Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

Experimental Studies of Neural Network Impedance Force Control for Robot Manipulators *

Seul Jung and Sun Bin Yim
Intelligent Systems and Emotional Engineering(ISEE) Laboratory
Department of Mechatronics Engineering
Chungnam National University, Daejon, Korea 305-764
and
T. C. Hsia
The Robotics Research Laboratory
The University of California, Davis
Davis, CA 95616

Abstract

In this paper, neural network force control is presented. Under the framework of impedance control, neural network is used to compensate for all the uncertainties from robot dynamics and unknown environment. Modified simple impedance function is realized after the convergence of neural network. Learning algorithms for neural network to minimize force error directly are designed. As a test-bed, the large x-y table robot is implemented. Experimental results have shown better force tracking when neural network is used.

1 Introduction

When the robot performs force tracking for a certain task several fundamental problems have to be solved. One problem is the uncertainty from the estimation of inexact robot dynamic equation specially when the inverse dynamic control algorithm is applied, which is one of the most popular control methods. Uncertainty in robot dynamics may yield joint space position tracking errors, and they lead to poor force tracking in the Cartesian space. Another problem is caused from the unknown environment. Unknown environment position and stiffness lead to poor force tracking due to incorrect estimation of desired force.

The impedance control is one of the main force control algorithms, which deals with the dynamic relationship between the robot and the environment. However, the impedance control is known to have lack of desired force tracking capability while hybrid force control has the capability [1, 2].

To give force tracking capability to the impedance control, the adaptive control technique in conjunction with the time-delayed control method has been applied to deal with unknown environment and unknown robot dynamics as well [3].

As for intelligent approach to force control, fuzzy neural network algorithms have been used to deal with unknown objects [4]. Neural force control based on impedance control that compensates for all the uncertainties has been proposed [5].

In this paper, as an extension of our previous paper [5], a single neural network is used to compensate for uncertainties from both robot dynamics and unknown environment under the framework of impedance force control. A new training signal is proposed to minimize force error directly.

Neural network training algorithm is implemented on a DSP board in PC for faster computing to make real time control possible. Intensive experimental studies are presented for the x-y table robot to investigate the robustness under uncertainties.

Experimental results show good force tracking for different stiffnesses such as steel and wood, which have the unknown bumped shape.

2 Dynamic Equation of Robot Manipulators

The dynamic equation of an n degrees-of-freedom robot manipulator in joint space coordinates are given

*This project has been supported in part by 1998 KOSEF and 1997 KRF research funds in Korea.

by :

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) = \tau - \tau_e \quad (1)$$

where the vectors $q, \dot{q}, \ddot{q}$ are the n x 1 joint angle, the n x 1 joint angular velocity, and the n x 1 joint angular acceleration, respectively; $D(q)$ is the n x n symmetric positive definite inertia matrix; $C(q,\dot{q})\dot{q}$ is the n x 1 vector of Coriolis and centrifugal torques; $G(q)$ is the n x 1 gravitational torques; τ_f is the n x 1 vector of actuator joint friction forces; τ is the n x 1 vector of actuator joint torques; τ_e is the n x 1 vector of external disturbance joint torques.

For simplicity, let us denote $h(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + \tau_f(\dot{q})$ so that equation (1) can be rewritten as

$$D(q)\ddot{q} + h(q,\dot{q}) = \tau - \tau_e \quad (2)$$

The Jacobian relationship between the joint velocity and the Cartesian space velocity allows to have the robot dynamic equation model including disturbance force in the Cartesian space as

$$D^*\ddot{X} + h^* = F - F_e \quad (3)$$

where $D^* = J^{T^{-1}}DJ^{-1}, h^* = J^{T^{-1}}h - D^*\dot{J}J^{-1}\dot{X} + J^{T^{-1}}\tau_f, F_e$ is the external force, and F is the control force.

In order to deal with the position tracking errors due to ill-estimation of robot dynamic model, we propose to use neural network as a compensator to cancel out those uncertain terms.

3 Review of Impedance Force Control

The basic impedance control concept is to regulate the relationship between force and position by setting suitable gains of impedance parameters [1].

The control law F for impedance force control is then

$$F = \hat{D}^*V + \hat{h}^* + F_e \quad (4)$$

where $\hat{D}^*, \hat{h}^*$ are estimates of D^*, h^* and F_e is the force exerted on the environment.

The Cartesian control input V is given by

$$V = \ddot{X}_r + M^{-1}[B(\dot{X}_r - \dot{X}) + K(X_r - X) - F_e] \quad (5)$$

where M B and K are diagonal n x n symmetric positive definite matrices of desired inertia, damping, and stiffness gains, respectively, and X_r is the reference end-point trajectory determined from environment position, environment stiffness and desired contact force.

Combining (3),(4), and (5) yields the closed loop tracking error equation

$$\ddot{\varepsilon} + M^{-1}[B\dot{\varepsilon} + K\varepsilon - F_e] = \hat{D}^{*^{-1}}[\Delta D^*\ddot{X} + \Delta h^*] \quad (6)$$

where $\Delta D^* = D^* - \hat{D}^*, \Delta h = h^* - \hat{h}^*$, and $\varepsilon = X_r - X$.

In the ideal case where $\Delta D^* = \Delta h^* = 0$, and if the reference trajectory X_r is known exactly, the closed-loop robot satisfies the target impedance relationship as

$$F_e = M\ddot{\varepsilon} + B\dot{\varepsilon} + K\varepsilon \quad (7)$$

Since there are always uncertainties in the robot dynamic model the ideal target impedance relationships (7) can not be achieved in general.

Another problem is the difficulty of knowing the environment stiffness accurately in advance. In order to achieve the desired contact force the reference trajectory X_r should be designed a priori based on environment stiffness and environment position. This difficulty can be solved by the modified force tracking impedance control in the next section.

4 Modified Impedance Function

The proposed neural network control structure is shown in Figure 1. Replacing X_r with X_e in (7) and subtracting the desired force F_d from F_e yields the new impedance function as [6]

$$F_e - F_d = M\ddot{E} + B\dot{E} + KE \quad (8)$$

where $E = X_e - X$ and X_e is the environment position.

For simplicity, we consider that force is applied to only one direction. Let f_d, f_e, m, b, k be elements of F_d, F_e, M, B, K respectively. Then, equation (8) becomes

$$m\ddot{e} + b\dot{e} + ke - f_e + f_d = 0 \quad (9)$$

where $e = x_e - x$.

In free space control, the control law can be obtained from (9) for $f_e = 0$ as follows:

$$m\ddot{e} + b\dot{e} + ke = -f_d \quad (10)$$

We note that when the desired force f_d is set to zero in the equation (10) and the desired trajectory is known exactly, then the robot is under position control. In force controllable direction, the desired force f_d in (10) serves as the driving force to enable robot to make contact with the environment for sure.

In contact space, since $f_e = k_e(x - x_e) = -k_e e$, the impedance function (9) becomes

$$m\ddot{e} + b\dot{e} + (k + k_e)e = -f_d \quad (11)$$

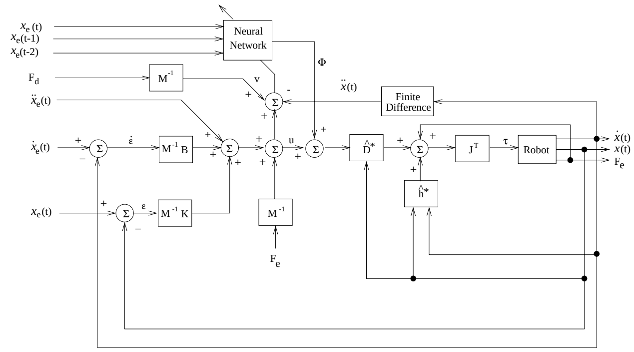

Figure 1: NN impedance force control structure

We see that (11) is asymptotically stable.

However, when we have unknown environment position the impedance function becomes

$$m\ddot{e}' + b\dot{e}' + ke' + k_e e = -f_d \tag{12}$$

where $e' - e = \delta x_e$. At the steady state of (12), $f_e \neq f_d$. Careful investigation of the impedance relation (12) suggests that setting the stiffness gain k to zero will satisfy the ideal steady state condition in $f_e = f_d$ for any k_e. So the proposed law is to set the stiffness gain $k = 0$ in the force controllable direction at the time of contact, that is (12) becomes

$$m\ddot{e}' + b\dot{e}' - f_e + f_d = 0 \tag{13}$$

Even though k_e is not known exactly in practice, the proper gains m and b based on the approximation of k_e can be chosen to achieve good transient response. Therefore, the proposed impedance function is simple, stable and robust in force tracking under unknown environment stiffness condition.

5 Neural Network Control Compensation

Since force control is done in the Cartesian space the control law should be represented by having the Jacobian relationship.

From Figure 1 the control law becomes

$$\tau = \hat{D}\ddot{q}(t) + \hat{h} + \tau_e \tag{14}$$

and

$$\ddot{U}(t) \approx \ddot{q}(t) = J^{-1}(V - \dot{J}\dot{q}) \tag{15}$$

where $V = \ddot{X}$.

From the impedance law (10) and (13), $\ddot{X}$ can be obtained as

$$V_p = \ddot{X}_d + M^{-1}(B\dot{\varepsilon} + K\varepsilon) + \Phi_p$$

$$V_f = \begin{cases} \ddot{X}_e' + M^{-1}(F_d + B\dot{E}' + KE') & \text{free} \\ \ddot{X}_e' + M^{-1}(F_d - F_e + B\dot{E}') + \Phi_f) & \text{contact} \end{cases} \tag{16}$$

where $\Phi_p\Phi_f$ is the outputs of neural network and $V = [V_p^T V_f^T]^T$ where V_p is for position controlled direction and V_f is for force controlled direction.

Combining (14) and (15) yields

$$\tau = \hat{D}J^{-1}(V - \dot{J}\dot{q}) + \hat{h} + \tau_e \tag{17}$$

Substituting the control law (17) into robot dynamic equation (2) yields

$$\hat{D}J^{-1}(V - \ddot{X}) = \Delta D\ddot{q} + \Delta h \tag{18}$$

where $\Delta D = D - \hat{D}, \Delta h = h - \hat{h}$. Arranging (18) gives error equation as

$$(V - \ddot{X}) = J\hat{D}^{-1}(\Delta D\ddot{q} + \Delta h) \tag{19}$$

Consider force controllable direction separately. Substituting (16) for contact into (19) yields

$$\ddot{X}_e' + M^{-1}(F_d - F_e + B\dot{E}') + \Phi_f - \ddot{X} = J\hat{D}^{-1}\Delta\tau \tag{20}$$

where the uncertainty terms $\Delta\tau = \Delta D\ddot{q} + \Delta h$.

Then (20) is simplified as

$$M\ddot{E}' + B\dot{E}' + F_d - F_e + M\Phi_f = MJ\hat{D}^{-1}\Delta\tau \tag{21}$$

Therefore neural network compensating signal Φ_f can compensate for those dynamic uncertainties $\Delta\tau$ as

$$M\ddot{E}' + B\dot{E}' + F_d - F_e = MJ\hat{D}^{-1}\Delta\tau - M\Phi_f \tag{22}$$

After the convergence, neural network output becomes

$$\Phi_f = J\hat{D}^{-1}\Delta\tau \tag{23}$$

Then the desired impedance function can be achieved as

$$M\ddot{E}' + B\dot{E}' + K_e E = -F_d \tag{24}$$

where $F_e = -K_e E$.

In the same manner we can obtain the following equation for position controllable direction.

$$M\ddot{\varepsilon} + B\dot{\varepsilon} + K\varepsilon = MJ\hat{D}^{-1}\Delta\tau - M\Phi_p \tag{25}$$

where Φ_p is a neural network output for position controlled direction.

6 Neural Compensator Design

When the environment is assumed to be flat such as $\ddot{X}_e = \dot{X}_e = 0$ (22) becomes

$$F_d - F_e = M\ddot{X} + B\dot{X} + MJ\hat{D}^{-1}\Delta\tau - M\Phi_f \tag{26}$$

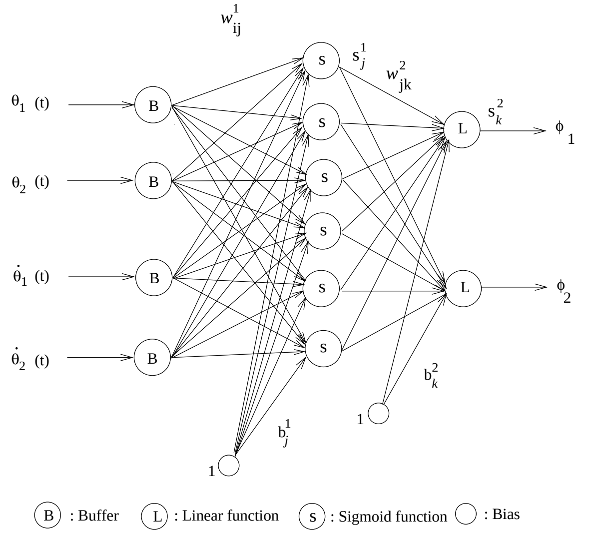

Figure 2: Neural Network Structure

If the environment position profile is not flat then good force tracking can not be achieved due to $\dot{X}, \ddot{X}$ terms. To overcome this problem we use neural network to cancel out those terms in (26).

In order to do that, the training signal for neural network should be designed properly so that the force error is minimized. The training signals for v_{force} are obtained from equations (16) as

$$v_{free} = \ddot{e} + \frac{1}{m}(b\dot{e} + ke + f_d) \tag{27}$$

$$v_{contact} = f_d - f_e \tag{28}$$

The objective function to be minimized is

$$\mathcal{J} = \frac{1}{2} v^T v \tag{29}$$

where $v = [v_{position}^T v_{force}^T]^T$.

Making use of the definition of v in equation (25) and (26) yields the gradient of $\mathcal{J}$ as

$$\frac{\partial J}{\partial w} = \frac{\partial v^T}{\partial w} v = -M \frac{\partial \Phi^T}{\partial w} v \tag{30}$$

in which the fact $\frac{\partial v^T}{\partial w} = -M \frac{\partial \Phi^T}{\partial w}$ is used. The neural network output is $\Phi = [\Phi_p^T \Phi_f^T]^T$.

The back-propagation update rule for the weights with a momentum term is

$$\Delta w(t) = \eta M \frac{\partial \Phi^T}{\partial w} v + \alpha \Delta w(t-1) \tag{31}$$

where η is the update rate and α is the momentum coefficient.

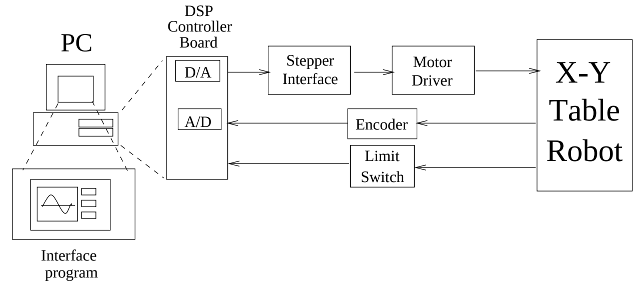

Figure 3: The overall system structure

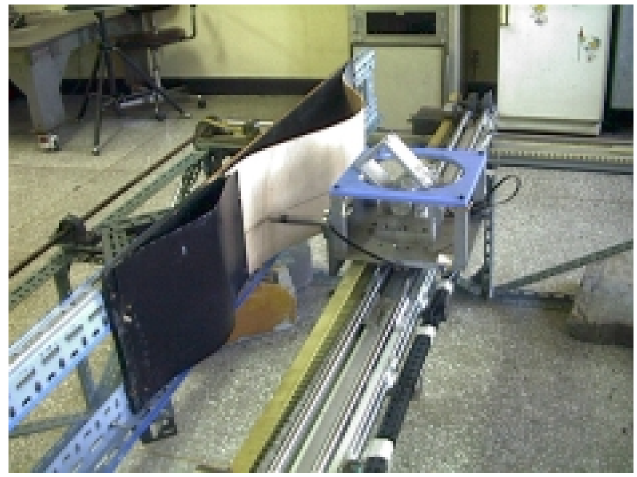

Figure 4: Force tracking experiment environment

Two layer feed-forward neural network is used as a compensator as shown in Figure 2. The output of hidden units are filtered by a nonlinear function

$$f(\cdot) = \frac{1 - exp(-(\cdot))}{1 + exp(-(\cdot))} \tag{32}$$

while the output units are linear. We have chosen six hidden neurons ($n_H = 6$) for our experiments. The back propagation algorithm parameters are: η is optimized and $\alpha = 0.9$. Weights are randomly selected.

7 Experimental Setup

The experimental setup for neural network impedance force control system is shown in Figure 3. The system consists of large x-y table robot, DSP controller in PC, and motor drivers. The ball bearing mounted end-effector is designed to reduce friction force as small as possible. The frictionless end-effector mounted with JR^3 force sensor resides on a cart in x direction of the large x-y table as shown in Figure 4. The end-effector can move on the x-y plane.

Since the robot is huge, axis movement of the large x-y table is actuated by 1:10 reduction geared type DC motors through timing belts in order to generate larger torques. Uncertainties are present due to design mechanism such as back lash caused from loose tension of the timing belt and unbalanced actuation by one side motor. The DC motors are driven by their own drivers and they are commanded from DSP board mounted in PC. The user can command the robot to move through the interface program written in C++. The DSP board is used to implement neural network algorithm in real time, which requires huge calculation time. We can achieve real time force control experiments approximately at 166Hz sampling frequency.

8 Experimental Results

8.1 Flexible Bumped Wood Environment

The sample force tracking performance for different environment surface location is shown in Figure 4. The robot is commanded to move y direction while pushing the wooden environment in x direction. The controller gains are selected as $K_D = diag[4,5], K_P = diag[0,85]$. The shape of thin wood is not flat but sinusoidal as shown in Figure 4. The environment is so flexible that the position is time-varying. The desired position for force controlled direction is not given in the controller while for position controlled direction is given in the controller. First of all, the performance of the modified impedance force controller without neural network was tested. The desired force is set to 5 N. The force tracking result is shown in Figure 5, which has large force tracking errors about 5 N. The large offset errors occur due to time-varying unknown environment profile. The corresponding position tracking is shown in Figure 6, which is a bumped shape.

Next, we try to test that neural network can compensate for those uncertainties from unknown environment position and stiffness. For the neural network parameters, the optimized learning rates $\eta = 0.000023$ is used.

The force tracking response using neural network is shown in Figure 7. The neural controller enables the system to be stable by compensating for those uncertainties. Since the environment is flexible, when the end-effector moves right after the peak point, slipping leads to no contact point. But it immediately regulates to maintain contact.

8.2 Unknown Steel Environment

We also tested for hard stiffness environment such as a bumped steel as shown in Figure 4. The controller gains are selected as $K_D = diag[140,5]$ and $K_P = diag[0,85]$ which give more over-damped motions at the force controlled direction than the previous case. The learning rate $\eta = 0.00001$ is used. We also tried the experiment for desired force as 5 N, but it is very difficult to make it stable due to the nature of large stiffness. So we increase the desired force for contact to 40 N. It is more interesting to see the performance of the proposed controller when the environment surface location is also time-varying as shown in Figure 8. Force tracking performance is shown in Figure 9. When environment surface location is totally unknown, the force tracking offset error occurred due to the unknown surface velocity profile of environment location as we assume $\dot{x}'_e = \ddot{x}'_e = 0$. Those uncertainties are successfully compensated by neural network. The simulation data showed that the force tracking result is very effective.

9 Conclusions

Experimental studies of neural network control technique for the modified simple impedance control of robot are presented. As for environment, we have tested bumped wood and steel whose environment position and stiffness are totally unknown to the controller. The proposed neural network controller has shown better force tracking under unknown environment stiffness and location by compensating for the uncertainties than that for without neural network.

When the environment has large stiffness, the regulating force should be large enough to make contact stable.

References

[1] N. Hogan, "Impedance control : An approach to manipulator, part i, ii, iii," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–24, March, 1985.

[2] M. H. Raibert and J. J. Craig, "Hybrid position and force control of robot manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 126–133, June, 1981.

[3] S. Jung and T. C. Hsia, "Adaptive force tracking impedance control of robot for cutting nonhomogeneous workpiece," *IEEE Conf. on Robotics and Automations*, pp. 1800–1804, May 1999.

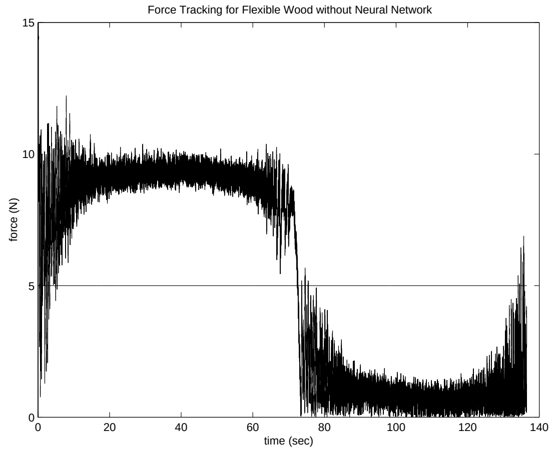

Figure 5: Force Tracking for Time-varying Flexible Wood Environment without NN

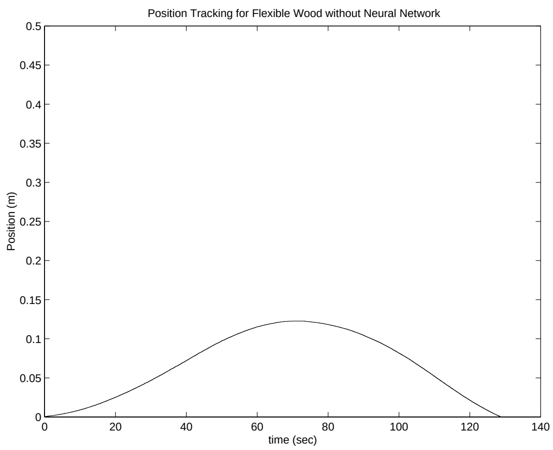

Figure 6: Position Tracking for Time-varying Flexible Wood Environment without NN

[4] Kazuo Kiguchi and Toshio Fukuda, "Position/force control of robot manipulators for geometrically unknown objects using fuzzy neural networks," *IEEE Trans. on Industrial Electronics*, vol. 47, no. 3, pp. 641–649, 2000.

[5] S. Jung and T. C. Hsia, "Robust neural force control scheme under uncertainties in robot dynamics and unknown environment," *IEEE Trans. on Industrial Electronics*, vol. 47, no. 2, pp. 403–412, April 2000.

[6] Sukhan Lee and Hahk Sung Lee, "Intelligent control of manipulators interfacing with an uncertain environment based on generalized impedance," *Proc. of IEEE Symposium on Intelligent Control*, pp. 61–66, 1991.

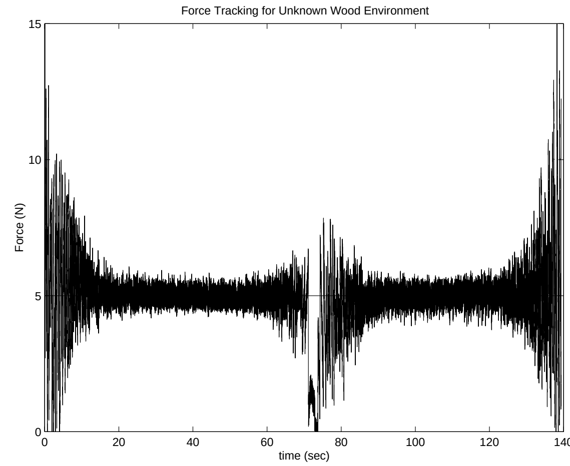

Figure 7: Force Tracking for Time-varying Flexible Wood Environment

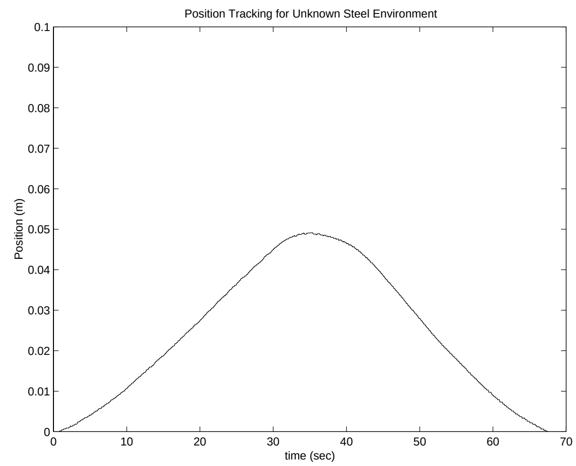

Figure 8: Position Tracking for Time-varying Unknown Steel Environment

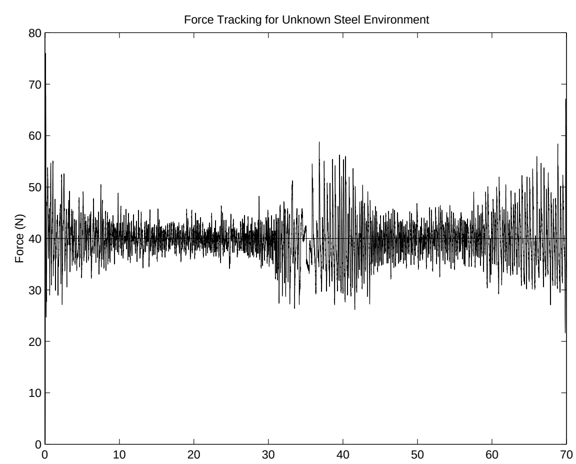

Figure 9: Force Tracking for Time-varying Unknown Steel Environment