Model Identification and Control Design for a Humanoid Robot

Wei He, *Senior Member, IEEE,* Weiliang Ge, Yunchuan Li, Yan-Jun Liu, *Member, IEEE*, Chenguang Yang, *Senior Member, IEEE*, Changyin Sun

***Abstract*—In this paper, a model identification and adaptive control design is performed on a humanoid robot based Devanit-Hartenberg (D-H) model. We focus on the modelling of the 6 degree-of-freedom (DOF) upper limb using recursive Newton-Euler (RNE) formula for the coordinate frame of each joint. To obtain sufficient excitation of the robot for modelling, the particle swarm optimization (PSO) method has been employed to optimize the trajectory of each joint, such that satisfied parameter estimation can be obtained. In addition, the estimated inertia parameters are taken as the initial values of the RNE based adaptive control design to achieve improved tracking performance. Simulation studies are carried out to verify the result of the identification algorithm and to illustrate the effectiveness of the control system.**

***Index Terms*—Humanoid Robot, Model Identification, Recursive Newton-Euler Formulation, Adaptive Control, Particle Swarm Optimization**

I. Introduction

With the rapid development of the robot industry and the urgent need of a wide range of robot functions, efforts spent on robotics research have increased significantly in recent decades. Modern robots are expected to perform various functions in addition to the fundamental functions like walking or speaking. The higher level is the functional task, which includes barrier avoidance, walking up stairs, face recognition, object localization and pattern learning [1]. Robots with upper limbs are generated in this environment where the artificial intelligent technology has explosive improvement.

In the research field of the model identification, quite a lot works have been carried out. Investigators have developed various methods for the model identification of robot manipulators [2]. The robot manipulators are designed and processed according to precise kinematic specifications. In recent years, the kinematics of robot manipulator has been intensively investigated [3]. The model parameters can be computed from the CAD/CAM database, but it is rarely adopted because of the unclear accuracy [4]. To improve the accuracy, the characteristic-equation based method is proposed in [5]. The dynamic equations of kinematic model of rigid bodies are derived by the NE or Lagrangian method [6], [7]. To solve the elastic-deformable or other complex problems, the finite element method (FEM) can be applied [8], [9]. In recent decades, the artificial intelligence (AI) method has been developed dramatically, and has been used in many fields [10]. The model parameters can also be identified by AI method, such as artificial neural network (ANN) [11]. In this paper, the model identification method is based on NE principle, and the identification trajectory is generated by another AI method, the particle swarm optimization (PSO) [12].

When a humanoid robot picks up or puts down something, the load of the manipulator changes significantly. This may cause the robotic control system unstable. To resolve this problem, the adaptive control scheme is proposed [13], [14], [15], [16], [17]. Fuzzy adaptive control is investigated in [18], [19], [20], [21], [22]. Many special conditions like input with dead-zone, output with constraint or discrete-time systems are considered in [23], [24], [25], [26], [27], [28], [29], [30]. In this way, the control system can adjust the physical parameters to the real uncertain condition [31], [32], [33]. In recent two decades, many approaches in adaptive control field have been studied and developed. Under the condition that the output is measurable, the adaptive neural network control is proposed in [34], [35], [36], [37], [38], [39], [40], [41], [42], [43]. Using the fuzzy logic theory, the behaviors of the unknown and uncertain dynamics of the robot can be learnt by fuzzy logic systems (FLS) [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54]. Under the condition of motion disturbances and parametric uncertainties, the robust control theory is applied to the adaptive control to improve the stability of the robot system [55], [56], [57], [58], [59].

However, because of the computational complexity of these approaches, such abundant theoretical and experiment results fail to improve the practical applicability of adaptive control algorithms, especially for manipulators with more than 4 DOFs. In this paper, we study the dynamical modeling and control design for the upper limb manipulator of a biped humanoid robot, HUBO, designed by Korea Advanced Institute of Science and Technology (KAIST). The internal parameters of upper limb of HUBO are identified. Based on the estimated

W. He is with the School of Automation and Electrical Engineering and the Key Laboratory of Advanced Control of Iron and Steel Process, University of Science and Technology Beijing, Beijing 100083, China. (Email: weihe@ieee.org)

W. Ge and Y. Li are with the School of Automation Engineering and Center for Robotics, University of Electronic Science and Technology of China, Chengdu 611731, China.

Y.-J. Liu is with the School of Sciences, Liaoning University of Technology, Jinzhou 121001, China.

C. Yang is with the Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea, SA1 8EN, United Kingdom. (Email: cyang@theiet.org)

C. Sun is with the School of Automation, Southeast University, Nanjing 210096, China.

This work was partially supported by the National Natural Science Foundation of China under Grant 61522302, 61533008, 61520106009, the National Basic Research Program of China (973 Program) under Grant 2014CB744206, the Engineering and Physical Sciences Research Council under Grant EP/L026856/1 and the Fundamental Research Funds for the China Central Universities of USTB under Grant FRF-TP-15-005C1.

parameters, the manipulator can track the desired trajectory efficiently via the proposed control.

As shown in Fig. 2, there are 6 DOFs in the upper limb of HUBO, i.e. 3 DOFs in the shoulder (yaw, pitch and roll), 2 DOFs in the elbow (yaw and pitch) and 1 DOF in the wrist (pitch). In order to avoid the computation complexity, the recursive algorithm based on the NE formulation is proposed [60]. In this recursive algorithm, the linear in parameters (LIPs) process can be computed in recursive algorithm based on NE formulation. In this way, the computation complexity of the adaptive control algorithm can be reduced dramatically. No matter how many DOFs the manipulator has, the handling methods are always same, except the processing time varies with the variation of DOF.

II. Problem Formulation and Preliminaries

A. Problem Formulation

Fig. 1 illustrates the structure of the model identification system. The aim of the model identification is to identify the dynamic model of the upper limb of the humanoid robot, which provides the foundation of control with high accuracy. It consists of two sub-modules.

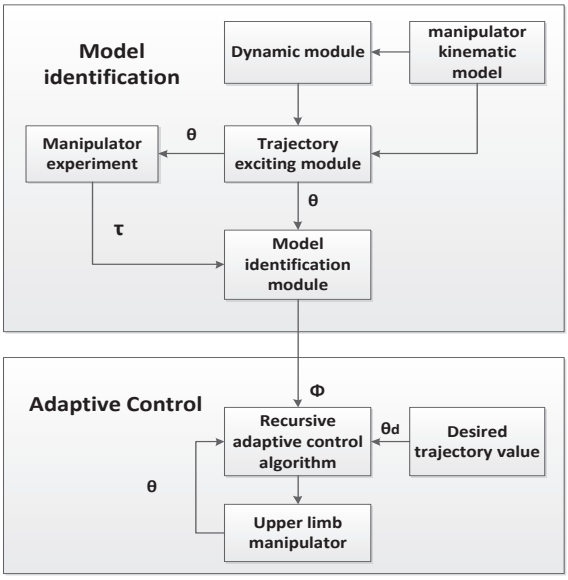

Fig. 1. System structure

In the model identification module, the robot kinematic model is treated as the input of the dynamic modeling sub-module, whose output is the dynamic model, which is also the input of trajectory planning sub-module. A family of exciting trajectories are generated by this module. These trajectories can be used in simulation experiments and actual robot experiments to collect the kinematic data and torque for the model identification. For model identification submodule, the kinematic configuration information of the corresponding robot, the experiment obtained data along with the exciting trajectories are the input. And then the identified inertia parameters of the model are given as the output.

In the adaptive control module, the desired trajectory generator is used to generate a trajectory which can be obtained with respect to the actual application condition and environment. The adaptive control is the kernel sub-module of this part. The adaptive control law is designed in this sub-module. Based on the desired and measured position and velocity of the robot joints, the adaptive control system can generate the actuator commands to the robot plant. Meanwhile a recursive algorithm based on NE formulator is used to reduce the algorithm complexity due to the high DOF.

B. Preliminaries

PSO method is a kind of group algorithm, which is based on the behavior of a group of living beings. Each particle in the group can move to a better place based on the adaptive degree. In PSO algorithm, each particle is assumed as a zero volume point flying in an m-dimension searching space. The flying speed can be adjusted based on its own and others' experience. We can suppose that the real time position of particle *i* is x_i and the best position is p_i. The best position of the group is p_g. The velocity of particle *i* is v_i. During each update cycle, both p_i and p_g are updated. The particle position and velocity can be updated as [61], [62],

$$v_i^{t+1} = c_0 v_i^t + c_1 r_1 (p_i - x_i^t) + c_2 r_2 (p_g - x_i^t), \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (2)$$

where c_0 is the inertial constant; c_1 and c_2 are the acceleration constant; r_1 and r_2 are the random number between 0 and 1, $c_0 v^t$ represents the influence of the particle velocity, $c_1 r_1 (p_i - x_i^t)$ represents the influence of individual experience and $c_2 r_2 (p_g - x_i^t)$ represents the influence of group experience, which is the cooperation of each particle. Based on the constriction factor method [63], the parameters c_0 is chosen as 0.72894 and c_1=c_2=1.49618. Using PSO algorithm, the analysis time can be speeded up without introducing extra error [64].

Assumption 1: [65] f(D(z,ξ))≤ f(z) and if $\xi \in$ S, then f(D(z,ξ))≤ f(ξ), where ξ_k is the vector generated from sample space $(\mathbb{R}^n, B, \mu_k)$; B is the σ region of the subset in $\mathbb{R}^n$; μ_k is the probability measure in B and D is the recursive method.

Theorem 1: [65] Any recursive satisfies the Assumption 1, has the convergence property.

Theorem 2: [66] Given the non-linear dynamic systems

$$\dot{x} = f(x,t), x(0) = x_0, \quad (3)$$

with an equilibrium point at the origin, and let N be a neighborhood of the origin, then origin **O** is stable in the sense of Lyapunov if for $x \in N$ there exists a scalar function $V(x,t) > 0$ and $\dot{V} \leq 0$.

III. Modeling and Identification

In this section, D-H model of the manipulator system transform matrix are given firstly. Secondly, Newton-Euler formulation is exploited to derive the model of manipulator. Then, the trajectories are excited and parameters are optimized using PSO algorithm.

A. Kinematic Modeling

To analyze the kinematics of the system, the model can be viewed as a three-link structure shown in Fig. 2. T_i are the coordinate frames of the joints. Using the positive direction of coordinate and the modified D-H model, the link transformation matrix ${}^{i-1}_{i}T$ can be obtained [67]:

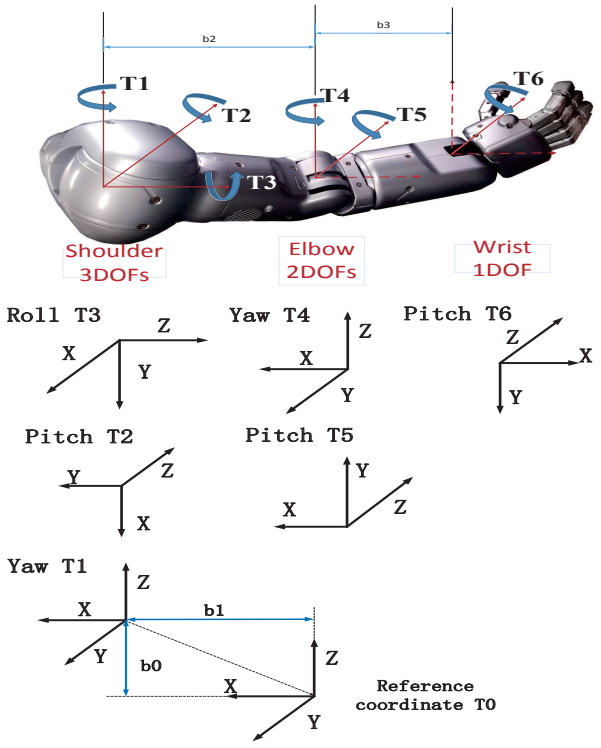

Fig. 2. Upper limb manipulator

Remark 1: T_1 to T_6 denote joint 1 to joint 6, respectively, and T_0 represents the reference coordinate, which is located at the center of chest.

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\theta_i d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4) $$

which, for short, can be written as (5)

$$ {}^{i-1}_{i}T = \begin{bmatrix} {}^{i-1}_{i}R & {}^{i-1}_{i}P \\ \mathbf{0} & 1 \end{bmatrix}. \quad (5) $$

According to the coordinate frame shown in figure 1, the D-H parameters can be obtain [67], which is shown in Table I.

Remark 2: b_1, b_2 and b_3 denote the distance between the center of chest and shoulder, the distance between the shoulder and elbow and the distance between elbow and wrist, respectively.

B. Dynamic Modeling

From the modified D-H model and the transform matrix discussed above, we may calculate the angular velocity ω_i, the angular acceleration $\dot{\omega}_i$ and the line acceleration $\dot{v}_i$:

TABLE I
D-H MODEL PARAMETER

$Joint$	a_i	α_i	d_i	θ_i
1	b_1	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
2	0	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
3	0	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
4	b_2	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
5	0	$\frac{\pi}{2}$	0	0
6	b_3	0	0	π

$$ \begin{aligned} \omega_i &= {}^{i-1}_{i}R\omega_{i-1} + z_i\dot{\theta}_i, \\ \dot{\omega}_i &= {}^{i-1}_{i}R\dot{\omega}_{i-1} + {}^{i-1}_{i}R\omega_{i-1} \times z_i\dot{\theta}_i + z_i\ddot{\theta}_i, \\ \dot{v}_i &= {}^{i-1}_{i}R[\dot{v}_{i-1} + \dot{\omega}_{i-1} \times {}^{i-1}_{i}P + \omega_{i-1} \times (\omega_{i-1} \\ &\quad \times {}^{i-1}_{i}P)], \end{aligned} \quad (6) $$

where the initial values of the angular velocity, the angular acceleration, the line acceleration and z_i are defined as:

$$ \omega_0^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \dot{\omega}_0^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \dot{v}_0^0 = \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix}, z_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (7) $$

As mentioned in [67], the Newton-Euler equation is shown in (8)

$$ f_i = I_i a_i + v_i \times I_i v_i, \quad (8) $$

where a_i, v_i and I_i represents the spatial acceleration, spatial velocity and spatial inertia , respectively, and f_i denotes the force applied on i th joint. The Newton-Euler equation can be further represented as

$$ f_i = \begin{bmatrix} \bar{I}_i\dot{\omega}_i + S(\omega_i)\bar{I}_i\omega_i - S(\ddot{d_{0i}})m_i c_i \\ m_i\ddot{d}_{0i} + S(\dot{\omega}_i)m_i c_i + S(\omega_i)S(\omega_i)m_i c_i \end{bmatrix}, \quad (9) $$

To formulate an estimation algorithm, (9) can be represented as (10)

$$ f_i = \begin{bmatrix} \mathbf{0} & -S(\ddot{d}_{0i}) & L(\dot{\omega}_i) + S(\omega_i)L(\omega_i) \\ \ddot{d}_{0i} & S(\dot{\omega}_i) + S(\omega_i)S(\omega_i) & \mathbf{0} \end{bmatrix} \begin{bmatrix} m_i \\ m_i c_i \\ \mathfrak{l}(\bar{I}_i) \end{bmatrix}, \quad (10) $$

where the mass moment $m_i c_i$ appears as a quantity to be estimated in combination. And $\ddot{d}_{0i} = \dot{v}_i + \omega_i \times v_i$, $S(x_i)$ is a 3×3 matrix shown in (11), $L(x_i)$ is a 3×6 matrix shown in (12), and $\mathfrak{l}(\bar{I}_i)$ is expressed as (13).

$$ S(x_i) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (11) $$

$$ L(x_i) = \begin{bmatrix} x_1 & x_2 & x_3 & 0 & 0 & 0 \\ 0 & x_1 & 0 & x_2 & x_3 & 0 \\ 0 & 0 & x_1 & 0 & x_2 & x_3 \end{bmatrix} \quad (12) $$

$$\mathfrak{l}(\bar{I}_i) = \begin{bmatrix} I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \end{bmatrix} \tag{13}$$

(10) can be expressed more compactly as

$$f_i = A_i \Phi_i, \tag{14}$$

where A_i is a 6×10 matrix which can be expressed as

$$A_i = \begin{bmatrix} \mathbf{0} & -S(\ddot{d}_{0i}) & L(\dot{\omega}_i) + S(\omega_i)L(\omega_i) \\ \ddot{d}_{0i} & S(\dot{\omega}_i) + S(\omega_i)S(\omega_i) & \mathbf{0} \end{bmatrix},$$

and Φ_i is a vector of 10 unknown inertia parameters which can be expressed as

$$\Phi_i = \begin{bmatrix} m_i \\ m_i c_i \\ \mathfrak{l}(\bar{I}_i) \end{bmatrix}.$$

f_{ij} is defined as the spatial force at joint i due to the movement of link j alone. Then f_{ii} is the spatial force at joint i due to movement of its own link which is the same as (14). The total spatial force ${}^i f_i$ at joint i is the sum of the spatial forces ${}^i f_{ij}$ for all joint from j to i

$${}^i f_i = \sum_{j=i}^{6} f_{ij} = \sum_{j=i}^{6} {}^i X_j^F A_j \Phi_j, \tag{15}$$

where ${}^i X_j^F$ is the spatial force transform matrix. For convenience, we note that ${}^i X_i^F = I_{6\times 6}$. Thus, we can derive an upper-diagonal matrix expression as (16)

$$\begin{bmatrix} {}^1 f_1 \\ {}^2 f_2 \\ \vdots \\ {}^6 f_6 \end{bmatrix} = \begin{bmatrix} {}^1 X_1^F A_1 & {}^1 X_2^F A_2 & \cdots & {}^1 X_6^F A_6 \\ \mathbf{0} & {}^2 X_2^F A_2 & \cdots & {}^2 X_6^F A_6 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & {}^6 X_6^F A_6 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_6 \end{bmatrix}, \tag{16}$$

Each ${}^i f_i$ must be translated into torque τ_i since only torque about the joint rotation axis z_i can be measured directly. For short, we can express it as follow

$$\tau^{(n)} = K^{(n)} \Phi, \tag{17}$$

where

$$\tau^{(n)} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_6 \end{bmatrix}, \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_6 \end{bmatrix}.$$

In this way, each elements of the torque vector or matrix can be expressed as

$$\tau_i = \begin{bmatrix} z_i \\ \mathbf{0} \end{bmatrix}^T {}^i f_i,$$

$$K_{ij} = \begin{bmatrix} z_i \\ \mathbf{0} \end{bmatrix}^T {}^i X_j^F A_j,$$

and $K_{ij} = \mathbf{0}_{1\times 10}$ if $i > j$. For this 6 links manipulator, τ is a 6×1 vector and K is a 6×60 matrix.

C. Trajectory Parametrization

It is obviously that the matrix K in (20) is just the function of the manipulator structure and the movement status of each joint, derived from (8) to (14). Using the measured joint angle, angular velocity and acceleration, the matrix K can be obtained based on (20). The accuracy of the numerical calculation can be influenced by the matrix condition number of K. So PSO method is proposed to generate a nice trajectory, which provides excitation to the system dynamic. Additionally, the measured angular position data are both band-limited and periodic. These characteristics make the processing more simpler and the parameter estimation more accurate. Arbitrary signal can be expressed as the Fourier expansion as (18). In this expression, the parameter θ can be designed as band-limited and periodic signal if N is finite. The upper limit of N is determined by the frequency response of the robot. Similarly, the angular velocity and angular acceleration can be expressed as follows.

$$\begin{aligned} \theta &= \theta_0 + \sum_{k=1}^{N} \big(\frac{a_k}{k\omega_f} \sin(k\omega_f t) - \frac{b_k}{k\omega_f} \cos(k\omega_f t) \big), \\ \dot{\theta} &= \dot{\theta}_0 + \sum_{k=1}^{N} \big(a_k \cos(k\omega_f t) + b_k \sin(k\omega_f t) \big), \\ \ddot{\theta} &= \ddot{\theta}_0 - \sum_{k=1}^{N} \big(k\omega_f a_k \sin(k\omega_f t) + k\omega_f b_k \cos(k\omega_f t) \big), \end{aligned} \tag{18}$$

where a_k and b_k are the coefficients of Fourier transformation; θ_0, $\dot{\theta}_0$, $\ddot{\theta}_0$ are the offsets of each joint trajectory; ω_f is the fundamental pulsation of the Fourier series; k is the frequency coefficient. In this paper, the constant N is set as 5, and ω_f is $0.1\times 2\pi$. The parameters needed to be optimized are a_k and b_k. As this manipulator system has 6 DOF, and for each joint there are 10 variables needed to be confirmed. The optimization vector has 60 dimensions.

D. Parameter Optimization for Trajectory

Based on the Theorem 1, the PSO has the convergence property. Meanwhile the PSO method guarantees convergence to the ideality limits in a shorter time for this optimization problem.

Remark 3: We can treat this parameter optimization for exciting trajectory as a multi-dimension nonlinear free model optimization problem. For K in (20) is a nonsingular matrix, the condition number of matrix $\tilde{S}$ can be defined as the objective function, where $\widetilde{S} = diag(\sigma_1, \sigma_2, \cdots, \sigma_m)$ is a part of S shown in (22). Thus the objective function is as Eq. (19) shown and the variables are Fourier transformation coefficient a_k and b_k in (18).

$$cond(S) \quad = \quad \|\tilde{S}\|_2 \|\tilde{S}^{-1}\|_2. \tag{19}$$

As shown in (1) and (2), the parameter x_i can be substituted as the Fourier coefficient vector. As mentioned above, the

minimization of the uncertainty on the model identification is a complex nonlinear free model optimization with motion constraint. There are two approaches for constraint setting: one is the constant boundary constraint, and the other is the special route from one position to another position or a special loop. The motion constraints are the limitation impacting on the angular position, angular velocity and angular acceleration. In this problem, the upper limit and lower limit are determined by the real structural in space and working condition. For example, the fourth DOF in the arm is the elbow joint, which only can rotate in negative angel. This is very similar as the elbow of human beings. As HUBO is a humanoid robot, we are not able to obtain the motion trajectory of each joint in advance. Therefore the constant boundary constraint method is used in this paper as Table II shown. The boundary constraint is the feasible region of this optimization problem.

TABLE II
CONSTRAINTS OF EACH JOINT

Location	Joint	Minimum	Maximum
Shoulder Yaw	1	$-90°$	$+90°$
Shoulder Pitch	2	$-170°$	$+170°$
Shoulder Roll	3	$0°$	$+180°$
Elbow Pitch	4	$-150°$	$0°$
Elbow Yaw	5	$-120°$	$+120°$
Wrist Pitch	6	$-90°$	$+90°$

As shown in Table II, for joint 1 and joint 6, the absolute values of the maximum and minimum constraint are not the same, which may influence the optimization effect of PSO algorithm. So the parameters θ_0, $\dot{\theta}_0$, $\ddot{\theta}_0$ in (18) are used to offset the asymmetry of the angular constraint.

E. Model Identification Method

With the torque sensors on each joint of the manipulator, τ_i can be measured directly. And (17) represents the dynamics of the joint for one sample point. In this model identification, there are M data points:

$$K = \begin{bmatrix} K^1 \\ K^2 \\ \vdots \\ K^M \end{bmatrix}, \tau = \begin{bmatrix} \tau^1 \\ \tau^2 \\ \vdots \\ \tau^M \end{bmatrix}, \tau = K\Phi, \qquad (20)$$

where K is a $6M \times 60$ matrix and τ is a $6M \times 1$ vector. For the structure inherent characteristic, the matrix K may be not full rank. So the least square method is needed. The estimated of Φ can be expressed as (21)

$$\Phi = (K^T K)^{-1} K^T \tau. \qquad (21)$$

However, $K^T K$ is not invertible due to the loss of rank from restricted degrees of freedom at the proximal links and the lack of full force-torque sensing.The inertia parameters can be divided into three groups: fully identifiable, identifiable in linear combinations and completely unidentifiable[68]. In this paper, a dividing method is proposed by analyzing the correlation of each column of K. If a column is all zero, the relevant parameter is completely unidentifiable. The column is identifiable in linear combinations when a nonzero column is abandoned and the rank of the matrix does not reduce. The other columns are fully identifiable. In this system, these 60 inertial parameters can be divided into the following categories:

(a) fully identifiable: I_{zz1}, m_2c_{x2}, m_2c_{y2}, I_{xx2}, I_{xy2}, I_{xz2}, I_{yz2}, I_{zz2}, m_3c_{x3}, m_3c_{y3}, I_{xx3}, I_{xy3}, I_{xz3}, I_{yz3}, I_{zz3}, m_4c_{x4}, m_4c_{y4}, I_{xx4}, I_{xy4}, I_{xz4}, I_{yz4}, I_{zz4}, m_5c_{x5}, m_5c_{y5}, I_{xx5}, I_{xy5}, I_{xz5}, I_{yz5}, I_{zz5}, m_6c_{x6}, m_6c_{y6}, I_{xx6}, I_{xy6}, I_{xz6}, I_{yz6}, I_{zz6};
(b) identifiable in linear combinations: m_2c_{z2}, I_{yy2}, m_3c_{z3}, I_{yy3}, m_4 m_4c_{z4}, I_{yy4}, m_5 m_5c_{z5}, I_{yy5}, m_6 m_6c_{z6}, I_{yy6};
(c) completely unidentifiable:m_1, m_1c_{x1}, m_1c_{y1}, m_1c_{z1}, I_{xx1}, I_{xy1}, I_{xz1}, I_{yy1}, I_{yz1}, m_2, m_3.

The following two methods can be applied to solve the nonsingular problem. One method is "Singular Value Decomposition (SVD)" of K in (20). For the matrix K is only the function of the geometry of the manipulator structure, it can be generated by simulation. The SVD of K can be expressed as [67]

$$K = USV^T, \qquad (22)$$

where U and V are orthogonal matrix, $S = diag(\sigma_1, \sigma_2, \cdots, \sigma_m, 0, \cdots, 0)$ is a diagonal matrix. σ_i is the nonzero singular value of K and parameter m represents the rank of K. Substituting (22) in (20), we have the new expression of τ as

$$\tau = USV^T\Phi. \qquad (23)$$

Considering $\Psi = V^T\Phi$ and $\zeta = U^{-1}\tau = U^*\tau$, (23) can be rewritten as

$$\zeta = S\Psi, \qquad (24)$$

where S is a $6M \times 60$ matrix, and Ψ is 60×1 matrix. As S is a diagonal matrix, $S_{6M\times 60}$ can be replaced as $\tilde{S}_{6M\times r}$, meanwhile $\Psi_{60\times 1}$ can be replaced as $\tilde{\Psi}_{r\times 1}$, where r is the rank of matrix S. The new inertia matrix $\hat{\Psi}$ can be estimated as

$$\hat{\Psi} = (\tilde{S}^T\tilde{S})^{-1}\tilde{S}^T\zeta. \qquad (25)$$

Considering $\hat{\Psi}_{r\times 1} = V^T_{r\times 60}\hat{\Phi}_{60\times 1}$, where $\hat{\Phi}$ is the independent variable vector of the system of equation. Because the number of independent variables is more than the number of the equations, the system of equation includes infinite roots. We can obtain the roots by consistently setting $60-r$ of the inertia parameters to zero, leaving only r parameters, which can be estimated by this method.

Another method is "Ridge Regression". In this method, the matrix K^TK is substituted by $K^TK + \varepsilon I_{10n}$. εI_{10n} is a identity diagonal matrix and ε represents a very small value which is much less than the smallest nonzero eigenvalue of K^TK. In this way, the estimated can be expressed as

$$\hat{\Phi} = (K^TK + \varepsilon I_{60})^{-1}K^T\tau. \qquad (26)$$

The expansion of (26) can be expressed as [67] shows

$$\hat{\Phi} = \sum_{j=1}^{60} (u_j^T f) \frac{\mu_j}{\mu_j^2 + \varepsilon} v_j, \tag{27}$$

where μ_j is the singular values of K, u_j and v_j are the columns of matrix U and V as shown in (22). Hence, a very small value of μ_j can be counteracted by ε. Though the solution can be influenced by the parameter ε, as long as its magnitude is suitable enough, the solution error can be controlled in an ideal range.

IV. Recursive Adaptive Control Design

The adaptive control of robotic manipulators has been studied actively in recent years. Many remarkable results in this field have been obtained owing to the advances in taking the nonlinear, time-varying and coupled nature of manipulator dynamics fully into consideration.

Usually, the computational complexity of these methods is very tedious for a 6-DOF manipulator because of the complicated linear-in-parameters (LIP) process. Therefore, the recursive adaptive control algorithm based on Newton-Euler formulation is employed. Using this recursive algorithm, the LIP can be computed in an iterative method and the computational complexity of the adaptive control can be reduced effectively. In this recursive way, the basic approaches are always the same no matter how many DOFs the manipulator has.

A. Adaptive Control

The dynamics of an n-link rigid robotic system without additional friction or external disturbance can be described as follows [69]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{28}$$

where $q \in R^n$ is the coordinates, n is the number of the links in the robotic system, $\tau \in R^n$ is the applied joint torque, $M(q) \in R^{n\times n}$ is a inertia matrix, $C(q,\dot{q})\dot{q} \in R^n$ denotes the centripetal and Coriolis torques, and $G(q) \in R^{n\times n}$ is the gravitational force. We can know some of properties of these system parameters.

Property 1: [69] The matrix $M(q)$ is symmetric and positive definite.

Property 2: [69] The matrix $\dot{M}(q) - 2C(q,\dot{q})$ is skew-symmetric.

Property 3: [69] Assuming there is no external disturbance, the left-hand side of the dynamic equation can be linearly parameterized as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Y(q,\dot{q},\ddot{q})\theta, \tag{29}$$

where $\theta \in R^p$ contains the system parameters, and $Y(q,\dot{q},\ddot{q}) \in R^{n\times p}$ is the regression matrix, which contains known functions of the signal $q(t)$, $\dot{q}(t)$ and $\ddot{q}(t)$.

Remark 4: The notation $\hat{(\cdot)}$ represents the computed or nominal value of $(\cdot)$, and indicates that the theoretically exact feedback linearization cannot be achieved in practice due to uncertainties in the robotic system. The error or mismatch $\tilde{(\cdot)} = (\cdot) - \hat{(\cdot)}$ is a measure of one's knowledge of the system dynamics. Note that $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$ since the parameter vector θ is a constant.

In this paper, the basic adaptive control law follows the method in [69]. Using the sliding model control method, both the steady state position error and velocity error can be eliminated. We denote the desired trajectory of $q(t)$ as $q_d(t)$. And the trajectory error is $\tilde{q} = q(t) - q_d(t)$. The control objective is to track the desired trajectory. We use the sliding surface error to estimate error as in [70],

$$s = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda\tilde{q}, \tag{30}$$

where Λ is a constant matrix whose real part eigenvalues are positive strictly. Then we know that if $r \to 0$, $\dot{\tilde{q}}$ and$\tilde{q} \to 0$ as $t \to \infty$ because the error of position and velocity can convergence to zero by the hyperplane of (30). Meanwhile the we define $q_r(t)$ as the reference trajectory to estimate $q_d(t)$ as follows

$$q_r = q_d + \Lambda \int_0^t \tilde{q}. \tag{31}$$

Then, $\dot{q}_r$ and $\ddot{q}_r$ can be represented as

$$\dot{q}_r = \dot{q}_d + \Lambda\tilde{q}, \tag{32}$$

$$\ddot{q}_r = \ddot{q}_d + \Lambda\dot{\tilde{q}}. \tag{33}$$

As mentioned before, we need to estimate robotic system parameters by LIP method so that the matrix Y is now a function of $\dot{q}_r$ and $\ddot{q}_r$

$$\hat{M}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{G}(q) = Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{\theta}. \tag{34}$$

We can design the control law and adaptive law as

$$\tau = Y(q,\dot{q},\ddot{q})\hat{\theta} - Kr, \tag{35}$$

$$\dot{\hat{\theta}} = -\Gamma^{-1}Y^T(q,\dot{q},\ddot{q})r, \tag{36}$$

where Γ and K are both diagonal constant matrixes. From the dynamics of the robotic system, we have

$$\begin{aligned} \tau &= M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \\ &= Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\theta - M(q)\dot{r} - C(q,\dot{q})r. \end{aligned} \tag{37}$$

According to control law (35) and (37), we have

$$\tilde{M}(q)\ dotr + C(q,\dot{q})r + Kr = Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\tilde{\theta}. \tag{38}$$

Considering the Lyapunov stability, we can design the Lyapunov function as

$$V(t) = \frac{1}{2}r^T M r + \frac{1}{2}\tilde{\theta}^T\Gamma\tilde{\theta}. \tag{39}$$

We can obtain the time derivative of the last formula

$$\dot{V}(t) = r^T M\dot{r} + \frac{1}{2}r^T\dot{M}r + \tilde{\theta}^T\Gamma\dot{\tilde{\theta}}. \tag{40}$$

Considering the property mentioned above and substituting the control law and adaptive law in, we have

$$\dot{V}(t) = -r^T K r \le 0. \tag{41}$$

Then, we have

$$\int_0^T r^T K dt = V(0) - V(T) \leq 0. \tag{42}$$

The property of the diagonal constant matrix K can be obtained

$$\lambda_{\min}(K) \int_0^T r^T r dt \leq V(0). \tag{43}$$

From the definition, $t \to \infty$, both $\tilde{\theta}$ and $\dot{\tilde{\theta}}$ converge to 0. So we know that $r \to 0$ as the $t \to \infty$. Thus, using this adaptive control law, the system can be guaranteed to achieve zero steady-state close loop error for both position and velocity [70].

B. Recursive Algorithm

In this paper, the initial value of adaptive control is supposed to be the estimated vector in model identification as shown in the second portion in the right side of (10). So θ in (35) can be defined as

$$\begin{aligned}\theta_i =&[m_i, mc_{i,x}, mc_{i,y}, mc_{i,z}, I_{i,xx}, I_{i,xy}, I_{i,xz},\\ &I_{i,yy}, I_{i,yz}, I_{i,zz}]^T.\end{aligned} \tag{44}$$

To simplify the description of the Slotine adaptive control method in (29), we can define τ_r as

$$\tau_r = \hat{H}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{G}(q) = Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\theta. \tag{45}$$

As we know, recursive algorithm is an open-loop model, in which each joint can be analyzed separately. The control and adaptive law can be written in a recursive way. We assume that Γ in (35) is consisted by positive definite matrix $P_i \in R^{10\times 10}$. The torque control and adaptive law in (35) can be represented as follows

$$\tau = \tau_r - K_D s, \tag{46}$$

$$\dot{\hat{\theta}}_i = -P_i \sum_{j=1}^{i} s_j y_{ji}^T, \tag{47}$$

where $\tau_{r,i} = \sum_{k=1}^{n} y_{ik}\theta_k$ is the ith element of τ_r, and $y_{ik} = (y_{ik,1}, y_{ik,2}, \cdots, y_{ik,10}) \in R^{1\times 10}$ is the element of Y in (35), which is given as

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ 0 & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_{nn} \end{bmatrix}. \tag{48}$$

As the coordinate frame in Fig. 2 shows that, all the joints are rotational. Just considering the condition of rotational, besides the (6), the other Newton-Euler dynamic formulation can be expressed as

$$\begin{aligned}\tau_i =& z_i^T n_i\\ F_i =& m_i\dot{v}_i + \dot{\omega}_i \times mc_i + \omega_i \times (\omega_i \times mc_i) +_{i+1}^{i} Rf_{i+1},\\ N_i =& I\dot{\omega}_i + \omega_i \times (I_i\omega_i) + mc_i \times \dot{v}_i +_{i+1}^{i} R(_{i+1}^{i}P\\ &\times f_{i+1} + n_{i+1}),\end{aligned} \tag{49}$$

where ω_i, $\dot{\omega}_i$ and $\dot{v}_i$ represent the angular velocity, angular acceleration and linear acceleration of frame i respectively; f_i and n_i are the force and moment exerted on link i by link $i-1$; I is the inertia tensor about the origin of frame i; mc_i is the mass moment of link i and c_i is the mass center. Using the recursive Newton-Euler equations and the derivation in, the force and torque matrix can be expressed as follows

$$\begin{aligned}\omega_i\{\dot{q}\} =&_{i-1}^{i}R\omega_{i-1}\{\dot{q}\} + z_i\dot{q}_i,\\ \omega_i\{\dot{q}_r\} =&_{i-1}^{i}R\omega_{i-1}\{\dot{q}_r\} + z_i\dot{q}_{r,i},\\ \alpha_i =&_{i-1}^{i}R\alpha_{i-1} +_{i-1}^{i} R\omega_{i-1}\{\dot{q}_r\} \times z_i\dot{q}_i\\ &+_{i-1}^{i} R\omega_{i-1}\{\dot{q}\} \times z_i\dot{q}_{i,r} + 2z_i\ddot{q}_{i,r},\\ \beta_i =&_{i-1}^{i}R\beta_{i-1} +_{i-1}^{i} R\Phi_{i-1,r_i^{i-1}}P,\end{aligned} \tag{50}$$

where

$$\Phi_i = \Gamma_i + [a_i\times], \tag{51}$$

$$\Gamma_i = [\omega_i\{\dot{q}_r\}\times][\omega_i\{\dot{q}\}\times] + ([\omega_i\{\dot{q}_r\}\times][\omega_i\{\dot{q}\}\times])^T \tag{52}$$

F_i and N_i are represented as follows

$$F_i = A_i a_i +_{i+1}^{i} RF_{i+1}, \tag{53}$$

$$N_i = B_i a_i +_{i+1}^{i} R(P_{i+1} \times F_{i+1} + N_{i+1}), \tag{54}$$

where

$$A_i = \begin{bmatrix} \beta & \vdots & \Phi_i & \vdots & 0 \end{bmatrix}_{3\times 10}, \tag{55}$$

$$B_i = \begin{bmatrix} 0 & \vdots & -[\beta_i\times] & \vdots & \Omega_i \end{bmatrix}_{3\times 10}. \tag{56}$$

Here Ω_i is defined as

$$\begin{aligned}\Omega_i =& I_i\alpha_i + \omega_i\{\dot{q}_r\} \times (I_i\omega_i\{\dot{q}\}) + \omega_i\{\dot{q}\}\\ &\times (I_i\omega_i\{\dot{q}_r\}),\end{aligned} \tag{57}$$

and $\alpha_{i,j}$ is the jth element of α_i and $\Gamma_{i,jk}$ is the (j,k) element of Γ_i. As the derivation in, the parameter y_{ik} in (47) can be expressed as

$$y_{ik} = (\mu_i^k)^T B_k + (\gamma_i^k)^T A_k, \tag{58}$$

where

$$\mu_i^k = {}_{k-1}^{k}R\mu_i^{k-1}, \tag{59}$$

$$h_{i+1}^k = {}_{k-1}^{k}R(h_{i+1}^{k-1} +_k^{k-1} p), \tag{60}$$

$$\gamma_i^k = \mu_i^k \times h_{i+1}^k, \tag{61}$$

and i is set from 1 to 6, and k is chosen from $i+1$ to 6. Besides of the parameters defined above, the others can be set as zero. From the (35) and (46), torque can be calculated.

V. Numerical Simulations and Discussion

In this section, the performance of the proposed method of model identification algorithm is verified through numerical simulations in Matlab. The effectiveness of the estimated inertia parameters are proved by comparing in the recursive adaptive control system with different initial conditions. Based on the system structure in Fig. 1, the numerical simulation in this section is divided into three parts: exciting optimal trajectory, model identification and recursive adaptive control.

A. Exciting Optimal Trajectory

The excitation of optimal trajectory is the first step of the model identification. Under the constraints described in Table

II and the objective function, the PSO method is adopted to obtain the optimal value. In this particular problem, the particle number is 50. The optimization algorithm will not stop until the condition number in (19) is less than 130. For the condition of model identification, the condition number less than 130 is enough for exciting the full statement of the arm structure. The optimization results of the Fourier coefficients are shown in Table III

From the simulation result, it is obvious that the PSO algorithm has a high convergence efficiency. Using the coefficient parameters in Table III and based on (18), the optimal trajectory can be obtained as shown in Fig. 3.

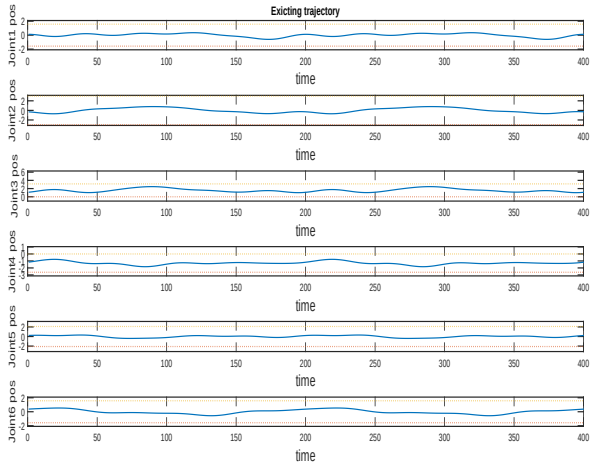

Fig. 3. Optimal trajectory

Remark 5: The Y-axis of Fig. 3 denotes the angle of each joint, in radian.

In Fig. 3 the solid lines are the joint trajectories, and the dotted lines are the constraint of each joint shown in Table II. The figure shows that all the excited trajectories are restrained in the constraint boundary.

B. Parameter Estimation

Since the actual function of the model identification is to obtain the model parameter for the adaptive control, the "Ridge regression" method is suitable for this condition [67]. Following that method, by substituting the trajectory data into the model identification module, the inertia parameters of the HUBO robot can be calculated. The measured and estimated torque are shown in Fig. 4. In this figure, the dotted line is the measured torque, and the solid line is the estimated torque. Fig. 4 illustrates that the measured and estimated torque are almost identical. The estimation error is very small, which is shown in Fig. 5

From the torque estimation result and estimation error, it is apparent that the torque of each joint can be estimated with a high accuracy. The torque estimation errors are less than $5 \times 10^{-7} Nm$. The high accuracy estimation result indicates that the estimated inertia parameters are suitable for this set of trajectories and torques. And the estimated inertia parameters are shown in Table IV.

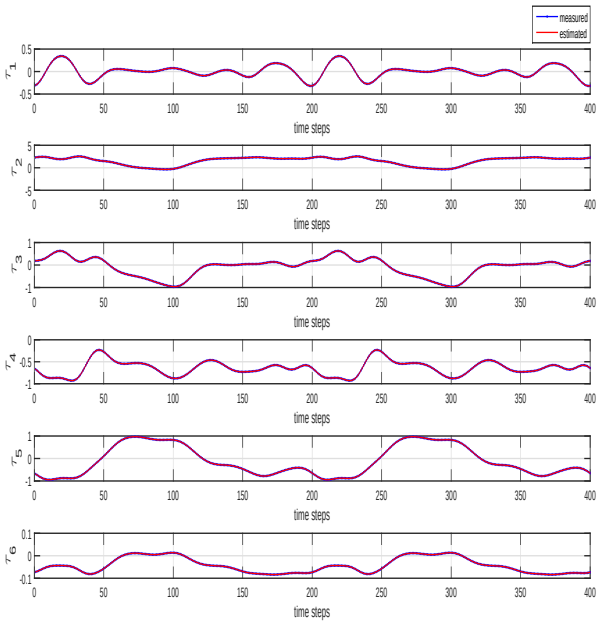

Fig. 4. Measured and estimated torque

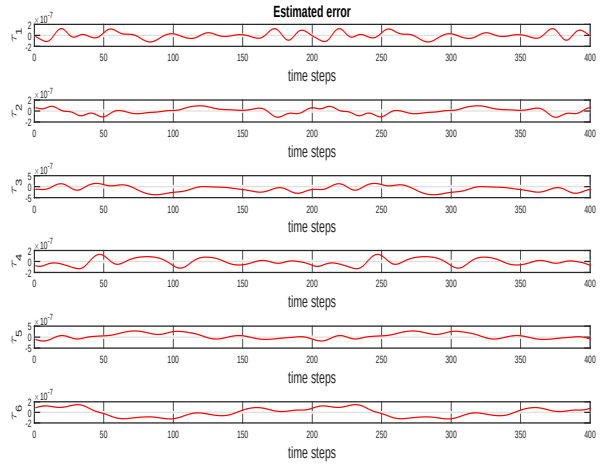

Fig. 5. Estimated error

C. Estimated Parameter Verification

For confirming of the validity of the estimated inertia parameters, a testing method is proposed. The testing method is run a simulation with a new trajectory different from the optimized one in Fig. 3. The new trajectory for testing is shown in Fig. 8, which is generated arbitrarily within the constraints. Taking the trajectory as the desired route, the estimated inertia parameters are shown in Table IV, which are used to estimate the new torque of each joint as (20), where K is the new matrix according to the new trajectory, andΦ is the estimated parameters. The effectiveness and generalization of the estimated parameters can be judged by estimated effect of the new trajectory. The small estimated error means that

TABLE III
COEFFICIENTS

Coefficients	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
a_1	0.0809	0.1578	0.0747	-3.084e-4	-0.0175	0.1089
a_2	0.1870	-0.1432	-0.1236	0.2508	0.3013	0.0044
a_3	-0.0406	-0.1306	0.3670	0.0597	-0.0538	0.1254
a_4	-0.1554	-0.1242	0.0960	0.3006	0.0482	0.0967
a_5	-0.0376	-0.0860	0.1920	-0.1368	0.0493	-0.1159
b_1	-0.1426	-0.3889	-0.2593	0.1383	0.0924	0.2347
b_2	0.1021	0.1311	0.3530	-0.103	-0.0603	0.1098
b_3	0.1690	0.0790	-0.1844	-0.0430	-0.0339	-0.2097
b_4	0.1530	0.3067	-0.4159	-0.0115	0.1136	-0.0243
b_5	0.3600	0.1896	-0.2709	-0.1878	0.2461	0.1022

TABLE IV
ESTIMATED PARAMETERS

Parameters	Unit	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
m_i	Kg	0	0	0	1.5370e-02	1.53699e-02	3.55626e-02
mc_x	$\mathrm{Kg}\cdot\mathrm{m}$	0	9.54103e-08	9.64515e-09	-2.665e-05	1.17065e-02	5.07750e-03
mc_y	$\mathrm{Kg}\cdot\mathrm{m}$	0	2.72188e-05	-8.1902e-02	8.8609e-03	-1.2679e-05	6.99648e-03
mc_z	$\mathrm{Kg}\cdot\mathrm{m}$	0	3.40486e-19	-2.7219e-05	8.3420e-02	-8.8609e-03	- 9.35302e-03
I_{xx}	$\mathrm{Kg}\cdot\mathrm{m}^2$	0	2.47756e-03	2.13150e-03	4.7763e-03	-4.4776e-03	0.51961e-03
I_{xy}	$\mathrm{Kg}\cdot\mathrm{m}^2$	0	7.64263e-07	2.64990e-08	1.4295e-06	4.92949e-08	- 3.23927e-05
I_{xz}	$\mathrm{Kg}\cdot\mathrm{m}^2$	0	-1.55390e-07	-6.6769e-08	2.0950e-07	2.98953e-03	- 2.81641e-04
I_{yy}	$\mathrm{Kg}\cdot\mathrm{m}^2$	0	2.68877e-03	2.89464e-04	-6.955e-04	1.47890e-02	5.11529e-03
I_{yz}	$\mathrm{Kg}\cdot\mathrm{m}^2$	0	-3.45664e-09	-1.4773e-10	1.2946e-05	6.67067e-09	- 1.89297e-04
I_{zz}	$\mathrm{Kg}\cdot\mathrm{m}^2$	5.16632e-03	-5.65875e-05	1.94927e-03	5.5351e-03	1.93014e-02	7.4893e-04

the estimated inertia parameters have good generalization and effectiveness.

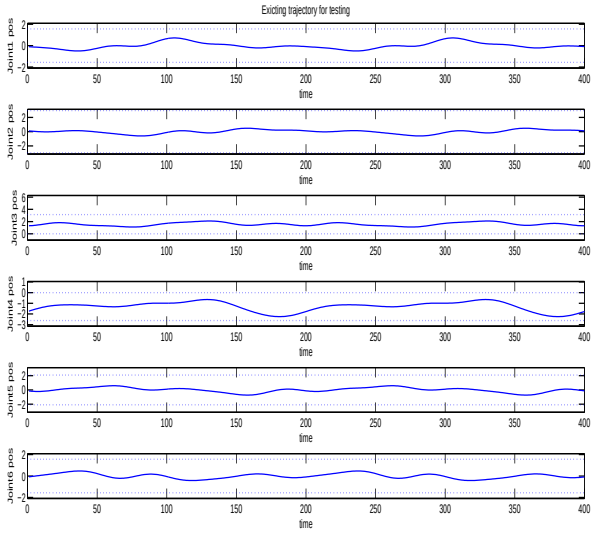

Fig. 8. Trajectory for testing

The estimation results and their errors are shown in Fig. 6 and Fig. 7.

Fig. 6 and Fig. 7 illustrate that the estimated torque error in the versification is less that $5\times10^{-7}N\cdot m$, so that the estimated inertia parameter in Table IV can be used to estimate other trajectories and the relevant torque primely, which means that the model identification result is accurate and generalized.

D. Recursive Adaptive Control

On the basis of the RNE adaptive control algorithm, the simulation experiment is provided in this section. We use zero as the initial position of each joint. The initial value of inertia parameters θ in (44) are the estimated data from the model identification module. The initial values of the other parameters in recursive algorithm are as follows: $\alpha_0 = 0$, $\beta_0 = 2z_0g$, $\Phi_0 = 0$, $h^i_{i+1} = 0$, and ${}^i\mu_i = z_i/2$, where g is the gravitational acceleration. The gain matrix K_D in (46) is given by $diag(22, 10, 13, 13, 11, 12)$, the sliding surface coefficient matrix Λ in (30) is given by $diag(8, 30, 5, 0.1, 13, 0.1)$ and adaptive law matrix P in Eq. (47) is given by $10^{-6}I_{60}$, where I_{60} is a 60×60 identity matrix. According to the set of initial values, the control simulation effect and control error are shown in Fig. 9 and Fig. 10. In Fig. 9, the solid and dotted lines represent the actual and desired trajectory of each joint, respectively. The desired trajectories are designed as

$$\theta_d = \theta_{d0} + Amp \cdot \sin(\omega_f t), \quad (62)$$

where $Amp = [180, 80, 180, 150, 120, 180] \cdot 0.5 \cdot \pi/180$, $\theta_{d0} = [0, 60, 90, -45, 0, 30] \cdot 0.5 \cdot \pi/180$ and $\omega_f = 5$. As shown in Fig. 9 and Fig. 10, it is illustrated that each joint can be controlled according to the desired trajectory, and the control error decreases with time. In this manner, the control performance of the adaptive control algorithm is verified.

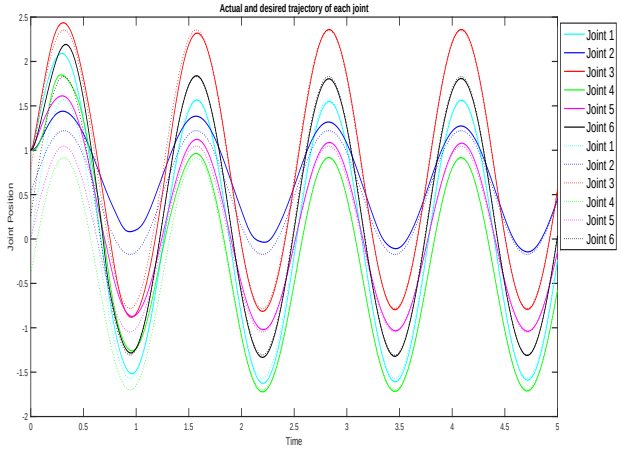

Fig. 9. Output of the robotic system with adaptive control

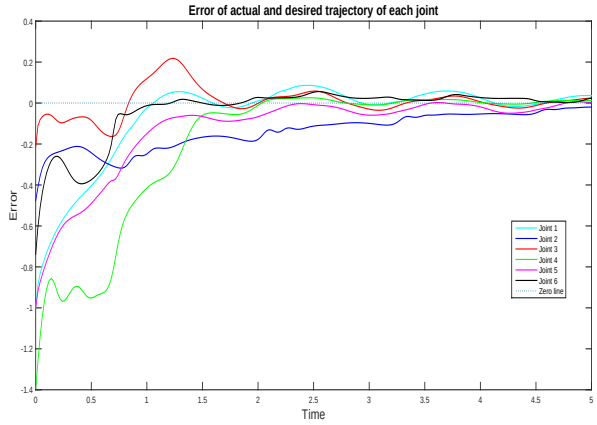

Fig. 10. Tracking error of adaptive control

To further evaluate the result of the estimated inertia parameters, we compare the control results of adaptive controller with the estimated parameters against without the estimated parameters. Obviously, the simulation results show the fact that the control scheme when inertia parameters are set as the estimated parameters shown in Table. IV is better than pure adaptive controller.

As shown in Fig. 11 and Fig. 12, when the values of the inertia parameters are set as the estimated parameters, the convergency rate of torques of each joint and the torque limits are improved, compared to that when only adaptive controller is presented. In this way, by choosing the estimated inertia parameters as the initial value of θ in (44), the nominal torque of the motor can be well controlled and the performance of the system is improved.

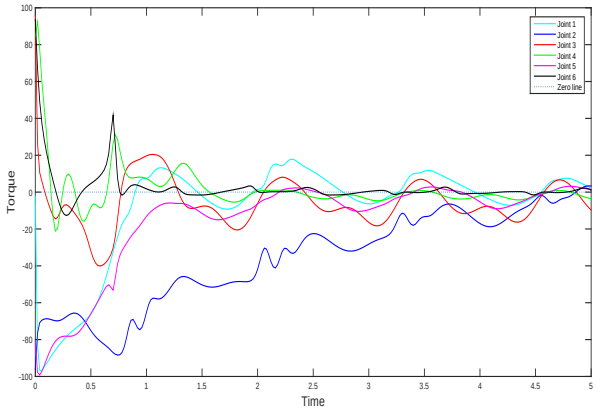

Fig. 11. Torque for the estimated inertia parameters

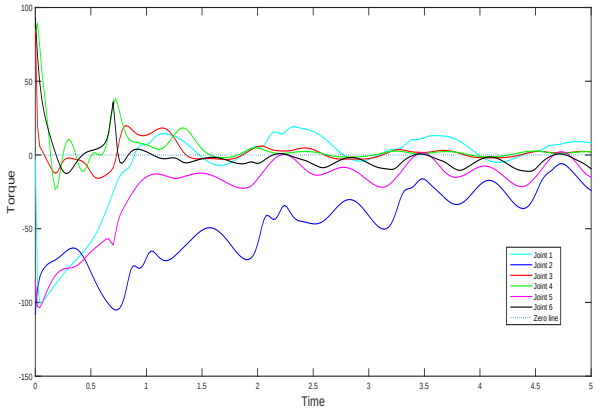

Fig. 12. Torque for adaptive control

Remark 6: The unit of Y-axis of Fig. 9 and Fig. 10 is radian, while the unit of Y-axis of Fig. 11 and Fig. 12 is $\mathrm{N} \cdot \mathrm{m}$.

From the simulation results, we know that the torque of the motor can be well controlled with high accuracy estimation inertia parameters. And the trajectory can tracking with small errors. Then in the future, precision of parameter estimation can be improved to deal with more complicated situation.

VI. Conclusion

In this paper, the D-H model of the upper limb of a humanoid robot is presented and the Newton-Euler formulation of the manipulator of the HUBO robot has been derived. The exciting trajectory generation method based on PSO has been proposed. Using this method, the accuracy and generalization of the estimation result can be guaranteed and the optimization efficiency has been improved. Based on the optimized trajectory, the structural inertia parameters have been estimated. To reduce the complexity of the computation, recursive adaptive control algorithm plays a pretty good role for improving the control performance. Using the estimated inertia parameters as the initial value in adaptive control progress, the torque convergency of joint motors has been improved. In this paper, we only consider modeling and control of the the upper limb

of a humanoid robot, HUBO. Future work includes model identification for the lower limbs of the humanoid robot, and the balancing control will be investigated further.

ACKNOWLEDGMENTS

The authors would like to thank the Editor-In-Chief, the Associate Editor and the anonymous reviewers for their constructive comments which helped improve the quality and presentation of this paper.

REFERENCES

[1] K.-H. Park, H.-E. Lee, Y. Kim, and Z. Z. Bien, "A steward robot for human-friendly human-machine interaction in a smart house environment," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 21–25, 2008.

[2] H. Mochiyama, "Kinematics for the whole arm of a serial-chain manipulator," *Advanced Robotics*, vol. 15, no. 2, pp. 255–275, 2001.

[3] D. Guo and Y. Zhang, "Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1203–1217, 2014.

[4] C. Atkeson, C. An, and J. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

[5] X. Ding and J. S. Dai, "Characteristic equation-based dynamics analysis of vibratory bowl feeders with three spatial compliant legs," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 164–175, 2008.

[6] J. Swevers, W. Verdonck, and J. Schutter, "Dynamic model identification for industrial robots," *IEEE Control Systems Magazine*, vol. 27, no. 5, pp. 58–71, 2007.

[7] J. Swevers, W. Verdonck, B. Naumer, S. Pieters, and E. Biber, "An experimental robot load identification method for industrial application," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 701–712, 2002.

[8] V. Portman, B. Sandler, and E. Zahavi, "Rigid 6-DOF parallel platform for precision 3-D micromanipulation," *International Journal of Machine Tools and Manufacture*, vol. 41, no. 9, pp. 1229–1250, 2001.

[9] M. Rognant, E. Courteille, and P. Maurine, "A systematic procedure for the elastodynamic modeling and identification of robot manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1085–1093, 2010.

[10] C. Wang and D. J. Hill, "Learning form neural control," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 130–146, 2006.

[11] A. Yazdizadeh, K. Khorasani, and R. Patel, "Identification of a two-link flexible manipulator using adaptive time delay neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31, no. 1, pp. 165–172, 2000.

[12] W. Ge, W. He, Y. Li, and C. Yang, "Model identification and adaptive control design for a 6 dofs manipulator," in *2015 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 461–466, IEEE, 2015.

[13] Z. Li, P. Y. Tao, S. S. Ge, M. Adams, and W. S. Wijesoma, "Robust adaptive control of cooperating mobile manipulators with relative motion," *IEEE Transactions on System, Man, and Cybernetics*, vol. 39, no. 1, pp. 103–115, 2009.

[14] R. Cui, J. Guo, and Z. Mao, "Adaptive backstepping control of wheeled inverted pendulums models," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 501–511, 2015.

[15] Y.-J. Liu and S.-C. Tong, "Barrier lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints," *Automatica*, vol. 64, no. 2, pp. 70–75, 2016.

[16] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.

[17] W. He, S. S. Ge, Y. Li, E. Chew, and Y. S. Ng, "Neural network control of a rehabilitation robot by state and output feedback," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 15–31, 2015.

[18] S.-C. Tong, Y. Li, Y.-M. Li, and Y.-J. Liu, "Observer-based adaptive fuzzy backstepping control for a class of stochastic nonlinear strict-feedback systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 6, pp. 1693–1704, 2011.

[19] Z. Liu, F. Wang, and Y. Zhang, "Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled dynamic," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1301–1312, 2015.

[20] C. L. P. Chen, G.-X. Wen, Y.-J. Liu, and Z. Liu, "Observer-based adaptive backstepping consensus tracking control for high-order nonlinear semi-strict-feedback multiagent systems," *IEEE Transactions on Cybernetics*, 2016, In Press, DOI: 10.1109/TCYB.2015.2452217.

[21] S. Tong, L. Zhang, and Y. Li, "Observed-based adaptive fuzzy decentralized tracking control for switched uncertain nonlinear large-scale systems with dead zones," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2015, In Press, DOI: 10.1109/TSMC.2015.2426131.

[22] S.-C. Tong, X.-L. He, and H.-G. Zhang, "A combined backstepping and small-gain approach to robust adaptive fuzzy output feedback control," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1059–1069, 2009.

[23] W. He and S. S. Ge, "Cooperative control of a nonuniform gantry crane with constrained tension," *Automatica*, vol. 66, no. 4, pp. 146–154, 2016.

[24] Y.-J. Liu and S.-C. Tong, "Adaptive NN tracking control of uncertain nonlinear discrete-time systems with nonaffine dead-zone input," *IEEE Transactions on Cybernetcs*, vol. 45, no. 3, pp. 497–505, 2015.

[25] W. He, S. Zhang, and S. S. Ge, "Adaptive control of a flexible crane system with the boundary output constraint," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 8, pp. 4126–4133, 2014.

[26] Y.-J. Liu, S. Tong, D.-J. Li, and Y. Gao, "Fuzzy adaptive control with state observer for a class of nonlinear discrete-time systems with input constraint," *IEEE Transactions on Fuzzy Systems*, 2016, In Press, DOI: 10.1109/TFUZZ.2015.2505088.

[27] W. He and S. S. Ge, "Vibration control of a flexible beam with output constraint," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5023–5030, 2015.

[28] C. L. P. Chen, G.-X. Wen, Y.-J. Liu, and F.-Y. Wang, "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1217–1226, 2014.

[29] W. He, S. S. Ge, and D. Huang, "Modeling and vibration control for a nonlinear moving string with output constraint," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 4, pp. 1886–1897, 2015.

[30] Y.-J. Liu, L. Tang, S.-C. Tong, and C. L. P. Chen, "Adaptive NN controller design for a class of nonlinear MIMO discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1007–1018, 2015.

[31] D. E. Miller, "A new approach to model reference adaptive control," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 743–757, 2003.

[32] J.-X. Xu and D. Huang, "Spatial periodic adaptive control for rotary machine systems," *Automatic Control, IEEE Transactions on*, vol. 53, no. 10, pp. 2402–2408, 2008.

[33] C. Yang, Z. Li, and J. Li, "Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 24–36, 2013.

[34] D. Liu, D. Wang, F.-Y. Wang, H. Li, and X. Yang, "Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2834–2847, 2014.

[35] D. Liu, D. Wang, and H. Li, "Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 418–428, 2014.

[36] Z. Li and C.-Y. Su, "Neural-adaptive control of single-master-multiple-slaves teleoperation for coordinated multiple mobile manipulators with time-varying communication delays and input uncertainties," *IEEE Transactions on Neural Networks and Learning System*, vol. 24, no. 9, pp. 1400–1413, 2013.

[37] Z. Liu, C. Chen, Y. Zhang, and C. L. P. Chen, "Adaptive neural control for dual-arm coordination of humanoid robot with unknown nonlinearities in output mechanism," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 521–532, 2015.

[38] S.-L. Dai, C. Wang, and F. Luo, "Identification and learning control of ocean surface ship using neural networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 801–810, 2012.

[39] S.-L. Dai, C. Wang, and M. Wang, "Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 111–123, 2014.

[40] Z. Liu, G. Lai, Y. Zhang, X. Chen, and C. L. P. Chen, "Adaptive neural control for a class of nonlinear time-varying delay systems

with unknown hysteresis.," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2129–2140, 2014.
[41] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004–2016, 2014.
[42] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 334–344, 2016.
[43] W. He, A. O. David, Z. Yin, and C. Sun, "Neural network control of a robotic manipulator with input deadzone and output constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2016, In Press, DOI: 10.1109/TSMC.2015.2466194.
[44] Y.-J. Liu and S. Tong, "Adaptive fuzzy control for a class of unknown nonlinear dynamical systems," *Fuzzy Sets and Systems*, vol. 22, no. 2, pp. 595–603, 2015.
[45] Y.-J. Liu, Y. Gao, S.-C. Tong, and Y.-M. Li, "Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 16–28, 2016.
[46] Y.-J. Liu, S.-C. Tong, and C. L. P. Chen, "Adaptive fuzzy control via observer design for uncertain nonlinear systems with unmodeled dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 2, pp. 275–288, 2013.
[47] Z. Li, L. Ding, H. Gao, G. Duan, and C.-Y. Su, "Trilateral tele-operation of adaptive fuzzy force/motion control for nonlinear teleoperators with communication random delays," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 610–623, 2013.
[48] Y.-J. Liu and S. Tong, "Adaptive fuzzy control for a class of nonlinear discrete-time systems with backlash," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1359–1365, 2014.
[49] Z. Li, S. Xiao, S. S. Ge, and H. Su, "Constrained multilegged robot system modeling and fuzzy control with uncertain kinematics and dynamics incorporating foot force optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2015, In Press, DOI: 10.1109/TSMC.2015.2422267.
[50] C. L. P. Chen, Y.-J. Liu, and G.-X. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 583–593, 2014.
[51] Y.-J. Liu and S. Tong, "Adaptive fuzzy identification and control for a class of nonlinear pure-feedback MIMO systems with unknown dead zones," *IEEE Transactions on Fuzzy Systems,*, vol. 23, no. 5, pp. 1387–1398, 2015.
[52] Z. Liu, F. Wang, Y. Zhang, X. Chen, and C. L. P. Chen, "Adaptive fuzzy output-feedback controller design for nonlinear systems via backstepping and small-gain approach," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1714–1725, 2014.
[53] Y. Gao and Y.-J. Liu, "Adaptive fuzzy optimal control using direct heuristic dynamic programming for chaotic discrete-time system," *Journal of Vibration and Control*, vol. 263, pp. 49–70, 2016.
[54] G.-X. Wen, C. L. P. Chen, Y.-J. Liu, and Z. Liu, "Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems," *IET Control Theory & Applications*, vol. 9, no. 13, pp. 1927–1934, 2015.
[55] M. Zareh and S. Soheili, "A modified model reference adaptive control with application to MEMS gyroscope," *Journal of Mechanical Science and Technology*, vol. 25, no. 8, pp. 2061–2066, 2011.
[56] C. Chen, Z. Liu, Y. Zhang, C. L. P. Chen, and S. Xie, "Saturated nussbaum function based approach for robotic systems with unknown actuator dynamics," *IEEE Transactions on Cybernetics*, 2016, DOI: 10.1109/TCYB.2015.2475363.
[57] R. Cui, Y. Li, and W. Yan, "Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT*," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2016, In Press, DOI: 10.1109/TSMC.2015.2500027.
[58] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011.
[59] L. Wang, Z. Liu, C. L. P. Chen, Y. Zhang, S. Lee, and X. Chen, "Energy-efficient SVM learning control system for biped walking robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 831–837, 2013.
[60] H. KaWasaki, T. Bito, and K. Kanzaki, "An efficient algorithm for the model-based adaptive control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 496–501, 1996.
[61] M. Pant, R. Thangaraj, and A. Abraham, "A new quantum behaved particle swarm optimization," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pp. 87–94, ACM, 2008.
[62] M. Clerc, *Particle swarm optimization*. John Wiley & Sons, 2010.
[63] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer, 2010.
[64] S. S. Rao and S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.
[65] F. Van Den Bergh, *An analysis of particle swarm optimizers*. PhD thesis, University of Pretoria, 2006.
[66] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.
[67] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.
[68] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.
[69] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.
[70] J. K. Liu, *Design and MATLAB simulation for robotic control system*. Tsinghua University Press, 2008.

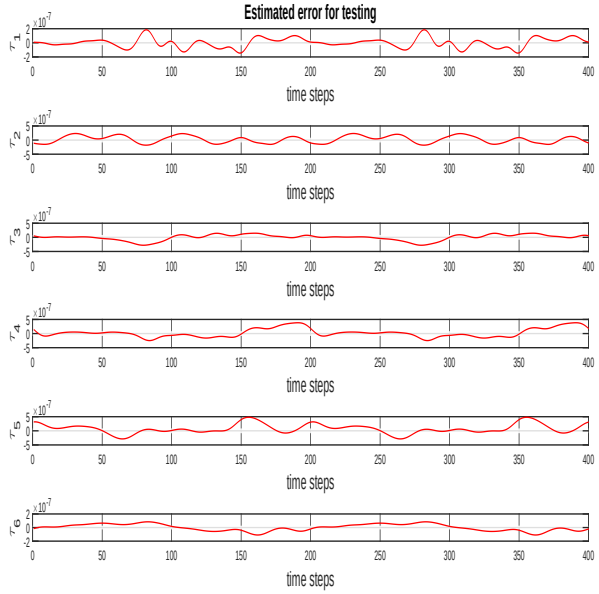

Fig. 6. Measured and estimated torque for testing

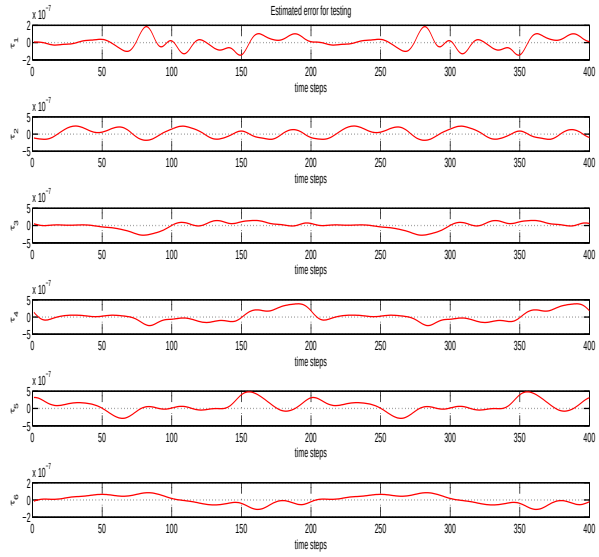

Fig. 7. Estimated error for testing