Springer

KSME

Journal of Mechanical Science and Technology 24 (6) (2010) 1337~1351
www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-010-0403-7

Real-time regulated sliding mode controller design of multiple manipulator space free-flying robot †

Hamid Khaloozadeh[1,*] and M. Reza Homaeinejad[2]

[1]*Department of Control Engineering, K.N. Toosi University of Technology, Tehran, Iran*
[2]*Department of Mechanical Engineering, K.N. Toosi University of Technology, Tehran, Iran*

(Manuscript received January 16, 2008; Revised July 13, 2009; Accepted April 1, 2010)

Abstract

The problem of controlling Space Free-flying Robots (SFFRs), which have many degrees of freedom caused by their mechanical manipulators, is challenging because of the strong nonlinearities and their heavy computational burden for the implementation of model-based control algorithms. In this paper, a chattering avoidance sliding mode controller is developed for SFFR as highly nonlinear-coupled systems. To fulfill stability requirements, robustness properties, and chattering elimination, a regulating routine is proposed to determine the proper positive values for the coefficient of sliding condition. To solve the run-time problem, an explicit direct relationship between the SFFR's output of actuators (force/torque) and the measurement of distances from the corresponding sliding surfaces is also assumed. To reach perfect performance, the parameters are estimated recursively using the Kalman filter as a parameter estimator. The explicit dynamics of a 14-DOF SFFR is derived using SPACEMAPLE, and the recursive prediction error method (RPEM) is used to parameterize the SFFR model. To alleviate the chattering trend, a multi-input sliding mode control law is proposed and applied to the given SFFR based on the online estimated dynamics to control its orientation and position to catch a moving target. To evaluate the new proposed algorithm in a more complicated condition, only on–off actuators are assumed for controlling the base of SFFR because it is the case in real systems. The obtained results show that the proposed regulated sliding mode controller can significantly reduce the chattering trend. Consequently, energy consumption will be substantially decreased, and running the control algorithm will be within a reasonable time duration.

Keywords: 14-DOF space free-flying robots; Variable structure systems; Parameter estimation; Kalman filtering

1. Introduction

A Space Free-flying Robot (SFFR) includes an actuated base equipped with one or more manipulators to perform a variety of tasks in orbit. Distinct from fixed-based manipulators, the spacecraft (base) of an SFFR responds to dynamic reaction forces caused by manipulator motions. To control such system, considering the dynamic coupling between the manipulators and the base is essential. To develop control systems for space assemblies, establishing a proper kinematics/dynamics model for the system is vital. This has been studied under the assumption of rigid elements [1, 2] and also of elastic elements [3, 4]. Various studies on the nonlinear control problem of such systems [5-7] have been conducted because of the complicated nonlinearities in space systems, maneuver time limitations, and restrictions in energy consumption. Systems that include uncertainties, such as parametric or structural uncertainties, also need appropriate strategies to be controlled. Two important approaches used for dealing against nonlinearities and uncertainties are robust control and adaptive control [8-10].

One of the main approaches to robust control is Sliding Mode Control (SMC) [11]; it is usually accompanied by a phenomenon called "chattering" [12]. Chattering should be avoided to reduce the energy consumption of the control system and to prevent any potential damages on actuators, especially in the case of the on–off type. Due to the high-frequency content of chattering, it can also easily stimulate flexible modes, which in turn may cause instability. To alleviate the chattering phenomenon, saturation functions instead of switching operators that degrade the control precision can be used. An auxiliary continuous control may be added to the control input obtained from the conventional sliding mode design [13]. This continuous control will eventually replace the time average of the discontinuous control in the steady state, and the switching gain goes to zero when the distance from the sliding surface decreases toward zero, thus eliminating chattering. Although the steady-state response of this controller is acceptable, this

† This paper was recommended for publication in revised form by Associate Editor Jong Hyeon Park

*Corresponding author. Tel.: +98 21 88462174, Fax.: +98 21 88462066
E-mail address: h_khaloozadeh@kntu.ac.ir

method is not very efficient for systems with time varying desired states. For multi-input nonlinear systems, where the equations are completely coupled, the selection of controller parameters to reach acceptable response is not straightforward [14-17]. A serious problem in the model-based control of robotic systems is that the computational burden imposed on the robot computer is heavy, causing the real-time exploitation of the control algorithm to take high costs [18]. To date, some solutions have been proposed to solve this problem based on identification and estimation strategies [19-33]. To solve this problem, we propose a model that can be supposed directly between actuators' outputs and the sensors measurements of distances from the sliding surfaces. Through this, the computational task will be rid of vigorous mathematical calculations of Jacobian matrices and can be therefore used in real time.

In this paper, focusing on the chattering phenomenon to fulfill energy limitations in space, a new approach is proposed to alleviate (ideally eliminate) the chattering trend. To fulfill the stability requirements, robustness properties, and chattering elimination, a regulating routine is proposed to determine the proper positive values for the coefficient of the sliding condition. The dynamic model of an SFFR is obtained through the SPACEMAPLE code. Derivation of the equations of motion results in explicit derivations of the system's mass matrix and of the vectors of nonlinear velocity terms and generalized forces. Unlike recursive dynamics formulations, the obtained dynamics model is very useful for dynamics analyses, design studies, and development of model-based control algorithms. The recursive PEM method is used for the parameterization of the explicit direct model between actuators' output and the measurement of the distances from the corresponding sliding surfaces. Afterwards, the 14-DOF SFFR is simulated as a highly nonlinear and coupled system. The actuators acting at the base of the SFFR are primarily assumed as continuous servos with saturation limits, and to illustrate the merits of the new proposed algorithm, they are subsequently considered as on–off discontinuous actuators. Despite the fact that the conventional sliding mode controller cannot successfully control the system in the latter case, the obtained results show that the proposed regulated sliding mode controller can substantially alleviate the chattering trend. Consequently, the system energy consumption will also be significantly decreased when the control algorithm is run in a reasonable time duration.

2. SFFR dynamics modeling

A typical maneuver of an SFFR is of relatively short length and duration; thus, microgravity and dynamical effects caused by orbital mechanics are negligible compared with control forces. Therefore, the motion of the system is considered with respect to an in-orbit inertial frame of reference (**XYZ**), and the system potential energy is taken as equal to zero. In Fig. 1, the general configuration of an SFFR and its essential coordinates are shown.

The general Lagrangian formulation for such system yields

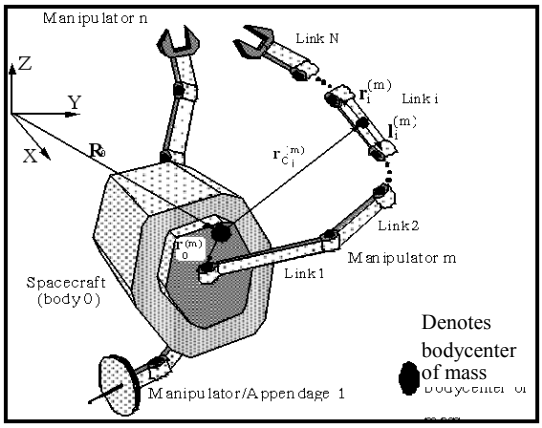

Fig. 1. General configuration and essential coordinate systems for the description of a space robotic system.

the following:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right)-\left(\frac{\partial T}{\partial q_i}\right)=Q_i;i=1,\cdots,N \tag{1}$$

where T is the system kinetic energy; N is the system degrees-of-freedom; and q_i, $\dot{q}_i$, and Q_i are the i-th elements of the vector of the generalized coordinates, generalized speeds, and generalized forces, respectively. To apply Eq. (1) and obtain dynamics equations, the system kinetic energy, T, must first be derived. This can be written as follows:

$$T=\frac{1}{2}\int_M \dot{\mathbf{R}}_P\cdot\ \dot{\mathbf{R}}_P\, dM \tag{2}$$

Where, M defines the system-distributed mass and $\dot{\mathbf{R}}_P$ is the velocity of an arbitrary point P, which can be evaluated based on the direct path kinematics approach for multiple manipulator SFFR with rigid elements that was developed in Moosavian and Papadopoulos (1997) as follows:

$$P\in Base:\dot{\mathbf{R}}_p^{(0)}=\dot{\mathbf{R}}_{C_0}+\omega_0\times\mathbf{r}_{p/C_0} \tag{3}$$

where

$$P\in Link_i^{(m)}:\dot{\mathbf{R}}_{p_i}^{(m)}=\dot{\mathbf{R}}_{C_0}+\omega_0\times\mathbf{r}_0^{(m)}+\sum_{k=1}^{i-1}\omega_k^{(m)}\times(\mathbf{r}_k^{(m)}-\mathbf{l}_k^{(m)})-\omega_i^{(m)}\times(\mathbf{l}_i^{(m)}-\mathbf{r}_{p/C_i^{(m)}})$$

describes the spacecraft center of mass velocity; $\mathbf{r}_{p/C_0}$ describes the position of P with respect to the spacecraft center of mass; vectors $\mathbf{l}_i^{(m)}$, $\mathbf{r}_i^{(m)}$ are the body-fixed vectors that describe the position of joints i and i+1 with respect to C_i, as seen in Fig. 1; and ω_0 and $\omega_k^{(m)}$ are the angular velocity of the spacecraft and of the k-th link of the m-th manipulator, respectively. For single DOF joints, the angular velocity of an individual body can be obtained as follows:

$$\omega_k^{(m)} = \omega_0 + \sum_{i=1}^{k} \dot{\theta}_i^{(m)} \mathbf{z}_i^{(m)} \qquad \begin{cases} m = 1, \cdots, \ n \\ k = 1, \cdots, \ N_m \end{cases} \tag{4}$$

where $\mathbf{z}_i^{(m)}$ is a unit vector along the axis of rotation of the i-th joint of the m-th manipulator, and $\dot{\theta}_i^{(m)}$ is the corresponding joint angle rate.

The substitution of Eq. (3) for $\dot{\mathbf{R}}_P$ into Eq. (2) yields the following:

$$T = \frac{1}{2} \int_M (\dot{\mathbf{R}}_{C_0} + \dot{\mathbf{r}}_{C_i} + \omega_i \times \mathbf{r}_{p/C_i})(\dot{\mathbf{R}}_{C_0} + \dot{\mathbf{r}}_{C_i} + \omega_i \times \mathbf{r}_{p/C_i}) \, dM \tag{5}$$

This can be simplified to obtain

$$T = T_0 + T_1 + T_2 \tag{6a}$$

with

$$T_0 = \frac{1}{2} M \, (\dot{\mathbf{R}}_{C_0} \cdot \ \dot{\mathbf{R}}_{C_0}) \tag{6b}$$

$$T_1 = \frac{1}{2} \left\{ \omega_0 \mathbf{I}_0 \omega_0 + \sum_{m=1}^{n} \sum_{i=1}^{N_m} (m_i^{(m)} \ \dot{\mathbf{r}}_{C_i}^{(m)} \ \dot{\mathbf{r}}_{C_i}^{(m)} + \omega_i^{(m)} \mathbf{I}_i^{(m)} \omega_i^{(m)}) \right\} \tag{6c}$$

(6d)

$$T_2 = \dot{\mathbf{R}}_{C_0} \cdot \left(\sum_{m=1}^{n} \sum_{i=1}^{N_m} m_i^{(m)} \ \dot{\mathbf{r}}_{C_i}^{(m)} \right) \tag{6e}$$

and $\dot{\mathbf{r}}_{C_i}^{(m)}$ describes the velocity of C_i, which can be obtained as follows:

$$\dot{\mathbf{r}}_{C_i}^{(m)} = \omega_0 \times \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1} \omega_k^{(m)} \times (\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \omega_i^{(m)} \times \mathbf{l}_i^{(m)} \qquad \begin{cases} m = 1, \cdots, \ n \\ i = 1, \cdots, \ N_m \end{cases}$$

Note that the expressions for T are in terms of invariant body-fixed vectors. To conduct the required differentiations in Eq. (1), appropriate transformation matrices for each term must be employed. The vector of the generalized coordinates is chosen as follows:

$$\mathbf{q} = (\mathbf{R}_{C_0}^T, \delta_0^T, \theta^T)^T \tag{7a}$$

This can be arranged as follows:

$$\mathbf{q} = (\mathbf{q}^{(0)^T}, \mathbf{q}^{(1)^T}, \cdots, \mathbf{q}^{(n)^T})^T \tag{7b}$$

where

$$\mathbf{q}^{(0)} = (\mathbf{R}_{C_0}^T, \mathrm{d}_0^T)^T \tag{7c}$$

$$\mathbf{q}^{(m)} = \mathrm{q}^{(m)} = (\theta_1^{(m)}, \theta_2^{(m)}, \cdots, \theta_{N_m}^{(m)})^T \tag{7d}$$

with δ_0 as the spacecraft Euler angles, and $\theta_i^{(m)}$ (i=1,…,N_m) describing the m-th manipulator joint angles. Using Eq. (6) and applying the general Lagrangian formulation, Eq. 1, the equations of motion are obtained as follows:

$$\mathbf{H}(\delta_0, \theta) \, \ddot{\mathbf{q}} \ + \mathbf{C}(\delta_0, \dot{\delta}_0, \theta, \dot{\theta}) = \mathbf{Q}(\delta_0, \theta) \tag{8}$$

where the vector of generalized coordinates **q** is defined in Eq. (7); **C** is an N×1 vector that contains all the nonlinear velocity terms (in a microgravity environment); and **Q** is the N×1 vector of the generalized forces (N = K + 6) given by the following:

$$\mathbf{Q} = \begin{Bmatrix} \mathbf{0}_{6\times1} \\ \tau_{K\times1} \end{Bmatrix} + \sum_{p=1}^{i_f} \mathbf{J}_{0,p}{}^T \mathbf{F}_{0,p} + \sum_{m=1}^{n} \sum_{i=1}^{N_m} \sum_{p=1}^{i_f} \mathbf{J}_{i,p}^{(m)\ T} \mathbf{F}_{i,p}^{(m)} \tag{9}$$

$\mathbf{F}_{0,p}$ is the p-th external force or moment applied on the spacecraft; $\mathbf{F}_{i,p}^{(m)}$ is the p-th external force or moment applied on the i-th body of the m-th manipulator; i_f is the number of applied forces or moments on the corresponding body; and $\mathbf{J}_{i,p}^{(m)}$ is a Jacobian matrix corresponding to the point of force or moment application. Eq. (9) can be obtained based on the definition of the generalized forces. This equation can be rearranged in a way that the actuator forces or torques are displayed explicitly. If all external forces, except the ones applied on the spacecraft, are zero, **Q** can be written as

$$\mathbf{Q} = \mathbf{J}_Q \begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \\ \tau_{K\times1} \end{Bmatrix} \tag{10}$$

where ${}^0\mathbf{f}_s$ and ${}^0\mathbf{n}_s$ are the net force and moment applied on the spacecraft, respectively, and $\mathbf{J}_Q$ is an N × N Jacobian matrix. For a well-designed system, $\mathbf{J}_Q$ remains nonsingular, (i.e., any required **Q** can be produced by the system's actuators). To obtain an explicit dynamics model of a multiple manipulator SFFR, mathematical analyses are presented to help calculate the mass matrix, the vector of nonlinear velocity terms, and the generalized forces.

3. Explicit dynamics model

3.1 Mass matrix

To obtain the mass matrix **H**, according to Eq. (8), the acceleration terms in each of the three formats have to be considered. Therefore, H_{ij} is computed by the following:

- Substituting each term of the system kinetic energy into the corresponding format

- Finding the coefficients of $\ddot{\mathbf{q}}$ in each format
- Adding the results obtained from the three formats for each term
- Adding the results obtained for all of the terms

Disregarding the details, this procedure eventually yields the following:

$$H_{ij} = M\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\frac{\partial \mathbf{R}_{C_0}}{\partial q_j} + \frac{{}^0\partial \omega_0}{\partial \dot{q}_i}\cdot \mathbf{I}_0 \cdot \frac{{}^0\partial \omega_0}{\partial \dot{q}_j} + \sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_i}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_j} + \frac{{}^k\partial \omega_k^{(m)}}{\partial \dot{q}_i}\mathbf{I}_k^{(m)}\frac{{}^k\partial \omega_k^{(m)}}{\partial \dot{q}_j}\right) + \left(\sum_{m=1}^{n}\sum_{k=1}^{N_m} m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_i}\right)\frac{\partial \mathbf{R}_{C_0}}{\partial q_j} + \left(\sum_{m=1}^{n}\sum_{k=1}^{N_m} m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_j}\right)\frac{\partial \mathbf{R}_{C_0}}{\partial q_i} \tag{11}$$

where $\omega_k^{(m)}$ is given by Eq. (4), and $\mathbf{r}_{C_k}^{(m)}$ can be substituted from the following (see Fig. 1):

$$\mathbf{r}_{C_i}^{(m)} = \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1}(\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \mathbf{l}_i^{(m)} \quad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{12}$$

The notation employed here is consistent to Kane and Levinson (1985) (i.e., a left superscript on partial derivatives refers to the frame where the differentiation has to be taken, whereas it is left blank for the inertial frame).

3.2 Vector of nonlinear terms

The vector of nonlinear velocity terms in Eq. (8) can be computed by dropping the acceleration terms in each of the obtained formats. Thus, C_i is computed following the same procedure as described for computing the H_{ij}, that is, by considering the coefficients of $\dot{\mathbf{q}}$ and any other term (except those which correspond to $\ddot{\mathbf{q}}$) in each format. This approach yields the following:

$$\mathbf{C}(\delta_0, \dot{\delta}_0, \theta, \dot{\theta}) = \mathbf{C}_1(\delta_0, \dot{\delta}_0, \theta, \dot{\theta})\,\dot{\mathbf{q}} + \mathbf{C}_2(\delta_0, \dot{\delta}_0, \theta, \dot{\theta}) \tag{13}$$

Where,

$$\mathrm{C}_{1ij} = M\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{R}_{C_0}}{\partial q_s \partial q_j}\dot{q}_s\right) + \frac{{}^0\partial \omega_0}{\partial \dot{q}_i}\cdot \mathbf{I}_0 \cdot \frac{{}^0\partial \omega_0}{\partial q_j} + \omega_0 \cdot \mathbf{I}_0 \cdot \frac{{}^0\partial^2 \omega_0}{\partial \dot{q}_i \partial q_j} + \sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_i}\cdot\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}_{C_k}^{(m)}}{\partial q_s \partial q_j}\dot{q}_s\right) + \frac{{}^k\partial \omega_k^{(m)}}{\partial \dot{q}_i}\cdot \mathbf{I}_k^{(m)} \cdot \frac{{}^k\partial \omega_k^{(m)}}{\partial q_j} + \omega_k^{(m)} \cdot \mathbf{I}_k^{(m)} \cdot \frac{{}^k\partial^2 \omega_k^{(m)}}{\partial \dot{q}_i \partial q_j}\right) \left(\sum_{s=1}^{N}\frac{\partial^2 R_{C_0}}{\partial q_s \partial q_i}\dot{q}_s\right)\cdot \sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_j}\right) + \frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)}\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}_{C_k}^{(m)}}{\partial q_s \partial q_j}\dot{q}_s\right) \tag{14}$$

and

$$\mathrm{C}_{2i} = -\left(\omega_0\,\mathbf{I}_0\frac{{}^0\partial \omega_0}{\partial q_i} + \sum_{m=1}^{n}\sum_{k=1}^{N_m}\omega_k^{(m)}\mathbf{I}_k^{(m)}\frac{{}^k\partial \omega_k^{(m)}}{\partial q_i}\right) \tag{15}$$

Expressing the angular velocity as a function of the Euler rates, vector $\mathbf{C}_2$ can be combined with the first term of Eq. (13a). The vector of nonlinear velocity terms can then be written as follows:

$$\mathbf{C}(\delta_0, \dot{\delta}_0, \theta, \dot{\theta}) = \breve{\mathbf{C}}(\delta_0, \dot{\delta}_0, \theta, \dot{\theta})\,\dot{\mathbf{q}} \tag{16}$$

This is a representation of nonlinear velocity terms preferred in the development of adaptive control algorithms.

3.3 Vector of generalized forces

As described in Eq. (9), if all external forces except the ones applied on the spacecraft are zero, the vector of generalized forces $\mathbf{Q}$ is written as follows:

$$\mathbf{Q} = \mathbf{J}_Q \begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \\ \mathrm{t}_{K\times 1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0}_{6\times 1} \\ \mathrm{t}_{K\times 1} \end{Bmatrix} + \mathbf{J}_0{}^T \begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \end{Bmatrix} \tag{17}$$

Assuming that ${}^0\mathbf{f}_s$ and ${}^0\mathbf{n}_s$ are applied at the spacecraft center of mass, $\mathbf{J}_0$ is defined as

$$\begin{Bmatrix} {}^0\dot{\mathbf{R}}_{C_0} \\ {}^0\omega_0 \end{Bmatrix} = \mathbf{J}_0\,\dot{\mathbf{q}} \tag{18}$$

Therefore, $\mathbf{J}_Q$ is obtained as follows:

$$\mathbf{J}_Q = \begin{bmatrix} \mathbf{T}_0 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{3\times 3} & \mathbf{S}_0{}^T & \mathbf{0}_{3\times K} \\ \mathbf{0}_{K\times 3} & \mathbf{0}_{K\times 3} & \mathbf{1}_{K\times K} \end{bmatrix}_{N\times N} \tag{19}$$

which can be substituted into Eq. (10) to obtain $\mathbf{Q}$. This completes the derivation of the dynamics model for a multiple-arm SFFR with rigid elements. The computation of the obtained dynamics equations can be conducted either by *numerical* or *symbolical programming tools* [2].

4. Regulated SMC Law

A multi-input nonlinear system can be defined as follows:

$$x_i^{(n_i)} = f_i(\mathbf{x}) + \sum_{j=1}^{m} b_{ij}(\mathbf{x})\;\; u_j; i, j = 1, \ldots m \tag{20}$$

where $\mathbf{u}$ describes the control input array, and $\mathbf{x}$ is defined as the state vector composed of $\mathbf{x}_i$'s as state arrays:

$$\mathbf{x}_i = [x_i \quad \dot{x}_i .... x_i^{n_i-1}]^T \tag{21}$$

where f_i represents the dynamics of the i[th] state as a function of state vector $\mathbf{x}$; b_{ij} is the corresponding element of input matrix "B" that describes the gain function of the j[th] input on sub-system "i"; n_i is the order of the corresponding differential equation; and "m" is the number of independent inputs. The control aim can be expressed as making the state vector $\mathbf{x}$ follow the desired time-dependent vector $\mathbf{x}_d$. In the presence of modeling uncertainties, it is assumed that all parametric uncertainties appear in the input matrix B or its estimated value $\hat{\mathbf{B}}$, and that it is nonsingular in the state space domain. Therefore, it can be written as follows:

$$\begin{aligned} &\mathbf{B} = (\mathbf{I} + \boldsymbol{\Delta})\hat{\mathbf{B}} \\ &|\Delta_{ij}| \le D_{ij} \qquad i, j = 1,.....,m \\ &|\hat{f}_i - f_i| \le F_i \end{aligned} \tag{22}$$

where $\hat{f}_i$ is the estimated value of f_i that can be obtained from the dynamics model, and Δ_{ij} can yield the error values of the input matrix estimation procedure. The exact value of Δ_{ij} is unknown, but the upper-bound limitation (i.e., D_{ij}) can be substituted. Therefore, the distance from a sliding surface is defined as

$$\begin{aligned} &s_i = \left(\frac{d}{dt} + \lambda_i\right)^{n_i-1} \tilde{x}_i \\ &\tilde{x}_i = x_i - x_{id} \end{aligned} \tag{23}$$

where λ_i 's are controller parameters and time constants in a low pass filter sequence [11], and $\tilde{x}_i$ describes the tracking error of x_i . Eq. (23) can be written as

$$s_i = x_i^{(n_i-1)} - x_{r_i}^{(n_i-1)} \tag{24}$$

where $x_{r_i}^{(n_i-1)}$ is computed based on the error between $\vec{x}_i$ and $\vec{x}_{i_d}$. For instance, considering a system with two state arrays and two independent inputs, it can be obtained by

$$s_i = (\frac{d}{dt} + \lambda_i).\tilde{x}_i = \dot{x}_i - (\dot{x}_{id} - \lambda_i \tilde{x}_i) \quad i = 1,2 \tag{25}$$

which yields the following:

$$x_{r_i} = \dot{x}_{i_d} - \lambda_i \tilde{x}_i \quad i = 1,2 \tag{26}$$

Therefore, in a general case, the control inputs must be determined in a way that satisfies the following sliding condition:

$$\frac{1}{2}\frac{d}{dt}s_i^2 \le -\eta_i |s_i| \qquad \eta_i > 0 \tag{27}$$

where η_i 's are controller parameters and are chosen as positive values reflecting the way states are converged to their sliding surfaces. The higher values of η_i indicate that the corresponding state reaches its sliding surface faster. Assuming that K_i 's are positive values that must be determined to satisfy the sliding condition (27), then the control law is obtained as

$$\mathbf{u} = \left(\hat{\mathbf{B}}\right)^{-1}\left[\mathbf{x}_r^n - \hat{\mathbf{f}} - \mathbf{Ksgn(s)}\right] \tag{28}$$

where K_i 's can be obtained using Filippov's Construction of Equivalent Dynamics [11] by calculating $\dot{s}_i = 0$ (for i = 1,…,n), which yields the following:

$$\begin{aligned} &(1 + D_{ii})K_i + \sum_{j=1} D_{ij}K_j = F_i + \sum_{j=1}^{n} D_{ij}\left|x_{ij}^n - \hat{f}_j\right| + \eta_i \\ &i = 1,...,n \end{aligned} \tag{29}$$

Eq. (29) defines a set of "*n*" equations with "*n*" unknowns (i.e., K_i 's). To solve these equations, η_i 's should be chosen. As explained before, η_i is a factor that indicates the speed of the corresponding state in approaching its sliding surface. Therefore, if η_i can be determined in such a way that based on the absolute value of the distance from the sliding surface, the speed of the corresponding state becomes lower and reaches zero on the surface, then the performance will be as it is desired, and chattering will be substantially alleviated, if not vanished. Therefore, rather than the conventional heuristic method of choosing η_i , that the selection is based on the following Etta Regulating Procedure (ERP) is proposed:

$$\begin{aligned} \eta_i(t) = (\eta_{0_i}\left|1 - e^{-|S_i|}\right|) \times \frac{1}{2}[1 - (1) \times \mathrm{sgn}(|\, s_i\,| - s_{acti}^*)] \\ + \eta_{0i} \times \frac{1}{2}[1 + (1) \times \mathrm{sgn}(|\, s_i\,| - s_{acti}^*)] \end{aligned} \tag{30}$$

A less complicated form is as follows:

$$\eta_i(t) = \eta_{0_i}\left|1 - e^{-|S_i|}\right| \tag{31}$$

In Eq. (30), the parameter s_{acti}^* is a positive constant value that specifies the activity margin of the ERP mechanism. For instance, if the transient response of the system is more important than the smoothness of the approaching rate of the system state vector toward the sliding plane, s_{acti}^* should be of small value, whereas the value of η_{0i} should be large enough. On the other hand, if the smoothness of the response is more important, s_{acti}^* can be given a large value. To select the initial value, η_{0i} , any large value can be chosen. In the absence of

uncertainties, there are various aspects in the conventional heuristic method of choosing η_i. First, in terms of noise effects, the controller must be capable of dealing against the separation of the state trend from the corresponding sliding surface. Therefore, with a reasonable high value of η_i, the noise effect will decrease. On the other hand, after reaching the states to the sliding surface, *chattering* may occur. Unfortunately, the amplitude of this phenomenon increases along with a higher value of η_i. Therefore, the noise rejection characteristics are in conflict with the chattering effect reduction. However, following the proposed procedure, starting with high initial values for η_{0i}, will compensate for the two effects and eventually alleviate the chattering. The main concern is to control the momentum of the system. In a conventional sliding mode controller with a constant switching gain, chattering will occur when the amplitude is proportional to the η_i. When the state vector of the system reaches the sliding surface, it will pass through the sliding surface because of the system's inertia if the momentum of the system is not equal to the desired momentum (according to the desired trajectory). Thus, when the state vector is in the vicinity of the sliding surface and approaches the surface, the amplitude of the chattering will decrease if the applied input is regulated to set the system momentum. On the other hand, that the value of η_i's will be set properly when the system state vector goes farther from the sliding surface must be guaranteed. Eq. (30) can be replaced by a simpler form such as Eq. (31); however, the regulating function is continuous in this case. The sliding mode controller equipped with the new proposed ERP has a broad margin of stability against parametric uncertainties, variation of controller parameters, and actuators limitations [14].

5. Recursive identification algorithms

Based on the previous sections, designing the sliding mode controller requires Jacobian matrices $\mathbf{J}_{\mathbf{Q}}, \mathbf{J}_{\dot{\mathbf{q}}}$, mass matrix (i.e., H), and the vector of the non-linear velocity terms C. The most common way to design a controller with respect to a given desired trajectory corresponds to the following steps:

(1) Determination of $\hat{\mathbf{Q}}$ based on the used control algorithm
(2) Conversion of $\hat{\mathbf{Q}}$ to $\mathbf{Q}$ using the $\mathbf{J}_{\dot{\mathbf{q}}}$ Jacobian matrix
(3) Quadrature of the motion equations described in the generalized coordinate and finding the $\mathbf{q}$ and $\dot{\mathbf{q}}$
(4) Conversion of $\mathbf{Q}$ to the actuator space using the $\mathbf{J}_{\mathbf{Q}}$ Jacobian matrix to command the actuators
(5) Conversion of $\dot{\mathbf{q}}$ to the task variables (output variables) ($\dot{\hat{\mathbf{q}}}$) using the $\mathbf{J}_{\dot{\mathbf{q}}}$ Jacobian matrix and the calculation of the tracking errors

Due to the presence of massive mathematical terms in the mentioned matrices and vector C, the computational burden imposed on the robot computer will be drastically high. Therefore, implementing the sliding mode control or any model-based control algorithm for real-time purposes is not possible. One way to solve this problem is by building a linear model and by setting its parameters corresponding to the robot's measured input(s) and output(s). On the other hand, because the considered robotic system is non-linear, the regulation of the parameterized model parameters should be continued with a priori knowledge of robot dynamics. To fulfill this, recursive identification methods can be used, with the dynamics of the robot assumed to consist of a number of prominent sub-systems. These sub-systems are shown as a couple (i, j). The index "i" indicates the i^{th} measured output, whereas the index "j" indicates the j^{th} measured input. The output of the system is considered the "distance from the sliding surface" because the control algorithm is chosen as the SMC; the input of the system is appointed as the "actuators force/moment" applied to the robotic system; and the spacecraft and sub-systems are the relationships between the input and output. By choosing these options, the complexity of the equations of motion is relieved completely, and simulated output can be used directly in the control algorithm. The recursive estimation of the parameters vector $\hat{\boldsymbol{\theta}}$ is given by the following difference equation:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)\left[\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right] \tag{32}$$

where $\hat{\boldsymbol{\theta}}(t)$ is the parameter vector estimation at time t; y(t) is the vector of the observed output; and $\hat{\mathbf{y}}(t)$ is the one-step prediction of the system's output with respect to a priori observations of the input and output. K(t) is the gain matrix of the estimation process, which determines the way the current prediction error ($\mathbf{y}(t) - \hat{\mathbf{y}}(t)$) affects the update of the parameter estimation.

The choice for the K(t) is typically the following equation:

$$\mathbf{K}(t) = \mathbf{Q}(t)\boldsymbol{\psi}(t) \tag{33}$$

In Eq. (33), the vector $\boldsymbol{\psi}(t)$ is the vector of regressors, and it contains old values of the observed input and output. The one-step prediction of the system is given by

$$\hat{\mathbf{y}}(t) = \boldsymbol{\psi}^T(t)\hat{\boldsymbol{\theta}}(t-1) \tag{34}$$

Matrix $\mathbf{Q}(t)$ regulates the adaptation gain and gives the direction when the estimation is pursued. The optimal value of $\mathbf{Q}(t)$ is obtained from Riccati's forward equation solution as it is given by

$$\begin{cases} \mathbf{Q}(t) = \dfrac{\mathbf{P}(t-1)}{R_2 + \boldsymbol{\psi}^T(t)\mathbf{P}(t-1)\boldsymbol{\psi}(t)} \\ \mathbf{P}(t) = \mathbf{P}(t-1) + \mathbf{R}_1 - \dfrac{\mathbf{P}(t-1)\boldsymbol{\psi}(t)\boldsymbol{\psi}^T(t)\mathbf{P}(t-1)}{R_2 + \boldsymbol{\psi}^T(t)\mathbf{P}(t-1)\boldsymbol{\psi}(t)} \end{cases} \tag{35}$$

In Eq. (35), drift matrix $\mathbf{R}_1$ is the covariance matrix of the true system parameters change, which logically assumes a random walk:

$$\begin{aligned} &\theta_0(t) = \theta_0(t-1) + w(t) \\ &E\left[w(t)w^T(s)\right] = R_1\delta(t-s) \end{aligned} \tag{36}$$

R_2 is the variance of the innovations in the true system if it is declared in the linear regression format as

$$\begin{cases} y(t) = \boldsymbol{\psi}^T(t)\boldsymbol{\theta}_0(t) + e(t) \\ R_2 = E\left[e^2(t)\right] \end{cases} \tag{37}$$

Finally, to solve the recursive equations, covariance matrix $\mathbf{R}_1$, variance R_2, $\mathbf{P}(0)$, and $\boldsymbol{\theta}(0)$ should be appointed in the beginning. To describe a general input–output linear model for a system with input u and output y, the following structure is introduced:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t-nk) + \frac{C(q)}{D(q)}e(t) \tag{38}$$

where

$$\begin{cases} A(q) = 1 + a_1 q^{-1} + ... + a_{na} q^{-na}; B(q) = b_1 q^{-1} + ... + b_{nb} q^{nb} \\ C(q) = c_1 q^{-1} + ... + c_{nc} q^{-nc}; F(q) = 1 + f_1 q^{-1} + ... + f_{nf} q^{-nf} \\ D(q) = 1 + d_1 q^{-1} + ... + d_{nd} q^{-nd} \end{cases} \tag{39}$$

Several elaborations of the general linear model given by Eq. (38) can be extracted, such as ARX, ARMAX, BJ, OE, and PEM [34-36]. By mathematical manipulation, the general model described by Eq. (38) can be rewritten as

$$P(q)y(t) = R(q)u(t-nk) + S(q)e(t) \tag{40}$$

Using the above difference equation, the corresponding regressor vectors can be written as follows:

$$\begin{aligned} \boldsymbol{\psi}(t) = [&-y(t-1) ... - y(t-np)\ u(t-nk-1) \\ &...u(t-nk-nr+1)\ e(t-1)...e(t-ns)]^T \end{aligned} \tag{41}$$

To reach the ARX or ARMAX structure, the order of the general model described by Eq. (38) should be chosen as

$$\begin{cases} ARX: np = na; nr = nb; ns = 0; \\ ARMAX: np = na; nr = nb; ns = nc; \end{cases} \tag{42}$$

5.1 Prediction error method algorithm

By minimizing a cost function defined for the prediction error $e(t)$, a general model is parameterized as follows [36]:

$$\begin{cases} \mathbf{e}(t) = [\mathbf{H}(q)]^{-1}[\mathbf{y}(t) - \mathbf{G}(q)\mathbf{u}(t)] \\ (\hat{\mathbf{G}}, \hat{\mathbf{H}}) = \arg\min \sum_{t=1}^{N} \left(\|\mathbf{e}(t)\|^2 \right) \end{cases} \tag{43}$$

After some mathematical manipulating, the following result can be obtained:

$$\begin{cases} \hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)[\mathbf{y}(t) - \hat{\mathbf{y}}(t)] \\ \mathbf{K}(t) = \mathbf{Q}(t)\boldsymbol{\psi}(t) \\ \mathbf{Q}(t) = \mathbf{Q}(t-1) - \dfrac{\mathbf{Q}(t-1)\boldsymbol{\psi}(t)\boldsymbol{\psi}^T(t)\mathbf{Q}(t-1)}{1 + \boldsymbol{\psi}^T(t)\mathbf{Q}(t-1)\boldsymbol{\psi}(t)} \\ \varepsilon(t) \approx \varepsilon(t, \hat{\boldsymbol{\theta}}(t-1)) \\ \boldsymbol{\psi}(t) \approx -\left[\dfrac{\partial}{\partial \boldsymbol{\theta}} \varepsilon(t, \hat{\boldsymbol{\theta}}(t-1))\right]^T \end{cases} \tag{44}$$

6. SMC controller design

Using the expressions for the kinetic and potential energies and by applying Lagrange's equations for a space robotic system as shown in Fig. 1, the dynamics model can be obtained as follows:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q} \tag{45}$$

where $\mathbf{q}$ is the vector of the generalized coordinates; $\mathbf{C}$ contains all nonlinear velocity and gravity terms; and $\mathbf{Q}$ is the vector of the generalized forces. Gravity terms are practically zero in microgravity environments and can therefore be neglected in the design of control laws. In terrestrial applications, these terms may cause static positioning errors in the control; in such case, they must be compensated separately. Therefore, in this paper, the $\mathbf{C}$ vector contains only nonlinear velocity terms. The output velocities $\dot{\hat{\mathbf{q}}}$, which are the rates of variables to be controlled (outputs), are obtained from the generalized velocities $\dot{\mathbf{q}}$ using a Jacobian matrix $\mathbf{J}_{\hat{\mathbf{q}}}$ as

$$\dot{\hat{\mathbf{q}}} = \mathbf{J}_{\hat{\mathbf{q}}}(\mathbf{q})\dot{\mathbf{q}} \tag{46}$$

Assuming that this Jacobian matrix is square and nonsingular, Eq. (45) can be written in terms of the output variables as follows:

$$\hat{\mathbf{H}}(\mathbf{q})\ddot{\hat{\mathbf{q}}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{Q}} \tag{47}$$

where

$$\begin{cases} \hat{\mathbf{H}} = \mathbf{J}_{\hat{q}}^{-T}\mathbf{H}\,\mathbf{J}_{\hat{q}}^{-1} \\ \hat{\mathbf{C}} = \mathbf{J}_{\hat{\mathrm{q}}}^{-\mathrm{T}}\mathbf{C} - \hat{\mathbf{H}}\,\dot{\mathbf{J}}_{\hat{\mathrm{q}}}\,\dot{\mathbf{q}} \\ \hat{\mathbf{Q}} = \mathbf{J}_{\hat{q}}^{-T}\mathbf{Q} \end{cases} \tag{48}$$

The above dynamics model is obtained through the SPACEMAPLE code [2]. The derivation of the equations of motion results in an explicit derivation of the system's mass matrix and of the vectors of nonlinear velocity terms and generalized forces. Unlike with recursive dynamics formulations,

the obtained dynamics model is very useful for dynamics analyses, design studies, and development of model-based control algorithms. To design a sliding mode controller based on Eq. (28), the dynamics of the system and the gain matrix of inputs are defined as follows:

$$\ddot{\mathbf{q}} = \mathbf{f} + \mathbf{B}\mathbf{u} \tag{49}$$

where

$$\begin{cases} \mathbf{f} = -\left[\hat{\mathbf{H}}(\mathbf{q})\right]\hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}) \\ \mathbf{B} = \left[\hat{\mathbf{H}}(\mathbf{q})\right]^{-1} \\ \mathbf{u} = \hat{\mathbf{Q}} \end{cases} \tag{50}$$

The distance from the sliding "hyper surface" is defined as an $n \times 1$ column vector, where n is the system's DOF, and its components are defined as follows:

$$\begin{aligned} s_i &= \left(\frac{d}{dt} + \lambda_i\right) e_i = \dot{e}_i + \lambda_i e_i \\ e_i &= \hat{q}_i - \hat{q}_{id} \end{aligned} \tag{51}$$

Following the sliding mode controller design described in Eqs. (20) to (29), the command input, $\hat{Q}_{com}$, is obtained in the robot workspace coordinate system as follows:

$$\hat{Q}_{com} = (B_{est})^{-1}[-f_{est} + \ddot{\hat{q}}_{des} - \lambda\dot{e} - K\ \mathrm{sgn}(s)] \tag{52}$$

In Eq. (52), $(.)_{est}$ describes the estimation from the system's dynamics vector or the input gain matrix. These estimations are usually obtained from the mathematical modeling of the system's dynamics. As mentioned before, uncertainties in the system's dynamics and uncertainties in the input gain matrix are considered in the controller design procedure. In Eq. (52), λ and K are the diagonal matrices with positive components. Matrix λ is determined heuristically with respect to the system's behavior in the transient and steady state. Based on various case studies, this matrix can be chosen at least five times more than the sliding condition parameter described in Eq. (27) to obtain acceptable performance. K is the switching matrix, and its components are determined from Eq. (29). The sgn (s) is a column vector, with its components being one multiplied by the "sign" function of the corresponding algebraic distance from the sliding surface:

$$\mathbf{sgn(s)} = \left[\mathrm{sgn}(s_1)\ \mathrm{sgn}(s_1)...\mathrm{sgn}(s_n)\right]^T \tag{53}$$

With a transform made by the corresponding Jacobian matrix, the command input described in the robot workspace can be mapped to the joint space (generalized coordinate) as follows:

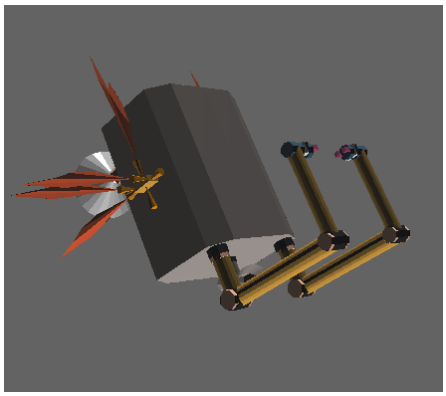

Fig. 2. A three-manipulator/appendage SFFR.

$$\begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \\ \tau_{K\times 1} \end{Bmatrix} = (\mathbf{J}_Q)^{-1} Q_{com} \tag{54}$$

Finally, the control input in the generalized coordinate space is obtained as

$$Q_{com} = (J_{\hat{q}})^T (B_{est})^{-1}[-f_{est} + \ddot{\hat{q}}_{des} - \lambda\dot{e} - K\ \mathrm{sgn}(s)] \tag{55}$$

Next, a 14-DOF SFFR is simulated to compare the performance of the new proposed RSMC algorithm and the conventional SMC.

7. Simulation results

In this section, the simulation results for several cases are presented. In the first step of simulation, we evaluate the performance of the new proposed SMC algorithm (i.e., RSMC along sinusoidal desired trajectories). As mentioned before, a PEM parameterized model is used in the control algorithm instead of a mathematical model. The performance of the prediction mechanism is shown in all simulations. Afterwards, the performance of the new proposed RSMC algorithm is evaluated through simulation and compared with the conventional SMC. The task aims to capture a moving object based on the planned trajectories. The system is a 14-DOF SFFR, as shown in Fig. 2 and described in [15]. The geometric and mass properties are given to the SPACEMAPLE code to obtain the dynamics model of the SFFR in a relevant symbolic format [2].

The actuators acting at the base of the SFFR are primarily assumed as continuous servos with saturation limits. In addition, they are subsequently considered as on–off actuators to illustrate the merits of the new proposed algorithm. Clearly, the actuators at the joints of the manipulators are continuous servos.

7.1 Servo actuators model

In the active control of SFFR, actuators have various per-

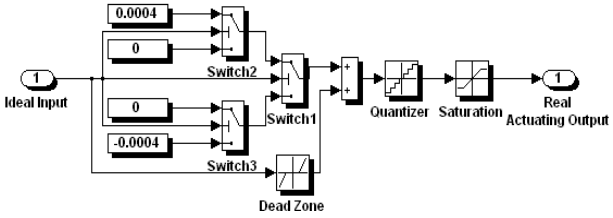

Fig. 3. Servo actuators model in simulink.

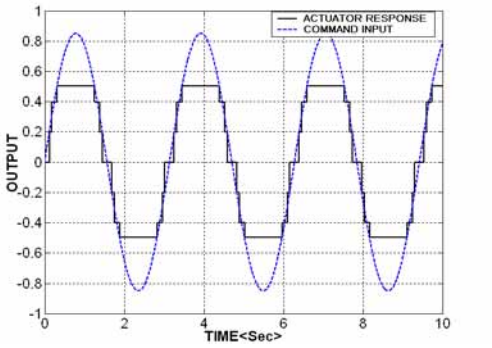

Fig. 4. Actuator response corresponding to a sinusoidal control demand.

formance characteristics. Real actuators cannot generate the exact command input sent from the controller, which in turn deteriorates the system performance. In summary, the most important problems are the following:

(1) Upper-bound value (saturation limit) that can be generated by the actuator

(2) Actuator resolution, namely, the capability of the actuator to track a continuous input command with the same kind of smoothness

(3) The finest value of input that can be generated by the actuator (input threshold)

The actuator with the above considerations, as modeled in Simulink, is shown in Fig. 3. Three switches plus a dead zone provide the threshold of input, that is, if the threshold of the input is 0.0004 N.m., the threshold of switches 1, 2, and 3 must be 0, 0.0004, and –0.0004, respectively (Fig. 3). To illustrate the actuator performance, a sample response is shown in Fig. 4. Studies on using on–off actuators in such systems can be found in [14].

In the first step of simulations, that the desired trajectories of the SFFR are arbitrary sinusoidal functions, as shown in Fig. 5, is assumed. The frequencies and amplitudes of each subsystem are different from the other sub-systems. In Fig. 6, two samples of the estimated distances from the sliding surfaces are shown. To show the chattering phenomenon and the performance of the ERP mechanism, an appropriate magnification is also conducted. The figure shows that the chattering drastically becomes calm due to the activity of the new SMC algorithm. The quality of the other sub-systems in the distance from the sliding surface is similar to that in Fig. 6. To show chattering in control inputs, the applied moments to the space-

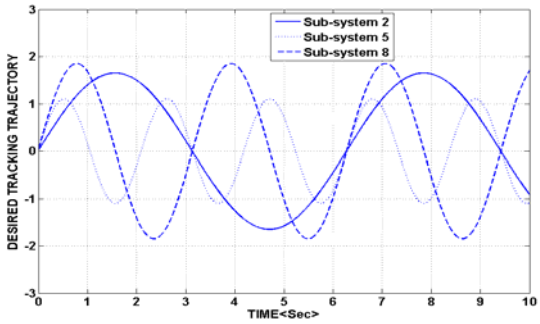

Fig. 5. Three desired trajectories for sub-systems 2, 5, and 8.

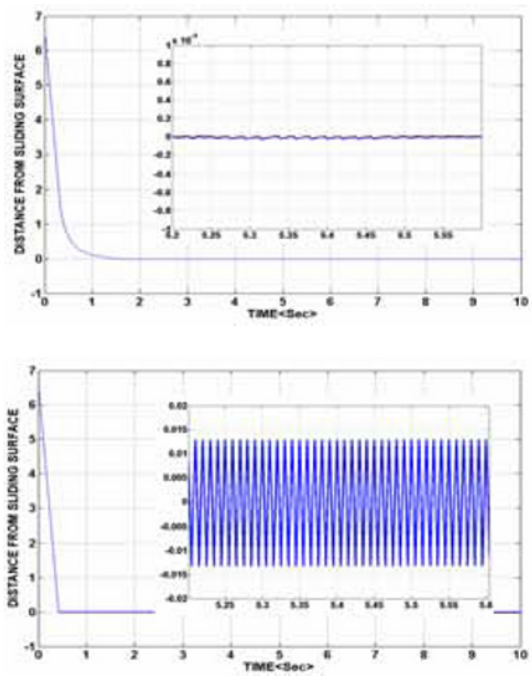

Fig. 6. Distance from the corresponding sliding surface and its magnification to show the chattering phenomenon (Up: RSMC; Down: CSMC).

craft is shown in Fig. 7, whereas one of the joints of the manipulators is shown in Fig. 8. The new proposed algorithm has a significant effect in decreasing chattering. Fig. 9 shows the results obtained from the recursive PEM. The orders of the model have been chosen as [2 3 1 1 1], and thus the number of parameters that must be evaluated recursively is 8. In the RSMC, due to the lesser intensity of chattering, the estimated parameters have smooth variations in comparison with those in CSMC. In Table 1, the mean values and variances of the estimation error are shown.

8. Trajectory planning to capture a moving target in space

In this section, the SMC control is applied to the SFFR along a designed trajectory to pursue a target. The escaping target that must be captured by the SFFR travels through

Table 1. Mean values and variances of estimation error for two controllers.

	Mean value			Variance		
Sub-system	(2, 7)	(5, 12)	(8, 14)	(2,7)	(5, 12)	(8, 14)
CSMC	0.0148	0.0096	0.0116	0.0712	0.0474	0.0409
RSMC	0.0122	0.0087	0.0082	0.0711	0.0474	0.0407

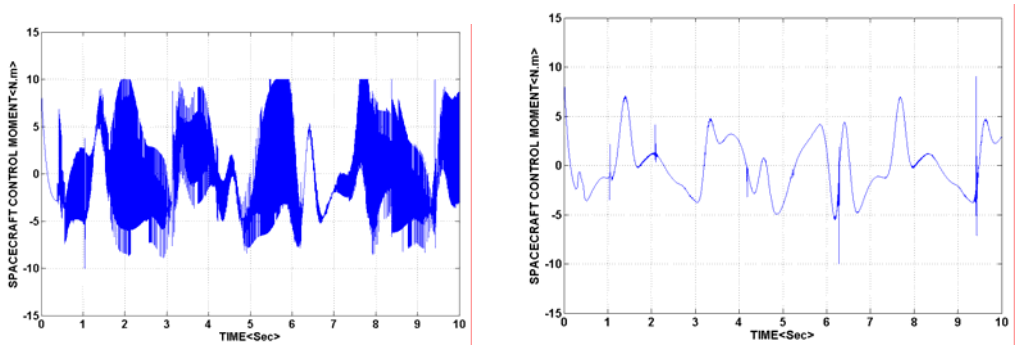

Fig. 7. Applied moment to the spacecraft (base) of the SFFR (Up: CSMC; Down: RSMC).

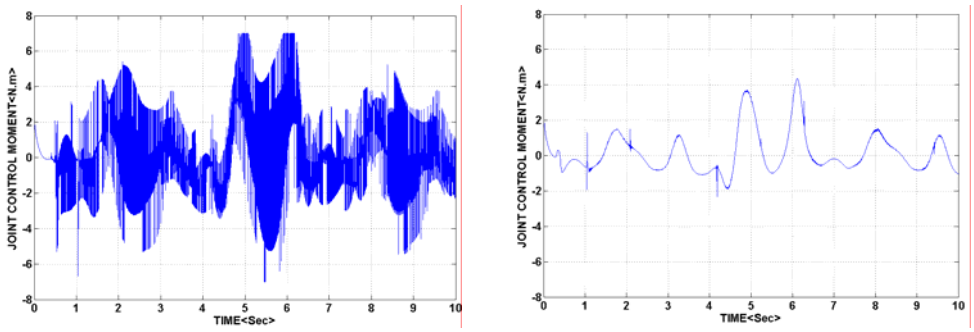

Fig. 8. Applied torque to the joint of the SFFR manipulator (Up: CSMC; Down: RSMC).

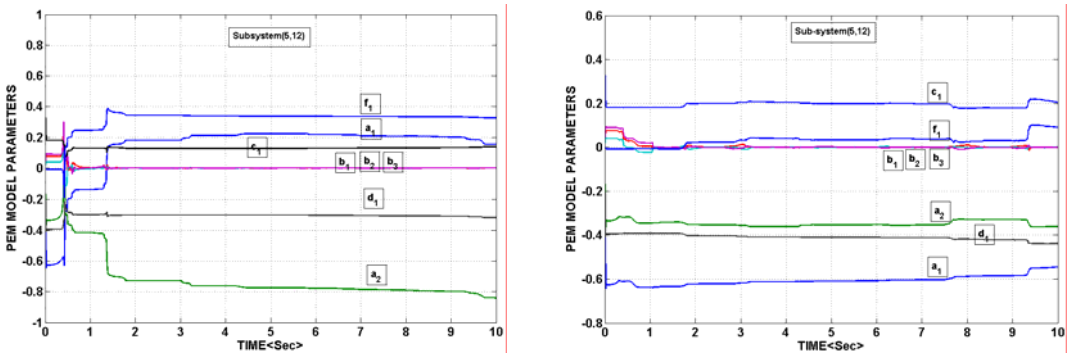

Fig. 9. Estimated parameters for the PEM model using recursive Kalman filter for the direct model between actuators' output and the measurements of robot sensors [Top: CSMC system; Bottom: RSMC system; sub-system (5,3)].

three-dimensional spaces with a constant velocity vector, as is the case for a passive object. The coordinated motion of the given SFFR is planned as follows:

The control of the spacecraft Euler angles that make the symmetric motion for the two end-effectors throughout the capturing mission and the control of the distance between the spacecraft's center of mass and the moving object fulfill the dexterous motion of the end-effectors. Upon reaching the moving object, the manipulators remain in their home configuration, except the antenna that must have a controlled orientation throughout the maneuver. When the object is reachable by end-effectors, the manipulators start moving to capture the object, while the control system must retain the desired Euler angles and distance between the spacecraft and the target.

Fig. 10 shows the tracking position errors for the two end-effectors. Until $t \approx 40 < Sec >$, the manipulators remain in their home configuration; the locks of the joints are released; and at this time their initial tracking error appears while time

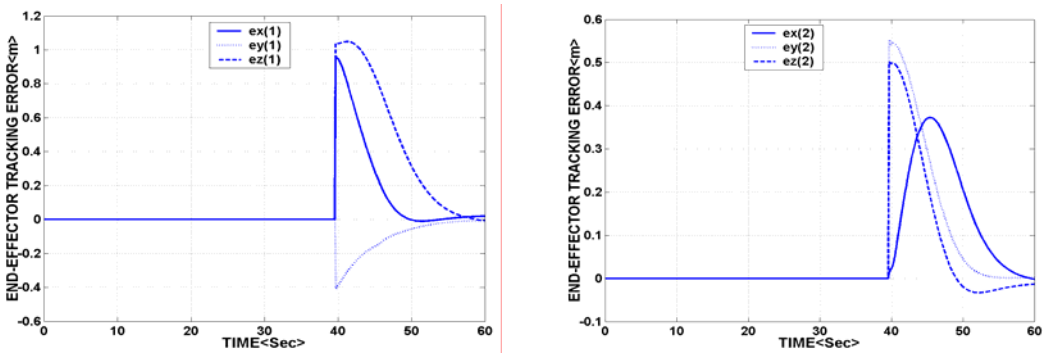

Fig. 10. Tracking position errors of the SFFR end-effectors (Left: First manipulator; Right: Second manipulator).

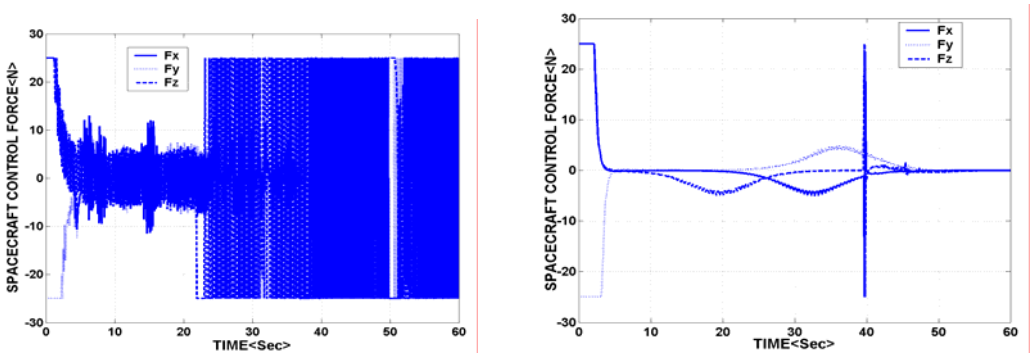

Fig. 11. Control forces applied to the base of the SFFR (Left: Conventional SMC; Right: RSMC).

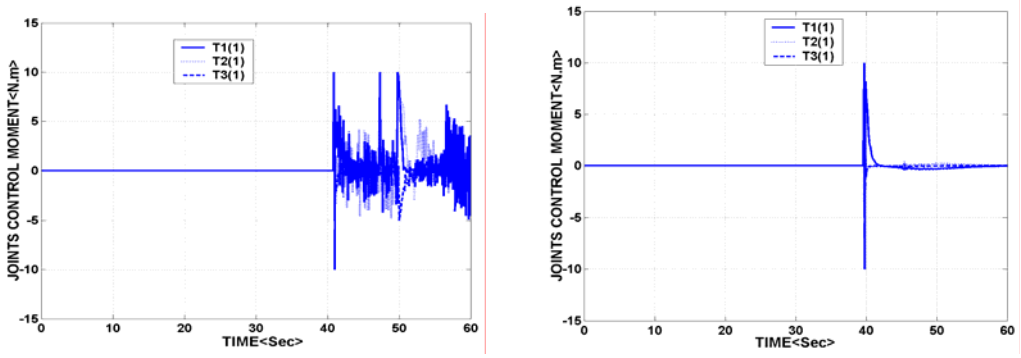

Fig. 12. Control torques applied to the first manipulator (Left: Conventional SMC; Right: RSMC).

vanishes. The tracking position errors are calculated in terms of the reference coordinate components. Tracking errors for both conventional SMC and RSMC algorithms are almost the same. However, in the new RSMC algorithm, the chattering phenomenon is reduced drastically, but the performance of the controller remains similar to the conventional SMC. Chattering can be observed in the control inputs as shown in Fig. 11.

To show the effect of the new chattering elimination algorithm, RSMC, the control forces applied to the base of SFFR (spacecraft), are shown in Fig. 11. The RSMC algorithm has a significant effect on the decrease in amplitude of the chattering phenomenon. The control moments applied to the base of the SFFR are similar to the forces shown in Fig. 11. In Fig. 12, the control torques applied to the joints of the first manipulator are shown, where those of the second manipulator are similar.

For comparison, *Chattering Intensity Factor (CIF)* is defined as follows:

$$CIF = \frac{\text{Energy Consumed under application of } a \text{ Control Law}}{\text{Nominal Required Value of Energy}} \tag{56}$$

where the nominal value is obtained by solving the inverse dynamics. This factor serves as a reference to compare the control algorithms in terms of energy consumption and in turn the chattering characteristics. In Fig. 13, the total energy consumed by both conventional SMC and RSMC algorithms is plotted. In the RSMC algorithm, the total consumed energy is considerably decreased compared with the conventional SMC due to the lower amplitude of the chattering. To show the

Table 2. Consumed energy by the conventional SMC and the new RSMC compared with the nominal required value of energy.

	SMC	RSMC	Nominal value
Consumed energy (Joule)	317.0090	53.3674	18.4534
CIF	17.1789	2.8920	1.00

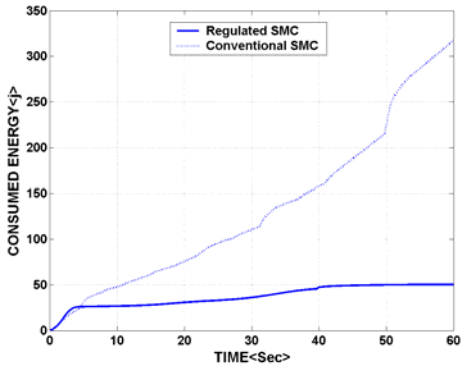

Fig. 13. Total energy consumed by the control system (Dashed line: Conventional SMC; Solid line: RSMC).

chattering elimination quantity, CIF values are compared in Table 2. The nominal value of energy consumption is obtained from inverse dynamics based on the given trajectories. The RSMC algorithm consumes approximately 2.9 times the nominal value, whereas the conventional SMC consumes 17.2 times the nominal value. These results show the major advantage of the new SMC algorithm in lowering the actuators' activity. In the next step, to show the effectiveness of the new proposed algorithm in handling the system in the presence of vigorous limitations in actuators, the force or moment actuators of the spacecraft are replaced by on–off ones.

9. Application of the new RSMC algorithm to the SFFR with on-off actuators at the base

In this step, that the actuators of the SFFR are of the on–off type to avoid delivering exactly the commanded control input produced by the control algorithm is assumed. To plan the law of on–off actuators, that the chattering-eliminated SMC commands are suitable signals due to the avoidance of control input oscillations around zero is proposed [13]. The model for the actuators is shown in Fig. 14. In this model, the constant amplitude is determined based on the average calculation of the required force or moment obtained from inverse dynamics. In Fig. 15, the tracking errors of the end-effectors for the RSMC are shown. Running several simulations with various control parameters shows that for such a complicated highly nonlinear dynamic system with on–off actuators, the conventional SMC cannot successfully control the system. This shows the absolute advantage of the new proposed RSMC compared with the conventional SMC. Fig. 16 presents the control force or moment in the x-direction. The other control inputs at the base are similar to those shown in Fig. 16, whereas joint torques are almost the same as those presented in the

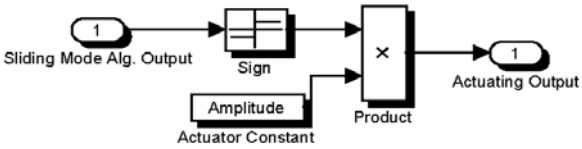

Fig. 14. Model for the on–off type actuators.

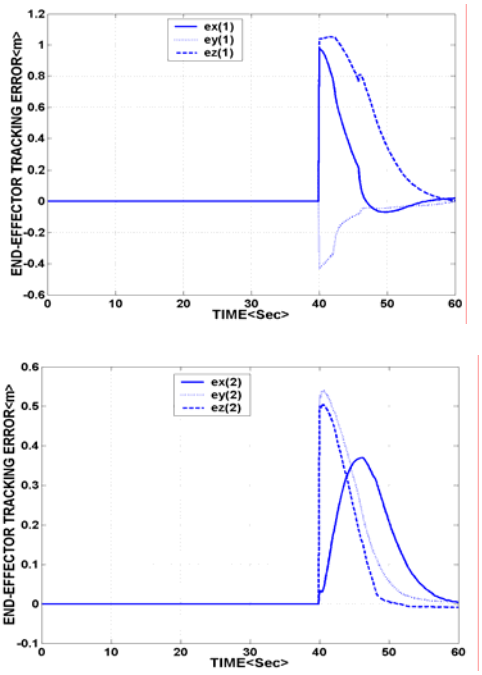

Fig. 15. Tracking position errors of the SFFR end-effectors using on–off actuators and applying RSMC (Left: First manipulator; Right: Second manipulator).

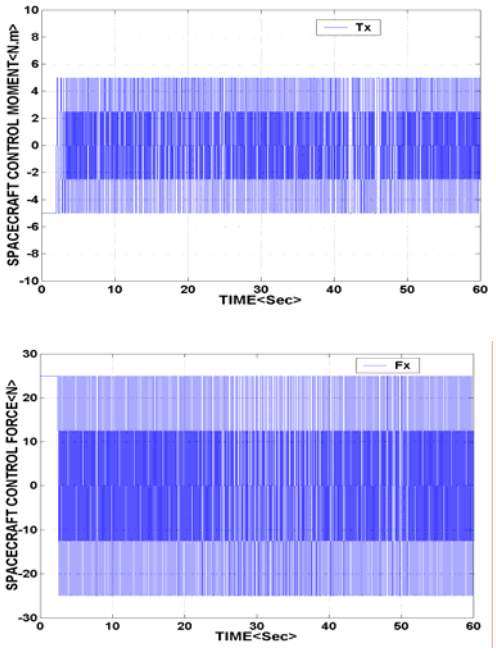

Fig. 16. Control force and control torques applied to the base of the SFFR by on–off actuators (Left: Control torque about x direction; Right: Control force along x direction).

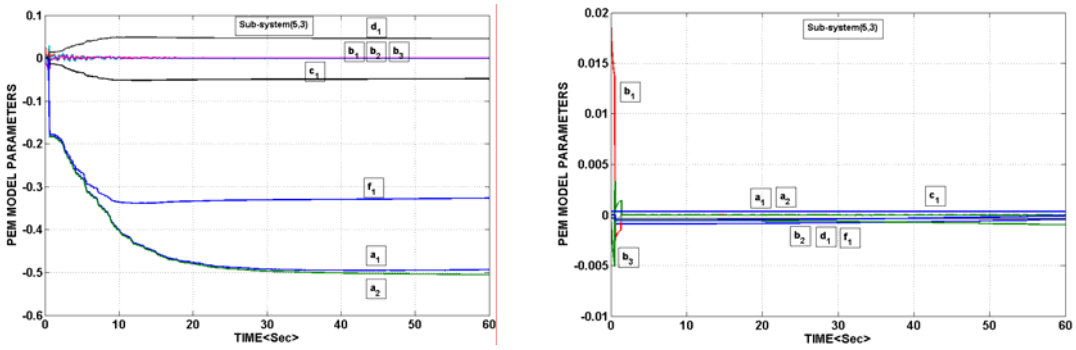

Fig. 17. Estimated parameters for the PEM model using the recursive Kalman filter for the direct model between actuators' output and the measurements of robot sensors (Top: CSMC system; Bottom: RSMC system; sub-system (5,3); continuous servos in the base).

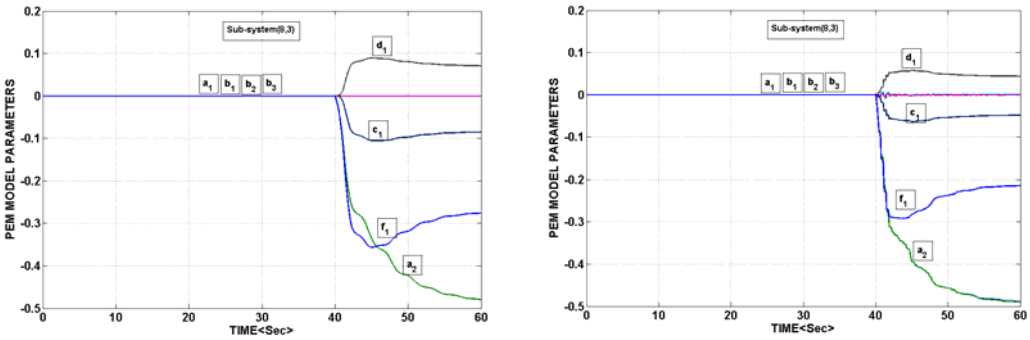

Fig. 18. Estimated parameters for the PEM model using the recursive Kalman filter for the direct model between actuators' output and the measurement of robot sensors (Top: CSMC; Bottom: RSMC; sub-system (8, 3); continuous servos in the base).

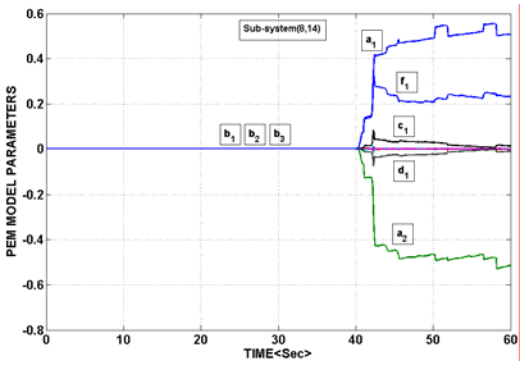

Fig. 19. Estimated parameters for the PEM model using the recursive Kalman filter for the direct model between actuators' output and the measurement of robot sensors (discontinuous servos at the base).

previous case. Fig. 17 plots the estimated parameters in the SFFR maneuver and shows the parameters for the sub-system (5, 3), that is, the system indicating that its input is the force along the z-direction and its output is the fifth distance from the sliding surface. Fig. 18 depicts the parameters for sub-system (8, 3) during the flight. As mentioned previously, until $t \approx 40\text{sec}$, the manipulators remain in their home configuration; afterwards, the locks of the joints are released, and their parameter tuning begins. This continues until both the estimation and tracking errors approach zero.

10. Conclusions

In this paper, an efficient chattering-eliminated sliding mode control algorithm is introduced and applied to an SFFR as a highly nonlinear coupled system. A chattering phenomenon results in significant energy dissipation and causes practical difficulties for actuators. To fulfill stability requirements, robustness properties, and chattering elimination, an RSMC algorithm is proposed to determine proper positive values for the coefficient of sliding condition. To solve the run-time problem, an explicit direct relationship between the SFFR's output of actuators (force or torque) and the measurement of distances from the corresponding sliding surfaces is assumed.

The model is parameterized optimally using the PEM. To reach acceptable performance, the parameters are estimated recursively using the Kalman filter as a parameter estimator. To control the orientation and position of the SFFR, a multi-input sliding mode control law is designed to catch a moving target using the estimated model in the control algorithm. The new approach alleviates the chattering trend. The obtained results show that the proposed regulated sliding mode controller can significantly alleviate the chattering trend, consequently substantially decreasing energy consumption. The run time of the control algorithm in all simulations is also ascertained to decrease completely, while both estimation and tracking errors remain in their acceptable range.

Acknowledgment

The authors would like to thank Prof. S. Ali A. Moosavian for his detailed comments and suggestions.

References

[1] S. Ali A. Moosavian and E. Papadopoulos, On the Kinematics of multiple manipulator space free-flyers, *Journal of Robotic Systems*, 15 (4) (1998) 207-216.

[2] S. Ali A. Moosavian and E. Papadopoulos, Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPL, *Journal of Advanced Robotics*, 18 (2) (2004) 223-244.

[3] J. L. Kuang, B. J. Kim, H. W. Lee and D. K. Sung, The attitude stability analysis of a rigid body with multi-elastic appendages and multi-liquid-filled cavities using the chateau method, *Journal of the Astronautical Space Sciences*, 15 (1) (1998) 209-220.

[4] H. W. Mah, V. J. Modi, Y. Morita and H. Yokota, Dynamics during slewing and translational maneuvers of the space station based MRMS, *Journal of the Astronautical Space Sciences*, 38 (4) (1990) 557-579.

[5] E. Papadopoulos and S. Ali A. Moosavian, Dynamics & control of space free-flyers with multiple arms, *Journal of Advanced Robotics*, 9 (6) (1995) 603-624.

[6] S. Ali. A. Moosavian, R. Rastegari and E. Papadopoulos, Multiple impedance control for space free-flying robots, *AIAA Journal of Guidance, Control, and Dynamics*, 28 (5) (2005) 939-947.

[7] S. Ali A. Moosavian and E. Papadopoulos, On the control of space free-flyers using multiple impedance control, Proc. IEEE Int. Conf. Robotics Automation, NM, USA, (1997) 853-858.

[8] S. Dubowsky and D. T. Des Forges, The application of model-referenced adaptive control to robotic manipulators, *ASME Journal of Dynamic Systems, Measurement & Control*, 101 (1979) 193-200.

[9] J. J. E. Slotine and W. Li, Adaptive robot control: a new perspective, Proc. of the 26th IEEE Conf. on Decision and Control, Los Angeles, CA, USA, (1987) 49-59.

[10] K. Youcef-Toumi and O. Ito, Controller design for systems with unknown dynamics, Proc. of American Cont. Conf., Minneapolis, USA, (1987) 836-844.

[11] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, (1991).

[12] J. Y. Hung, W. Gao and J. C. Hung, Variable structure control: a survey, *IEEE Transactions on Industrial Electronics*, 40 (1) (1993) 2-22.

[13] A. S. Mollah, S. M. Ullah, L. K. Wong, F. H. F. Leung and P. K. S. Tam, A chattering elimination algorithm for sliding mode control of uncertain non-linear systems, Elsevier Science Ltd, *Mechatronics,* 8 (1998) 765-775.

[14] S. Ali A. Moosavian and M. Reza Homaeinejad, Regulated sliding mode control of satellite rotation, Proc. of the IFAC workshop on Generalized Solutions in Control Problems (GSCP), Russia, (2004) .

[15] S. Ali A. Moosavian and E. Papadopoulos, Control of space free-flyers using modified transpose Jacobian algorithm, Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, France, (1997) 1500-1505.

[16] S. Ali A. Moosavian and M. Reza Homaeinejad, Regulated sliding mode control of robotic manipulators, Proc. of the Tehran International Congress on Manufacturing Engineering (TICME- 2006), Tehran, Iran, (2006).

[17] S. Ali A. Moosavian and M. Reza Homaeinejad, Variable structure control (VSC) of robotic manipulators, Proc. of the IEEE Workshop on Advanced Motion Control (AMC), Istanbul, Turkey, (2006).

[18] S. Ali A. Moosavian, Dynamics and control of free-flying robots in space: a survey, Proc. of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles- IAV2004, Lisbon, Portugal, (2004).

[19] D. Wang and G. Xu, Full-state tracking and internal dynamics of nonholonomic wheeled mobile robots, *IEEE/ASME Transactions on mechatronics*, 8 (2) (2003) 203-214.

[20] S. M. Savaresi, F. Previdi, A. Dester, S. Bittanti and A. Ruggeri, Modeling, identification, and analysis of limit-cycling pitch and heave dynamics in an ROV, *IEEE Journal of Oceanic Engineering*, 29 (2) (2004) 407-417.

[21] I. Markovsky, J. C. Willems, S. V. Huffel, B. DeMoor and R. Pintelon, Application of structured total least squares for system identification and model reduction, *IEEE Transactions on Automatic Control*, 50 (10) (2005) 1490-1500.

[22] M. Storace and O. DeFeo, Piecewise-linear approximation of nonlinear dynamical systems, *IEEE Transactions on Circuits and Systems*, 51 (4) (2004) 830-842.

[23] J. X. Xu, Y. J. Pan and T. H. Lee, Sliding mode control with closed-loop filtering architecture for a class of nonlinear systems, I*EEE Transactions on Circuits and Systems*, 51 (4) (2004) 168-173.

[24] J. O. Hahn, R. Rajamani, S. H. You and K. I. Lee, Real-time identification of road-bank angle using differential GPS, *IEEE Transactions on Control Systems Technology*, 12 (4) (2004) 589-599.

[25] S. Oucheriah, Exponential stabilization of linear delayed systems using sliding mode controllers, *IEEE Transactions on Circuits and Systems*, 50 (6) (2003) 826-830.

[26] S. Chen, X. Hong, C. J. Harris and X. Wang, Identification of nonlinear systems using generalized kernel models, *IEEE Transactions on Control Systems Technology*, 13 (3) (2005) 401-411.

[27] M. Azimi, P. Nasiopoulos and R. K. Ward, Offline and online identification of hidden semi-markov models, IEEE *Transactions on Signal Processing*, 53 (8) (2005) 2658-2663.

[28] V. Smídl and A. Quinn, Mixture-based extension of the AR model and its recursive Bayesian identification, IEEE *Transactions on Signal Processing*, 53 (9) (2005) 3530-3542.

[29] X. M. Ren, A. B. Rad, P. T. Chan and W. L. Lo, Online identification of continuous-time systems with unknown time delay, *IEEE Transactions on Automatic Control*, 50 (9) (2005) 1418-1422.

[30] N. Diolaiti, C. Melchiorri and S. Stramigioli, Contact impedance estimation for robotic systems, *IEEE Transactions on Robotics*, 21 (5) (2005) 925-935.

[31] L. Y. Wang, J. F. Zhang and G. G. Yin, System identification using binary sensors, *IEEE Transactions on Automatic Control*, 48 (11) (2003) 1892-1907.

[32] W. Spinelli, L. Piroddi and M. Lovera, On the role of prefiltering in nonlinear system identification, *IEEE Transactions on Automatic Control,* 50 (10) (2005) 1597-1602.

[33] J. Vörös, Identification of Hammerstein systems with time-varying piecewise-linear characteristics, IEEE Transactions on Circuits and Systems, 52 (12) (2005) 865-869.

[34] L. Ljung and T. Glad, *Modeling of Dynamic Systems*, Prentice Hall, Englewood Cliffs, N. J., (1994).

[35] L. Ljung, *System Identification-Theory for the User*, Prentice Hall, Upper Saddle River, N. J., 2nd edition, (1999).

[36] T. Söderström and P. Stoica, *System Identification*, Prentice Hall International, London. (1989).

Hamid Khaloozadeh was born in Mashhad, Iran on January 1965. He received the B.S. degree in control engineering from Sharif University of Technology, Iran in 1990, the M.Sc. degree in control engineering from K.N.Toosi University of Technology, Iran in 1993, and the Ph.D. degree in control and system engineering from Tarbiat Modarres University, Iran in 1998. Since 1998 to 2004 he was a faculty member at Ferdowsi University of Mashhad. He is currently an Associate Professor and teaches in the Department of Electrical and Computer Engineering in K.N.Toosi University of Technology. His research interests include system identification, optimal control, adaptive control, stochastic estimation and control, and time series analysis.