

DYNAMIC EXTERNAL FORCE FEEDBACK LOOP CONTROL OF A ROBOT MANIPULATOR USING A NEURAL COMPENSATOR—APPLICATION TO THE TRAJECTORY FOLLOWING IN AN UNKNOWN ENVIRONMENT

FARID FERGUENE, REDOUANE TOUMI

Faculty of Electronics and Computer Science
USTHB University, BP 32, El-Alia, 16111, Bab-Ezzouar, Algiers, Algeria
e-mail: fferguene@yahoo.fr, red_toumi@hotmail.fr

Force/position control strategies provide an effective framework to deal with tasks involving interaction with the environment. One of these strategies proposed in the literature is external force feedback loop control. It fully employs the available sensor measurements by operating the control action in a full dimensional space without using selection matrices. The performance of this control strategy is affected by uncertainties in both the robot dynamic model and environment stiffness. The purpose of this paper is to improve controller robustness by applying a neural network technique in order to compensate the effect of uncertainties in the robot model. We show that this control strategy is robust with respect to payload uncertainties, position and environment stiffness, and dry and viscous friction. Simulation results for a three degrees-of-freedom manipulator and various types of environments and trajectories show the effectiveness of the suggested approach compared with classical external force feedback loop structures.

Keywords: force/position control, external structure, neural control, robot manipulator.

1. Introduction

Most of the industrial robot manipulators in the 1980s carried out tasks of welding, painting, and handling. They were generally controlled in position or in velocity by training. The end effector, by describing its trajectory, practically never came into direct contact with the environment. However, in recent years, robot manipulators have been required to perform tasks of great skills, e.g., in assembly plants. Currently, the tasks in robotics lead more and more to interactions of the robot with its environment. Typical tasks are assembly and machining objects. Indeed, when the robot is in contact with an environment, the end effector of the robot manipulator cannot move freely in all directions. The resulting movement is a constrained motion, and the kinematics chain of the manipulator are closed in the environment. Because of the stiffness of both, the weak variation of the end-point trajectory can induce high efforts which, if they are not taken into account by the controller, can cause rebounds and the deterioration of the effector and/or the environment. When a robot manipulator touches an environment, it is necessary to simultaneously control the force applied to the environment and the position of the end effector in

the same environment.

There are two main methods of force control in the literature: impedance control and hybrid control. In impedance control systems, the approach (Hogan, 1985) consists of adjusting the robot end effector position in response to the contact force in such a way that a target impedance force relationship is satisfied. In hybrid control systems, the control loop is divided into two directions for force control and position control. Controllers with two full degrees of freedom make both a position and a force control action available for each task coordinate; the actual control action is built using selection matrices which determine a proper contribution to the control law for each task coordinate matrices (Raibert and Craig, 1981). In order to ensure stability, the scheme must take account of dynamic coupling effects arising during the manipulator motion (Khatib, 1987). However, using selection matrices by these schemes may modify the norm of the desired command vectors of the controllers (Perdereau, 1991).

Another force/position structure can be devised that closes an outer force control loop around an inner position control loop, which may be the ordinary position control system of the manipulator. This structure is known in the literature as the external force feedback loop method

(Schuter and Van Brussel, 1988). This structure makes it possible to control the force and the position simultaneously without needing to use selection matrices. The force control loop is designed to prevail over the position control loop in the event of a conflict, variations in the planned task giving priority to the force control that is generally a conventional PID controller.

Several approaches are proposed in the literature to return this controller adaptive without influencing the existing motion control system of the manipulator. The works (Saadia *et al.*, 1997; Saadia, 1997) proposed an adaptive neural network control scheme for force control of an assembly robot. This approach was tested only with an assembly task where the velocity motion was chosen very slow. A fuzzy PI control scheme is used in the force controller design in (Song and Li, 1995). It is shown that this scheme is adaptive with respect to environment stiffness variation. The performances of this approach impose that the force data provided by the force/torque sensor are not contaminated by noise. The use of good filters is necessary. However, no dynamic structure—to our knowledge—has been proposed in the literature which would be able to deal with the problem of parameter variations of the robot itself as well as of the environment which have in reality an impact on the dynamic response of a manipulator in contact.

Actually, the design of the dynamic external force feedback loop control law is straightforward if the models of the robot manipulator and the environment are known with high precision. Uncertainties in these models degrade system performance. In practice, a complete dynamic model of a robot is not known exactly, and the environment position and stiffness are also approximately known.

During the past years, several approaches have been proposed to solve the problem of motion/force tracking in the presence of uncertainties: one can mention variable structure control (VSC) (Yao *et al.*, 1992), adaptive control (Singh and Popa, 1995; Whitcomb, 1997) and recently robust adaptive control (Chiu *et al.*, 2004). Although these techniques improve system performances, they tend to increase the complexity of system dynamics, which requires special attention regarding system stability. The use of neural networks in control systems has increased in recent years because such usage does not require a detailed knowledge of mathematical and control theories, thereby reducing the development cost of controllers for complex systems, particularly nonlinear ones.

Neural networks have previously been applied to the control of manipulators. Kiguchi and Fakuda (1995; 1997) proposed that the intelligent position/force controller of robot manipulators could be controlled using fuzzy neural approaches, and Nakawono and Katagiri (2003) proposed a force and position controller for robot manipulators using a neurocontroller with genetic algo-

rithm based training. The disadvantage of these approaches lies in the use of the selection matrices in the control loops for the former, whereas for the latter the choice of the input of the neural networks depends directly on the desired task, and their applications are limited to simple cases where the frame attached to the plane and reference frame are supposed to be parallel, which limits the diversity of the tasks in an imposed constrained motion.

Jung and Hsia (1995; 1998; 2000) proposed a force/impedance controller using a neural network compensator for uncertainties of robotic manipulators. The suggested method uses a trajectory modification algorithm technique to determine the reference trajectory. The disadvantage of this method is that it is based on the detection of the effort felt to select the modified trajectory depending on a desired force, which can be sensitive to the noise generated by the force sensors. In (Ferguene and Toumi, 2005), it was shown that neural networks are very effective in compensating for motion/force tracking in the case of a simple planar surface.

In this paper, we propose a new dynamic external force feedback loop scheme using a neural network compensator for uncertainties. This paper is organized as follows: The classical external force feedback loop structure is addressed in Section 2. In Section 3, robot model in contact with an environment is formulated. Section 4 develops the dynamic external force feedback loop structure, where an analysis of stability was carried out. In Section 5, we present our neural approach applied to the control scheme. Section 6 shows simulation results for tasks carried out on two different environment types (planar and curved surface), by using a three-degrees-of-freedom industrial robot PUMA560 (Armstrong and Khatib, 1986), to demonstrate the robustness of the proposed scheme to payload uncertainties, position and environment stiffness, as well as dry and viscous frictions. Finally, conclusions are drawn in Section 7.

2. External force feedback loop scheme

In this control system (Fig. 1), the force error is converted in a suitable reference trajectory for the inner position controller which is defined in the Cartesian space or the angular space (Schuter and Van Brussel, 1988). When there are no constraints due to the environment (E), the force controller will be without effect and the position controller will continue to control the system. When constraints appear, the force controller allows the control of the generated efforts. However, this controller scheme as proposed in the literature generally neglects the effect of the dynamics of the manipulator (terms due to the inertial, Coriolis and centrifugal torques) because the velocity is supposed to be very slow and the effect of the contact forces dominates the dynamic disturbances generated over the system,

which is not always true with respect to the tasks in the constrained motion carried out by the robot. In what follows we propose a new dynamic external force feedback loop scheme developed in Cartesian space coordinates.

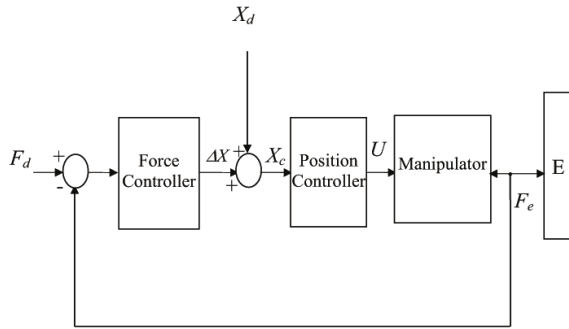


Fig. 1. External force feedback loop structure.

3. Dynamic equation of robot manipulators in contact with the environment

The dynamic equation of an n degrees-of-freedom manipulator in joint space coordinates can be given by

$$\tau = A(q)\ddot{q} + b(q, \dot{q})\dot{q} + g(q) + \tau_f + \tau_e, \quad (1)$$

where

- q , \dot{q} and \ddot{q} are the joint angle, the joint angular velocity, and the joint angular acceleration, respectively,
- $A(q)$ is the $n \times n$ symmetric positive-definite inertia matrix,
- $b(q, \dot{q})\dot{q}$ is the $n \times 1$ vector of Coriolis and centrifugal torques,
- $g(q)$ is the $n \times 1$ vector of gravitational torques,
- τ_f is the $n \times 1$ vector of actuator joint friction forces,
- τ_e is the $n \times 1$ vector of external disturbance joint torques,
- τ is the $n \times 1$ vector of actuator of actuator joint torque.

For simplicity, write

$$C(q, \dot{q})\dot{q} = b(q, \dot{q})\dot{q} + g(q).$$

Therefore, (1) can be rewritten as

$$\tau = A(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_f + \tau_e. \quad (2)$$

The relation between the joint velocity speed and the Cartesian space velocity can be expressed as

$$\dot{X} = J(q)\dot{q}, \quad (3)$$

where $J(q)$ represents the $n \times n$ Jacobian matrix of the manipulator that is supposed to be nonsingular. By differentiating (3), the Cartesian acceleration term can be found as

$$\ddot{X} = J(q)\ddot{q} + \dot{J}\dot{q}. \quad (4)$$

Then the equation of robot motion in the joint space can also be represented in Cartesian space coordinates by the relationship

$$\ddot{q} = J(q)^{-1}(\ddot{X} - \dot{J}\dot{q}). \quad (5)$$

Substituting (5) into (2) yields

$$\tau = A(q)J(q)^{-1}(\ddot{X} - \dot{J}\dot{q}) + C(q, \dot{q}) + \tau_f + \tau_e. \quad (6)$$

The actuator forces are related to the joint torques of the actuators through the Jacobian of the mechanism, i.e.,

$$\tau = J^T F. \quad (7)$$

The model of the robot in Cartesian space is thus given by

$$F = D\ddot{X} + H + F_f + F_e, \quad (8)$$

where

$$\begin{aligned} D &= (J^T)^{-1}AJ^{-1}, \\ H &= (J^T)^{-1}C - D\dot{J}J^{-1}\dot{X}, \\ F_f &= (J^T)^{-1}\tau_f, \\ F_e &= (J^T)^{-1}\tau_e. \end{aligned}$$

4. Dynamic external force feedback loop control scheme

A dynamic force/position control scheme is developed in this section according to the concept of the external force feedback loop. The dynamical model (8) represents a highly nonlinear and strongly coupled system for which the nonlinear dynamic decoupling approach (Boissonat and Faverjon, 1988) can be adopted in Cartesian space coordinates (see Fig. 2). The desired position X_d is modified by the term ΔX resulting from the force control loop. This leads to the following control structure:

$$F = \hat{D}M_d^{-1}U + \hat{H} + \hat{F}_e, \quad (9)$$

with

$$U = M_d\ddot{X}_d + K_v(\dot{X}_d - \dot{X}) + K_p(X_c - X), \quad (10)$$

where

- $X_c = X_d + \Delta X$ is the position command,

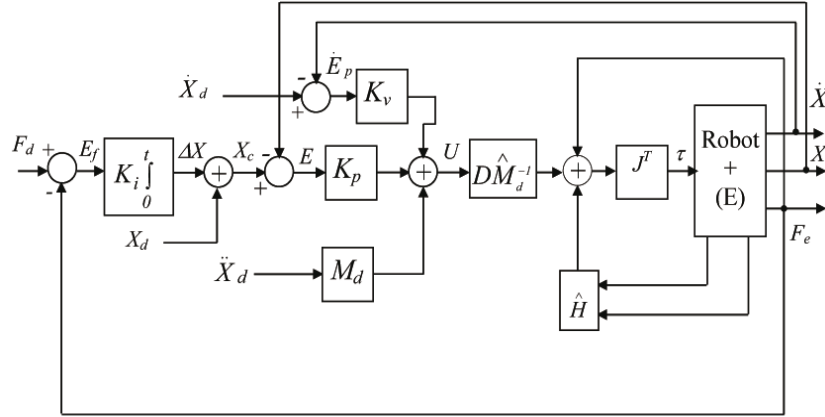


Fig. 2. Dynamic external force feedback control scheme.

- \hat{D}, \hat{H} are the estimates of D and H , respectively,
- F is the $n \times 1$ vector of generalized forces at the end effector,
- U is the $n \times 1$ vector of the decoupled end effector,
- M_d is the positive definite desired inertia matrix that must be diagonal to ensure dynamic decoupling, and
- \hat{F}_e is the measured contact force vector.

Combining (8)–(10) yields the closed loop tracking error dynamic equation

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E = \hat{D}^{-1}(\Delta D \ddot{X} + \Delta H + F_f + \Delta F_e), \quad (11)$$

where

$$\begin{aligned} \Delta D &= D - \hat{D}, \\ \Delta H &= H - \hat{H}, \\ \Delta F_e &= F_e - \hat{F}_e, \\ E &= X_c - X, \\ E_p &= X_d - X. \end{aligned}$$

In the ideal case, where the dynamic model is known perfectly, i.e., $\Delta D = \Delta H = \Delta F_e = 0$ and $F_f = 0$, the closed loop robot behavior is given by the following relation:

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E = 0. \quad (12)$$

According to Fig. 2, we can write

$$E = X_c - X = X_d + \Delta X - X = E_p + \Delta X. \quad (13)$$

If we replace (13) in (12), we will have

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p \Delta X = 0. \quad (14)$$

The force controller chosen here is given by

$$\Delta X = K_i \int_0^t E_f d\tau, \quad (15)$$

where $E_f = F_d - F_e$. In this case, (12) becomes

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p K_i \int_0^t E_f d\tau = 0. \quad (16)$$

This points out how E_f prevails over E_p . Indeed, in a steady state, the position error may be a constant with a force error equal to zero. This implies that the force control loop dominates over the position control loop in that the overall control system attempts to obtain $F_e = F_d$ even with a position error that differs from zero. M_d, K_p, K_v and K_i are diagonal matrices of the form λI (λ is a scalar and I is an identity matrix). It is noted that, if the task carried out by the effector ensures $F_d = F_e$, the system will be controlled by the usual dynamic Cartesian space control.

4.1. Stability analysis in elastic environments. We next want to analyze the performances of the proposed control when the manipulator interacts with the environment. Accurate modeling of the contact between the manipulator and the environment is usually difficult to obtain in an analytic form, due to the complexity of the physical phenomena involved in the interaction. It is then reasonable to resort to a simple and significant model relying on the robustness of the control system in order to reduce the effects of inaccurate modeling. The total elasticity, due to end effector force sensors and environments, is accounted through the compliance of the plane. Friction effects are neglected within the compliance of the operational range of interest. When the end effector is in contact with the plane, the model of the contact force is given by (only

translational motion and linear force components are studied, i.e., $n = 3$ is assumed)

$$F_e = K_e(X - X_e), \quad (17)$$

where X_e is a point of the plane at rest and K_e is a (3×3) constant symmetric stiffness matrix. Equation (17) induces the following relationship:

$$E_f = K_e E_p + F_d - K_e(X_d - X_e), \quad (18)$$

which, when substituted in (16), gives

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p K_i K_e \int_0^t E_p d\tau \\ = -K_p K_i \int_0^t F_d d\tau + K_p K_i K_e \int_0^t (X_d - X_e) d\tau \end{aligned} \quad (19)$$

Equation (19) represents a third-order vector system for which the gain matrices are to be suitably chosen in order to obtain asymptotic stability. Since the system is linear, its stability can be discussed in terms of the stability of the unforced system given by the following equation:

$$M_d \ddot{X} + K_v \dot{X} + K_p X + K_p K_i K_e \int_0^t X d\tau = 0. \quad (20)$$

Since the robot is in contact with its environment, in order to analyze (20) coupled by the matrix K_e , the following assumptions are made:

- The contact force is orthogonal to the plane for any vector $(X - X_e)$. Then, a base of $R(K_e)$ (the range space of matrix K_e) is the unit vector \vec{n} orthogonal to the plane and $\text{rank}(K_e) = 1$.
- All vectors $(X - X_e)$ which lie on the plane and do not contribute to the contact force are tangent in the plane. Then a base of $N(K_e)$ (the null space of the matrix K_e) is a pair of orthogonal unit vectors (\vec{t}_1, \vec{t}_2) lying in the plane.

Under these conditions, the columns of the matrix

$$R_{ot} = (n \ t_1 \ t_2) \quad (21)$$

form a set of orthonormal vectors constituting a base of \mathbb{R}^3 . According to the above remarks, the matrix K_e can be decomposed as

$$K_e = R_{ot} \text{diag}(k_e, 0, 0) R_{ot}^T = k_e n n^T, \quad (22)$$

with R_{ot} , as defined in (21), representing the rotation matrix from the frame attached to the plane to the reference frame. Then k_e is the stiffness coefficient, characterizing the contact along the orthogonal direction to the plane.

Let us introduce x_n, x_1, x_2 as the components of X in the frame attached to the plane defined by

$$\begin{pmatrix} x_n \\ x_1 \\ x_2 \end{pmatrix} = R_{ot}^T X. \quad (23)$$

Noting that

$$R_{ot} R_{ot}^T = n n^T + t_1 t_1^T + t_2 t_2^T = I, \quad (24)$$

Eqn. (20) can be written in the reference frame attached to the plane as follows:

$$\begin{aligned} n \left[m_d \ddot{x}_n + k_v \dot{x}_n + k_p x_n + k_p k_i k_e \int_0^t x_n d\tau \right] \\ + t_1 [m_d \ddot{x}_1 + k_v \dot{x}_1 + k_p x_1] \\ + t_2 [m_d \ddot{x}_2 + k_v \dot{x}_2 + k_p x_2] = 0. \end{aligned} \quad (25)$$

By the orthonormality of t_1, t_2 and n , (25) is equivalent to the following system of three scalar equations:

$$\begin{cases} m_d \ddot{x}_n + k_v \dot{x}_n + k_p x_n + k_p k_i k_e \int_0^t x_n d\tau = 0, \\ m_d \ddot{x}_1 + k_v \dot{x}_1 + k_p x_1 = 0, \\ m_d \ddot{x}_2 + k_v \dot{x}_2 + k_p x_2 = 0. \end{cases} \quad (26)$$

These equations show that the dynamics of the system (20) are decoupled in the space of variables x_n, x_1 and x_2 . The Routh criterion applied to the system (26) gives a necessary and sufficient condition of stability:

$$k_p, k_v, m_d, k_i > 0 \quad \text{and} \quad k_v > m_d k_i k_e,$$

with k_p, k_v, k_i and m_d being respectively the elements of matrices K_v, K_p, K_i, K_e and M_d .

Remark 1. An estimate of the stiffness coefficient k_e is used to tune the feedback gains based on a simple scalar differential equation.

Remark 2. In the case of a free motion ($F_d = F_e = 0$), (16) is uncoupled and becomes a second-order equation. The necessary and sufficient condition of its stability is checked for $m_d, k_v, k_p > 0$.

4.1.1. Case of unknown environment positions. If the environment position X_e is perfectly known, (19) can be rewritten while replacing X_d by X_e ($X_d = X_e$). If the environment position is unknown, then $X_d = X_e + \delta X_e = X_e'$, with δX_e being the position environment uncertainty. In this case, (19) will be written as

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p K_i K_e \int_0^t E_p d\tau \\ = -K_p K_i \int_0^t F_d d\tau - M_d \delta \ddot{X}_e - K_v \delta \dot{X}_e - K_p \delta X_e. \end{aligned} \quad (27)$$

The Laplace transform of (27) gives

$$E_p(s) = - [M_d s^3 + K_v s^2 + K_p s + K_p K_e K_i]^{-1} \times [K_p K_i F_d + (M_d s^3 + K_v s^2 + K_p s) \delta X_e(s)] \quad (28)$$

with $\delta X_e(s)$ as the input and $E_p(s)$ the output.

We suppose that the input (disturbance) is a step ($\delta X_e(s) = 1/s$), and that $F_d = 0$, the equilibrium point for the system (28), will be

$$E_p(\infty) = \lim_{s \rightarrow 0} s E_p(s) = 0 \Rightarrow X_\infty \rightarrow X_e \\ \Rightarrow F_e = K_e (X_\infty - X_e) \rightarrow 0.$$

This result proves that the system is compliant and that the force controller dominates the position controller.

4.1.2. Case of a curved surface. The analysis of the stability developed in Section 4.1 is valid only in the case of planar surfaces. If the surface is curved, the rotation matrix R_{ot} (Eqn. (21)) becomes a nontrivial function of a time varying contact point. Considering the complexity of the problem, we suppose that the surface is smooth and known analytically. In this case, we have the local stability of an equilibrium point. Indeed, the tangent plane at the point is a good approximation to the surface and R_{ot} can be considered to be constant. Quasi-static analysis can be made with respect to the succession of locally stable equilibrium points. The desired trajectory is selected such that R_{ot} varies slowly with respect to the overall system dynamics. Under these assumptions, the above stability conditions remain valid. In this case, the contact force model (17) can be rewritten in the form (Chiaverini and Sciavesco, 1993)

$$F_e = K_e(X_e(X))(X - X_e) \quad (29)$$

with

$$K_e(X_e) = k_e n(X_e) n^T(X_e). \quad (30)$$

However, bad performances of this control system can appear if the exact knowledge of the robot and environment models is not checked, i.e., ΔD , ΔH , ΔF_e and F_f are different from zero. We propose a new external force feedback loop structure based on a neural approach to compensate for these disturbances.

5. Neural structure of external force feedback loop control

In this section, we present a neural network controller designed to achieve disturbance rejection for the external force feedback loop system. The proposed scheme is given in Fig. 3. The idea is that the neural network output ϕ cancels out the uncertainties caused by an inaccurate

robot model in contact with its environment. The output signal ϕ of the neural network is added to the control input U whose dimension equals the number of degrees of freedom of the robot manipulator. Therefore, the vector ϕ is three dimensional. The control law becomes

$$F = \hat{D} M_d^{-1} (U + \phi) + \hat{H}. \quad (31)$$

Combining (31) and (8) yields the corresponding closed-loop error system as

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + F_e) - \phi. \quad (32)$$

In order to achieve the ideal case (Eqn. (12)), the output of the neural compensator is required to be

$$\phi = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + F_e). \quad (33)$$

Let

$$v = M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p. \quad (34)$$

Since the control objective is to generate ϕ to reduce v to zero in (32), we propose here to use v as the error signal for training the neural network. Thus the ideal value of ϕ at $v = 0$ is the same as the uncertainties given by (33). Equation (33) is nonlinear and depends on the position, velocity and acceleration of the end effector, as well as force contacts between the robot and environment. We propose this as neural network input. The weight updating law minimizes the objective function J , which is a quadratic function of the training signal v . The three-layer feedforward neural network structure is used as the compensator. It is composed of linear input and output layers and a nonlinear intermediate hidden layer that is a sigmoid function, which is bounded in magnitude between -1 and 1 in accordance with the following equation:

$$f(\cdot) = \frac{1 - \exp(-(\cdot))}{1 + \exp(-(\cdot))}. \quad (35)$$

The input is

$$[X_c^T(t) \quad \dot{X}_d^T(t) \quad \ddot{X}_d^T(t) \quad F_e^T(t)]^T.$$

According to Fig. 4, the output can be written in the form

$$\phi_k = \left[\sum_{j=1}^{n_H} w_{jk}^2 \left(\frac{1 - \exp(-(\sum_{i=1}^{n_I} x_{di} w_{ij}^1 + b_j^1))}{1 + \exp(-(\sum_{i=1}^{n_I} x_{di} w_{ij}^1 + b_j^1))} \right) \right] + b_k^2. \quad (36)$$

Here we use the following notation:

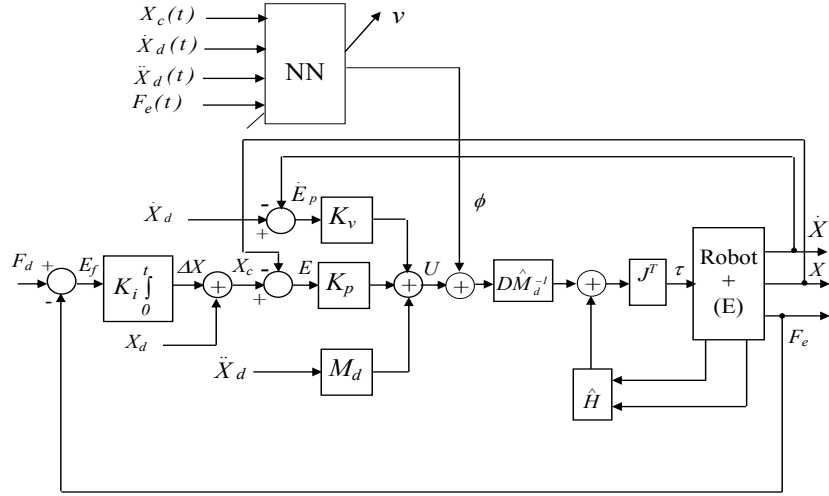


Fig. 3. Neural structure of the external force feedback loop.

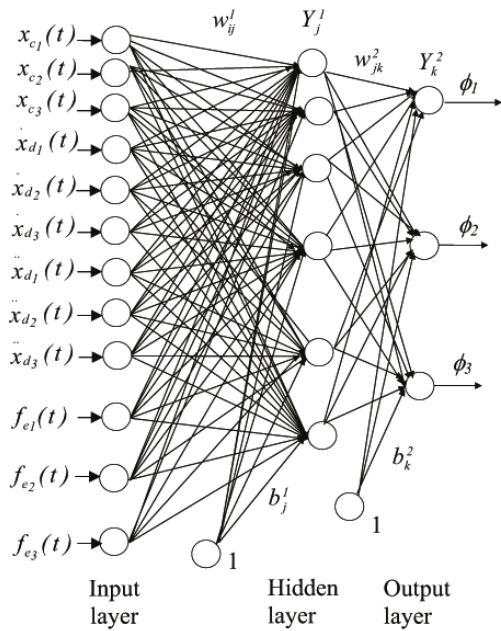


Fig. 4. External force feedback loop structure.

- n_I is the number of inputs,
- n_H is the number of hidden neurons,
- x_{di} is the i -th element of the input vector X_{NN} ,
- w_{ij}^1 are the weights between the i -th neuron of the input layer and the j -th neuron of the hidden layer,
- w_{jk}^2 are the weights of the j -th neuron of the hidden layer and the k -th neuron of the output layer,
- b_j^1 is the biased weight for the j -th hidden neuron,

- b_k^2 is the bias of the k -th neuron of the output.

Remark 3. In the absence of a general rule regarding the choice of the number of neurons of the hidden layer (Haykin, 1999), the minimal number of neurons of this layer is experimentally selected and it is fixed as six.

Remark 4. The network is adaptive online and the initialization of its weights is random. This neural controller performs the adaptive control process online.

The weights update law minimizes the objective function J , which is a quadratic function depending on the signal v of Eqn. (34) and is given by

$$J = \frac{1}{2} v^T v. \quad (37)$$

Differentiating (37) and making use of (32) yields the gradient of J ,

$$\frac{\partial J}{\partial w} = \frac{\partial v^T}{\partial w} v = -\frac{\partial \phi^T}{\partial w} v. \quad (38)$$

The adaptation of the weights is obtained by using the algorithm of retropropagation of the gradient given by the following equation:

$$\Delta w(t) = \eta \frac{\partial \phi^T}{\partial w} v + \mu \Delta w(t-1), \quad (39)$$

where η is the update rate and μ is the momentum coefficient (Haykin, 1999).

Remark 5. One can notice here that the convergence of this backpropagation algorithm depends on the good choice of parameters η and μ as well as the position and force controller parameters that must satisfy the stability condition given in Section 4.1.

Below we give the backpropagation algorithm:

Initialize weight vectors ($\Delta w = 0, w = 0$) for the work time simulation T_w .

1. Compute the network input.
2. Compute the network output by propagation the input vector:

$$Y_j^1 = \frac{1 - \exp\left(-\left(\sum_{i=1}^{n_I} x_{di}w_{ij}^1 + b_j^1\right)\right)}{1 + \exp\left(-\left(\sum_{i=1}^{n_I} x_{di}w_{ij}^1 + b_j^1\right)\right)},$$

$$\phi_k = Y_k^2 = \sum_{j=1}^{n_H} Y_j^1 w_{jk}^2 + b_k^2.$$

3. Compute the training signal v .
4. Compute Δw :

$$\Delta w_{ij}^1(t) = \eta \cdot 0.5 \cdot (1 - Y_j^1)^2 x_{di} \left[\sum_{k=1}^n v_k w_{jk}^2 \right] + \mu \Delta w_{ij}^1(t-1),$$

$$\Delta w_{jk}^2(t) = \eta v_k Y_j^1 + \mu \Delta w_{jk}^2(t-1),$$

$$\Delta b_k^2(t) = \eta v_k + \mu \Delta b_k^2(t-1),$$

$$\Delta b_j^1(t) = \eta \cdot 0.5 \cdot (1 - Y_j^1)^2 \left[\sum_{k=1}^n v_k w_{jk}^2 \right] + \mu \Delta b_j^1(t-1).$$

5. Adjust the network weights and then go to Step 1.

6. Implementation

In order to check the effectiveness of the neural force position control scheme, several simulations were carried out using a three-link robot manipulator. The parameters are taken from the first three links of the PUMA 560 arm (Fig. 5) (Armstrong and Khatib, 1986) (it is supposed that its last three joints are blocked). Simulations carried out aim at evaluating the performances of our scheme for constrained tasks.

Three simulation test are proposed in this article. The first consists in testing the compliance of the robot with respect to the unknown environment position. For that purpose, we introduced a position disturbance with a step motion. In this case, the environment is supposed to be planar and the frame attached to the plane is parallel to the reference frame. The second simulation suggested in this paper is a circular tracking in a tilted environment, and the third (most complex) consists in making the effector follow a curved surface. The controller gain can be obtained by successive tests. It is a delicate operation, in particular, when there are many tunable parameters. We propose a simple method which makes it possible to choose these controller gains.

Let us consider the third-order equation of the system (26). This equation can be written using the Laplace transform as follows:

$$\left(m_d s^3 + K_v s^2 + K_p s + K_p K_e K_i \right) x_n(s) = 0. \quad (40)$$

An adequate adjustment of the parameters can be obtained while forcing (40) to behave like a third-order system with frequency ω_0 and a damping ratio ξ adequately chosen, given by

$$G(s) = (s + \omega_0)(s^2 + 2\xi\omega_0 s + \omega_0^2). \quad (41)$$

Remark 6. The controller gains with the second-order equation of the system (26) will be chosen the same since the stability condition of the set remains checked for $m_d, k_v, k_p > 0$. The sampling period is fixed at $T_e = 5$ ms, and the initial weights of the network are fixed at zero ($w = 0$).

Simulation 1. A planar surface is considered in this simulation, characterized by $n = (1, 0, 0)^T$, and the environment position is at $X_e = (0.45, 0.04, 0.45)^T$ m (see Fig. 5). A step disturbance $\delta X_e = (0.005, 0, 0)^T$ m is in-

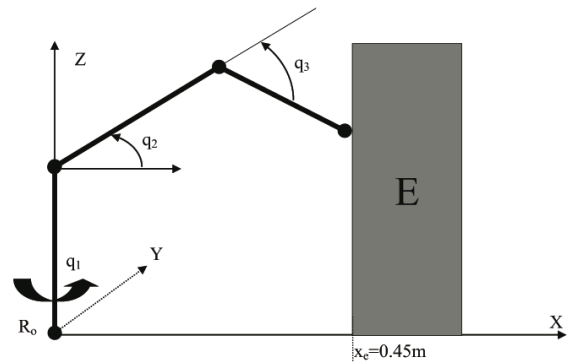


Fig. 5. Robot with tree axes and its environment.

roduced. The correct stiffness coefficient of the environment is assumed as $k_e = 10^4$ N/m. Let a known mechanical tool be attached to the third link ($m = 2$ kg). In order to minimize possible noise effects due to the sensors, we reduce the system bandwidth by choosing adequate (not too large) control gains (Dumas and Samson, 1987). For that purpose, we impose $w_0 = 20$ rad/s and $\xi = 1$. By identification, the control gains are selected as follows:

$$m_d = 1 \text{ kg}, \quad k_p = 1200 \text{ N/m}, \quad k_v = 60 \text{ s/m},$$

$$k_i = 6.66 \cdot 10^{-4} \text{ s}^{-1}.$$

The complete selection of the controller gains of the system is thus the following:

$$M_d = I, \quad K_p = 1200 \times I, \quad K_v = 60 \times I,$$

$$K_i = 6.66 \cdot 10^{-4} \times I.$$

The simulation results are given in Fig. 6. Note that, in the adapted case, the system is compliant with a complete recovery from the impact and a perfect accommodation of the end effector position with a null steady contact force achieved in less than 0.5 s, and this with respect to the unknown localization of the environment.

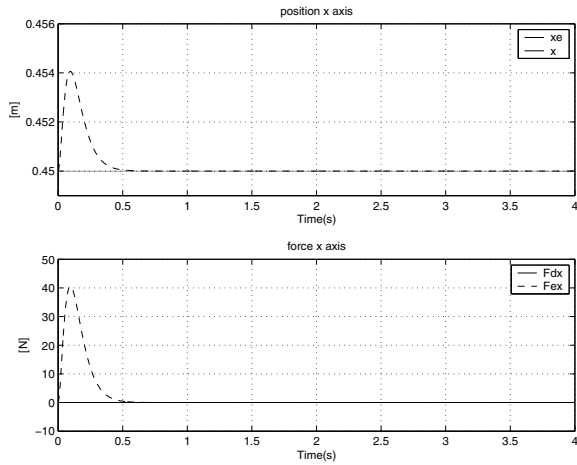


Fig. 6. Force and position tracking (update case).

To evaluate the robustness of our approach, we introduced the following uncertainties in the robot model and environment stiffness:

- Tool attached to the end effector $m = 10$ kg.
- Environment stiffness is increased to $k_e = 5 \times 10^4$ N/m.

We kept the same values of the parameters of the previous adjustment to which we added the parameters of the neural network (Fig. 4), chosen experimentally as $\eta = 0.0008$, $\mu = 0.9$. The simulation data showed that with a compensated system the rate of convergence is very fast, with a convergence time less than 0.5 s (Fig. 8). However, with an uncompensated system the force and position tracking errors are very large and amplified (Fig. 7). The disturbance signal vector v (Figs. 9 and 10) shows how this signal is reduced by our neural approach.

Simulation 2. In this simulation the proposed scheme is tested by tracking a circle in a tilted environment according to the normal $n = (\cos(\pi/4), 0, -\sin(\pi/4))^T$ with the desired effort $F_d = 5$ N shown in Fig. 11. The equations defining the desired circular trajectory in the refer-

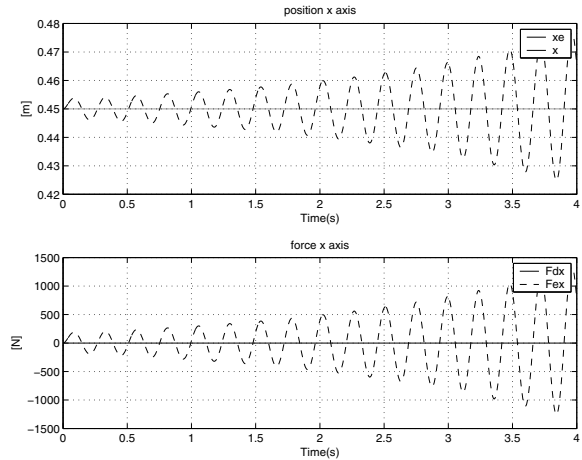


Fig. 7. Force and position tracking (without the neural compensator).

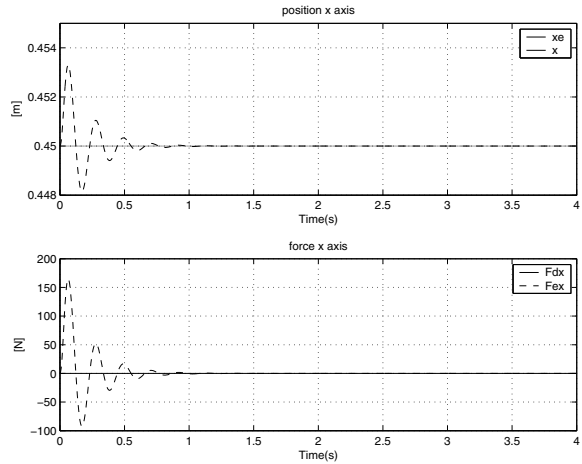


Fig. 8. Force and position tracking (with the neural compensator).

ence frame are

$$\begin{aligned} x(t) &= 0.106 \cos\left(\frac{2\pi}{T}t\right) + 0.424, \\ y_d(t) &= 0.15 \sin\left(\frac{2\pi}{T}t\right) + 0.3, \\ z_d(t) &= 0.1 \cos\left(\frac{2\pi}{T}t\right), \end{aligned}$$

with $T = 8$ s. To evaluate the robustness of our approach, we introduced the following uncertainties in the robot models and the environment:

- Tool attached to end effector $m = 10$ kg.
- Coulomb friction, and viscous friction forces added to each joint, given by

$$\tau_f = 0.8\dot{q} + 0.5\text{sign}(\dot{q}).$$

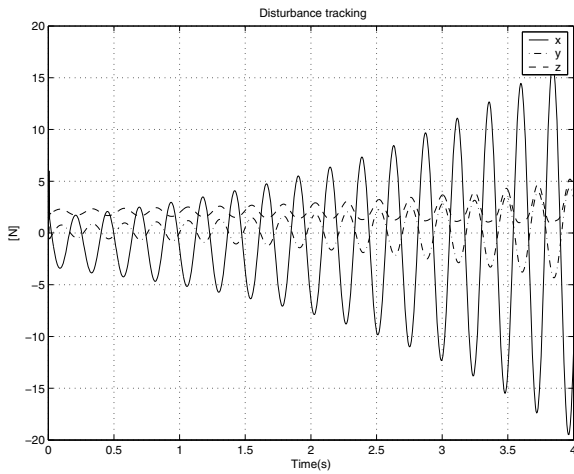


Fig. 9. Disturbance tracking (without the neural compensator).

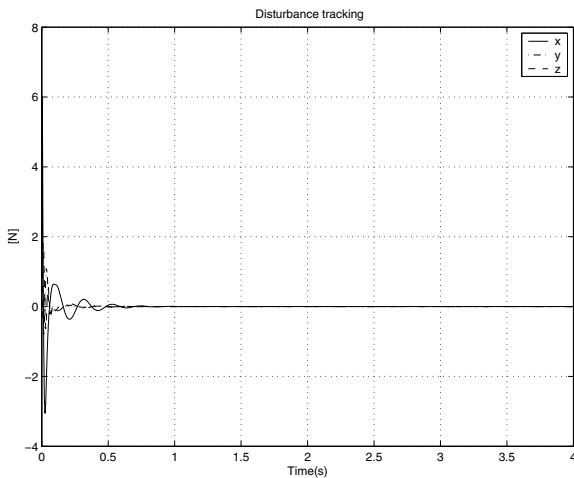


Fig. 10. Disturbance tracking (with the neural compensator).

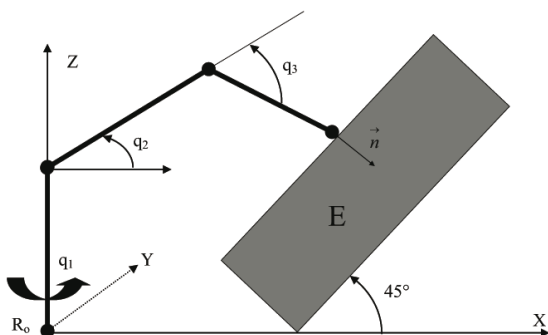


Fig. 11. Three-link robot with a tilted surface.

$$k_e = \begin{cases} 10000 \text{ N/m}, \\ 40000 \text{ N/m}, \\ 20000 \text{ N/m}, \\ 30000 \text{ N/m}. \end{cases}$$

Using the same values of the control parameters, the simulation data are represented by Figs. 12–18. One can notice a clear improvement in the trajectory following in the direction Y (free motion) (Figs. 12 and 13) by introducing the neural compensator (the errors in the directions X and Z represent the penetration of the effector in the wall). In the case of the effort response, Figs. 14 and 15 show that the introduction of abrupt variations in the stiffness generates a damped oscillation at the beginning of the variation in the stiffness, but it is quickly driven to the desired force with a convergence time less than 0.5 s by introducing the neural compensator. The comparison of Figs. 16 and 17 clearly shows that the disturbances are compensated by our neuronal approach. Figure 18 shows a circular trajectory tracking with a neural compensator.

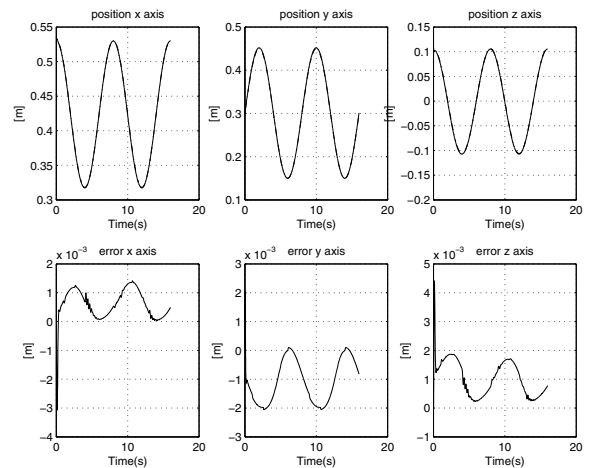


Fig. 12. Position tracking (without the neural compensator).

Simulation 3. The third simulation suggested in this article consists in making the end effector follow a trajectory on a curved surface. Let us recall that a task force/position control consists in following a trajectory on the surface considered by imposing a normal force F_d . The trajectory $X_d(t)$ to be followed by the end effector is supposed to be known at every time moment. When this trajectory is in the environment, it is additionally necessary to ensure a normal force whose norm $|F_d|$ is imposed. The components of the desired vector \vec{F}_d can then be deduced in the analysis of the task by the calculation of the normal \vec{n} at each point of the trajectory. Consider a sine-shaped surface given by

$$x_e = x_{es} + \Delta z \sin\left(\frac{2\pi}{l}(z_e - z_{es})\right), \quad (42)$$

- The discontinuous stiffness profile given by

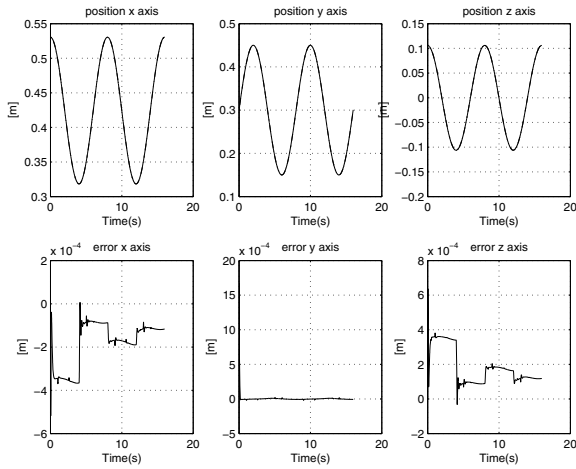


Fig. 13. Position tracking (with the neural compensator).

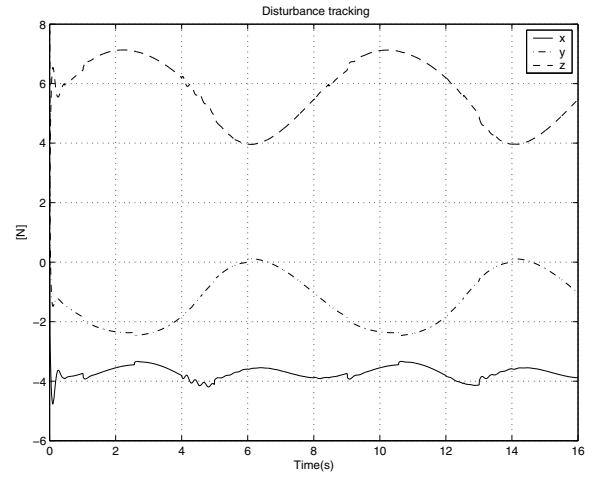


Fig. 16. Disturbance tracking (without the neural compensator).

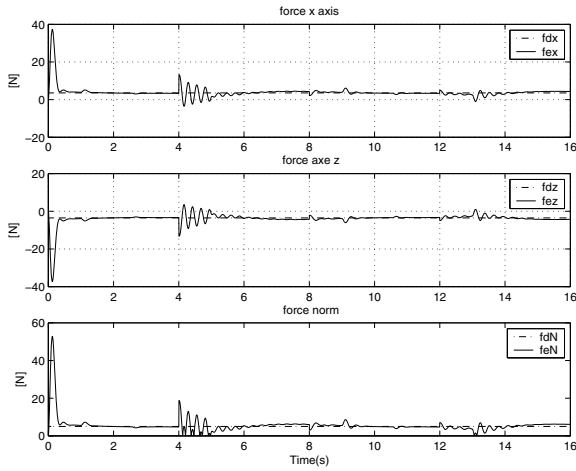


Fig. 14. Force tracking (without the neural compensator).

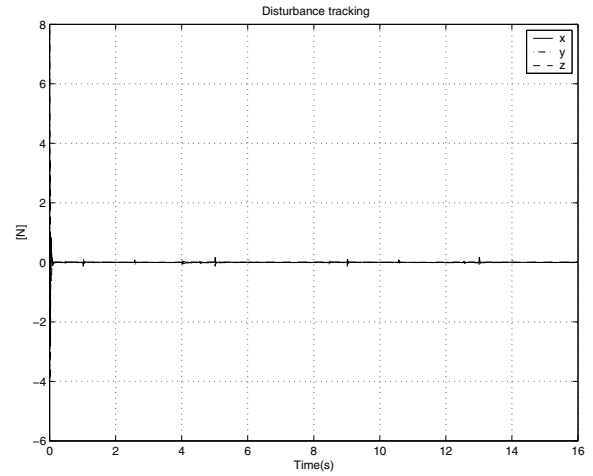


Fig. 17. Disturbance tracking (with the neural compensator).

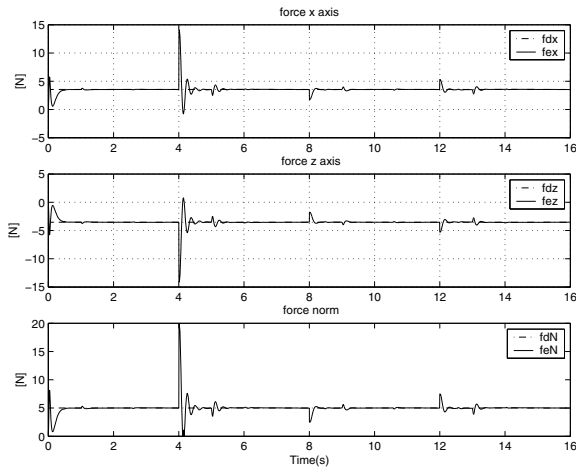


Fig. 15. Force tracking (with the neural compensator).

where x_{es} is the average amplitude of the surface, and Δz , l and z_{es} are respectively the amplitude, period and shift of the sine-wave along the z_e direction. The expression of the normal to the surface at the rest point is given by (Chiaverini and Sciavesco, 1993)

$$n(X_e) = \frac{1}{\sqrt{1 + \left(\Delta z \frac{2\pi}{l} \cos\left(\frac{2\pi}{l}(z_e - z_{es})\right) \right)^2}} \times \begin{pmatrix} 1 \\ 0 \\ \Delta z \frac{2\pi}{l} \cos\left(\frac{2\pi}{l}(z_e - z_{es})\right) \end{pmatrix}. \quad (43)$$

To validate our approach, we consider Eqn. (42) characterized by the values of the following parameters $z_{es} = 0$ m, $\Delta z = 0.03$ m, $l = 0.3$ m and $x_{es} = 0.2$ m. The velocity on the surface is equal to 0.05 m/s and the norm of the normal force applied is $|F_d| = 5$ N. The desired task consists in making the effector move from

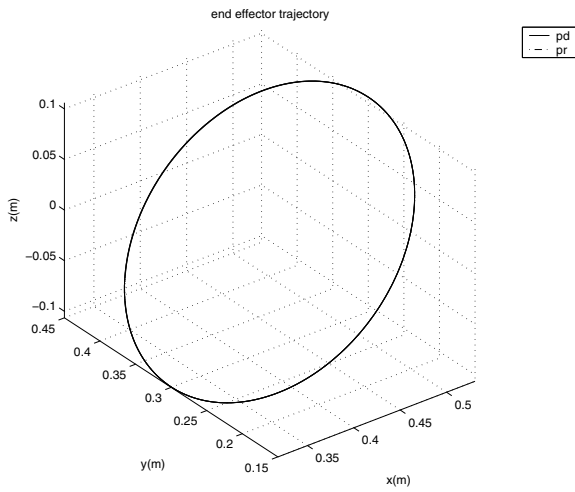


Fig. 18. End effector tracking of circular trajectory (with the neural compensator).

the position $A = (0.175, 0.3, 0.2)$ m towards the position $B = (0.175, 0.3, 0.5)$ m by continuing the surface defined by (42). We considered the same disturbances as previously, except that the variation in the stiffness in this case is supposed to be constant and known. The adjustment of the parameters of the controllers is maintained the same and the simulation results obtained are as follows (Figs. 19–25).

One can notice a clear improvement of the precision in position according to the fixed direction Y (Figs. 19 and 20) by introducing the neural compensator (the error directions X and Z represent the penetration of the effector in surface). In the case of the follow-up of the effort applied (Figs. 21 and 22), it is seen that the force tracking errors are very large for the classical structure, whereas the efforts are clearly followed while using the neuronal compensator, with a convergence time less than 0.5 s. The comparison between Figs. 23 and 24 clearly shows the compensation of the disturbances by our neural approach. Figure 25 represents the end point tracking of a curved surface in the plane (X, Z).

7. Conclusion

In this article, a neural network force/position strategy has been proposed. We used a classical external force feedback loop for which we developed its dynamic structure in Cartesian coordinates space. To improve the performance of this control structure, we implemented a neural compensator around this structure to compensate the possible disturbance on the system. This control structure was tested in simulations while supposing that the robot is equipped with effort, position, and velocity sensors and controlled by an effective calculator permitting implementation of this control scheme in real time. We considered

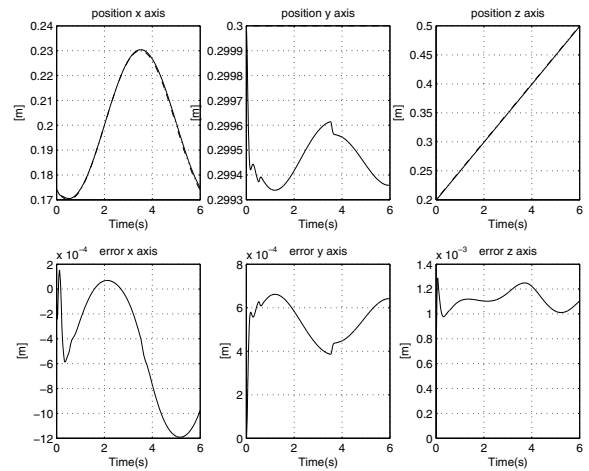


Fig. 19. Position tracking (without the neural compensator).

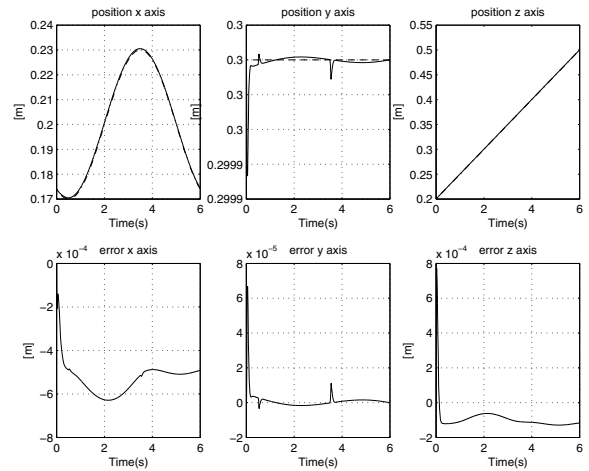


Fig. 20. Position tracking (with the neural compensator).

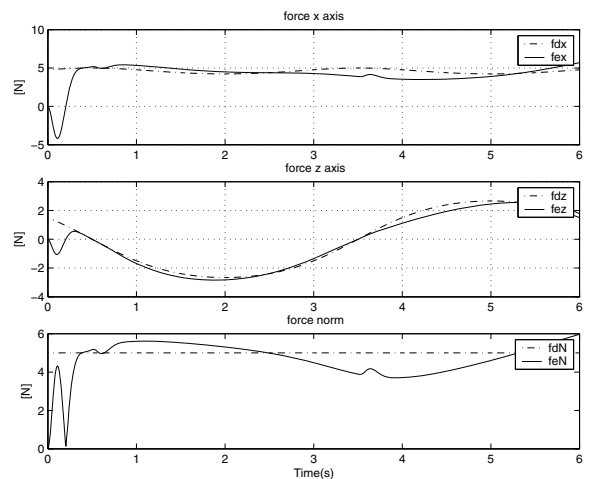


Fig. 21. Force tracking (without the neural compensator).

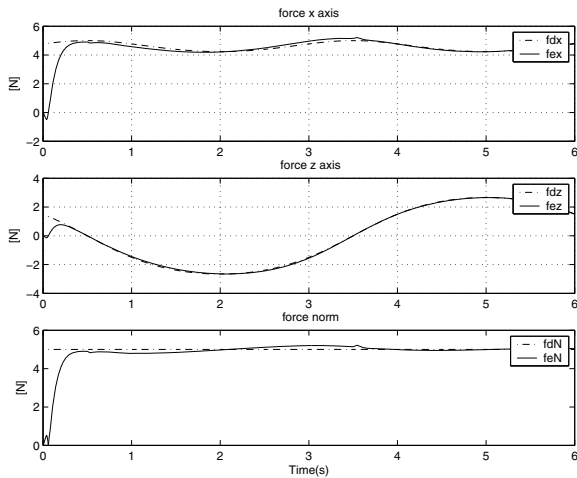


Fig. 22. Force tracking (with the neural compensator).

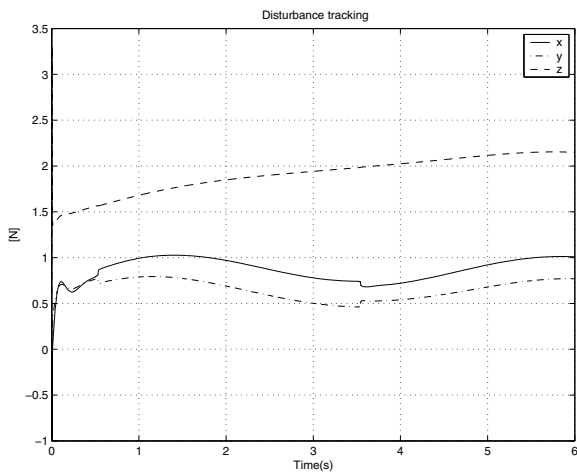


Fig. 23. Disturbance tracking (without the neural compensator).

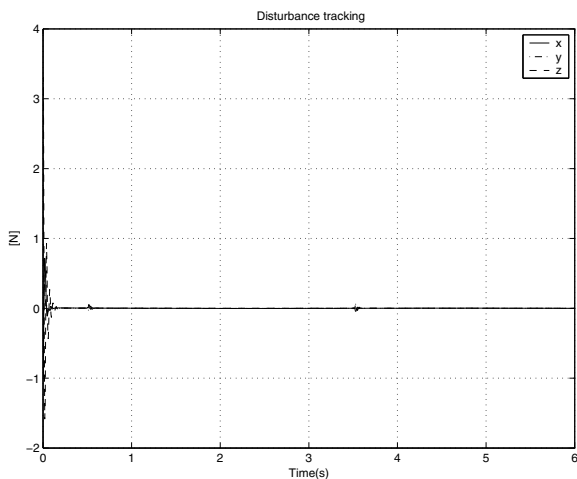


Fig. 24. Disturbance tracking (with the neural compensator).

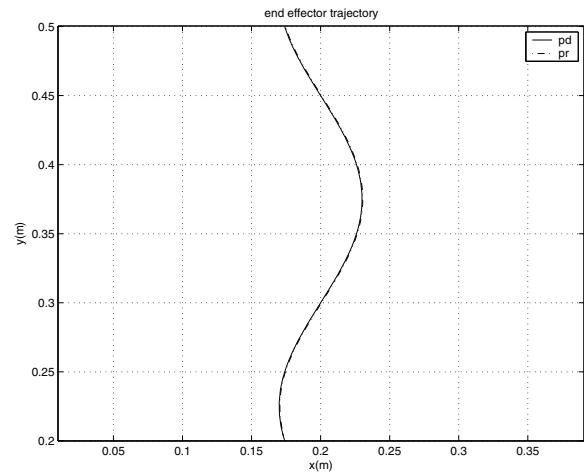


Fig. 25. End effector tracking of circular trajectory (with the neural compensator).

the uncertain case for the modeling of the robot and its environment (with variations in the payload, environment stiffness, and dry and viscous frictions). The results obtained by our approach improve in a meaningful way the performance of some conventional control structures proposed in the literature (Schuter and Van Brussel, 1988). Future work will consist in validating the results obtained on a complete simulation model of the robot PUMA 560 implementing the proposed control structures on a real robot and comparing our results with other approaches.

References

Armstrong, B. and Khatib, O. (1986). Explicit dynamic model and inertial parameters of PUMA 560 arm, *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, pp. 510–518.

Boissonat, J.D. and Faverjon, B. (1988). *Technique de la robotique*, Hermès, Paris.

Chiaverini, S. and Sciavisco, L. (1993). The parallel approach to force/position control of robotic manipulators, *IEEE Transactions on Robotics and Automation* 9(4): 361–373.

Degoulange, E. (1993). *Commande en effort d'un robot manipulateur à deux bras: Application au contrôle d'une chaîne cinématique fermée*, Ph.D. thesis, Université Montpellier II.

Chiu, C.S, Lian, K.Y. and Wu, T.C. (2004). Robust adaptive motion/force tracking design for uncertain constrained robot manipulators, *Automatica* 40(12): 2111–2119.

Dumas, R. and Samson, C. (1987). *Robust nonlinear control of robotic manipulators: Implementation aspect and simulation*, Internal report, INRIA, Rennes.

Fraisse, P. (1994). *Contribution à la commande robuste position/force des robots manipulateurs à architecture complexe: Application à un robot à deux bras*, Ph.D. thesis, Université Montpellier II.

- Ferguene, F. and Toumi R. (2005). A neural approach to force/position parallel control of robotic manipulators application to the follow-up of trajectory in unknown stiffness environment, *Proceedings of the International Conference on Computer Systems and Information Technology*, Algiers, Algeria, No. 1, pp. 247–251.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd Edn., Prentice-Hall, Englewood Cliffs, NJ.
- Hogan, N.R. (1985). Impedance control: An approach to manipulator. Parts I, II and III, *ASME Journal of Dynamics Systems, Measurement and Control* **107**(10): 1–24.
- Jung, S. and Hsia, T.C. (1995). On neural network application to robust impedance control of robot manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Japan, Vol. 1, pp. 869–874.
- Jung, S. and Hsia, T.C. (1998). Neural network impedance force control of robot manipulator, *IEEE Transactions on Industrial Electronics* **45**(3): 451–461.
- Jung, S. and Hsia, T.C. (2000). Robust neural force control design under uncertainties in robot dynamics and unknown environment, *IEEE Transactions on Industrial Electronics*, **47**(2): 403–412.
- Jung, S. and Hsia, T.C. (2001). Experimental studies of neural network impedance forces control for robot manipulators, *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 3453–3458.
- Khatib, O. and Burdick, J. (1986). Motion and force control of robot manipulators, *Proceeding of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, pp. 1381–1386.
- Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation, *ASME IEEE Journal of Robotics and Automation* **3**(1): 43–53.
- Kiguchi, K. and Fukuda, T. (1995). Robot manipulator contact force control application of fuzzy neural network, *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Japan, Vol. 1, pp. 875–880.
- Kiguchi, K. and Fukuda, T. (1997). Intelligent position/force controller for industrial robot manipulators-application of fuzzy neural networks, *IEEE Transactions on Industrial Electronics* **44**(6): 753–761.
- Nakawono, K. and Katagiri, M. (2003). Force and position control of robot manipulator using neurocontroller with GA based training, *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Kobe, Japan, pp. 1354–1357.
- Mills, J. and Goldenberg, A. (1989). Forces and position control of manipulators during constrained motion task, *IEEE Transactions on Robotics and Automation* **5**(1): 30–46.
- Perdereau, V. (1991). *Contribution à la commande hybride force/position: Application à la coopération de deux robots*, Ph.D. thesis, Université Pierre et Marie Curie.
- Raibert, M. and Craig, J. (1981). Hybrid position/force control of manipulators, *Transactions of ASME Journal of Dynamic Systems Measurement and Control* **103**(2): 126–133.
- Saadia, N. (1997). *Contribution à la commande hybride force-position des robots compliants selon une approche neuronale*, Ph.D. thesis, Université Paris XII.
- Saadia, N. Amirat, Y. and Djouani, J. (1997). Neural networks for force control of an assembly robot, *Proceedings of the IFAC Conference on Control of Industrial Systems*, Belfort, France, Vol. 2, pp. 551–557.
- Schuter, H. and Van Brussel, J. (1988). Compliant robot motion II with control approach based on external control loop, *International Journal of Robotics Research* **7**(4): 18–33.
- Singh, S.K. and Popa, D.O. (1995). An analysis of some fundamental problems in adaptive control of force and impedance behavior: Theory and experiments, *IEEE Transactions on Robotics and Automation* **11**(6): 912–921.
- Song, K.T and Li, H.P (1995). Design and experiment of fuzzy force controller for an industrial robot, *Proceedings of the National Science Council, Republic of China* **19**(1): 26–36.
- Surdilovic, D. (1999). Robust robot compliant motion control using intelligent adaptive impedance approach, *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, USA, pp. 2128–2133.
- Yao, B. Chan, S.P. and Wang, D. (1992). Robust motion and force control of robot manipulators in the presence of environmental constraint uncertainties, *Proceedings of the IEEE Conference on Decision and Control*, Tucson, AZ, USA, pp. 1875–1880.
- Saadia, N. (1997). Robust motion and force control of robot manipulators in the presence of environmental constraint uncertainties, *Proceedings of the IEEE Conference on Decision and Control*, Tucson, AZ, USA, pp. 1875–1880.
- Yoshikawa, T. (2000). Force control of robot manipulators, *Proceeding of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, pp. 220–226.
- Whitcomb, L.L. (1997). Adaptive model-based hybrid control of geometrically constrained robot arms, *IEEE Transactions on Robotics and Automation* **13**(1): 105–116.



Farid Ferguene received his engineer degree from the Houari Boumediene University of the Sciences and Technology, Algeria, in 1993. In 1994 he joined the Laboratory of Robotics (LRPE) of the same university, where he received his M.Sc. and Ph.D. degrees in 1997 and 2008, respectively. All his degrees are in electronic engineering. His research interests are force/position robot control and intelligent control.



Redouane Toumi has been a professor of automatic control at the Faculty of Electronics and Informatics of the Houari Boumediene University of Science and Technology in Algiers (USTHB, Algeria) since 1978. He is the head of the Laboratory of Robotics (LRPE). He graduated from the University of Algiers and received his Ph.D. degrees in electronics at the University of Paris (1977) and in automatic control (1985, USTHB, Algiers). His main research field covers stability control of systems and robotics.

Received: 12 June 2007

Revised: 27 February 2008