A NEURAL NETWORK-BASED IMPEDANCE CONTROLLER FOR A REDUNDANTLY ACTUATED CLOSED-CHAIN ROBOT MANIPULATOR [1]

A. Mesbah-Nejad, M. Moallem

Department of Electrical and Computer Engineering
The University of Western Ontario
London, Ontario, Canada, N6A 5B9

Abstract: In this paper, a neural network (NN) impedance controller is proposed to control position/force of the tip of a 3 DOF redundantly actuated closed-chain manipulator. The manipulator is in contact with an unknown environment. The structure of the controller is derived using a filtered error approach in which no off-line learning phase is needed. The actuator redundancy is resoled by augmentation of the Jacobian matrix of the manipulator. Simulation results are presented that illustrate strength of the proposed controller in the presence of model uncertainties and external disturbances.

Keywords: Redundant actuation, closed-chain manipulator, neural networks, impedance control.

1. INTRODUCTION

There is increasing interest in the applications of robots when they are interacting with an external environment. For example in robot assisted surgery, robots are expected to do surgical operations but with much higher accuracy than human hand. In such applications, robots are usually in contact with unknown environments. The motivation for this work comes from the fact that the robot controllers for these applications must not only be able to accurately control position/force exerted to the environment, they must also be able to interact with a desired compliance with respect to the environment.

A large class of controllers developed for constraint manipulators are based on the full knowledge of external environment. In the impedance control scheme, one of the position or force, and their ratio which is called compliance, or *impedance*, are controlled in each direction (Anderson and Spong, 1988). Impedance control is based on the concept that the controller is used to regulate the dynamic behavior between the robot manipulator motion and the force exerted on the environment (Hogan, 1987), rather than considering the motion and force control problems separately. In hybrid impedance control, the task space is decomposed into two subspaces of position and force. Then in the position subspace, instead of pure position control, an impedance controller is used. Impedance control works well when all the dynamics related terms in the robot and the environment are known. In practice, many of these parameters are unknown, or change over time. This becomes more important when the manipulator comes in contact with an unknown environment. In such a case, those methods cannot work very well since the controller does not know the dynamics of system to be controlled.

A great deal of research in neuro-control robotics has focused on systems with known parameters for the robot and environment. In (Kiguchi *et al.*, 2002), a fuzzy-neuro force controller of a robot manipula-

[1] This research was supported in part by grants from the Institute of Robotics and Intelligent Systems (IRIS), the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Canada Foundation for Innovation (CFI) under the New Opportunities Program.

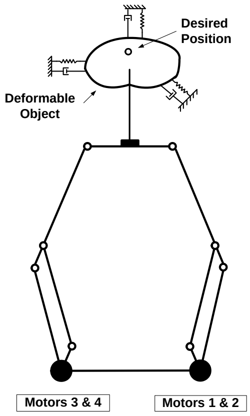

Fig. 1. An example of the application of the proposed controller

tor was presented in an unknown environment using off line learning. A neural network force tracking impedance control scheme was proposed in (Jung and Hsia, 2000). In this paper we present a NN impedance controller for a redundantly actuated closed-chain manipulator when it is in contact with an unknown environment which is different from previous works reported in the literature.

The structure of this paper is as follow: In section two the mathematical analysis of the problem with one control scheme is given to justify the use of proposed controller. In section three, the design of the controller is described. The proposed NN position tracking scheme is described in section 4. In section 5, two different control schemes are studied by simulations using a NN impedance position controller and a NN impedance position/force controller.

2. PROBLEM FORMULATION

One objective of this paper is to investigate position and force control strategies when the manipulator is in contact with a deformable environment, i.e., in needle insertion or drilling tasks. In these applications the end-effector is required to follow a prescribed position trajectory while maintaining appropriate contact forces or a specific compliance with the environment. Figure 1 shows a closed-chain redundantly actuated robot which was used for implementing the control algorithm. The details of the robot design and specifications can be found in (Mesbah-Nejad *et al.*, 2004). In order to formulate the problem, let us assume that the dynamics of the environment is given by

$$F = M_e\ddot{\delta} + B_e\dot{\delta} + K_e\delta \tag{1}$$

where

$$F = [f_x \;\; f_y \;\; f_\psi]^T \tag{2}$$

is the planar force-moment vector exerted on the environment by the end-effector. The environment is described by the terms M_e, B_e and K_e, which are mass, damping, and stiffness matrices. The term δ is the vector of generalized coordinates of the environment. The kinematic equations of the manipulator and environment are given by

$$X = h(q) \tag{3}$$

$$Y = i(\delta) \tag{4}$$

where X, Y represent the coordinates of the end-effector and the object's contact point, respectively. The functions $h\colon \Re^l \longrightarrow \Re^m$ and $i\colon \Re^m \longrightarrow \Re^n$ represent mapping from object and robot generalized coordinates to X and Y, respectively. l is the dimension of the joint space, m is the dimension of generalized coordinates of environment, and n is the dimension of the task space with $m \geq n$. It is assumed that both of h and i are twice continuously differentiable. When the manipulator is in contact with the environment, the dynamic equations of motion can be described in the following form

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_d = \tau + \tau_f \tag{5}$$

where $M(q)$ is the mass matrix, $N(q,\dot{q})$ is a vector contains all of nonlinearities in the robot dynamics (Coriolis, centrifugal, friction and gravity terms) and

$$\tau_f = J_h^T(q)F \tag{6}$$

is the vector of torques exerted on the environment and $J_h = \frac{\partial h}{\partial q}$ is the Jacobian matrix of the manipulator. When the robot is in contact with the environment, it follows from (3) and (4) that

$$h(q) = i(\delta)\cdot \tag{7}$$

For clarity and simplicity of notation, the arguments of the functions will be dropped subsequently. It is also assumed that the disturbance term τ_d in (5) is equal to zero. Taking the first and the second time derivatives of (7) gives

$$J_h\dot{\theta} = J_i\dot{\delta} \tag{8}$$

$$J_h\ddot{\theta} + \dot{J}_h\dot{\theta} = J_i\ddot{\delta} + \dot{J}_i\dot{\delta} \tag{9}$$

where $J_i = \frac{\partial i}{\partial \delta}$. Now solving (1) and (5) for $\ddot{\delta}$ and $\ddot{q}$ yields

$$\ddot{\delta} = M_e^{-1}(F - B_e\dot{\delta} - K_e\delta) \tag{10}$$

$$\ddot{q} = M^{-1}(\tau + J_h^T F - N)\cdot \tag{11}$$

Substituting (10) and (11) in (9) and solving it for F yields

$$\begin{aligned} F =& (J_iM_e^{-1} - J_hM^{-1}J_h^T)^{-1}\big[J_hM^{-1}(\tau - N) \\ &+ J_iM_e^{-1}(B_e\dot{\delta} + K_e\delta) + \dot{J}_h\dot{q} - \dot{J}_\delta\dot{\delta}\big]\cdot \end{aligned} \tag{12}$$

Now, Let us define

$$\tau = N + MJ_h^{-1}(v - \dot{J}_h\dot{q}) \tag{13}$$

where v is a pseudo control input. Then, substituting τ from (13) in (5), and using (3) results in

$$\ddot{X} = v - J_hM^{-1}J_h^T F\cdot \tag{14}$$

Also, substituting τ from (13) in (12), yields

$$\begin{aligned} F =& (J_iM_e^{-1} - J_hM^{-1}J_h^T)^{-1}\big[v \\ &+ J_iM_e^{-1}(B_e\dot{\delta} + K_e\delta) - \dot{J}_\delta\dot{\delta}\big]\cdot \end{aligned} \tag{15}$$

Let us define the control input v as

$$v = J_h M^{-1} J_h^T (F_d - v_n) - K_v(\dot{X}_d - \dot{X}) - K_p(X_d - X) \tag{16}$$

where X_d and F_d are the desired position and force vectors, respectively, and K_v and K_p are constant matrices representing the derivative and the proportional gains of the controller. Substituting v from (16) into (15) and (14) and rearranging the terms, after some manipulations we have

$$F_d - F = v_n - \Delta \tag{17}$$

and

$$\ddot{X} + K_v \dot{e} + K_p e = J_h M^{-1} J_h^T (F_d - F - v_n) \cdot \tag{18}$$

where

$$\Omega = (J_i M_e^{-1} - J_h M^{-1} J_h^T)^{-1}, \tag{19}$$

$$\begin{aligned} \Delta = & \left[\Omega J_h M^{-1} J_h^T - I\right](F_d - v_n) \\ & - \Omega \left[K_v(\dot{X}_d - \dot{X}) + K_p(X_d - X)\right] \\ & + \Omega \left[J_i M_e^{-1}(B_e \dot{\delta} + K_e \delta) - \dot{J}_\delta \dot{\delta}\right], \end{aligned} \tag{20}$$

$\dot{e} = \dot{X}_d - \dot{X}$, and $e = X_d - X$. Now, substituting (17) in (18) yields

$$\ddot{X} + K_v \dot{e} + K_p e = -J_h M^{-1} J_h^T v_n \cdot \tag{21}$$

Equation (17) states that by generating an appropriate control signal v_n, which approaches Δ, one may force the manipulator to track a desired force. As it can be seen from (20), in order to build the control signal to follow Δ, exact knowledge of the environment and manipulator dynamics are needed. On the other hand, (21) states that the position tracking error can ultimately approach zero if the v_n approaches zero. This means that achieving zero error while controlling position and force together is impossible. One can see from (20) and (21) that both force or position tracking errors can be reduced by selecting appropriate values for K_v and K_p. Therefore, depending on the application, one may choose to give priority to either force or position control and so be able to achieve acceptable position or force tracking errors by an appropriate selection of proportional and derivative gains. Nevertheless, one can see from (13) and (16) that the proposed control signal needs the full knowledge of robot and environment dynamics which are not always available. The above analysis leads us to use a NN control approach that does not require knowledge of robot and environment parameters.

3. NN IMPEDANCE CONTROLLER DESIGN

The target impedance is a desired dynamic behavior between the motion of the manipulator and the force exerted on the environment and may be defined as

$$M_d(\ddot{x_c} - \ddot{x}) + B_d(\dot{x_c} - \dot{x}) + K_d(x_c - x) = F_e \tag{22}$$

where x_c is the vector of commanded trajectories specified by the user which are bounded and twice differentiable. x is the vector of actual end-effector position trajectory, and M_d, B_d, K_d, are interpreted as the desired mass, damping, and stiffness matrices of a mass-spring damper system that quantifies the mechanical impedance relationship between the end-effector contact force F_e and the position error $(x_c - x)$. From the impedance model (22), one can see that impedance control objective can be realized if the end-effector position x and the commanded trajectory x_c satisfy this model. Therefore, if x closely tracks the desired trajectory x_d, the commanded trajectory can be found by solving (22) for x_c which yields

$$M_d \ddot{x_c} + B_d \dot{x_c} + K_d x_c = F_e + M_d \ddot{x_d} + B_d \dot{x_d} + K_d x_d \tag{23}$$

with $x_c(0) = x_d(0)$ and $\dot{x_c}(0) = \dot{x_d}(0)$. This equation may be interpreted as a filter with input x_d. The matrices M_d, B_d, K_d, and the online measurement of F_e, characterize the desired dynamic relationship between the end-effector position and the contact force through the specification of x_c. Therefore, impedance control can be done first by utilizing the contact force to modify the commanded trajectory according to the desired impedance dynamics and then by tracking x_c, which can be implemented by a position tracking controller. This approach can be used when precise control of position is required, for example in tasks such as drilling, sawing or needle insertion. If force needs to be controlled instead of position, (23) should be changed to

$$M_d \ddot{x_c} + B_d \dot{x_c} + K_d x_c = F_d + M_d \ddot{x_e} + B_d \dot{x_e} + K_d x_e \tag{24}$$

where F_d is the desired force and x_e is the actual environment position which can be measured online. Again x_c can be found by numerical integration of (24) with initial conditions $x_c(0) = x_e(0)$ and $\dot{x_c}(0) = \dot{x_e}(0)$. Now forcing a position controller to track x_c results in a desired contact force of F_d. A known problem of this control scheme is the lack of force tracking capability under an unknown environment (Jung and Hsia, 2000),(Lee and Lee, 1991). One way to improve the force tracking capability of the controller is by using the generalized impedance formulation proposed in (Lee and Lee, 1991). The problems are, using derivative of force and environment position which may add noise to the measured signals and variation of the target impedance from the impedance which was originally anticipated (24).

In this paper, in order to control the force exerted on the environment, the desired force is divided by the online measurement of environment stiffness to generate the desired position. The desired position is then supplied to the impedance filter (24) to generate the commanded position trajectory.

Implementation of the impedance control algorithm boils down to the implementation of a position tracking controller. In this section, the idea of the NN position tracking controller, which was proposed in (Lewis *et al.*, 1996) is used. However, modifications are made to the algorithm in the context of an impedance control algorithm. Refereing to (5, the inertia matrix $M(q)$ is symmetric, positive definite, and bounded. Also the Coriolis/centripetal vector $V_m(q,\dot{q})\dot{q}$ is quadratic in $\dot{q}$. V_m is bounded so $\parallel V_m \parallel \leq v_B \parallel q \parallel$, where v_B is upper bound of V_m. To obtain task space formulation, consider the relationship

$$x = h(q) \tag{25}$$

Taking its first and second time derivative and solving for $\dot{q}$ and $\ddot{q}$ we have

$$\dot{q} = J^{\#}(q)\dot{x} \qquad \ddot{q} = J^{\#}(q)\left(\ddot{x} - J(\dot{q})\dot{q}\right) + (I - J^{\#}J)\zeta \tag{26}$$

where $J = \frac{\partial h}{\partial q}$ is the task space Jacobian matrix and $J^{\#} = (J^T J)^{-1} J^T$ is the pseudo-inverse of the Jacobian matrix. ζ can be any vector. For brevity of notation, the arguments of the functions will be dropped in subsequent analysis. Substituting (26) in (5 yields the task space dynamics

$$\bar{M}\ddot{x} + \bar{V}\dot{x} + \bar{F}_r + \bar{G} + f_d = F \tag{27}$$

where x is the task space variable vector and

$$\begin{aligned} \bar{M} &= J^{\dagger} M J^{\#} & \bar{V} &= J^{\dagger}(V_m J^{\#} - M\dot{J}J^{\#}) \\ \bar{F}_r &= J^{\dagger} F_r & f_d &= J^{\dagger}\tau_d \\ \bar{G} &= J^{\#} G & F &= J^{\#}\tau \end{aligned} \tag{28}$$

and $J^{\dagger} = J^{\#^T}$.

Given the desired position trajectory x_d, the position tracking error and the filtered position tracking error can be defined by

$$e_p = x_d - x \tag{29}$$

$$r = \dot{e_p} + \Lambda e_p \tag{30}$$

where e_p is the position error and Λ is a positive definite design parameter matrix. The system (30) is stable as long as r is bounded and Λ is large enough. Taking the first and the second time derivatives of (29) and substituting the results in (27) yields

$$\bar{M}(\ddot{x_d} - \ddot{e_p}) + \bar{V}(\dot{x_d} - \dot{e_p}) + \bar{F}_r + \bar{G} + f_d = F \cdot \tag{31}$$

Differentiating (30) with respect to time and substituting $\ddot{e_p}$ and $\dot{e_p}$ in (27) results in the robot dynamics in terms of r given by

$$\bar{M}\dot{r} = -\bar{V}r + f(y) + f_d - F \tag{32}$$

where the $f(y)$ is defined by

$$f(y) = \bar{M}(\ddot{x_d} + \Lambda\dot{e_p}) + \bar{V}(\dot{x_d} + \Lambda e_p) + \bar{F}_r + \bar{G} \tag{33}$$

and vector y is defined as

$$y = [e_p^T \quad \dot{e_p}^T \quad x_d^T \quad \dot{x_d}^T \quad \ddot{x_d}^T]^T \cdot \tag{34}$$

Note that in (32) all of the potentially unknown robot parameters are included in $f(y)$ except the term $-\bar{V}r$. This term is cancelled out in controller stability analysis. It is assumed that all the states of the system are measurable. If not, then additional NN methods should be used to estimate the unmeasured states (Tian *et al.*, 2004). The proposed control law is defined as

$$F = \hat{f}(y) + K_v r + v \tag{35}$$

where $\hat{f}(y)$ is an estimate of f, Kv is a constant and $v(t)$ a robustifying term. Assuming that $f(y)$ is smooth enough, according to the universal approximation property of neural networks, there exist a two layer feedforward NN with one hidden layer having enough neurons and weight matrices V and W such that $f(y)$ can be approximated on a compact set S, by

$$f(y) = W^T\sigma(V^T y) + \varepsilon \tag{36}$$

where ε, the functional estimation error is bounded by ε_N (NN approximation inaccuracy) for $y \in S$. If the estimated weight matrices are defined as $\hat{V}$ and $\hat{W}$, the approximated function $\hat{f}(y)$ is

$$\hat{f}(y) = \hat{W}^T\sigma(\hat{V}^T y) \tag{37}$$

and the weights errors can be defined as $\tilde{V} = V - \hat{V}$, and $\tilde{W} = W - \hat{W}$.

The weight tuning formulas are given by the following theorem following a similar analysis in (Lewis *et al.*, 1996).

Theorem 1. Assume that the dynamics of the robot is expressed by (27). Also assume that the ideal weights and the desired trajectory are bounded. It is assumed that the NN approximation property holds for the function f given in (33). Then with the control law (35) where v is the robustifying term defined by

$$v(t) = \rho\left(\left\|\hat{Z}\right\|_F + Z_M\right) r \quad \text{with} \quad \rho > \alpha_8 \tag{38}$$

where subscript F stands for Frobenius norm. In (38), Z is a block diagonal matrix with diagonal elements V and W and with the upper bound of Z_M, α_8 is a positive value given by $\alpha_8 = \left[1 + \frac{1+\varsigma_{max}(\Lambda)}{\varsigma_{min}(\Lambda)}\right] Z_M$ where ς_{max} and ς_{min} are maximum and minimum singular values of Λ (Mesbah-Nejad, 2004), and ρ is the robustifying gain. Then with the following NN weight tuning algorithms

$$\dot{\hat{W}} = \eta_2\sigma(\hat{V}^T y)r^T - \eta_2\dot{\hat{\sigma}}\hat{V}^T y r^T - k\eta_2 \|r\| \hat{W} \tag{39}$$

$$\dot{\hat{V}} = \eta_1 y(\dot{\hat{\sigma}}^T\hat{W}r)^T - k\eta_1 \|r\| \hat{V} \tag{40}$$

where the learning rates η_1 and η_2 are symmetric positive definite matrices, and k is a positive constant value, the position tracking error $\|r\|$ is uniformly ultimately bounded.

The first terms of (39) and (40) are backpropagation terms. The last terms are *Narendra's e-modification* (Narendra and A.Annaswamy, 1987) which adds a quadrature term in $\left\|\tilde{W}\right\|_F$ and $\left\|\hat{V}\right\|_F$ to make the derivative of Lyapunov's function negative. The second term in (39) is added to prove that the NN weights are bounded. Boundedness of the weights is needed

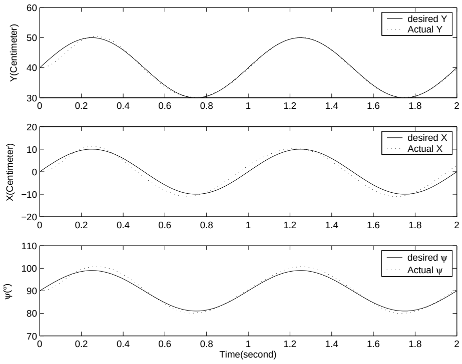

Fig. 2. Response of the PD position controller to the input trajectories (41), (42) and 43.

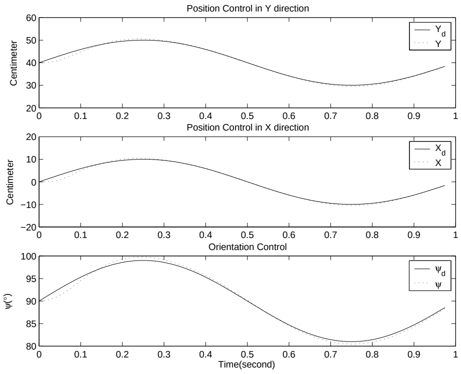

Fig. 3. Response of the MFNN Impedance position controller.

to show that the control input remains bounded. Although the error is not zero, with selection of controller gains it can be reduced to a small enough value.

At the beginning of the control process where all the weights of the NN controller are zero, the controller acts like a robust PD controller. The performance improves over time as the NN controller is trained. Another interesting feature of the proposed controller is that, when the end-effector is moving in the free space, the impedance filter is bypassed as there is no force. Thus the controller acts like a position controller. As soon as the end-effector touches the environment, the impedance filter comes into service. Therefore, there is no need to design two separate algorithms for free space movement and constrained space movement and no need to switch between them. This simplifies the design and removes the need for a switching mode controller.

4. SIMULATION RESULTS

To have a reference for comparison of the results of the proposed controller with a standard controller, a PD controller with feedback linearization method is designed. The following position trajectories were applied as inputs to the system

$$X_d = 0.1\sin(2\pi t) \tag{41}$$
$$Y_d = 0.4 + 0.1\sin(2\pi t) \tag{42}$$
$$\Psi = \frac{\pi}{2} + \frac{\pi}{20}\sin(2\pi t)\ \cdot \tag{43}$$

The outputs of the controller using feedback gains $K_p = 200,\ K_v = 10$ are shown in Figure 2. The tracking error increases with a decrease in the proportional gain. Although the value of overshoots and the final error can be adjusted by tuning PD gains, for achieving a tracking error of less than a millimeter, a relatively large proportional gain should be selected ($K_p \geq 400$). Evaluation of the controller in presence of disturbance in control signal and variation in the robot parameters indicates that the PD controller is not able to track the inputs very well. The function f in (33) which can be approximated by a NN is the sum of four parts which are related to the mass matrix, centrifugal and coriolis terms, gravity, and friction forces. The gravity part is zero as the manipulator is planar. The neural network controller can be partitioned into three individual 2-layer NNs, each implementing one term of f. This simplifies the design and reduces the computational load when some of the parameters such as mass matrix are well known. This also helps to utilize available knowledge of the system, for example, if the model of friction or its approximation, is known. The Sigmoid activation function can be used in the NN implementation, since the joints are revolute and joint variables appear in *sine* or *cosine* form, instead of using joint angles, their *sine* and *cosine* can be used. Another preprocessing uses $(\ddot{x}_d+\Lambda\dot{e}_p)$ and $(\dot{x}_d+\Lambda e_p)$ instead of e_p and $\dot{e}_p$ in (34).

A 2-layer NN structure was designed to estimate the nonlinear function (33). The signum function is used to model the friction of the joints. The environment is modelled with a mass-spring system with the following parameters

$$M_e = 0.5I_{3\times3} \quad B_e = I_{3\times3} \quad K_e = 10I_{3\times3}\cdot \tag{44}$$

For NN Impedance Position Control, the controller is designed to control positions in X and Y directions and the end-effector orientation while maintaining the required impedance. The NNs are implemented by 10 hidden layer neurons and Sigmoid activation function. The controller parameters are selected as $\Lambda = 100I_{3\times3}$, $K_v = 10I_{3\times3}$, $\rho = 1$, $Z_M = 10$. Also $\eta_1 = 50I_{10\times10}$, $\eta_2 = 10I_{10\times10}$, $k = 0.1$. The same position trajectories as PD controller (41, 42 and 43) are selected. The desired system impedance is set to

$$M_d = I_{3\times3} \quad B_d = 20I_{3\times3} \quad K_d = 500I_{3\times3}\cdot \tag{45}$$

The response of the controller appears in Figure 3. Comparing the position tracking results with those obtained in Figure 2 for the PD controller, the NN is able to achieve arbitrary small overshoot, undershoot, and tracking errors by adjusting the gains of the controller and the desired impedance. The controller neither needs to know any a-priori information about the dynamics of the robot or the environment nor does it require any initial training. It also gives good flexibility to the designer as there are many parameters in the system impedance to adjust. Two different disturbance

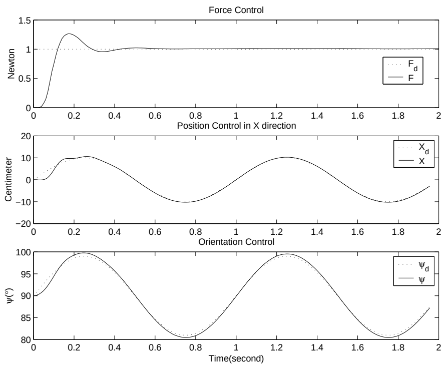

Fig. 4. Response of the MFNN Impedance position/force controller.

signals of $\tau_d = 0.2N.m$ and $\tau_d = 0.2\sin(20\pi t)N.m$ are added to control signal individually in order to evaluate robustness of the controller to external disturbance. The results indicate that the controller is robust to unknown but bounded disturbances. In order to evaluate the robustness of the controller in the presence of uncertainties in robot dynamics, the mass matrix M and the Coriolis-centrifugal vector N are reduced by a factor of 80% in the robot simulator. The response is almost identical to the one shown in Figure 3 which means that the controller is able to adapt with variations in robot parameters. The M_e, B_e and K_e terms in (44) have been changed from 0 to 0.9, 0 to 100 and 0 to 18, respectively, to evaluate the effects of different environments on the controller. The results show that as long as the desired impedance is achievable, the controller is able to control the system with significant changes in characteristics of the environment. For NN Impedance Position/Force Control, the controller is designed to control the force in Y direction, the position in X direction and the end-effector orientation while maintaining the required impedance. The NNs are implemented by 10 hidden layer neurons. The controller parameters are selected as $\Lambda = 100I_{3\times3}$, $K_v = 10I_{3\times3}$, $\rho = 1$, $Z_M = 10$. Also $\eta_1 = 50I_{10\times10}$, $\eta_2 = 10I_{10\times10}$, $k = 0.1$. It is assumed that the variation of K_e is not very large, so its derivative is set to zero. The same position tracking trajectories as the PD controller case (41 and 43) are selected for X and Ψ. The desired force is selected to be $F_Y = 1N$. The desired system impedance is given by (45). Response of the controller appears in Figure 4. Similar to the impedance/position case, the controller neither needs to know any a-priori information about the dynamics of the robot or the environment, nor does it require any initial training. The response of the controller in the presence of disturbance and uncertainties in robot dynamics indicates that force and position controllers are robust to the bounded external disturbance and it is able to adapt to the variations in the robot parameters.

5. CONCLUSION

In this paper, a NN impedance control scheme was developed to control the position/force of the end-effector of the manipulator when it is in contact with an unknown environment while maintaining a desired impedance. The controller was partitioned into three NNs working in parallel. The NNs do not need to be trained off-line. The controller was designed with a MFNN. Simulation results were presented in the presence of external disturbance and variation in model and environmental parameters and compared with a PD controller. The results show that the proposed controller can perform better than standard PD controller in terms of tracking error and robustness to noise and uncertainties in model dynamics.

REFERENCES

Anderson, R.J. and M.W. Spong (1988). Hybrid impedance control of robotic manipulators. *IEEE Journal of Robotics and Automation* **4**(5), 549–556.

Hogan, N. (1987). Stable execution of contact tasks using impedance control. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 4. pp. 1047–1054.

Jung, S. and T.C. Hsia (2000). Robust neural force control scheme under uncertainties in robot dynamics and unknown environment. *IEEE Transactions on Industrial Electronics* **47**(2), 403–412.

Kiguchi, K., K. Watanabe and T. Fukuda (2002). Generation of efficient adjustment strategies for a fuzzy-neuro force controller using genetic algorithms application to robot force control in an unknown environment. *Information Sciences* **145**(1-2), 113–126.

Lee, S. and H.S. Lee (1991). On force-tracking impedance control of robot manipulators. In: *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*. Vol. Aug. Arlington, VA USA. pp. 61–66.

Lewis, F.L., A. Yegildirek and K. Liu (1996). Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks* **7**(2), 388–399.

Mesbah-Nejad, A. (2004). Mechanical design and control of a redundant closed-chain manipulator. M.E.Sc. University of Western Ontario. London, Ontario, Canada.

Mesbah-Nejad, A., M. Madani, M. Moallem and R.V. Patel (2004). Design and optimization of a closed-chain mechanism for constrained position and force control. In: *3rd IFAC Symposium on Mechatronic Systems*. Sydney, Australia. pp. 751–756.

Narendra, K.S. and A.Annaswamy (1987). A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control* **32**(2), 134–145.

Tian, L., J. Wang and Z. Mao (2004). Constrained motion control of flexible robot manipulators based on recurrent neural networks. *IEEE Transactions on Systems, Man and Cybernetics, Part B* **34**(3), 1541–1552.