Sliding mode-based online fault compensation control for modular reconfigurable robots through adaptive dynamic programming

Hongbing Xia^{1,2} · Ping Guo¹

Received: 30 November 2020 / Accepted: 29 March 2021 / Published online: 26 April 2021
© The Author(s) 2021

Abstract

In this paper, a sliding mode (SM)-based online fault compensation control scheme is investigated for modular reconfigurable robots (MRRs) with actuator failures via adaptive dynamic programming. It consists of a SM-based iterative controller, an adaptive robust term and an online fault compensator. For fault-free MRR systems, the SM surface-based Hamilton–Jacobi–Bellman equation is solved by online policy iteration algorithm. The adaptive robust term is added to guarantee the reachable condition of SM surface. For faulty MRR systems, the actuator failure is compensated online to avoid the fault detection and isolation mechanism. The closed-loop MRR system is guaranteed to be asymptotically stable under the developed fault compensation control scheme. Simulation results verify the effectiveness of the present fault compensation control approach.

Keywords Adaptive dynamic programming · Sliding mode control · Fault compensation · Neural networks · Modular reconfigurable robots

Introduction

As modular reconfigurable robots (MRRs) take advantages of structural flexibility, low cost, excellent adaptability, etc., they often work in perilous and complex working environments, such as disaster rescue, deep space/sea exploration, smart manufacturing, and many other hazardous environments that human cannot involve directly [1–3].

The research work described in this paper was supported partially by the National Key Research and Development Program of China under Grant 2018AAA0100203, partially by the National Natural Science Foundation of China under Grants 61973330 and 61773075, partially by the Beijing Municipal Natural Science Foundation under Grant 4212038, partially by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China under Grant ICT2021B48, partially by the Fundamental Research Funds for the Central Universities under Grant 2019NTST25, and partially by the Key Scientific Research Project of Bengbu University under Grant 2019ZR01zd.

✉ Ping Guo
pguo@bnu.edu.cn

Hongbing Xia
xiahongbing@mail.bnu.edu.cn

¹ School of Systems Science, Beijing Normal University, Beijing 100875, China

² School of Electronic and Electrical Engineering, Bengbu University, Bengbu 233030, China

In recent years, several researches on MRRs control approaches have been attracted a great deal of attention, such as centralized control [4,5], distributed control [6,7], and decentralized control [8,9]. These control approaches mainly tackled force/position control problems [10,11], fault tolerant control problems [12,13], and so on. Despite above methods have achieved good performance, the designed controller always contain some adjustable parameters, which increase the design difficulty and structural complexity. Thus, more attention should be paid to simplify the control structure and reduce the computational burden.

As we know, the optimal control is one of the key requirements in modern control theory. It can not only ensure the stability of the control system, but also obtain proper optimal performance. Since the strong self-learning and optimization ability, adaptive dynamic programming (ADP) [14] was introduced and extensively investigated in optimal control by solving the Hamilton–Jacobi–Bellman equation (HJBE) without the “curse of dimensionality”. Therefore, more and more ADP-based control methods were investigated to deal with trajectory tracking [15,16], zero-sum games [17], uncertainties [18], and actuator saturation [19], etc. It is significant to point out that reinforcement learning (RL) and ADP are almost in the same spirit when dealing with optimal control problems. Therefore, RL is often regarded as the synonym for ADP. Up to now, a lot of ADP and RL methods have

been investigated [20–23]. Bai et al. designed an optimal control approach via the neural network (NN) technique and RL algorithm to tackle the nonstrict-feedback control problem [24] and input saturation [25]. For systems with known dynamics, Shi et al. [26] proposed an optimal tracking control (OTC) approach to handle time delay problems via integral RL and value iteration method. In [27], a novel approximate OTC strategy was addressed using an event-driven ADP algorithm to handle such problems. However, the aforementioned methods require accurate system dynamics, which is difficult to obtain in real industrial applications of MRRs. Recently, some model-free RL-based control methods have been presented. These control approaches depend merely on the input and the output measurement data of the controlled plant [28]. Actually, the model-free methods require large online or offline data to train NN, which wastes computation and training time.

Furthermore, MRRs work in hazardous environments for a long time may lead to the occurrence of failures, which can not only degrade the system performance, but even damage the surrounding workspace. As we all know, the actuator failure is always regarded as one of challenging failures to handle, because the occurrence of unknown actuator failures may easily cause serious deterioration of the system control performance compared with other fault scenarios. Furthermore, it is unrealizable to repair MRRs in hazardous environments. Hence, exploring a fault tolerant control (FTC) method is imperative to guarantee MRRs to continue working reliable in the presence of actuator failures.

FTC strategies mainly include passive FTC (PFTC) and active FTC (AFT-C). PFTC does not need fault detection and identification (FDI) unit. Over the last few decades, many PFTC approaches have been presented mainly based on quantitative feedback theory [29], linear matrix inequality [30], and H_∞ theory [31]. PFTC designs a fixed controller before fault occurs, thus it can only solve the known faults [32]. Alternatively, AFTC can effectively avoid the drawback of PFTC. AFTC obtains fault information via FDI, then readjusts or reconstructs the control law. Various AFTC approaches can be categorized into fault accommodation [33], fault reconfiguration [34], and fault compensation [35]. Owing to the better performance of AFTC, it is potential in robot manipulators [36], quadrotors [37], inverted pendulums [38] and other practical applications. Moreover, some FTC schemes have been developed through RL or ADP. Zhao et al. [32] employed the information of the fault observer to construct an improved cost function and utilized online iteration algorithm to develop a novel FTC method for nonlinear systems. Fan and Yang [39] investigated an FTC strategy to handle the time-varying actuator bias faults via ADP. In [40], an ADP-based stabilizing scheme for nonlinear systems with unknown actuator saturation was developed via NN compensation. However, these literatures have solved stabilizing

problems, rather than trajectory tracking, which is feasible to MRRs.

To get rapid response and convergence, sliding mode-based control schemes have been presented. Owing to the low sensitivity and robustness to system uncertainties and external disturbances, sliding mode control (SMC) reduces the necessity of accurate model, and is feasible to apply to design control systems no matter in normal or faulty conditions [41–43]. Hence, SMC methods always have been applied to systems with high nonlinearities, variable parameters and external disturbances, such as aircraft systems [44], direct current (DC) servomotors [45], multi-machine power systems [46] and MRRs [12].

Although previous ADP-based FTC methods can guarantee the stability of faulty system, we further require a faster control action in practice. Thus, motivated by [47], this paper develops a SM-based online fault compensation control (SMOFCC) scheme for MRRs with unknown actuator failures. For the fault-free case, the SM-based approximate optimal control (SMAOC) is derived using the SM-based iterative controller and an adaptive robust term. When the actuator failures occurs, the SMOFCC is obtained by adding an online fault compensator to SMAOC. The main contribution and novelties of this work are presented as follows.

- (1) The scheme extends the ADP-based SMC method to FTC problem for MRRs with unknown actuator failures, and the online fault compensation is achieved without FDI.
- (2) The proposed SMOFCC scheme, which is composed of SM-based iterative controller, an adaptive robust term and a fault compensator, can guarantee the MRR system to be asymptotically stable, rather than ultimately uniformly bounded (UUB) [3,32,39,40].
- (3) By employing the SMC technique, the developed SMOFCC has a faster control response compared to that based on tracking error feedback only [3].

The rest of this paper is organised as follows. In the next section, we present the problem statement. In the subsequent section, the SM-based control scheme for MRRs in fault-free case is presented. Then, an online fault compensator is developed to obtain the FTC, and the stability analysis is provided. The numerical simulation demonstrates the effectiveness of the SMOFCC before the final sections. In the last section, a brief conclusion is drawn.

Problem statement

The n -DOF (degree of freedom) MRR system with unknown actuator failures can be described by

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u - f_a, \quad (1)$$

where $q \in \mathbb{R}^n$ denotes the vector of joint displacements, $M(q) \in \mathbb{R}^{n \times n}$ denotes the nonsingular symmetric inertia matrix, $C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ denotes the Coriolis and centripetal force, $G(q) \in \mathbb{R}^n$ denotes the gravity term, and $u \in \mathbb{R}^n$ denotes the joint input torque, $f_a \in \mathbb{R}^n$ represents the unknown additive actuator failure.

Define the system state as $x = [x_1, x_2]^T = [q, \dot{q}]^T$, the MRR system (1) can be presented as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)(u - f_a) \\ y = x_1, \end{cases} \quad (2)$$

where $x \in \mathbb{R}^{2n}$ and $y \in \mathbb{R}^n$ are the state and the output vectors, respectively, $f(x) = M^{-1}(q)[-C(q, \dot{q}) - G(q)]$ and $g(x) = M^{-1}(q)$.

Assumption 1 The nonlinear functions $f(\cdot)$ and $g(\cdot)$ are Lipschitz continuous with $f(0) = 0$, i.e., $x = 0$ is the equilibrium point of system (2), and system (2) is controllable.

Assumption 2 The desired reference trajectory q_d , the velocity vector $\dot{q}_d$, and the acceleration vector $\ddot{q}_d$ are norm-bounded as [15]

$$\begin{bmatrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{bmatrix} \leq q_\kappa,$$

where $q_\kappa > 0$ is a known constant.

Assumption 3 The actuator failure f_a is norm-bounded as $\|f_a\| \leq \varsigma_M$, where ς_M is a positive constant.

For the fault-free case of the MRR system (2), i.e., $f_a = 0$, we define the nominal system as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)u_0 \\ y = x_1, \end{cases} \quad (3)$$

where u_0 is a SMAOC law.

The tracking error is defined as

$$e = x - x_\vartheta, \quad (4)$$

where $x_\vartheta = [q_\vartheta, \dot{q}_\vartheta]^T$ is the desired reference trajectory. Thus, the time derivative of the tracking error (4) becomes

$$\dot{e} = \dot{x} - \dot{x}_\vartheta. \quad (5)$$

To accelerate the convergence rate, we introduce the SM surface as

$$s = \dot{e} + \Lambda e, \quad (6)$$

where Λ is a positive definite matrix.

The time derivative of (6) is

$$\begin{aligned} \dot{s} &= \ddot{e} + \Lambda \dot{e} \\ &= f(x) + g(x)u_0 - \ddot{x}_\vartheta + \Lambda \dot{e} \\ &= f(x) + g(x)u_0 + \varphi, \end{aligned} \quad (7)$$

where $\varphi = -\ddot{x}_\vartheta + \Lambda \dot{e}$.

To realize the approximate optimal control, the SM-based iterative controller u_s is used to make the trajectory tracking error converge to the steady state, then the cost function is defined as

$$J(s(t)) = \int_0^\infty Z(s(t), u_s(t))dt, \quad (8)$$

where $Z(s, u_s) = s^T Q s + u_s^T R u_s$, $J(s(t)) \geq 0$ for arbitrary s and u_s , and $J(0) = 0$. $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$ are positive definite matrices.

Remark 1 From (6), we can observe that the SM surface consists of the position tracking error e and the velocity tracking error $\dot{e}$, rather than position tracking error e only. Thus, compare the optimal control with the SM signal to that with position tracking error feedback only, the optimal control with SM has a faster convergence and smaller overshoot. Furthermore, the SMC has low sensitivity and strong robustness to system uncertainties, and is easily implemented in practice.

Online fault compensation control design and stability analysis

Sliding mode-based HJBE

Definition 1 Considering the nominal MRR system (3), a SM-based iterative control strategy $\mu(s) \in \Psi(\Omega)$ is defined to be admissible subject to (8) on a compact set Ω , if $\mu(s)$ is continuous on Ω with $\mu(0) = 0$, $\mu(s)$ ensures system (2) to be convergence on Ω , and $J(s)$ is finite, $\forall s \in \Omega$ [3,27,32].

For each admissible control strategy $\mu(s) \in \Psi(\Omega)$ of system (3), where $\Psi(\Omega)$ is the set of admissible control, if the cost function (8) is continuously differentiable, then the nonlinear Lyapunov equation can be derived as

$$0 = Z(s, u_s) + (\nabla J(s))^T \dot{s}, \quad (9)$$

where $\nabla J(s) = \frac{\partial J(s)}{\partial s}$.

The Hamiltonian is defined as

$$H(s, u_s, \nabla J(s)) = Z(s, u_s) + (\nabla J(s))^T \dot{s}, \quad (10)$$

and the optimal cost function can be defined as

$$J^*(s(t)) = \min_{u_s} \int_0^\infty Z(s(t), u_s(t)) dt. \quad (11)$$

Based on the Bellman principle of optimization, $J^*(s)$ satisfies the HJBE

$$0 = \min_{u_s} H(s, u_s, \nabla J^*(s)). \quad (12)$$

Since $\frac{\partial H(s, u_s^*, \nabla J^*(s))}{\partial u_s^*} = 0$, the optimal control law can be obtained as

$$u_s^*(s) = -\frac{1}{2} R^{-1} g^T(x) \nabla J^*(s). \quad (13)$$

By equivalent transformation, (13) becomes

$$\nabla J^{*\top}(s) g(x) = -2u_s^{*\top}(s) R. \quad (14)$$

Online PI algorithm

According to [36,37,39,40], the solution of HJBE (12) can be approximated through the online PI algorithm when the system is in normal state. Unlike [3], the online PI algorithm is realized with the help of SM feedback signal, rather than system tracking error. The online PI algorithm is presented in Algorithm 1.

Algorithm 1 Online PI algorithm

Step 1. Select a small positive constant ζ , the maximum iteration time T . Let $l = 0$,

and begin with a given admissible control policy $\mu^{(0)}(s)$,

Step 2. (Policy evaluation) Let $l > 0$, based on the control policy $\mu^{(l)}(s)$, solve the following nonlinear Lyapunov equation $J^{(l)}$ for

$$0 = Z(s, \mu^{(l)}(s)) + (\nabla J^{(l)})^\top \dot{s},$$

Step 3. (Policy improvement) Update the control policy $\mu^{(l)}$ via

$$\mu^{(l+1)}(s) = -\frac{1}{2} R^{-1} g^T(x) \nabla J^{(l)}(s).$$

Step 4. If $\|J^{(l+1)}(s) - J^{(l)}(s)\| \leq \zeta$, stop and obtain the approximate optimal control; else, let $l = l + 1$, if $l \leq T$, return to Step 2.

where $W_c \in \mathbb{R}^M$ and $\sigma_c(s)$ denote the ideal weight vector and the activation function, respectively, M denotes the number of neurons in the hidden layer, and $\varepsilon_c(s)$ denotes the approximation error caused by critic NN (CNN) approximation. Then, from (15), we can obtain

$$\nabla J(s) = \nabla \sigma_c^\top(s) W_c + \nabla \varepsilon_c^\top(s). \quad (16)$$

According (16), the Hamiltonian (10) can be rewritten as

$$\begin{aligned} H(s, u_s, W_c) &= Z(s, u_s) + (W_c^\top \nabla \sigma_c(s)) \dot{s} \\ &= -\nabla \varepsilon_c^\top \dot{s} \\ &= s_{cH}, \end{aligned} \quad (17)$$

where s_{cH} is the residual error caused by the NN approximation.

To estimate W_c , the CNN (15) is approximated as

$$\hat{J}(s) = \hat{W}_c^\top \sigma_c(s) \quad (18)$$

and we can obtain $\nabla \hat{J}(s)$ from (18) that

$$\nabla \hat{J}(s) = \nabla \sigma_c^\top(s) \hat{W}_c. \quad (19)$$

Inserting (19) into (17), we have the approximate Hamiltonian as

Sliding mode-based critic neural network

The cost function $J(s)$ can be reconstructed by a single-layer NN as

$$J(s) = W_c^\top \sigma_c(s) + \varepsilon_c(s), \quad (15)$$

$$H(s, u_s, \hat{W}_c) = Z(s, u_s) + \hat{W}_c^\top \nabla \sigma_c(s) \dot{s} = s_c. \quad (20)$$

Denote $\varpi = \nabla \sigma_c(s) \dot{s}$, and assume that there exists a constant $\varpi_M > 0$ such that $\|\varpi\| \leq \varpi_M$. By minimizing the objective

function $E_c = \frac{1}{2}s_c^\top s_c$ with gradient descent algorithm, $\hat{W}_c$ should be updated by

$$\dot{\hat{W}}_c = -\beta_c s_c \varpi, \quad (21)$$

where $\beta_c > 0$ is the learning rate.

Define the weight approximation error as

$$\tilde{W}_c = W_c - \hat{W}_c. \quad (22)$$

From (17), (20) and (22), one has

$$s_c = s_{cH} - \tilde{W}_c^\top \varpi. \quad (23)$$

Then, the weight approximation error is updated by

$$\dot{\tilde{W}}_c = -\dot{\hat{W}}_c = \beta_c s_c \frac{\partial s_c}{\partial \hat{W}_c} = \beta_c (s_{cH} - \tilde{W}_c^\top \varpi) \varpi. \quad (24)$$

Inserting (16) to (13), the ideal SM-based iterative control strategy is expressed by

$$u_s = -\frac{1}{2}R^{-1}g^\top(x)(\nabla\sigma_c^\top(s)W_c + \nabla\varepsilon_c^\top(s)) \quad (25)$$

and it is approximated as

$$\hat{u}_s = -\frac{1}{2}R^{-1}g^\top(x)\nabla\sigma_c^\top(s)\hat{W}_c. \quad (26)$$

Theorem 1 *Considering the nominal MRR system (3), if the weight vector of the SM-based CNN is tuned by (21), the weight approximation error is guaranteed to be UUB.*

Proof Choose a Lyapunov function candidate as

$$L_1 = \frac{1}{2\beta_c} \tilde{W}_c^\top \tilde{W}_c. \quad (27)$$

Taking the time derivative of (27), we have

$$\begin{aligned} \dot{L}_1 &= \frac{1}{\beta_c} \tilde{W}_c^\top \dot{\tilde{W}}_c \\ &= \tilde{W}_c^\top (s_{cH} - \tilde{W}_c^\top \varpi) \varpi \\ &= \tilde{W}_c^\top s_{cH} \varpi - \|\tilde{W}_c^\top \varpi\|^2 \\ &\leq \frac{1}{2}s_{cH}^2 - \frac{1}{2}\|\tilde{W}_c^\top \varpi\|^2. \end{aligned} \quad (28)$$

Therefore, we can obtain $\dot{L}_1 \leq 0$ as long as $\tilde{W}_c$ lies outside the compact set $\Omega_c = \{\tilde{W}_c : \|\tilde{W}_c\| < \frac{s_{cH}}{\varpi_M}\}$. Thus, the CNN weight estimation error $\tilde{W}_c$ is UUB. This completes the proof.

Sliding mode-based approximate optimal control

In light of previous analysis, we can design the SMAOC law as

$$u_0 = u_s - \frac{\hat{\varphi} + k_1 s + k_2 \text{sgn}(s)}{g(x)}, \quad (29)$$

where $\text{sgn}(s) = [\text{sgn}(s_1), \text{sgn}(s_2), \dots, \text{sgn}(s_n)] \in \mathbb{R}^n$, k_1 and k_2 are positive definite constant matrices, and $\hat{\varphi}$ is a robust term which is tuned by

$$\dot{\hat{\varphi}} = \beta_\varphi s, \quad (30)$$

where $\beta_\varphi > 0$ is a positive definite matrix. According to the SMC theory and the proposed controller (29), the reachable condition $s^\top \dot{s} \leq 0$ ensures the MRR system states reach and stay on the SM surface.

Theorem 2 *Consider the nominal MRR system (3), the SM surface (6) and its time derivative (7). The tracking error of the MRR system can arrive and stay on the SM surface thereafter under the developed SMAOC law (29).*

Proof Choose a Lyapunov function candidate as

$$L_2 = \frac{1}{2}s^\top s + \frac{1}{2\beta_\varphi} \tilde{\varphi}^\top \tilde{\varphi}, \quad (31)$$

where $\tilde{\varphi} = \varphi - \hat{\varphi}$. Introducing the SMAOC law (29) into (31), we have

$$\begin{aligned} \dot{L}_2 &= s^\top \dot{s} - \frac{1}{\beta_\varphi} \dot{\hat{\varphi}} \tilde{\varphi} \\ &= s^\top (f(x) + g(x)u_s + \varphi - \hat{\varphi} - k_1 s - k_2 \text{sgn}(s)) \\ &\quad - \frac{1}{\beta_\varphi} \dot{\hat{\varphi}} \tilde{\varphi}. \end{aligned} \quad (32)$$

From Assumption 1, there exist two unknown positive constants D_f and D_g , s.t. $\|f(x)\| \leq D_f$ and $\|g(x)\| \leq D_g$. Using Young's inequality, (32) becomes

$$\begin{aligned} \dot{L}_2 &= D_f \|s\| + s^\top g(x)u_s - \lambda_{\min}(k_1)\|s\|^2 \\ &\quad - \lambda_{\min}(k_2)\|s\| + \tilde{\varphi}s - \frac{1}{\beta_\varphi} \dot{\hat{\varphi}} \tilde{\varphi} \\ &\leq -\lambda_{\min}(k_1)\|s\|^2 - \lambda_{\min}(k_2)\|s\| \\ &\quad + D_f \|s\| + D_g \|s\| \|u_s\| \\ &\leq -\delta \|s\|, \end{aligned} \quad (33)$$

where $\delta = \lambda_{\min}(k_2) - D_f - D_g \|u_s\|$. It implies that the system tracking errors reach the SMC and remain on it with $\delta \geq 0$. From $\dot{L}_2 \leq 0$, we can see that s and $\dot{s}$ are all bounded. $\dot{L}_2 \leq -\delta \|s\|$ means that its integral exists as

$$\begin{aligned}
& + D_g \|s\| \|u_s\| + s^T g(x)(\hat{f}_a - f_a) \\
& + \nabla J^{*T}(s)(f(x) + g(x)(u_0 + \hat{f}_a - f_a) + \varphi) \\
& - \frac{1}{\beta_f} \dot{\hat{f}}_a \tilde{f}_a \\
\leq & -\lambda_{\min}(k_1) \|s\|^2 - (\lambda_{\min}(k_2) - D_f) \|s\| \\
& + D_g \|s\| \|u_s\| \\
& + (s^T + \nabla J^{*T}(s)) g(x)(\hat{f}_a - f_a) \\
& - \lambda_{\min}(k_1) \|s\|^2 - \lambda_{\min}(k_2) \|s\| \\
& + \varphi_M \|s\| - Z(s, u_s) - \frac{1}{\beta_f} \dot{\hat{f}}_a \tilde{f}_a \\
\leq & -2\lambda_{\min}(k_1) \|s\|^2 - (2\lambda_{\min}(k_2) - D_f) \|s\| \\
& + \frac{1}{2} D_g^2 \|u_s\|^2 + \frac{1}{2} \|s\|^2 \\
& - s^T Q s - u_s^T R u_s + \left(2u_s^T R - s^T g(x) - \frac{1}{\beta_f} \dot{\hat{f}}_a\right) \tilde{f}_a \\
\leq & -2\lambda_{\min}(k_1) \|s\|^2 - v \|s\| \\
& + \frac{1}{2} D_g^2 \|u_s\|^2 + \frac{1}{2} \|s\|^2 - \lambda_{\min}(Q) \|s\|^2 \\
& - \lambda_{\min}(R) \|u_s\|^2 + \left(2u_s^T R - s^T g(x) - \frac{1}{\beta_f} \dot{\hat{f}}_a\right) \tilde{f}_a,
\end{aligned} \quad (41)$$

where $v=2\lambda_{\min}(k_2) - D_f$. Substituting (39) into (41), one can obtain

$$\begin{aligned}
\dot{L}_3 \leq & -\left(\lambda_{\min}(Q) + 2\lambda_{\min}(k_1) - \frac{1}{2}\right) \|s\|^2 \\
& - v \|s\| - \left(\lambda_{\min}(R) - \frac{1}{2} D_g^2\right) \|u_s\|^2 \\
\leq & -\Gamma_1 \|s\|^2 - \Gamma_2 \|u_s\|^2 - v \|s\|,
\end{aligned} \quad (42)$$

where $\Gamma_1 = \lambda_{\min}(Q) + 2\lambda_{\min}(k_1) - \frac{1}{2}$ and $\Gamma_2 = \lambda_{\min}(R) - \frac{1}{2} D_g^2$. Thus, we can obtain that $\dot{L}_3 \leq 0$ when $\lambda_{\min}(Q) + 2\lambda_{\min}(k_1) \geq \frac{1}{2}$, $\lambda_{\min}(R) \geq \frac{1}{2} D_g^2$, and $v \geq 0$. Hence, asymptotic stability of the MRR tracking error is ensured with the developed SMOFCC policy. This completes the proof.

Remark 3 The difference between model-free control and model-based control lies in whether the dynamic model of the controlled plant is known or not. In this paper, the SMOFCC scheme is designed based on known system dynamics, and it surely can be extended to model-free case as long as the system dynamics is available. To achieve this goal, one strategy is to employ the observer [15] or identifier [48] to estimate the system dynamics, and then directly applied it to propose the control method. While, another way is to develop a pure model-free control method, i.e., the controller is designed directly with system input-output data [28].

Table 1 Control parameters

Parameter	$[q_1, q_2, \dot{q}_1, \dot{q}_2]$	Q	R	β_c	β_φ
Value	$[1, 1, 0, 0]$	$6I_2$	$2I_2$	0.001	5
Parameter	W_{c0}	β_f	k_1	k_2	Λ
Value	$[10; 15; 20]$	60	$5I_2$	$5I_2$	1.2

Simulation study and results analysis

In this section, a 2-DOF MRR (see configuration b in [3]) is employed to verify the effectiveness of the theoretical results of SMOFCC comparatively.

The desired trajectories of two joint modules are defined as

$$q_d = \begin{bmatrix} q_{1d} \\ q_{2d} \end{bmatrix} = \begin{bmatrix} 0.4 \sin(0.6t) + 0.3 \cos(0.5t) \\ 0.2 \cos(0.9t) - 0.5 \sin(0.4t) \end{bmatrix} \quad (43)$$

and an unknown additive actuator failure is assumed to be

$$f_a(t) = \begin{cases} [0; 0], & 0 \leq t \leq 30s \\ [0.2 + 0.1 \sin(0.4t); 0], & 30s < t \leq 60s. \end{cases} \quad (44)$$

To approximate (8), we employ a CNN (18) with $M = 3$, the weight vector as $\hat{W}_c = [\hat{W}_{c1}, \hat{W}_{c2}, \hat{W}_{c3}]^T$, and the activation function as $\sigma_c = [s_1^2, s_1 s_2, s_2^2]^T$, respectively. Related initial weight vectors and control parameters are listed in Table 1.

To show the superiority of the proposed scheme, we compare the control performance of the developed SMOFCC scheme with the existing optimal control scheme that based on tracking error feedback only in [3]. Figures 2, 3, 4 and 5 illustrate the simulation results under the proposed control strategy, and Figs. 6, 7, 8 and 9 depict the simulation results under the control method in [3].

The designed compensator (39) is employed to estimate the fault amplitude online. From Figs. 2 and 6, we can observe that the estimated failure tracks the actual failure within less than 1 s after the system runs with the developed SMOFCC scheme, while it takes near 10 s with the control scheme in [3]. Moreover, the SMOFCC can obtain a smaller overshoot. That is to say, the estimated fault amplitude with the proposed compensator (39) has a smaller bias and a higher accuracy. Figure 3 shows that the actual trajectories track the desired ones within 3 s under the proposed scheme, and a faster convergence rate of MRR is obtained compared to that in [3]. Meanwhile, compared to Fig. 7, Fig. 3 shows that SMOFCC provides a faster convergence at the beginning of the system runs in fault free scenario. We can see from Fig. 4 that the tracking errors gradually decrease and reach steady state after 4 s, which indicates the above results more intuitively. Figures 5 and 9 illustrate the control inputs of two different

Fig. 2 Online fault estimation under the developed SMOFCC

Fig. 4 Tracking errors under the developed SMOFCC

Fig. 3 Trajectories tracking performance under the developed SMOFCC

control methods, respectively. We can see that the control input of joint 1 under the SMOFCC has a slight change after the actuator failure occurs at $t = 30$ s due to the online fault compensation. From above simulation results, we can see that the SMOFCC has a faster convergence rate, a smaller overshoot and a higher accuracy. That is because the SMOFCC scheme introduces the SMC technique which combines the

Fig. 5 Control inputs under the developed SMOFCC

position tracking error e and the velocity tracking error $\dot{e}$ similar to the proportion and derivation controller [49]. The proportion regulation improves the system convergence rate and reduces the system error, and derivation reduces system overshoot and tuning time. Besides, the FDI unit is removed since introducing the online fault estimation. Therefore, the fault diagnosis time is greatly reduced and a good fault tolerance performance can be obtained in spite of the actuator failure occurs. Furthermore, the closed-loop MRR system is asymptotically stable, rather than UUB. The system states can be recovered by the online fault compensation after the actuator failure occurs. Thus, the tracking performance under the designed SMOFCC is superior to that in [3]. In summary, the proposed control scheme performs a better tracking and

Fig. 6 Online fault estimation under the optimal control in [3]

Fig. 7 Trajectories tracking performance under the optimal control in [3]

fault tolerant performance for MRRs by introducing SM surface.

Conclusion

In this paper, we propose SMOFCC scheme which extends the ADP-based control with SMC technique to solve the

Fig. 8 Tracking errors under the optimal control in [3]

Fig. 9 Control inputs under the optimal control in [3]

FTC problem of MRRs with unknown actuator failures. The SMOFCC consists of the SM-based iterative controller, an adaptive robust term and an online fault compensator. Thus, the prior nominal controller that relies on the knowledge of accurate dynamic model can be relaxed based on the SMC technique. Hereafter, the stability of the closed-loop MRR system is ensured to be asymptotically stable, rather than UUB. Based on the online estimated actuator failures, the proposed SMOFCC scheme removes the FDI unit. Comparable simulation results show that the developed scheme can provide a faster convergence and a less overshoot than existing optimal control methods which were developed based on tracking error feedback only. In the future work, the approx-

imated optimal FTC problems for MRRs with other fault scenarios and noises will be further considered.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Gams A, Nemec B, Ijspeert AJ, Ude A (2014) Coupling movement primitives: interaction with the environment and bimanual tasks. *IEEE Trans Robot* 30(4):816–830
- Christoph HB, Jamie P (2018) Mori: a modular origami robot. *IEEE/ASME Trans Mechatron* 22(5):2153–2164
- Li Y, Xia H, Zhao B (2018) Policy iteration algorithm based fault tolerant tracking control: an implementation on reconfigurable manipulators. *J Electr Eng Technol* 13(4):1740–1751
- Meister E, Gutenkunst A, Levi P (2013) Dynamics and control of modular and self-reconfigurable robotic systems. *Int J Adv Intell Syst* 6(1&2):66–78
- Kirchoff S, Melek WW (2008) Distributed control of modular and reconfigurable robot with torque sensing. *Robotica* 26(1):75–84
- Li Z, Melek WW, Clark C (2013) Distributed fault detection for modular and reconfigurable robots with joint torque sensing: a prediction error based approach. *Mechatronics* 23(6):607–616
- Kirchoff S, Melek WW (2007) A saturation-type robust controller for modular manipulators arms. *Mechatronics* 17(4):175–190
- Li Z, Melek WW, Clark C (2009) Decentralized robust control of robot manipulators with harmonic drive transmission and application to modular and reconfigurable serial arms. *Robotica* 27(2):291–302
- Zhu M, Li Y (2010) Decentralized adaptive fuzzy sliding mode control for reconfigurable modular manipulators. *Int J Robust and Nonlinear Control* 20(4):472–488
- Li Y, Ding G, Zhao B (2016) Decentralized adaptive neural network sliding mode position/force control of constrained reconfigurable manipulators. *J Cent South Univ* 23:2917–2925
- Zhou F, Dong B, Li Y (2017) Torque sensorless force/position decentralized control for constrained reconfigurable manipulator with harmonic drive transmission. *Int J Control Autom Syst* 15(5):2364–2375
- Zhao B, Li C, Ma T, Li Y (2015) Multiple faults detection and isolation via decentralized sliding mode observer for reconfigurable manipulator. *J Electr Eng Technol* 10(6):2393–2405
- Zhao B, Li Y, Liu D (2017) Self-tuned local feedback gain based decentralized fault tolerant control for a class of large-scale nonlinear systems. *Neurocomputing* 235:147–156
- Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modeling. In: *Handbook of intelligent control: neural, fuzzy, and adaptive approaches*, chapter 13
- Zhao B, Liu D (2020) Event-triggered decentralized tracking control of modular reconfigurable robots through adaptive dynamic programming. *IEEE Trans Ind Electron* 67(4):3054–3064
- Zhao B, Luo F, Lin H, Liu D (2021) Particle swarm optimized neural networks based local tracking control scheme of unknown nonlinear interconnected systems. *Neural Netw* 134:54–63
- Fu Y, Fu J, Chai T (2015) Robust adaptive dynamic programming of two-player zero-sum games for continuous-time linear systems. *IEEE Trans Neural Netw Learn Syst* 26(12):3314–3319
- Zhang Q, Zhao D, Wang D (2018) Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming. *IEEE Trans Neural Netw Learn Syst* 29(1):37–50
- Yang H, Ying Li, Yuan H, Liu Z (2018) Adaptive dynamic programming for security of networked control systems with actuator saturation. *Inf Sci* 460–461:51–64
- Yang Y, Vamvoudakis KG, Modares H, Yin Y, Wunsch DC (2020) Hamiltonian-driven hybrid adaptive dynamic programming. *IEEE Trans Syst Man Cybern: Syst*. <https://doi.org/10.1109/TSMC.2019.2962103>
- Yang Y, Vamvoudakis KG, Modares H, Yin Y, Wunsch DC (2020) Safe intermittent reinforcement learning with static and dynamic event generators. *IEEE Trans Neural Netw Learn Syst* 31(12):5441–5455
- Wen G, Philip Chen CL, Ge SS, Yang H, Liu X (2019) Optimized adaptive nonlinear tracking control using actor-critic reinforcement learning strategy. *IEEE Trans Ind Inform* 15(9):4969–4977
- Zhao B, Liu D, Luo C (2020) Reinforcement learning-based optimal stabilization for unknown nonlinear systems subject to inputs with uncertain constraints. *IEEE Trans Neural Netw Learn Syst* 31(10):4330–4340
- Bai W, Li T, Tong S (2020) NN reinforcement learning adaptive control for a class of nonstrict-feedback discrete-time systems. *IEEE Trans Cybern* 50(11):4573–4584
- Bai W, Zhou Q, Li T, Li H (2020) Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation. *IEEE Trans Cybern* 50(8):3433–3443
- Shi J, Yue D, Xie X (2020) Adaptive optimal tracking control for nonlinear continuous-time systems with time delay using value iteration algorithm. *Neurocomputing* 396:172–178
- Zhang K, Zhang H, Jiang H, Wang Y (2018) Near-optimal output tracking controller design for nonlinear systems using an event-driven ADP approach. *Neurocomputing* 309(2):168–178
- Yang Y, Guo Z, Xiong H, Ding D, Yin Y, Wunsch DC (2019) Data-driven robust control of discrete-time uncertain linear systems via off-policy reinforcement learning. *IEEE Trans Neural Netw Learn Syst* 30(12):3735–3747
- Wu S, Grimble M, Wei W (2000) QFT based robust/fault tolerant flight control design for a remote pilotless vehicle. *IEEE Trans Control Syst Technol* 8(6):1010–1016
- Bonivento C, Isidori A, Marconi L, Paoli A (2004) Implicit fault-tolerant control: application to induction motors. *Automatica* 40(3):355–371
- Benosman M, Lum KY (2010) Passive actuators' fault-tolerant control for affine nonlinear systems. *IEEE Trans Control Syst Technol* 18(1):152–163
- Zhao B, Liu D, Li Y (2017) Observer based adaptive dynamic programming for fault tolerant control of a class of nonlinear systems. *Inf Sci* 384:21–33
- Chen W, Saif M (2006) An iterative learning observer for fault detection and accommodation in nonlinear time-delay systems. *Int J Robust Nonlinear Control* 16(1):1–19
- Zakharov A, Zattoni E, Yu M, Jounela SLJ (2015) A performance optimization algorithm for controller reconfiguration in

- fault tolerant distributed model predictive control. *J Process Control* 34:56–69
35. Seron MM, Jose ADD (2014) Robust actuator fault compensation accounting for uncertainty in the fault estimation. *Int J Adapt Control Signal Process* 28(12):1440–1453
 36. Mohamed AK, Yu X, Zhang Y (2018) Fault-tolerant cooperative control design of multiple wheeled mobile robots. *IEEE Trans Control Syst Technol* 26(2):756–764
 37. Avram RC, Zhang X, Muse J (2018) Nonlinear adaptive fault-tolerant quadrotor altitude and attitude tracking with multiple actuator faults. *IEEE Trans Control Syst Technol* 26(2):701–707
 38. Hmidi R, Brahim AB, Dhahri S, Hmida FB, Sellami A (2020) Sliding mode fault-tolerant control for Takagi–Sugeno fuzzy systems with local nonlinear models: application to inverted pendulum and cart system. *Trans Inst Meas Control*. <https://doi.org/10.1177/0142331220949366>
 39. Fan Q, Yang G (2016) Adaptive fault-tolerant control for affine non-linear systems based on approximate dynamic programming. *IET Control Theory Appl* 10(6):655–663
 40. Zhao B, Jia L, Xia H, Li Y (2018) Adaptive dynamic programming based stabilization of nonlinear systems with unknown actuator saturation. *Nonlinear Dyn* 93(4):2089–2103
 41. Utkin VI (1992) *Sliding modes in control and optimization*. Springer, Berlin
 42. Argha A, Li L, Su SW, Hung TN (2016) On LMI-based sliding mode control for uncertain discrete-time systems. *J Frankl Inst* 353(15):3857–3875
 43. Qin J, Ma Q, Gao F, Zheng WX (2018) Fault-tolerant cooperative tracking control via integral sliding mode control technique. *IEEE/ASME Trans Mechatron* 23(1):342–351
 44. Pazooki M, Mazinan AH (2017) Hybrid fuzzy-based sliding-mode control approach, optimized by genetic algorithm for quadrotor unmanned aerial vehicles. *Complex Intell Syst* 4(2):79–93
 45. Kommuri SK, Rath JJ, Veluvolu KC (2018) Sliding-mode based observer-controller structure for fault-resilient control in dc servomotors. *IEEE Trans Ind Electron* 65(51):918–929
 46. Sharifi E, Mazinan AH (2018) On transient stability of multi-machine power systems through Takagi–Sugeno fuzzy-based sliding mode control approach. *Complex Intell Syst* 4(3):171–179
 47. Zhao B, Liu D, Alippi C (2020) Sliding-mode surface-based approximate optimal control for uncertain nonlinear systems with asymptotically stable critic structure. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2019.2962011>
 48. Na J, Lv Y, Zhang K, Zhao J (2020) Adaptive identifier-critic-based optimal tracking control for nonlinear systems with experimental validation. *IEEE Trans Syst Man Cybern: Syst*. <https://doi.org/10.1109/TSMC.2020.3003224>
 49. Pan Y, Meng JE, Sun T, Xu B, Yu H (2017) Adaptive fuzzy PD control with stable H_∞ tracking guarantee. *Neurocomputing* 237:71–78

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.