Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

The Operational Space Formulation with Neural-Network Adaptive Motion Control

Dandy B. Soewandito *, **Denny Oetomo** **,
Marcelo H. Ang, Jr. *

* *Mechanical Engineering Department National University of Singapore 9 Engineering Drive 1, Singapore 117576, Singapore. Email: altec_1@yahoo.com (corresp. author), mpeangh@nus.edu.sg*
** *Department of Mechanical Engineering, The University of Melbourne, VIC 3010, Australia. Email: doetomo@unimelb.edu.au*

Abstract: An operational space controller that employs three-layer neural network (NN) adaptive motion control is presented in this paper. It is shown that the trajectory tracking errors and the NN weight errors are bounded and consequently the controller is shown to be stable. Comparative study is made between the performance of the proposed NN adaptive motion control strategy and the conventional inverse-dynamics control and PD plus gravity compensation controller. The simulation results show that the NN adaptive motion control can be as effective as the inverse-dynamics without the need for *a priori* knowledge of the system dynamics. Real-time implementation of the strategy was also conducted on a real PUMA 560 robot.

1. INTRODUCTION

To this day, many industrial serial manipulators are still limited to independent PID joint control. The usage of task space motion control is limited in the industries and even more so for simultaneous force-motion (compliant) control. Compliant control allows a manipulator to interact with its environment and it has many useful applications, such as to apply a controlled force needed for a manufacturing process (e.g. grinding, polishing), interacting with an object with a controlled force, or dealing with geometric uncertainty of the workpiece by establishing controlled contact forces. The operational space formulation for motion and force control [Khatib, 1987] provides a unified framework for this purpose, and its implementations are shown in [Jamisola et al., 2005, Khatib and Burdick, 1986]. Control for operational space framework is also known as inverse-dynamics control. It requires *a priori* knowledge of the robot dynamics. However, it is well-established that the system identification process for a robot is difficult to perform accurately [Armstrong et al., 1986, Jamisola et al., 1999].

The identification of Lagrangian dynamics for a particular robot is a two-step process. The first is to derive the symbolic expressions for the dynamic parameters, such as the inertia matrix, Coriolis/centrifugal, and gravity vector through closed-form Lagrangian energy considerations. Simplification follows to produce lumped parameters to reduce computational complexity in implementation. The second step is to measure the values of the dynamic parameters of the physical robot. Some experimental techniques have been proposed in the past, such as in [Jamisola et al., 1999], to measure the robot dynamics through its responses to specific commands sent to the joint actuators. However, it is difficult to obtain an accurate measurement and there is much room for human errors. It should also be noted that the errors in parameter identification accumulate with subsequent links in the serial manipulator.

Joint friction of a real robot is also difficult to obtain. There are three type of frictions: fluid/viscous friction, kinetic/Coulomb friction, and static friction. Friction parameters depend on the current ambient condition, so friction identification should ideally be performed immediately prior to the operation of the robot [Jamisola et al., 1999].

Early research efforts [Craig et al., 1986, Slotine and Li, 1987, Middleton and Goodwin, 1988, Ortega and Spong, 1988] turned to Linear-In-Parameter (LIP) adaptive control to seek answers for an easy-to-use high performance robot control strategy. In practice, however, LIP adaptive strategy still requires the derivation of the regression matrix out of the symbolic expressions of the dynamics equations, prior to the parameters estimation. Furthermore the dynamics equations in Cartesian space becomes more complicated than in joint space, making it difficult for implementation.

A three-layer Neural Network (NN) joint space adaptive motion control was proposed in [Lewis et al., 1996]. It was shown that satisfactory performance was achieved in the simulation. Several studies then followed in NN adaptive control in Cartesian space: such as in [Hu et al., 2000], [Kwan et al., 1994] where the task space force/motion NN adaptive control was formulated based upon model-based framework of [McClamroch and Wang, 1988]. However, this framework required the contact surface geometry, hence different surface would require different transformations making it difficult for real and robust implementation. Another work, [Ge et al., 1997] showed position only NN adaptive control in Cartesian space, which lacked the dexterity in accommodating the orientation control of the end-effector. Also while it uses the same dual-layer weight like [Lewis et al., 1996] but its first-layer weight is not tuned. The NN theory clearly states the dual-layer weight

NN can approximate any function, if only if both layers are tuned.

In this paper, a Neural Network motion controller in the operational space is proposed by adapting the joint space three-layer NN motion controller by [Lewis et al., 1996] into the operational space framework [Khatib, 1987]. The operational space formulation was selected due to its capability to accommodate position and orientation degrees-of-freedom and unified force/motion control, as well as its versatility in extension into redundant mechanisms and branching mechanisms Russakow et al. [1995]. The NN adaptive control flexibility is expected to remove the tedious effort of deriving and identifying the dynamic parameters required in the inverse-dynamics control inherent in the original operational space formulation. Stability analysis and proof is given in this paper as well as comparison to the performance of the inverse-dynamics and the PD plus gravity control strategies. Simulation and experimental results of Puma 560 are provided and discussed.

2. EFFECTOR DYNAMICS AND ITS PROPERTIES

The dynamics of a non-redundant serial manipulator can be expressed as:

$$\mathbf{M_x}(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{g_x}(\mathbf{q}) + \mathbf{f_x}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{F} \quad (1)$$

where $\mathbf{x} \in \Re^n$ denotes the operational space coordinates and $\mathbf{q} \in \Re^n$ the joint space coordinates. The matrices $\mathbf{M_x}(\mathbf{q}) \in \Re^{n\times n}$ and $\mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}}) \in \Re^{n\times n}$ denote the inertia and the Coriolis/centrifugal terms, respectively, while vectors $\mathbf{g_x}(\mathbf{q}) \in \Re^n$ and $\mathbf{f_x}(\mathbf{q},\dot{\mathbf{q}}) \in \Re^n$ denote the gravity and friction forces. Vector $\mathbf{F} \in \Re^n$ denotes the operational space forces as control input. The subscript '$\mathbf{x}$'denotes that the matrices and vectors are expressed in operational space. The relationships between the components of the joint space dynamic and the operational space dynamics of a manipulator in non-singular configuration can be expressed as:

$$\mathbf{M_x}(\mathbf{q}) = \mathbf{J}^{-\mathrm{T}}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}^{-1}(\mathbf{q}) \quad (2)$$

$$\mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}}) = [\mathbf{J}^{-T}(\mathbf{q})\mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{M_x}(\mathbf{q})\dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})]\mathbf{J}^{-1}(\mathbf{q}) \quad (3)$$

$$\mathbf{g_x}(\mathbf{q}) = \mathbf{J}^{-\mathrm{T}}(\mathbf{q})\mathbf{g}(\mathbf{q}) \quad (4)$$

$$\mathbf{f_x}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{J}^{-\mathrm{T}}(\mathbf{q})\boldsymbol{\tau}_f(\dot{\mathbf{q}}) \quad (5)$$

where $\mathbf{J}(\mathbf{q})$ is the basic Jacobian and $\mathbf{M}(\mathbf{q})$, $\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})$, $\mathbf{g}(\mathbf{q})$ are as stated above, but expressed in joint space. The joint friction $\boldsymbol{\tau}_f(\dot{\mathbf{q}})$ can be obtained by Xia et al. [2004]:

$$\boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \boldsymbol{\tau}_{vis}\dot{\mathbf{q}} + \left[\boldsymbol{\tau}_{cou} + \boldsymbol{\tau}_{sti}\exp^{(-\boldsymbol{\tau}_{dec}\dot{\mathbf{q}}^2)}\right]\mathrm{sgn}(\dot{\mathbf{q}}) \quad (6)$$

where $\mathrm{sgn}(\dot{q}) = +1, -1, 0$ if $\dot{q}$ = positive, negative and zero respectively and $\boldsymbol{\tau}_{vis}$, $\boldsymbol{\tau}_{cou}$, $\boldsymbol{\tau}_{sti}$, and $\boldsymbol{\tau}_{dec}$ are the viscous friction, Coulomb friction, stiction, and Stribeck effect, respectively. In order to develop the NN adaptive controller in the operational space framework, the following properties of the effector dynamics (1) are utilized:

Property 1. The operational space kinetic energy matrix $\mathbf{M_x}(\mathbf{q})$ is symmetric and positive definite due to (2) and the joint space kinetic energy $\mathbf{M}(\mathbf{q})$ is symmetric and positive definite by definition. Hence according to Rayleigh-Ritz theorem Slotine and Li [1991]:

$$M_m\|\mathbf{z}\|^2 \le \mathbf{z}^{\mathrm{T}}\mathbf{M_x}(\mathbf{q})\mathbf{z} \le M_M\|\mathbf{z}\|^2 \quad (7)$$

where M_m and M_M denote the minimum and maximum eigenvalues of $\mathbf{M_x}(\mathbf{q})$ respectively. Moreover any positive definite matrix $\mathbf{A}(\mathbf{y})$ satisfies:

$$A_m \le \|\mathbf{A}(\mathbf{y})\| \le A_M \quad (8)$$

where A_m and A_M denote the minimum and maximum eigenvalues of $\mathbf{A}(\mathbf{y})$ respectively. Unless otherwise specified, all norms are defined as 2-norm (Frobenius norm).

Property 2. $\dot{\mathbf{M}}_\mathbf{x}(\mathbf{q}) - 2\mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}})$ is skew-symmetric Lewis et al. [1993], hence

$$\mathbf{z}^{\mathrm{T}}\left(\dot{\mathbf{M}}_\mathbf{x}(\mathbf{q}) - 2\mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}})\right)\mathbf{z} = 0, \quad \mathbf{z} \in \Re^n \quad (9)$$

Property 3. In joint space, each term in $\boldsymbol{\tau}_f(\dot{\mathbf{q}})$ satisfies [Lewis et al., 1993]:

$$\|\boldsymbol{\tau}_{vis}\dot{\mathbf{q}}\| \le \tau_{vis,M}\|\dot{\mathbf{q}}\| \quad (10)$$

$$\|\boldsymbol{\tau}_{cou}\mathrm{sgn}(\dot{\mathbf{q}})\| \le \tau_{cou,M} \quad (11)$$

$$\|\boldsymbol{\tau}_{sti}\exp^{(-\boldsymbol{\tau}_{dec}\dot{\mathbf{q}}^2)}\mathrm{sgn}(\dot{\mathbf{q}})\| \le \tau_{sti,M} \quad (12)$$

Property 4. The operational space gravity vector $\mathbf{g_x}(\mathbf{q})$ is upper-bounded, from (4):

$$\|\mathbf{g_x}(\mathbf{q})\| \le g_M < \infty \quad (13)$$

3. NN ADAPTIVE MOTION CONTROLLER FORMULATION

3.1 NN Adaptive Motion Controller

In this section, the operational space NN-based adaptive motion controller is proposed as:

$$\mathbf{F} = \mathbf{K_v}\mathbf{r} + \hat{\mathbf{M}}_\mathbf{x}(\mathbf{q})\ddot{\mathbf{x}}_r + \hat{\mathbf{B}}_\mathbf{x}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}}_r + \hat{\mathbf{g}}_\mathbf{x}(\mathbf{q}) + \hat{\mathbf{f}}_\mathbf{x}(\mathbf{q},\dot{\mathbf{q}}) \quad (14)$$

In this paper, the estimate of a dynamic parameter m is represented as $\hat{m}$ and the error dynamics are represented by $\tilde{m} = m - \hat{m}$. The following terms are defined:

$$\begin{aligned} \ddot{\mathbf{x}}_r &= \ddot{\mathbf{x}}_d + \boldsymbol{\Lambda}\dot{\mathbf{e}} + \boldsymbol{\Lambda}_i\mathbf{e} \\ \dot{\mathbf{x}}_r &= \dot{\mathbf{x}}_d + \boldsymbol{\Lambda}\mathbf{e} + \boldsymbol{\Lambda}_i\int_0^{\tau=t}\mathbf{e}\,\mathrm{d}\tau \\ \mathbf{r} &= \dot{\mathbf{e}} + \boldsymbol{\Lambda}\mathbf{e} + \boldsymbol{\Lambda}_i\int_0^{\tau=t}\mathbf{e}\,\mathrm{d}\tau \end{aligned} \quad (15)$$

$\boldsymbol{\Lambda}, \boldsymbol{\Lambda}_i \in \Re^{n\times n}$ are positive diagonal matrices, $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ and $\dot{\mathbf{e}} = \dot{\mathbf{x}}_d - \dot{\mathbf{x}}$ are the trajectory tracking error and velocity error, $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$ and $\ddot{\mathbf{x}}_d$ are the desired operational space coordinates.

Combining the manipulator dynamics (1) and the proposed controller (14):

$$\mathbf{M_x}(\mathbf{q})\dot{\mathbf{r}} = -\mathbf{K_v}\mathbf{r} - \mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}})\mathbf{r} + \boldsymbol{\eta}; \quad (16)$$

where the uncertainties of the system $\boldsymbol{\eta}$

$$\boldsymbol{\eta} = \tilde{\mathbf{M}}_\mathbf{x}(\mathbf{q})\ddot{\mathbf{x}}_r + \tilde{\mathbf{B}}_\mathbf{x}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}}_r + \tilde{\mathbf{g}}_\mathbf{x}(\mathbf{q}) + \tilde{\mathbf{f}}_\mathbf{x}(\mathbf{q},\dot{\mathbf{q}}). \quad (17)$$

3.2 Three-Layer Neural Networks

Given N_1, N_2 and N_3 are the number of neurons in layer 1, 2 and 3, respectively, an input vector $\mathbf{z} \in \Re^{N_1}$ can be defined where v_{kl} is the first-to-second layer node weights, $l = 1,\ldots,N_1$, $k = 1,\ldots,N_2$, θ_k is the threshold offset, and w_{ik} is second-to-third layer node weights where $i = 1,\ldots,N_3$. For a three-layer NN with an output vector $y \in \Re^{N_3}$, we can write

$$y_i = \sum_{k=1}^{N_2} w_{ik}\ \sigma_k\left(\sum_{l=1}^{N_1} v_{kl}z_l\ +\ \theta_k\right); i = 1,\ldots,N_3 \quad (18)$$

and for an output matrix $y \in \Re^{N_3\times N_4}$, we can write

$$\begin{aligned} y_{ij} = \sum_{k=1}^{N_2} w_{ijk}\ \sigma_k\left(\sum_{l=1}^{N_1} v_{kl}z_l\ +\ \theta_k\right); \\ i = 1,\ldots,N_3, j = 1,\ldots,N_4 \end{aligned} \quad (19)$$

where function $\sigma(\cdot)$ is differentiable such as sigmoid and hyperbolic functions. We can write (18) and (19) compactly in vector form ($\mathbf{y}$) and matrix form ($\mathbf{Y}$):

$$\mathbf{y} = \{\mathbf{W}\}^{\mathrm{T}}\boldsymbol{\sigma}\left(\{\mathbf{V}\}^{\mathrm{T}}\mathbf{z}\right), \quad \mathbf{Y} = \{\mathbf{W}\}^{\mathrm{T}}\boldsymbol{\sigma}\left(\{\mathbf{V}\}^{\mathrm{T}}\mathbf{z}\right) \tag{20}$$

3.3 Uncertainties $\boldsymbol{\eta}$ in NN terms

Since the estimate of a dynamic parameter m is represented as $\hat{m}$ and the error dynamics are represented by $\tilde{m} = m - \hat{m}$ then the uncertainties $\boldsymbol{\eta}$ (17) can be written as

$$\begin{aligned}\boldsymbol{\eta} = &(\mathbf{M_x} - \hat{\mathbf{M}}_\mathbf{x})(\mathbf{q})\ddot{\mathbf{x}}_r + (\mathbf{B_x} - \hat{\mathbf{B}}_\mathbf{x})(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}}_r \\ &+ (\mathbf{g_x} - \hat{\mathbf{g}}_\mathbf{x})(\mathbf{q}) + (\mathbf{f_x} - \hat{\mathbf{f}}_\mathbf{x})(\mathbf{q},\dot{\mathbf{q}})\end{aligned} \tag{21}$$

From NN theory, given unlimited number of hidden layer, three layer NNs with ideal weights can approximate any functions. But in practice we have limited number of hidden layer nodes, thus the dynamic terms $\mathbf{M_x}(\mathbf{q})$, $\mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}})$, $\mathbf{g_x}(\mathbf{q})$, and $\mathbf{f_x}(\mathbf{q},\dot{\mathbf{q}})$ can be approximated by three-layer NNs with "optimum"weights $\{\mathbf{V}\}, \{\mathbf{W}\}$ and approximation error $\boldsymbol{\varepsilon}$:

$$\mathbf{M_x}(\mathbf{q}) = \{\mathbf{W_M}\}^{\mathrm{T}}\boldsymbol{\sigma}_\mathbf{M}(\{\mathbf{V_M}\}^{\mathrm{T}}\mathbf{z_M}) + \boldsymbol{\varepsilon}_\mathbf{M} \tag{22}$$

$$\mathbf{B_x}(\mathbf{q},\dot{\mathbf{q}}) = \{\mathbf{W_B}\}^{\mathrm{T}}\boldsymbol{\sigma}_\mathbf{B}(\{\mathbf{V_B}\}^{\mathrm{T}}\mathbf{z_B}) + \boldsymbol{\varepsilon}_\mathbf{B} \tag{23}$$

$$\mathbf{g_x}(\mathbf{q}) = \{\mathbf{W_g}\}^{\mathrm{T}}\boldsymbol{\sigma}_\mathbf{g}(\{\mathbf{V_g}\}^{\mathrm{T}}\mathbf{z_g}) + \boldsymbol{\varepsilon}_\mathbf{g} \tag{24}$$

$$\mathbf{f_x}(\mathbf{q},\dot{\mathbf{q}}) = \{\mathbf{W_f}\}^{\mathrm{T}}\boldsymbol{\sigma}_\mathbf{f}(\{\mathbf{V_f}\}^{\mathrm{T}}\mathbf{z_f}) + \boldsymbol{\varepsilon}_\mathbf{f} \tag{25}$$

Likewise the estimated dynamic terms $\hat{\mathbf{M}}_\mathbf{x}(\mathbf{q})$, $\hat{\mathbf{B}}_\mathbf{x}(\mathbf{q},\dot{\mathbf{q}})$, $\hat{\mathbf{g}}_\mathbf{x}(\mathbf{q})$, and $\hat{\mathbf{f}}_\mathbf{x}(\mathbf{q},\dot{\mathbf{q}})$ are approximated by estimated weights $\{\hat{\mathbf{V}}_i\}, \{\hat{\mathbf{W}}_i\}$ where subscript $i = M, B, g, f$ represents the individual dynamical terms.

The following generic NN expressions are then defined for ease of representation such that:

$$\begin{aligned}\mathbf{L}_i &= \{\mathbf{W}_i\}^{\mathrm{T}}\boldsymbol{\sigma}_i(\{\mathbf{V}_i\}^{\mathrm{T}}\mathbf{z}_i) \\ \hat{\mathbf{L}}_i &= \{\hat{\mathbf{W}}_i\}^{\mathrm{T}}\boldsymbol{\sigma}_i(\{\hat{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i) \\ \tilde{\mathbf{L}}_i &= \mathbf{L}_i - \hat{\mathbf{L}}_i;\end{aligned} \tag{26}$$

where $\mathbf{L}_i$, $\hat{\mathbf{L}}_i$, and $\tilde{\mathbf{L}}_i$ represent the actual, estimated, and error, of the variables accordingly.

Hence, using the generic NN expressions, the uncertainties (21) can be written as

$$\begin{aligned}\boldsymbol{\eta} = &(\mathbf{L}_M - \hat{\mathbf{L}}_M)\ddot{\mathbf{x}}_r + (\mathbf{L}_B - \hat{\mathbf{L}}_B)\dot{\mathbf{x}}_r + (\mathbf{L}_g - \hat{\mathbf{L}}_g) \\ &+ (\mathbf{L}_f - \hat{\mathbf{L}}_f) + \boldsymbol{\varepsilon}\end{aligned} \tag{27}$$

where the total approximation error $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_\mathbf{M}\ddot{\mathbf{x}}_r + \boldsymbol{\varepsilon}_\mathbf{B}\dot{\mathbf{x}}_r + \boldsymbol{\varepsilon}_\mathbf{g} + \boldsymbol{\varepsilon}_\mathbf{f}$. To compute $\boldsymbol{\eta}$ (27), it is necessary to compute the generic form of

$$\mathbf{L}_i - \hat{\mathbf{L}}_i = \{\mathbf{W}_i\}^{\mathrm{T}}\boldsymbol{\sigma}\left(\{\mathbf{V}_i\}^{\mathrm{T}}\mathbf{z}_i\right) - \{\hat{\mathbf{W}}_i\}^{\mathrm{T}}\boldsymbol{\sigma}\left(\{\hat{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i\right). \tag{28}$$

The error in the sigmoid function of the first to second layer weights is calculated as:

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}\left(\{\mathbf{V}\}^{\mathrm{T}}\mathbf{z}\right) - \boldsymbol{\sigma}\left(\{\hat{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z}\right). \tag{29}$$

From the Taylor series expansion

$$\boldsymbol{\sigma}(\mathbf{k})\big|_{\mathbf{k}=\hat{\mathbf{k}}} = \boldsymbol{\sigma}(\hat{\mathbf{k}}) + \frac{d\boldsymbol{\sigma}(\mathbf{k})}{d\mathbf{k}}(\mathbf{k} - \hat{\mathbf{k}}) + O(\mathbf{k} - \hat{\mathbf{k}}) \tag{30}$$

where $O(\mathbf{k} - \hat{\mathbf{k}})$ denotes higher order terms. Note that $\boldsymbol{\sigma}'(\mathbf{k}) = \frac{\mathbf{d}\boldsymbol{\sigma}(\mathbf{k})}{d\mathbf{k}}\big|_{\mathbf{k}=\widehat{\mathbf{k}}}$, $\boldsymbol{\sigma}$ is differentiable, thus $\boldsymbol{\sigma}'$ exist. Hence $\boldsymbol{\sigma}(\{\mathbf{V}\}^{\mathrm{T}}\mathbf{z})\big|_{\{\mathbf{V}\}^{\mathrm{T}}\mathbf{z}=\{\hat{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z}}$ in (29) can be written as

$$\begin{aligned}\boldsymbol{\sigma}(\{\mathbf{V}\}^{\mathrm{T}}\mathbf{z}) = &\boldsymbol{\sigma}(\{\hat{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z}) + \boldsymbol{\sigma}'(\{\hat{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z})\{\tilde{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z} \\ &+ O(\{\tilde{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z})\end{aligned} \tag{31}$$

For simplification of notation, it is defined that $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_i(\{\mathbf{V}_i\}^{\mathrm{T}}\mathbf{z}_i)$, $\hat{\boldsymbol{\sigma}}_i = \boldsymbol{\sigma}_i(\{\hat{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i)$, and $\boldsymbol{\sigma}_i = \hat{\boldsymbol{\sigma}}_i + \tilde{\boldsymbol{\sigma}}_i$.

Using (31), we can rewrite (29) as:

$$\tilde{\boldsymbol{\sigma}} = \hat{\boldsymbol{\sigma}}'\{\tilde{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z} + O\left(\{\tilde{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z}\right), \tag{32}$$

and substituting (32) into (28) yields:

$$\begin{aligned}&\mathbf{L}_i - \hat{\mathbf{L}}_i \\ &= \{\mathbf{W}_i\}^{\mathrm{T}}\boldsymbol{\sigma}_i - \{\hat{\mathbf{W}}_i\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_i - \{\mathbf{W}_i\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_i + \{\mathbf{W}_i\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_i \\ &= \{\tilde{\mathbf{W}}_i\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_i + \{\mathbf{W}_i\}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}}_i \\ &= \{\tilde{\mathbf{W}}_i\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_i + \{\mathbf{W}_i\}^{\mathrm{T}}\left[\hat{\boldsymbol{\sigma}}_i'\{\tilde{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i + O\left(\{\tilde{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i\right)\right] \\ &= \{\tilde{\mathbf{W}}_i\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_i + (\{\hat{\mathbf{W}}_i\}^{\mathrm{T}} + \{\tilde{\mathbf{W}}_i\}^{\mathrm{T}})\left[\hat{\boldsymbol{\sigma}}_i'\{\tilde{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i + O\left(\{\tilde{\mathbf{V}}_i\}^{\mathrm{T}}\mathbf{z}_i\right)\right]\end{aligned} \tag{33}$$

Using (33), the uncertainties $\boldsymbol{\eta}$ (27) can be divided into

$$\boldsymbol{\eta} = \boldsymbol{\xi} + \mathbf{w} \tag{34}$$

where

$$\begin{aligned}\boldsymbol{\xi} = &\left(\{\tilde{\mathbf{W}}_\mathbf{M}\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_\mathbf{M}\right)\ddot{\mathbf{x}}_r + \left(\{\tilde{\mathbf{W}}_\mathbf{B}\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_\mathbf{B}\right)\dot{\mathbf{x}}_r + \{\tilde{\mathbf{W}}_\mathbf{g}\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_\mathbf{g} \\ &+ \{\tilde{\mathbf{W}}_\mathbf{f}\}^{\mathrm{T}}\hat{\boldsymbol{\sigma}}_\mathbf{f} + \left(\{\hat{\mathbf{W}}_\mathbf{M}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{M}\{\tilde{\mathbf{V}}_\mathbf{M}\}^{\mathrm{T}}\mathbf{z_M}\right)\ddot{\mathbf{x}}_r \\ &+ \left(\{\hat{\mathbf{W}}_\mathbf{B}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{B}\{\tilde{\mathbf{V}}_\mathbf{B}\}^{\mathrm{T}}\mathbf{z_B}\right)\dot{\mathbf{x}}_r \\ &+ \{\hat{\mathbf{W}}_\mathbf{g}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{g}\{\tilde{\mathbf{V}}_\mathbf{g}\}^{\mathrm{T}}\mathbf{z_g} + \{\hat{\mathbf{W}}_\mathbf{f}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{f}\{\tilde{\mathbf{V}}_\mathbf{f}\}^{\mathrm{T}}\mathbf{z_f}\end{aligned} \tag{35}$$

and the "whole"errors $\mathbf{w}$

$$\begin{aligned}\mathbf{w} = &\left(\{\tilde{\mathbf{W}}_\mathbf{M}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{M}\{\tilde{\mathbf{V}}_\mathbf{M}\}^{\mathrm{T}}\mathbf{z_M}\right)\ddot{\mathbf{x}}_r + \left(\{\tilde{\mathbf{W}}_\mathbf{B}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{B}\{\tilde{\mathbf{V}}_\mathbf{B}\}^{\mathrm{T}}\mathbf{z_B}\right)\dot{\mathbf{x}}_r \\ &+ \{\tilde{\mathbf{W}}_\mathbf{g}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{g}\{\tilde{\mathbf{V}}_\mathbf{g}\}^{\mathrm{T}}\mathbf{z_g} + \{\tilde{\mathbf{W}}_\mathbf{f}\}^{\mathrm{T}}\boldsymbol{\sigma}'_\mathbf{f}\{\tilde{\mathbf{V}}_\mathbf{f}\}^{\mathrm{T}}\mathbf{z_f} \\ &+ \left(\{\mathbf{W_M}\}^{\mathrm{T}}O(\{\tilde{\mathbf{V}}_\mathbf{M}\}^{\mathrm{T}}\mathbf{z_M}\right)\ddot{\mathbf{x}}_r + \left(\{\mathbf{W_B}\}^{\mathrm{T}}O(\{\tilde{\mathbf{V}}_\mathbf{B}\}^{\mathrm{T}}\mathbf{z_B}\right)\dot{\mathbf{x}}_r \\ &+ \{\mathbf{W_g}\}^{\mathrm{T}}O(\{\tilde{\mathbf{V}}_\mathbf{g}\}^{\mathrm{T}}\mathbf{z_g} + \{\mathbf{W_f}\}^{\mathrm{T}}O(\{\tilde{\mathbf{V}}_\mathbf{f}\}^{\mathrm{T}}\mathbf{z_f} + \boldsymbol{\varepsilon}\end{aligned} \tag{36}$$

This division is necessary because in the Lyapunov analysis (Section 3.4), it becomes evident that only $\boldsymbol{\xi}$ term can be canceled by the weight updates. A non-strict assumption that $\mathbf{w}, \boldsymbol{\xi}$ are bounded is defined, i.e.:

$$\|\mathbf{w}\| \leq w_M, \quad \|\boldsymbol{\xi}\| \leq \xi_M. \tag{37}$$

This is because:

- $\{\mathbf{W}\}, \{\mathbf{V}\}$ and $\boldsymbol{\varepsilon}$ are bounded, since the actual dynamics are bounded.
- $\hat{\boldsymbol{\sigma}}$, its derivative and $O(\{\tilde{\mathbf{V}}\}^{\mathrm{T}}\mathbf{z})$ are bounded for differentiable functions like sigmoid, tanh, RBF functions,
- the desired trajectories $\|\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d\| \leq x_{d,M}$ are bounded by design,
- $\{\tilde{\mathbf{W}}\}$ and $\{\tilde{\mathbf{V}}\}$ are assumed bounded, implying that $\{\hat{\mathbf{W}}\}$ and $\{\hat{\mathbf{V}}\}$ are also bounded. (This assumption is validated when $\{\tilde{\mathbf{W}}\}$ and $\{\tilde{\mathbf{V}}\}$ are shown to be bounded in Section 3.4.)
- $\dot{\mathbf{e}}$ and $\mathbf{e}$ are assumed bounded, implying $\dot{\mathbf{x}}, \mathbf{x}$ and $\dot{\mathbf{q}}, \mathbf{q}$ are also bounded. (This assumption is validated when $\dot{\mathbf{e}}$ and $\mathbf{e}$ are shown to be bounded in Section 3.4.)
- $\ddot{\mathbf{x}}_r$ and $\dot{\mathbf{x}}_r$ can be assumed bounded because $\dot{\mathbf{e}}$, $\mathbf{e}$ are assumed bounded (by the previous point) and the desired trajectories are bounded.

3.4 Stability Analysis

For the stability analysis, $\mathbf{Z} = diag[\mathbf{W}, \mathbf{V}]$ is defined, such that $\|\mathbf{Z}\| \leq Z_M$, where $\mathbf{W} = diag[\mathbf{W_M}, \mathbf{W_B}, \mathbf{W_g}, \mathbf{W_f}]$ and $\mathbf{V} = diag[\mathbf{V_M}, \mathbf{V_B}, \mathbf{V_g}, \mathbf{V_f}]$.

Theorem 1. Let the NN weight updates be provided as:

$$\dot{\hat{\mathbf{W}}}_{\mathbf{M}ij} = \mathbf{F}_{\mathbf{M}ij}(\hat{\boldsymbol{\sigma}}_{\mathbf{M}}\, r_i\, \ddot{x}_{rj} - \kappa\|\mathbf{r}\|\hat{\mathbf{W}}_{\mathbf{M}ij}) \tag{38}$$

$$\dot{\hat{\mathbf{V}}}_{\mathbf{M}k} = \mathbf{G}_{\mathbf{M}k} \quad (\mathbf{z}_{\mathbf{M}}\, \hat{\boldsymbol{\sigma}}'_{\mathbf{M}k}(\sum_{i=1}^{n}\sum_{j=1}^{n} \hat{\mathbf{W}}_{\mathbf{M}ijk}\, r_i\, \ddot{x}_{rj}) - \kappa\|\mathbf{r}\|\hat{\mathbf{V}}_{\mathbf{M}k}) \tag{39}$$

$$\dot{\hat{\mathbf{W}}}_{\mathbf{B}ij} = \mathbf{F}_{\mathbf{B}ij}(\hat{\boldsymbol{\sigma}}_{\mathbf{B}}\, r_i\, \dot{x}_{rj} - \kappa\|\mathbf{r}\|\hat{\mathbf{W}}_{\mathbf{B}ij}) \tag{40}$$

$$\dot{\hat{\mathbf{V}}}_{\mathbf{B}k} = \mathbf{G}_{\mathbf{B}k}(\mathbf{z}_{\mathbf{B}}\, \hat{\boldsymbol{\sigma}}'_{\mathbf{B}k}(\sum_{i=1}^{n}\sum_{j=1}^{n} \hat{\mathbf{W}}_{\mathbf{B}ijk}\, r_i\, \dot{x}_{rj}) - \kappa\|\mathbf{r}\|\hat{\mathbf{V}}_{\mathbf{B}k}) \tag{41}$$

$$\dot{\hat{\mathbf{W}}}_{\mathbf{g}i} = \mathbf{F}_{\mathbf{g}i}(\hat{\boldsymbol{\sigma}}_{\mathbf{g}}\, r_i - \kappa\|\mathbf{r}\|\hat{\mathbf{W}}_{\mathbf{g}i}) \tag{42}$$

$$\dot{\hat{\mathbf{V}}}_{\mathbf{g}k} = \mathbf{G}_{\mathbf{g}k}(\mathbf{z}_{\mathbf{g}}\, \hat{\boldsymbol{\sigma}}'_{\mathbf{g}k}(\sum_{i=1}^{n} \hat{\mathbf{W}}_{\mathbf{g}ik}\, r_i) - \kappa\|\mathbf{r}\|\hat{\mathbf{V}}_{\mathbf{g}k}) \tag{43}$$

$$\dot{\hat{\mathbf{W}}}_{\mathbf{f}i} = \mathbf{F}_{\mathbf{f}i}(\hat{\boldsymbol{\sigma}}_{\mathbf{f}}\, r_i - \kappa\|\mathbf{r}\|\hat{\mathbf{W}}_{\mathbf{f}i}) \tag{44}$$

$$\dot{\hat{\mathbf{V}}}_{\mathbf{f}k} = \mathbf{G}_{\mathbf{f}k}(\mathbf{z}_{\mathbf{f}}\, \hat{\boldsymbol{\sigma}}'_{\mathbf{f}k}(\sum_{i=1}^{n} \hat{\mathbf{W}}_{\mathbf{f}ik}\, r_i) - \kappa\|\mathbf{r}\|\hat{\mathbf{V}}_{\mathbf{f}k}) \tag{45}$$

where output nodes $i, j = 1, \ldots, n$ and hidden nodes $k = 1, \ldots, k_i$ (the subscript 'i'$\equiv M, B, g, f$). Then the region of attraction of the error dynamics (16), is given by:

$$\mathbf{S} = \left\{\mathbf{r} : b_{\mathbf{r}} < \|\mathbf{r}(t)\| < \infty,\ \tilde{\mathbf{Z}} : b_{\tilde{\mathbf{Z}}} < \|\tilde{\mathbf{Z}}(t)\| < \infty\right\} \tag{46}$$

where $b_{\mathbf{r}}$ and $b_{\tilde{\mathbf{Z}}}$ are defined later in equations (56) and (57), respectively.

Having defined the error dynamics in (16) and the uncertainties $\boldsymbol{\eta}$ (34), the Lyapunov function can be defined as

$$V(\mathbf{r}, \tilde{\mathbf{Z}}) = \frac{1}{2}\,\mathbf{r}^{\mathrm{T}}\mathbf{M}_{\mathbf{x}}(\mathbf{q})\mathbf{r} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{\mathbf{W}}_{\mathbf{M}ij}^{\mathrm{T}}\, \mathbf{F}_{\mathbf{M}ij}^{-1}\, \tilde{\mathbf{W}}_{\mathbf{M}ij} + \ldots + \frac{1}{2}\sum_{k=1}^{k_f} \tilde{\mathbf{V}}_{\mathbf{f}k}^{\mathrm{T}}\, \mathbf{G}_{\mathbf{f}_k}^{-1}\, \tilde{\mathbf{V}}_{\mathbf{f}k} \tag{47}$$

where $\tilde{\mathbf{W}}_{\mathbf{M}ij} \in \Re^{k_M}, \ldots, \tilde{\mathbf{W}}_{\mathbf{f}i} \in \Re^{k_f}$ and $\tilde{\mathbf{V}}_{\mathbf{M}k} \in \Re^{l_M}, \ldots, \tilde{\mathbf{V}}_{\mathbf{f}_k} \in \Re^{l_f}$ and $\mathbf{F}_{\mathbf{M}ij}^{-1} \in \Re^{k_M \times k_M}, \ldots, \mathbf{F}_{\mathbf{f}_i}^{-1} \in \Re^{k_f \times k_f}$ and $\mathbf{G}_{\mathbf{M}_k}^{-1} \in \Re^{l_M \times l_M}, \ldots, \mathbf{G}_{\mathbf{f}_k}^{-1} \in \Re^{l_f \times l_f}$ are positive diagonal matrices. With l_i is the input nodes size (the subscript 'i'$\equiv M, B, g, f$).

Then the error dynamics (16), property 2 and the uncertainties $\boldsymbol{\eta}$ (34) are substituted into $\dot{V}(\mathbf{r}, \tilde{\mathbf{Z}})$ to yield:

$$\dot{V}(\mathbf{r}, \tilde{\mathbf{Z}}) = -\mathbf{r}^{\mathrm{T}}\mathbf{K}_{\mathbf{v}}\mathbf{r} + \mathbf{r}^{\mathrm{T}}\mathbf{w} + \boldsymbol{\nu} \tag{48}$$

where

$$\begin{aligned}\boldsymbol{\nu} = &\sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{\mathbf{W}}_{\mathbf{M}ij}^{\mathrm{T}} \left(\mathbf{F}_{\mathbf{M}ij}^{-1}\, \dot{\tilde{\mathbf{W}}}_{\mathbf{M}ij} + \hat{\boldsymbol{\sigma}}_{\mathbf{M}} r_i \ddot{x}_{rj}\right) \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{\mathbf{W}}_{\mathbf{B}ij}^{\mathrm{T}} \left(\mathbf{F}_{\mathbf{B}ij}^{-1}\, \dot{\tilde{\mathbf{W}}}_{\mathbf{B}ij} + \hat{\boldsymbol{\sigma}}_{\mathbf{B}} r_i \dot{x}_{rj}\right) \\ &+ \sum_{i=1}^{n} \tilde{\mathbf{W}}_{\mathbf{g}i}^{\mathrm{T}} \left(\mathbf{F}_{\mathbf{g}i}^{-1}\, \dot{\tilde{\mathbf{W}}}_{\mathbf{g}i} + \hat{\boldsymbol{\sigma}}_{\mathbf{g}} r_i\right) + \sum_{i=1}^{n} \tilde{\mathbf{W}}_{\mathbf{f}_i}^{\mathrm{T}} \left(\mathbf{F}_{\mathbf{f}_i}^{-1}\, \dot{\tilde{\mathbf{W}}}_{\mathbf{f}i} + \hat{\boldsymbol{\sigma}}_{\mathbf{f}} r_i\right)\end{aligned} \tag{49}$$

$$\begin{aligned}&+ \sum_{k=1}^{l_M} \tilde{\mathbf{V}}_{\mathbf{M}_k}^{\mathrm{T}} \left(\mathbf{G}_{\mathbf{M}_k}^{-1}\, \dot{\hat{\mathbf{V}}}_{\mathbf{M}k} + \mathbf{z}_{\mathbf{M}} \hat{\boldsymbol{\sigma}}'_{\mathbf{M}k}(\sum_{i=1}^{n}\sum_{j=1}^{n} \hat{\mathbf{W}}_{\mathbf{M}ijk} r_i \ddot{x}_{rj})\right) \\ &+ \sum_{k=1}^{l_B} \tilde{\mathbf{V}}_{\mathbf{B}_k}^{\mathrm{T}} \left(\mathbf{G}_{\mathbf{B}_k}^{-1}\, \dot{\hat{\mathbf{V}}}_{\mathbf{B}k} + \mathbf{z}_{\mathbf{B}} \hat{\boldsymbol{\sigma}}'_{\mathbf{B}k}(\sum_{i=1}^{n}\sum_{j=1}^{n} \hat{\mathbf{W}}_{\mathbf{B}ijk} r_i \dot{x}_{rj})\right) \\ &+ \sum_{k=1}^{l_g} \tilde{\mathbf{V}}_{\mathbf{g}_k}^{\mathrm{T}} \left(\mathbf{G}_{\mathbf{g}_k}^{-1}\, \dot{\hat{\mathbf{V}}}_{\mathbf{g}k} + \mathbf{z}_{\mathbf{g}} \hat{\boldsymbol{\sigma}}'_{\mathbf{g}k}(\sum_{i=1}^{n} \hat{\mathbf{W}}_{\mathbf{g}ik} r_i)\right) \\ &+ \sum_{k=1}^{l_f} \tilde{\mathbf{V}}_{\mathbf{f}_k}^{\mathrm{T}} \left(\mathbf{G}_{\mathbf{f}_k}^{-1}\, \dot{\hat{\mathbf{V}}}_{\mathbf{f}k} + \mathbf{z}_{\mathbf{f}} \hat{\boldsymbol{\sigma}}'_{\mathbf{f}k}(\sum_{i=1}^{n} \hat{\mathbf{W}}_{\mathbf{f}ik} r_i)\right)\end{aligned}$$

It can be seen that ν is made up of the derivatives of $\dot{\tilde{\mathbf{W}}}$ and $\dot{\tilde{\mathbf{V}}}$ as well as $\boldsymbol{\xi}$. The idea is to cancel $\boldsymbol{\xi}$ with $\dot{\tilde{\mathbf{W}}}$. Hence, it is shown that only $\boldsymbol{\xi}$ can be accommodated by the updates of the weights. Furthermore, $-\dot{\tilde{\mathbf{W}}} = \dot{\hat{\mathbf{W}}}$, since $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$ and $\mathbf{W}$ is constant. With the weight updates (38)– (45), $\boldsymbol{\nu}$ becomes

$$\begin{aligned}\boldsymbol{\nu} = &\kappa\|\mathbf{r}\| \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{\mathbf{W}}_{\mathbf{M}ij}^{\mathrm{T}} \hat{\mathbf{W}}_{\mathbf{M}ij} + \ldots + \kappa\|\mathbf{r}\| \sum_{i=1}^{n} \tilde{\mathbf{W}}_{\mathbf{f}_i}^{\mathrm{T}} \hat{\mathbf{W}}_{\mathbf{f}i} \\ &+ \kappa\|\mathbf{r}\| \sum_{k=1}^{k_M} \tilde{\mathbf{V}}_{\mathbf{M}_k}^{\mathrm{T}} \hat{\mathbf{V}}_{\mathbf{M}k} + \ldots + \kappa\|\mathbf{r}\| \sum_{k=1}^{k_f} \tilde{\mathbf{V}}_{\mathbf{f}_k}^{\mathrm{T}} \hat{\mathbf{V}}_{\mathbf{f}k} \\ \leq &-\kappa\|\mathbf{r}\|\|\tilde{\mathbf{Z}}\|^2 + \kappa\|\mathbf{r}\|\|\tilde{\mathbf{Z}}\| Z_M\end{aligned} \tag{50}$$

Equation (50) is obtained by making use of

$$\langle \tilde{\mathbf{W}}, \hat{\mathbf{W}} \rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{\mathbf{W}}_{\mathbf{M}ij}^{\mathrm{T}} \hat{\mathbf{W}}_{\mathbf{M}ij} + \ldots + \sum_{i=1}^{n} \tilde{\mathbf{W}}_{\mathbf{f}_i}^{\mathrm{T}} \hat{\mathbf{W}}_{\mathbf{f}i} \tag{51}$$

$$\langle \tilde{\mathbf{V}}, \hat{\mathbf{V}} \rangle = \sum_{k=1}^{k_M} \tilde{\mathbf{V}}_{\mathbf{M}_k}^{\mathrm{T}} \hat{\mathbf{V}}_{\mathbf{M}k} + \ldots + \sum_{k=1}^{k_f} \tilde{\mathbf{V}}_{\mathbf{f}_k}^{\mathrm{T}} \hat{\mathbf{V}}_{\mathbf{f}k} \tag{52}$$

$$\langle \tilde{\mathbf{Z}}, \hat{\mathbf{Z}} \rangle = \langle \tilde{\mathbf{V}}, \hat{\mathbf{V}} \rangle + \langle \tilde{\mathbf{W}}, \hat{\mathbf{W}} \rangle. \tag{53}$$

where $\hat{\mathbf{Z}} = \mathbf{Z} - \tilde{\mathbf{Z}}$, and therefore

$$\langle \tilde{\mathbf{Z}}, \hat{\mathbf{Z}} \rangle = \langle \tilde{\mathbf{Z}}, \mathbf{Z} \rangle - \|\tilde{\mathbf{Z}}\|^2 \leq \|\tilde{\mathbf{Z}}\|\|\mathbf{Z}\| - \|\tilde{\mathbf{Z}}\|^2 \leq \|\tilde{\mathbf{Z}}\| Z_M - \|\tilde{\mathbf{Z}}\|^2. \tag{54}$$

Due to (8) and the assumption $\mathbf{w} \leq w_M$, it is possible to show that:

$$\dot{V}(\mathbf{r}, \tilde{\mathbf{Z}}) \leq - \|\mathbf{r}\| \left[K_{v,m}\|\mathbf{r}\| - w_M + \kappa(\|\tilde{\mathbf{Z}}\| - \frac{Z_M}{2})^2 - \frac{\kappa Z_M^2}{4}\right] \tag{55}$$

Hence, if

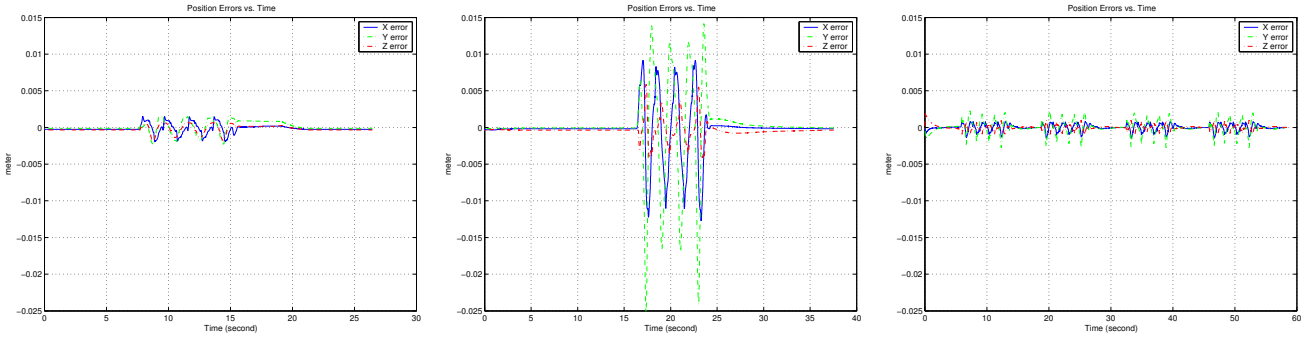
$$\|\mathbf{r}\| > \frac{w_M + \kappa \mathbf{Z}_M^2/4}{K_{v,m}} \equiv b_{\mathbf{r}}, \quad \text{or} \tag{56}$$

$$\|\tilde{\mathbf{Z}}\| > \sqrt{\frac{w_M}{\kappa} + \frac{Z_M^2}{4}} + \frac{Z_M}{2} \equiv b_{\tilde{\mathbf{Z}}} \tag{57}$$

then the terms in the square bracket of the right hand side of (55) evaluate to a positive number, and $\dot{V}(\mathbf{r}, \tilde{\mathbf{Z}}) \leq -\beta\|\mathbf{r}\| < 0$. According to Lyapunov's extension theorem LaSalle [1960] this demonstrates:

$$\lim_{t\to\infty} \|\mathbf{r}(t)\| = b_{\mathbf{r}}, \quad \lim_{t\to\infty} \|\tilde{\mathbf{Z}}(t)\| = b_{\tilde{\mathbf{Z}}}. \tag{58}$$

This shows that $\mathbf{r}$ and $\tilde{\mathbf{W}}, \tilde{\mathbf{V}}$ are bounded. It can be shown through classical control that a bounded reference signal $\mathbf{r}$ (15) yields error signals $\lim_{t\to\infty} \mathbf{e} = 0$ and $\dot{\mathbf{e}}$, $\int_0^{\tau=t} \mathbf{e}\, \mathrm{d}\tau$ that are also bounded.



(a) (b) (c)

Fig. 1. End-effector tracking error with respect to the reference trajectory for (a) inverse dynamics control (b) PD control with gravity compensation and (c) NN adaptive control strategies - obtained through simulation.

4. PERFORMANCE EVALUATION

The proposed strategy is studied through simulation and real-time implementation on a PUMA 560 robot. In addition to the proposed NN adaptive motion control proposed in this paper, two other types of control strategies are performed for comparison: (i) the model-based inverse dynamics motion controller in Operational Space Formulation [Khatib, 1987] – without friction compensation and (ii) Proportional-plus-Derivative (PD) control with gravity model compensation. A periodic circular trajectory for end-effector position with a constant orientation was set as the desired path for all cases (simulation and real-time implementation) of the experimentations.

4.1 Robot Simulation

A simulation study was carried out for the proposed controller. A dynamic model of a 6 DOF manipulator of PUMA 560 was utilized. To give a more realistic situation, the robot simulator includes joint friction dynamics in addition to the inertia, Coriolis and centrifugal and gravity terms. The performance of the inverse dynamics (Operational Space Formulation) and the PD with gravity compensation controllers are shown in Fig. 1(a) and Fig. 1(b), respectively. Note again that gravity is compensated through dynamic modelling in the PD controller (Fig.1(b)).

The performance of the proposed NN based adaptive control is shown in Fig. 1(c). Without any prior knowledge of the robot dynamics, the controller was initialized with zero weights. The proposed NN adaptive control was shown to effectively learn and reduce the tracking errors. Table 1 shows that the proposed NN control strategy yields comparable performance to inverse dynamics strategy, without prior knowledge of the robot dynamics. It should be noted that the inverse dynamic controller does not include the friction model as generally it is difficult to obtain the coefficients of friction in implementation as friction varies with many factors, such as temperature, presence of dust and dirt, humidity, etc. However, the friction model is included in the proposed NN adaptive controller, as described in the derivation. In the implementation, the coefficients of friction are determined online with NN adaptive method, hence identification is not an issue. Figure 2 also shows the boundedness and stability of the norms of NN weights.

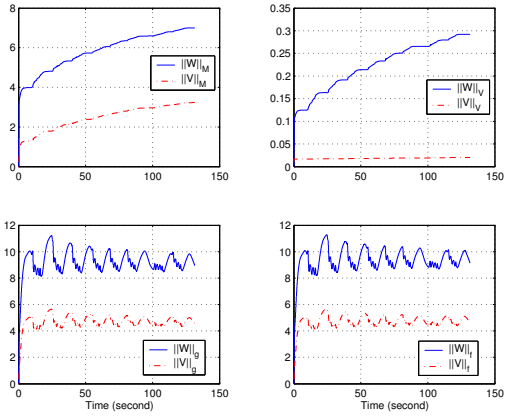


Fig. 2. Time history of the NN weights - simulation.

4.2 Real-time Robot Implementation

The strategy was implemented real-time on a PUMA 560 which does not provide joint velocity feedback. The joint velocities $\dot{\mathbf{q}}$ are obtained by employing backward difference algorithm of joint positions $\mathbf{q}$ in conjunction with low pass filter. Hence, only the estimated operational space velocities $\dot{\hat{\mathbf{x}}}$ are available, using $\dot{\hat{\mathbf{x}}} = \mathbf{J}\dot{\hat{\mathbf{q}}}$. Figure 3 shows that the performance of NN controller fails to produce similar performance as in the simulation, although the NN weights still converge to a stable set of values. Figure 4 shows the boundedness and stability of the norms of NN weights. It is found that in real-time, the values of velocity error gain $\mathbf{K_v}$ in the control law that can be selected is lower than that in simulation, due to the quality of the velocity feedback signal that can be obtained.

	Inverse-dynamics	PD + gravity	NN controller
X_{error}	1.3	12.5	0.7
Y_{error}	1.7	25	2.2
Z_{error}	1.5	6	1.3

Table 1. Comparison of maximum position error - simulation

	Inverse-dynamics	PD + gravity	NN controller
X_{error}	2.5	12.5	10
Y_{error}	2.5	20.0	17
Z_{error}	2.5	7.0	7.5

Table 2. Comparison of maximum position error - real-time

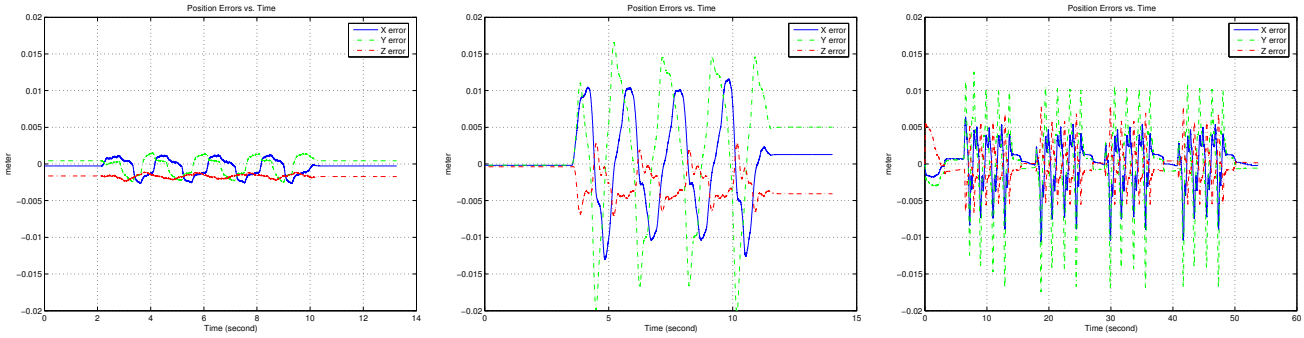


Fig. 3. End-effector tracking error with respect to the reference trajectory for (a) inverse dynamics control (b) PD control with gravity compensation and (c) NN adaptive control strategies - obtained through real-time implementation.

5. CONCLUSIONS AND FUTURE WORKS

At this point, it is possible to conclude that it is feasible to construct an NN adaptive controller with a potential performance comparable to that of inverse-dynamics strategy, without any prior knowledge of the system dynamics. However it does highlight the problem in real robot implementation where joint velocity feedback is not provided. The quality of the velocity signal feedback affects the performance of the adaptive NN controller, which can be further improved, through several strategies such as observer controller.

REFERENCES

B. Armstrong, O. Khatib, and J. Burdick. The explicit dynamic model and inertial parameters of the puma 560 arm. pages 510–518, San Francisco, CA, April7–10, 1986.

J.J. Craig, P. Hsu, and S.S. Sastry. Adaptive control of mechanical manipulators. pages 190–195, San Francisco, CA, April7–10, 1986.

S.S. Ge, C.C. Hang, and L.C. Woon. Adaptive neural network control of robot manipulators in task space. 44 (6):746–752, December 1997.

S. Hu, M.H. Ang, Jr, and H. Krishnan. Neural network controller for constrained robot manipulators. pages 1906–1911, San Francisco, CA, May 2000.

R.S. Jamisola, M.H. Ang, Jr., T.M. Lim, O. Khatib, and S.Y. Lim. Dynamics identification and control of an industrial robot. In *Proc. 9th Int. Conf. Adv. Robot.*, pages 323–328, Tokyo, Japan, October25–27, 1999.

R.S. Jamisola, D. Oetomo, M. H. Ang, Jr., O. Khatib T.M. Lim, , and S.Y. Lim. Compliant motion using mobile manipulator: An operationalspace formulation approach to aircraft canopy polishing. *Adv. Robot.*, 19(5):613–634, Jun 2005.

O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. RA-3(1):43–53, February 1987.

O. Khatib and J. Burdick. Motion and force control of robot manipulators. pages 1381–1386, San Francisco, CA, April7 – 10, 1986.

C.M. Kwan, A. Yeşildirek, and F.L. Lewis. Robust force/motion control of constrained robots using neural network. In *Proc. IEEE Conf. Decision and Control*, pages 1862–1867, Lake Buena Vista, Florida, December 1994.

J.P. LaSalle. Some extensions of Liapunov's second method. *IEE Trans. Circuit Theory*, pages 520–527, 1960.

F.L. Lewis, C.T. Abdallah, and D.M. Dawson. *Control of Robot Manipulators*. MacMillan, NY, 1993.

F.L. Lewis, A. Yesildirek, , and K. Liu. Multilayer neural-net robot controller with guaranteed tracking performance. 7:388–399, March 1996.

N.H. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. 33(5):419–426, May 1988.

R.H. Middleton and G.C. Goodwin. Adaptive computed torque control for rigid link manipulations. *Syst. Contr. Lett.*, 10:9–16, 1988.

R. Ortega and M. Spong. Adaptive motion control of rigid robots: A tutorial. In *Proc. IEEE Conf. Decision and Control*, pages 1576–1584, Austin, TX, December 1988.

J. Russakow, O. Khatib, and S. Rock. Extended operational space formulation for serial-to-parallel chain (branching) manipulators. In *Proc. IEEE Int. Conf. Robot. Automat.*, volume 1, pages 1056–1061, Nagoya, Japan, May 1995.

J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall, 1991.

J.-J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *Int. J. Robot. Res.*, 6(3):49–59, 1987.

Q.H. Xia, S.Y. Lim, , M.H. Ang, Jr, and T.M. Lim. Adaptive joint friction compensation using a model-based operational space velocity observer. pages 3081–3086, New Orleans, LA, April26 – May1 2004.

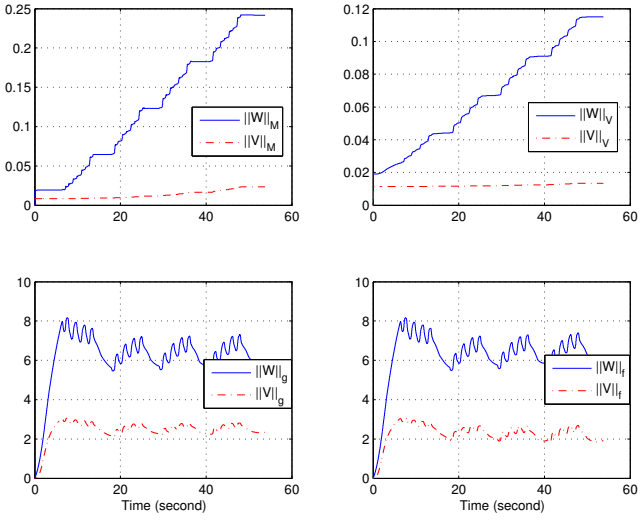


Fig. 4. Time history of the NN weights - real-time.