1 Introduction

Intelligent mobile manipulators have been given extensive attention in recent years since they have many applications such as in modern factories for transporting materials, in dangerous fields for dismantling bombs or moving nuclear infected objects, and at home for serving disabled persons. A nonholonomic mobile modular manipulator is normally composed of a m-wheeled nonholonomic mobile platform and a n-degree of freedom (DOF) onboard modular manipulator. This combination extends the workspace of the entire robot dramatically. Building up the dynamic model for such kind of robot is a challenging task due to the interactive motions between the manipulator and the mobile platform [1,2]. Also the motion planning is a hot research topic in mobile robot [3]. As for a mobile manipulator, the trajectory following task becomes even more complex and difficult to achieve [4,5]. Furthermore, stability is another concerning issue since the probability of tip-over increases due to this kind of mechanical structure [6]. Therefore, real-time tip-over prevention for a mobile modular manipulator without affecting the end-effector motion task is of great requisite. In this paper, the redundancy of a mobile modular manipulator is investigated to avoid tipping over of the entire robot by adjusting its self-motions. At another aspect, neural networks (NNs) and fuzzy logic systems (FLS) have been widely used for robotic control because of their ability to approximate arbitrary linear or nonlinear systems. Combining these two techniques, neural-fuzzy system (NFS) possess both of their advantages, i.e., low-level learning and computational power of NNs, and high-level human-like IF-THEN rule thinking and reasoning abilities of FLS. In this paper, NFS and adaptive technique are combined together to construct an ANFC.

This paper presents a practical method for automatic tip-over prevention and path following control of a redundant nonholonomic mobile modular manipulator. According to modular robot concept, the mobile platform is treated as a special module attached to the base of the modular manipulator, then an integrated structure is constructed and its dynamic modeling is performed. A new tip-over stability criterion based on the supporting forces is proposed in consideration of inertia, gravity, and acceleration. An online fuzzy logic (FL) self-motion planner and an adaptive neural-fuzzy controller (ANFC) are presented: The former is used to generate desired self-motions in a real-time manner, and the latter is used to prevent the robot from tipping over and to control the end-effector to follow a desired spatial trajectory at the same time. The proposed algorithm does not need any a priori knowledge of dynamic parameters and can suppress bounded external disturbances effectively. Simulation results for a real robot validate the dynamic modeling method and the controller design algorithm.

[DOI: 10.1115/1.2229253]
2 Dynamic Modeling

A three-wheel nonholonomic mobile modular manipulator moving on a slope is illustrated in Fig. 1(a). In Fig. 1, the coordinate systems are defined as follows: $O_x Y_x Z_x$ forms an inertial base frame, and $O_{x'} Y_{x'} Z_{x'}$ is a frame fixed on the slope; while $O_{y} X_{y} Y_{y} Z_{y}$ is a frame mounted on the mobile platform. $O_{m}(x_m,y_m)$ is selected as the midpoint of the line segment connecting the two driving-wheel contacting points with the motion plane; $Z_{m}$ is along the line segment mentioned above, and $Z_{m}$ is parallel with $Z_{y}$ passing through $O_{m}$. The notations shown in Fig. 1(b) are detailed in the nomenclature.

Supposing $\Delta t=t^{t+1}-t^{t}=0$, from Fig. 1(b), we can obtain

$$
\Delta S_L = \theta_L \left( r_L - \frac{d_m}{2} \right), \quad \Delta S_R = \theta_L \left( r_L + \frac{d_m}{2} \right)
$$

$$
\Delta S_M = \Delta \theta_M = \left| O'M_{o_m} O_m \right|, \quad \Delta \theta_M = \Delta \phi_m
$$

$$
\Delta x_m = x_m^{t+1} - x_m^t = \left| O'M_{o_m} O_m \right| \cdot \cos \phi_m
$$

$$
\Delta y_m = y_m^{t+1} - y_m^t = \left| O'M_{o_m} O_m \right| \cdot \sin \phi_m
$$

Noticing that Eq. (1) holds all the time, the superscript “$t$” can be omitted, the following equation can be derived:

$$
\dot{x}_m = \frac{r_f}{2} \cdot C_m \cdot (\phi_L + \phi_R)
$$

$$
\dot{y}_m = \frac{r_f}{2} \cdot S_m \cdot (\phi_L + \phi_R)
$$

Where $S_m = \sin \phi_m, C_m = \cos \phi_m$.

Defining $\xi=[x_m, y_m, \phi_L, \phi_R, \phi_m]^T$ as general coordinates of the mobile platform, then from Eq. (2), we can obtain

$$
\dot{\xi} = A(\xi) \cdot \dot{\xi} + S(\xi) \cdot \ddot{\xi}
$$

In short, $\ddot{\xi} = S(\xi) \cdot [\phi_L, \phi_R, \phi_m]^T$, and $A(\xi) \cdot \dot{\xi} = 0$.

Then, the nonholonomic constraints can be derived by

$$
A(\xi) \cdot S(\xi) = 0
$$

By taking the mobile platform as a special module attached to the modular manipulator, the following transformation matrices can be derived:
matrix of the

\[ \mathbf{E} = \begin{bmatrix} C_{33} & -C_{35} & \cdots & -S \cdot x_m \cdot C \cdot S \cdot x_m \cdot C \end{bmatrix} \]

\[ \mathbf{m} \cdot T = \begin{bmatrix} C_{33} & -C_{35} & \cdots & -S \cdot x_m \cdot C \cdot S \cdot x_m \cdot C \end{bmatrix} \]

\[ \mathbf{n} \cdot T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \]

Where \( S \) is the inclined angle of the slope; \( m \) is the transformation matrix of the first module of the mobile manipulator with respect to frame \( O_{x}X_{y}Z_{z} \).

Then according to Denavit-Hartenberg notation, transformation matrix of the \( i \)-th module with respect to \( O_{x}X_{y}Z_{z} \) can be derived by \( \mathbf{b}_{i}^T = \mathbf{b}_{i-1}^T \cdot \mathbf{a}_{i} \), where \( \mathbf{a}_{i} \) is the transformation matrix of module \( i \) with respect to module \( i-1 \). Furthermore, the position vectors \( (p_x, p_y, p_z) \) can be observed from the fourth column of matrix \( \mathbf{b}_{i}^T \).

Defining \( \xi = [q_1 \ldots q_n]^T \), \( q = [\phi_1 \ \phi_2 \ q_1 \ldots q_n]^T \), assuming \( x \) be a vector for task-space variables, from Eq. (3), we can obtain

\[ \dot{x} = \mathbf{S} \cdot \dot{q}, \quad \ddot{x} = \mathbf{S} \cdot \ddot{q} \]

In short, \( \dot{\xi} = \mathbf{S} \cdot \dot{q} \), \( \dot{\xi} = \mathbf{S} \cdot \ddot{q} \), \( \dot{I}_n = \mathbf{S} \cdot \dot{I}_n \). Defining \( \xi = [\dot{q} \ \ddot{q} \ \dot{I}_n]^T \), the mobility of \( n \times 2 \) joint space is in concern. Hence, then Eq. (6) yields

\[ \ddot{\xi} + (I_{n+2} - J^T \cdot J) \cdot \dot{\chi} = \mathbf{V} \]

Where \( J^T \cdot (J^T \cdot J)^{-1} \) is the Moore-Penrose generalized inverse of Jacobian matrix \( J \); \( \chi \) is an arbitrary \( n+2 \)-dimensional joint-velocity vector.

Let \( J_n \in \mathbb{R}^{(n+2) \times (n+1)} \) be a matrix with all its columns as the normalized bases of \( N(J_0) \), the null space of \( J \), then

\[ J \cdot J_n = 0_{(n+1) \times 1}, \quad J^T_n \cdot J_n = I_{n+1} \]

\[ J^T_n \cdot J^T = 0_{(n+1) \times 3}, \quad J_n^T \cdot J_n^T = I_{n+2} - J^T \cdot J \]

\[ J^T_n \cdot J^T = 0_{(n+1) \times 3}, \quad J_n^T \cdot (I_{n+2} - J^T \cdot J) = J^T_n \]

Defining \( \ddot{\chi} = J^T_n \cdot \dot{\chi} \), from Eqs. (6)-(8), we can obtain

\[ \ddot{\chi} = J^T_n \cdot \dot{\chi} + J^T_n \cdot \dot{\chi} + J_n \cdot \ddot{\chi} = J^T_n \cdot \dot{\chi} + J_n \cdot \ddot{\chi} \]

The constrained dynamics for the nonholonomic mobile manipulator can be determined by [26]

\[ M \cdot \ddot{\xi} + V \cdot \dot{\xi} + G = B \cdot (\tau + J^T \cdot F_{\text{exc}}) + C \cdot \lambda \]

Where \( M, V, \) and \( G \) represents the inertial matrix, the centripetal and coriolis matrix, and the gravitational force vector, respectively; \( F_{\text{exc}} = [F_x \ F_y \ F_z]^T \) is the external force vector added to the end-effector; \( \tau = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4 \ \tau_5]^T \) denotes a vector for driving torques; \( \lambda = [\lambda_1 \ \lambda_2 \ \lambda_3]^T \) are Lagrange multipliers associated with nonholonomic constraints; \( B = [0_{(n+2) \times 3} \ I_{n+1}] \), \( C = [A(\xi) \ 0_{(n+2) \times 1}] \).

From Eq. (6), \( \ddot{\xi} = \mathbf{S} \cdot \ddot{q} + \mathbf{V} \cdot \dot{\xi} + \mathbf{G} \cdot \xi = \mathbf{\tau} \)

Where \( \mathbf{\tau} = J^T \cdot F_{\text{exc}} \). The terms \( \mathbf{S} \cdot \ddot{q}, \mathbf{V} \cdot \dot{\xi} \), and \( \mathbf{G} \cdot \xi = \mathbf{\tau} \) are eliminated. Substituting Eq. (9) into Eq. (11) yields:

\[ \mathbf{\tau} = J^T \cdot \dot{\chi} + (\mathbf{\dot{\mathbf{V}}} - \mathbf{\ddot{M}} \cdot \dot{\chi} \cdot J^T) \cdot [\dot{\chi} \ \ddot{\chi}] \]

Defining \( x_n = [x^T \ x_n^T]^T \), \( J^T_n = [J^T_n \ J_n]^T \), left multiplying Eq. (10) yields

\[ \dot{x} = J^T_n \cdot \dot{x} + J_n \cdot \ddot{x} \]

Fig. 2 Tip-over stability analysis for a mobile manipulator on a slope

Journal of Dynamic Systems, Measurement, and Control

\[ \ddot{M} \cdot \dot{x}_E + \ddot{V} \cdot \dot{x}_E + \ddot{G} = \tau \]  
\[ \text{where} \quad \ddot{M} = (J^*)^T \ddot{M} \cdot (J^*), \quad \ddot{V} = (J^*)^T \ddot{V} \quad (\ddot{V} - \ddot{M} \cdot J^*) \cdot (J^*), \quad \ddot{G} = (J^*)^T \ddot{G}, \quad \ddot{\tau} = (J^*)^T \ddot{\tau}. \]

**Remark 1.** If \( J \) is full rank, then matrix \( (J^*)^{-1} \) is invertible, and \( J_k = (J_k^*)^{-1} \).

**Remark 2.** Matrix \( \ddot{M} \) is symmetric, i.e., \( \ddot{M}^T = \ddot{M} \); and for any \( \gamma \in R^{n \times 2} \), \( \gamma^T \cdot \ddot{M} \cdot \gamma = 0 \).

**Remark 3.** The matrix \( \ddot{M} - 2 \ddot{V} \) is skew-symmetric, i.e., for any \( \gamma \in R^{n \times 2} \), \( \gamma^T \cdot (\ddot{M} - 2 \ddot{V}) \cdot \gamma = 0 \).

**Remark 4.** The dynamics is linear with respect to structure parameters \( \phi \), i.e., \( \ddot{M} \cdot \dot{x}_E + \ddot{V} \cdot \dot{x}_E + \ddot{G} = Y(\ddot{\zeta}, \dot{\phi}, \dot{\theta}) \cdot \ddot{G} \).

**Remark 5.** The matrices \( \ddot{M}, \ddot{V}, \ddot{G} \) are all bounded as long as the Jacobian matrix \( J_k \) is full rank.

### 3 Tip-Over Stability Analysis

To the three-wheel nonholonomic mobile platform analyzed in this paper, there are a totally of three points contacting with the ground. On the assumption of stiff ground surface, the ground is a kind of single-bounded constraint, i.e., the ground can only support the robot to prevent it from plunging into the ground, but cannot pull it back to prevent from leaving the ground. So, the supporting forces should be non-negative. If all these three wheels touch the ground, the three contacting points can determine a plane and the robot is stable. However, if one of the constraint forces is approaching to zero, the wheel will leave the ground, tip-over and instability will occur, as shown in Fig. 2(a). Hence constraint forces becoming zero can be treated as a critical tip-over stability state. Under this circumstance, even a small disturbance force may topple down the mobile modular manipulator. From the analysis above, the tip-over stability is determined by the minimum supporting force acting on these three wheels. And the most stable situation is that all the burden is distributed evenly to the three wheels, i.e., \( N_{fl} = N_{fr} = N_t \). Then, a tip-over stability criterion based on supporting forces can be defined as

\[ \sigma_{sa} = \sqrt{\left( N_{fl} - N_t \right)^2 + (N_{fr} - N_t)^2 + (N_t - N_t)^2} \]

where \( N_t = (N_{fl} + N_{fr} + N_r)/3 \).

To simplify the calculation, the external forces and the gravity acceleration are projected to \( O_0 X_0 Y_0 Z_0 \), i.e., \( F_{ext} = R^T \cdot F_{ext} \) and \( g = R^T \cdot [0 0 -9.81]^T \). Here, \( R \) is the rotation matrix of frame \( O_0 X_0 Y_0 Z_0 \) with respect to frame \( O_0 X_0 Y_0 Z_0 \). The tip-over stability criterion [19] assumed that the mobile platform move and rotate in constant velocities, the criterion is extended to variable velocity conditions in this paper. Considering that \( O_0 X_0 Y_0 Z_0 \) is not an inertial frame, the Newton’s second law cannot be used directly. Then, inertial forces should be added during the course of analysis. Force analysis for a nonholonomic mobile modular manipulator moving on a slope is shown in Fig. 2(b). From Figs. 1(b) and 2(b), supporting forces of the three wheels can be derived in the same way as given by [19]. Because of page limitation, we only give the results

\[ N_f = \sum_{i=0}^{n} \left[ m_i \left[ \left( m_i g + m_i a_i \right) \cdot \left( m_i p_i - \left( m_i g + m_i a_i \right) \cdot \left( m_i p_i \right) \right) \right] - \sum_{i=0}^{n} \left[ m_i \left( m_i p_i \right) \right] \right] \]

\[ = \frac{2}{d_m} \left[ \left( m_i g \cdot \left( d_m + m_i p_i \right) \right) \right] - \sum_{i=0}^{n} \left[ m_i \left( m_i p_i \right) \right] \]

\[ N_{fl} = \sum_{i=0}^{n} \left[ m_i \left[ \left( a_i + g \right) \cdot \left( m_i p_i - \left( a_i + g \right) \cdot \left( m_i p_i \right) \right) \right] - \sum_{i=0}^{n} \left[ m_i \left( m_i p_i \right) \right] \right] \]

\[ = \frac{2}{d_m} \left[ \left( m_i g \cdot \left( d_m + m_i p_i \right) \right) \right] - \sum_{i=0}^{n} \left[ m_i \left( m_i p_i \right) \right] \]

\[ N_{fr} = \sum_{i=0}^{n} \left[ m_i \left[ \left( a_i + g \right) \cdot \left( m_i p_i - \left( a_i + g \right) \cdot \left( m_i p_i \right) \right) \right] - \sum_{i=0}^{n} \left[ m_i \left( m_i p_i \right) \right] \right] \]

\[ = \frac{2}{d_m} \left[ \left( m_i g \cdot \left( d_m + m_i p_i \right) \right) \right] - \sum_{i=0}^{n} \left[ m_i \left( m_i p_i \right) \right] \]

\[ \dot{x}_i = d\dot{\theta}_i / dt = \left[ m_i \dot{\theta}_i (t + \Delta t) - m_i \dot{\theta}_i (t) \right] / \Delta t \]

\[ \dot{\phi}_i = d\phi_i / dt = \left[ \phi_i (t + \Delta t) - \phi_i (t) \right] / \Delta t \]

\[ \dot{\theta}_i = d\theta_i / dt = \left[ \theta_i (t + \Delta t) - \theta_i (t) \right] / \Delta t \]

The desired task-space trajectory, velocity, and acceleration \((x_d, \dot{x}_d, \ddot{x}_d)\) can all be determined in advance. However, the desired self-motions \((\dot{x}_d, \dot{\theta}_d, \ddot{x}_d)\) have to be generated in a real time manner. To prevent the robot from tipping over, desired self-
motions are generated by online FL planners. Inputs to these FL planners are $k_{pi} = \partial x_{Ed}/\partial q_{i}$, whose limits can be derived from Eqs. (14) and (15) or determined by a trial-and-error approach [27], while the outputs are $\dot{x}_{Ed}$. Assuming that $\dot{q} \in [\dot{q}^-, \dot{q}^+]$, then the output fuzzy sets can be determined: $\dot{x}_{Ed} \in [\dot{x}_{Ed}^-, \dot{x}_{Ed}^+]$, with $\dot{x}_{Ed}^- = \dot{q}^- - \dot{q}$. In these FLS, triangular membership functions are employed, as shown in Fig. 3. To each input or output fuzzy set, nine fuzzy centroids are defined, i.e., negative very big (NVB), negative big (NB), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), positive big (PB), and positive very big (PVB).

The basic idea for the FL inference is very simple, i.e., to minimize the tip-over stability criterion by adjusting the self-motions. Expressed in the form of fuzzy rules, we can obtain

$$
\text{if } k_{pi} \text{ is NVB, then } \dot{x}_{Ed} \text{ is PVB}
$$

$$
\text{if } k_{pi} \text{ is NB, then } \dot{x}_{Ed} \text{ is PB}
$$

$$
\text{if } k_{pi} \text{ is NM, then } \dot{x}_{Ed} \text{ is PM}
$$

$$
\text{if } k_{pi} \text{ is NS, then } \dot{x}_{Ed} \text{ is PS}
$$

$$
\text{if } k_{pi} \text{ is ZE, then } \dot{x}_{Ed} \text{ is ZE}
$$

$$
\text{if } k_{pi} \text{ is PS, then } \dot{x}_{Ed} \text{ is NS}
$$

$$
\text{if } k_{pi} \text{ is PM, then } \dot{x}_{Ed} \text{ is NM}
$$

$$
\text{if } k_{pi} \text{ is PB, then } \dot{x}_{Ed} \text{ is NB}
$$

$$
\text{if } k_{pi} \text{ is PVB, then } \dot{x}_{Ed} \text{ is NVB}
$$

(17)

By applying product inference and centroid defuzzifier, the desired self-motions $\dot{x}_{Ed}$ can be determined by

$$
\dot{x}_{Ed} = \sum_{i=1}^{q} \left\{ \mu_{k_{pi}} \cdot b_{\dot{x}_{Ed}^i} \right\} / \sum_{i=1}^{q} \left\{ \mu_{k_{pi}} \right\}
$$

(18)

Where $\mu_{k_{pi}}$ is the input membership function for the $k$th rule; $b_{\dot{x}_{Ed}^i}$ is the corresponding centroid of the consequent fuzzy set.

Then, according to Eq. (9), the desired trajectory in null space $\ddot{x}_{Ed}$ can be obtained. Furthermore, $\ddot{x}_{Ed}$ and $\dot{x}_{Ed}$ can be determined by differential and integral actions, i.e., $\ddot{x}_{Ed}(t) = d\ddot{x}_{Ed}/dt$ and $\dot{x}_{Ed}(t) + \Delta(t) = \ddot{x}_{Ed}(t) + \int_{t_0}^{t} \dot{x}_{Ed}(\tau) d\tau dt$.

Let $x_{Ed} = [x_{Ed}^T, \ddot{x}_{Ed}^T]^T$, the error systems are defined by

$$
e(t) = x_{Ed}(t) - x_{Ed}(t)
$$

$$
\dot{s}(t) = \dot{s}(t) - \Delta \cdot e(t)
$$

(19)

Where $\Delta$ is the constant positive definite matrix.

Substituting Eq. (19) into Eq. (13), yields

$$
\tilde{M} \cdot \ddot{s}(t) + \tilde{V} \cdot \dot{s}(t) + \tilde{M} \cdot \ddot{x}(t) + \tilde{V} \cdot \dot{x}(t) + \tilde{G} = \tilde{r}
$$

(20)

It is verified that the multiple inputs single output (MISO) FLS with center average defuzzifier, product inference rule and singleton fuzzifier, and Gaussian membership function can uniformly approximate any nonlinear functions over a compact set to any degree of accuracy, see [24] for details. If the FLS described above is realized by a NN, a NFS can be obtained, as shown in Fig. 4. The MISO NFS output is given by [27].
Fig. 5 An adaptive neural-fuzzy controller

\[
y = f_{\text{NFS}} = \sum_{j=1}^{N_{\text{NFS}}} \left\{ \prod_{i=1}^{N_{\text{I}}} \exp \left[ -\frac{(x_i - \mu_{ji})^2}{\sigma_{ji}} \right] \right\}
\]

Where \( x_i \) is the \( i \)th input variable; \( w_i \) denotes the point at which the output membership function for the \( j \)th rule achieving its maximum value; \( i = 1, 2, \ldots, N_{\text{I}}, j = 1, 2, \ldots, N_{\text{NFS}} \) where \( N_{\text{I}} \) and \( N_{\text{NFS}} \) represent the number of input variables and rules, respectively. \( \mu_{ji} \) and \( \sigma_{ji} \) are the mean and standard derivations of the Gaussian membership functions accordingly.

From Remark 5 and Eq. (19), the last three terms on the left side of Eq. (20) are all bounded and belong to compact sets as long as the Jacobian matrix \( J \) is full rank. Then, according to the universal approximation theory mentioned above, each element of their sum can be approximated by a MISO NFS in the form of Fig. 4. Furthermore, from the nonholonomic velocity constraints and the error system defined above, inputs to these NFS can be selected as \( x_{\text{in}} = [\xi^T \quad \dot{\xi}^T \quad \dot{\xi}_{\text{f}}^T \quad \dot{x}_{\text{f}}^T]^T \). Then

\[
h(x_{\text{in}}) = M \cdot \dot{\xi}(t) + \bar{V} \cdot \dot{\xi}(t) + \bar{G} = h_{\text{NFS}}(x_{\text{in}}) + \varepsilon(x_{\text{in}}) \tag{22}
\]

Where \( h_{\text{NFS}} \in \mathcal{R}^{n+2} \) is the NF approximation of \( h \) with its elements \( h_{\text{NFS}}=f_{\text{NFS}}(x_{\text{in}}, \sigma_k, \wr_k, w_k) \); here \( \sigma_k, \wr_k \in \mathcal{R}^{n \times N_{\text{NFS}}}, \) and \( w_k \) \( \in \mathcal{R}^{N_{\text{NFS}}} \) are adjustable parameter matrices for NFS; \( \varepsilon_k \) are the corresponding approximated errors, \( k=1,2,\ldots,n+2 \).

Supposing \( x_{\text{in}} = [x_{\text{in}}^T \quad x_{\text{in}}^T] \), \( h \in [h^T \quad h^T] \), then the adjustable parameters can be initialized as follows:

\[
\begin{align*}
\mu_{kji} &= x_{\text{in}} + j \cdot \frac{x_{\text{in}} - x_{\text{in}}}{N_{\text{NFS}}} \\
\sigma_{kji} &= \frac{x_{\text{in}} - x_{\text{in}}}{N_{\text{NFS}}} \\
w_{kji} &= h_{kji} + j \cdot \frac{h_{kji} - h_{kji}}{N_{\text{NFS}}} 
\end{align*}
\]

Let the estimators of \( \mu_{kji}, \sigma_{kji}, \) and \( w_{kji} \) be \( \hat{\mu}_{kji}, \hat{\sigma}_{kji}, \) and \( \hat{w}_{kji} \), respectively, the Taylor series expansions of \( h_{\text{NFS}} \) around \( \hat{h}_{\text{NFS}} = f_{\text{NFS}}(x_{\text{in}}, \sigma_k, \wr_k, w_k) \) can be expressed by

\[
\begin{align*}
\tilde{h}_{\text{NFS}} &= \sum_{j=1}^{N_{\text{NFS}}} \sum_{i=1}^{N_{\text{I}}} \left[ \frac{\partial f_{\text{NFS}}(x_{\text{in}}, \sigma_k, \wr_k, w_k)}{\partial \sigma_{kji}} \cdot \hat{\sigma}_{kji} + O(\sigma_{kji}^2) \right] \\
+ \sum_{j=1}^{N_{\text{NFS}}} \sum_{i=1}^{N_{\text{I}}} \left[ \frac{\partial f_{\text{NFS}}(x_{\text{in}}, \sigma_k, \wr_k, w_k)}{\partial \wr_{kji}} \cdot \hat{\wr}_{kji} + O(\wr_{kji}^2) \right] \\
+ \sum_{j=1}^{N_{\text{NFS}}} \left[ \frac{\partial f_{\text{NFS}}(x_{\text{in}}, \sigma_k, \wr_k, w_k)}{\partial w_{kji}} \cdot \hat{w}_{kji} + O(w_{kji}^2) \right] 
\end{align*}
\]

Where \( \sigma_{kji} = \sigma_{kji} - \hat{\sigma}_{kji}, \quad \wr_{kji} = \sigma_{kji} - \hat{\wr}_{kji}, \quad \hat{w}_{kji} = w_{kji} - \hat{w}_{kji} \), and \( \tilde{h}_{\text{NFS}} \)

Fig. 6 A real robot and external disturbances introduced in case 2
\[ V_\lambda = \frac{1}{2} \cdot s^T \cdot \hat{M} \cdot s + \frac{1}{2} \cdot \left( \int_0^t s(t) dt \right)^T \cdot K_I \cdot \left[ \int_0^t s(t) dt \right] + \frac{1}{2} \cdot \sum_{k=1}^{n+2} \sum_{i=1}^{N_k} \left( \tilde{\sigma}_{ij}^2 + \tilde{\sigma}_{kj}^2 \right) + \frac{1}{2} \cdot \left( \tilde{w}_{kj}^2 + \tilde{w}_{kj}^2 \right) \geq 0 \]

The time derivative of Lyapunov candidate is as follows:

\[ \dot{V}_\lambda = \frac{1}{2} \cdot s^T \cdot \hat{M} \cdot s + s^T \cdot \hat{K}_P \cdot s + \frac{1}{2} \cdot \left( \int_0^t s(t) dt \right)^T \cdot K_I \cdot \left[ \int_0^t s(t) dt \right] + \frac{1}{2} \cdot \sum_{k=1}^{n+2} \sum_{i=1}^{N_k} \left( \tilde{\sigma}_{ij}^2 + \tilde{\sigma}_{kj}^2 \right) + \frac{1}{2} \cdot \left( \tilde{w}_{kj}^2 + \tilde{w}_{kj}^2 \right) \]

(31)

From Eq. (29), we have

\[ s^T \cdot \left( \hat{M} \cdot \hat{K}_P \cdot s + \frac{1}{2} \cdot \left( \int_0^t s(t) dt \right)^T \cdot K_I \cdot \left[ \int_0^t s(t) dt \right] \right) = -s^T \cdot \hat{V} \cdot s - s^T \cdot \hat{K}_P \cdot s \]

(32)

Notice that \( \tilde{\sigma}_{ij} = w_{ij} \), \( \tilde{\sigma}_{kj} = w_{kj} \), \( \hat{w}_{ij} = \hat{w}_{ij} \), \( \hat{w}_{kj} = \hat{w}_{kj} \). Substituting Eqs. (27) and (32) into Eq. (31), and considering Eqs. (24) and (26) and Remark 3 at the same time, yields

\[ V_\lambda \leq -s^T \cdot \hat{K}_P \cdot s \leq 0 \]

(33)

From Eqs. (30) and (33), \( V_\lambda \) is a Lyapunov function. Furthermore, if \( s = 0 \), \( V_\lambda \) and \( V_\lambda \) may equal to zero. According to LaSalle’s theorem, the system is asymptotically stable and \( s \rightarrow 0 \) as \( t \rightarrow +\infty \).

To ensure the system far away from singularities, the Jacobian matrix \( J \) is required to be full rank, i.e., \( \det(J) \neq 0 \). A desired singularity-free path can be predefined.

5 Simulation Results

The mobile modular manipulator in this simulation is constructed by combining a three-wheel nonholonomic mobile platform, called “Pioneer 3-DXe,” with a 4-DOF modular manipulator, named “POWER-CUBE,” as shown in Fig. 6(a) in our laboratory. From Fig. 6(a), the last joint “joint4” has no influence on the positions of the end-effector, so \( n = 3 \) in this case.

To ensure the system far away from singularities, the Jacobian matrix \( J \) is required to be full rank, i.e., \( \det(J) \neq 0 \). Then, a desired singularity-free path can be predefined.

To verify the effectiveness of the control algorithm presented in this paper, simulations for two control schemes are performed: (1) The FL self-motion planner and the ANFC (FLP and ANFC); (2) an inverse dynamics controller (IDC) designed in task space [26]. The performance of the FLP and ANFC is compared with that of the IDC for three different cases. All the joint angles and velocities are initialized to be zero. Simulation time is selected as 20 s.
Case 1: This case is designed to test the self-motion stabilization characteristic of the proposed algorithm. The robot is supposed to move on a horizontal plane, i.e., $\theta_b = 0$ deg. Both FLP and ANFC and IDC are used to control the end-effector to follow a similar parabola spatial trajectory. For the FLP and ANFC, each element of $h$ is approximated by a NFS, and all the adjustable parameters are initialized according to Eq. (23). The gain matrices or constants are selected by Eq. (34). For the IDC, precise dynamic parameters are assumed to be absolutely known in advance, and no external disturbances are introduced. The structure of the IDC is given by Eq. (35), which does not consider self-motions $K_P = \text{diag}[100], \quad \Gamma_{\alpha_{ij}} = 0.1, \quad \Lambda = \text{diag}[2.0]$

$$K_I = \text{diag}[10.0], \quad \Gamma_{\alpha_{ij}} = 0.1, \quad N_r = 200$$

$$K_e = \text{diag}[20.0], \quad \Gamma_{\omega_{ij}} = 0.1$$

$$\tau = \ddot{\bar{M}}f[\ddot{\bar{x}}_d - \ddot{\bar{q}} + K_P(\ddot{\bar{x}}_d - x) + K_P(\ddot{\bar{x}}_d - x)] + \ddot{\bar{V}}_q + \ddot{\bar{G}} - J^T F_{\text{ext}}$$

Where $K_P = \text{diag}[100]$ and $K_I = \text{diag}[10]$ are proportional and differential gain matrices. Proof for the stability and convergent characteristics of this controller can be found in [26].

Figure 7 gives the simulation results for case 1. The left side is for IDC and the right side is for ANFC. Tracking locus, position, and control torques are shown in (a), (c), (e), and (g) for IDC, and (b), (d), (f), and (h) for ANFC.
errors, velocity errors, and control torques are shown, respectively, from up to down. It is not difficult to find from these figures that the dynamic modeling method proposed in this paper is effective and the ANFC can stabilize the system effectively by controlling self-motions. However, the simulation results for IDC just present an example for torque instability caused by self-motions.

Case 2: This case is designed to verify the disturbance suppression abilities of the controllers. The desired trajectory is a sine-like spatial curve and $\theta_1=0$ deg. For the FLP and ANFC, all parameters are selected in the same way as that in case 1. For the IDC, nominal dynamic parameters are adopted, which are deviated from the real values by 10%. A series of disturbance external forces are introduced, as shown in Fig. 6(b). By considering self-motions created by the FLP, the controller is given by

$$\tau = \bar{M} \cdot \dot{J}_E \cdot [\ddot{x}_{Ed} - \dot{J}_E \cdot \dot{q} - K_D \cdot \dot{e} - K_P \cdot e] + \bar{V} \cdot \dot{q} + \bar{G} - J^T \cdot F_{ext}$$

(36)

Where $K_P=\text{diag}(100)$ and $K_D=\text{diag}(10)$.

Simulation results for case 2 are shown in Fig. 8. Figures 8(a) and 8(b) give the desired and controlled locus for IDC and ANFC,
respectively. The tracking position and velocity errors are shown in Figs. 8–8 accordingly. Figures 8(g) and 8(h) show time-variant control torques. From these figures, the ANFC behaves better than IDC when parameter uncertainties and external disturbances exist. Comparing the simulation results of IDC with those in case 1, we can conclude that it is the self-motions that cause the system almost divergent in case 1.

Case 3: This case is designed to examine the tip-over prevention ability of the proposed algorithm. An elliptical like spacial trajectory is adopted and $\theta_2$ is assumed to be 30 deg. Parameters for the FLP and ANFC are the same as those in case 1. The same structure as in case 2 is adopted for the IDC. The desired self-motion is not created from the FLP, but is selected corresponding to the following optimization problem: $\min(q^T \cdot q)$ subject to $\dot{x} = J \cdot \dot{q}$.

Simulation results for case 3 are presented in two parts. Tracking locus, position errors, as well as velocity errors are shown in Figs. 9(a)–9(f), respectively. Figures 10(a) and 10(b) give the self-motions. Figure 10(c) presents the time-variant minimum supporting forces for IDC. The tip-over stability measure for the FLP and ANFC is shown in Figure 10(d). Time-variant control torques are given by Figs. 10(e) and 10(f), respectively. From these figures, the FLP and ANFC presented in this paper can prevent the entire robot from tipping over effectively. However, the IDC may have to be faced with such a danger.

Remark 7. In this paper, the FLP and ANFC are designed in task space directly, so calculation for the inverse Jacobian matrix can be avoided. However, this control scheme needs some task-space information, such as $x$ and $\dot{x}$, which is difficult to measure. In this case, the task-space information is calculated by forward kinematics, and differential kinematics, respectively. Another solution is to construct a vision-based control system. However, such a system will not be widely used in practice until a very powerful but not too expensive camera with high speed data processing ability appears. Furthermore, the mobile robot “Pioneer 3-DXe” adopted in this paper is encapsulated so well that no application programming interfaces are provided for direct motor control. So, to verify the proposed algorithm on the real mobile modular manipulator, there are still a great deal of work to do, both on software programming and hardware designing. This paper aims to present a theoretical analysis method and the experiment will be carried out in our future study.

6 Conclusions

In this paper, a general redundant nonholonomic mobile modular manipulator composed of a three-wheel nonholonomic mobile platform and an n-DOF onboard modular manipulator is investigated. Based on the modular robotic concept, an integrated dynamic modeling method is presented which takes the nonholonomic constraints, the interactive motions, and the self-motions into thorough consideration. The tip-over stability criterion based
on wheel supporting forces is extended, and an online FL self-
motion planner is proposed to prevent the entire robot from over-
turn. An ANFC is devised to control the end-effector to follow a
desired spatial trajectory. This algorithm does not need any a
priori dynamic parameters and can suppress bounded external dis-
turbances effectively. Simulation results for a real redundant non-
holonomic mobile modular manipulator prove that the dynamic
modeling method and the controller design algorithm proposed in
this paper are effective.

Acknowledgment
This work was supported by the Research Committee of the
University of Macau under Grant No. RG082/04-05S/LYM/FST.
The authors also appreciate the comments and suggestions from
anonymous reviewers and editors.

Nomenclature

- \( r_r \) = radius of the castor wheel for the mobile
  platform
- \( m_i \) = mass of the \( i \)th module for the modular
  manipulator
- \( m_f \) = mass of the driving wheels for the mobile
  platform
- \( r_f \) = radius of the driving wheels for the mobile
  platform
- \( \theta_r \) = yaw angle of the mobile platform at interval
  \([r', r''+1]\)
- \( r_i \) = steering radius of the mobile platform at inter-
  val \([r', r''+1]\)
- \( l_r \) = distance from the fixed bar to the driving
  wheel axis
- \( d_r \) = distance from the fixed bar to the castor wheel
  axis
- \( \phi_r \) = rotating angle of the castor wheel around its
  own axis
- \( \beta_r \) = rotating angle of the castor wheel around the
  fixed bar
- \( \phi_m \) = heading angle of the mobile platform
- \( d_m \) = distance between the two driving wheels
- \( l_G \) = offset of the first module with respect to \( O_m \)
  along \( X_m \)
- \( h_G \) = offset of the first module with respect to \( O_m \)
  along \( Z_m \)
- \( m_c \) = mass of the cart (including all components in
  the box)
- \( N_r \) = supporting forces on the castor wheel
- \( \phi_L \) = rotating angle of the left driving wheel

Fig. 10 Simulation results for case 3—part 2
\[ \phi_R = \text{rotating angle of the right driving wheel} \]
\[ a^*_L = \text{linear acceleration of the cart} \]
\[ N_{fL} = \text{supporting forces on the left driving wheel} \]
\[ N_{fR} = \text{supporting forces on the right driving wheel} \]
\[ A_{m} = \text{advance of the point } O_m \text{ at the time interval} \]
\[ \Delta S_L = \text{advance of the left driving wheel at interval} \]
\[ \Delta S_R = \text{advance of the right driving wheel at interval} \]
\[ m^*_i = \text{angular acceleration vector for the } i^{th} \text{ module} \]
\[ m^*_L = \text{angular acceleration of the left driving wheel} \]
\[ m^*_R = \text{angular acceleration of the right driving wheel} \]
\[ p_c = \text{coordinate vector of the end-effector} \]
\[ p_{c1} = \text{coordinate vector of center of mass (cm) for the cart} \]
\[ a_{c1} = \text{linear acceleration vector of the } i^{th} \text{ module} \]
\[ r_i = \text{inertial moment of the } i^{th} \text{ module with respect to } X_m \]
\[ r_i = \text{inertial moment of the } i^{th} \text{ module with respect to } Y_m \]
\[ m() = \text{refers to the frame } O_mX_mY_mZ_m \]

References