ISA Transactions 96 (2020) 60–68

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators

Mien Van [a,*], Xuan Phu Do [b], Michalis Mavrovouniotis [c]

[a] Institute of Research and Development, Duy Tan University, Danang 550000, Viet Nam
[b] MediRobotics Laboratory, Mechatronics and Sensor Systems Technology, Vietnamese-German University, Viet Nam
[c] KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Cyprus

HIGHLIGHTS

- A novel self-tuning fuzzy PID-NFTSM (STFPID-NFTSM) and time delay estimation (TDE) is developed.
- The proposed controller possesses faster transient response, lower steady state error, guarantees a finite time convergence.
- The stability of the closed-loop hybrid system is demonstrated based on Lyapunov function.
- Compared to other state-of-the-art controllers, the proposed method has superior performance.

ARTICLE INFO

Article history:
Received 24 April 2018
Received in revised form 3 June 2019
Accepted 18 June 2019
Available online 26 June 2019

Keywords:
Fuzzy logic
Fault tolerant control
Robot manipulator
Sliding mode control
Robust control

ABSTRACT

In this work, a new robust controller is developed for robot manipulator based on an integrating between a novel self-tuning fuzzy proportional–integral–derivative (PID)-nonsingular fast terminal sliding mode control (STF-PID-NFTSM) and a time delay estimation (TDE). A sliding surface based on the PID-NFTSM is designed for robot manipulators to get multiple excited features such as faster transient response with finite time convergence, lower error at steady-state and chattering elimination. However, the system characteristics are hugely affected by the selection of the PID gains of the controller. In addition, the design of the controller requires an exact dynamics model of the robot manipulators. In order to obtain effective gains for the PID sliding surface, a fuzzy logic system is employed and in order to get an estimation of the unknown dynamics model, a TDE algorithm is developed. The innovative features of the proposed approach, i.e., STF-PID-NFTSM, is verified when comparing with other up-to-date advanced control techniques on a PUMA560 robot.

© 2019 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

For enhancing the reliability of the uses of robot manipulators in practical applications, fault monitoring and fault tolerant control (FTC) have been extensively studied in the last decade [1,2]. The goal of the fault monitoring scheme is to monitor if a fault occurs, while the FTC is developed to handle unexpected behaviors of the system due to the effects of faults. Generally, the approaches for FTC can be executed by using either passive FTC (PFTC) or active FTC (AFTC) [3]. Compared to the AFTC, the PFTC compensates faults' effects quicker since it does not require feedback from the fault diagnosis observer, which is required in the design of the AFTC [4]. However, since there is no online information feedback, the PFTC needs to depress the highest faults' effects in the system, and therefore, a high robustness controller is required for this control scheme. Hence, many efforts have been spent to develop high robustness PFTC systems in the literature [5,6]. Sliding mode control (SMC) has shown to supply exceptional robustness property to tackle the matched uncertainties in the system [7]. The great features of the SMC have been considerably utilized for robust FTC [8,9]. Generally, the conventional SMC uses a linear sliding manifold, a discontinuous sign function and an assumed upper bounded value of the uncertainties in its design. However, this design principle leads to several limitations for the control system, including: (1) it cannot supply a defined finite time convergence; (2) it has slow response when facing with fast variants of faults; (3) it provides big oscillation(chattering) because of the discontinuous term; and, (4) it is challenging to get the upper bound value of the unknown nonlinear function in the system (it requires a lot of experiments). From the practical point of view, these drawbacks would reduce the performance of the FTC system significantly.

In order to gain the merits of the SMC and discard the drawbacks, many advance techniques have been investigated, such as: (1) non-singular fast terminal sliding mode control (NFTSMC) to

* Corresponding author.
E-mail address: vanmien1@gmail.com (M. Van).

https://doi.org/10.1016/j.isatra.2019.06.017
0019-0578/© 2019 ISA. Published by Elsevier Ltd. All rights reserved.

get faster finite time stability [10,11]; (2) integral SMC to obtain faster response to the effects of the faults [12]; and (3) high-order SMC (HOSMC) to cope with the chattering phenomenon [13,14]. Even though the first three drawbacks of the conventional SMC has been well studied and addressed separately by an effective approach mentioned above, in many practical applications, the system must obtain all three properties (i.e., fast transient response against the effects of fault, stability and convergence in a finite time and free-chattering) simultaneously. To achieve this feature, a backstepping NFTSMC has been developed in [15]. However, this approach does not preserve the full merits of the PID controller, which could dominate the robustness and fast convergence of the system. To integrate the full features of the PID controller into the system, new PID-NFTSMC has been initially studied in previous work [16]. The analyzed results shown that the developed controller provided very quick response against the effects of faults and obtained a finite time convergence. However, the approach in [16] has two major drawbacks. First, the system characteristics strongly rely upon the selection of the PID's parameters (i.e., the proportional, the integral and the derivative gains parameters). And second, the partial component of the system's dynamics model needs to be known in advance.

One way to overcome the needs of prior knowledge of the dynamics model in the design, which is considered as one of the major drawbacks of the SMC, is to use adaptive techniques [17,18]. However, these approaches provide slow convergence speed when the unknown function is large. In addition, for the large bound value of the unknown function, the adaptive sliding parameter needs to adapt to a huge value, which, unfortunately, generates higher chattering in the system. Therefore, to reduce the chattering, the sliding gain needs to be reduced. To obtain this requirement, learning methods using neural network (NN) [19,20] or fuzzy logic [21,22] has been developed for approximating the uncertain nonlinear function. After obtaining the accurate model based on the learning methods, the sliding gain only needs to estimate the approximation error, which is usually small, and consequently, the chattering is significantly reduced. However, the use of learning techniques result in complicated designs of learning laws and require high computational burden due to the weight's learning process. In order to get simpler designs and lower computational burden, an alternative time delay estimation (TDE) technique has been developed [23,24].

In this work, a new hybrid control method, which combines the PID-NFTSM control, a TDE and a self-tuning fuzzy mechanism, is proposed to compensate the aforementioned limitations of the previous work in [16] for PFTC of robot manipulators. At the beginning, the proposed control method reconstructs a PID-NFTSM sliding surface taken from [16] for the robot manipulators. Although the integration of the PID and the NFTSM sliding surfaces has several advantages, it possesses a severe drawback: the performance of the system is strongly dependent on numerous major parameters. Tuning these parameters such that the system can obtain the best performance is usually a difficult task in research and development. In the literature, a self-tuning mechanism based on fuzzy logic system has shown to provide effective parameters for the system [25–27]. Hence, in order to effectively tune the parameters of the controllers, a fuzzy logic system is designed. Then, a TDE method is explored for removing the required prior knowledge of the dynamics model during constructing the proposed controller.

In summary, in comparison to other up-to-date innovation techniques, the merits of the proposed PFTC method can be pointed out as follows:

- Compared to the conventional NFTSMC [10,11], the integral SMC [12] and the HOSMC [13], the proposed method, i.e., STF-PID-NFTSMC, preserves the merits of all these controllers simultaneously. Hence, the proposed method provides fast finite time convergence, high robustness against faults' effects, and chattering elimination in the system.
- Compared to the previous works based on backstepping NFTMC (BNFTSMC) [15] and PID-NFTSMC [16], the contributions and merits of the proposed method are pointed out as follows. In the work [15], an integral nonsingular fast terminal sliding (INFTSM) surface is proposed (Eq. (7) in [15]). Then, the first and second derivatives of the sliding surface are computed (Eqs. (9) and (10) in [15]). Combining the proposed sliding surface and its first and second derivative, a third-order state-space model is obtained (Eq. (11) in [15]). Finally, a backstepping control is designed to stabilize the third-order state-space system. It can be observed that the design in [15] contains an integral component (Eq. (7) in [15]), a proportional component (Eq. (9) in [15]) and a derivative component (Eq. (10) in [15]). Therefore, it can be roughly considered as a proportional–integral–derivative (PID)-nonsingular fast terminal sliding mode (PID-NFTSM) controller (or we can call PID-NFTSM-like controller) with the gains $K_p = 1, K_i = 1$, and $K_d = 1$. Obviously, this approach does not preserve the full features of the PID controller. In the work [16], a proportional integral derivative-nonsingular fast terminal sliding mode (PID-NFTSM) surface is proposed. This approach can preserve the merits of both the PID and NFTSM controllers (the gains, i.e., K_p, K_i and K_d, are selected based on experiences). Thus, this approach generally provides better performance than the approach in [15] in terms of quick response and fast finite time convergence. However, it preserves two major drawbacks mentioned above (the system characteristics strongly rely upon the selection of the PID's parameters and the partial component of the system's dynamics model needs to be known in advance). In the proposed method in this paper, i.e., STF-PID-NFTSMC, effective parameters of the controller are obtained based on a fuzzy logic system so that the performance of the system is improved. Although the learning method based on fuzzy approximation has been developed to approximate the model uncertainty and faults in [16], its computational load is high as discussed above. Hence, in this work, the developed TDE helps to remove the awareness of the robot's dynamics model and significantly reduce the computational load of the system since the computational burden of the TDE is much lower compared to the use of fuzzy approximation technique in [16].
- The complexity and stability issues of the hybrid system have been solved properly.
- A comprehensive comparison with the existing PFTC methods such as the computed torque control (CTC), the PID, the PID-SMC, the NFTSMC and the PID-NFTSMC has been implemented to show the superior of the proposed strategy.

The next sections are constructed as follows. Section 2 states the problem. Section 3 presents the design of robust fault tolerant control using the PID-NFTSMC controller and the time-delay estimation. The design of the self-tuning fuzzy PID-NFTSMC and the time-delay estimation is introduced in Section 4. Discussions on the simulation results for the performance of the proposed strategy are presented in Section 5. Conclusions are provided in Section 6.

2. Problem statement

A typical dynamics model of an n-DOF (degree-of-freedom) robot manipulator is described as follows [11]:

$$\ddot{q} = M^{-1}(q)\left(\tau - \Xi(q,\dot{q}) - \Pi(\dot{q},\tau_d)\right) + \eta(t - T_f)\phi(q,\dot{q},\tau) \qquad (1)$$

where $q \in \Re^n$, $\dot{q} \in \Re^n$, $\ddot{q} \in \Re^n$ are the angle, the angular velocity, and the angular acceleration of the joints, respectively. $M(q) \in \Re^{nxn}$ represents the inertia matrix. $\tau \in \Re^n$ is the control input torque. $\Xi(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q)$, in which $C(q, \dot{q}) \in \Re^n$, $G(q) \in \Re^n$ represents the Coriolis/ Centripetal forces and the gravity force, respectively. $\Pi(\dot{q}, \tau_d) = F(\dot{q}) + \tau_d$, in which $F(\dot{q}) \in \Re^n$ is the friction matrix and τ_d is the disturbance torque. The term $\phi(q, \dot{q}, \tau) \in \Re^n$ is a vector representing the faults. In this paper, we consider the effects of the component faults and actuator faults since they usually occur in the robot system. Both can be represented by the term $\phi(q, \dot{q}, \tau)$. Finally, $\eta(t - T_f) \in \Re^n$ represents the variant of the fault function with respect to time, where T_f indicates the moment that the faults occur. These functions are defined in the same way as in [11].

The inertia matrix $M(q)$ is assumed to satisfy the following condition:

$$0 < \lambda_{\min}\{M(q)\} \leq \|M\| \leq \lambda_{\max}\{M(q)\} \leq d, \quad d > 0 \tag{2}$$

where $\lambda_{\max}\{M\}$ and $\lambda_{\min}\{M\}$ denotes the biggest and smallest eigenvalues of matrix M, respectively.

Eq. (1) can be rewritten as follows:

$$\ddot{q} = M^{-1}(q)\tau + M^{-1}(q)\left(-\Xi(q, \dot{q}) - \Pi(\dot{q}, \tau_d)\right) + \eta(t - T_f)\phi(q, \dot{q}, \tau). \tag{3}$$

Let $x_1 = q$, $x_2 = \dot{q}$ and $u = \tau$, Eq. (3) can be represented by a form of state space model [11]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1}(x_1)u + f(x_1, x_2) + \Delta(x_1, x_2, u, \tau_d) \\ y &= x_1 \end{aligned} \tag{4}$$

where $f(x_1, x_2) = M^{-1}(q)\left(-\Xi(q, \dot{q})\right)$ is a sum of known functions and $\Delta(x_1, x_2, u, \tau_d) = M^{-1}(q)\left(-\Pi(\dot{q}, \tau_d)\right) + \gamma(t - T_f)\phi(q, \dot{q}, \tau)$ represents the lumped unknown function. When the system works in normal condition, the component $\Delta(x_1, x_2, u, \tau_d)$ represents the effects of the uncertainties and disturbance, whereas when the faults occur, the term $\Delta(x_1, x_2, u, \tau_d)$ will cover all the effects of the uncertainties, disturbances and faults.

Assumption 1. The uncertainty model is bounded by:

$$|\Delta(x_1, x_2, u, \tau_d)| \leq \Delta_0 \tag{5}$$

Assumption 2. The derivative of the uncertainty model is bounded by:

$$|\dot{\Delta}(x_1, x_2, u, \tau_d)| \leq \Delta_1 \tag{6}$$

where Δ_0 and Δ_1 are constants.

The above assumptions are quite general in the literature [4,6], that is, the FTC is applied for "not exploding systems". This leads to practical satisfaction of the Lipschitz condition in the considered operation region [4].

3. Robust fault tolerant control using PID-NFTSMC and time delay estimation

3.1. Robust PID-NFTSMC fault tolerant control

The selection of the sliding surface when designing SMC significantly impacts on the system's tracking outcomes. The sliding surface is selected in such a way that if it converges to the origin, then the system can obtain the expected outcomes. Let $e = x_1 - x_d$, $\dot{e} = x_2 - \dot{x}_d$ be the trajectory tracking errors, where x_d and $\dot{x}_d$ denote a desired trajectory and a derivative of the desired trajectory, respectively. In order to achieve fast finite time convergence and eliminate the singular problem, a NFTSM sliding surface is configured as [10,11]:

$$s = e + k_1 e^{[\lambda]} + k_2 \dot{e}^{[p/q]} \tag{7}$$

where $k_1 = diag\,(k_{11}, k_{12}, \ldots, k_{1n}) \in \Re^{n\times n}$ and $k_2 = diag\,(k_{21}, k_{22}, \ldots, k_{2n}) \in \Re^{n\times n}$ are two positive definite matrices, the two positive odd numbers p and q are selected to satisfy the conditions $1 < p/q < 2$ and $\lambda > p/q$.

Then, based on the sliding surface s, a sliding surface based on PID-NFTSM is proposed as [16]:

$$s_{PID}(t) = K_p s(t) + K_i \int_0^t s(t)dt + K_d \frac{ds(t)}{dt} \tag{8}$$

where K_p, K_i and K_d are the proportional, the integral and the derivative gains, respectively. Obviously, the system response depends very much on the selection of these parameters. Therefore, these parameters need to be effectively tuned.

The derivative of the sliding surface s is derived as:

$$\begin{aligned} \frac{ds(t)}{dt} &= \dot{e} + k_1\lambda|e|^{\lambda-1} \cdot \dot{e} + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot \ddot{e} \\ &= \dot{e} + k_1\lambda|e|^{\lambda-1} \cdot \dot{e} + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot (\dot{x}_2 - \ddot{x}_d) \end{aligned} \tag{9}$$

Inserting Eq. (4) into (9), we have

$$\begin{aligned} \frac{ds(t)}{dt} &= \dot{e} + k_1\lambda|e|^{\lambda-1} \cdot \dot{e} + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \\ &\quad \cdot \left(M^{-1}(x_1)u + f(x_1, x_2) + \Delta(x_1, x_2, u, \tau_d) - \ddot{x}_d\right) \end{aligned} \tag{10}$$

From Eqs. (8) and (10) it can be observed that the gain K_d is proportional to the control input u. Therefore, the variation of this parameter has a significant effect on the stability of the whole system. For simplicity in guaranteeing the stability of the system, we choose $K_d = 1$. Then, from results in Eq. (10), Eq. (8) can be rewritten as:

$$\begin{aligned} s_{PID} &= K_p s + K_i \int_0^t s dt + \dot{e} + k_1\lambda|e|^{\lambda-1} \cdot \dot{e} \\ &\quad + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot \left(M^{-1}(x_1)u + f(x_1, x_2)\right. \\ &\quad \left. + \Delta(x_1, x_2, u, \tau_d) - \ddot{x}_d\right) \\ &= K_p s + K_i \int_0^t s dt + \dot{e} + k_1\lambda|e|^{\lambda-1} \cdot \dot{e} \\ &\quad + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot (f(x_1, x_2) - \ddot{x}_d) \\ &\quad + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} M^{-1}(x_1)u + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot \Delta(x_1, x_2, u, \tau_d) \end{aligned} \tag{11}$$

Let $\Gamma(x_1, x_2, x_d, \dot{x}_d, \ddot{x}_d) = K_p s + K_i \int_0^t s(t)dt + \dot{e} + k_1\lambda|e|^{\lambda-1} \cdot \dot{e} + k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot (f(x_1, x_2) - \ddot{x}_d)$ be the lumped known function, $\Lambda(\dot{e}, \Delta) = k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot \Delta(x_1, x_2, u, \tau_d)$ be the lumped unknown function and $\Omega(\dot{e}, x_1) = k_2\frac{p}{q}|\dot{e}|^{(p/q)-1} \cdot M^{-1}(x_1)$. Then, Eq. (11) can be simplified as follows:

$$s_{PID} = \Gamma(x_1, x_2, x_d, \dot{x}_d, \ddot{x}_d) + \Omega(\dot{e}, x_1)u + \Lambda(\dot{e}, \Delta) \tag{12}$$

The proposed sliding mode controller is developed as follows:

$$u_{PID} = -\Omega^{+}(\dot{e}, x_1)\left(u_{eq} + u_r\right) \tag{13}$$

where $\Omega^{+}(\dot{e}, x_1)$ denotes the pseudoinverse of $\Omega(\dot{e}, x_1)$, i.e., $\Omega^{+}(\dot{e}, x_1) = \left(\Omega^T(\dot{e}, x_1)\Omega(\dot{e}, x_1)\right)^{-1}\Omega^T(\dot{e}, x_1)$, $\Omega^T(\dot{e}, x_1)$ denotes

the transpose of the matrix $\Omega(\dot{e}, x_1)$, and the equivalent control is designed as follows:

$$u_{eq} = \Gamma\left(x_1, x_2, x_d, \dot{x}_d, \ddot{x}_d\right) \tag{14}$$

and the switching term is made of:

$$\dot{u}_r = (\Upsilon + a) sign(s_{PID}) \tag{15}$$

where the constant Υ is chosen such that $\delta_\Lambda = \frac{d(\Lambda(\dot{e}, \Delta))}{dt}$ and $|\delta_\Lambda| \leq \Upsilon$, and a is a small positive constant.

Theorem 1. *Consider the system model described in Eq. (4) and the proposed sliding surface in Eqs. (7) and (8), if the composite controller in Eqs. (13), (14) and (15) are employed as the control input to the system, then sliding surface is stable and convergent to zero.*

Proof. Inserting Eqs. (13)–(15) into Eq. (12), we have

$$s_{PID} = -u_r + \Lambda\left(\dot{e}, \Delta\right) \tag{16}$$

Differentiating the sliding variable in Eq. (16) with respect to time, we have

$$\dot{s}_{PID} = -\dot{u}_r + \delta_\Lambda \tag{17}$$

Consider the following Lyapunov function candidate

$$V = \frac{1}{2} s_{PID}^T s_{PID} \tag{18}$$

Differentiating Eq. (18) and combining the results with Eqs. (16) and (17), one obtains

$$\begin{aligned} \dot{V} &= s_{PID}^T \dot{s}_{PID} \\ &= s_{PID}^T \left(-\dot{u}_r + \delta_\Lambda\right) \\ &= s_{PID}^T \left(-(\Upsilon + a) sign(s_{PID}) + \delta_\Lambda\right) \\ &= -\Upsilon \left|s_{PID}\right| + \delta_\Lambda s_{PID} - a \left|s_{PID}\right| \leq -a \left|s_{PID}\right| \end{aligned} \tag{19}$$

As a result, the sliding surface (8) is stable and convergent according to the Lyapunov criterion. This proof is completed.

Remark 1. Compared to the conventional NFTSMC [10,11] and other existing conventional sliding mode surfaces [8,9], the proposed PID sliding surface in Eq. (8) possesses three major terms as analyzed in our previous work [16]. First, the presence of the component $K_p s(t)$ helps to maintain the properties of the conventional NFTSMC. Second, the presence of the component $K_i \int_0^t s dt$ helps to get high robustness property like a manner to the integral SMC. Third, the presence of the component $K_d \frac{ds(t)}{dt}$ helps to get the chattering elimination similar to the HOSMC. Therefore, theoretically, the proposed sliding surface inherits the major benefits of the NFTSMC, the integral SMC and the HOSMC simultaneously.

3.2. Adaptive PID-NFTSMC and TDE for robust fault tolerant control

In the developed control law in Eq. (13), the chattering is massively reduced because the discontinuous function is under integral, but it cannot be eliminated entirely by this method. In addition, the sliding gain of the controller is chosen by utilizing the prior knowledge of $|\delta_\Lambda|$. However, this assumption prevents the applicability of the proposed algorithm because it is difficult to satisfy the assumption in real applications. To reduce the chattering and remove the assumption, a time delay estimation technique and an adaptive law are developed in this subsection.

From Eq. (12), the unknown component, i.e., $\Lambda\left(\dot{e}, \Delta\right)$, at the time t can be determined as:

$$\Lambda(\dot{e}, \Delta)_{(t)} = s_{PID_{(t)}} - \Gamma(x_1, x_2, x_d, \dot{x}_d, \ddot{x}_d)_{(t)} - \Omega(\dot{e}, x_1)_{(t)} u_{(t)} \tag{20}$$

However, since the control input u at the time t is not available, Eq. (20) is not calculable. To get an estimation, TDE is employed. The underlying principle of the TDE is to approximate the unknown component at the time t using its previous value at the time delay $(t - L)$ subject to the sufficiently small value of the time delay L. Actually, when the Assumptions 1 and 2 are satisfied and the time delay L is sufficiently small, the following result holds [11]:

$$\Lambda(\dot{e}, \Delta)_{(t)} \cong \Lambda(\dot{e}, \Delta)_{(t-L)} \tag{21}$$

Therefore, the unknown function can be approximated by:

$$\hat{\Lambda}(\dot{e}, \Delta)_{(t)} \triangleq \Lambda(\dot{e}, \Delta)_{(t-L)} \tag{22}$$

where $\hat{\Lambda}(\dot{e}, \Delta)_{(t)}$ is the estimation of the unknown function $\Lambda(\dot{e}, \Delta)_{(t)}$ at time t.

From Eqs. (12) and (22), we have:

$$\begin{aligned} \hat{\Lambda}(\dot{e}, \Delta)_{(t)} &\triangleq \Lambda(\dot{e}, \Delta)_{(t-L)} \\ &= s_{PID_{(t-L)}} - \Gamma(x_1, x_2, x_d, \dot{x}_d, \ddot{x}_d)_{(t-L)} \\ &\quad - \Omega(\dot{e}, x_1)_{(t-L)} u_{(t-L)} \\ &= u_{TDE_{(t)}} \end{aligned} \tag{23}$$

Then, the unknown function can be expressed by:

$$\Lambda\left(\dot{e}, \Delta\right) = u_{TDE} + \varepsilon \tag{24}$$

where ε denotes the error due to TDE approximation. The following assumption is considered:

Assumption 3. the derivative of the approximation error ε is bounded by:

$$\delta_\varepsilon = \left|\frac{d\varepsilon(t)}{dt}\right| \leq \kappa \tag{25}$$

where κ is an unknown constant.

When the sample time of the system, i.e., L, is small, the approximation error ε and its derivative expressed in Eq. (25) will be bounded due to the estimation capability of the TDE. Therefore, Assumption 3 can be satisfied in the practical applications.

The proposed controller is now designed as follows:

$$u_{PID} = -\Omega^+(\dot{e}, x_1)\left(u_{eq} + u_{TDE} + u_{ar}\right) \tag{26}$$

where u_{eq} and u_{TDE} are introduced in Eqs. (14) and (23), respectively.

The adaptive law is designed as follows:

$$\dot{u}_{ar} = (\hat{\kappa} + a) sign(s_{PID}) \tag{27}$$

where $\hat{\kappa}$ is the estimation of the bounded value κ. It is updated by the following law:

$$\dot{\hat{\kappa}} = \frac{1}{c} \left|s_{PID}\right| \tag{28}$$

where c is the adaptation gain.

Theorem 2. *Consider the dynamics model of the system described in Eq. (4) and the proposed sliding surface in Eqs. (7) and (8), if the TDE in Eq. (23) is employed to approximate the unknown dynamics model and the controllers defined in Eqs. (26)–(28) are employed as the control input to the system, then the stability and convergence of the system are guaranteed.*

Proof. Inserting the proposed control law defined in Eqs. (26)–(28) into Eq. (12), we have:

$$s_{PID} = -u_{ar} + \varepsilon \tag{29}$$

Differentiating the sliding variable in Eq. (29) with respect to time and using Eq. (25), we have:

$$\dot{s}_{PID} = -\dot{u}_{ar} + \delta_{\varepsilon} \tag{30}$$

Let $\tilde{\kappa} = \kappa - \hat{\kappa}$ be the adaptation error. Consider the following Lyapunov function candidate

$$V = \frac{1}{2} s_{PID}^T s_{PID} + \frac{1}{2} c \tilde{\kappa}^T \tilde{\kappa} \tag{31}$$

Differentiating Eq. (31) and combining the results with Eqs. (28) and (30), one obtains

$$\begin{aligned} \dot{V} &= s_{PID}^T \dot{s}_{PID} + c(\hat{\kappa} - \kappa)\dot{\hat{\kappa}} \\ &= s_{PID}^T \left(-\dot{u}_r + \delta_{\varepsilon}\right) + c(\hat{\kappa} - \kappa)\dot{\hat{\kappa}} \\ &= s_{PID}^T \left(-(\hat{\kappa} + a) sign(s_{PID}) + \delta_{\varepsilon}\right) + (\hat{\kappa} - \kappa)\,|s_{PID}\,| \\ &= -\kappa\,|s_{PID}| + \delta_{\varepsilon} s_{PID} - a\,|s_{PID}| \;\leq -a\,|s_{PID}| \end{aligned} \tag{32}$$

As a result, the output of Theorem 2 is guaranteed according to the Lyapunov criterion and this completes our proof.

Remark 2. Due to the effects of disturbance, the sliding surface s_{PID} cannot maintain $s_{PID} = 0$ in infinite time. Therefore, the adaptation gain κ in Eq. (28) keeps increasing continuously. This symptom is known as the "parameter drift problem". In order to eliminate this problem, the following adaptive law can be employed:

$$\dot{\hat{\kappa}} = \begin{cases} 0, & \text{if } |s_{PID}| < \rho \\ \dfrac{1}{c}\,|s_{PID}\,|, & \text{if } |s_{PID}| \geq \rho \end{cases} \tag{33}$$

where ρ is a small positive constant.

Remark 3. The selected sliding surface in Eq. (7) and the proposed controllers in Eqs. (26)–(28) requires the full measurement of the states. However, in some practical robot models, only the position is measurable. Therefore, to estimate the velocity and acceleration of the system, the following second-order exact differentiator (SOED) proposed in [14] is employed:

$$\begin{aligned} \dot{z}_0 &= \upsilon_0 \\ \upsilon_0 &= -m_1 |z_0 - x_1|^{2/3} sign(z_0 - x_1) + z_1 \\ \dot{z}_1 &= \upsilon_1 \\ \upsilon_1 &= -m_2 |z_1 - \upsilon_0|^{1/2} sign(z_1 - \upsilon_0) + z_2 \\ \dot{z}_2 &= -m_3 sign(z_2 - \upsilon_1). \end{aligned} \tag{34}$$

where $m_i (i = 1, 2, 3)$ are the gains that need to be effectively selected.

After converging, the SOED can provide

$$z_0 = \hat{x}_1,\ z_1 = \hat{x}_2,\ z_2 = \dot{\hat{x}}_2 \tag{35}$$

In this sense, the measurement of x_2 and $\dot{x}_2$ in the proposed sliding surface in Eq. (7) and the controllers in Eqs. (26)–(28) can be replaced by the estimations z_1 and z_2, respectively. For example, the derivative of the tracking error can be re-defined as $\dot{e} = z_1 - \dot{x}_d$, etc.

Remark 4. The SOED in Eq. (34) provides the accurate states estimation regardless of what the control input is. It means that the states estimation based on SOED and the controller can be independently constructed. Therefore, the closed-loop observer-controller can obtain the full merit features like the controller with full state measurement.

Remark 5. The major drawback of the learning technique is that the computational burden is high due to the weights learning process [19–22]. This paper employed the TDE method to significantly reduce the computational burden of the system as it is simple in computing.

4. Design of self-tuning Fuzzy-PID-NFTSMC for robust fault tolerant control using time delay estimation

In the proposed controller, it is important to select the effective values for the gains K_p and K_i so that the system satisfies the desired performance, i.e., fast convergence, small oscillatory and less overshoot. In this work, these parameters are automatically tuned by a fuzzy logic system. Through fuzzy knowledge, the tuners can be established as follows:

$$K_a(t) = K_{a0} + \beta \Delta K_a(t), \tag{36}$$

where a is p or i, K_{a0} is the constant learning rate, $\Delta K_a(t)$ is the tuning parameter, which is achieved based on a fuzzy logic system and β is a constant coefficient. A common structure of fuzzy logic includes four parameters: fuzzifier, knowledge base, inference engine, and defuzzifier.

Let $x = [x_1, x_2, \ldots, x_n]^T \in \Re^n$ be the input and $y \in \Re$ be the output of the fuzzy system, a set of conditional "IF-THEN" rules is created to establish the fuzzy rule base, as follows:

$$Rule\ l: \text{If } x_1 \text{ is } A_1^l \text{ and } x_2 \text{ is } A_2^l \text{ and} \ldots \text{and } x_n \text{ is } A_n^l \text{ then } y \text{ is } B^l, \tag{37}$$

where A_k^l and B^l denotes the linguistic variables of the input and output of the fuzzy set, respectively. The symbol $k = 1, \ldots, n$ indicates the number of inputs, whereas the symbol $l = 1, \ldots, m$ indicates the number of fuzzy IF-THEN rules.

In this work, two inputs: absolute error, i.e, $|e(k)|$, and absolute derivative of error, i.e., $|de(k)|$, are used as inputs of the fuzzy logic system. Each input is defined by:

$$\{Z,\ VS,\ S,\ M,\ B\} \tag{38}$$

where $Z = Zero$, $VS = Very\ Small$, $S = Small$, $M = Medium$, and $B = Big$, as shown in Fig. 1.

The fuzzy output is calculated as:

$$u(k) = \frac{\sum_{j=1}^{m} h_j u_j}{\sum_{j=1}^{m} h_j} \tag{39}$$

where $h_j = \mu_j(|e(k)|) \wedge \mu_j(|de(k)|)$, $\mu_j(|e(k)|)$ and $\mu_j(|de(k)|)$ are the membership functions (MFs) of the fuzzy subsets $|e(k)|$ and $|de(k)|$. The operator $\wedge$ denotes the minimum. There are several MFs that can be used for the controller, however triangular MFs have been selected in this paper due to their simplicity and straightforward implementation. The MFs for $|e(k)|$ and $|de(k)|$ can be formed with five triangular functions with equal base and 50% overlap with neighboring MFs. Fig. 1a shows MFs outlined on the commonly normalized area between 0 and 1. The MFs for the control output, $\Delta K_a(a = p, i)$, is defined on $[0, 4]$, as shown in Fig. 1b.

Generally, there is no definite technique for defining the rules for the fuzzy logic system. In fact, the set of rules are commonly established based on the knowledge about the operation principles of the control system. By analyzing the dynamics behavior and the output error, the tuning rules for $\Delta K_a(a = p, i)$ in this work are designed as in Table 1. The following three factors are used to establish the fuzzy rules:

1. When the error $|e(k)|$ and the derivative of error $|de(k)|$ are small, the output ΔK_p should be small while the output ΔK_i should be large to maintain the small tracking error.

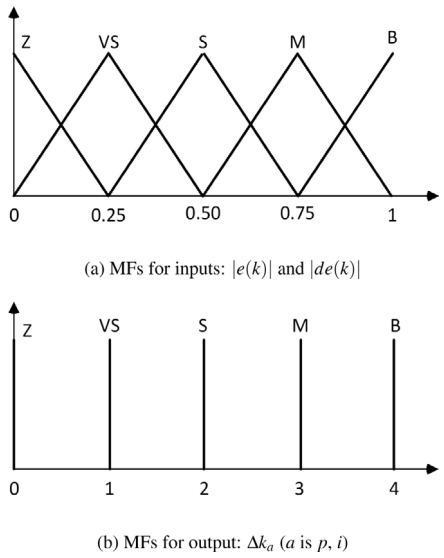


Fig. 1. Membership functions of fuzzy inputs and output of the fuzzy PI-D sliding surface.

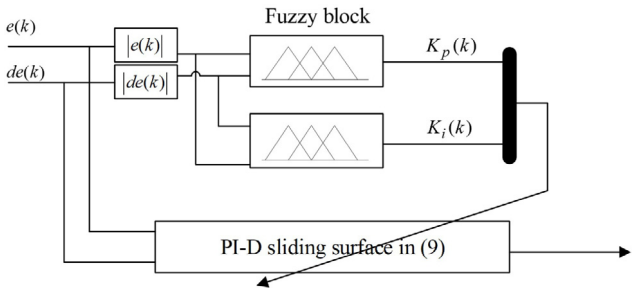


Fig. 2. Self-tuning fuzzy PI-D sliding surface architecture.

2. When the error $|e(k)|$ is large, the output ΔK_p should be large to force the system to converge to zero faster. Meanwhile, the output ΔK_i should be small to avoid large oscillation and overshoot.
3. Other rules are configured based on the behavior observations of the system through the simulations. For example, when $|e(k)|$ is **M** and $|de(k)|$ is **S**, we can set the rule for ΔK_i first, and then adjust the rule for ΔK_p. Then, observe and compare the dynamic behavior of the system for each selected rule of ΔK_p. The best behavior of the system is selected as the final rule, i.e., rule **M** in Table 1. After obtaining the rule for ΔK_p, we adjust the rule ΔK_i such that the best system behavior can be achieved, i.e., the rule **VS** is obtained. Finally, the rule **(M,VS)** is established when $|e(k)|$ is **M** and $|de(k)|$ is **S**. Similar procedure can be performed for selecting other fuzzy rules.

The structure of the tuning mechanism for K_p and K_i is illustrated in Fig. 2. The architecture of the proposed controller is interpreted in Fig. 3.

Remark 6. The design of the self-tuning fuzzy logic for PID/PI controllers is quite standard in the literature. The importance and novelty of the design of fuzzy logic for such applications is to construct the effective fuzzy rules. This paper employs this well-known technique to select the effective proportional and integral gains of the PID sliding surface.

Remark 7. The system stability under the use of the controller defined in Eq. (26) has been proved in Eq. (32). At the same time, the employment of the fuzzy logic to tune the PID gains does not affect the system's stability. Therefore, the system's stability can be guaranteed thoroughly.

5. Results and discussions

In this section, the effectiveness of the developed control strategy is verified. Without loss of generality, the PUMA560 robot [28] is used as a benchmark for testing. For the sake of results' evaluation, only the first three joints of the robot are considered. For the configuration of the PUMA560 robot and its kinematic and dynamics model, the reader can refer to [28]. In the practical engineering application, the exact information of the dynamics model of the robot is difficult to obtain. Therefore, we assume that the following friction and disturbance components existed in the robot dynamics:

$$F(q)+\tau_d=\begin{bmatrix}1.5\dot{q}_1+0.5\sin(3q_1)+1.2sin(\dot{q}_1)\\ 2.3\dot{q}_2-1.2\sin(2q_2)+0.95sin(\dot{q}_2)\\ -2.1\dot{q}_3-1.6\sin(3q_3)+0.75sin(\dot{q}_3)\end{bmatrix} \tag{40}$$

The robot is assumed to follow the following trajectory:

$$x_d=\left[\cos(t/5\pi)-1,\ \cos(t/5\pi+\frac{\pi}{2}),\ \sin(t/5\pi+\frac{\pi}{2})-1\right]^T \tag{41}$$

The proposed controller, i.e., STF-PID-NFTSMC, is compared with up-to-date innovative PFTC techniques such as the computed torque control (CTC), the proportional-derivative-integral (PID), the NFTSMC [10,11], the PID-SMC [12] and the PID-NFTSMC [16]. The parameters used in this simulation of the controllers are selected as in Table 2. To verify the performance of the controllers, we consider their performances on the robot when it works in two circumstances: normal and fault operations.

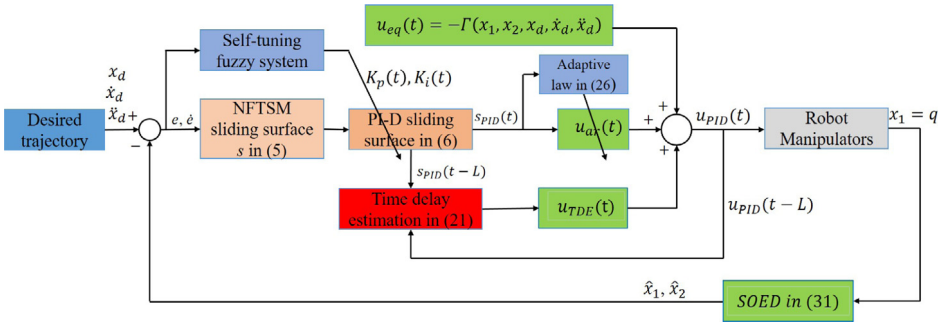
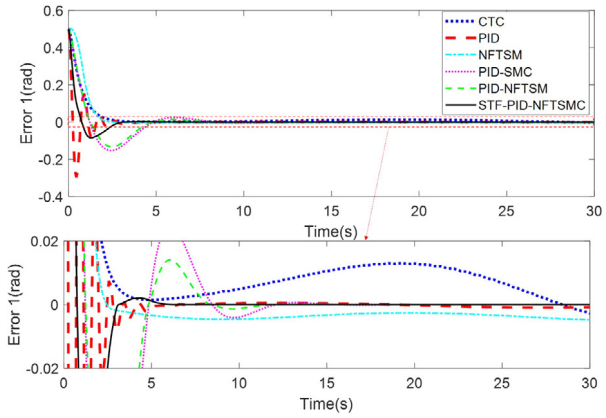
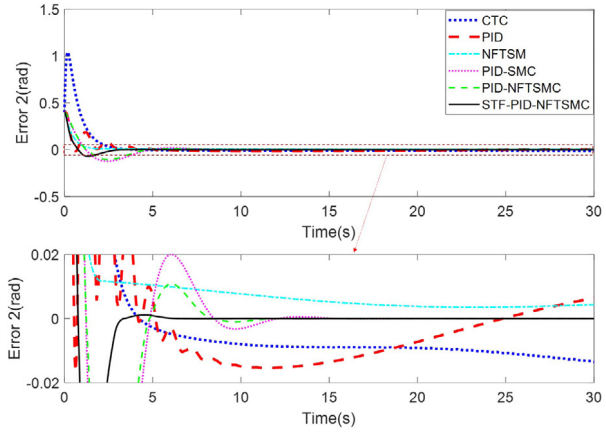


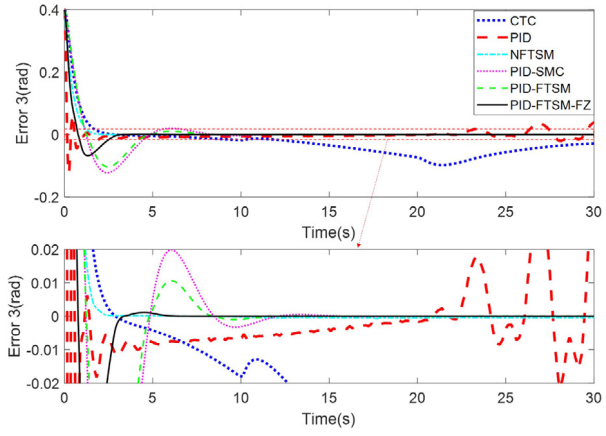
Fig. 3. Sketch of the proposed controller, i.e., STF-PID-NFTSMC and TDE.



(a) Error 1

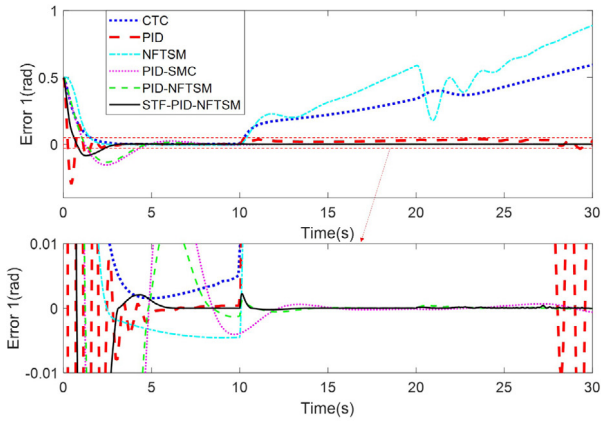


(b) Error 2

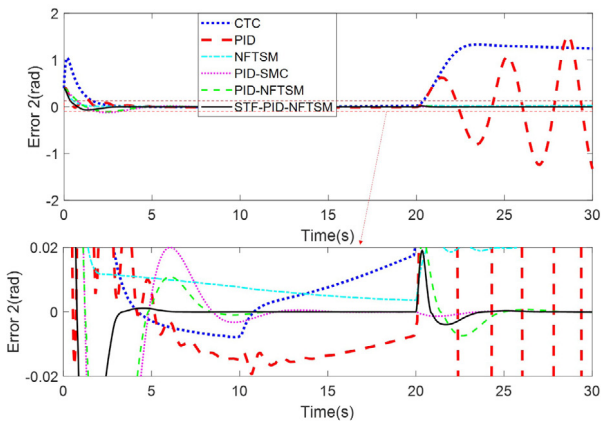


(c) Error 3

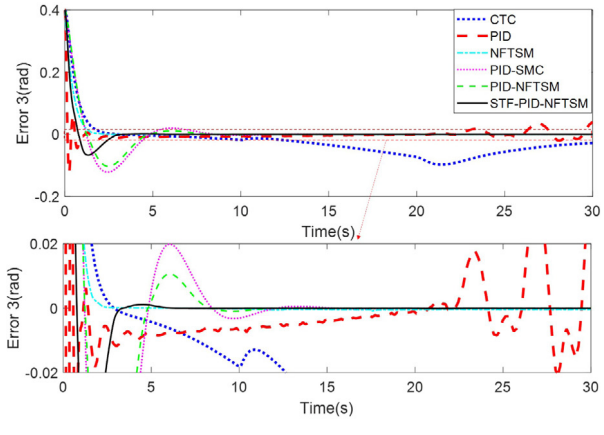
Fig. 4. Tracking errors of joints when the system in normal operation.



(a) Error 1



(b) Error 2



(c) Error 3

Fig. 5. Tracking errors of joints when the system in fault operation.

5.1. Comparisons when robot works in normal operation

The tracking errors of the controller in normal operation are shown in Fig. 4. It can be observed from Figs. 4a and 4b that the PID and the CTC provide good tracking response (fast convergence and less oscillatory) when the system is being affected by small effects of model uncertainties. However, for the system containing large uncertainties, the PID and the CTC provide unsatisfied performance, as shown in Fig. 4c. On the other hand, using sliding mode control techniques, i.e., the PID-SMC, the NFTSMC, the PID-NFTSMC and the STF-PID-NFTSMC, the system provides better tracking performance against the effects of both small and large uncertainties than that of the CTC and the PID controllers, as shown in Fig. 4. Interestingly, when the details of Fig. 4 are zoomed out, it can be observed that the steady-state error (SSE) of the PID-SMC, the PID-NFTSMC and the STF-PID-NFTSMC are much smaller than that of the NFTSMC. These results proved that the integration of the PID sliding surface into the SMC has reduced the SSE of the system significantly. In addition, one more interesting point that can be obtained from Fig. 4 is that the responses of the PID-SMC and the PID-NFTSMC are quite similar when the parameters of the PID sliding surface of these

controllers have the same value. This means that the selection of the PID gains are major and the responses of the controllers are strongly dependent on the selection of the gain values. Therefore, these parameters should be optimized. Using the fuzzy logic to automatically select the PID gains, the performance of the system is better, i.e., the performance of the STF-PID-NFTSMC is better than that of the PID-NFTSMC.

5.2. Comparisons when robot works in fault operation

In this work, the following fault function is generated into the system:

$$\phi(q,\dot{q},\tau) = \begin{bmatrix} (30\sin(q_1q_2) + 4\cos(\dot{q}_1q_2) + 15cos(\dot{q}_1\dot{q}_2)) & t \geq 10 \\ -0.8u_2, & t \geq 20 \\ 0 \end{bmatrix} \tag{42}$$

It means that the fault $30\sin(q_1q_2)+4\cos(\dot{q}_1q_2)+15cos(\dot{q}_1\dot{q}_2)$ is assumed to occur in the first actuator from time $t = 10$ s and another fault $-0.8u_2$ is existed in the second actuator from the time $t = 20$ s. The tracking errors of the controllers when the faults occur are presented in Fig. 5. For the convenience of comparisons, the root-mean-square-error (RMSE) and convergence time (CT) of the controllers are also shown in Table 3. From Fig. 5 and Table 3, it can be observed that the controllers such as the CTC, the PID, and the NFTSMC provide worse tracking performance when the faults occur; the system becomes unstable and tends to be out of control. In particular, in Figs. 5a and 5b, the controllers PID, CTC and NFTSMC provide unexpected motion of the robot from the beginning of the faults at time $t = 10$ s and time $t = 20$ s. The integration of the PID into the SMC has increased the robustness of the system significantly: the controllers PID-SMC, PID-NFTSMC and STF-PID-NFTSMC suppress the effects of the faults very quick, as shown in Fig. 5.

In addition, the comparisons between the PID-NFTSMC and the STF-PID-NFTSMC show that the self-tuning fuzzy mechanism helps to enhance the system performance, i.e., the STF-PID-NFTSMC provides less overshoot, faster transient response and faster convergence than that of the PID-NFTSMC. This is clearly demonstrated from the tracking error and convergent time reported in Table 3. The fuzzy tuning parameters are shown in Fig. 6. The results are matched with the analysis in Section 4 and can be explained as follows. From the initial position and when the faults occur at time $t = 10$ s and time $t = 20$ s, the tracking errors and the derivatives of the tracking errors are large, the gain K_p is large and the gain K_i is small to force the system to converge faster and simultaneously avoid large oscillatory. When the system converges, i.e., the error and its derivative are small, the gain K_p is small and the gain K_i is larger to maintain the small tracking error. The control efforts of the developed controller, i.e., STF-PID-NFTSMC, are reported in Fig. 7. It can be observed from Fig. 7 that the chattering is significantly reduced by using the developed control strategy.

Remark 8. The control efforts of other controllers, i.e., the CTC, the PID, the NFTSMC and the PID-based SMC, are not presented in this paper to shorten the length of the paper. However, since the proposed method, i.e., STF-PID-NFTSMC, provides higher tracking performance than other controllers, as discussed previously, and also provides continuous control inputs, we can conclude that: in this simulation analysis, with the selected parameters in Table 1, the proposed method surpasses other controllers with regard to transient response, tracking precision and chattering elimination.

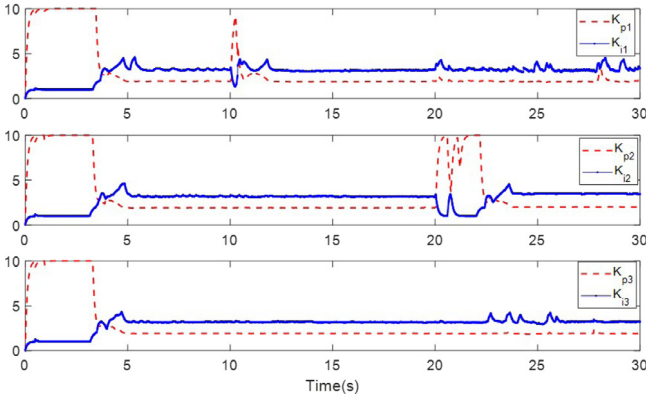


Fig. 6. Online tuning gains, K_p, K_i, using self-tuning fuzzy logic when the system in fault operation.

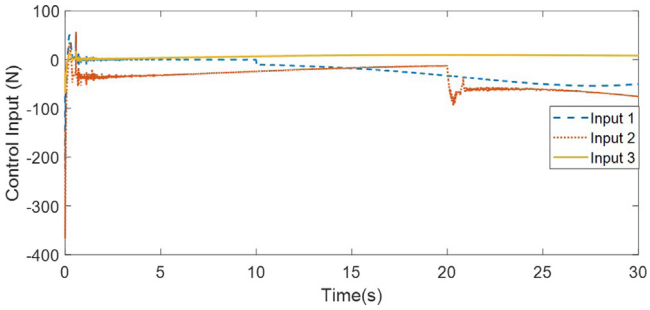


Fig. 7. Control input of the proposed controller, i.e, STF-PID-NFTSMC.

Table 1
Constructed fuzzy tuning rule.

Δk_p, Δk_i		$\lvert de(k)\rvert$				
		Z	VS	S	M	B
$\lvert e(k)\rvert$	Z	VS, B	VS, B	Z, B	Z, B	Z, B
	VS	VS, B	VS, B	VS, B	Z, M	Z, M
	S	S, M	S, M	S, M	VS, S	VS, S
	M	M, Z	M, Z	M, VS	S, VS	S,VS
	B	B, Z	B, Z	B, Z	B, Z	M, Z

Table 2
Parameter settings of the investigated controllers.

Controller	*Parameters*	*Value*
CTC	K_p, K_d	200,10
PID	K_p, K_i, K_d	200, 100, 10
PID-SMC	K_p, K_i, K_d	200, 100, 10
	Δ, ζ, ν	20, 1, 0.1
NFTSMC	k_1, k_2, λ, p, q	10, 5, 1.4, 9, 7
	Δ, ζ, ν	20, 1, 0.1
PID-NFTSMC	k_1, k_2, λ, p, q	10, 5, 1.4, 9, 7
	m_1, m_2, m_3	10, 10, 10
	c, a	0.5, 0.1
STF-PID-NFTSMC	PID-NFTSM term	Same values as the PID-NFTSMC
	The fuzzy rules	As in Table 1 and Fig. 1.
	K_{p0}, K_{i0}, β	4, 2, 1.5

Remark 9. It must be noted that Assumptions 1 and 2 may not be satisfied for abrupt faults since the changing rate of the faults is very high. However, due to the presence of the integral term in the proposed PID-NFTSM sliding surface in Eq. (10), the system can compensate the fault very quickly. Therefore, the system will have a higher probability to avoid "exploding". In fact, this is one of the major contributions of this work. The reports shown in Figs. 3 and 4 support our statement.

Table 3
Tracking errors and Convergence Time (CT) of the system when the faults occur under the inputs of the controllers.

Controller	E_1	E_2	E_3	CT_1	CT_2	CT_3
CTC	0.2363	0.3875	0.0424	3.75	3.85	4.00
PID	0.1270	0.2295	0.0396	6.12	10.50	4.80
PID-SMC	0.0306	0.0190	0.0213	8.21	8.45	11.50
NFTSMC	0.3539	0.1395	0.0116	3.72	3.20	2.10
PID-NFTSMC	0.0195	0.0164	0.0152	7.45	8.05	8.14
STF-PID-NFTSMC	0.087	0.0075	0.0067	3.51	3.52	3.10

6. Conclusions

This paper continues to enhance the performance of the fault accommodation scheme for robot manipulators found on earlier results in [16]. First, the PID-NFTSMC is reconstructed and applied for the dynamics model of robot manipulators. Then, in order to enhance the performance of the PID-NFTSMC, a self-tuning fuzzy mechanism and an approximation technique based on TDE are developed, resulting in a new control method called self-tuning fuzzy PID-NFTSMC (STF-PID-NFTSMC). Compared to the conventional PID-NFTSMC, the proposed approach improves the transient response and provides less overshoot and steady-state error since the major parameters of the PID gains are selected effectively based on a fuzzy logic system. In addition, the integration of the TDE helps to eliminate the prior knowledge requirement of the exact system dynamics and reduce the computational burden as well. The proposed approach is then compared with the conventional PID-NFTSMC and other up-to-date FTC innovation methodologies. The simulation results verify the outstanding features of the proposed strategy.

For future work it will be interesting to investigate the effects of the input saturation, output constraints of the system and the sensor faults in the design.

Acknowledgments

This work is partially supported by the EU Framework Programme for Research and Innovation H2020 under grant agreement No. 739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Van M, Ge SS, Ceglarek D. Fault estimation and accommodation for virtual sensor bias fault in image-based visual servoing using particle filter. IEEE Trans Ind Inf 2018;14(4):1312–22.
[2] Zhao B, Skjetne R, Blanke M, Dukan F. Particle filter for fault diagnosis and robust navigation of underwater robot. IEEE Trans Control Syst Technol 2014;22(6):2399–407.
[3] Gao Z, Cecati C, Ding SX. A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches. IEEE Trans Ind Electron 2015;62(6):3768–74.
[4] Jiang B, Staroswiecki M, Cocquempot V. Fault accommodation for nonlinear dynamic systems. IEEE Trans Automat Control 2006;51(9):1578–83.
[5] Shen Z, Ma Y, Song Y. Robust adaptive fault-tolerant control of mobile robots with varying center of mass. IEEE Trans Ind Electron 2018;65(3):2419–28.
[6] Zuo Z, Ho DW, Wang Y. Fault tolerant control for singular systems with actuator saturation and nonlinear perturbation. Automatica 2010;46(3):569–76.
[7] Utkin V. Sliding modes on control and optimization. Berlin, Germany: Springer-Verlag; 1992.
[8] Wang B, Zhang Y. An adaptive fault-tolerant sliding mode control allocation scheme for multirotor helicopter subject to simultaneous actuator faults. IEEE Trans Ind Electron 2018;65(5):4227–36.
[9] Zhang D, Liu G, Zhou H, Zhao W. Adaptive sliding mode fault-tolerant coordination control for four-wheel independently driven electric vehicles. IEEE Trans Ind Electron 2018;65(11):9090–100.
[10] Qiao L, Zhang W. Adaptive second-order fast nonsingular terminal sliding mode tracking control for fully actuated autonomous underwater vehicles. IEEE J Ocean Eng 2019;44(2):363–85.
[11] Van M, Ge SS, Ren H. Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control. IEEE Trans Cybern 2017;47(7):1681–93.
[12] Van M, Ge SS, Ren H. Robust fault-tolerant control for a class of second-order nonlinear systems using an adaptive third-order sliding mode control. IEEE Trans Syst Man Cybern Syst 2017;47(2):221–8.
[13] Dehkordi NM, Sadati N, Hamzeh M. A robust backstepping high-order sliding mode control strategy for grid-connected dg units with harmonic/interharmonic current compensation capability. IEEE Trans Sustain Energy 2017;8(2):561–72.
[14] Levant A. Robust exact differentiation via sliding mode technique. Automatica 1998;34(3):379–84.
[15] Van M, Mavrovouniotis M, Ge SS. An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. IEEE Trans Syst Man Cybern Syst 2019;49(7):1448–58.
[16] Van M. An enhanced robust fault tolerant control based on an adaptive fuzzy pid-nonsingular fast terminal sliding mode control for uncertain nonlinear systems. IEEE/ASME Trans Mechatronics 2018;23(3):1362–71.
[17] Basin MV, Yu P, Shtessel YB. Hypersonic missile adaptive sliding mode control using finite- and fixed-time observers. IEEE Trans Ind Electron 2018;65(1):930–41.
[18] Hoffstadt T, Maas J. Adaptive sliding-mode position control for dielectric elastomer actuators. IEEE/ASME Trans Mechatronics 2017;22(5):2241–51.
[19] Lu X, Zhang X, Zhang G, Fan J, Jia S. Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties. ISA Trans 2019;86:201–14.
[20] Salgado I, Chairez I. Adaptive unknown input estimation by sliding modes and differential neural network observer. IEEE Trans Neural Netw Learn Syst 2018;29(8):3499–509.
[21] Dass A, Srivastava S. Identification and control of dynamical systems using different architectures of recurrent fuzzy system. ISA Trans 2019;85:107–18.
[22] Chen Z, Shan C, Zhu H. Adaptive fuzzy sliding mode control algorithm for a non-affine nonlinear system. IEEE Trans Ind Inf 2007;3(4):302–11.
[23] Wang Y, Yan F, Chen J, Ju F, Chen B. A new adaptive time-delay control scheme for cable-driven manipulators. IEEE Trans Ind Inf 2019;15(6):3469–81.
[24] Baek J, Kwon W, Kim B, Han S. A widely adaptive time-delayed control and its application to robot manipulators. IEEE Trans Ind Electron 2019;66(7):5332–42.
[25] Choi HH, Yun HM, Kim Y. Implementation of evolutionary fuzzy pid speed controller for pm synchronous motor. IEEE Trans Ind Inf 2015;11(2):540–7.
[26] Duan X, Deng H, Li H. A saturation-based tuning method for fuzzy pid controller. IEEE Trans Ind Electron 2013;60(11):5177–85.
[27] Mallick N, Mukherjee V. Self-tuned fuzzy-proportional–integral compensated zero/minimum active power algorithm based dynamic voltage restorer. IET Gener Transm Distrib 2018;12(11):2778–87.
[28] Armstrong B, Khatib O, Burdick J. The explicit dynamic model and inertial parameters of the puma 560 arm. In: Proc. IEEE int. conf. robot. autom, Vol. 3. 1986, 510-18.