2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

Sliding Mode Adaptive Neural-network Control for Nonholonomic Mobile Modular Manipulators

Yangmin Li and Yugang Liu
Department of Electromechanical Engineering, Faculty of Science and Technology,
University of Macau, Av. Padre Toms Pereira S.J., Taipa, Macao S.A.R., P.R.China
ymli@umac.mo, ya27401@umac.mo

Abstract

A general mobile modular manipulator can be defined as a *m*-wheeled holonomic/nonholonomic mobile platform combining with a *n*-degree of freedom modular manipulator. This paper presents a sliding mode adaptive neural-network controller for trajectory following of nonholonomic mobile modular manipulators in task space. Dynamic model for the entire mobile modular manipulator is established in consideration of nonholonomic constraints. Multilayered perceptrons (MLP) are used as estimators to approximate the dynamic model of the mobile modular manipulator. Sliding mode control and direct adaptive technique are combined together to suppress bounded disturbances and modeling errors caused by parameter uncertainties. Simulations are performed to demonstrate that the established models are valid and the control method is effective.

Keywords: sliding mode control, adaptive control, neural network, nonholonomic mobile modular manipulator

1 Introduction

Modular manipulators have been attracting more and more researchers because of their reconfigurability and adaptability. Traditional modular manipulators mounted on a fixed base whose mobility is limited. In this paper, we investigate an assemble structure through attaching a mobile platform to a modular manipulator base in order to extend its workspace. Since there exist interactions between the modular manipulator and the mobile platform as well as nonholonomic constraints of the mobile platform, the modeling problem becomes extremely complex. At the same time, a trajectory following task becomes difficult to accomplish for this kind of robot.

Regarding related research work, parameters identification for modular manipulators were investigated in [1]. The dynamic characteristics between the mobile platform and the manipulator were studied [2]. A neural-network based methodology was developed for the motion control of mobile manipulators [3]. Regarding to research work on adaptive control and sliding mode control, adaptive technique and sliding mode control were combined together [4]. A sliding mode controller is presented for trajectory tracking control of nonholonomic wheeled mobile robots [5].

In the previous research, the mobile platform and the manipulator modeling was usually treated separately and nonholonomic mobile robotic control was mostly limited to kinematic velocity control, few work on dynamic torque control. In this approach, the entire mobile modular manipulator is modeled as an integrated structure. Furthermore, most of the neural-network controllers were designed in joint space, but few in task space. However, in practical applications, the task is usually specified to the end-effector in world space. Up to now, neural network estimator, adaptive technique, and sliding mode controllers were mostly used for control of manipulators, few work are found on nonholonomic mobile manipulators. A mobile modular manipulator is a complex high nonlinear system, whose dynamic parameters are difficult to forecast precisely. In fact, it is almost impossible to obtain exact dynamic models because of such uncertainties as nonlinear frictions and flexibilities of the joints and links of robot. For mobile modular manipulators, not only interactions between the mobile platform and its atop manipulator but also unknown disturbances caused by terrain irregularity make the control task difficult to achieve. Sliding mode control provides a natural rejection to external disturbances by a high-frequency commuted control action that constrains the error trajectories to stay on the sliding subspace. Adaptive control technique does not rely on precise prior knowledge of dynamic parameters, and it can suppress such errors caused by parameter uncertainties effectively. Furthermore, it can counteract the negative influence of high-frequency switching caused by sliding mode controller.

A dynamic model is established for the entire robot. A sliding mode adaptive neural-network controller is designed in task space to control the end-effector following a desired trajectory. Simulations are carried out to verify the effectiveness of the modeling and controller design method. Finally, some conclusions are presented.

2 Dynamic Modeling

2.1 Nonholonomic Constraints

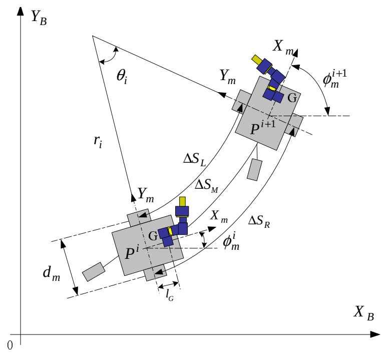

Figure 1: platform-motion.

In this paper, we analyze a three-wheeled nonholonomic mobile platform, which has two driving wheels and one castor wheel. Assume that the robot just move on a horizontal plane, then the motion of the mobile platform on the plane can be described as shown in Fig. 1, from which we can easily obtain

$$\begin{cases} \dot{x}_m = \frac{r_f \cdot C_m}{2}\left(\dot{\phi}_R + \dot{\phi}_L\right) \\ \dot{y}_m = \frac{r_f \cdot S_m}{2}\left(\dot{\phi}_R + \dot{\phi}_L\right) \\ \dot{\phi}_m = \frac{r_f}{d_m} \cdot \left(\dot{\phi}_R - \dot{\phi}_L\right) \end{cases} \tag{1}$$

Where $S_m = \sin(\phi_m)$,$C_m = \cos(\phi_m)$; r_f is the radius of the driving wheel; ϕ_L and ϕ_R are rotating angles of the left driving wheel and the right driving wheel respectively.

Let the general coordinates of the mobile platform be $\xi = \begin{bmatrix} x_m & y_m & \phi_m & \phi_L & \phi_R \end{bmatrix}^T$. From Eq. (1) we can obtain

$$\dot{\xi} = \begin{bmatrix} r_f \cdot C_m/2 & r_f \cdot C_m/2 \\ r_f \cdot S_m/2 & r_f \cdot S_m/2 \\ -r_f/d_m & r_f/d_m \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_L \\ \dot{\phi}_R \end{bmatrix} \tag{2}$$

Eq.(2) can be written as $\dot{\xi} = S(\xi) \cdot \begin{bmatrix} \dot{\phi}_L & \dot{\phi}_R \end{bmatrix}^T$.

From Eq. (1) and Fig. 1, we can obtain

$$\begin{bmatrix} C_m & S_m & -d_m/2 & -r_f & 0 \\ C_m & S_m & d_m/2 & 0 & -r_f \\ S_m & -C_m & 0 & 0 & 0 \end{bmatrix} \cdot \dot{\xi} = 0 \tag{3}$$

Eq.(3) can be written as $A(\xi) \cdot \dot{\xi} = 0$.

Equation (3) is just the nonholonomic constraint of the mobile platform. Substituting Eq. (2) into (3) yields

$$A(\xi) \cdot S(\xi) = 0 \tag{4}$$

Taking the mobile platform as a special module attached at the bottom of the modular manipulator, and according to Denavit-Hartenberg notation, we can obtain transformation matrix of the end-effector with respect to its inertial base frame ${}^B_E T$. Furthermore, we can obtain the position vectors and the posture vectors of the end-effector with respect to frame B. Then, Z-Y-Z Euler angles can be obtained [6].

Define $q = \begin{bmatrix} \phi_L & \phi_R & q_1 & \cdots & q_N \end{bmatrix}^T \in \Re^{(N+2)}$,
$x = \begin{bmatrix} p_x & p_y & p_z & \phi & \theta & \psi \end{bmatrix}^T \in \Re^6$,
$\zeta = \begin{bmatrix} \xi^T & q_1 & \cdots & q_N \end{bmatrix}^T \in \Re^{(N+5)}$,
then the Jacobian matrix can be derived as follows

$$J = \frac{\partial x}{\partial \zeta} \cdot \begin{bmatrix} S(\xi) & 0_{5\times N} \\ 0_{N\times 2} & I_{N\times N} \end{bmatrix} \tag{5}$$

2.2 Dynamic Modeling

Supposed that the mobile platform moves on a horizontal plane, it has a constant potential energy U_m. Then, the Lagrange function can be derived by

$$\begin{aligned} L &= \tfrac{1}{2} \cdot m_c \cdot \dot{x}_m^2 + \tfrac{1}{2} \cdot m_c \cdot \dot{y}_m^2 + \tfrac{1}{2} \cdot I_c \cdot \dot{\phi}_m^2 - U_m \\ &+ \tfrac{1}{2} \cdot I_f \cdot \dot{\phi}_L^2 + \tfrac{1}{2} \cdot I_f \cdot \dot{\phi}_R^2 - \sum_{i=0}^{N} \left(m_i \cdot \vec{g}^T \cdot \vec{p}_{ci}\right) \\ &+ \sum_{i=0}^{N} \left(\tfrac{1}{2} \cdot \vec{v}_{ci}^T \cdot m_i \cdot \vec{v}_{ci} + \tfrac{1}{2} \cdot \vec{\omega}_i^T \cdot I_i \cdot \vec{\omega}_i\right) \end{aligned} \tag{6}$$

Where m_c is the mass of the cart including all driving units in the box; I_c is the inertial moment around the vertical axis; I_f is the inertial moment of the fixed wheel; $\vec{g} = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}^T$ is a gravity acceleration vector; $\vec{p}_{ci}$ and $\vec{v}_{ci}$ are the coordinate vectors and linear velocity vectors of the mass center of the ith module with respect to frame i respectively; ,$\vec{\omega}_i$, m_i ,and I_i are angular velocity vectors, the masses,and the inertial moments accordingly.

The constrained dynamics for the nonholonomic mobile modular manipulator can be determined as follows

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\zeta}}\right)^T - \left(\frac{\partial L}{\partial \zeta}\right)^T = B(\zeta)\tau + C(\zeta)\lambda \tag{7}$$

Where $B(\zeta) = \begin{bmatrix} 0_{3\times(N+2)} \\ I_{(N+2)\times(N+2)} \end{bmatrix} \in \Re^{(N+5)\times(N+2)}$, $\tau = \begin{bmatrix} \tau_L & \tau_R & \tau_1 & \cdots & \tau_N \end{bmatrix}^T \in \Re^{(N+2)}$, $C(\zeta) = \begin{bmatrix} A^T(\xi) \\ 0_{N\times 3} \end{bmatrix} \in \Re^{(N+5)\times 3}$, $\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix}^T \in \Re^3$. $\lambda_i = \lambda_i(\zeta)(i = 1,2,3)$ are Lagrange multipliers associated with the constraints; τ_L, τ_R, and τ_i are the driving torques of the left wheel, the right wheel and the ith joint respectively.

Separating the 1st order index and the 2nd order index from Eq. (7) yields

$$M(\zeta) \cdot \ddot{\zeta} + V\left(\zeta, \dot{\zeta}\right) \cdot \dot{\zeta} + G(\zeta) = B(\zeta) \cdot \tau + C(\zeta) \cdot \lambda \tag{8}$$

Where $M(\zeta) \in \Re^{(N+5)\times(N+5)}$ is the inertia matrix; $V\left(\zeta,\dot{\zeta}\right) \in \Re^{(N+5)\times(N+5)}$ is the centripetal and coriolis matrix; $G(\zeta) \in \Re^{(N+5)}$ denotes the gravitational force.

From Eq. (2), we can obtain

$$\dot{\zeta} = \bar{S}(\zeta)\cdot \dot{q} \tag{9}$$

Where $\bar{S}(\zeta) = \begin{bmatrix} S(\xi) & 0_{5\times N} \\ 0_{N\times 2} & I_{N\times N} \end{bmatrix} \in \Re^{(N+5)\times(N+2)}$.

Differentiating Eq. (9) yields

$$\ddot{\zeta} = \dot{\bar{S}}(\zeta)\cdot \dot{q} + \bar{S}(\zeta)\cdot \ddot{q} \tag{10}$$

Substituting Eq. (10) into (8), and left multiplying $\bar{S}^T(\zeta)$, yields

$$\bar{M}(\zeta)\cdot \ddot{q} + \bar{V}\left(\zeta,\dot{\zeta}\right)\cdot \dot{q} + \bar{G}(\zeta) = \bar{\tau} \tag{11}$$

Where $\bar{M}(\zeta) = \bar{S}^T(\zeta)\cdot M(\zeta)\cdot \bar{S}(\zeta)$, $\bar{V}\left(\zeta,\dot{\zeta}\right) = \bar{S}^T(\zeta)\cdot \left[V\left(\zeta,\dot{\zeta}\right)\cdot \bar{S}(\zeta) + M(\zeta)\cdot \dot{\bar{S}}(\zeta)\right]$, $\bar{G}(\zeta) = \bar{S}^T(\zeta)\cdot G(\zeta)$, $\bar{\tau}(\zeta) = \bar{S}^T(\zeta)\cdot B(\zeta)\cdot \tau$, and the term $\bar{S}(\zeta)^T \cdot C(\zeta) = 0_{(N+2)\times 3}$ disappears.

Equation (11) can be expressed in task space as

$$\tilde{M}(\zeta)\cdot \ddot{x} + \tilde{V}\left(\zeta,\dot{\zeta}\right)\cdot \dot{x} + \tilde{G}(\zeta) = \tilde{\tau} \tag{12}$$

Where $\tilde{M}(\zeta) = J^{-T}\cdot \bar{M}(\zeta)\cdot J^{-1}$, $\tilde{V}\left(\zeta,\dot{\zeta}\right) = J^{-T}\cdot \left[\bar{V}\left(\zeta,\dot{\zeta}\right) - \bar{M}(\zeta)\cdot J^{-1}\cdot \dot{J}\right]\cdot J^{-1}$, $\tilde{G}(\zeta) = J^{-T}\cdot \bar{G}(\zeta)$, $\tilde{\tau} = J^{-T}\cdot \bar{\tau}$.

Properties:

1) $\tilde{M}(\zeta)$ is positive and symmetric, i.e.

$$\tilde{M}^T(\zeta) = \tilde{M}(\zeta) > 0 \tag{13}$$

2) Matrix $\tilde{N}\left(\zeta,\dot{\zeta}\right) = \dot{\tilde{M}}(\zeta) - 2\tilde{V}\left(\zeta,\dot{\zeta}\right)$ is skew-symmetric, i.e.

$$\tilde{N}^T\left(\zeta,\dot{\zeta}\right) = -\tilde{N}\left(\zeta,\dot{\zeta}\right) \tag{14}$$

3) $\tilde{M}(\zeta)$, $\tilde{V}\left(\zeta,\dot{\zeta}\right)$, $\tilde{G}(\zeta)$, and $\tilde{\tau}$ are all bounded as long as J is nonsingular.

3 Sliding Mode Adaptive Neural-network Controller Design

3.1 Neural-network Modeling

It is verified that a multilayer perceptron trained with the back-propagation algorithm can approximate any continuous multivariate functions to any desired degree of accuracy, provided that sufficient hidden neurons are available [7]. To ensure rapid convergence, single-output multilayer perceptrons with only one hidden layer are used in this paper, see Fig. 2.

In Fig. 2, the output of the MLP is as follows

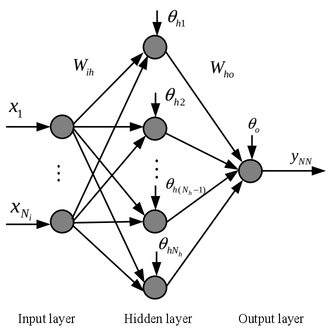

Figure 2: A single-output MLP with one hidden layer.

$$y_{NN} = \sum_{j=1}^{N_h}\left[\varphi\left(\sum_{i=1}^{N_i} x_i\cdot w_{ihij} + \theta_{hj}\right)\cdot w_{hoj}\right] + \theta_o \tag{15}$$

Where $\varphi(\circ)$ is the activation function, which is hyperbolic tangent function in this paper; N_i and N_h represent the number of input-layer and hidden-layer neurons respectively; x_i is the ith input to the neural network; w_{ihij} represents the weight connecting the ith input-layer neuron and the jth hidden-layer neuron; θ_{hj} denotes the threshold of the jth hidden-layer neuron; w_{hoj} is the weight connecting the jth hidden-layer neuron and the output neuron; θ_o is threshold of the output neuron.

Let y_d represent the desired output to be approximated. Then, the instantaneous squared errors is

$$\varepsilon = \frac{1}{2}\cdot \left(y_d - y_{NN}\right)^2 \tag{16}$$

BP algorithm corrects the weights of MLP neural networks using steepest descent method. To increase the rate of convergence and yet avoid the danger of instability, the generalized delta rule is introduced

$$\begin{aligned} w_{ihij}(k+1) &= w_{ihij}(k) + \alpha\Delta w_{ihij}(k) + \eta\frac{\partial\varepsilon}{\partial w_{ihij}} \\ w_{hoj}(k+1) &= w_{hoj}(k) + \alpha\Delta w_{hoj}(k) + \eta\frac{\partial\varepsilon}{\partial w_{hoj}} \\ \theta_{hj}(k+1) &= \theta_{hj}(k) + \alpha\Delta\theta_{hj}(k) + \eta\frac{\partial\varepsilon}{\partial\theta_{hj}} \\ \theta_o(k+1) &= \theta_o(k) + \alpha\Delta\theta_o(k) + \eta\frac{\partial\varepsilon}{\partial\theta_o} \end{aligned} \tag{17}$$

Where $\alpha \in (-1,1)$ is the momentum constant; η represents the learning-rate parameter; k denotes the number of iterations; $\Delta(\circ)(k) = (\circ)(k) - (\circ)(k-1)$; The local gradient can be calculated as follows

$$\begin{aligned} \frac{\partial\varepsilon}{\partial\theta_o} &= \frac{\partial\varepsilon}{\partial e}\cdot\frac{\partial e}{\partial y_{NN}}\cdot\frac{\partial y_{NN}}{\partial\theta_o} = -e \\ \frac{\partial\varepsilon}{\partial w_{hoj}} &= \frac{\partial\varepsilon}{\partial e}\cdot\frac{\partial e}{\partial y_{NN}}\cdot\frac{\partial y_{NN}}{\partial w_{hoj}} = -e\cdot\varphi(\circ) \\ \frac{\partial\varepsilon}{\partial\theta_{hj}} &= -e\cdot w_{hoj}\cdot\frac{b}{2a}\cdot[a+\varphi(\circ)]\cdot[a-\varphi(\circ)] \\ \frac{\partial\varepsilon}{\partial w_{ihij}} &= -e\cdot w_{hoj}\cdot\frac{b}{2a}\cdot[a+\varphi(\circ)]\cdot[a-\varphi(\circ)]\cdot x_i \end{aligned} \tag{18}$$

From **propertiey 3** listed above, $\tilde{M}(\zeta), \tilde{V}\left(\zeta,\dot{\zeta}\right)$, and $\tilde{G}(\zeta)$ can be bounded as long as the matrix J is nonsingular. Therefore their elements can all be approximated by multilayer perceptrons [8]. From Eq. (12), we can see that $\tilde{M}(\zeta)$, and $\tilde{G}(\zeta)$ are functions of ζ

only, but $\tilde{V}\left(\zeta,\dot{\zeta}\right)$ is function of both ζ and $\dot{\zeta}$. So, neural networks used to approximate $\tilde{M}(\zeta)$, and $\tilde{G}(\zeta)$ can have much less hidden-layer neurons than those for $\tilde{V}\left(\zeta,\dot{\zeta}\right)$. Moreover, from the nonholonomic velocity constraint of the mobile platform, inputs of the neural networks used to approximate $\tilde{V}\left(\zeta,\dot{\zeta}\right)$ can be selected as $\chi=\left[\begin{array}{cc}\zeta & \dot{q}\end{array}\right]^T$, then

$$\begin{array}{l}\tilde{M}(\zeta)=M_{NN}(\zeta)+E_M(\zeta)\\ \tilde{G}(\zeta)=G_{NN}(\zeta)+E_G(\zeta)\\ \tilde{V}\left(\zeta,\dot{\zeta}\right)=V_{NN}(\chi)+E_V(\chi)\end{array}\quad(19)$$

Where $M_{NN}(\zeta)=\{m_{NNij}\}\in\Re^{6\times6}$, $G_{NN}(\zeta)=\{g_{NNi}\}\in\Re^{6}$, and $V_{NN}(\chi)=\{v_{NNij}\}\in\Re^{6\times6}$ are MLP neural-network approximations of $\tilde{M}(\zeta)$, $\tilde{G}(\zeta)$, and $\tilde{V}\left(\zeta,\dot{\zeta}\right)$ respectively. $E_M(\zeta)=\{\varepsilon_{Mij}\}\in\Re^{6\times6}$, and $E_G(\zeta)=\{\varepsilon_{Gi}\}\in\Re^{6}$, and $E_V(\chi)=\{\varepsilon_{Vij}\}\in\Re^{6\times6}$ are corresponding error matrices.

3.2 Controller Design

Assume the desired trajectory, velocity, and acceleration in the task space be $x_d(t)$, $\dot{x}_d(t)$, and $\ddot{x}_d(t)$. The goal of the sliding mode adaptive NN controller is to make $x(t)$ track $x_d(t)$ as close as possible with significant uncertainty. To avoid measuring accelerations, the error system can be defined as follows

$$\begin{array}{l}e(t)=x(t)-x_d(t)\\ \dot{x}_\gamma(t)=\dot{x}_d(t)-\Lambda\cdot e(t)\\ \gamma(t)=\dot{x}(t)-\dot{x}_\gamma(t)=\dot{e}(t)+\Lambda\cdot e(t)\end{array}\quad(20)$$

Where $\gamma(t)$ is the tracking error measure; $\Lambda\in\Re^{6\times6}$ is a constant positive definite matrix.

From Eq. (20), we can obtain

$$\begin{array}{l}\dot{x}(t)=\gamma(t)+\dot{x}_\gamma(t)\\ \ddot{x}_\gamma(t)=\ddot{x}_d(t)-\Lambda\cdot\dot{e}(t)\\ \ddot{x}(t)=\dot{\gamma}(t)+\ddot{x}_\gamma(t)\end{array}\quad(21)$$

Substituting Eq. (21) into (12) and considering Eq. (19) at the same time, yields

$$\begin{array}{l}\tilde{M}(\zeta)\cdot\dot{\gamma}(t)+\tilde{V}(\chi)\cdot\gamma(t)+M_{NN}(\zeta)\cdot\ddot{x}_\gamma(t)\\ +V_{NN}(\chi)\cdot\dot{x}_\gamma(t)+G_{NN}(\zeta)+E(\chi)=\tilde{\tau}\end{array}\quad(22)$$

Where $E(\chi)=E_M(\zeta)\cdot\ddot{x}_\gamma(t)+E_V(\chi)\cdot\dot{x}_\gamma(t)+E_G(\zeta)$.

The sliding mode adaptive NN controller presented in this paper is given by Eq. (23), and a control system diagram is shown in Fig. 3.

$$\begin{array}{rl}\tau &=\left[\bar{S}^T(\zeta)\cdot B(\zeta)\right]^{-1}\cdot J^T\cdot\left[\hat{M}_{NN}(\zeta)\cdot\ddot{x}_\gamma(t)\right.\\ &+\hat{V}_{NN}(\chi)\cdot\dot{x}_\gamma(t)+\hat{G}_{NN}(\zeta)-K_I\cdot\int_0^t\gamma(t)\,dt\\ &\left.-K_P\cdot\gamma(t)-K_M\cdot\ddot{x}_\gamma(t)-K_V\cdot\dot{x}_\gamma(t)-K_G\right]\end{array}\quad(23)$$

Where $\hat{M}_{NN}=\{\hat{m}_{NNij}\}$, $\hat{V}_{NN}=\{\hat{v}_{NNij}\}$, $\hat{G}_{NN}=\{\hat{g}_{NNi}\}$ are estimators of M_{NN}, V_{NN}, and G_{NN} respectively and these three terms construct an adaptive model-based controller; $K_P\in\Re^{6\times6}$, $K_I\in\Re^{6\times6}$ are the proportional gain matrices and the integral gain matrices respectively and these two terms form a PID controller; The last three terms form a sliding mode controller with their parameters selected as follows:

$$\begin{array}{l}K_M=\{k_{Mij}\cdot sgn\left(\gamma_i\cdot\ddot{x}_{\gamma j}\right)\},k_{Mij}>|\varepsilon_{Mij}|\\ K_V=\{k_{Vij}\cdot sgn\left(\gamma_i\cdot\dot{x}_{\gamma j}\right)\},k_{Vij}>|\varepsilon_{Vij}|\\ K_G=\{k_{Gi}\cdot sgn\left(\gamma_i\right)\},k_{Gi}>|\varepsilon_{Gi}|\end{array}\quad(24)$$

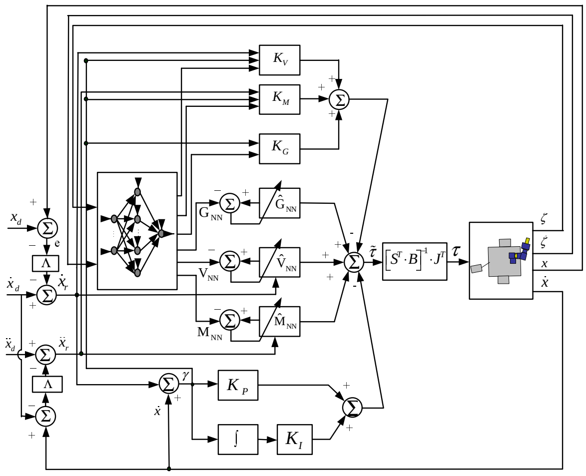

Figure 3: A sliding mode adaptive NN controller.

Substituting Eq. (23) into (22), and considering Eqs. (11-12) at the same time, yields

$$\begin{array}{l}\tilde{M}(\zeta)\cdot\dot{\gamma}(t)+\tilde{V}(\chi)\cdot\gamma(t)+K_I\cdot\left[\int_0^t\gamma(t)\,dt\right]\\ +K_P\cdot\gamma(t)+K_M\cdot\ddot{x}_\gamma(t)+K_V\cdot\dot{x}_\gamma(t)+K_G+E(\chi)\\ =\breve{M}_{NN}(\zeta)\cdot\ddot{x}_\gamma(t)+\breve{V}_{NN}(\chi)\cdot\dot{x}_\gamma(t)+\breve{G}_{NN}(\zeta)\end{array}\quad(25)$$

Where

$$\begin{array}{l}\breve{M}_{NN}(\zeta)=\hat{M}_{NN}(\zeta)-M_{NN}(\zeta)\\ \breve{V}_{NN}(\chi)=\hat{V}_{NN}(\chi)-V_{NN}(\chi)\\ \breve{G}_{NN}(\zeta)=\hat{G}_{NN}(\zeta)-G_{NN}(\zeta).\end{array}\quad(26)$$

Theorem: If $K_P>0$ and $K_I>0$, the closed-loop system described by Eq. (25) is asymptotically stable under the following parameter adaptation laws given by Eq. (27). The error signals are convergent along with time, i.e., $e(t),\dot{e}(t)\to0$ as $t\to\infty$. Furthermore, the signals in this system are all bounded.

$$\begin{array}{l}\dot{\hat{m}}_{NNij}=-\Gamma_{Mij}\cdot\left(\gamma_i\cdot\ddot{x}_{\gamma j}+\sigma_{Mij}\cdot|\gamma_i|\cdot\breve{m}_{NNij}\right)\\ \dot{\hat{v}}_{NNij}=-\Gamma_{Vij}\cdot\left(\gamma_i\cdot\dot{x}_{\gamma j}+\sigma_{Vij}\cdot|\gamma_i|\cdot\breve{v}_{NNij}\right)\\ \dot{\hat{g}}_{NNi}=-\Gamma_{Gi}\cdot\left(\gamma_i+\sigma_{Gi}\cdot|\gamma_i|\cdot\breve{g}_{NNi}\right)\end{array}\quad(27)$$

Where $\Gamma_{Mij},\Gamma_{Vij},\Gamma_{Gi},\sigma_{Mij},\sigma_{Vij}$, and σ_{Gi} are all positive constants.

Proof: Considering the following Lyapunov candidate

$$\begin{array}{rl}V_S&=\frac{1}{2}\cdot\left[\int_0^t\gamma(t)\,dt\right]^T\cdot K_I\cdot\left[\int_0^t\gamma(t)\,dt\right]\\ &+\frac{1}{2}\cdot\sum_{j=1}^{6}\sum_{i=1}^{6}\left(\frac{\breve{m}^2_{NNij}}{\Gamma_{Mij}}\right)+\frac{1}{2}\cdot\sum_{j=1}^{6}\sum_{i=1}^{6}\left(\frac{\breve{v}^2_{NNij}}{\Gamma_{Vij}}\right)\\ &+\frac{1}{2}\cdot\sum_{i=1}^{6}\left(\frac{\breve{g}^2_{NNi}}{\Gamma_{Gi}}\right)+\frac{1}{2}\cdot\gamma^T(t)\cdot\tilde{M}(\zeta)\cdot\gamma(t)>0\end{array}\quad(28)$$

The time derivative of Lyapunov candidate is

$$\begin{aligned}\dot{V}_S &= \gamma^T(t)\cdot\left\{\tilde{M}(\zeta)\cdot\dot{\gamma}(t)+K_I\cdot\left[\int_0^t\gamma(t)\,dt\right]\right\}\\ &+\sum_{j=1}^{6}\sum_{i=1}^{6}\left(\frac{\breve{m}_{NNij}}{\Gamma_{Mij}}\cdot\dot{\breve{m}}_{NNij}+\frac{\breve{v}_{NNij}}{\Gamma_{Vij}}\cdot\dot{\breve{v}}_{NNij}\right)\\ &+\sum_{i=1}^{6}\left(\frac{\breve{g}_{NNi}}{\Gamma_{Gi}}\cdot\dot{\breve{g}}_{NNi}\right)+\gamma^T(t)\cdot\frac{\dot{\tilde{M}}(\zeta)}{2}\cdot\gamma(t)\end{aligned} \tag{29}$$

Substituting Eq. (25) into (29) and considering Eqs. (14) and (24) at the same time, yields

$$\begin{aligned}\dot{V}_S &\le \sum_{j=1}^{6}\sum_{i=1}^{6}\left\{\breve{m}_{NNij}\cdot\left[\gamma_i(t)\cdot\ddot{x}_{\gamma j}(t)+\frac{\dot{\breve{m}}_{NNij}}{\Gamma_{Mij}}\right]\right\}\\ &+\sum_{j=1}^{6}\sum_{i=1}^{6}\left\{\breve{v}_{NNij}\cdot\left[\gamma_i(t)\cdot\dot{x}_{\gamma j}(t)+\frac{\dot{\breve{v}}_{NNij}}{\Gamma_{Vij}}\right]\right\}\\ &+\sum_{i=1}^{6}\left\{\breve{g}_{NNi}\left[\gamma_i(t)+\frac{\dot{\breve{g}}_{NNi}}{\Gamma_{Gi}}\right]\right\}-\gamma^T(t)\cdot K_P\cdot\gamma(t)\end{aligned} \tag{30}$$

Substituting Eq. (27) into (30), and considering that $\dot{M}_{NN}=\{0\}$, $\dot{V}_{NN}=\{0\}$, $\dot{G}_{NN}=\{0\}$, yields

$$\begin{aligned}\dot{V}_S &\le -\gamma^T(t)\cdot K_P\cdot\gamma(t)-\sum_{i=1}^{6}\left\{\sigma_{Gi}\cdot|\gamma_i(t)|\cdot\breve{g}_{NNi}^2\right\}\\ &-\sum_{j=1}^{6}\sum_{i=1}^{6}\left\{\sigma_{Mij}\cdot|\gamma_i(t)|\cdot\breve{m}_{NNij}^2\right\}\\ &-\sum_{j=1}^{6}\sum_{i=1}^{6}\left\{\sigma_{Vij}\cdot|\gamma_i(t)|\cdot\breve{v}_{NNij}^2\right\}\le 0\end{aligned} \tag{31}$$

From Eqs. (28) and (31), we can conclude that V_S is a Lypunov function. Furthermore, when and only when $\gamma=0$ and $\dot{V}_S=0$ hold. It can be concluded that the system is asymptotically stable and $\gamma\to 0$ as $t\to+\infty$ based on LaSalle's theorem.

Define $\ell_p=\left\{x(t)\in\Re^n:\|x\|_p<\infty\right\}$ which represents $p-norm$ for $x(t)$. Then,
(1) From Eqs. (28) and (31), we can see that $\gamma(t)\in\ell_2$. From Eq. (20), we can easily obtain $e(t)\in\ell_2\cap\ell_\infty, \dot{e}(t)\in\ell_2$ and in fact $e(t)\to 0$ as $t\to\infty$.
(2) Since $\dot{V}_S\le 0, V_S\in\ell_\infty$, which implies that $\breve{m}_{NNij},\breve{v}_{NNij},\breve{g}_{NNij}\in\ell_\infty$. Furthermore, from Eq. (12) we can see that if the Jacobian matrix J keeps nonsingular, the parameter approximations are bounded, $m_{NNij},v_{NNij},g_{NNi}\in\ell_\infty$, so $\hat{m}_{NNij},\hat{v}_{NNij},\hat{g}_{NNi}\in\ell_\infty$. Moreover, the desired trajectory, velocity, and acceleration are all bounded, i.e., $x_d(t),\dot{x}_d(t),\ddot{x}_d(t)\in\ell_\infty$, from Eqs. (21) and (25), we can conclude that $\dot{\gamma}(t)\in\ell_\infty$. Then the control torque in Eq. (23) is bounded, i.e., $\tau(t)\in\ell_\infty$. That is to say, all the signals in the system are bounded.
(3) Considering that $\gamma(t)\in\ell_2$ and $\dot{\gamma}(t)\in\ell_\infty$, we can conclude that $\gamma(t)\to 0$ as $t\to\infty$, which is followed by $\dot{e}(t)\to 0$ as $t\to\infty$.

4 Simulation Results

In this simulation, a mobile modular manipulator which is composed of a three-wheeled nonholonomic mobile platform and a four-degree of freedom modular manipulator is investigated. Each element of matrix $\tilde{M}$ and $\tilde{G}$ is approximated by a three-layered MLP neural network in node numbers of 9, 100, and 1. Learning rates are all selected as $\eta_G=\eta_M=0.01$. However, elements for matrix $\tilde{V}$ are approximated by three-layered MLP neural networks with node numbers of 15, 500, and 1. And learning rates are all selected as $\eta_V=0.005$. A selected errors versus epochs curve in the process of MLP training is shown in Fig. 4.

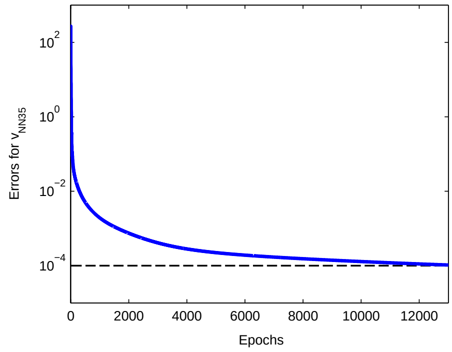

Figure 4: Curve for errors-versus-epochs.

The sliding mode adaptive neural-network controller designed in this paper is used to control the mobile modular manipulator to follow the desired trajectory shown in Fig. 5. Simulation time interval is chosen as 4 seconds. The gain matrices for the controller are selected as: $\Lambda=diag\{2.0\},K_P=diag\{1.0\},K_I=diag\{0.001\},\Gamma_M=\{0.1\},\Gamma_V=\{1.0\},\Gamma_G=\{1.0\},\sigma_M=\{0.05\},\sigma_V=\{0.001\},\sigma_G=\{0.1\}$.

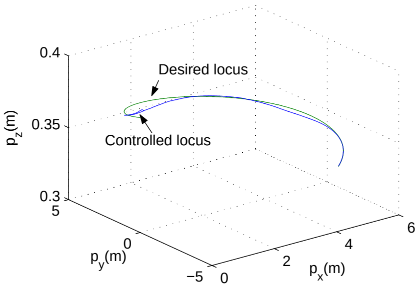

Figure 5: Locus for the end-effector.

The controlled locus is also shown in Fig. 5. Fig. 6 gives the tracking Euler angular errors. The tracking velocity errors are shown in Figures 7 and 8. Time variant control torques are shown in Figure 9. From these figures, we can see that small position and posture errors can be eliminated by this controller. If proper gain matrices are selected, the initial errors can be eliminated soon.

5 Conclusions

This paper investigates the end-effector trajectory following problem of nonholonomic mobile modular manipulators. Firstly, a kind of dynamic modeling method for entire nonholonomic mobile modular manipulators is presented in consideration of nonholonomic constraints. Secondly, three-layered

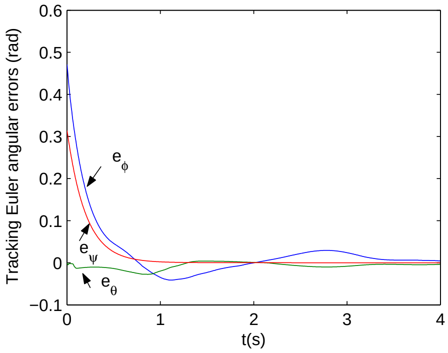

Figure 6: Tracking Euler angular errors.

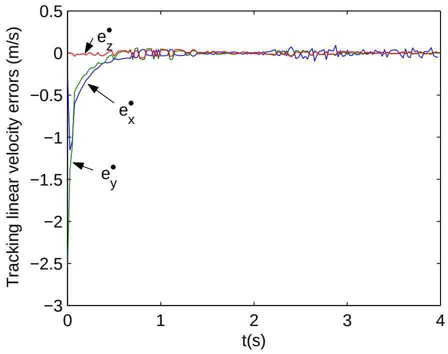

Figure 7: Tracking linear velocity errors.

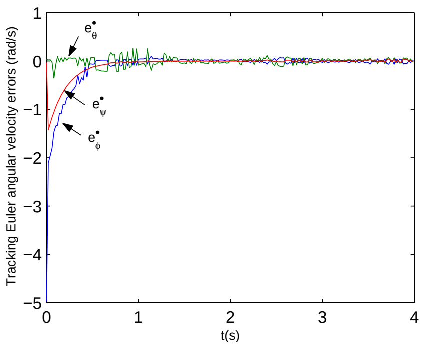

Figure 8: Tracking Euler angular velocity errors.

MLP neural networks are exploited to approximate dynamic models of mobile modular manipulators, and a sliding mode adaptive neural-network controller is designed to control the end-effector to follow a desired trajectory given in task space. Lastly, a simulation is carried out which has verified the effectiveness of the dynamic model and the sliding mode adaptive neural-network controller discussed in this paper.The dynamic modeling method and the sliding mode adaptive neural-network controller proposed in this paper can easily be extended to study some other mobile manipulators as well.

6 Acknowledgements

This work was supported by the Research Committee of University of Macau under Grant RG024/03-04S/LYM/FST.

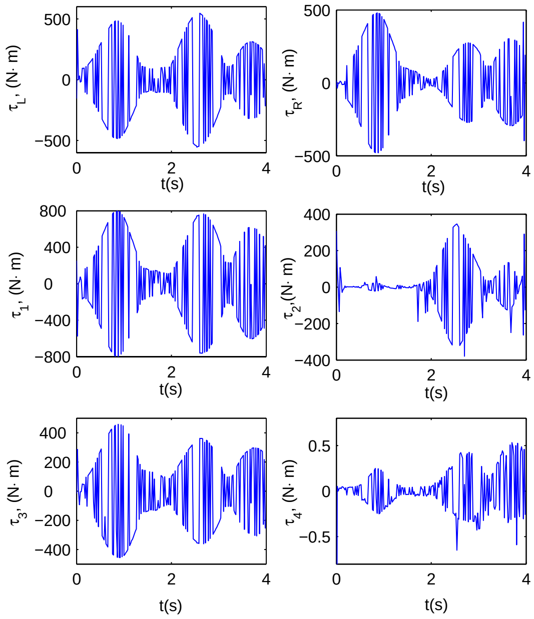

Figure 9: Time-variant driving torques.

References

[1] Li, Y., Liu,X., Peng,Z., and Liu,Y., "Identification of Joint Parameters for Modular Robots Based on Fuzzy Theory and Genetic Algorithms", *Robotica*,Vol. 20, No. 5, pp.509-517(2002).

[2] Yamamoto, Y. and Yun, X., "Effect of the Dynamic Interaction on Coordinated Control of Mobile Manipulators", *IEEE Transactions Robotics and Automatomation*,Vol.12, No.5,pp.816-824(1996).

[3] Sheng, L. and Goldenberg, A. A., "Neural-Network Control of Mobile Manipulators", *IEEE Transactions on Neural Networks*, Vol.12, No.5, pp.1121-1133(2001).

[4] Bartolini, G., Ferrara, A., Giacomini, L., and Usai, E., "Properties of a combined adaptive/second-order sliding mode control algorithm for some classes of uncertain nonlinear systems", *IEEE Trans. Automatic Control*, Vol. 45, No.7, pp. 1334-1341(2000).

[5] Yang, J.M. and Kim, J.H., "Sliding Mode Motion Control of Nonholonomic Mobile Robots", *IEEE Control Systems Magazine*, Vol.19, No.2, pp.15-23(1999).

[6] Li,Y. and Liu, Y., "Control of a Mobile Modular Manipulator Moving on a Slope", *Proc. of IEEE International Conference on Mechatronics,Istanbul,Turkey*, pp.135-140(2004).

[7] Haykin, S., *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, 1994.

[8] Ge, S.S., Lee, T.H. and Harris, C.J., *Adaptive Neural Network Control of Robotic Manipulators*, World Scientific Publishing Co. Pte. Ltd,(1998).