Robotica (2014) volume 32, pp. 375–399. © Cambridge University Press 2013
doi:10.1017/S0263574713000775

Function approximation technique-based adaptive virtual decomposition control for a serial-chain manipulator

Hayder F. N. Al-Shuka†*, B. Corves† and Wen-Hong Zhu‡

†*Department of Mechanism and Machine Dynamics, RWTH Aachen University, Aachen, Germany*
‡*Canadian Space Agency, Canada*

(Accepted June 26, 2013. First published online: August 6, 2013)

SUMMARY
The virtual decomposition control (VDC) is an efficient tool suitable to deal with the full-dynamics-based control problem of complex robots. However, the regressor-based adaptive control used by VDC to control every subsystem and to estimate the unknown parameters demands specific knowledge about the system physics. Therefore, in this paper, we focus on reorganizing the equation of the VDC for a serial chain manipulator using the adaptive function approximation technique (FAT) without needing specific system physics. The dynamic matrices of the dynamic equation of every subsystem (e.g. link and joint) are approximated by orthogonal functions due to the minimum approximation errors produced. The control, the virtual stability of every subsystem and the stability of the entire robotic system are proved in this work. Then the computational complexity of the FAT is compared with the regressor-based approach. Despite the apparent advantage of the FAT in avoiding the regressor matrix, its computational complexity can result in difficulties in the implementation because of the representation of the dynamic matrices of the link subsystem by two large sparse matrices. In effect, the FAT-based adaptive VDC requires further work for improving the representation of the dynamic matrices of the target subsystem. Two case studies are simulated by Matlab/Simulink: a 2-R manipulator and a 6-DOF planar biped robot for verification purposes.

KEYWORDS: Virtual decomposition control; Adaptive control; Function approximation technique; Computational complexity; Biped robot; Orthogonal functions.

1. Introduction

For complex robotic systems, such as humanoid robots or any robot having the number of degrees of freedom (DOFs) larger than 6 DOFs, difficulties are encountered in the implementation of the control algorithms. Therefore, over 30 years, the robotics researchers have focused on the problem of computational efficiency. Many efficient $O(n)$ algorithms have been developed for inverse[1–6] and forward dynamics[7–10] of robotic systems. For more literature on the efficient dynamic algorithms, we refer to refs. [11, 12]. However, the adaptive control algorithm which deals with controlling the robotic systems despite their uncertain parameters may decrease the computational efficiency of the dynamics $O(n)$ algorithms. Fu *et al.*[13] have shown that the combined identification and control algorithms can be computed in $O(n^3)$ time despite using recursive Newton–Euler (NE) formulation. Therefore, most researchers have used local controller (decentralized controller) such as proportional-integral-derivative (PID) family for controlling complex dynamic systems.[14–19] The problem now is that the independent joint control considers the coupling effects of the other bodies or links as disturbances, which is unrealistic.[20] Dallali *et al.*[21] have made a comparison between multivariable and decentralized control strategies for a 10-DOF humanoid robot. It was shown that the multivariable linear quadratic regulator, which takes the entire system dynamics into account, has better robustness and energy efficiency.

* Corresponding author. E-mail: al-shuka@igm.rwth-aachen.de

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

One of the efficient tools to deal with full-dynamics-based control for complex robotic systems is the virtual decomposition control (VDC) suggested by Wen-Hong Zhu and his colleagues.[22–29] The authors have introduced the notion of virtual stability that deals with the interaction between the subsystems. If every subsystem is virtually stable, the entire robotic system is stable. The dynamics of every subsystem is derived using the NE method, which satisfies the low computational time. The virtual decomposition process is flexible in a manner that every subsystem can be further decomposed into sub-subsystems provided that the virtual stability is guaranteed. The VDC has been applied to multiple robot manipulators, space robots, and biped robots etc.[20] Antonelli *et al.*[30] have used this approach for the adaptive control of underwater vehicle-manipulator systems, with modification in the stability analysis. The authors have used the task-priority approach in analyzing the stability instead of the virtual stability mentioned above.

However, the VDC still adopts the regressor approach as an adaptive control, though with respect to subsystems. It should be pointed out that the regressor-based approach may pose difficulties in practical implementation due to the complexity of the regressor matrix which embodies the full dynamics of the robotic systems.[31] Therefore, much attention has been focused on to evade the regressor computations.[31–39] The function approximation technique (FAT) is an essential tool for approximating the unknown parameters of the dynamic system without using the regressor matrix. It has been used successfully for low dimensional robotic systems without considering the computational complexity of the proposed algorithms.[40–42]

In the conventional adaptive control, which is defined here as the control of the robotic system entirely, the regressor computational complexity is a function of both the number of DOFs and the uncertain parameters, whereas the FAT computational complexity is a function of both the number of DOFs and the number of the basis functions used in the approximation technique. Therefore, difficulties may arise for higher DOF robots. The situation is different with the VDC. The maximum numbers of DOFs of each subsystem are six and three for links and joints, respectively. Consequently, the regressor computations are functions of the number of uncertain parameters which are equal to 13 for a link subsystem in free space.[20] However, the dimension of the vector of the estimated dynamic parameters could be very large (about 100) for a rigid body moving in a fluid considering all hydrodynamic and damping terms.[30,43] As a result, this motivates us to reorganize the equation of the VDC of a serial-chain manipulator in terms of FAT. The virtual stability of every subsystem and the stability of the entire system are proved in this work. The control of each link and joint is performed individually. The friction effects on the joints include Coulomb, viscous, Stribeck and offset friction. The computational complexity of FAT-based adaptive VDC is considered and compared with the regressor-based adaptive VDC.

This paper can be organized as follows. Section 2 summarizes a review on VDC while Section 3 shows briefly the FAT. Section 4 is divided into four sections. Section 4.1 deals with the FAT-based adaptive VDC for each link, while the same method is shown for each joint in Section 4.2. Section 4.3 proves stability of the entire system. The end of this section is closed up with the computational complexity of the FAT comparing with the regressor approach. Section 5 shows the simulation results and discussions. The conclusions are summarized in Section 6.

2. Review of the VDC

When the DOFs of the robotic system are larger than 6, difficulties are encountered in the implementation of control algorithms. One possible solution of this problem is the VDC.[3] VDC is a powerful tool to control every subsystem (links and joints) individually, while maintaining the L_2 and L_∞ stability and convergence of the entire robotic system. Some important definitions, lemmas and theorems, cited from ref. [20], are given this section to understand the notion of the VDC. To perform the VDC of each subsystem, the virtual power flows are introduced to characterize the dynamic interaction among the subsystems.

Definition 1. [20] With respect to the frame $\{A\}$, the virtual power flow is defined as the inner product of the linear/angular velocity error vector and the force/moment error vector, that is,

$$p_A = \left({}^A v_r - {}^A v\right)^T \left({}^A f_r - {}^A f\right), \tag{1}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

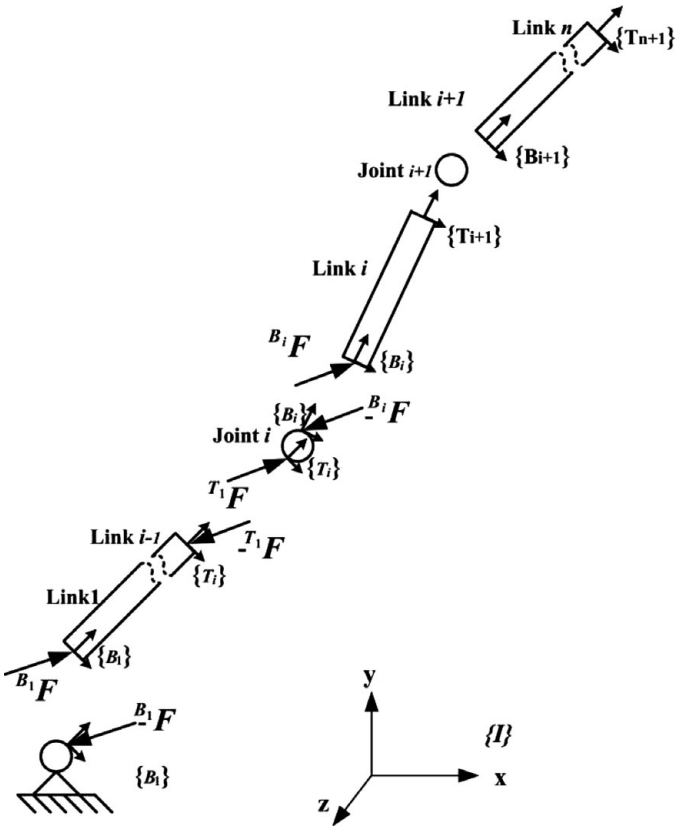


Fig. 1. Virtual decomposition of a serial-chain manipulator.

where ${}^{A}v_r \in \mathbb{R}^6$ and ${}^{A}f_r \in \mathbb{R}^6$ represent the required (design) vector of ${}^{A}v \in \mathbb{R}^6$ and ${}^{A}f \in \mathbb{R}^6$, respectively. This virtual power flow is equal and opposite in sign with the neighboring subsystem at the interaction.

To allow the complex robotic system to be decomposed into subsystems, the notion of the virtual cutting point is introduced.

Definition 2. [20] A cutting point is a directed separation interface that conceptually cuts through a rigid body. At the cutting point, the two parts resulting from the virtual cut maintain equal position and orientation. The cutting point is interpreted as a driving cutting point by one part and is simultaneously interpreted as a driven cutting point by another part. A force vector $f \in \mathbb{R}^3$ and a moment vector $m \in \mathbb{R}^3$ are exerted from one part to which the cutting point is interpreted as a driving cutting point to the other part to which the cutting point is interpreted as a driven cutting point.

After introducing the two important definitions, next we will explain the concept of the virtual stability of every subsystem that guarantees the stability of the entire robotic system. Because each subsystem is dealt individually, the NE method is used for the derivation of its dynamics. In the VDC, it is useful to derive the dynamic equation of every subsystem with respect to a local frame located at one end of the subsystem oriented according to the Denavite–Hartenberg representation.

We consider a fixed-base serial-chain manipulator with revolute and prismatic joints. Thus, the links are numbered from 1 to n, where the base link is numbered as zero link. Figure 1 shows a link (i), where $i = 1, \ldots, n$, connected to other links via mechanical joints at its ends. This link has one driving cutting point associated with the frame $\{T_{i+1}\}$ and one driven cutting point associated with the frame $\{B_i\}$. Thus, the joint (i) has one driven cutting point associated with the frame $\{B_i\}$ and one driving cutting point associated with frame $\{T_i\}$. As aforementioned, because the principle of VDC depends on the flow of virtual power through each subsystem, the dynamic equation will be derived with respect to the local frame $\{B_i\}$ associated with every subsystem or link. The dynamic equation

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

of the link (i) can be written as[20]

$$\boldsymbol{M}_{B_i}\,{}^{B_i}\dot{\boldsymbol{v}} + \boldsymbol{C}_{B_i}\,{}^{B_i}\boldsymbol{v} + \boldsymbol{g}_{B_i} = {}^{B_i}\boldsymbol{f}^*, \quad i = 1, \ldots, n, \tag{2}$$

where $\boldsymbol{M}_{B_i} \in \mathbb{R}^{6\times 6}$ is a symmetric positive definite mass matrix of the link (i). Because the mass matrix is derived with respect to the local frame $\{B_i\}$, it contains constant value elements.

Note that ${}^{B_i}\boldsymbol{v}$ and ${}^{B_i}\dot{\boldsymbol{v}} \in \mathbb{R}^6$ are the generalized velocity and acceleration vector of the link (i) with respect to the local frame $\{B_i\}$. These are augmented vectors such that they contain the translational and rotational components.

- $\boldsymbol{C}_{B_i} \in \mathbb{R}^{6\times 6}$ is a skew-symmetric matrix representing the Coriolis and centripetal terms of the link (i).
- $\boldsymbol{g}_{B_i} \in \mathbb{R}^6$ is the gravity vector of the link (i).
- ${}^{B_i}\boldsymbol{f}^* \in \mathbb{R}^6$ is the augmented net force/moment vectors at the frame$\{B_i\}$.

The augmented linear/angular velocity vector can be written as

$${}^{B_i}\boldsymbol{v} = \boldsymbol{z}\dot{\boldsymbol{q}}_{\boldsymbol{i}} + {}^{B_i}\boldsymbol{U}_{B_{i+1}}^{T}\,{}^{B_{i-1}}\boldsymbol{v}, \tag{3}$$

where $z = [000001]^T$ and $[001000]^T$ for the revolute and the prismatic joint respectively, $\dot{q}_i$ is the angular velocity of joint (i), and

$${}^{B_i}\boldsymbol{U}_{B_{i+1}} = \begin{bmatrix} {}^{B_i}\boldsymbol{R}_{B_{i+1}} & 0_{3\times 3} \\ \boldsymbol{S}({}^{B_i}r_{B_i,B_{i+1}})\,{}^{B_i}\boldsymbol{R}_{B_{i+1}} & {}^{B_i}\boldsymbol{R}_{B_{i+1}} \end{bmatrix}, \tag{4}$$

where ${}^{B_i}\boldsymbol{R}_{B_{i+1}}$ is the rotation matrix of frame $\{B_{i+1}\}$ with respect to $\{B_i\}$, and $S(\cdot)$ is a skew-symmetric matrix of the vector ${}^{B_i}\boldsymbol{r}_{B_i,B_{i+1}}$ that represents the vector directed from the frame $\{B_i\}$ to the frame $\{B_{i+1}\}$.

Equation (4) represents the linear/angular velocity transformation matrix $\in \mathbb{R}^{6\times 6}$ that transforms the linear/angular velocity vector from the frame $\{B_{i+1}\}$ to the frame $\{B_i\}$. As with Jacobians, the force/moment transformation uses the transpose of Eq. (4).[44]

The controller structure for known physical dynamic parameters can be written according to Slotine and Li[45] and Zhu[20]

$${}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} = \boldsymbol{M}_{B_i}\,{}^{B_i}\dot{\boldsymbol{v}}_r + \boldsymbol{C}_{B_i}\,{}^{B_i}\boldsymbol{v}_r + \boldsymbol{g}_{B_i} + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s}, \tag{5}$$

where${}^{B_i}v_r \in \mathbb{R}^6$is the required velocity or the reference velocity which can be written according to Eq. (3) as follows:

$${}^{B_i}\boldsymbol{v}_r = \boldsymbol{z}\dot{\boldsymbol{q}}_{ri} + {}^{B_i}\boldsymbol{U}_{B_{i+1}}^{T}\,{}^{B_i-1}\boldsymbol{v}_{\boldsymbol{r}}. \tag{6}$$

Note that $\boldsymbol{K}_i \in \mathbb{R}^{6\times 6}$ is a diagonal matrix representing the gain of the feedback controller, and

$${}^{B_i}\boldsymbol{s} = {}^{B_i}\boldsymbol{v}_r - {}^{B_i}\boldsymbol{v}, \tag{7}$$

which is a measure of the tracking accuracy.[45]

Substituting Eqs. (5) and (7) into Eq. (2), we get the stable closed loop control system

$$\boldsymbol{M}_{B_i}\,{}^{B_i}\dot{\boldsymbol{s}} + \boldsymbol{C}_{B_i}\,{}^{B_i}\boldsymbol{s} + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s} = 0. \tag{8}$$

This implies that ${}^{B_i}\boldsymbol{s}$ is an output of stable first-order filter, assuming that ${}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*}$is equal to ${}^{B_i}\boldsymbol{f}^*$. Let us define a scalar non-negative function candidate as

$$V_{l_i} = (1/2)^{B_i}\boldsymbol{s}^T\boldsymbol{M}_{B_i}\,{}^{B_i}\boldsymbol{s}. \tag{9}$$

Its time derivative can be computed as

$$\dot{V}_{l_i} = {}^{B_i}\boldsymbol{s}^T\boldsymbol{M}_{B_i}\,{}^{B_i}\boldsymbol{s}. \tag{10}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

To simplify Eq. (10) further and get the desired result of the virtual stability of the link (i), we need an equivalent expression for the value $\boldsymbol{M}_{B_i}{}^{B_i}\dot{\boldsymbol{s}}$ .Therefore, by subtracting Eq. (2) from Eq. (5), we get

$$\begin{aligned} {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*} &= \boldsymbol{M}_{B_i}{}^{B_i}\dot{\boldsymbol{v}}_r + \boldsymbol{C}_{B_i}{}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} + \boldsymbol{g}_{B_i} + \boldsymbol{K}_i{}^{B_i}\boldsymbol{s} - \boldsymbol{M}_{B_i}{}^{B_i}\dot{\boldsymbol{v}} - \boldsymbol{C}_{B_i}{}^{B_i}\boldsymbol{v} - \boldsymbol{g}_{B_i} \\ &= M_{B_i}{}^{B_i}\dot{\boldsymbol{s}} + C_{B_i}{}^{B_i}s + K_i{}^{B_i}s. \end{aligned} \tag{11}$$

Rearranging Eq. (11), we obtain

$$\boldsymbol{M}_{B_i}{}^{B_i}\dot{\boldsymbol{s}} = {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*} - \boldsymbol{K}_i{}^{B_i}\boldsymbol{s} - \boldsymbol{C}_{B_i}{}^{B_i}\boldsymbol{s}. \tag{12}$$

Substituting Eq. (12) into Eq. (10), we get

$$\dot{V}_{l_i} = {}^{B_i}\boldsymbol{s}^T({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*} - \boldsymbol{K}_i{}^{B_i}\boldsymbol{s} - \boldsymbol{C}_{B_i}{}^{B_i}\boldsymbol{s}) = {}^{B_i}\boldsymbol{s}^T({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*}) - {}^{B_i}\boldsymbol{s}^T\boldsymbol{K}_i{}^{B_i}\boldsymbol{s} - {}^{B_i}\boldsymbol{s}^T\boldsymbol{C}_{B_i}{}^{B_i}\boldsymbol{s}. \tag{13}$$

The last term vanishes due to the skew-matrix property of the Coriolis and centripetal matrix. Thus, Eq. (13) becomes

$$\dot{V}_{l_i} = -{}^{B_i}\boldsymbol{s}^T\boldsymbol{K}_i{}^{B_i}\boldsymbol{s} + {}^{B_i}\boldsymbol{s}^T({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*}). \tag{14}$$

We will explore the last term further to know the notion of the virtual stability of each subsystem that ensures the stability of the entire robotic system. The first term of Eq. (13) can be expanded to give

$$\left({}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{v}\right)^T\left({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*}\right) = \left({}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{v}\right)^T\left({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{f}\right) - \left({}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{v}\right)^T{}^{B_i}\boldsymbol{U}_{T_{i+1}}\left({}^{T_{i+1}}\boldsymbol{f}_{\boldsymbol{r}} - {}^{T_{i+1}}\boldsymbol{f}\right), \tag{15}$$

where

$${}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} = {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{U}_{T_{i+1}}{}^{T_{i+1}}\boldsymbol{f}_{\boldsymbol{r}},\ {}^{B_i}\boldsymbol{f}^{*} = {}^{B_i}\boldsymbol{f} - {}^{B_i}\boldsymbol{U}_{T_{i+1}}{}^{T_{i+1}}\boldsymbol{f}. \tag{16}$$

According to Definition 1, Eq. (15) becomes

$$\left({}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{v}\right)^T\left({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*}\right) = p_{B_i} - p_{T_{i+1}}, \tag{17}$$

where p_{B_i} and $p_{T_{i+1}}$ are the virtual power flows at the driven cutting point and the driving cutting point respectively of the link (i). Thus, Eq. (14) can be written as

$$\dot{V}_{l_i} = -{}^{B_i}\boldsymbol{s}^T\boldsymbol{K}_i{}^{B_i}\boldsymbol{s} + p_{B_i} - p_{T_{i+1}}. \tag{18}$$

The second and third terms of Eq. (18) represent the virtual power flows (Definition 1). The appearance of virtual power flows at the time derivative of a non-negative function is necessary to ensure that this subsystem is virtually stable. This term is equal and opposite in sign with the adjacent subsystem at the interaction. Thus, the scalar non-negative function of the entire system will not contain these terms, and it will be negative semi-definite. This will be obvious in the next sections. We will close up this section by the following important notions regarding virtual stability and the stability of the entire system.

Definition 3.[20] A subsystem that is virtually decomposed from a complex robot is said to be virtually stable with an affiliated vector $x(t)$ being a virtual function in L_∞ and its affiliated vector $y(t)$ being a virtual function in L_2, if and only if there exists a non-negative function:

$$v(t) \geq (1/2)\boldsymbol{s}(t)^T\boldsymbol{M}\boldsymbol{s}(t) \tag{19}$$

such that

$$\dot{v} \leq -\boldsymbol{y}(t)^T\boldsymbol{Q}\boldsymbol{y}(t) - \boldsymbol{x}(t) + \sum_{\{A\}\in\varphi} p_A - \sum_{\{C\}\in\psi} p_c \tag{20}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

hold, subject to

$$\int_0^\infty x(t)d\tau \geq -\gamma_s. \tag{21}$$

With $0 \leq \gamma_s < \infty$, where $\boldsymbol{M}$ and $\boldsymbol{Q}$ are two positive-definite matrices, set φ contains frames being placed at the driven cutting point of this subsystem and set ψ contains frames being placed at the driven cutting points of this subsystem, and p_A and p_C denote the virtual power flows.
This definition explains the notion of the virtual stability of the subsystem and the structure of the derivative of the non-negative function of the subsystem.

Lemma 1.[20] Consider a non-negative piecewise continuous function $v(t)$ defined as

$$v(t) \geq (1/2)\boldsymbol{s}(t)^T \boldsymbol{M}\boldsymbol{s}(t), \tag{22}$$

with $\boldsymbol{s}(t) \in \mathbb{R}^n$, $n \geq 1$ and $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ being a symmetric positive-definite matrix. If the time derivative of $v(t)$ is governed by

$$\dot{v} \leq -\boldsymbol{y}(t)^T \boldsymbol{Q}\boldsymbol{y}(t) - x(t), \tag{23}$$

where $\boldsymbol{y}(t) \in \mathbb{R}^m$, $m \geq 1$, $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ is a symmetric positive-definite matrix, and $x(t)$ is subject to

$$\int_0^\infty x(t)d\tau \geq -\gamma_s. \tag{24}$$

With $0 \leq \gamma_s < \infty$, it follows that

$$\begin{aligned} &\boldsymbol{v}(t) \in L_\infty, \\ &\boldsymbol{s}(t) \in L_\infty, \\ &\boldsymbol{y}(t) \in L_2. \end{aligned} \tag{25}$$

This lemma demonstrates the L_2 and L_∞ stability of the entire system. This lemma is equivalent to Barbalat's lemma.

Theorem 1.[20] Consider a complex robot that is virtually decomposed into subsystems. If every subsystem is virtually stable in the sense of Definition 3, then all virtual functions in L_2 are functions in L_2 and all virtual functions in L_∞ are functions in L_∞.

This theorem explains the equivalence between the virtual stability of every subsystem and the stability of the entire complex robot.

3. Review of the FAT

FAT is an essential tool for approximation of the unknown parameters of the dynamic system for adaptive control purposes. In general, the procedure employed in constructing the FAT-based adaptive control includes the selection of a suitable approximator for the uncertain parameters of the dynamic system, choosing an appropriate update adaptation law for the weighting coefficient and designing the controller structure [46]. In this section, we will explain briefly the notion of the FAT, but the adaptation law and the controller are discussed later in the next section.

There are different classes of approximators, such as polynomials, splines, radial basis functions, cerebellar model articulation controller (CMAC), multilayer perceptron, fuzzy approximation and wavelets.[46] In this paper, we will focus on the orthogonal functions due to their capability of achieving minimum approximation errors as shown in the following theorem.

Theorem 2.[47] If $\{\phi_0, \ldots, \phi_n\}$ is an orthogonal set of functions on an interval $[a, b]$ with respect to the weight function, then the least-squares approximation to f on $[a, b]$ with respect to the weight

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

function is

$$P(t) = \sum_{k=0}^{n} a_k \phi_k(t). \tag{26}$$

Thus, we can approximate every function by orthogonal basis functions as

$$f(t) \cong P(t) = \sum_{k=0}^{n} a_k \phi_k(t). \tag{27}$$

We can write Eq. (27) in vector form as

$$f(t) \cong \boldsymbol{w}^T \boldsymbol{z}, \tag{28}$$

where $\boldsymbol{w}$ is the weighting vector and z is the orthogonal basis vector.

The dynamic equation of the link (i) subsystem or even the joint (i) subsystem (in the case of a spherical joint) consists of dynamic matrices, which are required to be approximated by FAT for adaptive control purposes. This requires separation of the dynamic matrices into two matrices representing the weighting constant matrix and the basis function matrix. Thus, the curse of dimensionality should be faced.[46] There are two main representations that have been introduced by Huang and Chien.[40] The first one is the tensor product of the weighting and the basis matrices which results in reasonable computational complexity. However, this representation has not been used due to its incompatibility to the conventional matrix operations. In contrast, the second representation shown below is used due to its compatibility.[40]

Let the matrix $\boldsymbol{M} \in R^{n \times n}$

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & \dots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \dots & m_{nn} \end{bmatrix} \cong \begin{bmatrix} \boldsymbol{w}_{11}^T z_{11} & \dots & \boldsymbol{w}_{1n}^T z_{1n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{w}_{n1}^T z_{n1} & \dots & \boldsymbol{w}_{nn}^T z_{nn} \end{bmatrix}. \tag{29}$$

Equation (29) can be represented as follows:

$$M = \boldsymbol{W}^T \boldsymbol{Z}, \tag{30}$$

where

$$\boldsymbol{W}^T = \left[\begin{array}{cccc|cccc|c|cccc} \boldsymbol{w}_{11}^T & 0 & \dots & 0 & \boldsymbol{w}_{12}^T & 0 & \dots & 0 & \dots & \boldsymbol{w}_{1n}^T & 0 & \dots & 0 \\ 0 & \boldsymbol{w}_{21}^T & \dots & 0 & 0 & \boldsymbol{w}_{22}^T & \dots & 0 & \dots & 0 & \boldsymbol{w}_{2n}^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{w}_{n1}^T & 0 & 0 & \dots & \boldsymbol{w}_{n2}^T & \cdots & 0 & 0 & \dots & \boldsymbol{w}_{nn}^T \end{array}\right], \tag{31}$$

$$\boldsymbol{Z}^T = \left[\begin{array}{cccc|cccc|c|cccc} z_{11}^T & z_{21}^T & \dots & z_{n1}^T & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & z_{12}^T & z_{22}^T & \dots & z_{n2}^T & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \cdots & z_{1n}^T & z_{2n}^T & \dots & z_{nn}^T \end{array}\right], \tag{32}$$

If the number of terms of the approximation orthogonal functions is $\boldsymbol{\beta}$, every element in the matrices $\boldsymbol{W}$, $\boldsymbol{Z}$ including the zero elements is a vector of $\boldsymbol{R}^{\beta \times 1}$. Thus, $\boldsymbol{W} \in \boldsymbol{R}^{n^2 \beta \times n}$ and $\boldsymbol{Z} \in \boldsymbol{R}^{n^2 \beta \times n}$. The same technique can be used to approximate the vector.

Despite the useful form of Eqs. (30)–(32), this representation can lead to a large computation time, which grows exponentially with the dimension of the matrix. The computational complexity issue will be discussed in Section 4.4. To reduce this computational complexity, the lattice-based approximation and CMAC can be used as alternative tools, which could be considered in future work.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

4. The Dynamics and FAT-Based Adaptive VDC

In this section, we will focus on the modification of adaptive VDC using FAT instead of the regressor-based approach. We will prove the validity of this approach in guaranteeing the virtual stability of each subsystem and the stability of the entire system. A comparison table between the FAT-based adaptive VDC and the regressor-based adaptive VDC will be listed at the end of this section, including a summary of the controller structure, the update adaptive laws and the computational complexity issue.

4.1. The dynamics and FAT-based adaptive VDC of the link (i)

The dynamic equation of the link (i) subsystem and its controller can be described by Eqs. (2) and (5), respectively. However, there are likely uncertain dynamic parameters in the dynamic equation of the subsystem. Therefore, the estimated values should be used. Thus, the adaptive controller structure can be written as

$$ {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} = \hat{\boldsymbol{M}}_{B_i}\,{}^{B_i}\dot{\boldsymbol{v}}_{\boldsymbol{r}} + \hat{\boldsymbol{C}}_{B_i}\,{}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} + \hat{\boldsymbol{g}}_{B_i} + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s}. \tag{33}$$

Using Eq. (7) and substituting Eq. (33) into Eq. (2), we get

$$ \boldsymbol{M}_{B_i}({}^{B_i}\dot{\boldsymbol{v}}_r - {}^{B_i}\dot{\boldsymbol{s}}) + \boldsymbol{C}_{B_i}({}^{B_i}\boldsymbol{v}_r - {}^{B_i}\boldsymbol{s}) + \boldsymbol{g}_{B_i} = \hat{\boldsymbol{M}}_{B_i}\,{}^{B_i}\dot{\boldsymbol{v}}_r + \hat{\boldsymbol{C}}_{B_i}\,{}^{B_i}\boldsymbol{v}_r + \hat{\boldsymbol{g}}_{B_i} + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s}. \tag{34}$$

Expanding Eq. (34), we get the closed-loop control system

$$ \boldsymbol{M}_{B_i}\,{}^{B_i}\dot{\boldsymbol{s}} + \boldsymbol{C}_{B_i}\,{}^{B_i}\boldsymbol{s} + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s} = \tilde{\boldsymbol{M}}_{B_i}\,{}^{B_i}\dot{\boldsymbol{v}}_r + \tilde{\boldsymbol{C}}_{B_i}\,{}^{B_i}\boldsymbol{v}_r + \tilde{\boldsymbol{g}}_{B_i}, \tag{35}$$

where

$$ \tilde{\boldsymbol{M}}_{B_i} = \boldsymbol{M}_{B_i} - \hat{\boldsymbol{M}}_{B_i}, \tilde{\boldsymbol{C}}_{B_i} = \boldsymbol{C}_{B_i} - \hat{\boldsymbol{C}}_{B_i}, \tilde{\boldsymbol{g}}_{B_i} = \boldsymbol{g}_{B_i} - \hat{\boldsymbol{g}}_{B_i}. $$

The objective of the adaptive control is to estimate $\boldsymbol{M}_{B_i}$, $\boldsymbol{C}_{B_i}$, $\boldsymbol{g}_{B_i}$ such that $\tilde{\boldsymbol{M}}_{B_i}, \tilde{\boldsymbol{C}}_{B_i}, \tilde{\boldsymbol{g}}_{B_i} \to \boldsymbol{0}$, which results in position and approximation errors converge to zero. In the FAT-based adaptive approach, we will treat every element of the matrices $\boldsymbol{M}_{B_i}, \boldsymbol{C}_{B_i}, \boldsymbol{g}_{B_i}$ as functions of time. Thus, each element of $\boldsymbol{M}_{B_i}, \boldsymbol{C}_{B_i}, \boldsymbol{g}_{B_i}$ can be approximated by orthogonal functions that have a sufficient number of terms to give an accurate representation of the estimated parameters. In this paper, Chebyshev polynomials are used for the approximation of the system matrices. We will use the presentation of the FAT described before to approximate the matrices and vectors of the dynamic subsystem equation. Thus,

$$ \boldsymbol{M}_{B_i} = \boldsymbol{W}_{M_i}^{T}\boldsymbol{Z}_{M_i} + \boldsymbol{\varepsilon}_{M_i}, \boldsymbol{C}_{B_i} = \boldsymbol{W}_{C_i}^{T}\boldsymbol{Z}_{C_i} + \boldsymbol{\varepsilon}_{C_i}, \boldsymbol{g}_{B_i} = \boldsymbol{W}_{g_i}^{T}\boldsymbol{z}_{g_i} + \boldsymbol{\varepsilon}_{g_i}, \tag{36}$$

where $\boldsymbol{W}_{M_i} \in \mathbb{R}^{6^2\beta_{Mi}\times 6}$, $\boldsymbol{W}_{C_i} \in \mathbb{R}^{6^2\beta_{Ci}\times 6}$ and $\boldsymbol{W}_{g_i} \in \mathbb{R}^{6\beta_{gi}\times 6}$ are the constant weighting matrices of the mass matrix, Coriolis and centripetal matrix and the gravity vector, respectively, for the link (i), while $\boldsymbol{Z}_{M_i} \in \mathbb{R}^{6^2\beta_{Mi}\times 6}$, $\boldsymbol{Z}_{C_i} \in \mathbb{R}^{6^2\beta_{Ci}\times 6}$ and $\boldsymbol{z}_{g_i} \in \mathbb{R}^{6\beta_{gi}\times 1}$ represent the corresponding basis function matrices. $\boldsymbol{\varepsilon}_{M_i}$, $\boldsymbol{\varepsilon}_{C_i}$ and $\boldsymbol{\varepsilon}_{g_i}$ are the approximation errors.

The estimated matrices of $\boldsymbol{M}_{B_i}, \boldsymbol{C}_{B_i}, \boldsymbol{g}_{B_i}$ can be approximated as

$$ \hat{\boldsymbol{M}}_{Bi} = \hat{\boldsymbol{W}}_{M_i}^{T}\boldsymbol{Z}_{M_i}, \quad \hat{C}_{B_i} = \hat{\boldsymbol{W}}_{C_i}^{T}Z_{C_i}, \quad \hat{\boldsymbol{g}}_{B_i} = \hat{\boldsymbol{W}}_{g_i}^{T}\boldsymbol{Z}g_i. \tag{37}$$

Thus, we can rewrite Eqs. (2), (33) and (35) as follows:

$$ {}^{B_i}\boldsymbol{f}^{*} = \hat{\boldsymbol{W}}_{M_i}^{T}\boldsymbol{Z}_{M_i}^{B_i}\dot{\upsilon} + \hat{\boldsymbol{W}}_{C_i}^{T}Z_{C_i}^{B_i}\upsilon + \hat{\boldsymbol{W}}_{g_i}^{T}\boldsymbol{Z}g_i + \sum \boldsymbol{\varepsilon}, \tag{38}$$

$$ {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} = \hat{\boldsymbol{W}}_{M_i}^{T}\boldsymbol{Z}_{M_i}^{B_i}\dot{\boldsymbol{v}}_r + \hat{\boldsymbol{W}}_{C_i}^{T}\boldsymbol{Z}_{C_i}^{B_i}\boldsymbol{v}_r + \hat{\boldsymbol{W}}_{g_i}^{T}\boldsymbol{Z}g_i + \boldsymbol{K}_i^{B_i}\boldsymbol{s}, \tag{39}$$

$$ \boldsymbol{M}_{B_i}\,{}^{B_i}\dot{\boldsymbol{s}} + \boldsymbol{C}_{B_i}\,{}^{B_i}\boldsymbol{s} + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s} = \tilde{\boldsymbol{W}}_{M_i}^{T}\boldsymbol{Z}_{M_i}\,{}^{B_i}\dot{\boldsymbol{v}}_r + \tilde{\boldsymbol{W}}_{C_i}^{T}\boldsymbol{Z}_{C_i}\,{}^{B_i}\boldsymbol{v}_r + \tilde{\boldsymbol{W}}_{g_i}^{T}\boldsymbol{z}_{g_i} + \sum \boldsymbol{\varepsilon}. \tag{40}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

We will assume that a sufficient number of basis functions are used in the approximation. Consequently, the approximation errors may be neglected in Eqs. (38) and (40).[40] It is the time to derive the update adaptive law of the weighting matrices, which have constant value elements, for the link (i) subsystem and prove its virtual stability. The scalar non-negative function (V_{l_i}) of the link (i) can be calculated as

$$V_{l_i} = \frac{1}{2} {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} \boldsymbol{M}_{B_i} {}^{B_i}\boldsymbol{s} + \boldsymbol{Tr}(\tilde{\boldsymbol{W}}_{M_i}^{\boldsymbol{T}} \boldsymbol{Q}_{M_i} \tilde{\boldsymbol{W}}_{M_i} + \tilde{\boldsymbol{W}}_{C_i}^{\boldsymbol{T}} \boldsymbol{Q}_{C_i} \tilde{\boldsymbol{W}}_{C_i} + \tilde{\boldsymbol{W}}_{g_i}^{\boldsymbol{T}} \boldsymbol{Q}_{g_i} \tilde{\boldsymbol{W}}_{g_i}), \tag{41}$$

where $\boldsymbol{Q}_{M_i} \in \mathbb{R}^{6^2\beta_{Mi} \times 6^2\beta_{Mi}}$, $\boldsymbol{Q}_{C_i} \in \mathbb{R}^{6^2\beta_{Ci} \times 6^2\beta_{Ci}}$ and $\boldsymbol{Q}_{g_i} \in \mathbb{R}^{6\beta_{Mi} \times 6\beta_{Mi}}$ are positive-definite weighting matrices of the mass matrix, Coriolis and centripetal matrix and the gravity vector, respectively, for the link (i). These matrices represent the adaptation gain matrices, which are responsible for speeding up the adaptation procedure.

Taking the derivative of Eq. (41) and substituting Eq. (7), we obtain

$$\begin{aligned} \dot{V}_{l_i} = {} & ({}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{v})^{\boldsymbol{T}} \boldsymbol{M}_{B_i} {}^{B_i}\dot{\boldsymbol{v}}_{\boldsymbol{r}} - \left({}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{v}\right)^{\boldsymbol{T}} \boldsymbol{M}_{B_i} {}^{B_i}\dot{\boldsymbol{v}} \\ & - \boldsymbol{Tr}\left(\tilde{\boldsymbol{W}}_{M_i}^{\boldsymbol{T}} \boldsymbol{Q}_{M_i} \dot{\tilde{\boldsymbol{W}}}_{M_i} + \tilde{\boldsymbol{W}}_{C_i}^{\boldsymbol{T}} \boldsymbol{Q}_{C_i} \dot{\tilde{\boldsymbol{W}}}_{C_i} + \tilde{\boldsymbol{W}}_{g_i}^{\boldsymbol{T}} \boldsymbol{Q}_{g_i} \dot{\tilde{\boldsymbol{W}}}_{g_i}\right). \end{aligned} \tag{42}$$

Subtracting Eq. (38) from Eq. (39), we get

$$-\boldsymbol{M}_{B_i} {}^{B_i}\dot{\boldsymbol{v}} = {}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*} - \hat{\boldsymbol{M}}_{B_i} {}^{B_i}\dot{\boldsymbol{v}}_{\boldsymbol{r}} - \hat{\boldsymbol{C}}_{B_i} {}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} - \boldsymbol{K}_i \left({}^{B_i}\boldsymbol{v}_r - {}^{B_i}\boldsymbol{v}\right) + \boldsymbol{C}_{B_i} {}^{B_i}\boldsymbol{v} + \boldsymbol{g}_{B_i} - \hat{\boldsymbol{g}}_{B_i}. \tag{43}$$

Substituting (43) into (42) with simplifications, we get

$$\begin{aligned} \dot{V}_{l_i} = {} & -{}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} \boldsymbol{K}_i {}^{B_i}\boldsymbol{s} + {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} \left({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*}\right) - \boldsymbol{Tr}\left(\tilde{\boldsymbol{W}}_{M_i}^{\boldsymbol{T}} \left(-\boldsymbol{Z}_{M_i} \dot{\boldsymbol{v}}_r {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} + Q_{M_i} \dot{\tilde{\boldsymbol{W}}}_{M_i}\right)\right) \\ & - \boldsymbol{Tr}\left(\tilde{\boldsymbol{W}}_{C_i}^{\boldsymbol{T}} \left(-\boldsymbol{Z}_{C_i} {}^{B_i}\boldsymbol{v}_r {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} + \boldsymbol{Q}_{C_i} \dot{\tilde{\boldsymbol{W}}}_{C_i}\right)\right) - \boldsymbol{Tr}\left(\tilde{\boldsymbol{W}}_{g_i}^{\boldsymbol{T}} \left(-\boldsymbol{z}_{g_i} {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} + \boldsymbol{Q}_{g_i} \dot{\tilde{\boldsymbol{W}}}_{g_i}\right)\right). \end{aligned} \tag{44}$$

To get a virtual stable subsystem, it is necessary to zero each of the last three terms. Thus, we can get the update adaptive laws of Eq. (45):

$$\begin{aligned} \dot{\tilde{\boldsymbol{W}}}_{M_i} &= \boldsymbol{Q}_{M_i}^{-1} \boldsymbol{Z}_{M_i} {}^{B_i}\dot{\boldsymbol{v}}_r {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}}, \\ \dot{\tilde{\boldsymbol{W}}}_{C_i} &= \boldsymbol{Q}_{C_i}^{-1} \boldsymbol{Z}_{C_i} {}^{B_i}\boldsymbol{v}_{\boldsymbol{r}} {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} \\ \dot{\tilde{\boldsymbol{W}}}_{g_i} &= \boldsymbol{Q}_{g_i}^{-1} \boldsymbol{z}_{g_i} {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}}. \end{aligned} \tag{45}$$

We can conclude from Eq. (44):

$$\dot{V}_{l_i} \leq -{}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} \boldsymbol{K}_i {}^{B_i}\boldsymbol{s} + {}^{B_i}\boldsymbol{s}^{\boldsymbol{T}} \left({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_i}\boldsymbol{f}^{*}\right). \tag{46}$$

Equation (46) coincides with Eqs. (14) and (18) in Section 2 such that this subsystem (link (i)) is virtually stable according to Definition 3.

Let us investigate Eq. (46) further by applying it on the simulation example (planar 2-R robot).

For link 1:

$${}^{B_1}\boldsymbol{s}^{\boldsymbol{T}} \left({}^{B_1}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_1}\boldsymbol{f}^{*}\right) = p_{B_1} - p_{T_2}. \tag{47}$$

For link 2, which is free contact end

$${}^{B_2}\boldsymbol{s}^{\boldsymbol{T}} \left({}^{B_2}\boldsymbol{f}_{\boldsymbol{r}}^{*} - {}^{B_2}\boldsymbol{f}^{*}\right) = p_{B_2}. \tag{48}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

Consequently, the non-negative accompanying functions of each link are

$$\dot{V}_{l_1} \leq -{}^{B_1}\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{K}_1{}^{B_1}\boldsymbol{s} + p_{B_1} - p_{T_2}, \tag{49}$$

$$\dot{V}_{l_2} \leq -{}^{B_2}\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{K}_2{}^{B_2}\boldsymbol{s} + p_{B_2}. \tag{50}$$

Recall Eqs. (49) and (50) until Section 4.3 to make out how the virtual power terms will be cancelled after adding the non-negative accompanying functions of the links with that of the joints to produce the non-negative accompanying function of the entire system.

4.2. The dynamics and FAT-based adaptive control of the joint (i)

There are two types of drive transmission systems for robotic joint systems. The first is the direct drive joints, in which the inertia of the motor is included in the corresponding links.[20,15] In this case, the dynamics of the joint is neglected. The second type of the system deals with a high gear transmission assuming that the inertial forces/torques act along the joint axis.[20] In the latter case, the dynamic equation of the joint (i) can be described as[20,48,49]

$$J_i\ddot{q}_i + k_{ci}sign(\dot{q}_i) + k_{vi}\dot{q}_i + k_{si}sign(\dot{q}_i)e^{-(\dot{q}_i/\eta_{si})} + b_i = \tau_i^* \quad \text{for} \quad i = 1, \ldots, n, \tag{51}$$

where J_i represents the equivalent inertia of the joint (i), $q_i, \dot{q}_i, \ddot{q}_i$ are the angular position, velocity and acceleration of the ith joint, respectively, k_{ci}, k_{vi} and k_{si} denote the Coulomb friction coefficient, viscous friction coefficient and Stribeck friction coefficient, respectively, η_{si} is the rate of decay of the Stribeck friction effect, b_i is the friction offset term and τ_i^* represents the net torque applied to the joint (i). The net torque of the joint (i) can be described as

$$\tau_i^* = \tau_{ci} - \boldsymbol{z}^{T\,B_i}\boldsymbol{f}, \tag{52}$$

where τ_{ci} is the input control torque of the joint (i) and the second term represents the output torque of the joint (i) toward the links.

The required angular velocity and the corresponding velocity error of the joint (i) can be written as

$$\dot{q}_{ri} = \dot{q}_{di} + \lambda_i(q_{di} - q_i), \tag{53}$$

$$s_i = \dot{e}_i + \lambda_i e_i = (\dot{q}_{di} - \dot{q}_i) + \lambda_i(q_{di} - q_i), \tag{54}$$

where q_{di} and its derivatives refer to the desired states of the joint (i) subsystem, and λ_i is the position gain. From Eq. (53), we can get

$$\dot{q}_i = \dot{q}_{di} + \lambda_i(q_{di} - q_i) = \dot{q}_{ri} - s_i. \tag{55}$$

The adaptive controller of the joint (i) can be constructed considering the estimated parameters of Eq. (51) as

$$\tau_{ri}^* = \hat{J}_i\ddot{q}_{ri} + \hat{k}_{ci}sign\,(\dot{q}_{ri}) + \hat{k}_{vi}\dot{q}_{ri} + \hat{k}_{si}sign\,(\dot{q}_{ri})\,e^{-\left(\frac{\dot{q}_{ri}}{\eta_{si}}\right)} + \hat{b}_i + k_i s_i, \tag{56}$$

where k_i is the feedback gain of the suggested controller.

Using FAT-based adaptive control and collecting the Coulomb friction torque, Stribeck friction torque and the friction offset torque in one term for less time consumption, Eqs. (51), (56) and the closed-loop control can be expressed, respectively, as

$$\tau_i^* = \boldsymbol{w}_{J_i}^T\boldsymbol{z}_{I_i}(\dot{q}_{ri} - \dot{s}_i) + \boldsymbol{w}_{eqi}^T\boldsymbol{z}_{eqi} + \boldsymbol{w}_{vi}^T\boldsymbol{z}_{vi}(\dot{q}_{ri} - s_i) + \sum \boldsymbol{\varepsilon}, \tag{57}$$

$$\tau_{ri}^* = \hat{\boldsymbol{w}}_{J_i}^T\boldsymbol{z}_{I_i}\ddot{\boldsymbol{q}}_{ri} + \hat{\boldsymbol{w}}_{eqi}^T\boldsymbol{z}_{eqi} + \hat{\boldsymbol{w}}_{vi}^T\boldsymbol{z}_{vi}\dot{\boldsymbol{q}}_{ri} + k_i s_i, \tag{58}$$

$$J_i\dot{s}_i + k_{vi}s_i = \hat{\boldsymbol{w}}_{I_i}^T\boldsymbol{z}_{I_i}\dot{\boldsymbol{q}}_{ri} + \hat{\boldsymbol{w}}_{eqi}^T\boldsymbol{z}_{eqi} + \hat{\boldsymbol{w}}_{vi}^T\boldsymbol{z}_{vi}\dot{\boldsymbol{q}}_{ri} + \sum \boldsymbol{\varepsilon}, \tag{59}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

where $\boldsymbol{w}_{J_i}$ denotes the constant weighting vector of the equivalent inertia of the joint (i) having a dimension of $\beta_{J_i} \times 1$, $\boldsymbol{w}_{eqi}$ is the equivalent weighting vector of the Coulomb friction, Stribeck friction and friction offset torques with a dimension of $\beta_{eqi} \times 1$, $\boldsymbol{w}_{vi}$ refers to the constant weighting vector of the viscous friction coefficients having a dimension of $\beta_{vi} \times 1$, whereas the corresponding Chebyshev basis function vectors are $\boldsymbol{z}_{J_i}$, $\boldsymbol{z}_{eqi}$ and $\boldsymbol{z}_{vi}$ with dimensions of $\beta_{J_i} \times 1$, $\beta_{eqi} \times 1$ and $\beta_{vi} \times 1$, respectively. Note that $\hat{\boldsymbol{w}}_{J_i}$, $\hat{\boldsymbol{w}}_{eqi}$ and $\hat{\boldsymbol{w}}_{vi}$ refer to the difference between the actual weighting vector and the corresponding estimated ones.

We will assume that a sufficient number of basis functions are used in the approximation. Consequently, the approximation errors could be neglected in Eqs. (57) and (59).[40] It is the time to derive the update adaptive laws for the estimated dynamic parameters for the joint (i) and prove its virtual stability.

Let the scalar non-negative accompanying function (V_{ji}) of the joint (i) be defined as

$$V_{ji} = (1/2)J_i s_i^2 + (1/2)(\hat{\boldsymbol{w}}_{J_i}^T \boldsymbol{Q}_{J_i} \hat{\boldsymbol{w}}_{I_i} + \hat{\boldsymbol{w}}_{eqi}^T \boldsymbol{Q}_{eqi} \hat{\boldsymbol{w}}_{eqi} + \hat{\boldsymbol{w}}_{vi}^T \boldsymbol{Q}_{vi} \hat{\boldsymbol{w}}_{vi}), \tag{60}$$

where $\boldsymbol{Q}_{J_i} \in \mathbb{R}^{\beta_{Ji} \times \beta_{Ji}}$, $\boldsymbol{Q}_{eqi} \in \mathbb{R}^{\beta_{eqi} \times \beta_{eqi}}$ and $\boldsymbol{Q}_{vi} \in \mathbb{R}^{\beta_{vi} \times \beta_{vi}}$ are positive-definite weighting matrices of the equivalent joint inertia, equivalent matrix of the Coulomb, Stribeck and offset friction torques, and the viscous friction torque, respectively, for the joint (i). These matrices denote the adaptation gain matrices.

Taking the derivative of Eq. (60), we have

$$\dot{V}_{ji} = J_i s_i \dot{s}_i - \left(\hat{\boldsymbol{w}}_{J_i}^T \boldsymbol{Q}_{J_i} \dot{\hat{\boldsymbol{w}}}_{J_i} + \hat{\boldsymbol{w}}_{eqi}^T \boldsymbol{Q}_{eqi} \dot{\hat{\boldsymbol{w}}}_{eqi} + \hat{\boldsymbol{w}}_{vi}^T \boldsymbol{Q}_{vi} \dot{\hat{\boldsymbol{w}}}_{vi}\right). \tag{61}$$

Subtracting Eq. (57) from Eq. (58), we get

$$\boldsymbol{w}_{J_i}^T \boldsymbol{z}_{J_i} \dot{\boldsymbol{s}}_i = \hat{\boldsymbol{w}}_{J_i}^T \boldsymbol{z}_{I_i} \ddot{\boldsymbol{q}}_{ri} + \hat{\boldsymbol{w}}_{eqi}^T \boldsymbol{z}_{eqi} + \hat{\boldsymbol{w}}_{vi}^T \boldsymbol{z}_{vi} \dot{\boldsymbol{q}}_{ri} - \boldsymbol{w}_{vi}^T \boldsymbol{z}_{vi} s_i + \tau_{ri}^* - \tau_i^* - k_i s_i. \tag{62}$$

By substituting Eq. (62) into Eq. (61), we get the following relation:

$$\begin{aligned} \dot{V}_{ji} = {} & -k_i s_i^2 - k_{vi} s_i^2 + s_i(\tau_{ri}^* - \tau_i^*) - (\hat{\boldsymbol{w}}_{J_i}^T (\boldsymbol{Q}_{J_i} \dot{\hat{\boldsymbol{w}}}_{J_i} - z_{I_i} \ddot{q}_{ri} s_i)) \\ & - \left(\hat{\boldsymbol{w}}_{eqi}^T (\boldsymbol{Q}_{eqi} \dot{\hat{\boldsymbol{w}}}_{eqi} - z_{eqi} s_i)\right) - (\hat{\boldsymbol{w}}_{vi}^T (\boldsymbol{Q}_{vi} \dot{\hat{\boldsymbol{w}}}_{vi} - z_{vi} \dot{q}_{ri} s_i)). \end{aligned} \tag{63}$$

By zeroing the last five terms to get update adaptive laws for the weighting vectors, we have

$$\dot{\hat{\boldsymbol{w}}}_{J_i} = \boldsymbol{Q}_{J_i}^{-1} \boldsymbol{z}_{J_i} \ddot{q}_{ri} s_i, \quad \dot{\hat{\boldsymbol{w}}}_{eqi} = \boldsymbol{Q}_{eqi}^{-1} \boldsymbol{z}_{eqi} s_i, \quad \dot{\hat{\boldsymbol{w}}}_{vi} = \boldsymbol{Q}_{vi}^{-1} \boldsymbol{z}_{vi} \dot{\boldsymbol{q}}_{ri} s_i. \tag{64}$$

From Eq. (63), we have

$$\dot{V}_{ji} \leq -k_i s_i^2 - k_{vi} s_i^2 + s_i(\tau_{ri}^* - \tau_i^*). \tag{65}$$

The third term of Eq. (65) refers to the difference in the virtual power flow through the joint (i). We expand the last term further:[20]

$$\begin{aligned} s_i(\tau_{ri}^* - \tau_i^*) &= -(\dot{q}_{ri} - \dot{q}_i) \boldsymbol{z}^T \left({}^{B_i}\boldsymbol{f}_{\boldsymbol{r}} - {}^{B_i}\boldsymbol{f}\right) \\ &= \left[\left({}^{B_i}\boldsymbol{v}_r - {}^{B_i}\boldsymbol{v}\right) - {}^{T_i}\boldsymbol{U}_{B_i}^T \left({}^{T_i}\boldsymbol{v}_r - {}^{T_i}\boldsymbol{v}\right)\right]^T \left({}^{T_i}\boldsymbol{f}_r - {}^{T_i}\boldsymbol{f}\right) \\ &= -\left({}^{B_i}\boldsymbol{v}_r - {}^{B_i}\boldsymbol{v}\right)^T \left({}^{B_i}\boldsymbol{f}_r - {}^{B_i}\boldsymbol{f}\right) + \left({}^{T_i}\boldsymbol{v}_r - {}^{T_i}\boldsymbol{v}\right)^T {}^{T_i}\boldsymbol{U}_{B_i}^T \left({}^{B_i}\boldsymbol{f}_r - {}^{B_i}\boldsymbol{f} = -p_{B_i} + p_{T_i}. \right. \end{aligned} \tag{66}$$

Thus, Eq. (65) becomes

$$\dot{V}_{ji} \leq -k_i s_i^2 - k_{vi} s_i^2 - p_{B_i} + p_{T_i}. \tag{67}$$

According to Definition 3, the joint (i) subsystem is virtually stable. Let us apply Eq. (67) to the first simulation example (2-R manipulator).

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

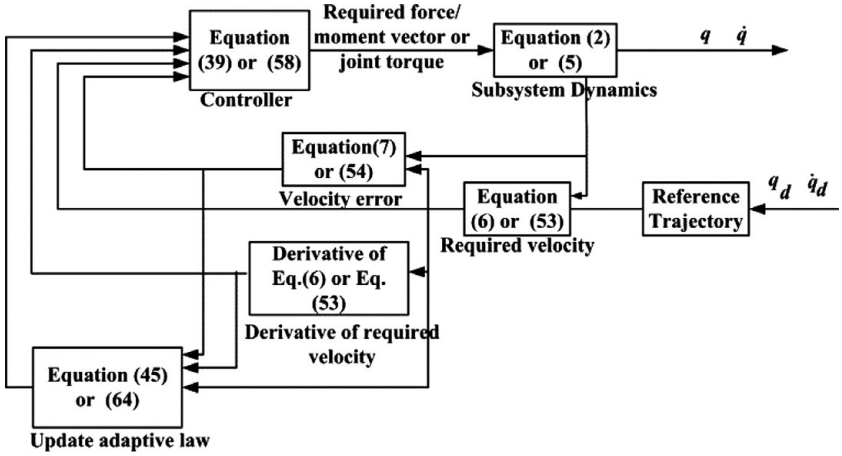


Fig. 2. FAT-based adaptive VDC algorithm structure.

For joint 1

$$s_1(\tau_{r1}^* - \tau_1^*) = -p_{B_1}. \tag{68}$$

For joint 2

$$s_2(\tau_{r2}^* - \tau_2^*) = -p_{B_2} + p_{T_2}. \tag{69}$$

Consequently, the non-negative accompanying functions of the joints become

$$\dot{V}_{j1} \le -k_{j1}s_1^2 - k_{v1}s_1^2 - p_{B_1}, \tag{70}$$

$$\dot{V}_{j2} \le -k_{j2}s_2^2 - k_{v2}s_2^2 - p_{B_2} + p_{T_2}. \tag{71}$$

Recall (70) and (71) until the next subsection to see how the virtual power terms will be cancelled after adding the non-negative accompanying functions of the links with the joints to produce a non-negative function of the entire system.

4.3. The stability of the entire system

Let us select the non-negative function of the entire system as follows:

$$V = \sum_{i=1}^{n} V_{li} + \sum_{i=1}^{n} V_{ji}. \tag{72}$$

Substituting Eqs. (46) and (67) into Eq. (72), we get

$$\dot{V} \le -\sum_{i=1}^{n} {}^{B_i}\boldsymbol{s}^T \boldsymbol{K}_i {}^{B_i}\boldsymbol{s} + \sum_{i=1}^{n} p_{B_i} - p_{T_{i+1}} - \sum_{i=1}^{n}(k_{ji}s_i^2 + k_{vi}s_i^2 + \sum_{i=1}^{n} -p_{B_i} + p_{T_i}. \tag{73}$$

For our simulation example, we obtain

$$\dot{V} \le -\sum_{i=1}^{2} {}^{B_i}\boldsymbol{s}^T \boldsymbol{K}_i {}^{B_i}\boldsymbol{s} - \sum_{i=1}^{2}\left(k_{ji}s_i^2 + k_{vi}s_i^2\right). \tag{74}$$

According to Eq. (74), Lemma 1 and Theorem 1, the entire system is stable in the sense of the L_2 and L_∞ stability. One can follow Zhu,[20] who proves the asymptotic stability of the states of the entire system. Figure 2 shows the structure of the FAT-based adaptive VDC for any subsystem (link and joint).

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

Table I. Computational complexities of some arithmetic operations.[50]

$\alpha \in \mathbb{R}^1, \mathbf{A} \in \mathbb{R}^{M\times N}$, $\mathbf{C} \in \mathbb{R}^{N\times L}$, $\mathbf{D} \in \mathbb{R}^{N\times N}$, $\mathbf{a}, b \in \mathbb{R}^N$ and $\boldsymbol{c} \in \mathbb{R}^M$			
Expression	Description	Products flops	Summation flops
αa	Vector scaling	N	–
αA	Matrix scaling	MN	–
$\boldsymbol{a}^T\boldsymbol{b}$	Inner product	N	$N-1$
$\boldsymbol{a}c^T$	Outer product	MN	–
Ab	Matrix–vector product	MN	$M(N-1)$
AC	Matrix–matrix product	MNL	$ML(N-1)$
AD	Diagonal matrix–matrix product	MN	–

4.4. The computational complexity of the VDC

In this subsection, we will discuss the computational complexity of the FAT-based adaptive VDC and the regressor-based adaptive VDC for comparison purposes. We can classify the computations as follows:

1. The computational complexity of the coefficients of the dynamic NE equation of the link (i) and joint (i) subsystems denoted by Eqs. (2) and (51), respectively.
2. The computational complexity of the required velocity of the link (i) and joint (i) subsystems denoted by Eqs. (6) and (53), respectively.
3. The computational complexity of the velocity error vector of the link (i) and joint (i) subsystems denoted by Eqs. (7) and (54), respectively.
4. The computational complexity of the update adaptive law of the link (i) and joint (i) subsystems denoted by Eqs. (45) and (64) for the FAT approach and Eqs. (75) and (77) for the regressor approach.
5. The computational complexity of the adaptive controller of the link (i) and joint (i) subsystems denoted by Eqs. (33) and (58), respectively, for the FAT approach and Eqs. (76) and (78) for the regressor approach.

Remark 1. Table I shows the computational complexities of the most necessary arithmetic operations used in our computations.

Remark 2. For the dynamic equation of the link (i), it is usual to evaluate the computational complexities inherent in obtaining the coefficients of the dynamic NE equations.[13] However, all required computations are evaluated for the required velocity, velocity error, adaptive law and the controller.

Remark 3. Since the dynamic equation of the joint (i) subsystem is a scalar function, the computational complexities of the dynamic equations (51), the required velocity denoted by Eq. (53) and the velocity error denoted by Eq. (54) are neglected. We focus on the evaluation of the computational complexities of the vector operations of the joint (i) for the update adaptive law and its controller.

Remark 4. All the adaptation gains of Eqs. (45), (64), (75) and (77) are always considered diagonal matrices. As a result, this will alleviate the computations.

Tables II and III show the computational complexities required for the adaptive control of the link (i) and joint (i) subsystems, respectively. While Table IV includes a comparison between the FAT approach and the regressor matrix approach.

From the above tables, one can notice the following important points:

(1) The computational complexity of the control algorithm of the FAT for the link (i) is a first-degree function to the number of the basis functions. Although the apparent advantages of the FAT are that the adaptation law will evade the regressor matrix, the computational complexities seem insufficient in the implementation. This large computations result from the unfortunate representation of the dynamic matrices of the link subsystem into two sparse matrices of

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

Table II. Computational complexities of the link (i) including the FAT and the regressor matrix approaches.

Adaptive control approach	Control algorithm	Multiplication flops	Summation flops
Function approximation technique (FAT)	1.N-E dynamic equation denoted by Eq. (2).	243	138
	2. Required velocity vector denoted by Eq. (6).	71	50
	3. Velocity error vector denoted by Eq. (7).	42	36
	4. Update adaptive law denoted by Eq. (45).	1338β	360β
	5. Adaptive controller denoted by Eq. (39).	$2628\ \beta+108$	$2628\ \beta+12$
	Total (excluding the dynamic equation)	$3966\ \beta+221$	$2988\ \beta+98$
Regressor matrix approach	1. NE dynamic equation denoted by Eq. (2).	243	138
	2. Required velocity vector denoted by Eq. (6).	71	50
	3. Velocity error vector denoted by Eq. (7).	42	36
	4. Update adaptive law which can be written as follows: $\dot{\hat{\boldsymbol{\theta}}}_i = \boldsymbol{\Gamma}_{\mathbf{i}}^{-1} \boldsymbol{Y}_i^{\boldsymbol{T}\,{}^{B_i}}\boldsymbol{s}$(75)[b]	$12x^a$	$5x$
	5. Adaptive controller which can be selected as ${}^{B_i}\boldsymbol{f}_{\boldsymbol{r}}^{*} = \boldsymbol{Y}_i\hat{\boldsymbol{\theta}}_i + \boldsymbol{K}_i\,{}^{B_i}\boldsymbol{s}$ (76)	$6x+6$	$6x-6$
	Total (excluding the dynamic equation)	$18x+119$	$11x+80$

[a]Here, x denotes the uncertain dynamic parameters of the link (i) subsystem.
[b]Here, $\hat{\boldsymbol{\theta}}_{\boldsymbol{i}} \in \mathbb{R}^{x\times 1}$ is the uncertain dynamic vector, $\boldsymbol{\Gamma}_{\mathbf{i}} \in \mathbb{R}^{x\times x}$ represents the adaptive gain matrix which is a positive-definite matrix, and $\boldsymbol{Y}_{\boldsymbol{i}} \in \mathbb{R}^{6\times x}$ is the regressor matrix of the link (i).

Table III. Computational complexities of the joint (i) including the FAT and the regressor matrix approach.

Adaptive control approach	Control algorithm	Multiplication flops	Summation Flops
	4. Update adaptive law denoted by Eq. (64).	3β	–
Function approximation technique (FAT)	5. Adaptive controller denoted by Eq. (59).	3β	$3(\beta-1)$
	Total	6β	$3\beta-3$
Regressor matrix approach	4. Update adaptive law which can be written as follows: $\dot{\hat{\boldsymbol{\theta}}}_{\boldsymbol{i}} = \boldsymbol{\Gamma}_{\mathbf{ji}}^{-1}\boldsymbol{Y}_{ji}^{\boldsymbol{T}}(\dot{q}_{ri}-\dot{q}_i)$ (77)[b]	y^a	–
	5. Adaptive controller which can be selected as $\tau_{ri}^{*} = \boldsymbol{Y}_{ji}\hat{\boldsymbol{\theta}}^{ji} + k_{ji}(\dot{q}_{ri}-\dot{q}_i)$ (78)	y	$y-1$
	Total	$2y$	$y-1$

[a]Note that y is the number of the uncertain dynamic parameters of the joint (i).
[b]Here, $\hat{\boldsymbol{\theta}}_{ji} \in \mathbb{R}^{y\times 1}$ is the uncertain dynamic vector of the joint (i), $\boldsymbol{\Gamma}_{\mathbf{ji}} \in \mathbb{R}^{y\times y}$ represents the adaptive gain matrix which is a positive-definite matrix, and $\boldsymbol{Y}_{ji} \in \mathbb{R}^{1\times y}$is the regressor matrix of the joint (i).

dimensions $36\beta \times 6$. For the alleviation of the computational complexity, we refer to Section 3 for details.

(2) The FAT-based adaptive VDC can be used successfully for the control of the joint (i) subsystem as shown in Table III.

(3) One should do not forget that the regressor matrix is full of nonlinear functions of joint positions and their derivatives. In the evaluation of the computational complexity, it is assumed that each element of the regressor matrix consists of one term. From Table II, it can be noticed that the regressor approach for controlling the link (i) is more efficient in the implementation comparing with the FAT. However, if the number of the uncertain dynamic parameters increases, such as in the underwater robotics, the difficulties of the implementation of the regressor approach should be faced. Frankly speaking, FAT is a fascinating and strong method for adaptive control if the appropriate representation of approximation of the dynamic matrices is selected. As a rule of thumb, using the regressor approach to the adaptive control of the link (i) subsystem may be more efficient than the FAT. However, using FAT in the adaptive control of the joint (i) subsystem can give very good results due its effectively dealing with the friction effects. There is a tradeoff between the two methods.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

Table IV. A comparison between FAT-based adaptive VDC and the regressor-based adaptive VDC.

FAT-based adaptive VDC (the present work)	Regressor-based adaptive VDC[20,30]
1. For link (i), $i = 1, \ldots, n$	1. For link (i), $i = 1, \ldots, n$
• Dynamic equation	• Dynamic equation
$M_{B_i}{}^{B_i}\dot{v} + C_{B_i}{}^{B_i}v + g_{B_i} = {}^{B_i}f^*$ (2)	$M_{B_i}{}^{B_i}\overset{+}{v}C_{B_i}{}^{B_i}v + g_{B_i} = {}^{B_i}f^*$ (2) (2)
• Controller	• Controller
${}^{B_i}f_r^* = \hat{\dot{W}}_{M_i}^T Z_{M_i}{}^{B_i}\dot{v}_r + \hat{\dot{W}}_{C_i}^T Z_{C_i}{}^{B_i}v_r + \hat{\dot{W}}_{g_i}^T z_{g_i} + K_i{}^{B_i}s$ (39)	${}^{B_i}f_r^* = \hat{M}_{B_i}{}^{B_i}\dot{v}_r + \hat{C}^{B_i}v_r + \hat{g}_{B_i} + K_i{}^{B_i}s = Y_i\hat{\theta}_i + K_i{}^{B_i}s$ (76)
• Adaptive laws	• Adaptive law
$\hat{\dot{W}}_{M_i} = Q_{M_i}^{-1} Z_{M_i}{}^{B_i}\dot{v}_r{}^{B_i}s^T$ $\hat{\dot{W}}_{C_i} = Q_{C_i}^{-1} Z_{C_i}{}^{B_i}v_r{}^{B_i}s^T$ $\hat{\dot{W}}_{g_i} = Q_{g_i}^{-1} z_{g_i}{}^{B_i}s^T$ (45)	$\dot{\hat{\theta}}_i = \Gamma_{\mathbf{i}}^{-1} Y_i^{T\,B_i}s$ (75)
• The computational complexity	• The computational complexity
(a) Multiplication: $3966\beta + 221$	(a) Multiplication: $18x + 119$
(b) Summation: $2988\beta + 98$	(b) Summation: $11x + 80$
2. For joint (i),$i = 1, \ldots, n$	2. For joint (i), $i = 1, \ldots, n$
• Dynamic equation	• Dynamic equation
$I_i\ddot{q}_i + k_{ci}sign(\dot{q}_i) + k_{vi}\dot{q}_i + k_{si}sign(\dot{q}_i)e^{-(\dot{q}_i/\eta_{si})} + b_i = \tau_i^*$ (51)	$I_i\ddot{q}_i + k_{ci}sign(\dot{q}_i) + k_{vi}\dot{q}_i + k_{si}sign(\dot{q}_i)e^{-(\dot{q}_i/\eta_{si})} + b_i = \tau_i^*$ (51)
• Controller	• Controller
$\tau_{ri}^* = \hat{w}_{J_i}^T z_{I_i}\ddot{q}_{ri} + \hat{w}_{eqi}^T z_{eqi} + \hat{w}_{vi}^T z_{vi}\dot{q}_{ri} + k_i s_i$ (58)	$\tau_{ri}^* = Y_{ji}\hat{\theta}_{ji} + k_{ji}(\dot{q}_{ri} - \dot{q}_i)$ (78)
• Adaptive laws	• Adaptive law
$\hat{\dot{w}}_{J_i} = Q_{J_i}^{-1} z_{I_i}\dot{v}_{ki}s_i$ $\hat{\dot{w}}_{eqi} = Q_{eqi}^{-1} z_{eqi}s_i$ $\hat{\dot{w}}_{vi} = Q_{vi}^{-1} z_{vi}v_{ri}s_i$ (60)	$\dot{\hat{\theta}}_i = \Gamma_{\mathbf{ji}}^{-1} Y_{ji}^T(\dot{q}_{ri} - \dot{q}_i)$ (77)
• The computational complexity	• The computational complexity
(a) Multiplication: 6β	(a) Multiplication: $2y$
(b) Summation: $3\beta - 3$	(b) Summation: $y - 1$

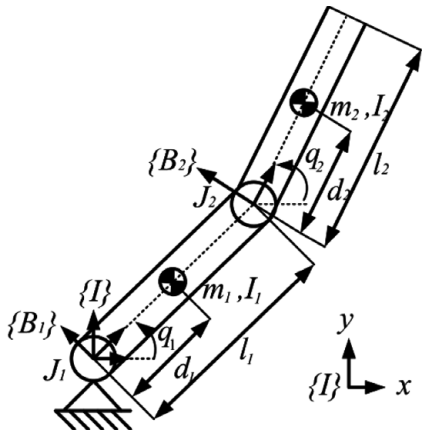


Fig. 3. The 2-R manipulator.

5. Simulation Results and Discussions

5.1. Case study 1. The 2-R planar manipulator

A planar 2-R manipulator is considered first to apply our approach as shown in Fig. 3. It is assumed that the two joints are actuated with high gear transmission. Tables V and VI show the physical parameters of the simulation example and the gains used for each subsystem. The equation of motion of each link and joint subsystem is described in Eqs. (2) and (51), respectively. As aforementioned,

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

Table V. Physical parameters for the planar 2-R robot.

The subsystem	Physical parameters
Links	$l_1 = l_2 = 0.75$ m, $l_{c1} = l_{c2} = 0.375$ m, $I_1 = I_2 = 0.0234$ kg m^2
	$J_1 = 0.04$ kg m^2, $J_2 = 0.02$ kg m^2
Joints	$k_{c1} = k_{c2} = k_{s1} = k_{s1} = 0.5$ Nm, $b_1 = b_2 = 0$
	$k_{v1} = k_{v2} = 0.5$ Nm s/rad, $\eta_{s1} = \eta_{s2} = 0.1$ rad/s

Table VI. Gain values for links and joints using FAT-VDC.

Subsystem		K_i	λ_i	Q_M^{-1}	Q_C^{-1}	Q_g^{-1}
Link 1		$\begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix}$	60	$I_{99\times 99}$	$I_{99\times 99}$	$150\ I_{99\times 99}$
Link 2		$\begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix}$	60	$I_{99\times 99}$	$I_{99\times 99}$	$150\ I_{99\times 99}$
		k_{ji}	λ_i	$Q_{J_i}^{-1}$	Q_{eqi}^{-1}	Q_{vi}^{-1}
Joint 1	Case 1	70	60	$100 I_{11\times 11}$	$150 I_{11\times 11}$	$100\ I_{11\times 11}$
	Case 2	70	60	$I_{11\times 11}$	$30 I_{11\times 11}$	$I_{11\times 11}$
Joint 2	Case 1	70	60	$100\ I_{11\times 11}$	$150 I_{11\times 11}$	$100 I_{11\times 11}$
	Case 2	70	60	$I_{11\times 11}$	$30 I_{11\times 11}$	$I_{11\times 11}$

it is necessary to derive the equation of motion of the link (i) ($i = 1, 2$) subsystem with respect to a local frame $\{B_i\}$ aligned with the link (i) as shown in Fig. 3. The mass, Coriolis and centripetal, and the gravity matrices of the link (i) can be described by the following equations:

$$M_{B_i} = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i d_i & m_i d_i \\ 0 & m_i d_i & I_i + m_i d_i^2 \end{bmatrix}, \quad i = 1, 2 \tag{79}$$

$$C_{B_i} = \begin{bmatrix} 0 & -m_i & -m_i d_i \\ m_i & 0 & 0 \\ m_i d_i & 0 & 0 \end{bmatrix} \dot{q}_i, \quad i = 1, 2 \tag{80}$$

$$g_{B_i} = \begin{bmatrix} m_i \sin(q_i)\, g \\ m_i \cos(q_i)\, g \\ m_i d_i \cos(q_i)\, g \end{bmatrix}, \quad i = 1, 2 \tag{81}$$

One of the important features of the VDC is the required augmented linear/angular velocity vector, which can be obtained from Eq. (6) or from the following equation:

$$^{B_i}v_r = \begin{pmatrix} ^{B_i}R_I \begin{pmatrix} \dot{x}_{B_i r} \\ \dot{y}_{B_i r} \end{pmatrix} \\ \dot{q}_i \end{pmatrix}, \quad i = 1, 2 \tag{82}$$

You should note that q_i represents the absolute angular displacement of the link (i). The details of the notations are given in Fig. 3.

The net force/moment vector at each local frame $\{B_i\}$ can be calculated as

$$\begin{aligned} ^{B_2}f_r^* &= {}^{B_2}f_r, \\ ^{B_1}f_r &= {}^{B_1}f_r^* + {}^{B_1}U_{B_2}\, {}^{B_2}f_r, \end{aligned} \tag{83}$$

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

where ${}^{B_i} f_r^*$ denotes the net force/moment vector applied on the link (i) with respect to the frame $\{B_i\}$, while ${}^{B_2} f_r$ represents the force/moment vector that is applied on the frame $\{B_i\}$, from which the required joint torques can be obtained as described in Eq. (52).

Thus, we have understood some important details of the dynamics of each subsystem. Next, we will describe the adaptive FAT of each subsystem, which is important in tracking the desired trajectory with unknown parameters of the dynamic subsystem. The desired target trajectory for our target robot could be selected as follows:[14]

$$q_{d1} = (30^0 * pi/180^0)(1 - \cos(2\pi t)),$$
$$q_{d2} = (45^0 * pi/180^0)(1 - \cos(2\pi t)). \tag{84}$$

Using the VDC strategy described in Fig. 2, the two link and the two joint subsystems are controlled individually. Ten terms of basis functions ($\beta = 10$) are used for all subsystems during simulation. Increasing the number of the basis function terms can result in better accuracy at the expense of large computation time. Although the small number of basis function terms can result in bad estimation, the subsystem could still be stable.[15] We assume that we did not have information about the initial conditions of the updating weighting matrices of Eqs. (45) and (60). Therefore, the initial conditions of the mentioned matrices are set to null matrices. We tried to use arbitrary initial conditions, but there was no noticeable change in the results. The adaptation gain matrices of Eqs. (45) and (60) play an important role in speeding up the adaptation process and in the improvement of the position error. We manipulated the adaptation gain matrices of the each joint subsystem only due to its easy manipulation. Thus, two cases of using different control parameters are considered for each joint as shown in Table VI. Higher values of adaptation gains could lead to less position error and better estimated parameters as shown in Figs. 4 and 5. The results of the joint subsystem are shown only due to the limited space on the pages. Although there are uncertain parameters, the system achieves precise tracking with bounded parameters.

The FAT-based adaptive VDC is compared with the conventional FAT-adaptive control–classical control that deals with the entire robotic systems as shown in Figs. 6 and 7.

Remark 5. A comparison is made to investigate the computational complexity of the FAT-based adaptive VDC with the regressor-based adaptive VDC used in refs.[20, 30]. It is found that the FAT approach for link (i) subsystems needed 3 s while the regressor-based approach needed approximately 0.5 s, as expected in Section 4.4 and Table IV. Note that ($\beta = 10$, $x = 3$, $y = 4$). A computer with a 2.1-GHz processor was used for simulation purposes.

5.2. Case study 2. Six-DOF planar biped robot

In this section, a 6-DOF biped robot is controlled using the FAT-based adaptive VDC. The biped robots have gained much attention for decades. It is aimed to make them able to assist or even substitute human beings in performing special tasks. In addition, studying biped robots is important in order to understand the human locomotion and to develop and improve control strategies for prosthetic and orthotic limbs. However, some challenges are encountered in the design and control of biped robots due to their high DOFs, having varying configuration during gait cycle and the unilateral foot-ground contact. Therefore, most researchers have attempted to use simple models to control their robots. Different control strategies have been used such as local controller,[21,51] optimal control,[52,53] and the robust sliding mode control[54,55] with simplified models. Therefore, VDC can be considered a powerful tool to deal with such a complex system. Zhu[20] has applied the regressor-based adaptive control to control the 5-DOF biped robot considering the actuator dynamics. In this section, a 6-DOF biped robot is simulated with FAT-based VDC. Since our study focuses on an open-chain robot, the control of the biped robot during the single support phase is developed with the following assumptions:

1. It can move in sagittal plane only with frictionless actuated direct drive joints.
2. Walking pattern generators of the biped robot were obtained from our previous work.[56] In effect, the physical parameters were taken from the same reference.
3. The support foot is in full contact during the single support phase. Therefore, its dynamics is not considered.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

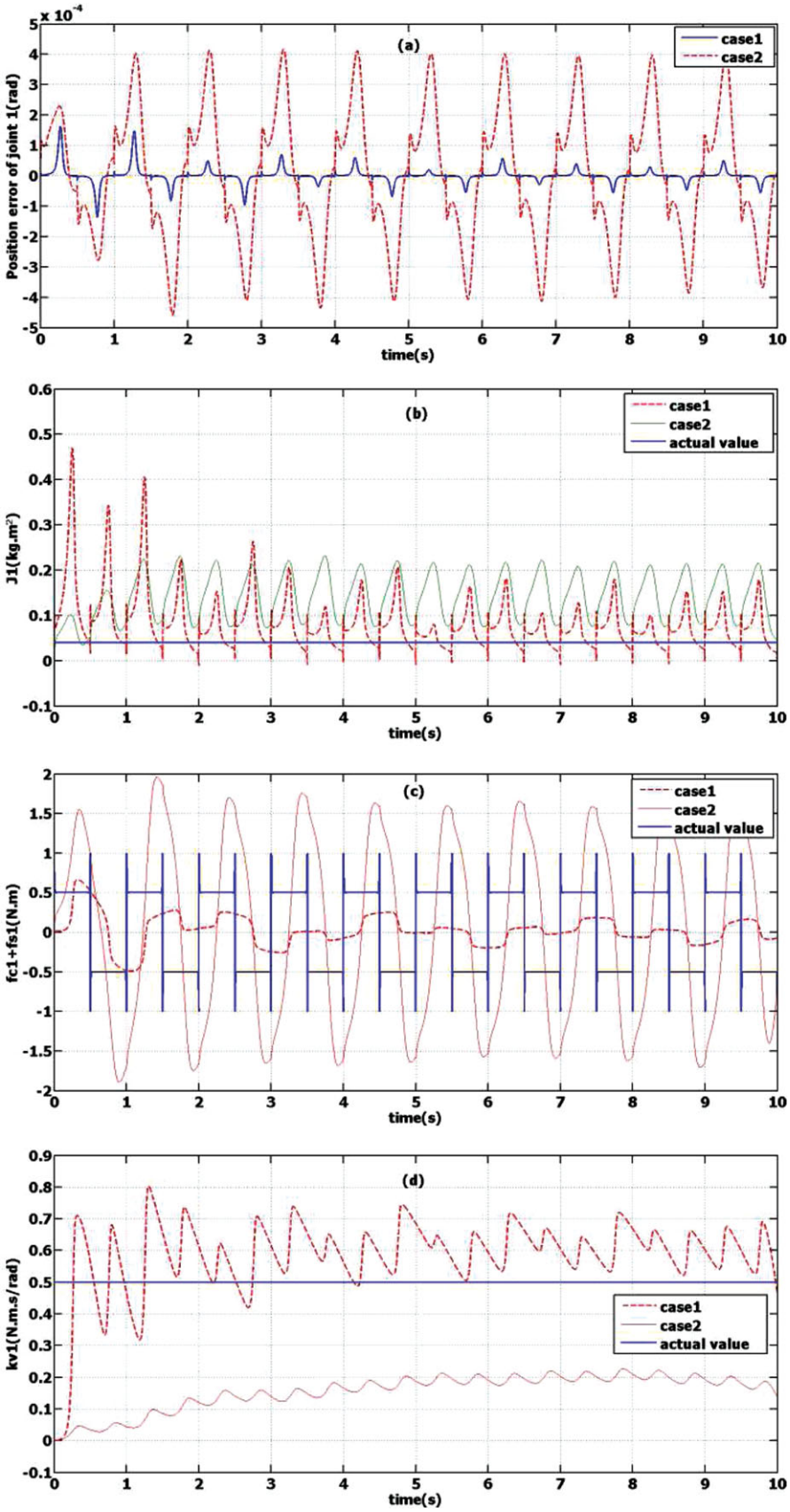


Fig. 4. (Colour online) FAT-based VDC results for joint 1: (a) position error, (b) estimated equivalent inertia, (c) estimated equivalent term of Coulomb and Stribeck friction torques and (d) estimated viscous friction coefficient.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

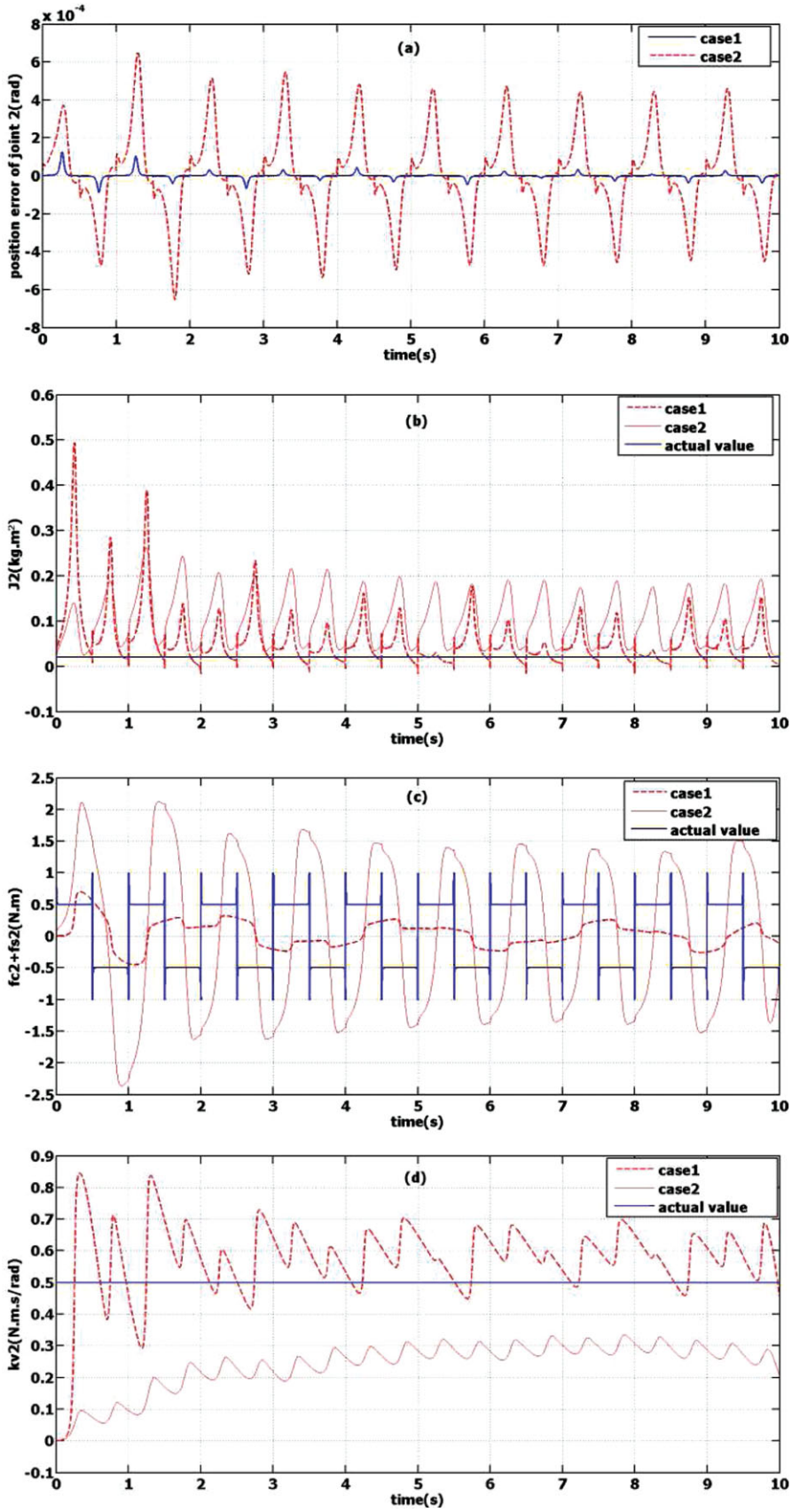


Fig. 5. (Colour online) FAT-based VDC results for joint 2: (a) position error, (b) estimated equivalent inertia, (c) estimated equivalent term of Coulomb and Stribeck friction torques and (d) estimated viscous friction coefficient.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

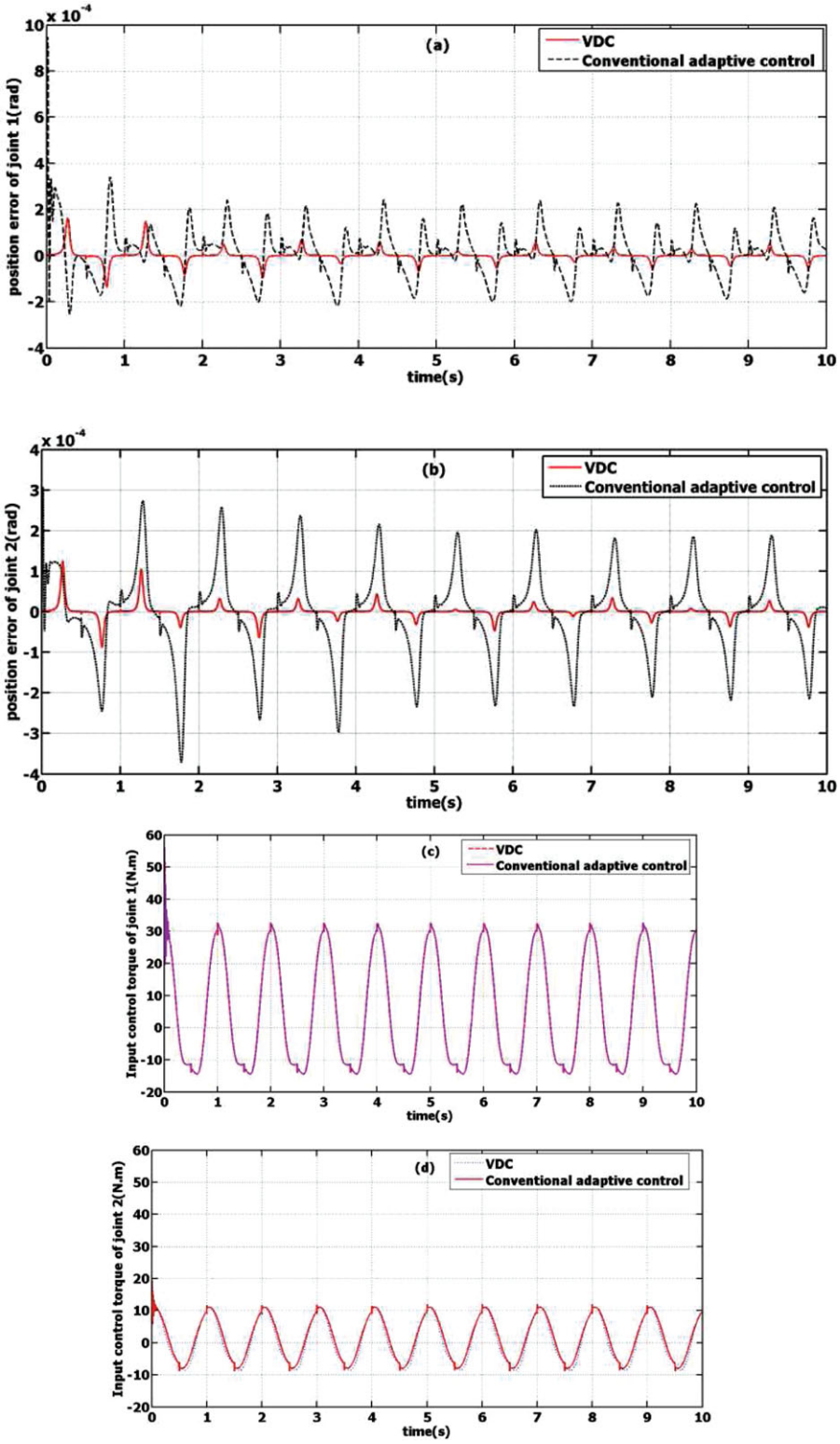


Fig. 6. (Colour online) Comparison between FAT-based adaptive VDC and conventional FAT-based adaptive control: (a) position error of joint 1, (b) position error of joint 2, (c) input control torque of joint 1 and (d) input control torque of joint 2.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

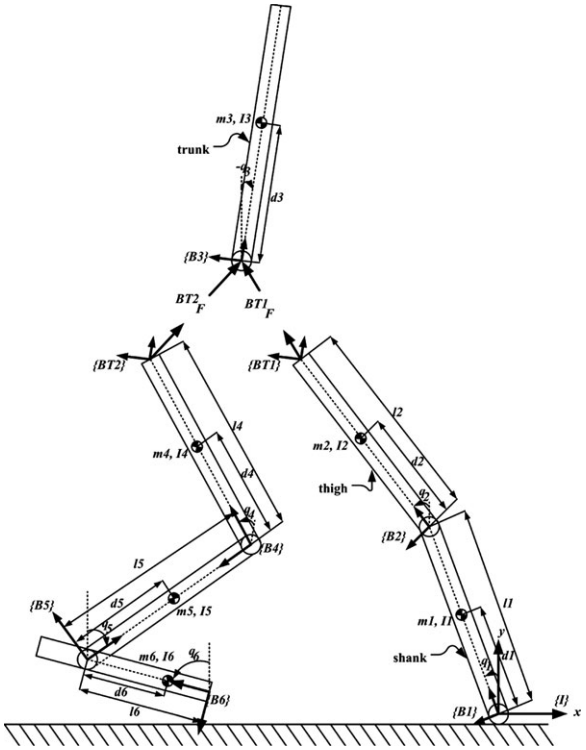


Fig. 7. The 6-DOF biped robot.

The dynamic equation of the link (i) subsystem can be described according to Eq. (2), with dynamic matrices exactly the same as Eqs. (79)–(81) with replacing q_i in the mentioned equations with $q_i + pi/2$. See Fig. 7 for the structure of the biped mechanism and the notations used. Equation (82) can provide us the required augmented velocity, which is important in the adaptive control algorithm. The force/moment vector applied on each link at each local frame $\{B_i\}$ can be calculated as

$$
\begin{aligned}
{}^{B_1}f &= {}^{B_1}f_r^* + {}^{B_1}U_{B_2}\,{}^{B_2}f_r,\\
{}^{B_2}f_r &= {}^{B_2}f_r^* + {}^{B_1}U_{B_2}\,{}^{B_{T_1}}f_r,\\
{}^{B_3}f_r &= {}^{B_{T_1}}f_r + {}^{B_{T_2}}f_r = {}^{B_3}f_r^*,\\
{}^{B_4}f_r &= {}^{B_4}f_r^* + {}^{B_1}U_{B_2}\,{}^{B_{T_2}}f_r,\\
{}^{B_5}f_r &= {}^{B_5}f_r^* + {}^{B_1}U_{B_2}\,{}^{B_4}f_r,\\
{}^{B_6}f_r &= {}^{B_6}f_r^* + {}^{B_1}U_{B_2}\,{}^{B_5}f_r.
\end{aligned}
\tag{85}
$$

The following points should be noted:

1. The required net force/moment vector ${}^{B_i}f_r^*$ can be obtained from the simulated link (i) subsystem as shown in Fig. 2.
2. The net force/force moment vector of the swing foot ${}^{B_6}f_r$ is equal to zero.
3. Because the dynamics of each joint is neglected, the required joint torques can be obtained from Eq. (52) with $\tau_i^* = 0, i = 1, 2, \ldots, 6$.

Using the VDC strategy described by Fig. 2, every link subsystem of the biped robot is controlled individually. It is suitable to start our strategy from the support leg (link 1) and ending with swing foot (link 6). It should be noted that the force/moment vector at the end of the swing foot (link 6) is zero, which is exploited to find the distribution of the force/moment vector at the hip/legs interaction. The same procedure and gains of the 2-R manipulator are used in this example. Figure 8 shows the

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

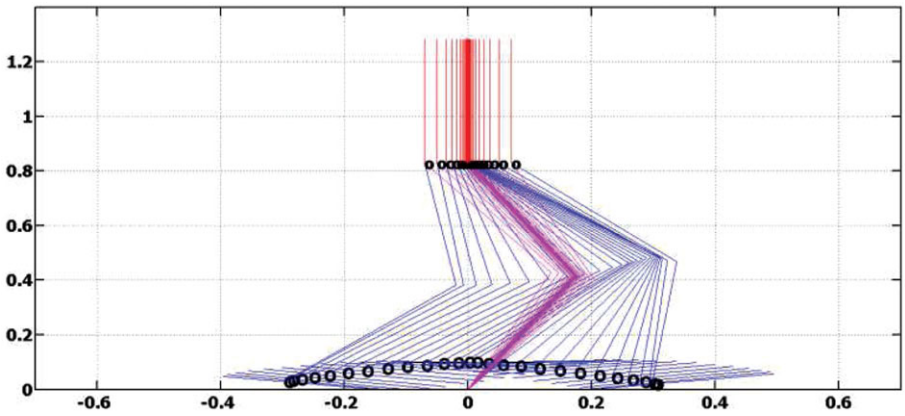


Fig. 8. (Colour online) Stick diagram of the 6-DOF biped robot.

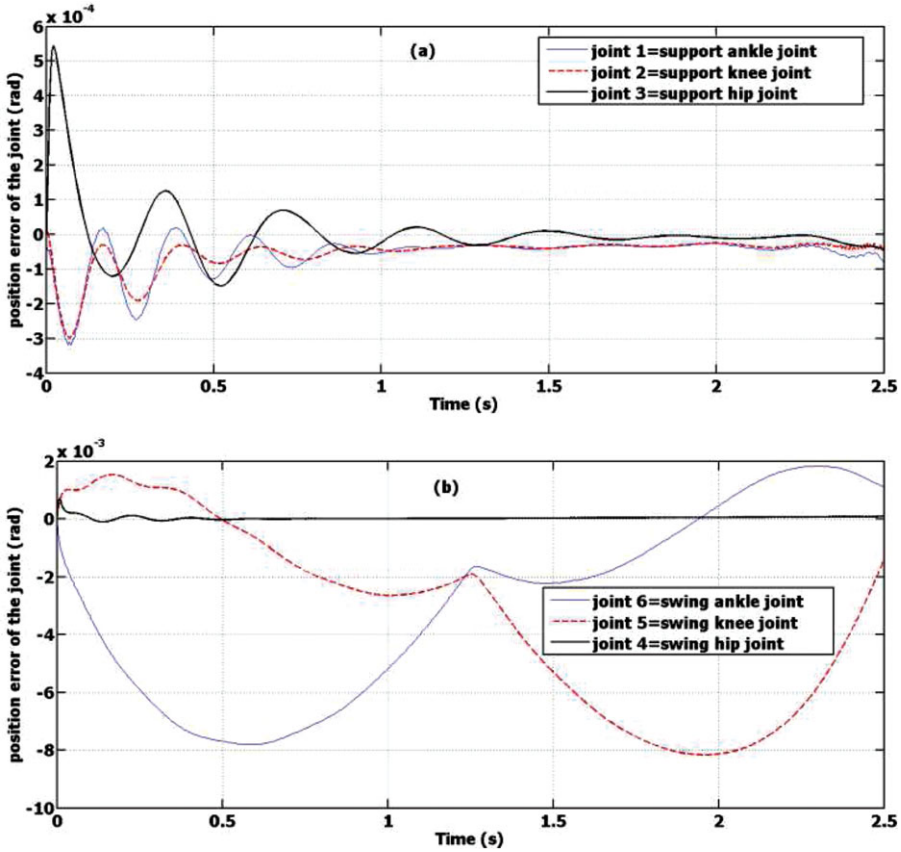


Fig. 9. (Colour online) Position errors of the biped robot: (a) support leg and (b) swing leg.

schematic diagram of the biped mechanism. There is a dark region at the middle of the trunk motion because the center of gravity of the biped robot will slow down at this region. In effect, two subphases can be encountered during the single support phase: initial deceleration and final acceleration. For details, we refer to ref. [56]. The position error of the six joints is shown in Fig. 9 for the support and the swing legs, respectively. The input control torques of the joints are illustrated in Fig.10 for the support and the swing legs, respectively.

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

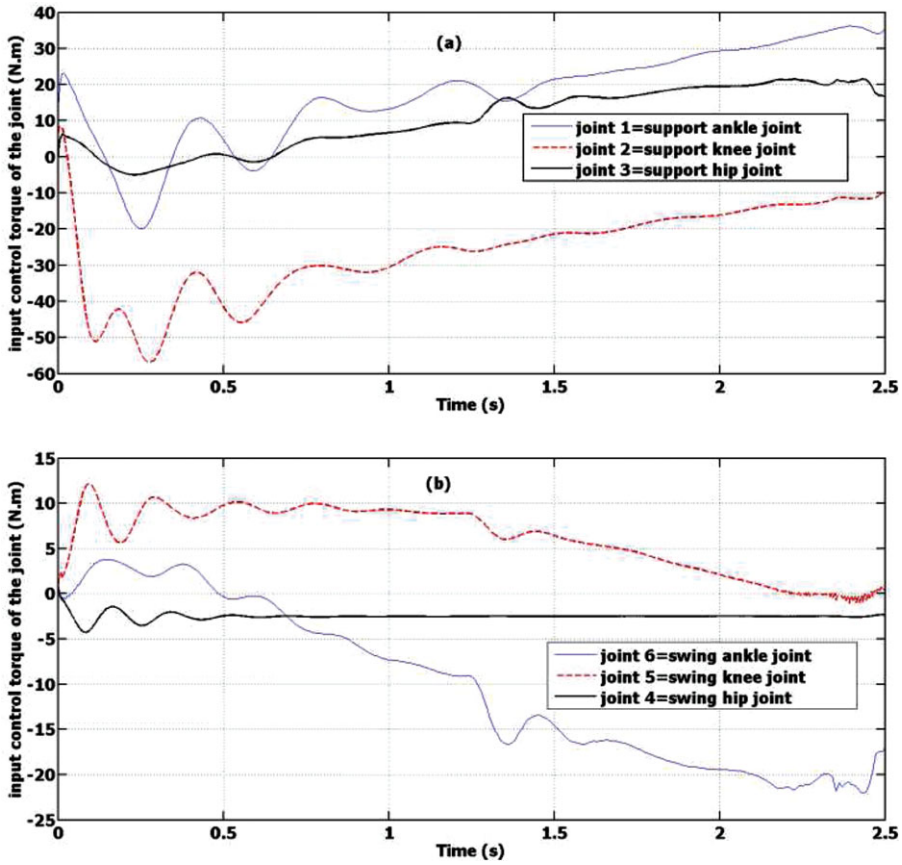


Fig. 10. (Colour online) Control torques of the biped robot: (a) support leg and (b) swing leg.

6. Conclusions

VDC is an efficient and powerful tool for the control of a complex robotic system. If the dynamic robotic system encounters uncertainty in its parameters, adaptive control should be used. Most adaptive schemes use the regressor matrix, which consists of nonlinear functions of the joint positions and their derivatives. Larger uncertain parameters lead to complex computations of the regressor approach, such as the underwater robots. Therefore, the FAT-based adaptive VDC is explored in this paper. Although the apparent advantages of the FAT such that the adaptation law will evade the regressor matrix, the computational complexities of the link subsystem seem insufficient in the implementation. The large computations result in the unfortunate representation of the dynamic matrices of the link subsystem with two sparse matrices of dimensions of $36\beta \times 6$. For the alleviation of computational complexities, the lattice-based approximation and CMAC are alternative tools that could be used for this purpose. However, The FAT-based adaptive VDC can be used successfully for the control of the joint due to its dynamic scalar equation. Frankly speaking, FAT is a fascinating and strong method for adaptive control if the appropriate representation of the approximation of the dynamic matrices can be selected. As a rule of thumb, using the regressor approach in the adaptive control of the link may be more efficient than the FAT. However, using FAT in the adaptive control of the joint can give very good results due to its effectively dealing with the friction effects. There is a tradeoff between the two methods.

References

1. J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "On-line computational scheme for mechanical manipulator," *Trans. ASME J. Dyn. Syst. Meas. Control* **102**(2), 69–76 (1980).
2. J. M. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and comparative study of dynamics formulation complexity," *IEEE. Trans. Syst. Man Cybern.* **SMC-10** (11), 730–736 (1980).

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

3. M. W. Walker and D. E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *Trans. ASME J. Dyn. Syst. Meas. Control* **104**(3), 205–211 (1982).
4. C. A. Balafoutis and R. V. Patel, "Efficient computation of manipulator, inertia matrices and the direct dynamics problem," *IEEE Trans. Syst. Man Cybern.* **19**(4), 1313–1321 (1989).
5. J. Angeles, O. Ma and A. Rojas, " An algorithm for the inverse dynamics of *n*-axis general manipulator using Kane's formulation of dynamical equations," *Comput. Math. Appl.* **17**(12), 1545–1561 (1998).
6. S. K. Saha, "Dynamics of serial multibody systems using the decoupled natural orthogonal complement matrices," *ASME J. Appl. Mech.* **66**, 986–996 (1999).
7. R. Featherstone, "The calculation of robot dynamics using articulated-body inertias", *Int. J. Robot. Res.* **2**(1), 13–30 (1983).
8. H. Brandle, R. Johanni and M. Otter, "A Very Efficient Algorithm for the Simulation of Robots and Similar Multibody Systems without Inversion of the Mass Matrix," **In**: *Proceedings of IFAC/IFIP/IMACS International Symposium on Theory of Robots*, Vienna, Austria (Dec. 1986), pp. 95–100.
9. A. Mohan and S. K. Saha, "A recursive, numerically stable and efficient simulation algorithm for serial robots," *Int. J. Multibody Syst. Dyn.* **17**(4), 291–319 (2007).
10. K. Lee and G. S. Chirikjian, "A New Perspective on O(n) Mass-Matrix Inversion for Serial Revolute Manipulators," **In**: *Proceeding of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 2005), pp. 4722–4726 .
11. R. Featherstone and D. Orin, "Robot Dynamics: Equations and Algorithms," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'00*, San Francisco, CA, USA (Apr. 2000), vol. 1, pp. 826–834.
12. S. K. Saha, "Recursive Dynamics Algorithms for Serial, Parallel and Closed-Chain Multibody Systems," *Indo-US Workshop on Protein Kinematics and Protein Conformations*, *IISC*, Bangalore (Dec. 2007).
13. K. S. Fu, R. C. Gonzalez and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence* (McGraw-Hill, New York, USA, 1987).
14. N. R. Sandell, J. R. P. Varaiya, M. Athans and M. G. Safonov, "Survey of decentralized control methods for large scale systems", *IEEE Trans. Autom. Control* **AC-23**(2), 108–128 (1978).
15. M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control* (John Wiley & Sons, New York, USA, 1989).
16. J. Boha, K. Belda and M. Valasue, "Decentralized control of redundant parallel robot construction," **In**: *Proceeding of the 10th Mediterranean Conference on Control and Automation*, Libson, Portugal (Jul. 2002), pp. 9–12.
17. J. Ohri, L. Dewan and M. K. Soni, "Tracking Control of Robots Using Decentralized Robust PID Control for Friction and Uncertainty Compensation," **In**: *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, CA, USA (Oct. 2007).
18. K. Xu, Integrating Centralized and Decentralized Approaches for Multi-Robot Coordination *Ph.D. Thesis* (Mechanical and Aerospace Engineering, New Brunswick, New Jersey, USA, 2010).
19. G. Leena and G. Ray, "A set of decentralized PID controller for an *n*-link robot manipulator," *Indian Acad. Sci.* **37**, 405–423 (2012).
20. W.-H. Zhu, *Virtual Decomposition Control: Toward Hyper Degrees of Freedom Robots* (Springer-Verlag, Berlin, Heidelberg, 2010).
21. H. Dallali, G. A. Medrano-Corda and M. Brown, "A Comparison of Multivariable and Decentralized Control Strategies for Robust Humanoid Walking," **In:** *Proceedings of UKACC International Conference on Control*, Coventry, UK (Sept. 2010).
22. W.-H. Zhu, Y. G. Xi and Z. J. Zhang, "Coordinative Control of Two Space Robots Based on Virtual Decomposition," **In**: *Proceedings of the 3rd IEEE Conference on Control Application*, Glasgow, Scotland (1994) pp. 327–332.
23. W.-H. Zhu, Y. G. Xi, Z. J. Zhang, Z. Bien and J. De Schutter, "Virtual decomposition based control for generalized high dimensional robotic systems with complicated structure," *IEEE Trans. Robot. Autom.* **13**(3), 411–436 (1997).
24. W.-H. Zhu, Z. Bien and J. De Schutter, "Adaptive motion/force control of coordinated multiple manipulators with joint flexibility based on virtual decomposition," *IEEE Trans. Autom. Control* **43**(1), 46–60 (1998).
25. W.-H. Zhu and J. De Schutter, "Adaptive control of mixed rigid/flexible joint robot manipulators based on virtual decomposition," *IEEE Trans. Robot. Autom.* **15**(2), 310–317 (1999).
26. W. H. Zhu and J. De Schutter, "Adaptive control of electrically driven space robots based on virtual decomposition," *AIAA J. Guid. Control Dyn.* **22**(2), 329–339 (1999).
27. W.-H. Zhu and J. De Schutter, "Experimental verifications of virtual decomposition based motion/force control," *IEEE Trans. Robot. Autom.* **18**(3), 379–386 (2002).
28. W.-H. Zhu and J. De Schutter, "Virtual Decomposition Based Motion/Force Control of an Industrial Manipulator KUKA361," **In**: *Preprint of 15th IFAC World Congress*, Barcelona, Spain (2002), pp. 1902–1907.
29. W.-H. Zhu, C. Lange and M. Callot, "Virtual Decomposition Control of a Planar Flexible-Link Robot," **In**: *Preprint of 17th IFAC World Congress*, Seoul, Korea (2008) pp. 1697–1702.
30. G. Antonelli, F. Caccavale and S. Chiaverini, "A Virtual-Decomposition Based Approach to Adaptive Control of Underwater Vehicle-Manipulator Systems," **In**: *The 9th Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia (2001).

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press

31. Y. D. Song, "Adaptive Motion Tracking Control of Robot Manipulators-Non-Regressor Based Approach," **In**: *Proceeding of the IEEE International Conference on Robotics and Automation*, San Diego, CA, USA (May 1994), vol. 4, pp. 3008–3013.
32. N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controller for robotic manipulators," *Int. J. Robot. Res.* **9**(3), 74–92 (1990).
33. W. S. Lu and Q. H. Meng, "Recursive Computation of Manipulator Regressor and Its Application to Adaptive Motion Control of Robots," **In**: *Proceedings of the IEEE Conference on Communication, Computation and Signal Processing*, Victoria, BC, USA (May 1991) pp. 170–173.
34. H. Kawasaki, T. Bito and K. Kanzaki, "An efficient algorithm for model-based adaptive controller of robot manipulators," *IEEE Trans. Robot. Autom.* **12**(3), 496–501 (1996).
35. J. H. Yang, "Adaptive Tracking Control for Manipulators with Only Position Feedback," **In**: *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, Alberta, Canada (May 1999) vol. 3, pp. 1740–1745.
36. Z. Qu and J. Dorsey, "Robust tracking control of robots by a linear feedback law," *IEEE Trans. Autom. Control* **36**(9), 1081–1084 (1991).
37. J. S. Park, Y. A. Jiang, T. Hesketh and D. J. Clements, "Trajectory Control of Manipulators Using Adaptive Sliding Mode Control," **In**: *Proceedings of the IEEE, Southeastcon'94*, Miami, FL, USA (Apr. 1994) pp. 142–146.
38. J. Yuan and Y. Stepanenko, "Adaptive PD Control of Flexible Joint Robots without Using the High-Order Regressor," **In**: *Proceedings of the 36th Midwest Symposium on Circuits and Systems*, Detroit, MI, USA (Aug. 1993), vol. 1, pp. 389–393.
39. C. Y. Su and Y. Stepanenko, "Adaptive Control for Constrained Robots without Using Regressor," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, USA (Apr. 1996), vol. 1, pp. 264–269.
40. A.-C. Huang and M.-C. Chien, *Adaptive Control of Robot Manipulators: A Unified Regressor-Free Approach* (World Scientific, Singapore/USA, 2010).
41. S. Cong, Y. Liang and W. Shang, "Function approximation-based sliding mode adaptive control for time-varying uncertain nonlinear systems," *Nonlinear Dyn.* **54**(5), 223–230 (2008).
42. J.-W. Liang, H.-Y. Chen and Y.-T. Tsu, "FAT-based Adaptive Sliding Mode Control for a Piezoelectric-Actuated System," **In**: *Proceedings of the IEEE International Conference on Control and Automation*, Christchurch, New Zealand (Dec. 2009) pp. 848–853.
43. T. Fossen, *Guidance and Control of Ocean Vehicles* (John Wiley & Sons, Chicheste, UK, 1994).
44. J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. (Pearson, New Jersey, USA, 2005).
45. J.-J. E. Slotine and W. Li, *Applied Nonlinear Control* (Prentice-Hall, New Jersey, USA, 1991).
46. A. Farrell and M. M. Polycarpou, *Adaptive Approximation Based: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches* (John Wiley & Sons, New York, USA, 2006).
47. J. D. Faires and R. L. Burden, *Numerical Methods*, 3rd ed. (Brooks Cole, Pacific Grove, CA, USA, 2002).
48. H. Chaoui, P. Sicard and W. Gueaieb, "ANN-based adaptive control of robotic manipulators with friction and joint elasticity," *IEEE Trans. Ind. Electron.* **56**(8), 3174–3187 (2009).
49. C. Chen, M. H. Cheng, C. Yang and J. Chen, "Robust Adaptive Control for Robot Manipulators with Friction," **In**: *Proceedings of the IEEE 3rd International Conference on Innovative Computing Information and Control (ICICIC'08)*, Dalian, Liaoning, China (Jun. 2008) p. 422.
50. R. Hunger, Floating Point Operations in Matrix–Vector Calculus, *Technical Report* (Associate Institute for Signal Processing, Technische Universitäte Mümchen, 2007).
51. S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi, "Biped Walking Stabilization Based on Inverted Pendulum Tracking," **In**: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Taiwan (2010) pp. 4489–4496 .
52. L. Roussel, C. Canudas-de-Wit and A. Goswami, "Generation of Energy Optimal Complete Gait Cycles for Biped Robots," **In**: *Proceedings of the IEEE Conference on Robotics and Automations*, Leuven, Belgium (May 1998) vol. 3, pp. 2036–2041.
53. C. Azevedo, P. Poignet and B. Esspiau, "On Line Optimal Control for Biped Robots," **In**: *IFAC, 15th Triennial World Congress*, Barcelona, Spain (2002).
54. S. Tzafestas, M. Raibert and C. Tzafestas, "Robust sliding–mode control applied to a 5-link biped robot," *J. Intell. Robot. Syst.* **15**, 76–133 (1996).
55. X. Mu and Q. Wu, "Dynamic Modeling and Sliding Mode Control of a Five-Link Biped Robot during the Double Support Phase," **In**: *Proceeding of the 2004 American Control Conference*, Boston, MA, USA (2004) pp. 2609–2614.
56. H. F. N. Al-Shuka and B. Corves, "On the walking pattern generators of biped robot," *J. Autom. Control* **1**(2), 149–155 (2013).

https://doi.org/10.1017/S0263574713000775 Published online by Cambridge University Press