HYBRID IMPEDANCE CONTROL OF ROBOT MANIPULATORS WITH NEURAL NETWORKS COMPENSATION

**Silvério J. C. Marques Luís Filipe Baptista
José M. G. Sá da Costa**

*Technical University of Lisbon, Instituto Superior Técnico
Department of Mechanical Engineering, GCAR/IDMEC
Avenida Rovisco Pais, 1096 Lisboa Codex, Portugal
Phone: +351-1-8418190, Fax: +351-1-8498097
E-mail:* marques@gcar.ist.utl.pt

Abstract: This article presents an hybrid impedance control approach with neural network for the compensation of robot manipulator's modelling errors. The proposed algorithm, consists of an outer hybrid impedance control loop that generates the reference acceleration to an inner inverse dynamics control loop. In order to improve the controller robustness, a compensation action of the manipulator modelling errors, is introduced, acting on the target acceleration. This compensation action is based on a neural network model achieved by minimising the modelling errors along the manipulator trajectory. The neural network algorithm uses an error training signal to model errors, that is minimised along the trajectory. The performance of the hybrid impedance control system with neural network compensation, is illustrated by computer simulations with a two degree-of-freedom PUMA 560 robot, which end-effector is forced to move along a frictionless surface located perpendicular to a horizontal plane. The results obtained, reveal an accurate force tracking and position control in robotic tasks, where is assumed significant uncertainties in the robot dynamic model.

Keywords: Robotics, Neural networks, Hybrid impedance control, Force control, Inverse dynamics control

1. INTRODUCTION

Force control of robotic manipulators has been an active area of research during the last decade. During that time, many approaches to force control have been proposed. Whitney (Whitney 1987), surveyed these approaches in 1987. To date the core of the force control approaches can be classified as either impedance control or hybrid control. The approaches classified as impedance control (Hogan 1985), do not attempt to control force explicitly but rather to control the relationship between force and position of the end-effector in contact with the environment. Controlling the position, therefore leads to an implicit force control. Alternatively, hybrid control (Raibert and Craig 1981, Yoshikawa 1987), separates the robotic force task into two subspaces, the force controlled subspace and the position controlled subspace. Two independent controllers are then designed for each subspace. In 1988, Anderson and Spong (Anderson and Spong 1988), proposed a new method to combine these two general algorithms into a uniform control strategy. This approach termed Hybrid Impedance Controller combines the strategies of hybrid control and impedance control, and can be reduced to either approach. It separates the task space into two subspaces, an explicit force controlled subspace and an impedance controlled position subspace.

In the literature, it has been shown that neural networks are very efficient to compensate errors in trajectory tracking of robots in free motion, as in the paper of Tso and Lin (Tso and Lin 1996) and Kim and Lewis (Kim and Lewis 1996), and in force control, as in the papers of Jung and Hsia (Jung and Hsia 1995*a*), (Jung and Hsia 1995*b*) where they applied on-line neural networks schemes to robot control in constrained motion.

In this paper, the hybrid impedance control approach proposed in (Anderson and Spong 1988) is extended to include a neural network compensation of the manipulator modelling errors. The neural network compensation action follows the work of Jung and Hsia (Jung and Hsia 1995*b*) for an impedance controller, but extend the scheme to the hybrid impedance approach, in order to yield a stable and robust force control system.

This paper is organised into 6 sections. Section 2 summarises the modelling of the manipulator and environment. Section 3 presents the hybrid impedance controller. Section 4 discusses the neural network compensation action and the overall control system. Simulation results are presented and analysed in Section 5. Finally, conclusions are drawn in Section 6.

2. MODELLING

2.1 *Manipulator dynamics*

The dynamic equation of rigid manipulator in constrained motion, is given by

$$\boldsymbol{M}(q)\ddot{q} + \boldsymbol{C}(q,\dot{q})\dot{q} + \boldsymbol{g}(q) + \boldsymbol{d}(\dot{q}) = \boldsymbol{\tau} - \tau_e \quad (1)$$

where q, $\dot{q}$ and $\ddot{q}$ correspond to the $(n \times 1)$ vectors of joints angular positions, velocities and accelerations, respectively. $\boldsymbol{M}(q)$ represents the $(n \times n)$ symmetric positive definite inertia matrix, $\boldsymbol{C}(q,\dot{q})\dot{q}$, describes the $(n\text{x}1)$ vector of Coriolis and centripetal effects, $\boldsymbol{g}(q)$ accounts for gravitational terms and $\boldsymbol{d}(\dot{q})$ for the frictional terms in joint coordinates, respectively. The term $\boldsymbol{J}$ represents the Jacobian matrix that relates the joint velocity to the linear and angular velocities of the end-effector, the term $\boldsymbol{\tau}$ represents the $(n \times 1)$ vector of applied joint torques and $\boldsymbol{f}_e = J^{-T}\tau_e$ denotes the $(n \times 1)$ vector of generalised forces exerted by the end-effector on the environment and measured by the wrist force sensor. If the robot is not redundant and it is not in a singular position, (1) can be written in cartesian space as

$$\boldsymbol{M}_x(x)\ddot{x} + \boldsymbol{C}_x(x,\dot{x})\dot{x} + \boldsymbol{g}_x(x) + \boldsymbol{d}_x(\dot{x}) = \boldsymbol{f} - \boldsymbol{f}_e \quad (2)$$

where x is a six-dimensional vector representing the position and orientation of the manipulator end-effector and the terms of (2), are given by

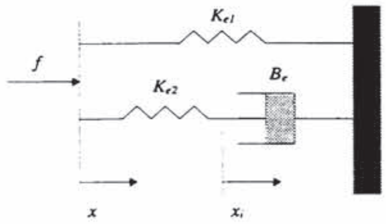

Fig. 1. Environment dynamic model.

$$\boldsymbol{M}_x(x) = \boldsymbol{J}^{-T}\boldsymbol{M}(q)\boldsymbol{J}^{-1} \quad (3)$$

$$\boldsymbol{C}_x(x,\dot{x}) = \boldsymbol{J}^{-T}(\boldsymbol{C} - \boldsymbol{M}(q)\boldsymbol{J}^{-1}\dot{\boldsymbol{J}})\boldsymbol{J}^{-1} \quad (4)$$

$$\boldsymbol{g}_x(x) = \boldsymbol{J}^{-T}\boldsymbol{g}(q) \quad (5)$$

$$\boldsymbol{d}_x(\dot{x}) = \boldsymbol{J}^{-T}\boldsymbol{d}(\dot{q}) \quad (6)$$

$$\boldsymbol{f} = \boldsymbol{J}^{-T}\boldsymbol{\tau} \quad (7)$$

2.2 *Environment*

The accurate modelling of the contact between the manipulator and the environment is usually difficult to obtain, due to the complex physical behaviour of the robot interaction with the contact surface. In this work, the environment is modelled as proposed by Mills and Lockhorst (Mills and Lockhorst 1993), and are assumed the following conditions:

- The manipulator is a rigid structure, which do not deform as a result of the contact with the working surface;
- Frictionless contact between the end-effector and the surface;
- The environment is modelled by noninteracting dynamic subsystems.

This environment model guarantee an important property, i.e. the forces resulting from the contact with the surface vary continuously in time. The continuous behaviour of the environment model during initial contact is ensured by the introduction of a serial spring/damper system. Figure 1 shows the environment model with linear springs $\boldsymbol{K}_{ei}$ and damper $\boldsymbol{B}_e$. The equations of the model (Mills and Lockhorst 1993), are given by:

$$\boldsymbol{f}_e = (\boldsymbol{K}_{e1} + \boldsymbol{K}_{e2})x - \boldsymbol{K}_{e2}x_i \quad (8)$$

$$\dot{x}_i = \frac{\boldsymbol{K}_{e2}}{\boldsymbol{B}_e}(x - x_i) \quad (9)$$

3. HYBRID IMPEDANCE CONTROLLER

Anderson and Spong (Anderson and Spong 1988) proposed a combination of the two main classes of robotic force control methodologies: impedance control and hybrid control, and termed this controller as *Hybrid Impedance Controller*. This control algorithm uses a inner and outer loop control

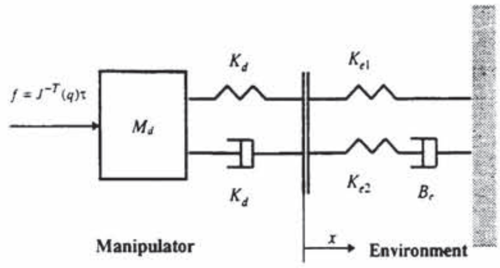

Fig. 2. One d.o.f. cartesian manipulator: $\boldsymbol{B}_e$, $\boldsymbol{K}_{e1}$, $\boldsymbol{K}_{e2}$ - environment parameters. $\boldsymbol{M}_d$, $\boldsymbol{B}_d$, $\boldsymbol{K}_d$ - Desired robot impedance parameters.

concept. The inner loop consists of a linearization of the robot dynamics and the outer loop consists of determining an input to the inner loop to achieve a broader goal such as trajectory tracking or force following. The design of the hybrid impedance controller is accomplished in the task space. The task space is split into two independently controlled subspaces, an impedance controlled subspace and a force controlled subspace. The object of the hybrid impedance controller is to design an algorithm for each subspace such that the dynamics exhibited by the system are replaced by more desirable dynamics. Figure 2 represents the one-degree-of-freedom cartesian manipulator case.

3.1 *Force controlled subspace*

In the force controlled subspace, the aim of the controller is to follow a desired force trajectory (Bickel and Tomizuka 1995). Additionally, in order to reduce impact forces when the manipulator loses and then regains contact with the environment, it is desirable to keep velocities low even in the presence of a large force error. This dynamics is given by

$$\boldsymbol{M}_d\ddot{x} + \boldsymbol{B}_d\dot{x} = \boldsymbol{f}_d - \boldsymbol{f}_e \tag{10}$$

Solving (10) for acceleration, the desired reference acceleration for the force controlled subspace is obtained

$$\ddot{x}_{tf} = \boldsymbol{M}_d{}^{-1}(-\boldsymbol{B}_d\dot{x} + \boldsymbol{f}_d - \boldsymbol{f}_e) \tag{11}$$

If this acceleration can be tracked, the desired dynamics given in (10) is feasible.

3.2 *Position controlled subspace*

In the impedance controlled subspace the objective of the controller is to track a desired position trajectory in the absence of any forces implied by the environment and to react as a desired impedance in the presence of these environmental forces (Bickel and Tomizuka 1995). The dynamics can be represented by

$$\boldsymbol{M}_d\ddot{e} + \boldsymbol{B}_d\dot{e} + \boldsymbol{K}_d e = \boldsymbol{f}_e \tag{12}$$

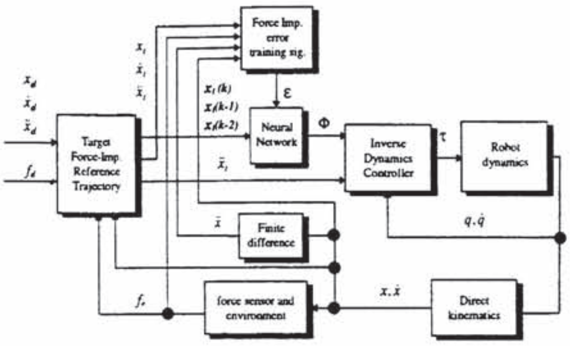

Fig. 3. Hybrid impedance control scheme with neural network compensation.

where $\ddot{e} = \ddot{x}_d - \ddot{x}$, $\dot{e} = \dot{x}_d - \dot{x}$ and $e = x_d - x$. Solving (12) for acceleration and considering $\boldsymbol{f}_e = 0$ for the impedance controlled subspace, the desired subspace reference acceleration is obtained

$$\ddot{x}_{ti} = \ddot{x}_d + \boldsymbol{M}_d{}^{-1}(\boldsymbol{B}_d\dot{e} + \boldsymbol{K}_d e) \tag{13}$$

Note that (11) and (13) constitute an outer loop control design that provides the following target acceleration

$$\ddot{x}_t = \left[\, \ddot{x}_{tf} \ \ddot{x}_{ti} \,\right]^T \tag{14}$$

to the inner inverse dynamics control loop in order to linearize the robot dynamics and track $\ddot{x}_t$.

4. NEURAL NETWORK COMPENSATOR

The neural network compensation scheme proposed in this paper, is an extension of the approach given in (Jung and Hsia 1995*b*) for the case of two subspaces. In this work, an hybrid impedance control structure is adopted and a different error training signal for the neural network compensator scheme is used, as shown in Figure 3. In this control approach, the target accelerations (14) obtained from the outer hybrid impedance control loop are corrected in order to compensate the robot modelling errors. This correction operates in a differentiated form on the two subspaces: force subspace when external forces acts in those directions and impedance subspace where no constraints exists. To obtain those corrections, consider the inverse dynamics control law

$$\boldsymbol{f} = \hat{\boldsymbol{M}}_x(\ddot{x}_t + \Phi) + \hat{\boldsymbol{C}}_x + \hat{\boldsymbol{g}}_x + \boldsymbol{f}_e \tag{15}$$

where $\hat{\boldsymbol{M}}_x$, $\hat{\boldsymbol{C}}_x$, $\hat{\boldsymbol{g}}_x$ are estimates of $\boldsymbol{M}_x$, $\boldsymbol{C}_x$, $\boldsymbol{g}_x$ in (2), Φ is the neural network output and $\ddot{x}_t$ is given by (14). Considering the uncertainties in the manipulator model (2), as $\Delta\boldsymbol{M}_x = \boldsymbol{M}_x - \hat{\boldsymbol{M}}_x$, $\Delta\boldsymbol{C}_x = \boldsymbol{C}_x - \hat{\boldsymbol{C}}_x$, and $\Delta\boldsymbol{g}_x = \boldsymbol{g}_x - \hat{\boldsymbol{g}}_x$, it is not possible to obtain an ideal target acceleration when no compensation exists for those modelling errors. One way to compensate those uncertainties is to add the output of a neural network (Φ) to the acceleration $\ddot{x}_t$, which is trained by on-line learning or pattern learning. After some mathematical manipulations, the closed-loop error equations for force and impedance subspaces, and then use those errors as a training vector signal to the neural network.

4.1 *Force-position error modelling signal*

In the force subspace, combining (11) and (15) with (2), yields the following equation

$$\boldsymbol{M_d}^{-1}(-\boldsymbol{B_d}\dot{x}+\boldsymbol{f_d}-\boldsymbol{f_e})-\ddot{x} = \hat{\boldsymbol{M_x}}^{-1}\Psi_e-\Phi \quad (16)$$

where Ψ_e is the uncertainty in the robot model, given by

$$\Psi_e = \Delta\boldsymbol{M_x}\ddot{x} + \Delta\boldsymbol{C_x} + \Delta\boldsymbol{g_x} + \boldsymbol{d_x} \quad (17)$$

Thus is possible to define the force subspace error signal ε_f as

$$\varepsilon_f = -\ddot{x} + \boldsymbol{M_d}^{-1}(-\boldsymbol{B_d}\dot{x} + \boldsymbol{f_d} - \boldsymbol{f_e}) \quad (18)$$

In the impedance subspace, combining (13) and (15) with (2), yields the following equation

$$\ddot{e} + \boldsymbol{M}_d^{-1}(\boldsymbol{B_d}\dot{e} + \boldsymbol{K_d}e) = \hat{\boldsymbol{M_x}}^{-1}\Psi_e - \Phi \quad (19)$$

Thus, is possible to define the impedance subspace error signal ε_i as

$$\varepsilon_i = \ddot{e} + \boldsymbol{M_d}^{-1}(\boldsymbol{B_d}\dot{e} + \boldsymbol{K_d}e) \quad (20)$$

The force/impedance subspaces error signal ε

$$\varepsilon = \begin{bmatrix}\varepsilon_f & \varepsilon_i\end{bmatrix}^T \quad (21)$$

can then be used as a training signal of the NN scheme, in order to provide a corrective action to the target acceleration $\ddot{x}_t$. The corrected acceleration, is then used to drive the robot minimising the manipulator modelling errors along the trajectory. The objective is to reduce to zero or minimise the error signal $\varepsilon = \Psi - \Phi$, where $\Psi = \hat{\boldsymbol{M_x}}^{-1}\Psi_e$. Thus, the neural network with backpropagation learning rule will update the synaptic weights in order to generate a vector Φ that follows the vector of robot model uncertainties Ψ, minimising the error signal ε.

4.2 *Neural Network Compensation Design*

The adopted architecture of the neural network shown in Figure 4 has two hidden layers, where the first one is a sigmoid layer and the second is a linear layer. The inputs to the NN are the target positions over three consecutive sampling times $\boldsymbol{X} = [x_t(t)^T \; x_t(t-1)^T \; x_t(t-2)^T]^T$, where x_t is obtained by double integration of $\ddot{x}_t$. The backpropagation update rule with a learning rate η and momentum α will give the delta vectors ΔW_i and ΔB_i in order to update of the synaptic weights for the first (W_1, B_1) and second layers (W_2, B_2), as shown in Figure 4, accordingly:

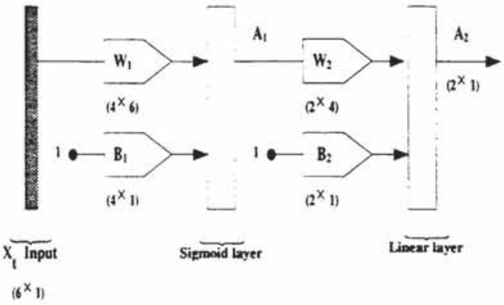

Fig. 4. Neural network structure.

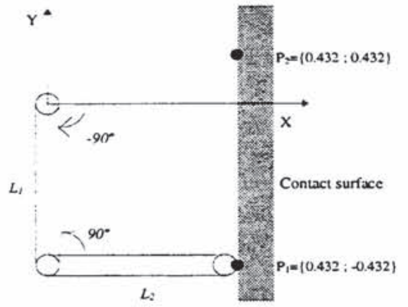

Fig. 5. Two link manipulator in vertical plane and environment.

$$\Delta W_1 = \alpha\Delta W_1 + (1-\alpha)\eta D_1 \boldsymbol{X}^T \quad (22)$$

$$\Delta B_1 = \alpha\Delta B_1 + (1-\alpha)\eta D_1 \quad (23)$$

$$\Delta W_2 = \alpha\Delta W_2 + (1-\alpha)\eta\varepsilon A_1^T \quad (24)$$

$$\Delta B_2 = \alpha\Delta B_2 + (1-\alpha)\eta\varepsilon \quad (25)$$

where

$$D_1 = (A_1 \odot (1 - A_1)) \odot (W_2^T\varepsilon) \quad (26)$$

is the derivative of the error for a layer of log-sigmoid neurons, where A_1 is the output of the log-sigmoid layer, and the operator $\odot$ represents the elementwise multiplication of two vectors.

5. SIMULATION RESULTS

In this section, the overall control scheme presented in Section 4 is applied to an industrial robot through computer simulation. The robot chosen for this simulation study is the two d.o.f. PUMA 560 planar robot in vertical plane (Figure 5). The simulation environment incorporates a detailed model of the robot, including the nonlinear arm dynamics and joint friction, which provides the basis for a realistic evaluation of the controller performance. For the particular robot under study, the numerical values of the link parameters are chosen as m_1 = 15.91 Kg, m_2 = 11.36 Kg, $l_1 = l_2$ = 0.432 m, so that they represent links 2 and 3 of the Unimation PUMA 560 arm.
In all the simulations, a constant time step of 1 ms was used in the controller implementa-

tion, while the dynamic model of the robot is simulated in the MATLAB/SIMULINK environment (The Mathworks 1992) using Runge-Kutta fourth order integration method.

In this study, a frictionless contact surface modelled as in (8, 9), with $\boldsymbol{K}_{e1} = \boldsymbol{K}_{e2} = 10000$ N/m and $\boldsymbol{B}_e = 100$ Ns/m is placed in the robot workspace, as shown in Figure 5. This surface is located at $x_e = 0.432$ m perpendicular to the horizontal plane. It is considered in all the simulations that the end-effector is already in contact with the surface and always maintain contact with the surface during the execution of the task (Figure 5). The control schemes were tested for a end-effector reference position trajectory, from $p_1 = [0.432\ \ -0.432]$ to $p_2 = [0.432\ \ 0.432]$ with a cycloidal reference force with maximum value $\boldsymbol{f}_d = 10$ N in 3 seconds, as shown in Figure 7. The impedance parameters in (12) are defined as $\boldsymbol{M}_d{=}diag[2.5\ \ 2.5]$, $\boldsymbol{K}_d{=}diag[250\ \ 250]$ and the elements of $\boldsymbol{B}_d$ are set to provide critical damping in the impedance model, that is $\boldsymbol{B}_d = diag[2\sqrt{m_{11}^2 k_{11}^2}\ \ 2\sqrt{m_{22}^2 k_{22}^2}]$. The backpropagation neural network parameters are chosen as $\alpha = 0.8$ and $\eta = 0.08$. The weights are initially selected by an off-line training procedure and deviations of 20% are applied in all the terms of the inverse dynamics controller.
In Figures 6-8, are shown the results for the hybrid impedance controller without NN compensation. Those results reveal significant force and position errors due to the presence of uncertainties in the inverse dynamics controller. In this case, the uncertainties are not cancelled, as shown in Figure 6. In Figures 9-11 are shown the results for the proposed hybrid impedance control scheme with NN compensator which reveal an accurate force and position tracking capability. In Figure 9 is shown that training signal ε converge rapidly to zero due to a fast learning behaviour of the compensator. This action minimises the modelling errors to small values in about 0.2 seconds and limit the oscillations in the joint torques to a short period of time at the beginning of the trajectory. In Figure 11 the y coordinate error and the force error are presented, for the hybrid impedance with NN compensation case.

In Table 1 the sum squared errors (SSE) of position, velocity of y coordinate and force f_e are presented.

Sum Squared Errors		
SSE	Without NN comp.	With NN comp.
SSE(y)	1.184	3.37×10^{-6}
SSE($\dot{y}$)	2.115	0.002
SSE(f_e)	1.44×10^4	58.519

Table 1. Sum squared errors of y coordinate and contact force f_e.

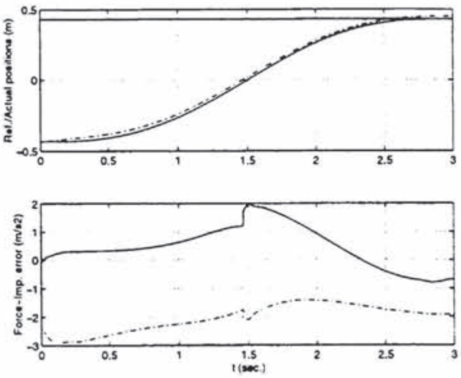

Fig. 6. Reference x_d and actual positions x. (Note: solid-actual; dashdot-reference). Force-impedance error signal ε without NN compensation. (Note: solid - x coord. ; dashdot - y coord.)

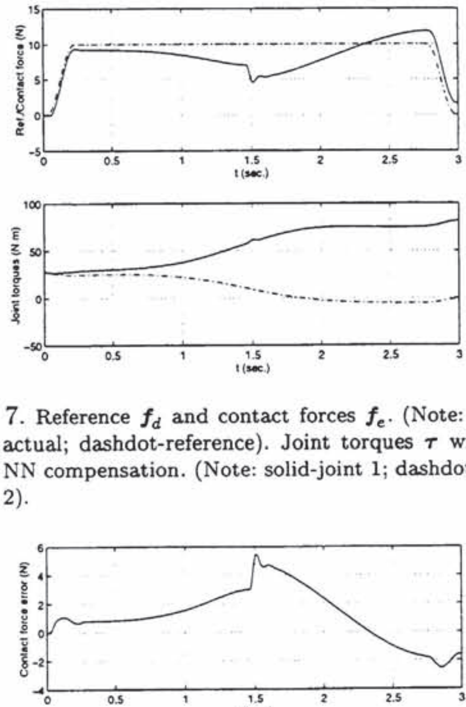

Fig. 7. Reference $\boldsymbol{f}_d$ and contact forces $\boldsymbol{f}_e$. (Note: solid-actual; dashdot-reference). Joint torques $\boldsymbol{\tau}$ without NN compensation. (Note: solid-joint 1; dashdot-joint 2).

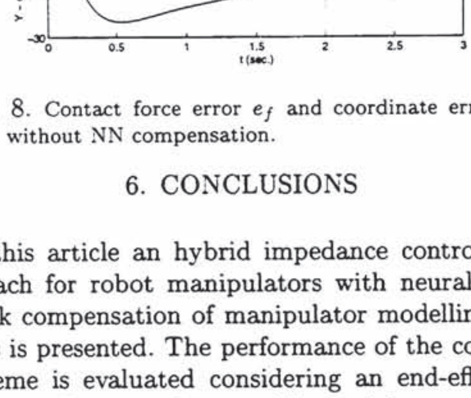

Fig. 8. Contact force error e_f and coordinate error e_y without NN compensation.

6. CONCLUSIONS

In this article an hybrid impedance control approach for robot manipulators with neural network compensation of manipulator modelling errors is presented. The performance of the control scheme is evaluated considering an end-effector position/force tracking problem with uncertainties in the inverse dynamics controller. The results obtained reveal a good force/position tracking performance in robotic tasks that require explicit

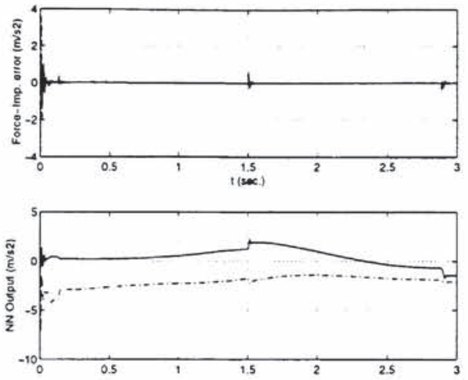

Fig. 9. Force-impedance error signal ε and NN output Φ with NN compensation. (Note: solid - x coord. ; dashdot - y coord.)

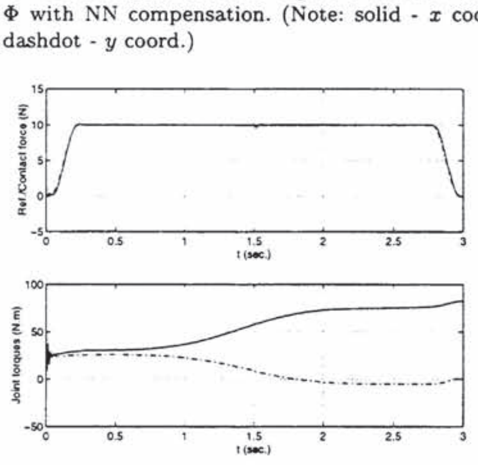

Fig. 10. Reference $\boldsymbol{f}_d$ and contact forces $\boldsymbol{f}_e$. (Note: solid-actual; dashdot-reference). Joint torques $\boldsymbol{\tau}$ with NN compensation. (Note: solid-joint 1; dashdot-joint 2).

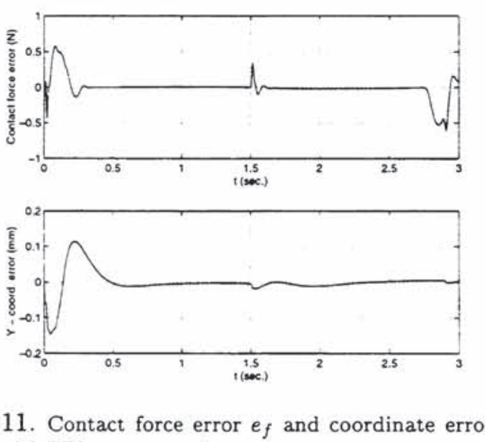

Fig. 11. Contact force error e_f and coordinate error e_y with NN compensation.

force tracking capability.

Future research will cover the improvement of the NN learning algorithm performance, a more accurate estimation of the cartesian accelerations used in the controller and the stability analysis in the context of the hybrid impedance control approach.

7. REFERENCES

Anderson, R. J. and M. W. Spong (1988). Hybrid impedance control of robotic manipulators. *IEEE Transactions on Robotics and Automation* **4**, 549–556.

Bickel, R. J. and M. Tomizuka (1995). Disturbance observer based hybrid impedance control of robotic manipulators. *American Control Conference.*

Hogan, Neville (1985). Impedance control: an approach to manipulation: Part I-II-III. *Journal of Dynamic Systems, Measurement, and Control* **107**, 1–24.

Jung, Seul and T. C. Hsia (1995*a*). Neural network techniques for robust force control of robots manipulators. *Proc. of the IEEE Int. Symposium on Intelligent Control* **1**, 111–116.

Jung, Seul and T. C. Hsia (1995*b*). On neural network application to robust impedance control of robots manipulators. *Proc. IEEE International Conference on Robotics and Automation* **1**, 869–874.

Kim, Young H. and Frank L. Lewis (1996). Output feedback control of rigid robots using dynamic neural networks. *Proc. IEEE International Conference on Robotics and Automation* pp. 1293–1928.

Mills, J. K. and David M. Lockhorst (1993). Control of robotic manipulators during general task execution: a discontinuous approach. *Proc. IEEE International Conference on Robotics and Automation* **12**, 146–163.

Raibert, M. and J. Craig (1981). Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control* **102**, 126–133.

The Mathworks, Inc. (1992). *Matlab User's Guide.* The Mathworks, Inc. 24 Prime Park Way, Natick, MA 01760.

Tso, S. K. and N. L. Lin (1996). Neural-network-based adaptive controller for uncertainty compensation of robot manipulators. *Proc. of IFAC, 13th Trienal World Congress* **K**, 61–66.

Whitney, D. E. (1987). Historical prespective and state of the art in robot force control. *International Journal of Robotics Research* **6**, 3–14.

Yoshikawa, T. (1987). Dynamic hybrid position/force control of robot manipulators-description of hand constraints and calculation of joint driving force. *IEEE Transactions on Robotics and Automation* **RA-3**, 386–392.