesting tools to perform the optimal estimations of the unknown cited matrices and vector.

3 Neural network approximations

The considered function approximator is an Multi-Layer Perceptron (MLP) neural network with one hidden layer and the output is linear. The approximations have the structure $\theta_1^T\varphi(\theta_2^T v)$ where θ_2 is the input-hidden layer weights of the neural network, θ_1 is the hidden-output weights of the neural network, φ provides an activation function of the hidden neurons and v is the input vector signal of the neural network.

The functions to be approximated are all the elements of the matrix $M^*(q)$, the elements of the matrix $C^*(q,\dot{q})$ and the elements of the vector $H^*(q,\dot{q})$ and are denoted by

$$\begin{aligned} m_{ij}^*(z) &= \alpha_{1ij}^T\varphi(\alpha_{2i}^T z) + \epsilon_{m_{ij}}(z) \\ c_{ij}^*(\acute{z}) &= \beta_{1ij}^T\varphi(\beta_{2i}^T \acute{z}) + \epsilon_{c_{ij}}(\acute{z}) \\ h_i^*(\acute{z}) &= \gamma_{1i}^T\varphi(\gamma_{2i}^T \acute{z}) + \epsilon_{h_i}(\acute{z}) \\ z &= q; \quad \acute{z} = (q,\dot{q}) \end{aligned}$$

where $\left\|\epsilon_{(.)}(\acute{z})\right\| < \bar{\epsilon}(\acute{z})$, with a known and sufficiently small $\bar{\epsilon}(\acute{z}) \in C^1$, $\alpha_{2i}^T \in R^{PM\times n}$, $\alpha_{1ij}^T \in R^{PM}$, $\beta_{2i}^T \in$

$R^{p_C\times(2n)}$, $\beta_{1ij}^T \in R^{p_C}$, $\gamma_{2i}^T \in R^{p_H\times(2n)}$ and $\gamma_{1i}^T \in R^{p_H}$. The activation function is of sigmoidal form.

In matrix form, these approximations can be written:

$$\begin{aligned} M^*(z) &= \alpha_1^T\varphi(\alpha_2^T z) + \epsilon_M(z) \\ C^*(\dot{z}) &= \beta_1^T\varphi(\beta_2^T \dot{z}) + \epsilon_C(\dot{z}) \\ H^*(\dot{z}) &= \gamma_1^T\varphi(\gamma_2^T \dot{z}) + \epsilon_H(\dot{z}) \end{aligned}$$

where:

$$\begin{aligned} \alpha_1^T &= [\alpha_{11}^T, \cdots, \alpha_{1m}^T] \\ \beta_1^T &= [\beta_{11}^T, \cdots, \beta_{1m}^T] \end{aligned}$$

and:

$$\varphi(\alpha_2^T z) = \begin{bmatrix} \varphi(\alpha_{21}^T z) & \underline{0} & \cdots & \underline{0} \\ \underline{0} & \varphi(\alpha_{22}^T z) & & \vdots \\ \vdots & & \ddots & \underline{0} \\ \underline{0} & \cdots & \underline{0} & \varphi(\alpha_{2m}^T z) \end{bmatrix};$$

$$\varphi(\beta_2^T \dot{z}) = \begin{bmatrix} \varphi(\beta_{21}^T \dot{z}) & \underline{0} & \cdots & \underline{0} \\ \underline{0} & \varphi(\beta_{22}^T \dot{z}) & & \vdots \\ \vdots & & \ddots & \underline{0} \\ \underline{0} & \cdots & \underline{0} & \varphi(\beta_{2m}^T \dot{z}) \end{bmatrix}$$

with $\alpha_{1i}^T \in \mathsf{R}^{n\times p_M}$, $\beta_{1i}^T \in \mathsf{R}^{n\times p_C}$, $\gamma_1^T \in \mathsf{R}^{n\times p_H}$.

The estimations of matrix $M^*(z)$, matrix $C^*(\dot{z})$ and vector $H^*(\dot{z})$ are given by:

$$\begin{aligned} \hat{M}^*(z) &= \hat{\alpha}_1^T\varphi(\hat{\alpha}_2^T z) \\ \hat{C}^*(\dot{z}) &= \hat{\beta}_1^T\varphi(\hat{\beta}_2^T \dot{z}) \\ \hat{H}^*(\dot{z}) &= \hat{\gamma}_1^T\varphi(\hat{\gamma}_2^T \dot{z}) \end{aligned} \quad (5)$$

with $\hat{\alpha}_2$, $\hat{\alpha}_1$, $\hat{\beta}_2$, $\hat{\beta}_1$, $\hat{\gamma}_2$ and $\hat{\gamma}_1$ are neural networks parameters which will be provided by an adaptation algorithm based on stability analysis.

With regard to linear parametrized networks [5][8], the advantage of the MLP networks is certainly the relatively reduced number of parameters. It is clear that this number depends on the dimension of the input, nevertheless this dependence is not exponential. The drawback of this type of networks is their non-linear parametrization. However, an alternative to treat these non-linearities is to use development in Taylor series of functions $\varphi(\theta^T z)$ around the estimated parameter $(\hat{\theta}^T z)$. It can be written as follows:

$$\varphi(\theta^T z) = \varphi(\hat{\theta}^T z) - \varphi'(\hat{\theta}^T z)\tilde{\theta}^T z - O(\hat{\theta}^T z) \quad (6)$$

with $\varphi'(\hat{z}) = d\varphi(z)/dz|_{z=\hat{z}}$, and $O(z)$ represents terms of superior order, their values are

$$O(\hat{\theta}^T z) = (\varphi(\hat{\theta}^T z) - \varphi(\theta^T z)) - \varphi'(\hat{\theta}^T z)\tilde{\theta}^T z \quad (7)$$

It is well known that sigmoidal functions φ and their derivatives φ' are bounded, then for (7) we can determine the approximation error bounds with Taylor series, that are such as:

$$||O(z)|| \leq c_1 + c_2||\tilde{\theta}||_F||z|| \quad (8)$$

where c_i $(i = 1, 2)$ are positive constants calculated from the expressions of φ and φ'.

4 Neural adaptive controller

The main objective of the controller is to achieve trajectory tracking of the robot en-effector. The desired trajectory is then defined in operational space by the variable x_d of dimension m. The tracking errors and reference signals are defined by:

$$\begin{aligned} &e = x - x_d; \quad \dot{e} = \dot{x} - \dot{x}_d; \quad s = \dot{e} + \Lambda e; \\ &\dot{x}_r = \dot{x}_d - \Lambda e; \quad \ddot{x}_r = \ddot{x}_d - \Lambda\dot{e} \end{aligned} \quad (9)$$

Λ is a diagonal positive matrix.

The proposed control law is given by:

$$u = \hat{M}^*(z)\ddot{x}_r + C^*(\dot{z})\dot{x}_r + \hat{H}(\dot{z}) - K_v s + w \quad (10)$$

$K_v > 0$ is a gain matrix. The signal w is used to compensate the approximation errors, it will be defined later.

The neural network parameter adaptation laws are defined as follows:

$$\begin{aligned} \dot{\hat{\alpha}}_1 &= -\kappa\Gamma_M||s||\hat{\alpha}_1 - \Gamma_M(\varphi(\hat{\alpha}_2^T z) + \varphi'(\hat{\alpha}_2^T z)\hat{\alpha}_2^T z)\ddot{x}_r s^T \\ \dot{\hat{\alpha}}_2 &= -\kappa\Gamma_M||s||\hat{\alpha}_2 - \Gamma_M z\ddot{x}_r s^T\hat{\alpha}_1^T\varphi'(\hat{\alpha}_2^T z) \\ \dot{\hat{\beta}}_1 &= -\kappa\Gamma_C||s||\hat{\beta}_1 - \Gamma_C(\varphi(\hat{\beta}_2^T \dot{z}) + \varphi'(\hat{\beta}_2^T \dot{z})\hat{\beta}_2^T \dot{z})\dot{x}_r s^T \\ \dot{\hat{\beta}}_2 &= -\kappa\Gamma_C||s||\hat{\beta}_2 - \Gamma_C\dot{z}\dot{x}_r s^T\hat{\beta}_1^T\varphi'(\hat{\beta}_2^T \dot{z}) \\ \dot{\hat{\gamma}}_1 &= -\kappa\Gamma_H||s||\hat{\gamma}_1 - \Gamma_H(\varphi(\hat{\gamma}_2^T \dot{z}) + \varphi'(\hat{\gamma}_2^T \dot{z})\hat{\gamma}_2^T \dot{z})s^T \\ \dot{\hat{\gamma}}_2 &= -\kappa\Gamma_H||s||\hat{\gamma}_2 - \Gamma_H\dot{z}s^T\hat{\gamma}_1^T\varphi'(\hat{\gamma}_2^T \dot{z}) \end{aligned} \quad (11)$$

with κ positive gain and $\Gamma_{\{M,C,H\}}$ positive matrices.

The error dynamics of the system in closed loop is obtained using the control law (10) with signals defined by (9):

$$\begin{aligned} M^*(z)\dot{s} \;=\; & -K_v s - C^*(\dot{z})s + w \\ & +\tilde{M}^*(z)\ddot{x}_r + \tilde{C}^*(\dot{z})\dot{x}_r + \tilde{H}(\dot{z}) \\ & +\epsilon(\dot{z}, \dot{x}_r, \ddot{x}_r) \end{aligned} \quad (12)$$

with $\epsilon(x,\dot{x}_r,\ddot{x}_r) = \epsilon_M(z)\ddot{x}_r + \epsilon_C(\acute{z})\dot{x}_r + \epsilon_H(\acute{z})$ and $\widetilde{(.)} = \widehat{(.)} - (.)$

Using the matrix approximations given by (5) and the development in the first order Taylor series of the activation function, the equation (12) can be rewritten:

$$\begin{aligned} M^*(z)\dot{s} &= -C^*(\acute{z})s - K_v s + w \\ &- \left(\hat{\alpha}_1^T\varphi'(\hat{\alpha}_2^T z)\tilde{\alpha}_2^T z + \tilde{\alpha}_1^T\varphi(\hat{\alpha}_2^T z)\right)\ddot{x}_r \\ &- (\hat{\beta}_1^T\varphi'(\hat{\beta}_2^T \acute{z})\tilde{\beta}_2^T \acute{z} + \tilde{\beta}_1^T\varphi(\hat{\beta}_2^T \acute{z}))\dot{x}_r \\ &- \left(\hat{\gamma}_1^T\varphi'(\hat{\gamma}_2^T \acute{z})\tilde{\gamma}_2^T \acute{z} + \tilde{\gamma}_1^T\varphi(\hat{\gamma}_2^T \acute{z})\right) + \varepsilon(\acute{z},\dot{x}_r,\ddot{x}_r) \end{aligned} \tag{13}$$

with

$$\varepsilon(\acute{z},\dot{x}_r,\ddot{x}_r) = \epsilon(\acute{z},\dot{x}_r,\ddot{x}_r) + \varepsilon_\varphi(\acute{z},\dot{x}_r,\ddot{x}_r)$$

and $\varepsilon_\varphi(\acute{z},\dot{x}_r,\ddot{x}_r)$ are disturbances due to the first order Taylor series approximations. They can be represented as follows:

$$\begin{aligned} \varepsilon_\varphi &= \tilde{\gamma}_1^T\varphi'(\hat{\gamma}_2^T \acute{z})\tilde{\gamma}_2^T \acute{z} - \gamma_1^T O(\hat{\gamma}_2^T \acute{z}) \\ &+ (\tilde{\beta}_1^T\varphi'(\hat{\beta}_2^T \acute{z})\tilde{\beta}_2^T \acute{z} - \beta_1^T O(\hat{\beta}_2^T \acute{z}))\dot{x}_r \\ &+ (\tilde{\alpha}_1^T\varphi'(\hat{\alpha}_2^T z)\tilde{\alpha}_2^T z - \alpha_1^T O(\hat{\alpha}_2^T z))\ddot{x}_r \end{aligned}$$

While using the Frobenius norm of matrix, we can write:

$$||\epsilon|| \le ||\epsilon_M||||\ddot{x}_r|| + ||\epsilon_C||||\dot{x}_r|| + ||\epsilon_H|| \tag{14}$$

and:

$$\begin{aligned} ||\epsilon_\varphi|| \le & (c_1||\tilde{\alpha}_1||||\alpha_2||||z|| + ||\alpha_1||||O_M||)||\ddot{x}_r|| \\ & + c_2||\tilde{\gamma}_1||||\gamma_2||||\acute{z}|| + ||\gamma_1||||O_H|| \\ & + (c_3||\tilde{\beta}_1||||\beta_2||||\acute{z}|| + ||\beta_1||||O_C||)||\dot{x}_r|| \end{aligned} \tag{15}$$

Let θ, $\hat{\theta}$ and $\tilde{\theta}$ matrices that contain respectively, all parameters, $\alpha_{1,2}, \beta_{1,2}, \gamma_{1,2}$, all estimated parameters $\hat{\alpha}_{1,2}, \hat{\beta}_{1,2}, \hat{\gamma}_{1,2}$, and all parameter errors $\tilde{\alpha}_{1,2}, \tilde{\beta}_{1,2}, \tilde{\gamma}_{1,2}$. Considering that:

- $||\theta||_F \le ||\theta||_{\max}$
- $||\left[\begin{array}{ccc} x_d & \dot{x}_d & \ddot{x}_d \end{array}\right]|| \le Y_d$ (Y_d is a bounded signal)
- $||\acute{z}|| \le p_1 Y_d + p_2||s||$ (p_1 and p_2 are positive real numbers function of λ)
- $||\ddot{x}_r|| \le p_3 Y_d + p_4||s||$

Recalling the Taylor approximation and manipulating the previous equations, we get:

$$||\epsilon_\varphi + \epsilon|| = ||\varepsilon|| \le A_1 + A_2||\tilde{\theta}|| + A_3||\tilde{\theta}||||s|| \tag{16}$$

where A_1, A_2, and A_3 are positive real constants function of $\theta_{\max}$, Y_d, λ, φ as well as approximation properties of the used network (ϵ_M, ϵ_C and ϵ_H).

For the system defined by the error equation (13), we propose the following term of compensation:

$$w = -K_\theta(||\hat{\theta}|| + ||\theta||_{\max})s \tag{17}$$

with:

$$K_\theta \ge A_3 \tag{18}$$

Note that in conception of compensation term (17), the estimation of $||\theta||_{\max}$ is necessary. However, the parameters of the network don't have any physical significance, this value can only be chosen in a heuristic way.

Let us consider the following Lyapunov function:

$$V = V_1 + V_2 + V_3 + V_4 \tag{19}$$

with:

$$\begin{aligned} V_1 &= \frac{1}{2}s^T M^*(z)s \\ V_2 &= \frac{1}{2}(\mathrm{tr}(\tilde{\alpha}_1^T\Gamma_M^{-1}\tilde{\alpha}_1) + \mathrm{tr}(\tilde{\alpha}_2^T\Gamma_M^{-1}\tilde{\alpha}_2)) \\ V_3 &= \frac{1}{2}(\mathrm{tr}(\tilde{\beta}_1^T\Gamma_C^{-1}\tilde{\beta}_1) + \mathrm{tr}(\tilde{\beta}_2^T\Gamma_C^{-1}\tilde{\beta}_2)) \\ V_4 &= \frac{1}{2}(\mathrm{tr}(\tilde{\gamma}_1^T\Gamma_H^{-1}\tilde{\gamma}_1) + \mathrm{tr}(\tilde{\gamma}_2^T\Gamma_H^{-1}\tilde{\gamma}_2)) \end{aligned}$$

its derivative with respect to the time is given by:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3 + \dot{V}_4 \tag{20}$$

with:

$$\begin{aligned} \dot{V}_1 &= s^T M^*(z)\dot{s} + \frac{1}{2}s^T \dot{M}^*(z)s \\ \dot{V}_2 &= \mathrm{tr}\left(\tilde{\alpha}_1^T\Gamma_M^{-1}\dot{\hat{\alpha}}_1\right) + \mathrm{tr}\left(\tilde{\alpha}_2^T\Gamma_M^{-1}\dot{\hat{\alpha}}_2\right) \\ \dot{V}_3 &= \mathrm{tr}\left(\tilde{\beta}_1^T\Gamma_C^{-1}\dot{\hat{\beta}}_1\right) + \mathrm{tr}\left(\tilde{\beta}_2^T\Gamma_C^{-1}\dot{\hat{\beta}}_2\right) \\ \dot{V}_4 &= \mathrm{tr}\left(\tilde{\gamma}_1^T\Gamma_H^{-1}\dot{\hat{\gamma}}_1\right) + \mathrm{tr}\left(\tilde{\gamma}_2^T\Gamma_H^{-1}\dot{\hat{\gamma}}_2\right) \end{aligned}$$

Using equations (13), (17) and (11), the expression of (20) is transformed as:

$$\begin{aligned} \dot{V} &= -s^T K_v s - \kappa||s||\,\mathrm{tr}(\tilde{\theta}^T(\tilde{\theta} + \theta)) \\ & \quad -K_\theta(||\hat{\theta}|| + ||\theta||_{\max})||s||^2 + s^T\varepsilon \end{aligned}$$

Using the following inequality:

$$\mathrm{tr}(\tilde{\theta}^T(\tilde{\theta}+\theta)) \geq ||\tilde{\theta}||^2 - ||\tilde{\theta}||||\theta||$$

we can write:

$$\begin{aligned}\dot{V} \;\leq\; & -\lambda_{\min}(K_v)||s||^2 - \kappa||s||(||\tilde{\theta}||^2 - ||\tilde{\theta}||||\theta||) \\ & -K_\theta(||\hat{\theta}|| + ||\theta||_{\max})||s||^2 + ||s||\,||\varepsilon||\end{aligned}$$

From inequalities (16) and (18), we can write $\dot{V} \leq 0$ if one of these conditions is verified:

$$||s|| \geq \frac{A_1 + (\theta_{\max} + A_2/\kappa)^2/4}{\lambda_{\min}(K_v)} \tag{21}$$

or:

$$||\tilde{\theta}|| \geq \frac{(\theta_{\max} + A_2/\kappa)}{2} + \sqrt{(A_1 + \frac{(||\theta|| + A_2/\kappa)^2/4)}{\kappa}} \tag{22}$$

$\dot{V}$ is negative outside of a compact set defined by (21) or (22), therefore s and $\tilde{\theta}$ are bounded. ■

Remark 1 *In (21), the radius of convergence of s depends of $\lambda_{\min}(K_v)$, so it is much smaller if K_v is large.*

Remark 2 *The proposed algorithm modifies parameters in the same way as the back-propagation. Indeed, the adaptation of parameters is achieved from the output toward the input of the neural network.*

The performance of the proposed controller have been tested in simulation on a three revolute joints planar robot manipulator ($n = 3$, $m = 2$).

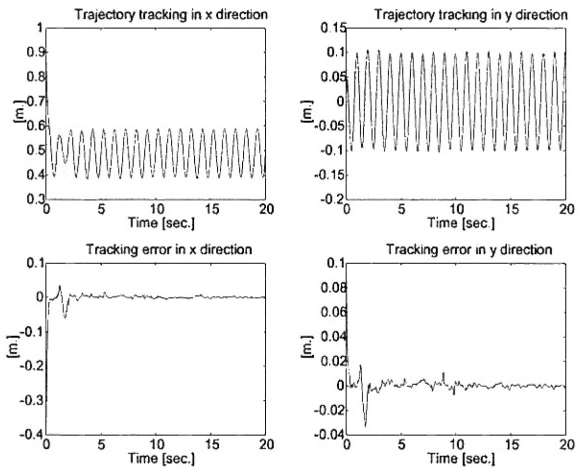

Figure 1: Trajectory tracking when the controller without optimisation is used.

We used one hidden layer neural networks with 5 neurons and tangent hyperbolic as activation function. The end-effector desired trajectory is a circle.

The control law parameters are chosen as:

$$K_v = 20 \quad \Lambda = 10 \quad \kappa = 0.01$$
$$K_w = 0.1 \quad \eta_{\{1,2\}\{m,c,h\}} = 5 \quad ||\theta||_{\max} = 10$$

The obtained results show a good trajectory tracking achieved by the robot end-effector (figures 1) but the joint positions are very large (figure 2).

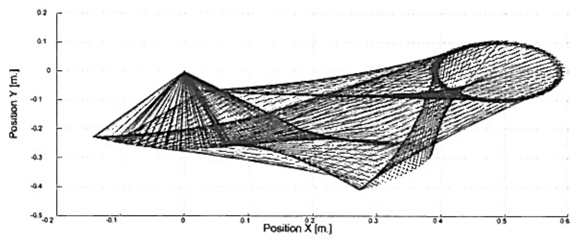

Figure 2: Configurations of the robot when the proposed controller without minimization is used. The desired trajectory is the circle.

We propose in what follows to minimize the energy of displacement via a supplementary variable h that depends on joint variables. It is deduced from convex quadratic criterion according to joint positions. The minimization of the cited criterion is equivalent to control the new variable h around zero.

5 Modified neural adaptive controller

We underlined through the previous results that the proposed controller was very efficient for trajectory tracking but unfortunately it does not minimize joint displacements. To improve the controller with regard to optimal robot configurations, a convex criterion function of joint displacements is introduced. The main objective is then to optimize this criteria via a new variable h which ensures convergence to global minimum. It is introduced in the previous controller to form an extended geometric model as follows:

$$X(q) = F(q) = \begin{bmatrix} x \\ h \end{bmatrix} = \begin{bmatrix} \phi(q) \\ h(q) \end{bmatrix} \tag{23}$$

The chosen expression of $h(q)$ results from minimization of an index $\Omega(q)$ function of joint displacements:

$$\Omega(q) = \frac{1}{2}\sum_{i=1}^{n} q_i^2 = \frac{1}{2}q^T q \tag{24}$$

This index is well convex with regard to the variable q. In this case, the problem is to find joint positions that minimizes $\Omega(q)$ under the geometric constraint $x = \phi(q)$. Thus, let's consider the following extended objective function:

$$\Theta(q, \lambda) = \Omega(q) + \lambda^T(x - \phi(q)) \tag{25}$$

λ is a vector of dimension m representing multipliers of Lagrange. The necessary condition of optimality can be written under the following form:

$$\begin{cases} \frac{\partial\Theta}{\partial q} = 0 \Rightarrow \frac{\partial\Omega}{\partial q} = \left(\frac{\partial\phi}{\partial q}\right)^T \lambda = J^T\lambda \\ \frac{\partial\Theta}{\partial\lambda} = 0 \Rightarrow x = \phi(q) \end{cases} \tag{26}$$

This condition can be rewritten using the non-zero matrix N translating the nul space of the Jacobean matrix J ($JN = 0$). Indeed, while multiplying the first equation of the system (26) by N^T, the necessary condition of optimality becomes:

$$N^T \frac{\partial\Omega(q)}{\partial q} = 0 \tag{27}$$

As the criterion Ω is convex with regard to q, we can conclude that the condition (27) is necessary and sufficient [7]. The choice of the variable h is then given by:

$$h(q) = N^T \frac{\partial\Omega(q)}{\partial q} = N^T q \tag{28}$$

and its desired trajectory value h_d is defined as:

$$h_d = \dot{h}_d = \ddot{h}_d = 0 \tag{29}$$

Let's consider the following reference signals for the new variable h:

$$\begin{cases} \dot{h}_r = \dot{h}_d + \lambda e_h; & e_h = h_d - h \\ \ddot{h}_r = \ddot{h}_d + \lambda \dot{e}_h; & \dot{e}_h = \dot{h}_d - \dot{h} \end{cases} \tag{30}$$

where:

$$\dot{h} = \frac{\partial h(q)}{\partial t} = \frac{\partial h(q)}{\partial q}\dot{q} = J_h \dot{q} \tag{31}$$

The augmented space desired trajectories are defined by the following vectors:

$$X_d = \begin{bmatrix} x_d \\ h_d \end{bmatrix}; \quad \dot{X}_d = \begin{bmatrix} \dot{x}_d \\ \dot{h}_d \end{bmatrix}; \quad \ddot{X}_d = \begin{bmatrix} \ddot{x}_d \\ \ddot{h}_d \end{bmatrix}$$

The extended model kinematics can be written as follows:

$$\dot{X} = J_e \dot{q}; \qquad J_e = \begin{bmatrix} J \\ J_h \end{bmatrix}$$

The proposed controller with these new formulations, as well as the algorithm of adaptation, have the same structure that those defined previously, and it is sufficient to replace x by X; x_d by X_d and J by J_e.

To show the efficiency of the modified proposed controller we have used the same gains as in the previous case and those relative to the new variable h are defined as:

$$K_v(3,3) = 10; \qquad \lambda = 5$$

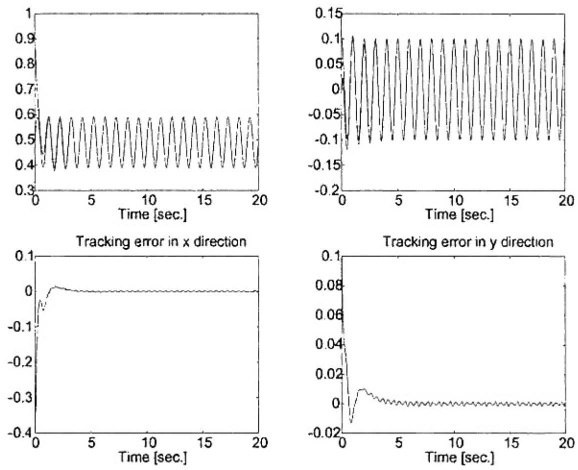

Figure 3: Trajectory tracking when the modified controller is used (with optimisation).

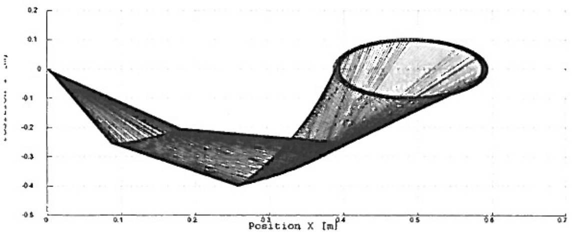

Figure 4: The configurations of the robot manipulator when neural controller with minimization of joint positions is used.

Simulation results with the modified controller are given by figures (3, 4 and 5). In addition to good trajectory tracking of the end-effector (figure 3) we can see that the robot uses minimal energy displacements (figure 4), especially in comparison with results given by the first neural controller (figure 2). The figure (5) shows the trajectory of the new variable h during

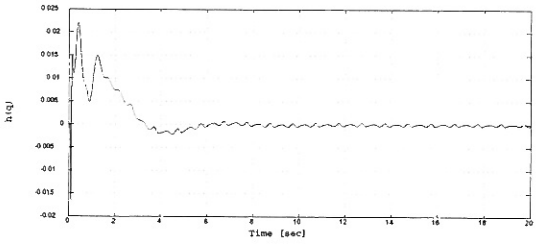

Figure 5: Trajectory of the variable h.

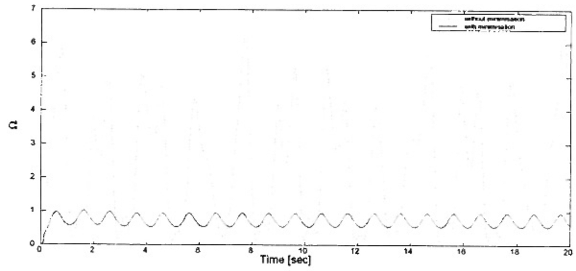

Figure 6: The index $\Omega(q)$ for the two situations (with minimization of energy in solid line and without minimization of energy in discontinuous line).

the first 20 seconds of simulation. We notice that this parameter converges to zero.

A comparison is well demonstrated on figure (6) representing the quadratic criterion given by equation (24) in the two cases: first neural controller (without constraint) and the modified one (with constraint). We notice that the behavior of the robot is better in the second situation. These results justify of lucid way the utility of the proposed method.

6 Conclusion

We proposed in this paper a stable adaptive neural controller designed in Cartesian space for redundant robot manipulator. The a priori knowledge on the dynamic model was reduced to its structure and its fundamental properties. Simulations achieved on the model of a 3-dof robot manipulator evolving in the vertical plan, show the efficiency of the proposed approach. It is necessary to notice that there is no online inversion what avoids numeric problems relative to matrix singularity. After a lapse of adaptation time, the desired trajectory tracking achieved by the end-effector is improved distinctly and the consumption of energy is minimized when the relative constraint to the minimization of the displacement energy is introduced. This principle of control with constraint can also be applied in the case of stubborn joint and obstacle avoidance.

The proposed method can be applied in the same way for parallel robot manipulators of which direct geometric and kinematics models are not easy to get.

References

[1] A. Cherif, V. Perdereau, and M. Drouin. On-line neural network algorithm for the constrained motion planning of redundant manipulators. In *Proc. of MCCS'97*, Minneapolis, USA, 1997.

[2] S. W. Kim, K. B. Park, and J. J. Lee. Redundancy resolution of robot manipulators using optimal kinematic control. In *Proc. of ICRA*, San Diego, 1994.

[3] C. A. Klein and C. H. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. on Systems Man and Cybernetics*, pages 245–250, 1983.

[4] A. De Luca, L. Lanari, and G. Oriolo. Control of redundant robots on cyclic trajectories. In *Proc. of ICRA*, Nice, France, 1992.

[5] D. Y. Meddah and A. Benallegue. A stable neuro-adaptive controller for rigid robot manipulators. *Journal od Intelligent and Robotic Systems*, 20:181–193, 1997.

[6] Y. Nakamura and H. Hanafusa. Optimal redundancy control of robot manipulators. *International Journal of Robotics Research*, 6(1):32–42, 1987.

[7] Amar Ramdane-Cherif. *Inversion des Modèles Géométrique et Cinématique d'un Robot Redondant : Une solution Neuronale Adaptative.* PhD thesis, Université Pierre et Marie Curie, 1998.

[8] R. M. Sanner and J.J. E. Slotine. Gaussian networks for direct adaptive control. *IEEE Trans. on Neural Networks*, 3(6):837–863, Nov. 1992.

[9] T. Shamir. Repeatability of redundant manipulators: Mathematical solution of the problem. *IEEE Tran. on Automatic Control*, 35:1004–1009, 1988.