University of Kent

Kent Academic Repository

Zhao, Dongya, Gao, Shouli, Li, Fei, Yan, Xinggang and Spurgeon, Sarah K. (2024)
Model-free adaptive tensor product control for a class of nonlinear systems.
Control Engineering Practice, 147 . ISSN 0967-0661.

Downloaded from
https://kar.kent.ac.uk/105347/ The University of Kent's Academic Repository KAR

The version of record is available from
https://doi.org/10.1016/j.conengprac.2024.105912

This document version
Author's Accepted Manuscript

DOI for this version

Licence for this version
CC BY-NC-ND (Attribution-NonCommercial-NoDerivatives)

Additional information

Versions of research works

Versions of Record
If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts
If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in ***Title of Journal*** , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries
If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

Model-Free Adaptive Tensor Product Control for a Class of Nonlinear Systems

Dongya Zhao*[a], Shouli Gao[a], Fei Li[a], Xinggang Yan[b], Sarah K. Spurgeon[c]

[a] College of New Energy, China University of Petroleum (East China),Qingdao, 266580, Shandong, China
[b] School of Engineering, University of Kent,Canterbury, CT2 7NT, United Kingdom
[c] Department of Electronic and Electrical Engineering, University College London, London, WC1E 7JE, United Kingdom

Abstract

This paper presents a new data driven control for a class of discrete-time multiple input and multiple output nonlinear systems which uses the principles of model-free adaptive control and the tensor product. The design of a linearization model using the notion of the tensor product overcomes reset problems in the calculation of the pseudo partial derivative and also improves control performance. The stability of the proposed method is guaranteed by theoretical analysis. Numerical simulation results and an experimental trial are presented to validate the effectiveness of the proposed approach.

Keywords:
Model-free adaptive control, Tensor product, Dynamic linearization, Nonlinear systems.

1. Introduction

Modern control theory developed rapidly in the 1950s, producing many model based control methods including adaptive control, robust control and sliding mode control. From these foundations, remarkable achievements have been achieved in the applied control of industrial processes as well as, for example, automotive and aerospace systems. With the rapid development of science and technology, the scale of enterprises continues to increase, as does the complexity. This evolution provides increasing challenges in the establishment of accurate mathematical models based on physical modelling. In addition, these systems usually have the ability to produce, manipulate, store and exchange large amounts of data. The effective use of a large number of offline and/or online data and knowledge to design a model-independent data-driven controller is now the focus of significant research efforts in Hou and Wang (2013); Hassani et al. (2018); Hou et al. (2017).

Data-driven control (DDC) is a design method that only uses process data to formulate control strategies. This has greatly expanded the application of control in practice. DDC has underpinned the development of a series of control methods, which can be generally classified by the characteristics of the data used. Offline DDC uses a large amount of pre-collected data to build the system model and develops the controller on this basis. Much of the work can be done offline, making the application process computationally efficient (Silva et al., 2002; Meng et al., 2017; Roman et al., 2016). However, offline DDC requires a large quantity of data and the adaptability of the controller can be poor when the system changes. In contrast, with online DDC all model and controller parameters can be updated with real-time data. Therefore, control performance is guaranteed as process parameters are adjusted, while the controlled system is not dependent on historical data. With the rapid development of digital computing, this practice of trading computational efficiency for robustness of the controlled system is becoming more acceptable and easier to implement (Şahin and Güzeliş, 2016; Qurashi et al., 2020; Safonov and Tsao, 1997a). The iterative feedback tuning method is a model-free method that calculates the unbiased gradient of an indicator function on the controller parameters (Roman et al., 2022). Iterative learning control is aimed at a class of nonlinear dynamic systems that run repeatedly on finite intervals (He et al., 2018; Meng and Moore, 2017; Chen et al., 2020). Their application scope is limited to areas such as batch production and robots performing repetitive tasks. The unfalsified control iteratively determines the pre-constructed controller parameter set by obtaining real-time data as in Safonov and Tsao (1997a) and Wang et al. (2007). However it is difficult to generate a candidate set using only online data. Model-free adaptive control (MFAC) overcomes the limitations of the above mentioned methods. In Chi et al. (2021), the basic idea is to replace the original model with a dynamic linearized model near the trajectory of the controlled system by using a pseudo gradient vector (Hou and Jin, 2011a; Xiong et al., 2022; Hou and Xiong, 2019).

Since the MFAC method was first introduced in Hou (1994), after nearly 30 years of continuous theoretical innovation and practical application, systematic design methods have been developed. MFAC control system design and a stability proof for single-input single-output (SISO) nonlinear systems are given in Hou (1994) and Hou and Jin (2011b). Combined with the stability theory of multiple-input multiple-output (MIMO) nonlinear systems, a theo-

retically rigorous MFAC control paradigm has been formed in Xiong et al. (2022); Hou and Jin (2011a). MFAC has been widely used in applications such as power systems (Cheng et al., 2022), vehicles (Xiong et al., 2022) and robots (Xue et al., 2022).

In MFAC, a pseudo partial derivative (PPD) is introduced to simplify the dynamic linearization model structure and avoid the design of a high-order controller. Dynamic linearization is robust to the time varying nature of the system parameters, structure (Hou and Xu, 2009). For SISO and MIMO nonlinear systems, dynamic linearization techniques have been extended from compact-form dynamic linearization to partial-form dynamic linearization and full-form dynamic linearization (Hou and Jin, 2013). Further, disturbances and unknown system states are considered in the dynamic linearization process (Chi et al., 2019; Gao et al., 2023). In order not to affect the control direction, the PPD is required to be a diagonally dominant matrix and the PPD is required to be reset to its initial value if it does not satisfy conditions for diagonal dominance during an iteration. However, as more and more information is used for dynamic linearization, the order of the parameter matrix corresponding to the dynamic system becomes greater. This requires a lot of system calculations and means that more parameters need to be reset. Note that frequent resets will cause the parameter matrix to remain at the initial value for a long time, which will seriously reduce the ability of the PPD to follow the equilibrium point of the nonlinear system. This means that reset affects the adaptive ability of the controller, and even affects the tracking performance of the system.

The tensor product (TP) model transformation is a relatively new approach proposed by Baranyi and Yam when studying control of quasi linear parametric systems (Baranyi et al., 2003). Over the past decade, the TP model transformation (TPMT) has been favoured by many scholars and has been used to solve controller design problems in various practical systems, such as chaotic systems (Baranyi, 2004), translation oscillators with rotating actuators (Petres et al., 2007) and visual servos (Matuško et al., 2015). the TP based MFAC can be used to develop a new dynamic linearization approach and eliminate the need for resets. The TP function and time-varying weight coefficient replace PPD to obtain a dynamic linearized model based on the TP. Since the TP function is obtained using a real-time mesh to divide the parameter space, the iteration process is not involved, so that reset can be avoided.

According to the above analysis, TP based MFAC can better deal with nonlinear systems with unknown models. However, the parameter space for TP meshing is difficult to determine a priori. This introduces difficulties in the design of a model-free adaptive tensor product control (MFATPC). In this paper, a novel dynamic grid dividing method is developed for MFAC, where the parameter space can be computed by using the rate of change of the system output.

To design an adaptive control algorithm in the model-free TP approach, the dynamic linearization method in Hou and Xiong (2019) can be used to obtain the dynamic model. Then, the TP function is used to replace the original PPD to construct a new system parameter matrix to avoid the reset problem. To deal with the computational burden caused by the TP, high order singular value decomposition (HOSVD) is introduced to simplify the TP function. Compared to the general TP method (Baranyi, 2004), the approach is extended to unknown nonlinear systems. Moreover, compared to more general MFAC (Hou and Xiong, 2019), the proposed method avoids the reset problem and the resulting controller has strong adaptive properties.

The outline of this paper is as follows: The problem is formulated, and the basic assumptions are given in Section II. A novel dynamic linearization TP data model is developed in Section III. A MFATPC is designed in Section IV. A simulation study and experimental trial are presented in Section V. Finally, conclusions are drawn in Section VI.

2. Problem formulation

Consider a MIMO nonlinear discrete-time system described by

$$y(k+1) = F(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)) \quad (1)$$

where $u(k) = [u_1(k), u_2(k), \ldots, u_n(k)]^T \in R^n$ is the input vector and $y(k) = [y_1(k), y_2(k), \ldots, y_n(k)]^T \in R^n$ is the output vector, respectively. $F(\cdot)$ is an unknown nonlinear function. $n_y, n_u > 0$ are unknown system orders.

For the system (1), the following Assumptions are presented to make the problem tractable (Hou and Xiong, 2019).

Assumption 1: The PPD of the unknown nonlinear function $F(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u))$ with respect to the control inputs $u(k)$ are continuous.

Assumption 2: System (1) satisfies a generalized Lipschitz condition, that is, $\|\Delta y(k+1)\| < b\|\Delta u(k)\|$ for any k and $\|\Delta u(k)\| \neq 0$, where $b > 0$, $\Delta y(k+1) = y(k+1) - y(k)$, $\Delta u(k+1) = u(k+1) - u(k)$.

Remark 1: From a practical point of view, these Assumptions are reasonable and acceptable. Assumption 1 is a typical condition for many control designs, which can be satisfied by many nonlinear systems. Assumption 2 poses a limitation on the rate of change of the system output prior to the design of any control law.

Consider the MIMO nonlinear discrete-time system (1) satisfying Assumptions 1 and 2, such that system (1) can be equivalently expressed in any of the following three forms of dynamic linearization model (Hou and Jin, 2013). The three corresponding parameter matrices are referred to as the PPD, the pseudo gradient (PG) and the pseudo Jacobian matrix (PJM), respectively, which are bounded.

i) Compact-form dynamic linearization data model

$$\Delta y(k+1) = \Phi(k)\Delta u(k) \tag{2}$$

where $\Phi(k) \in R^{n\times n}$ denotes the PPD.

ii) Partial-form dynamic linearization data model

$$\Delta y(k+1) = \Phi_{p,L}(k)\Delta u_L(k) \tag{3}$$

where $\Delta u_L(k) = u_L(k) - u_L(k-1)$, L is the control input linearization length constant, $\Phi_{p,L}(k) = [\Phi_1(k) ... \Phi_L(k)] \in R^{n\times nL}$ denotes the PG, for $i = 1, 2, ..., L$, $\Phi_i(k) \in R^{n\times n}$, $u_L(k) = [u(k)^T, ..., u(k-L+1)^T]^T$.

iii) Full-form dynamic linearization data model

$$\Delta y(k+1) = \Phi_{f,L_y,L_u}(k)\Delta H_{L_y,L_u}(k) \tag{4}$$

where $\Delta H_{L_y,L_u}(k) = H_{L_y,L_u}(k) - H_{L_y,L_u}(k-1)$ and L_y and L_u are the output linearization length constant and the input linearization length constant, respectively.

$$\begin{aligned} H_{L_y,L_u}(k) &= [u(k)^T, ..., u(k-L_u+1)^T, y(k)^T, \\ &\quad ..., y(k-L_y+1)^T]^T \end{aligned} \tag{5}$$

The linearization length constants satisfy $0 \le L_y \le n_y$ and $1 \le L_u \le n_u$ and $\Phi_{f,L_y,L_u}(k) = [\Phi_1(k) ... \Phi_{L_u+L_y}(k)] \in R^{n\times n(L_u+L_y)}$ is the PJM.

Taking (2) as an example, the controller design is as follows (Hou and Jin, 2011a)

$$u(k) = u(k-1) + \frac{\lambda' \Phi(k)^T (y_d(k+1) - y(k))}{\mu' + \|\Phi(k)\|^2} \tag{6}$$

where $y_d(k+1)$ is the desired trajectory, $\mu' > 0$ is a weighting factor and $\lambda' \in (0, 2]$ is a weighting factor.

The parameter matrix $\Phi(k)$ is designed by considering the following objective function

$$\begin{aligned} J(\Phi(k)) &= \|\Delta y(k+1) - \Phi(k)\Delta u(k-1)\|_2^2 \\ &\quad + \mu \left\|\Phi(k) - \hat{\Phi}(k-1)\right\|_2^2 \end{aligned}$$

where $\mu > 0$ is a weighting factor.

Minimising this objective function yields the following estimation algorithm for $\Phi(k)$

$$\begin{aligned} \hat{\Phi}(k) &= \hat{\Phi}(k-1) \\ &\quad + \frac{\lambda\left(\Delta y(k) - \hat{\Phi}(k-1)\Delta u(k-1)\right)\Delta u(k-1)^T}{\mu + \|\Delta u(k-1)\|^2} \end{aligned} \tag{7}$$

where $\lambda \in (0, 2]$ is a weighting factor,

$$\hat{\Phi}(k) = \begin{bmatrix} \hat{\phi}_{11}(k) & \hat{\phi}_{12}(k) & \cdots & \hat{\phi}_{1n}(k) \\ \hat{\phi}_{21}(k) & \hat{\phi}_{22}(k) & \cdots & \hat{\phi}_{2n}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\phi}_{n1}(k) & \hat{\phi}_{n2}(k) & \cdots & \hat{\phi}_{nn}(k) \end{bmatrix} \in R^{n\times n}$$

In order to ensure rigour in the controller stability analysis, the following reset mechanism is required.

$$\hat{\phi}_{ii}(k) = \begin{cases} \hat{\phi}_{ii}(1) & \text{if} \quad \left|\hat{\phi}_{ii}(k)\right| < b_2^* \\ & \text{or} \quad \left|\hat{\phi}_{\text{ii}}(\text{k})\right| > ab_2^* \\ & \text{or} \quad \text{sign}\left(\hat{\phi}_{ii}(k)\right) \ne \text{sign}\left(\hat{\phi}_{ii}(1)\right) \\ \hat{\phi}_{ii}(k) & \text{else} \end{cases} \tag{8}$$

$$\hat{\phi}_{ij}(k) = \begin{cases} \hat{\phi}_{ij}(1) & \text{if} \quad \left|\hat{\phi}_{ij}(k)\right| > b_1^* \\ & \text{or} \quad \text{sign}\left(\hat{\phi}_{ij}(k)\right) \ne \text{sign}\left(\hat{\phi}_{ij}(1)\right) \\ \hat{\phi}_{ij}(k) & \text{else} \end{cases} \tag{9}$$

where $b_1^* > 0$, $a \ge 1$, $b_2^* > b_1^*(2a+1)(n-1)$, $i, j = 1, 2, ..., n$, $i \ne j$.

Remark 2: From (7) it can be seen that the system parameter matrix $\hat{\Phi}(k)$ of the MFAC is obtained by an iterative approach. In order to ensure convergence of the control algorithm, it is required that $\hat{\Phi}(k)$ satisfies a diagonal dominance condition, otherwise a reset mechanism must be adopted. The actual implementation strategy of MFAC is to solve $\hat{\Phi}(k)$ by (7) - (9) , which includes a reset mechanism, and then substitute $\hat{\Phi}(k)$ into (6) to replace $\Phi(k)$ to determine the control law.

Motivating Numerical Example:

A MIMO nonlinear system is given by Hou and Jin (2013),

$$\begin{cases} y_1(k+1) = \dfrac{2.5y_1(k)y_1(k-1) + 0.09u_1(k)u_1(k-1)}{1 + y_1^2(k) + y_1^2(k-1)} \\ \qquad + 1.2u_1(k) + 1.6u_1(k-2) \\ \qquad + 0.09u_1(k)u_2(k-1) + 0.5u_2(k) \\ \qquad + 0.7\sin(0.5(y_1(k) + y_1(k-1))) \\ \qquad \cos(0.5(y_1(k) + y_1(k-1))), \\ y_2(k+1) = \dfrac{5y_2(k)y_2(k-1)}{1 + y_1^2(k) + y_1^2(k-1)} + u_2(k) \\ \qquad + 1.1u_2(k-1) + 1.4u_2(k) + 0.5u_1(k) \end{cases}$$

The desired trajectories are given as follows

$$\begin{cases} y_{1d} = 0.5 + 0.25\cos(0.25\pi k/10) + 0.25\sin(0.5\pi k/10) \\ y_{2d} = 0.5 + 0.25\sin(0.25\pi k/10) + 0.25\sin(0.5\pi k/10) \end{cases}$$

The values used to initialise the system parameter estimation are $\phi_{111}(1) = 0.5$, $\phi_{121}(1) = 0$, $\phi_{211}(1) = 0$, $\phi_{221}(1) = 0.5$. The full details of the full-form dynamic linearization MFAC can be found in Chapter V of Hou and Jin (2013). The results of the numerical examples are shown in Figure 1 - Figure 3. As can be seen in Figure 3, $\hat{\phi}_{121}$ and $\hat{\phi}_{211}$ are frequently reset. In particular, $\hat{\phi}_{111}$ starts to reset frequently after about 500 seconds, and the control performance becomes worse, as shown in Figure 1 and 2. The integral of the time squared error (ITSE) is shown in Table 1, where $ITSE = \sum_{k=1}^{N} T^*(e(k))^2$, T^* is the sampling time, N is the end time. For clarity, steady state

data is used to compute the ITSE. It can be seen that when $\hat{\phi}_{111}$ starts to reset frequently after 500 seconds, the system performance is worse than that before 500 seconds. This demonstrates that system performance levels reduce when resets appear frequently.

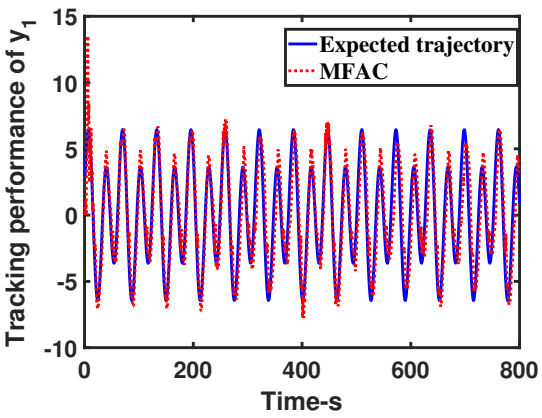


Figure 1: Tracking performance of y_1

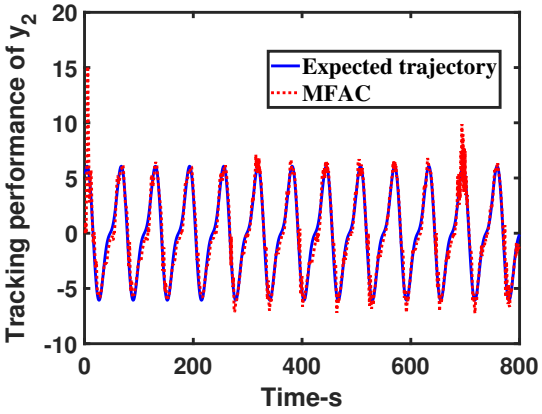


Figure 2: Tracking performance of y_2

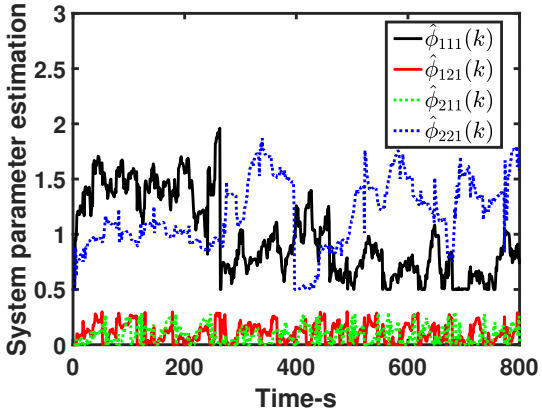


Figure 3: System parameter estimation with full-form dynamic linearization

Table 1: ITSE for MFAC

	300s-500s	500s-700s
y_1	178	661
y_2	203	1377

Remark 3: From (2) to (4), the order of the system parameter matrix increases rapidly. When the order of the system parameter matrix is high, MFAC inevitably faces more parameter iterations. More parameters mean that the reset probability in the iterative process is higher, the adaptive ability of the control system reduces and the control accuracy may be adversely affected.

Remark 4: It can be seen from (7) that the parameter matrix is mainly affected by $\Delta y(k)$. If $\Delta y(k)$ changes significantly, $\hat{\Phi}(k)$ will also change significantly, causing the value of $\hat{\Phi}(k)$ to exceed the set range and triggering the reset mechanism. Frequent resetting causes $\hat{\Phi}(k)$ to stay at the initial value for a long time, which means the dynamic linearization model may be far away from the equilibrium point, and this in turn leads to reduced control performance. The influence of the rate of output change on $\hat{\Phi}(k)$ can be reduced by increasing μ and decreasing λ to ensure that $\hat{\Phi}(k)$ reduces the reset. However, this also means that $\hat{\Phi}(k)$ will remain almost constant if the output changes little, reducing the adaptive ability of the scheme.

From the above analysis, it can be seen that MFAC adopts an iterative method to obtain the system parameter matrix, which in turn causes reset problems. The objective of this paper is to design a MFATPC for a class of unknown nonlinear systems and to solve the reset problem. The dynamic linear model (2) will be further modified to obtain a tensor product linearization model (TPLM). The system parameter matrix is obtained by mesh division, which avoids the problem of resetting caused by iteration. The TP set and the real-time weight are used to replace the system parameter matrix in the MFAC. Then, a MFATPC for nonlinear systems can be designed. Stability analysis, simulation and experimentation are carried out, providing a basis to support the development of a new MFAC.

3. Tensor product based dynamic linearization

The relationship between the rate of change of the control input and the rate of change of the output is reflected in the system parameter $\hat{\Phi}(k)$. As $u(k)$ and $y(k)$ each increase by one dimension, $\hat{\Phi}(k)$ changes from $n \times n$ to $(n+1) \times (n+1)$. This growth rate and reset mechanism makes it impossible to directly design controllers using traditional MFAC methods in some high-dimensional systems or systems where it is required to track a time-varying trajectory. The purpose of this section is to develop a TP based dynamic linearization model to replace the MFAC dynamic linear model. The system parameter matrix of the TPLM based on replaces the system parameters in

the MFAC to avoid the impact of reset mechanisms and high-dimensional data iteration processes. To ensure the computational efficiency of the system, this section also introduces a HOSVD to reduce the dimensionality of the system parameter matrix.

3.1. Tensor product linearization model

Lemma 1: Consider the MIMO nonlinear discrete-time system (1) satisfying Assumptions 1 and 2. The nonlinear system (1) at $y(k) \in \mathbb{D}$ can then be expressed by the following TPLM

$$\Delta y\left(k\right)=\sum\nolimits_{z=1}^{Z}\omega_{z}\left(p\left(k\right)\right)\Phi^{z}\Delta u\left(k\right)$$

where $\omega_{z}\left(p\left(k\right)\right)$ is a basis function, $p\left(k\right)\in R^{N}$ denotes time-varying parameters and $\Phi^{z}=\Phi\left(z\right)$ is the vertex.

Proof: In the partial format dynamic linearization data model based on (3), the system parameter matrix $\hat{\Phi}_{p,L}\left(k\right)$ is time-varying at each time instant k, so the system can be converted to a polytopic model

$$\Delta y(k)=\psi\left(p\left(k\right)\right)\Delta u\left(k\right) \tag{10}$$

where $\psi\left(p\left(k\right)\right)\triangleq\sum_{z=1}^{Z}\omega_{z}\left(p\left(k\right)\right)\Phi^{z}$ is a polytopic matrix with Φ_{z} as the vertex, $\omega_{z}\left(p\left(k\right)\right)$ as the basis function.

There are N-dimensional time-varying parameters $p(k)$. Assuming that all the system matrix parameters are in the polytopic matrix $\psi_{r}\in\{\Phi_{1},\Phi_{2},...\}$, and changing the weight r_{z} of the convex combination of the defined vertex system to the basis function $\omega_{z}\left(p\left(k\right)\right)$, the following the TP form of the parameter matrix is obtained

$$\psi\left(p\left(k\right)\right)=\sum\nolimits_{z=1}^{Z}\omega_{z}\left(p\left(k\right)\right)\Phi_{z} \tag{11}$$

Therefore, the TP form of the model is
$\Delta y\left(k\right)=\sum_{z=1}^{Z}\omega_{z}\left(p\left(k\right)\right)\Phi_{z}\Delta u\left(k\right)$. Q.E.D.

Remark 5: The TPLM is a superset of the original nonlinear system. Thus, the TPLM is valid not only for the original nonlinear system itself, but for all TPLM in the vertex convex hull. The application range shows that the model has some inherent robustness(Baranyi, 2006). The set of the TP vertices is a better description of the real system dynamics and thus can be more helpful for controller design, especially when there are disturbances and/or other interferences affecting the system.

3.2. Higher order singular value decomposition

In this section, methods are presented to design vertices and weights in the TPLM and to reduce the computation by using HOSVD.

Definition 1: (Petres, 2006) Consider the n-module product of tensor $A\in R^{J_{1}\times J_{2}\times\ldots\times J_{n}\times\ldots\times J_{N}}$ and matrix $U\in R^{I_{N}\times J_{N}}$. The n-mode matrix TP $B=A\times_{n}U$, $B\in R^{J_{1}\times J_{2}\times\ldots\times I_{n}\times\ldots\times J_{N}}$, then the element in B is

$$B_{j_{1},j_{2},\ldots,i_{n},\ldots,j_{N}}=\sum\nolimits_{j_{n}=1}^{J_{N}}A_{j_{1},j_{2},\ldots,j_{n},\ldots,j_{N}}\times U_{j_{n},i_{n}} \tag{12}$$

Lemma 2: (De Lathauwer et al., 2000) If the HOSVD is established according Appendix A. The discarded V_{n}^{d} contains non zero singular values $\sigma_{I_{n}+1}^{(n)}$, ... , $\sigma_{J_{n}}^{(n)}$, the upper bound of the approximation error is

$$\|\ \hat{\Phi}\left(k\right)-\psi^{hosvd}\ \|\leq\sum_{n=1}^{N}\left(\sum_{i_{n}=I_{n}+1}^{J_{n}}\left(\sigma_{i_{n}}^{(n)}\right)^{2}\right)$$

where ψ^{hosvd} is given in Appendix A.

The TPLM is obtained by following the steps in Appendix A, and the system parameter matrix (11) can then be rewritten as the following TPLM representation

$$\psi(p(k))=\psi\otimes_{n=1}^{N}\omega_{z}\left(p_{n}\right) \tag{13}$$

where ψ is a tensor of order $n+2$ and $\omega_{z}\left(p_{n}\right)$ is a weight function. $\otimes$ represents the Kronecker product. More details can be found in Appendix A.

Remark 6: In general, it is complex and difficult to obtain the linearized representation of an unknown nonlinear system using analytical methods. However, MFATPC provides an efficient solution based on numerical calculations. The combined dynamic linearization and TPLM approach is applicable to a wide range of nonlinear systems and is highly versatile.

3.3. Parameter matrix design

Expand (7) according to time k

$$\begin{aligned}\hat{\Phi}\left(k\right)&=\Delta y\left(k\right)\Gamma(k-1)+\hat{\Phi}\left(k-1\right)\\&\quad-\hat{\Phi}\left(k-1\right)\Delta u\left(k-1\right)\Gamma(k-1)\\&=\Delta y\left(k\right)\Gamma(k-1)\\&\quad+\hat{\Phi}\left(k-1\right)\left(I-\Delta u\left(k-1\right)\Gamma(k-1)\right)\end{aligned} \tag{14}$$

where $\Gamma(k)=\frac{\lambda\Delta u(k)^{T}}{\mu+\|\Delta u(k)\|^{2}}$.

Expand $\hat{\Phi}\left(k-1\right)$ in formula (14) to obtain

$$\begin{aligned}\hat{\Phi}\left(k\right)&=\Delta y\left(k\right)\Gamma(k-1)\\&\quad+\Delta y\left(k-1\right)\Gamma(k-2)(I-\Delta u\left(k-1\right)\Gamma(k-1))\\&\quad+\hat{\Phi}\left(k-2\right)\prod_{i=1}^{2}\left(I-\Delta u\left(k-i\right)\Gamma(k-i)\right)\end{aligned} \tag{15}$$

Then, (15) can be rewritten as

$$\begin{aligned}\hat{\Phi}\left(k\right)&=\Delta y\left(k\right)\Gamma(k-1)+\hat{\Phi}\left(1\right)\prod_{i=1}^{k-1}\left(I-\Delta u\left(i\right)\Gamma(i)\right)\\&\quad+\sum_{i=1}^{k-1}\prod_{j=k-2}^{i}\left(I-\Delta u\left(j\right)\Gamma(j)\right)\Delta y(i+1)\Gamma(i)\end{aligned} \tag{16}$$

Then, all the models and weights of the sample points can be obtained using the method in Appendix A.

Remark 7: If the control target suddenly changes or is disturbed, the system parameters in MFAC will change beyond the predetermined range and trigger the reset mechanism. Further, for systems with rapidly changing output trajectories, the parameter $\hat{\Phi}(k)$ in the corresponding MFAC remains at the initial value for a long time due to the reset, seriously affecting the control performance of the system.

In contrast to the MFAC parameter matrix (7), the new parameter matrix (16) is simply expanded based on the values of $\Delta y(k)$ and $\Delta u(k)$ at time k, and the weights are obtained by interpolation. The design method in this paper does not involve iterative calculations to avoid the reset phenomenon in (8) and (9).

4. Model-free adaptive tensor product control

In this section, the MFATPC is developed. The controller design and stability analysis follows.

First set the time-varying parameters corresponding to $\Delta y(k)$ and $\Delta u(k)$ at time k to p_n. The MFATPC is designed based on the following objective function

$$\begin{aligned} J\left(u\left(k\right)\right) &= \left\|y_d\left(k+1\right) - y\left(k+1\right)\right\|_2^2 \\ &\quad + \mu' \left\|u\left(k\right) - u\left(k-1\right)\right\|_2^2 \end{aligned} \tag{17}$$

where $\mu' > 0$ is a weighting factor.

Substituting the TPLM into the objective function (17) yields the following control law

$$\begin{cases} u_z\left(k\right) = \dfrac{\lambda' \psi_z(p(k)) \left(y_d\left(k+1\right) - y\left(k\right)\right)}{\mu' + \left\|\psi_z(p(k))\right\|^2} \\ u\left(k\right) = u\left(k-1\right) + \displaystyle\sum_{z=1}^{Z} \omega_z\left(p\left(k\right)\right) u_z\left(k\right) \end{cases} \tag{18}$$

where $\lambda' \in (0, 1]$ is a weighting factor, $u_z\left(k\right)$ is the control corresponding to the zth vertex system, and Z is the maximum number of vertices reserved.

Remark 8: The implementation strategy of MFATPC is to first obtain the system parameters of the TP through the steps in Appendix A, and then substitute them into (18) to obtain the control law. Compared to traditional MFAC (6), the method (18) proposed in this paper is a combination of a series of linear systems, where $\psi(p(k))$ is a TPLM coefficient which avoids the reset phenomenon.

Theorem 1: For system (14), under Assumptions 1-2, considering Lemma 2, if the feedback control is designed as (18), $\psi(p(k))$ are estimated from steps 1-4 in Appendix A, then $\tilde{\Phi}(k)$ and $e(k)$ is bounded.

Proof: Define the PPD estimation error as

$$\tilde{\Phi}(k) = \Phi(k) - \psi(p(k)) \tag{19}$$

Define the system parameter estimation error as

$$\tilde{\psi}(p(k)) = \psi(p(k)) - \hat{\Phi}(k) \tag{20}$$

Subtract $\Phi(k)$ from both sides of the parameter estimation algorithm (7)

$$\begin{aligned} \tilde{\Phi}^*(k) &= \tilde{\Phi}^*(k-1) + \Delta\Phi(k) \\ &\quad - \frac{\lambda \hat{\Phi}(k-1) \Delta u\left(k-1\right) \Delta u\left(k-1\right)^T}{\mu + \left\|\Delta u\left(k-1\right)\right\|^2} \end{aligned} \tag{21}$$

where $\tilde{\Phi}^*(k) = \hat{\Phi}(k) - \Phi(k)$, $\Delta\Phi(k) = \Phi(k) - \Phi(k-1)$, $\tilde{\Phi}(k) = [\tilde{\phi}_1^T(k), ..., \tilde{\phi}_n^T(k)]^T$, $\tilde{\phi}_i(k) = [\tilde{\phi}_{i1}(k), ..., \tilde{\phi}_{in}(k)]$ for $i = 1, 2, ..., n$ and $\Phi(k) = [\phi_1(k)^T, ..., \phi_n(k)^T]^T$, $\phi_i(k) = [\phi_{i1}(k), ..., \phi_{in}(k)]$ for $i = 1, 2, ..., n$.

The parameter estimation error can be rewritten as

$$\begin{aligned} \tilde{\phi}_i^*(k) &= \tilde{\phi}_i^*(k-1) + \Delta\phi_i(k) \\ &\quad - \frac{\lambda \tilde{\phi}_i(k-1) \Delta u\left(k-1\right) \Delta u\left(k-1\right)^T}{\mu + \left\|\Delta u\left(k-1\right)\right\|^2} \end{aligned} \tag{22}$$

Substitute $\Gamma(k)$ into (22) and take the norm

$$\begin{aligned} &\left\|\tilde{\phi}_i^*(k)\right\| \\ &= \left\|\tilde{\phi}_i^*(k-1) + \Delta\phi_i(k) - \tilde{\phi}_i^*(k-1) \Delta u\left(k-1\right) \Gamma(k-1)\right\| \\ &\leq \left\|\tilde{\phi}_i^*(k-1) \left(I - \Delta u\left(k-1\right) \Gamma(k-1)\right)\right\| + \left\|\Delta\phi_i(k)\right\| \\ &\leq \left\|\tilde{\phi}_i^*\left(k-1\right) \left(I - \Delta u\left(k-1\right) \Gamma(k-1)\right)\right\| + 2b \end{aligned} \tag{23}$$

Squaring the first term on the right hand side of (23) gives

$$\begin{aligned} &\left\|\tilde{\phi}_i^*(k-1) \left(I - \Delta u\left(k-1\right) \Gamma(k-1)\right)\right\|^2 \\ &= \left\|\tilde{\phi}_i^*(k-1)\right\|^2 \left\|\left(I - \Delta u\left(k-1\right) \Gamma(k-1)\right)\right\|^2 \\ &= \left\|\tilde{\phi}_i^*(k-1)\right\|^2 + \left\|\tilde{\phi}_i^*(k-1)\right\|^2 \left\|\Delta u\left(k-1\right) \Gamma(k-1)\right\|^2 \\ &\quad - 2\left\|\tilde{\phi}_i^*(k-1)\right\|^2 \Delta u\left(k-1\right) \Gamma(k-1) \\ &= \left\|\tilde{\phi}_i^*(k-1)\right\|^2 + \left(-2 + \Delta u\left(k-1\right) \Gamma(k-1)\right) \\ &\quad \times \left\|\tilde{\phi}_i^*(k-1)\right\|^2 \Delta u\left(k-1\right) \Gamma(k-1) \\ &= \left\|\tilde{\phi}_i^*(k-1)\right\|^2 + \left(-2 + \Delta u\left(k-1\right) \Gamma(k-1)\right) \\ &\quad \times \frac{\lambda \left\|\tilde{\phi}_i^*(k-1) \Delta u\left(k-1\right)\right\|^2}{\mu + \left\|\Delta u\left(k\right)\right\|^2} \end{aligned} \tag{24}$$

Since $0 < \lambda \leq 2$ and $\mu > 0$, the following inequality holds

$$\begin{aligned} &-2 + \left\|\Delta u\left(k-1\right) \Gamma(k-1)\right\| = \\ &-2 + \frac{\lambda \left\|\Delta u\left(k-1\right) \Delta u(k-1)^T\right\|}{\mu + \left\|\Delta u\left(k-1\right)\right\|^2} \\ &< 0 \end{aligned} \tag{25}$$

Combining (24) and (25) implies that there exists a constant $0 < d_1 < 1$ such that

$$\left\|\tilde{\phi}_i^*(k-1) \left(I - \Delta u\left(k-1\right) \Gamma(k-1)\right)\right\| \leq d_1 \left\|\tilde{\phi}_i^*(k-1)\right\| \tag{26}$$

Substituting (26) into (23) yields

$$\begin{aligned} \left\|\tilde{\phi}_i^*(k)\right\| &\leq d_1 \left\|\tilde{\phi}_i^*(k-1)\right\| + 2b \\ &\leq d_1 \left\|\tilde{\phi}_i^*(k-2)\right\| + 2d_1 b + 2b \\ &\leq d_1^{k-1} \left\|\tilde{\phi}_i^*(1)\right\| + \frac{2b\left(1 - d_1^{k-1}\right)}{1 - d_1} \end{aligned} \tag{27}$$

Since $0 < d_1 < 1$, $d_1^{k-1} \to 0$ when $k \to \infty$. Then

$$\left\|\tilde{\phi}_i^*(k)\right\| \leq \frac{2b}{1-d_1} \tag{28}$$

Since $\tilde{\phi}_i^*(k)$ is bounded, $\tilde{\Phi}^*(k)$ is bounded. Combining (20), the system parameter error is

$$\begin{aligned}\left\|\tilde{\Phi}(k)\right\| &= \|\Phi(k) - \psi(p(k))\| \\ &\leq \left\|\Phi(k) - \hat{\Phi}(k)\right\| + \left\|\psi(p(k)) - \hat{\Phi}(k)\right\| \\ &\leq \frac{2b}{1-d_1} + \left\|\psi(p(k)) - \hat{\Phi}(k)\right\|\end{aligned} \tag{29}$$

According to Lemma 2, $\|\psi(p(k)) - \hat{\Phi}(k)\|$ is bounded. It follows that the errors in the system parameters $\tilde{\Phi}(k)$ are bounded.

Substituting the TPLM (14) and controller algorithm (18) into the tracking error equation yields

$$e(k+1) = e(k) - S(k) \otimes_{n=1}^{N} \omega_n(p_n(k))\Delta u(k) \tag{30}$$

Since $\sum_{n=1}^{N} \omega_n(p(k)) = 1$, (30) can be written as

$$e(k+1) = \left(I - \frac{\lambda'\psi(p(k))\psi(p(k))^T}{\mu'\|\psi(p(k))\|^2}\right) e(k) \tag{31}$$

Since $0 < \lambda' \leq 2$ and $\mu' > 0$, the following inequality holds

$$0 < \left\| I - \frac{\lambda'\psi(p(k))\psi(p(k))^T}{\mu'\|\psi(p(k))\|^2} \right\| < 1 \tag{32}$$

Let $d_2 = \left\| I - \frac{\lambda'\psi(p(k))\psi(p(k))^T}{\mu'\|\psi(p(k))\|^2} \right\|$. Taking the norm of both sides of (20) yields

$$\begin{aligned}\|e(k+1)\| &\leq \left\| I - \frac{\lambda'\psi(p(k))\psi(p(k))^T}{\mu'\|\psi(p(k))\|^2} \right\| \|e(k)\| \\ &\leq d_2^k \|e(1)\|\end{aligned} \tag{33}$$

Since $0 < d_2 < 1$, $d_2^k \to 0$ when $k \to \infty$. Then $e(k)$ is bounded. Thus, $\tilde{\Phi}(k)$ and $e(k)$ are bounded. Q.E.D.

Remark 9: Compared with MFAC (Hou and Jin, 2013; Hou and Xiong, 2019), the dynamic linearization model in this paper is further transformed. The original system parameter matrix is not obtained from an iterative calculation but from a parameter matrix sequence and its weighted combination. Since the parameter matrix sequence and the weight are updated in real time and do not involve an iterative process, no reset is triggered and the adaptive capability is improved. Compared with traditional TP control, the TPLM is extended from linear time-varying systems to unknown nonlinear systems, and its application scope is broadened.

Remark 10: For the proposed control method, it can be obtained from Eq. (30) that

$$\begin{aligned}e(k+1) &= \left(I - \frac{\lambda'\psi(p(k))\psi(p(k))^T}{\mu'\|\psi(p(k))\|^2}\right) e(k) \\ &= K_p e(k)\end{aligned}$$

where $K_p = I - \frac{\lambda'\psi(p(k))\psi(p(k))^T}{\mu'\|\psi(p(k))\|^2}$.

Therefore, the controller parameters in this paper can be adjusted with the help of a proportional parameter adjustment method used in PID control. For the full-form dynamic linearization data model, the controller becomes a complete PID controller form. More details are given in Section III of Hou and Xiong (2019).

5. Simulation and experimental verification

5.1. SIMULATION

Case 1: Example of a general numerical simulation

To validate the proposed approach, comparisons are made between the proposed control and a classical MFAC scheme (Hou and Xiong, 2019).

A MIMO nonlinear system is considered as given by Hou and Jin (2013)

$$\begin{cases} x_{11}(k+1) = \frac{x_{11}^2(k)}{1+x_{11}^2(k)} + 0.3x_{12}(k) \\ x_{12}(k+1) = \frac{x_{11}^2(k)}{1+x_{12}^2(k)+x_{21}^2(k)+x_{22}^2(k)} + u_1(k) \\ x_{21}(k+1) = \frac{x_{21}^2(k)}{1+x_{21}^2(k)} + 0.2x_{22}(k) \\ x_{22}(k+1) = \frac{x_{21}^2(k)}{1+x_{11}^2(k)+x_{12}^2(k)+x_{22}^2(k)} + u_2(k) \\ y_1(k) = x_{11}(k) \\ y_2(k) = x_{21}(k) \end{cases}$$

The desired trajectories are given as follows

$$\begin{cases} x_{11d} = 0.5 + 0.25\cos(0.25\pi k/10) + 0.25\sin(0.5\pi k/10) \\ x_{21d} = 0.5 + 0.25\sin(0.25\pi k/10) + 0.25\sin(0.5\pi k/10) \end{cases}$$

The initial conditions are given as $x_{11}(0) = x_{21}(0) = 0.5$, $x_{12}(0) = x_{22}(0) = 0$, $u(0) = 0$. Approximating the original system on the parameter space $[y_1(k-1) + 0.5\Delta y_n(k), y_1(k+1)] \times ... \times [y_n(k) + 0.5\Delta y_1(k), y_n(k+1)]$ at time k.

The controller parameters are listed in Table 2.

Table 2: Values of the controller parameters

MFATPC	$\mu = 1\ \lambda' = 1\ \mu' = 0.01$
MFAC	$\mu = 1\ \lambda' = 1\ \mu' = 0.01\ \lambda = 1$

Table 3: ITSE for the two methods

	x_{11}	x_{21}
MFATPC	2.9856	5.7666
MFAC	4.2016	12.5389

The simulation results show that the MFATPC algorithm can effectively track systems with fast trajectory changes. Figure 4 - Figure 5 show the tracking performance. Figure 6 shows the system parameter estimate

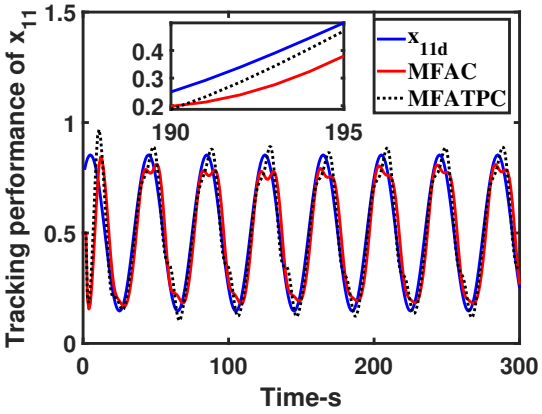


Figure 4: Tracking performance of x_{11}

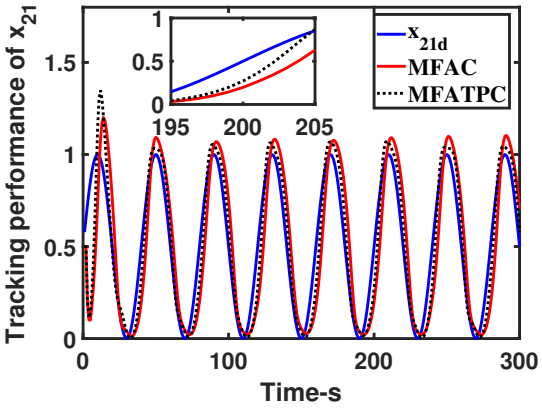


Figure 5: Tracking performance of x_{21}

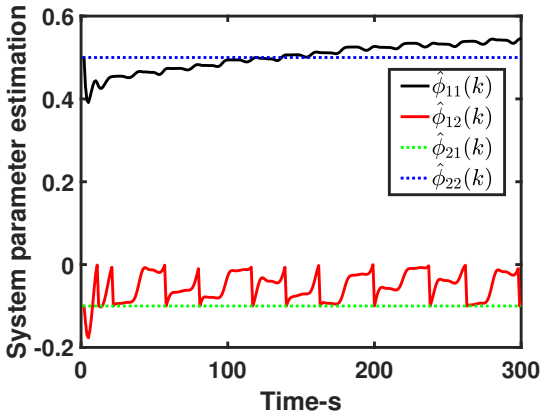


Figure 6: System parameter estimation with MFAC

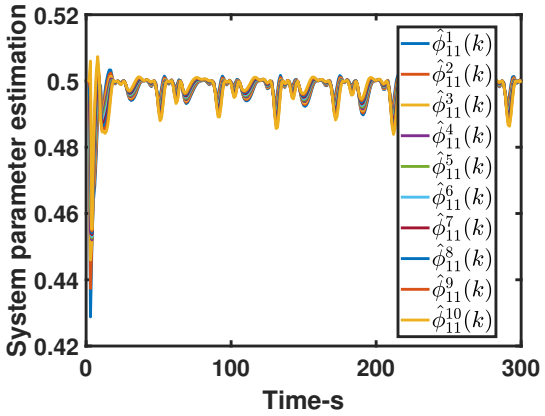


Figure 7: System parameter estimation ϕ_{11} with MFATPC

using MFAC. Figure 7 shows the system parameter estimate ϕ_{11} using MFATPC. To save space, take ϕ_{11} as an example for presentation. Figure 8 shows the corresponding control signals. From Figure 4 and Figure 6 it can be seen that $\hat{\Phi}_{21}(k)$ and $\hat{\Phi}_{22}(k)$ remain at their initial value and lose their ability to adapt using MFAC. When the direction of the tracking trajectory changes, $\hat{\Phi}_{12}(k)$ is reset periodically, and the corresponding MFAC has worse tracking performance. Compared to Figure 6, Figure 7 shows that the method proposed in this paper only expands around the initial point and does not require a reset process. Further there are no significant fluctuations in the system parameters that would adversely affect system performance. The ITSE for the MFATPC and MFAC are given in Table 3. It can be seen that the ITSE for the MFATPC is much smaller than that for the MFAC case. The simulation results further confirm the effectiveness of the proposed method.

Case 2: Continuous stirred tank reactor (CSTR) simulation

The dimensionless discrete dynamics model of the CSTR system (Zhao et al., 2015) is as follows

$$\begin{cases} x_1(k+1) = T\left(-a_1 x_1(k) + a_5(1-x_1(k))e^{\left(\frac{x_2(k)}{1+x_2(k)/a_4}\right)}\right) + x_1(k) \\ x_2(k+1) = T(-a_1 x_1(k) + a_2 * a_5(1-x_1(k))e^{\left(\frac{x_2(k)}{1+x_2(k)/a_4}\right)} \\ \qquad + a_3(u(k) - x_2(k))) + x_2(k) \\ y(k) = x_2(k) \end{cases}$$

where $x_1(k)$ and $x_2(k)$ are state variables representing dimensionless concentrations and temperatures, respectively, $y(k)$ is the system output and $u(k)$ is the control input. The parameters can be taken as $T = 0.1$, $a_1 = 1.0$, $a_2 = 8.0$, $a_3 = 0.3$, $a_4 = 20.0$, $a_5 = 0.072$. The desired trajectory is $y_d = 4$.

The initial conditions are given as $x_1(0) = 0.5$, $x_2(0) = 3$, $u(0) = 0$. The original system is approximated on the parameter space $[y_1(k-1) + 0.6\Delta y_n(k), y_1(k+1)] \times ... \times [y_n(k) + 0.6\Delta y_1(k), y_n(k+1)]$ at time k.

The controller parameters are listed in Table 4.

Table 4: Values of the controller parameters

MFATPC	$\mu = 0.4\ \lambda' = 1\ \mu' = 0.5$
MFAC	$\mu = 0.4\ \lambda' = 1\ \mu' = 0.5\ \lambda = 0.2$

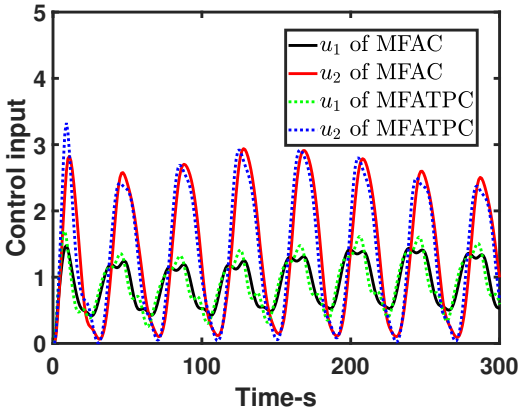


Figure 8: Time response of control inputs

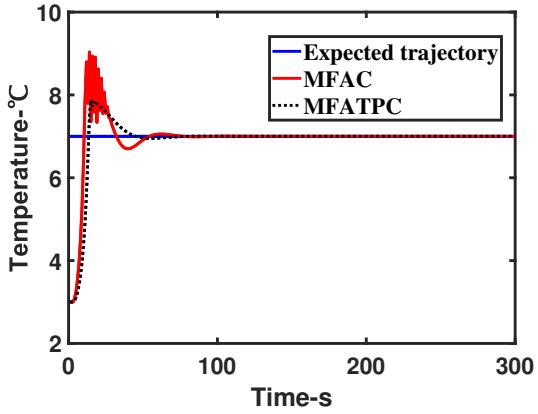


Figure 10: The temperature of the CSTR

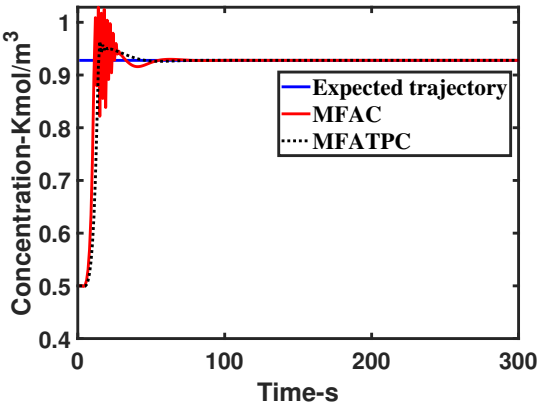


Figure 9: The concentration of the CSTR

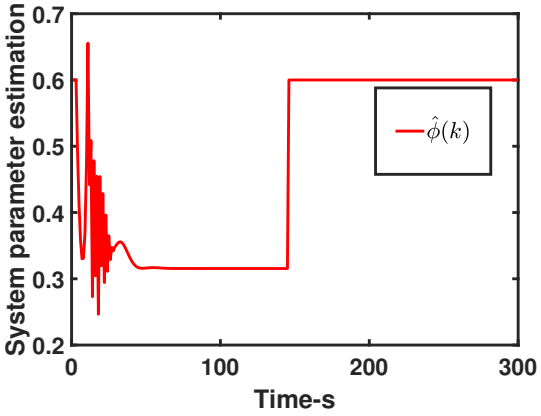


Figure 11: System parameter estimation with MFAC

The simulation results show that MFATPC and MFAC have similar tracking performance. However MFAC has to reset the parameter estimates which may affect the adaptive ability. Figure 9 - Figure 10 show the tracking performance. Figure 11 shows the system parameter estimate using MFAC. Figure 12 shows the system parameter estimate ϕ using MFATPC. Figure 13 shows the corresponding control signals. From Figure 9, Figure10 and Figure11, it can be seen that $\hat{\Phi}(k)$ in the MFAC algorithm is frequently reset at the beginning compared to the method proposed in this paper, which may affect its adaptive ability and the tracking performance. From Figure 12, it can be seen that the method proposed in this paper does not require any reset process which ensures the control signal is smoother.

5.2. EXPERIMENTAL TRIAL

In this subsection, an experimental platform belonging to the research team of China University of Petroleum (East China) is used to verify the algorithm. For details, see Zhang et al. (2021) and Gao et al. (2023). The initial flow rate of the raw material tank is 10L/h, and after running for 30 minutes, it increases to 15L/h. The temperature of the raw material tank is 17.6 °C. The desired temperature of the reactor is set at 31 °C for ten minutes and then set to 33 °C. Both experiments were switched to automatic control from an initial temperature of 29°C. The parameters of the MFATPC, MFAC and PID controllers are given in Table 5.

The experimental results are shown in Figure 14 - Figure 18 and the ITSE results are shown in Table 6. It can be seen that the proposed method ensures that the temperature converges to the desired trajectory. Compared with MFAC and PID, the proposed method not only has a smaller overshoot but also a smaller steady state error. It can be seen from Figure 16 and Figure 17 that the MFAC has been undergoing reset as compared to the method proposed in this paper. Consideration of the ITSE also reflects the better control performance of the proposed method. Experimental results further verify the effectiveness of the proposed method.

Remark 11: Comparing the performance between matlab simulation and experimental trial, it is clear that the performances are similar. The proposed approach has better performance than that of the MFAC. Further, the proposed approach does not require any reset in the implementation.

Remark 12: Compared to the model-based distributed control approach in Zhang et al. (2021), the proposed ap-

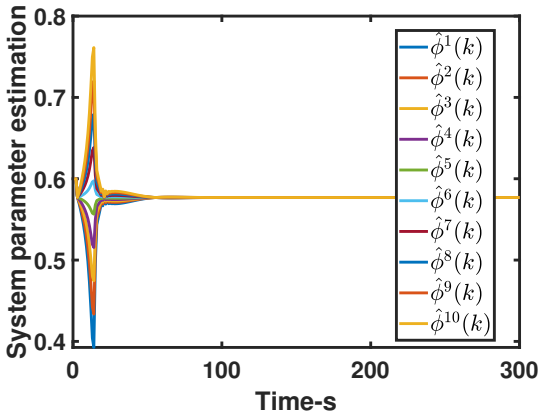


Figure 12: System parameter estimation ϕ with MFATPC

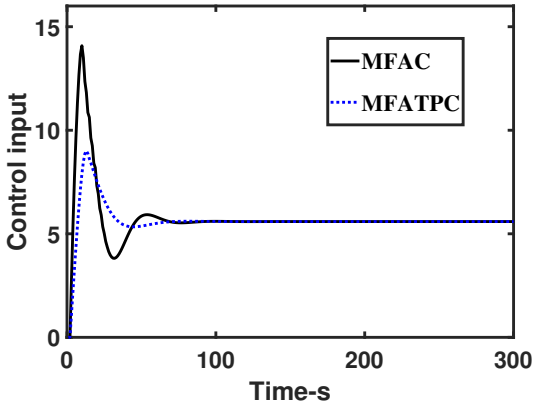


Figure 13: Time response of control inputs

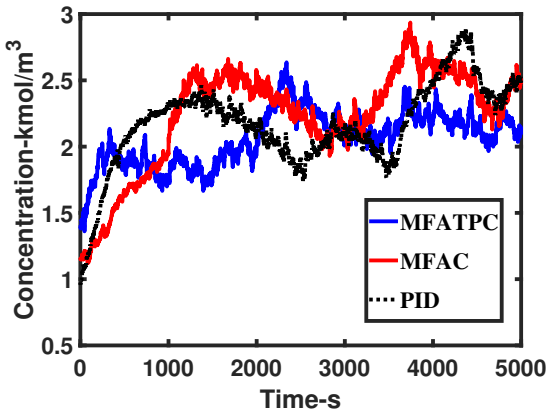


Figure 14: The concentration of CSTR

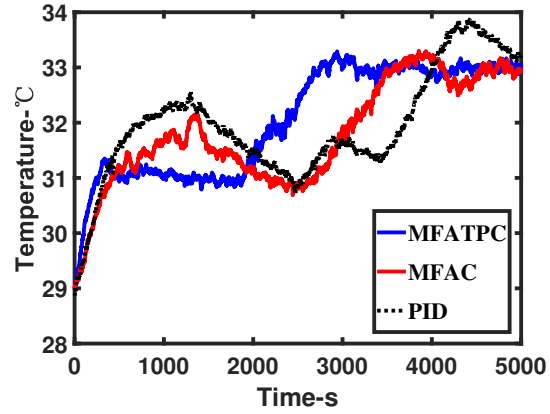


Figure 15: The temperature of the CSTR

proach is a model-free control approach. Compared to the state feedback based model-free control method in Gao et al. (2023), the proposed approach improves the dynamic linearised model to avoid the reset phenomenon.

Table 5: Values of the controller parameters

	Values of the controller parameters
MFATPC	$\mu = 0.01$ $\lambda' = 0.2$ $\mu' = 10$
MFAC	$\mu = 0.01$ $\lambda' = 0.2$ $\mu' = 10$ $\lambda = 1$
PID	$K_P = 25$ $K_D = 0$ $K_I = 0.01$

Table 6: ITSE for the two methods

	Temperature
MFATPC	456.54
MFAC	2369.4
PID	3245.8

6. Conclusion

This paper presents a class of MFATPC for discrete-time MIMO nonlinear systems. Compared with the traditional model-free adaptive control technique, the method transforms the original dynamic linearized model into a TPLM by establishing a new TP based global linearization method. At the same time, the parameter matrix of TPLM is solved by tensor expansion to avoid the reset phenomenon in MFAC and to improve the adaptive capability of the control. Finally, simulation and experimental results demonstrate the superiority of the method. In industrial implementation, the control parameters can be tuned according to the PID parameter tuning method. Future research will focus on the effect of noise in the data.

Acknowledgments

This work is supported by the National Nature Science Foundation of China under grant no. 61973315 and no. 61473312.

Appendix A. The steps to obtain the tensor product model.

This section converts dynamic linear systems into TPLM . Before discussing the details of the proposed method,

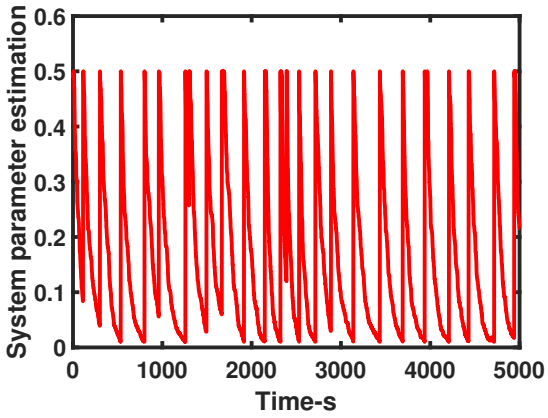


Figure 16: System parameter estimation by MFAC

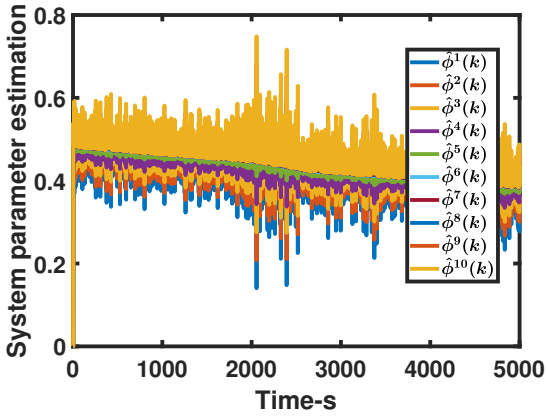


Figure 17: System parameter estimation ϕ by MFATPC

the basics of HOSVD and its regularisation are briefly reviewed for readability.

Lemma 3: (Baranyi, 2004) (HOSVD) Every tensor $\mathcal{A} \in R^{I_1 \times I_2 \times \cdots I_N}$ can be written as the product

$$\mathcal{A} = S \otimes_{n=1}^{N} V_n = S \times_1 V_1 \times_2 V_2 \times_3 ... \times_N V_N$$

where $V_n = [v_{1,n} v_{2,n} \cdots v_{I_N,n}] \in R^{I_N \times I_N}$ is a unitary matrix called the n-mode singular matrix, $S \in R^{I_1 \times I_2 \times \cdots I_N}$ is termed the core tensor and $S \times_n V_n$ is the n-mode matrix TP.

Lemma 4: (Baranyi, 2004) Matrix V_n can always be transformed to matrix $\overline{V}_n$ subject to

$$S \otimes_{n=1}^{N} V_n = \overline{S} \otimes_{n=1}^{N} \overline{V}_n$$

where notation $\overline{V}_n$ is a normalized representation of V_n.

Step 1) Sampling the given function

The system parameter matrix is transformed into a tensor form by meshing. The sampling mesh ensures the effectiveness of the controller design for the original nonlinear system.

Define a rectangular sampling mesh of $J_1 \times J_2 \times ... \times J_N$ on the region $\mathbb{D} = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_N, b_N]$. Calculate

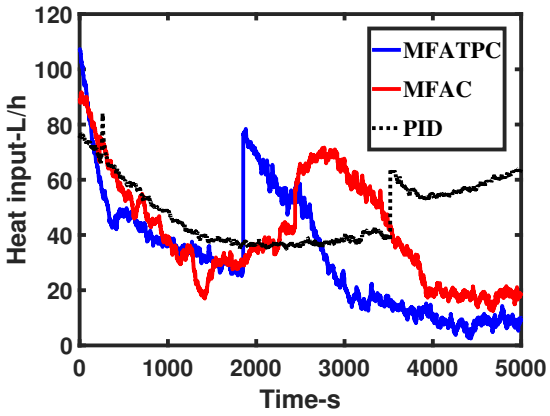


Figure 18: Time response of control inputs

the value of $\psi(p(k))$ at each sampling point $p_{j_1,...,j_N}$ and store it in a tensor ψ^s of order $N+2$ of $J_1 \times J_2 \times ... \times J_N \times n \times n$, then the element in ψ^s is

$$\psi^s(j_1, j_2, .., j_n, :, :) = \psi(p_{j_1,...,j_N}) \tag{A.1}$$

Thus, the TP model takes the form of

$$\Delta y(k+1) = \sum_{n=1}^{N} \omega_n(p(k)) \psi(p(k)) \Delta u(k) \tag{A.2}$$

where each $\psi(p(k))$ represents a linear model.

Step 2) Extracting the basis model

The number of models divided by meshes is often large. To improve computational efficiency, HOSVD is needed to extract the basis model. The n-module product and the corresponding weight matrix are obtained by higher order singular value decomposition.

The HOSVD algorithm is executed only in the $1-N$ dimensions of the sampling tensor.

$$\psi^s = \psi^* \otimes_{n=1}^{N} [V_n V_n^d] \tag{A.3}$$

where ψ^* is the tensor before the higher order singular value decomposition, the size of matrix V_n is the matrix consisting of the singular value vectors corresponding to the singular values retained when the singular value decomposition is performed, V_n^d is the matrix consisting of the singular value vectors corresponding to the singular values discarded when performing the singular value decomposition.

Retaining I_n number of singular values on the dimensions and discarding the rest with their corresponding singular vectors stored in V_n^d yields

$$\psi^s \approx \psi^{hosvd} \otimes_{n=1}^{N} V_n^{hosvd} \tag{A.4}$$

where ψ^{hosvd}, V_n^{hosvd} are obtained by ψ^* and V_n conversion respectively.

Step 3) Basis model normalization

In order to satisfy the summation normalisation condition (A.4), a further transformation must be executed.

Let $\sum V_n^{hosvd}$ denote the row sum of $(V_n^{hosvd})^T$. If $\sum V_n^{hosvd}$ is zero, then the auxiliary matrix

$$\psi_{n1} = \mathrm{diag}(\sum V_n^{hosvd})$$

otherwise

$$\psi_{n1} = I_{I_n \times I_n} + [\mathbf{0}_{I_n \times (\widetilde{I}-1)} \quad \Sigma(V_n)^T - \mathbf{1}_{I_n} \quad \mathbf{0}_{I_n \times (I_n - \widetilde{I})}]$$

where $\widetilde{I}$ denotes the $\widetilde{I}th$ element of the matrix. If $\sum V_n^{hosvd}$ is zero, then auxiliary matrix $\psi_{n2} = V_n^{hosvd}\psi_{n1}$, otherwise $\psi_{n2} = [V_n^{hosvd} \quad V_n^{d,hosvd} \sum V_n^{d,hosvd}] \begin{bmatrix} \psi_{n1} & 0 \\ 0 & 1 \end{bmatrix}$.

Let $\min \psi_{\mathrm{n2}}$ represent the minimum value of all elements in ψ_{n2}. If $\min \psi_{\mathrm{n2}} < -1$,then $\epsilon_n = \frac{1}{|\min \psi_{\mathrm{n2}}|}$, otherwise $\epsilon_n = 1$. Let $K_{\psi_{n2}}$ represent the number of columns in ψ_{n2}, then auxiliary matrix $\psi_{n3} = \frac{1}{K_{\psi_{n2}} + \epsilon_n}(\epsilon_n I_{K_{\psi_{n2}}} + 1_{K_{\psi_{n2}} \times K_{\psi_{n2}}})$. $\overline{V}_n = \psi_{n2}\psi_{n3}$.

$\psi_{(n)}$ is the n-model matrix of ψ, as follows

$$\psi_{(n)} = \begin{cases} \psi_{n3}^{-1}\psi_{n1}^{-1}\psi_{(n)}^{hosvd}, & \text{if} \quad \sum V_n^{d,hosvd} = 0 \\ \psi_{n3}^{-1} \begin{bmatrix} \psi^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \psi_{(n)}^{hosvd} \\ 0 \end{bmatrix}, & \text{otherwise} \end{cases}$$

where $\psi_{(n)}^{hosvd}$ is the n-model matrix of ψ^{hosvd}.

Step 4) Extract vertex tensor and weight function

The vertex tensor is a tensor composed of multiple linear system parameter matrices, and the weight function is the weight of the vertex tensor at different times.

Define the value of the weight function $\omega_z(p_{j_n})$ in (11) at the sampling mesh as the j line of V_n

$$\omega_z(p_{j_n}) = \overline{V}_n(j_n, :)^T \tag{A.5}$$

Therefore, the weight function can be obtained by performing piecewise interpolation with discrete values defined by (A.5).

In summary, the TPLM of the system parameter matrix (13) can now be obtained. At this point, all steps of the TPLM have been completed.

Numerical Example:

A polytope linearised model is as follows

$$\mathrm{S}(p) = \begin{bmatrix} -5(\rho-1)(\rho-2) & \cos(\rho-1) \\ \rho - 2.1 & \sin(\rho-1) \end{bmatrix} \tag{A.6}$$

Assuming the region is $p = [-0.5 : 0.5]$, the sampling rate $J = 5$, and the sampling tensor S^s is a $5 \times 2 \times 2$ order tensor, with the specific values of

$$\mathrm{S}^s(1,:,:) = \begin{bmatrix} -18.75 & 0.0707 \\ -2.6 & -0.9975 \end{bmatrix}$$

$$\mathrm{S}^s(2,:,:) = \begin{bmatrix} -14.0625 & 0.3153 \\ -2.35 & -0.9490 \end{bmatrix}$$

$$\mathrm{S}^s(3,:,:) = \begin{bmatrix} -10 & 0.5403 \\ -2.1 & -0.8415 \end{bmatrix}$$

$$\mathrm{S}^s(4,:,:) = \begin{bmatrix} -6.5625 & 0.7317 \\ -1.85 & -0.6816 \end{bmatrix}$$

$$\mathrm{S}^s(5,:,:) = \begin{bmatrix} -3.75 & 0.8776 \\ -1.6 & -0.4794 \end{bmatrix}$$

The calculations show that the non-zero 1-mode singular values of S^s are $27.034, 1.803, 0.127, 0.004$, respectively.

By retaining the largest singular value, it can be concluded that ψ^{hosvd} and $\omega_z(p_{j_n})$ are $1 \times 2 \times 2$ and 5×1, respectively, with the following values

$$\psi^{hosvd} = \begin{bmatrix} 26.5775 & -0.7363 \\ 4.5652 & 1.7630 \end{bmatrix}$$

$$\omega_z(p_{j_n}) = \begin{bmatrix} -0.7006 & -0.5287 & -0.3793 & -0.2526 & -0.1484 \end{bmatrix}^T$$

References

Baranyi, P., 2004. TP model transformation as a way to LMI-based controller design. IEEE Transactions on Industrial Electronics, 51(2), 387–400.

Baranyi, P., 2006. Output feedback control of two-dimensional aeroelastic system. Journal of Guidance, Control, and Dynamics, 29(3), 762–767.

Baranyi, P., Tikk, D., Yam, Y., Patton, R.J., 2003. From differential equations to PDC controller design via numerical transformation. Computers in Industry, 51(3), 281–297.

Chen, Q., Shi, H., Sun, M., 2020. Echo state network-based backstepping adaptive iterative learning control for strict-feedback systems: An error-tracking approach. IEEE Transactions on Cybernetics, 50(7), 3009–3022.

Cheng, L., Wu, W., Qiu, L., Liu, X., Ma, J., Zhang, J., Fang, Y., 2022. An improved data-driven based model predictive control for zero-sequence circulating current suppression in paralleled converters. International Journal of Electrical Power and Energy Systems, 143, 1–9.

Chi, R., Hui, Y., Huang, B., Hou, Z., 2021. Active disturbance rejection control for nonaffined globally Lipschitz nonlinear discrete-time systems. IEEE Transactions on Automatic Control, 66(12), 5955–5967.

Chi, R., Hui, Y., Zhang, S., Huang, B., Hou, Z., 2019. Discrete-time extended state observer-based model-free adaptive control via local dynamic linearization. IEEE Transactions on Industrial Electronics, 67(10), 8691-8701.

De Lathauwer, L., De Moor, B., Vandewalle, J., 2000. An introduction to independent component analysis. Journal of Chemometrics: A Journal of the Chemometrics Society, 14(3), 123–149.

Gao, S., Zhao, D., Yan, X., Spurgeon, S.K., 2023. Model-free adaptive state feedback control for a class of nonlinear systems. IEEE Transactions on Automation Science and Engineering, 1, 1–13.

Hassani, V., Pascoal, A.M., Onstein, T.F., 2018. Data-driven control in marine systems. Annual Reviews in Control, 46, 343–349.

He, W., Meng, T., Huang, D., Li, X., 2018. Adaptive boundary iterative learning control for an Euler-Bernoulli beam system with input constraint. IEEE Transactions on Neural Networks and Learning Systems, 29(5), 1539–1549.

Hou, Z., 1994. The parameter identification, adaptive control and model free learning adaptive control for nonlinear systems. Ph.D. thesis. Northeastern University.

Hou, Z., Chi, R., Gao, H., 2017. An overview of dynamic-linearization-based data-driven control and applications. IEEE Transactions on Industrial Electronics, 64(5), 4076–4090.

Hou, Z., Jin, S., 2011a. Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems. IEEE Transactions on Neural Networks, 22(12), 2173–2188.

Hou, Z., Jin, S., 2011c. A novel data-driven control approach for a class of discrete-time nonlinear systems. IEEE Transactions on Control Systems Technology, 19(6), 1549–1558.

Hou, Z., Jin, S., 2013. Model free adaptive control: theory and applications. CRC Press, Leiden, Netherlands.

Hou, Z., Wang, Z., 2013. From model-based control to data-driven control: survey, classification and perspective. Information Sciences, 235, 3–35.

Hou, Z., Xiong, S., 2019. On model-free adaptive control and its stability analysis. IEEE Transactions on Automatic Control, 64(11), 4555–4569.

Hou, Z., Xu, J., 2009. On data-driven control theory:the state of the art and perspective. Acta Automatica Sinica, 35(6), 650–667.

Matuško, J., Ileš, Š., Kolonić, F., Lešić, V., 2015. Control of 3d tower crane based on tensor product model transformation with neural friction compensation. Asian Journal of Control, 17(2), 443–458.

Meng, D., Moore, K.L., 2017. Convergence of iterative learning control for siso nonrepetitive systems subject to iteration-dependent uncertainties. Automatica, 79(2), 167–177.

Meng, W., Xie, S.Q., Liu, Q., Lu, C.Z., Ai, Q., 2017. Robust iterative feedback tuning control of a compliant rehabilitation robot for repetitive ankle training. IEEE/ASME Transactions on Mechatronics, 22(1), 173–184.

Petres, Z., 2006. Polytopic decomposition of linear parameter-varying models by tensor-product model transformation. Budapesti Műszaki és Gazdaságtudományi Egyetem.

Petres, Z., Baranyi, P., Korondi, P., Hashimoto, H., 2007. Trajectory tracking by TP model transformation: Case study of a benchmark problem. IEEE Transactions on Industrial Electronics, 54(3), 1654–1663.

Qurashi, M., Ma, T., Chaniotakis, E., Antoniou, C., 2020. PC-SPSA: Employing dimensionality reduction to limit SPSA search noise in DTA model calibration. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1635–1645.

Roman, R., Radac, M., Precup, R., 2016. Multi input multi output system experimental validation of model-free control and virtual reference feedback tuning techniques. IET Control Theory and Applications, 10(12), 1395–1403.

Roman, R.C., Precup, R.E., Hedrea, E.L., Preitl, S., Zamfirache, I.A., Bojan-Dragos, C.A., Petriu, E.M., 2022. Iterative feedback tuning algorithm for tower crane systems. Procedia Computer Science, 199, 157–165.

Safonov, M.G., Tsao, T.C., 1997a. The unfalsified control concept and learning. IEEE Transactions on Automatic Control, 42(6), 2819–2824.

Şahin, S., Güzeliş, C., 2016. Online learning arma controllers with guaranteed closed-loop stability. IEEE Transactions on Neural Networks and Learning Systems, 27(11), 2314–2326.

Silva, G.J., Datta, A., Bhattacharyya, S.P., 2002. New results on the synthesis of PID controllers. IEEE Transactions on Automatic Control, 47(2), 241–252.

Wang, R., Paul, A., Stefanovic, M., Safonov, M.G., 2007. Cost detectability and stability of adaptive control systems. International Journal of Robust and Nonlinear Control, 17(5), 549–561.

Xiong, S.s., Hou, Z.s., 2022. Model-free adaptive control for unknown mimo nonaffine nonlinear discrete-time systems with experimental validation. IEEE Transactions on Neural Networks and Learning Systems, 33(4), 1727–1739.

Xue, G., Liu, Y., Shi, Z., Guo, L., Li, Z., 2022. Research on trajectory tracking control of underwater vehicle manipulator system based on model-free adaptive control method. Journal of Marine Science and Engineering, 10(5), 652–662.

Zhang, S., Zhao, D., Spurgeon, S.K., 2021. Output feedback robust distributed model predictive control for parallel systems in process networks with competitive characteristics. Control Engineering Practice, 113(6), 104842.

Zhao, D., Zhu, Q., Dubbeldam, J., 2015. Terminal sliding mode control for continuous stirred tank reactor. Chemical engineering research and design, 94, 266–274.