IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 6, 2019, accepted July 16, 2019, date of publication July 26, 2019, date of current version August 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931198

Data-Driven Model-Free Adaptive Predictive Control for a Class of MIMO Nonlinear Discrete-Time Systems With Stability Analysis

YUAN GUO[1], ZHONGSHENG HOU[1,2], (Senior Member, IEEE), SHIDA LIU[3], AND SHANGTAI JIN[1]

[1]Advanced Control Systems Laboratory, School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[2]School of Automation, Qingdao University, Qingdao 266071, China
[3]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China

Corresponding author: Zhongsheng Hou (zhshhou@bjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61433002, Grant 61833001, Grant 61703019, and Grant 61573054, and in part by the Beijing Natural Science Foundation under Grant L161007.

ABSTRACT In this study, a model-free adaptive predictive control (MFAPC) method is proposed for a class of unknown nonlinear non-affine multiple-input and multiple-output (MIMO) systems based on a novel dynamic linearization technique and a new time-varying Pseudo–Jacobian matrix (PJM) parameter. The advantages of the proposed method are that it does not need the model information in the control system design, and it can avoid a short-sighted control decision and shows better control performance by integrating the idea of predictive control. The applicability and effectiveness of the proposed control scheme have been verified through rigorous mathematical analysis and extensive simulations.

INDEX TERMS Model-free adaptive predictive control (MFAPC), multiple-input and multiple-output (MIMO) nonlinear system, model-free adaptive control (MFAC), data-driven control (DDC).

I. INTRODUCTION

With the modern computer science developing rapidly, there have been great changes in large-scale industrial processes. The production equipment and technology we used today are becoming increasingly complicated. Due to the complexity of equipment and system, it is quite difficult to obtain a precise mathematical model in practice. Therefore, model-based control methods are difficult to be applied. At the same time, a great deal of useful data is generated and stored in the industrial process. When the precise mathematical models are not available, the use of the offline or online process data for control system design becomes very meaningful. Therefore, the study of the data-driven control (DDC) theory are of significance in industrial fields [1].

Up to present, there are a few of data-driven control (DDC) methods, for example, the reference [2] combines off-policy learning with experience replay, and proposes reinforcement learning-based adaptive optimal exponential tracking control for continuous-time linear systems with unknown dynamics. The reference [3] presents an intelligent PID algorithm, which uses on-line numerical differentiator based fast estimation and recognition technology to design controller, avoiding complicated and time-consuming parameter tuning. The references [4]–[6] propose a series of control methods based on virtual reference feedback tuning (VRFT), which is a method of identifying controller parameters using a set of off-line I/O data of controlled objects. There also exist other data-driven control methods, including proportion integral differential control (PID) [7], [8], iterative learning control (ILC) [9], [10], frequency domain robust control [11], iterative feedback tuning (IFT) [12], [13], model-free sliding mode control [14], lazy learning (LL) [15], model-free adaptive control (MFAC) [16] and so on. These control methods bypass the steps of modeling and design controllers directly through offline or online input and output data.

The associate editor coordinating the review of this manuscript and approving it for publication was Sun Junwei.

Among the DDC methods mentioned above, the MFAC method was originally presented in 1994 [16]. The MFAC method merely utilize the online input-output (I/O) data to design controller. The main idea of this method is that using the novel concept of pseudo partial derivative (PPD), to build an equivalent dynamic linearization data model at each operation point of the closed-loop system of the nonlinear system. Then, the system's PPD is online estimated by using system I/O data, and the controller is designed using the equivalent dynamic linearization data model according to some weighted one-step-ahead cost functions. This method does not need precise model and identification process, so it has the advantages of simple controller structure, simple controller parameter on-line tuning algorithm, small calculation burden, convenient implementation and strong robustness.

The MFAC method has been successfully applied to many actual systems, for example, the polymerization reaction process [17], the quadrotor aircraft [18], the twin rotor aerodynamic systems [19], the autonomous parking systems [20], the wide-area power system stabilizer [21], the multivariable industrial processes [22], the robot control [23], the traffic control [24], the launch vehicle [25] and so on.

Until now, the MFAPC for the unknown single input and single output (SISO) nonlinear non-affine plant had been well developed and a few applications were also reported. The literature [26] proposed a compact format dynamic linearization (CFDL) data model based MFAPC method and applied it to the phase splits control strategy of urban traffic networks. The literature [27] proposed a MFAPC method based on lazy learning and applied it to the control of three-tank water level. For the syngas industry, the literature [28] proposed a partial-format dynamic linearization data model based MFAPC method by using the local learning and applied it to the oxygen concentration control problem. To the best knowledge of the authors, the corresponding results of the MFAPC for the unknown multiple-input and multiple-output (MIMO) nonlinear non-affine plant have not been founded yet.

In this paper, the MFAPC method for a class of unknown MIMO nonlinear non-affine plant discrete-time systems is proposed. The feature of the MFAPC is that the controller is designed only depending on the system I/O data, and the system model information is not required. With the addition of predictive control ideas, this algorithm can make better control performance comparing to the control scheme without the prediction consideration.

The main contributions are as follows. Firstly, the unknown MIMO system is transformed into input-output data model by using dynamic linearization method. The Pseudo-Jacobian matrix (PJM) in the input-output data model is estimated and predicted on-line to realize the predictive control for the unknown weak decoupling MIMO system. Secondly, a novel stability analysis method based on contraction mapping is used to prove the tracking error convergence and the BIBO stability. The other one is that, due to the utilization of the prediction of future input and output information, the proposed MFAPC method can make a better control performance with good robustness which has been demonstrated by simulations.

The differences between this work and existing works are as follows. Compared with the MFAPC algorithm for SISO nonlinear systems [26]–[28], it is designed for MIMO nonlinear systems. The coupling between input and output variables of MIMO system can be compensated due to the on-line estimation of the $\boldsymbol{\Phi}(k)$ using the algorithm (19), and the auto decoupling in some extent can be realized for the unknown MIMO nonlinear system. It is noted that the stability analysis of this work is a contraction mapping based method, rather than the Lyapunov stability theory based, which is a novel in our adaptive control community. Compared with the MFAC algorithm of MIMO nonlinear system [31], this algorithm considers the future input and output information in the control performance index, and adds the prediction algorithm of pseudo-Jacobian matrix, thereby ensuring a good control performance and enhancing the robustness to unknown disturbance.

The rest of the paper is arranged as follows. In section 2, the MFAPC control scheme is designed based on a novel compact-form dynamic linearization technique for a class of discrete-time MIMO nonlinear systems. In section 3, the bounded-input and bounded-output (BIBO) stability of the CFDL-MFAPC schemes are analyzed. In section 4, through a series of simulation studies, the effectiveness, correctness and applicability of the MFAPC scheme are verified, and conclusions are obtained in section 5.

II. CONTROL SYSTEM DESIGN

The investigated MIMO nonlinear system is described as

$$\boldsymbol{y}(k+1) = \boldsymbol{f}(\boldsymbol{y}(k), \cdots, \boldsymbol{y}(k-n_y), \boldsymbol{u}(k), \cdots, \boldsymbol{u}(k-n_u)), \quad (1)$$

where $\boldsymbol{f}(\cdots) = (f_1(\cdots), \cdots, f_m(\cdots))^T \in \prod_{n_u+n_y+2} \mathrm{R}^m \mapsto \mathrm{R}^m$ denotes an unknown function. $\boldsymbol{u}(k) \in R^m$ is the control input at time k and $\boldsymbol{y}(k) \in R^m$ the system output; n_y and n_u are unknown constants.

A. COMPACT FORM DYNAMICAL LINEARIZATION

The two assumptions are given for the MIMO system (1) as follows.

Assumption 1 ([31]): The partial derivative of $\boldsymbol{f}_i(\cdots)$, $i = 1, 2, \cdots, m$, is continuous for the input $\boldsymbol{u}(k)$.

Assumption 2 ([31]): The MIMO system (1) satisfies the generalized *Lipschitz* condition, i.e., for $\forall k_1 \neq k_2, k_1 \geq 0, k_2 \geq 0, \boldsymbol{u}(k_1) \neq \boldsymbol{u}(k_2)$, it has $\|\boldsymbol{y}(k_1+1) - \boldsymbol{y}(k_2+1)\| \leq b\|\boldsymbol{u}(k_1) - \boldsymbol{u}(k_2)\|$, where $b > 0$.

Remark 1: In the practice viewpoint, assumptions 1 and 2 are both reasonable. Assumption 1 describes the typical condition for common nonlinear systems. Assumption 2 means that if the rate of change of control input is bounded, the variation rate of the system output is also finite, which can also be considered from an energy point of view that the finite input energy to a given physical plant can not lead

to the output energy of the given physical plant going to infinity. For the more theoretical explanations and the reasonality of assumption 2, they can be found in following literatures [31], [32].

Under the assumptions of 1 and 2, the following theorem 1 is presented.

Theorem 1: Considering the system (1) satisfying assumptions 1 and 2, if $\|\Delta \boldsymbol{u}(k)\| \neq 0$ for all k, then a time-varying matrix $\boldsymbol{\Phi}(k)$ termed Pseudo-Jacobian matrix (PJM) exists, *s.t.*, the system (1) can be described as the compact form dynamic linearization (CFDL) data model as follows.

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

with bounded $\boldsymbol{\Phi}(k)$ for any k, where

$$\boldsymbol{\Phi}(k) = \begin{bmatrix} \phi_{11}(k) & \phi_{12}(k) & \cdots & \phi_{1m}(k) \\ \phi_{21}(k) & \phi_{22}(k) & \cdots & \phi_{2m}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m1}(k) & \phi_{m2}(k) & \cdots & \phi_{mm}(k) \end{bmatrix} \in R^{m\times m},$$

is the unknown bounded PJM of system (1).

Proof 1: See reference [31].

Another assumption needs to be made in order to make a rigorous analysis for the closed-loop system stability.

Assumption 3 ([31]): The PJM $\boldsymbol{\Phi}(k)$ satisfies the diagonally dominant condition, that is, $|\phi_{ij}(k)| < b_1, b_2 \leq |\phi_{ii}(k)| \leq \alpha b_2$, $\alpha \geq 1$, $i = 1, 2\cdots, m$, $j = 1, 2, \cdots, m$, $i \neq j$, and the signs of all elements in $\boldsymbol{\Phi}(k)$ are fixed, where b_1, b_2 are two positive constants, and satisfy $b_2 > b_1 (2\alpha + 1)(m-1)$.

Remark 2: Assumption 3 is a description about the coupling relationship between input and output in closed-loop data. For a class of unknown MIMO non-linear systems under this weak coupling condition with the unavailability of the plant model and only the system I/O data, the coupling among the variables of the system is described via this diagonal dominant matrix $\boldsymbol{\Phi}(k)$, which can reflect data relationship between control input and system output. Strictly speaking, assumption 3 cannot be checked in practice. The reason is that the MFAPC is a data-driven control method, which means that the plant model cannot be obtained, and only the I/O data till the current time instant k is available for the MFAPC control systems design. No any future data of the open-loop or the closed-loop of the controlled plant can be used for the check. Of course, if the future I/O data after the time k of the controlled system is assumed to be sufficient, complete and available, then this condition could be checked. In fact, this condition is not very restrictive. It includes the LTI systems, and the nonlinear systems whose models are assumed to have continuously bounded partial derivatives with respect to its variables. Further, lots of practical plants can satisfy this assumption, and already validated by the practical applications, such as, [6], [19], [31].

Next, the MFAPC controller is designed based on the CFDL data model.

B. CONTROLLER DESIGN

On the basis of the above CFDL data model, a one-step prediction equation can be obtained as follows.

$$\boldsymbol{y}(k+1) = \boldsymbol{y}(k) + \boldsymbol{\Phi}(k)\Delta \boldsymbol{u}(k). \tag{3}$$

According to (3), we can give the N-step-ahead prediction equations.

$$\begin{cases} \boldsymbol{y}(k+1) = \boldsymbol{y}(k) + \boldsymbol{\Phi}(k)\Delta \boldsymbol{u}(k), \\ \boldsymbol{y}(k+2) = \boldsymbol{y}(k+1) + \boldsymbol{\Phi}(k+1)\Delta \boldsymbol{u}(k+1) \\ \quad = \boldsymbol{y}(k) + \boldsymbol{\Phi}(k)\Delta \boldsymbol{u}(k) + \boldsymbol{\Phi}(k+1)\Delta \boldsymbol{u}(k+1), \\ \quad \vdots \\ \boldsymbol{y}(k+N) = \boldsymbol{y}(k+N-1) + \boldsymbol{\Phi}(k+N-1)\Delta \boldsymbol{u}(k+N-1) \\ \quad = \boldsymbol{y}(k+N-2) + \boldsymbol{\Phi}(k+N-2)\Delta \boldsymbol{u}(k+N-2) \\ + \boldsymbol{\Phi}(k+N-1)\Delta \boldsymbol{u}(k+N-1) \\ \quad \vdots \\ \quad = \boldsymbol{y}(k) + \boldsymbol{\Phi}(k)\Delta \boldsymbol{u}(k) \\ + \cdots + \boldsymbol{\Phi}(k+N-1)\Delta \boldsymbol{u}(k+N-1). \end{cases} \tag{4}$$

Let $\boldsymbol{Y}_{Nm}(k+1)$, as shown at the top of the next page, where $\boldsymbol{Y}_{Nm}(k+1)$ denotes the N-step-ahead prediction vector of the output, $\Delta \boldsymbol{U}_{Nm}(k)$ denotes an increment vector of the system input, $\boldsymbol{I}_{m\times m}$ is an identity matrix of $m \times m$, $\boldsymbol{0}$ is a zero matrix of $m \times m$, and N_u is the system input horizon.

So, the equation (4) can be rewritten in a compact form,

$$\boldsymbol{Y}_{Nm}(k+1) = \boldsymbol{E}(k)\boldsymbol{y}(k) + \boldsymbol{A}(k)\Delta \boldsymbol{U}_{Nm}(k). \tag{5}$$

If $\Delta \boldsymbol{u}(k+j-1) = 0, j > N_u$, then prediction equation (5) becomes

$$\boldsymbol{Y}_{Nm}(k+1) = \boldsymbol{E}(k)\boldsymbol{y}(k) + \boldsymbol{A}_1(k)\Delta \boldsymbol{U}_{N_u m}(k), \tag{6}$$

where,

$$\boldsymbol{A}_1(k) = \begin{bmatrix} \boldsymbol{\Phi}(k) & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{\Phi}(k) & \boldsymbol{\Phi}(k+1) & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Phi}(k) & \boldsymbol{\Phi}(k+1) & \cdots & \boldsymbol{\Phi}(k+N_u-1) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Phi}(k) & \boldsymbol{\Phi}(k+1) & \cdots & \boldsymbol{\Phi}(k+N_u-1) \end{bmatrix} \in R^{Nm\times N_u m},$$

$$\boldsymbol{\Phi}(k) = \begin{bmatrix} \phi_{11}(k) & \phi_{12}(k) & \cdots & \phi_{1m}(k) \\ \phi_{21}(k) & \phi_{22}(k) & \cdots & \phi_{2m}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m1}(k) & \phi_{m2}(k) & \cdots & \phi_{mm}(k) \end{bmatrix} \in R^{m\times m},$$

$$\Delta \boldsymbol{U}_{N_u m}(k) = \left[\Delta \boldsymbol{u}^T(k), \Delta \boldsymbol{u}^T(k+1), \cdots, \Delta \boldsymbol{u}^T(k+N_u-1)\right]^T.$$

As shown above, we obtained the N-step forward prediction data model. Next, considering the tracking error and

$$\begin{cases} \boldsymbol{Y}_{Nm}(k+1) = \left[\boldsymbol{y}^T(k+1), \boldsymbol{y}^T(k+2), \cdots, \boldsymbol{y}^T(k+N)\right]^T, \\ \Delta\boldsymbol{U}_{Nm}(k) = \left[\Delta\boldsymbol{u}^T(k), \Delta\boldsymbol{u}^T(k+1), \cdots, \Delta\boldsymbol{u}^T(k+N-1)\right]^T, \\ \boldsymbol{E}(k) = [\boldsymbol{I}_{m\times m}, \boldsymbol{I}_{m\times m}, \cdots, \boldsymbol{I}_{m\times m}]^T, \\ \boldsymbol{A}(k) = \begin{bmatrix} \boldsymbol{\Phi}(k) & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Phi}(k) & \boldsymbol{\Phi}(k+1) & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{\Phi}(k) & \boldsymbol{\Phi}(k+1) & \cdots & \boldsymbol{\Phi}(k+N_u-1) & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{\Phi}(k) & \boldsymbol{\Phi}(k+1) & \cdots & \boldsymbol{\Phi}(k+N_u-1) & \cdots & \boldsymbol{\Phi}(k+N-1) \end{bmatrix}_{Nm\times Nm} \end{cases}$$

control input variation, the criterion function for control input is constructed as follows.

$$J = \sum_{i=1}^{N} \left\|\boldsymbol{y}^*(k+i) - \boldsymbol{y}(k+i)\right\|^2 + \lambda \sum_{j=0}^{N_u-1} \|\Delta\boldsymbol{u}(k+j)\|^2, \tag{7}$$

where the weighting factor λ is positive constant, and $\boldsymbol{y}^*(k+i)$ is the desired system output at time $k+i$, $i = 1, 2, \cdots, N$.

Let $\boldsymbol{Y}^*_{Nm}(k+1) = \left[\boldsymbol{y}^{*T}(k+1), \cdots, \boldsymbol{y}^{*T}(k+N)\right]^T$, then the cost function (7) is rewritten as:

$$J = \left[\boldsymbol{Y}^*_{Nm}(k+1) - \boldsymbol{Y}_{Nm}(k+1)\right]^T \left[\boldsymbol{Y}^*_{Nm}(k+1) - \boldsymbol{Y}_{Nm}(k+1)\right] + \lambda\Delta\boldsymbol{U}^T_{N_um}(k)\Delta\boldsymbol{U}_{N_um}(k). \tag{8}$$

Substituting (6) into (8), the control law can be obtained through the optimality condition $\frac{\partial J}{\partial \boldsymbol{U}_{N_um}(k)} = 0$.

$$\Delta\boldsymbol{U}_{N_um}(k) = \left[\boldsymbol{A}_1^T(k)\boldsymbol{A}_1(k) + \lambda\boldsymbol{I}\right]^{-1}\boldsymbol{A}_1^T(k)\left[\boldsymbol{Y}^*_{Nm}(k+1) - \boldsymbol{E}(k)\boldsymbol{y}(k)\right]. \tag{9}$$

The above algorithm includes the calculation of matrix inverse. The calculation of matrix inverse is time-consuming if the dimension of system I/O data is very large. Therefore, (9) can be simplified as follows.

$$\Delta\boldsymbol{U}_{N_um}(k) = \frac{\boldsymbol{A}_1^T(k)\left[\boldsymbol{Y}^*_{Nm}(k+1) - \boldsymbol{E}(k)\boldsymbol{y}(k)\right]}{\lambda + \|\boldsymbol{A}_1(k)\|^2}. \tag{10}$$

Thus, based on the receding horizon principle, the control input of system at time k is obtained as follows.

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \boldsymbol{g}^T\Delta\boldsymbol{U}_{N_um}(k), \tag{11}$$

where $\boldsymbol{g} = [\boldsymbol{I}_{m\times m}, \boldsymbol{0}_{m\times m}, \cdots, \boldsymbol{0}_{m\times m}]^T \in R^{m\times N_um}$, $\boldsymbol{0}_{m\times m} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{m\times m}$.

Remark 3: The dynamical linearization technique is applied to achieve the virtual dynamic linearization model at each optional point. Moreover, the PJM in this virtual dynamic linearization model contains all system information, including the model uncertainties. Besides, there also exist other references for addressing system uncertainties, such as references [29] and [30]. The main difference between them is that the modeling uncertainties are addressed directly by uncertainty description using the known model or by a implicitly estimation with PJM using the I/O data.

C. PJM ESTIMATION AND PREDICTION

Since $\boldsymbol{A}_1(k)$ in equation (9) contains the unknown time-varying PJM parameter $\boldsymbol{\Phi}(k), \boldsymbol{\Phi}(k+1), \cdots, \boldsymbol{\Phi}(k+N_u+1)$. In following, we will discuss the PJM estimation and prediction algorithms in order to implement the controller (10) (11) in practice.

1) PJM ESTIMATION ALGORITHM

Firstly, we design the estimation algorithm of Pseudo-Jacobian matrix. The modified projection algorithm is used to estimate $\boldsymbol{\Phi}(k)$ which is based on the optimization of following criterion function.

$$J(\boldsymbol{\Phi}(k)) = \|\Delta\boldsymbol{y}(k) - \boldsymbol{\Phi}(k)\Delta\boldsymbol{u}(k-1)\|^2 + \mu\left\|\boldsymbol{\Phi}(k) - \hat{\boldsymbol{\Phi}}(k-1)\right\|^2, \tag{12}$$

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

Minimizing cost function (12) yields the following projection algorithm,

$$\hat{\boldsymbol{\Phi}}(k) = \hat{\boldsymbol{\Phi}}(k-1) + \left[\Delta\boldsymbol{y}(k) - \hat{\boldsymbol{\Phi}}(k-1)\Delta\boldsymbol{u}(k-1)\right]\Delta\boldsymbol{u}^T(k-1) \cdot \left[\Delta\boldsymbol{u}(k-1)\Delta\boldsymbol{u}^T(k-1) + \mu\boldsymbol{I}\right]^{-1}. \tag{13}$$

Since the complex matrix inversion computation is involved in (13), which will leads to many difficulties when it is used in practice. Thus we will modify the PJM estimation algorithm (13) into a simplified version as follows.

$$\hat{\boldsymbol{\Phi}}(k) = \hat{\boldsymbol{\Phi}}(k-1) + \frac{\eta\left[\Delta\boldsymbol{y}(k) - \hat{\boldsymbol{\Phi}}(k-1)\Delta\boldsymbol{u}(k-1)\right]\Delta\boldsymbol{u}^T(k-1)}{\mu + \|\Delta\boldsymbol{u}(k-1)\|^2} \tag{14}$$

where $\eta \in [0, 2]$, μ is a positive parameter, and

$$\hat{\boldsymbol{\Phi}}(k) = \begin{bmatrix} \hat{\phi}_{11}(k) & \hat{\phi}_{12}(k) & \cdots & \hat{\phi}_{1m}(k) \\ \hat{\phi}_{21}(k) & \hat{\phi}_{22}(k) & \cdots & \hat{\phi}_{2m}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\phi}_{m1}(k) & \hat{\phi}_{m2}(k) & \cdots & \hat{\phi}_{mm}(k) \end{bmatrix} \in R^{m\times m}$$

is the estimation of unknown PJM $\boldsymbol{\Phi}(k)$. Moreover, in order to improve the tracking capacity of the algorithm (14) for time-varying parameters, according to assumption 3, the reset mechanism of the above algorithm is set as below:

$$\begin{aligned}\hat{\phi}_{ii}(k) &= \hat{\phi}_{ii}(1), \text{ if } \left|\hat{\phi}_{ii}(k)\right| < b_2 \text{ or } \left|\hat{\phi}_{ii}(k)\right| > ab_2 \text{ or} \\ &\quad sign(\hat{\phi}_{ii}(k)) \neq sign(\hat{\phi}_{ii}(1)), \quad i = 1, \cdots, m. \\ \hat{\phi}_{ij}(k) &= \hat{\phi}_{ij}(1), \text{ if } \left|\hat{\phi}_{ij}(k)\right| > b_1 \text{ or} \\ &\quad sign(\hat{\phi}_{ij}(k)) \neq sign(\hat{\phi}_{ij}(1))i, \quad j = 1, \cdots, m, i \neq j.\end{aligned}$$

2) PJM PREDICTION ALGORITHM

Since $\boldsymbol{\Phi}(k+1), \cdots, \boldsymbol{\Phi}(k+N_u-1)$ cannot be directly calculated from the I/O data till sample time k, $\boldsymbol{\Phi}(k+1), \cdots, \boldsymbol{\Phi}(k+N_u-1)$ need to be predicted according to the past estimated values $\boldsymbol{\Phi}(1)\cdots, \boldsymbol{\Phi}(k)$.

Assume that the estimated values $\boldsymbol{\Phi}(1)\cdots, \boldsymbol{\Phi}(k)$ have been calculated by equation (14) at time k. By use of these estimated values, and an autoregressive (AR) prediction model as follows:

$$\begin{aligned}&\hat{\boldsymbol{\Phi}}(k+1) \\ &= \boldsymbol{\theta}_1(k)\hat{\boldsymbol{\Phi}}(k)+\boldsymbol{\theta}_2(k)\hat{\boldsymbol{\Phi}}(k-1)+\cdots+\boldsymbol{\theta}_{n_p}(k)\hat{\boldsymbol{\Phi}}(k-n_p+1),\end{aligned} \tag{15}$$

where n_p is the fixed model order, which is usually set to 2∼7 as recommended by Reference [33], the predicted PJM can be obtained.

Therefore, the prediction equation becomes

$$\begin{aligned}\hat{\boldsymbol{\Phi}}(k+j) &= \boldsymbol{\theta}_1(k)\hat{\boldsymbol{\Phi}}(k+j-1) + \boldsymbol{\theta}_2(k)\hat{\boldsymbol{\Phi}}(k+j-2) \\ &\quad + \cdots + \boldsymbol{\theta}_{n_p}(k)\hat{\boldsymbol{\Phi}}(k+j-n_p).\end{aligned} \tag{16}$$

where $\boldsymbol{\theta}_i(k) \in R^{m\times m}$, $i = 1, 2, \cdots, n_p$.

Let

$$\begin{aligned}\boldsymbol{\theta}(k) &= \left[\boldsymbol{\theta}_1(k), \cdots, \boldsymbol{\theta}_{n_p}(k)\right]^T, \\ \hat{\boldsymbol{\varphi}}(k-1) &= \left[\hat{\boldsymbol{\Phi}}^T(k-1), \cdots, \hat{\boldsymbol{\Phi}}^T(k-n_p)\right]^T, \\ \hat{\boldsymbol{\Phi}}(k) &= \boldsymbol{\theta}^T(k)\hat{\boldsymbol{\varphi}}(k-1).\end{aligned}$$

For the parameter determination of the $\boldsymbol{\theta}(k)$, we will use the following optimization to get the parameter updating law.

$$J(\boldsymbol{\theta}(k)) = \left\|\hat{\boldsymbol{\Phi}}^T(k) - \hat{\boldsymbol{\varphi}}^T(k-1)\boldsymbol{\theta}(k)\right\|^2 + \delta\left\|\boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}(k-1)\right\|^2. \tag{17}$$

Minimizing the function (17), we have the following projection algorithm,

$$\begin{aligned}\boldsymbol{\theta}(k) &= \boldsymbol{\theta}(k-1) \\ &\quad + \frac{\hat{\boldsymbol{\varphi}}(k-1)}{\delta + \left\|\hat{\boldsymbol{\varphi}}(k-1)\right\|^2}\left[\hat{\boldsymbol{\Phi}}^T(k) - \hat{\boldsymbol{\varphi}}^T(k-1)\boldsymbol{\theta}(k-1)\right],\end{aligned} \tag{18}$$

where $\delta \in [0, 1]$ is a positive constant.

III. CONTROL SYSTEM DESIGN AND STABILITY ANALYSIS

A. CONTROL SCHEME

By integrating controller algorithm (10)-(11), PJM estimation algorithm (14) and PJM prediction algorithm (16) and (18), the MFAPC scheme is designed as below.

$$\begin{aligned}\hat{\boldsymbol{\Phi}}(k) &= \hat{\boldsymbol{\Phi}}(k-1) \\ &\quad + \frac{\eta\left[\Delta\boldsymbol{y}(k) - \hat{\boldsymbol{\Phi}}(k-1)\Delta\boldsymbol{u}(k-1)\right]\Delta\boldsymbol{u}^T(k-1)}{\mu + \|\Delta\boldsymbol{u}(k-1)\|^2},\end{aligned} \tag{19}$$

$$\begin{aligned}\hat{\phi}_{ii}(k) &= \hat{\phi}_{ii}(1), \text{ if } \left|\hat{\phi}_{ii}(k)\right| < b_2 \text{ or } \left|\hat{\phi}_{ii}(k)\right| > ab_2 \text{ or} \\ &\quad sign(\hat{\phi}_{ii}(k)) \neq sign(\hat{\phi}_{ii}(1)), \quad i = 1, \cdots, m.\end{aligned} \tag{20}$$

$$\begin{aligned}\hat{\phi}_{ij}(k) &= \hat{\phi}_{ij}(1), \text{ if } \left|\hat{\phi}_{ij}(k)\right| > b_1 \text{ or} \\ &\quad sign(\hat{\phi}_{ij}(k)) \neq sign(\hat{\phi}_{ij}(1)), \\ &\quad i, j = 1, \cdots, m; i \neq j.\end{aligned} \tag{21}$$

$$\begin{aligned}\boldsymbol{\theta}(k) &= \boldsymbol{\theta}(k-1) + \frac{\hat{\boldsymbol{\varphi}}(k-1)}{\delta + \left\|\hat{\boldsymbol{\varphi}}(k-1)\right\|^2} \\ &\quad \times \left[\hat{\boldsymbol{\Phi}}^T(k) - \hat{\boldsymbol{\varphi}}^T(k-1)\boldsymbol{\theta}(k-1)\right],\end{aligned} \tag{22}$$

$$\boldsymbol{\theta}(k) = \boldsymbol{\theta}(1), \text{ if } \|\boldsymbol{\theta}(k)\| \geq M, \tag{23}$$

$$\begin{aligned}\hat{\boldsymbol{\Phi}}(k+j) &= \boldsymbol{\theta}_1(k)\hat{\boldsymbol{\Phi}}(k+j-1) + \boldsymbol{\theta}_2(k)\hat{\boldsymbol{\Phi}}(k+j-2) + \cdots \\ &\quad + \boldsymbol{\theta}_{np}(k)\hat{\boldsymbol{\Phi}}(k+j-np), \\ &\quad j = 1, 2, \cdots, N_u - 1,\end{aligned} \tag{24}$$

$$\begin{aligned}\hat{\phi}_{ii}(k+j) &= \hat{\phi}_{ii}(1), \text{ if } \left|\hat{\phi}_{ii}(k+j)\right| < b_2 \text{ or } \left|\hat{\phi}_{ii}(k+j)\right| > ab_2 \\ &\quad \text{or } sign(\hat{\phi}_{ii}(k+j)) \neq sign(\hat{\phi}_{ii}(1)), \\ &\quad i = 1, \cdots, m,\end{aligned} \tag{25}$$

$$\begin{aligned}\hat{\phi}_{ij}(k+j) &= \hat{\phi}_{ij}(1), \text{ if } \left|\hat{\phi}_{ij}(k+j)\right| > b_1 \text{ or} \\ &\quad sign(\hat{\phi}_{ij}(k+j)) \neq sign(\hat{\phi}_{ij}(1))i, \\ &\quad j = 1, \cdots, m; i \neq j,\end{aligned} \tag{26}$$

$$\Delta\boldsymbol{U}_{N_u m}(k) = \frac{\hat{\boldsymbol{A}}_1^T(k)\left[\boldsymbol{Y}^*_{Nm}(k+1) - \boldsymbol{E}(k)\boldsymbol{y}(k)\right]}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}, \tag{27}$$

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \boldsymbol{g}^T\Delta\boldsymbol{U}_{N_u m}(k), \tag{28}$$

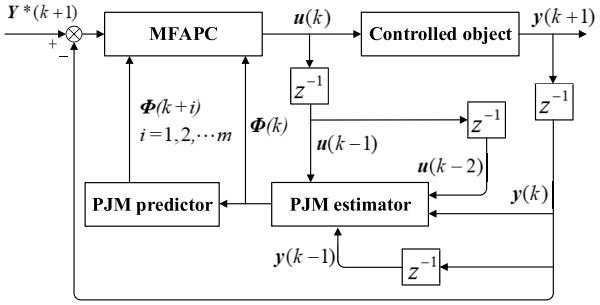

FIGURE 1. The diagram of MFAPC.

where $\boldsymbol{g} = [\boldsymbol{I}_{m\times m}, \boldsymbol{0}_{m\times m}, \cdots, \boldsymbol{0}_{m\times m}]^T$,

$$\hat{\boldsymbol{A}}_1(k) = \begin{bmatrix} \hat{\boldsymbol{\Phi}}(k) & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \hat{\boldsymbol{\Phi}}(k) & \hat{\boldsymbol{\Phi}}(k+1) & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{\Phi}}(k) & \hat{\boldsymbol{\Phi}}(k+1) & \cdots & \hat{\boldsymbol{\Phi}}(k+N_u-1) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{\Phi}}(k) & \hat{\boldsymbol{\Phi}}(k+1) & \cdots & \hat{\boldsymbol{\Phi}}(k+N_u-1) \end{bmatrix} \in R^{Nm\times N_u m},$$

$$\hat{\boldsymbol{\Phi}}(k) = \begin{bmatrix} \hat{\phi}_{11}(k) & \hat{\phi}_{12}(k) & \cdots & \hat{\phi}_{1m}(k) \\ \hat{\phi}_{21}(k) & \hat{\phi}_{22}(k) & \cdots & \hat{\phi}_{2m}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\phi}_{m1}(k) & \hat{\phi}_{m2}(k) & \cdots & \hat{\phi}_{mm}(k) \end{bmatrix} \in R^{m\times m}.$$

The Fig. 1 is the flow chart of MFAPC algorithm.

Remark 4: The prediction step size N is a key parameter in the MFAPC algorithm, and its size can be adjusted. In general, the prediction step size should be large enough to include the dynamic characteristics of the system.

B. STABILITY ANALYSIS

Lemma 1: [34]: Let $\boldsymbol{A} = (\alpha_{ij}) \in C^{n\times n}$. For each $1 \le i \le n$, denote the Gerschgorin disk as $D_i = \left\{ z \left| |z-\alpha_{ii}| \le \sum_{j=1, j\ne i}^{n} |\alpha_{ij}| \right. \right\}$, $z \in C^{n\times n}$, and the Gerschgorin domain is denotes a union of all Gerschgorin disks, i.e., $D_A = \bigcup_{i=1}^{n} D_i$. Each eigenvalue of $\boldsymbol{A}$ lie in D_A.

Theorem 2: For a regulation problem $\boldsymbol{y}^*(k+1) = \boldsymbol{y}^* = const$, if the system (1) satisfies assumptions 1-3, and the corresponding control scheme (19)-(28) is applied, then a constant $\lambda_{\min} > 0$ must exist, s.t., for $\forall\lambda > \lambda_{\min}$, it has

1) The tracking error sequence is convergent, that is, $\lim_{k\to\infty} \|\boldsymbol{y}(k+1) - \boldsymbol{y}^*\|_v = 0$, where $\|\cdot\|_v$ is the consistent norm.

2) The system inputs and outputs are bounded, i.e., the sequences $\{\boldsymbol{y}(k)\}$ and $\{\boldsymbol{u}(k)\}$ are bounded.

Proof: Appendix A. ■

Remark 5: From the proof, we can see that the persistently exciting condition is not required when the proposed control scheme is used for the adaptive control of an unknown nonlinear MIMO system.

Remark 6: From the proof of theorem 2 we can see that the closed-loop stability and convergence of the regulation problem of an unknown nonlinear system can be guaranteed. It is noteworthy that the proposed algorithm is a pure data-driven control method, in which the controller design process does not contain any model information. When the desired signal $\boldsymbol{y}^*$ is a time-varying signal, the proof of error convergence and stability can also be addressed by the following method.

A new controlled plant can be constructed,

$$\boldsymbol{c}(k+1) = \boldsymbol{y}(k+1) - \boldsymbol{y}^*(k+1) \tag{29}$$

where $\boldsymbol{y}^*(k+1)$ is a time-varying desired signal, and $\boldsymbol{y}(k+1)$ is the output of controlled plant (1).

The controlled plant (29) is a nonlinear discrete system as well. Therefore, the tracking issue of the controlled plant $\boldsymbol{y}$ to time-varying desired signal $\boldsymbol{y}^*$ can be transformed into the regulator problem of the controlled plant (29) to the time-invariant desired signal. In a sequel, we can claim that the tracking problem of the MIMO nonlinear system is also be proven.

IV. SIMULATIONS

In order to verify the correctness and the effectiveness of the proposed MFAPC scheme, extensive simulations are made in this section. Part A presents a series of numerical simulations, and part B is the simulations based on actual two degree of freedom manipulator system.

It is noteworthy that the system model in following simulations is merely used to generate the system I/O data, not to design the controller.

A. NUMERICAL SIMULATION

The system studied in example 1 is a numerical MIMO discrete system [31], and three other DDC algorithms are compared with MFAPC algorithm, including prototype MFAC [31], intelligent PID [3] and VRFT [4].

The MFAC control scheme is:

$$\boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \frac{\rho\boldsymbol{\Phi}^T(k)(\boldsymbol{y}^*(k+1) - \boldsymbol{y}(k))}{\lambda + \|\boldsymbol{\Phi}(k)\|^2}, \tag{30}$$

where $\boldsymbol{\Phi}(k)$ is estimated using (19)-(21).

For intelligent PID, the control structure is set to:

$$\begin{cases} u_1 = -\dfrac{F_1}{\alpha_1} + \dfrac{\dot{y}_1^*}{\alpha_1} + k_{p1}\cdot e_1 + k_{i1}\cdot \displaystyle\int e_1 \,, \\ u_2 = -\dfrac{F_2}{\alpha_2} + \dfrac{\dot{y}_2^*}{\alpha_2} + k_{p2}\cdot e_2 + k_{i2}\cdot \displaystyle\int e_2 \,. \end{cases} \tag{31}$$

where, $[F_i]_e = [\dot{y}_i]_e - u$, $[\bullet]_e$ is the estimate of a given quantity. the tracking error $e_i = y_i - y_i^*$, $i = 1, 2$

For VRFT, the controller scheme is set as the PID controller:

$$\begin{cases} \Delta u_1(k) = k_{p1}\cdot\Delta e_1(k) + k_{i1}\cdot e_1(k) \\ \qquad\qquad + k_{d1}(\Delta e_1(k) - \Delta e_1(k-1)) \,, \\ \Delta u_2(k) = k_{p2}\cdot\Delta e_2(k) + k_{i2}\cdot e_2(k) \\ \qquad\qquad + k_{d2}(\Delta e_2(k) - \Delta e_2(k-1)) \,. \end{cases} \tag{32}$$

TABLE 1. The parameter setting of algorithms.

MFAPC	$\eta = 0.8, \rho = 1, \mu = 1, \lambda = 4.5,$ $\boldsymbol{\Phi}_c(1) = \boldsymbol{\Phi}_c(2) = \boldsymbol{\Phi}_c(3) = \boldsymbol{\Phi}_c(4) = \begin{bmatrix} 0.5 & 0.02 \\ 0.01 & 0.5 \end{bmatrix}^T$.
MFAC	$\eta = 0.8, \rho = 1, \mu = 1, \lambda = 0.56,$ $\boldsymbol{\Phi}_c(1) = \boldsymbol{\Phi}_c(2) = \boldsymbol{\Phi}_c(3) = \boldsymbol{\Phi}_c(4) = \begin{bmatrix} 0.5 & 0.01 \\ 0.007 & 0.5 \end{bmatrix}^T$.
i-PID	$\alpha_1 = 0.86, k_{p1} = 0.4, k_{i1} = 0.38,$ $\alpha_2 = 0.8, k_{p2} = 0.42, k_{i2} = 0.35,$
VRFT	$k_{p1} = 0.63, k_{i1} = 0.86, k_{d1} = 0.32,$ $k_{p2} = 0.42, k_{i2} = 0.63, k_{d2} = 0.2.$

where, the reference model is $M_i(z) = \alpha_i/(1 - \alpha_i z^{-1})$, $i = 1, 2$. the weight function is $W_i(z) = 1$, $i = 1, 2$. According to the optimization method in [4], k_p, k_i and k_d are optimized respectively, the values are shown in the table 1.

Example 1: [31]:

The MIMO discrete nonlinear system can be described as:

$$\begin{cases} x_{11}(k+1) = \dfrac{x_{11}^2(k)}{1+x_{11}^2(k)} + 0.3x_{12}(k), \\ x_{12}(k+1) = \dfrac{x_{11}^2(k)}{1+x_{12}^2(k)+x_{21}^2(k)+x_{22}^2(k)} + a(k)u_1(k), \\ x_{21}(k+1) = \dfrac{x_{21}^2(k)}{1+x_{21}^2(k)} + 0.2x_{22}(k), \\ x_{22}(k+1) = \dfrac{x_{21}^2(k)}{1+x_{11}^2(k)+x_{12}^2(k)+x_{22}^2(k)} + b(k)u_2(k), \\ y_1(k+1) = x_{11}(k+1), \\ y_2(k+1) = x_{21}(k+1), \end{cases} \tag{33}$$

where

$$\begin{aligned} a(k) &= 1 + 0.1\sin(2\pi k/1500); \\ b(k) &= 1 + 0.1\cos(2\pi k/1500). \end{aligned} \tag{34}$$

The initial condition of the system is

$$\begin{aligned} &u_1(i) = u_2(i) = 0.5, \quad i = 1, 2, 3, 4. \\ &x_{11}(j) = x_{21}(j) = 0.5, \quad x_{12}(j) = x_{22}(j) = 0,\ j = 1, 2, 3, 4. \end{aligned}$$

In this subsection, the tracking ability of MFAPC algorithm and other three DDC algorithms to the same desired output is compared, and two kinds of desired signals, constant signal and time-varying signal, are simulated respectively.

1) CONSTANT DESIRED SIGNAL

In this case, the desired signal $\boldsymbol{y}^*$ is set as follows:

$$\begin{cases} y_1^*(k) = 1.25, \\ y_2^*(k) = \begin{cases} 1.2, & 0 < k \le 300, \\ 2.1, & 300 < k \le 700 \\ 1.1, & 700 < k \le 1000 \end{cases} \end{cases} \tag{35}$$

The parameter setting of the four algorithms are shown in Table 1.

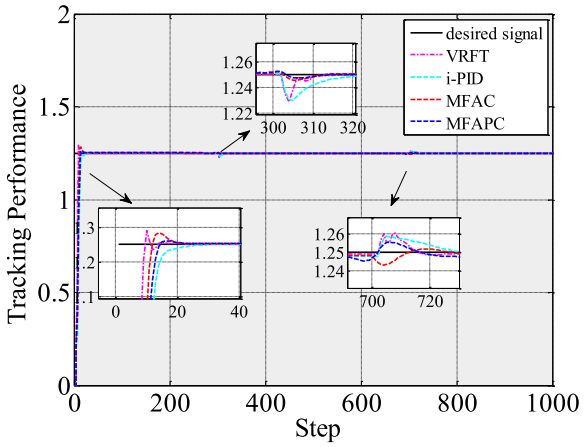

FIGURE 2. Tracking performance of y_1 with a constant desired signal.

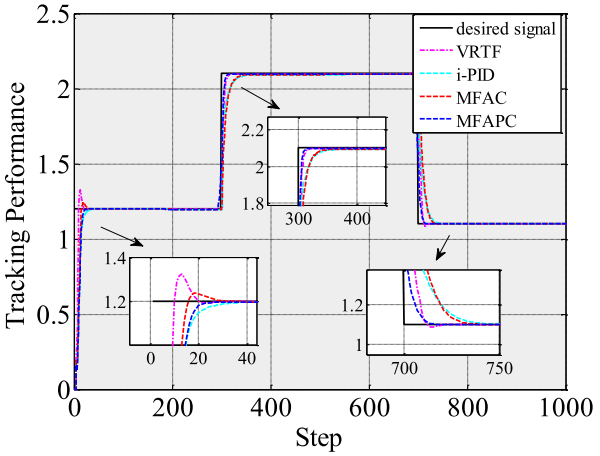

FIGURE 3. Tracking performance of y_2 with a constant desired signal.

The simulation results of this case are shown in Figure 2-5, and the overshoot index is shown in Table 2-3.

The simulation results illustrate that the four algorithms have good control performance. However, it can be observed from Figure 2-3 and table 2-3 that VRFT algorithm and MFAC algorithm have obvious overshoot. The MFAPC algorithm and i-PID algorithm have low overshoot. Furthermore, compared with the i-PID algorithm, the convergence speed of MFAPC algorithm is faster. In addition, when the desired signal changes abruptly, MFAPC algorithm can predict the change of the desired signal and track the signal ahead of time. Therefore, the tracking performance of MFAPC algorithm is better than the other three algorithms.

From the Figure 4-5 and the table of overshoot index, it can be concluded that the overshoot value is related to the change rate of the control input. The control input curves of MFAPC algorithm and i-PID algorithm are relatively smooth, so they have the characteristics of low overshoot.

In addition, as shown in Figure 2, when the control input u_2 changes abruptly, as can be seen in step 300 and step 700, the tracking signal y_1 of the MFAPC algorithm has less influence and the output curve is smoother. The reason is that

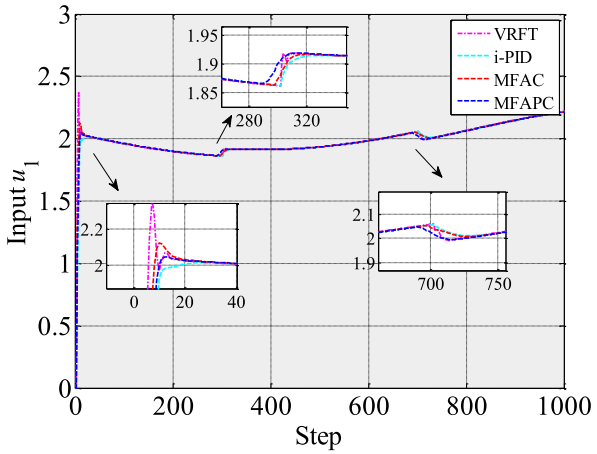

FIGURE 4. Control input u_1 with a constant desired signal.

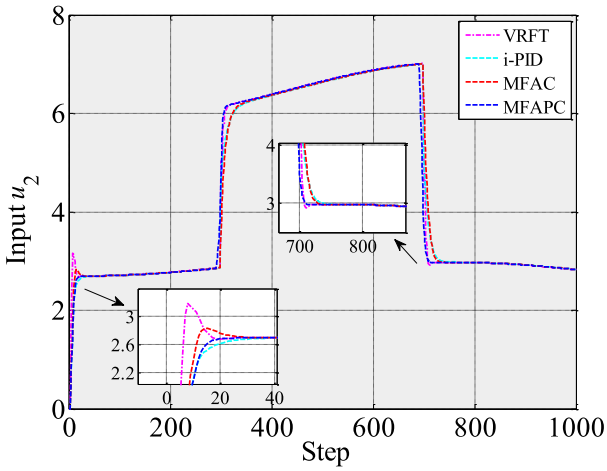

FIGURE 5. Control input u_2 with a constant desired signal.

TABLE 2. The overshoot of y_1.

	MFAPC	MFAC	iPID	VRFT
Overshoot	0.164%	2.74%	0.184%	3.248%

TABLE 3. The overshoot of y_2.

	MFAPC	MFAC	i-PID	VRFT
Overshoot	0.103%	3.116%	0.112%	6.04%

the intelligent PID algorithm and the VRFT algorithm need to design the controllers separately for each output, which leads to improper addressing of system coupling problems, while the MFAPC algorithm can realize the decoupling of the MIMO system by estimating the PJM.

It should be pointed out that the parameters of VRFT algorithm are offline and not adaptive, so the parameter tuning is time consuming and difficult. In contrast, the MFAC and MFAPC algorithms estimate parameters based on online I/O data, and the parameter tuning is simpler and more convenient. In addition, compared with MFAC algorithm, MFAPC algorithm makes full use of future input and output information and has better control performance.

TABLE 4. The running time.

	MFAPC	MFAC	i-PID	VRFT
Time	0.3601	0.0623	0.0482	0.0103

Remark 7: Considering the computational complexity and computing time of MFAPC algorithm, we calculate the running time of four algorithms respectively, as shown in Table 4.

Since the parameters of MFAPC algorithm are updated online, involving the estimation and prediction of PJM, the calculation time of MFAPC algorithm is longer than that of other three algorithms. This is a disadvantage of the MFAPC algorithm.

The above time contains 1000 sampling times of the algorithm, so it can be obtained that the MFAPC algorithm runs for one cycle with a time of 0.3601 ms.

Therefore, the MFAPC algorithm has the advantage of low computational complexity, because it only involves the operation of addition, subtraction, multiplication and division. Specifically, if the control period required by the controlled object is greater than the calculation time of the algorithm, which is 0.3601 ms, the algorithm can run on embedded controllers, such as 8-bit controllers, high-performance controller, etc. The choice of the controller depends on the accuracy requirement of the controlled object. For low-speed systems, such as water tank control and other process control systems, with low sampling time and control accuracy requirements, controllers with low-configuration can be applied, such as 8-bit controllers. For high-speed systems, such as motor control systems, high-performance controllers should be used.

2) TIME-VARYING DESIRED SIGNAL

In this case, the desired signal becomes a time-varying signal,

$$\begin{cases} y_1^*(k) = 0.5 + 0.25\cos(0.25\pi k/100) + 0.25\sin(0.5\pi k/100), \\ y_2^*(k) = 0.5 + 0.25\sin(0.25\pi k/100) + 0.25\cos(0.5\pi k/100). \end{cases} \tag{36}$$

The parameter setting of the four algorithms are shown in Table 5.

The simulations are presented in Figure 6-9, and the performance indicators are shown in Table 6-7 with the root mean square of error and the total sum of squares of control inputs included, described as follows.

The root mean square of error:

$$e_{RMS} = \sqrt{(\sum_{k=1}^{N} e(k)^2)/N}.$$

The total sum of squares of control inputs:

$$\Delta u_{TSS} = \sum_{k=1}^{N} \Delta u(k)^2.$$

TABLE 5. the parameter setting of algorithms.

MFAPC	$\eta = 1, \rho = 1, \mu = 1, \lambda = 4,$ $\boldsymbol{\Phi}_c(1) = \boldsymbol{\Phi}_c(2) = \boldsymbol{\Phi}_c(3) = \boldsymbol{\Phi}_c(4) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}^T$.
MFAC	$\eta = 1, \rho = 1, \mu = 1, \lambda = 0.5,$ $\boldsymbol{\Phi}_c(1) = \boldsymbol{\Phi}_c(2) = \boldsymbol{\Phi}_c(3) = \boldsymbol{\Phi}_c(4) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}^T$.
i-PID	$\alpha_1 = 0.98, k_{p1} = 1.13, k_{i1} = 0.4,$ $\alpha_2 = 0.98, k_{p2} = 1.12, k_{i2} = 0.38.$
VRFT	$k_{p1} = 0.9, k_{i1} = 0.56, k_{d1} = 1.1,$ $k_{p2} = 1.1, k_{i2} = 0.98, k_{d2} = 0.9.$

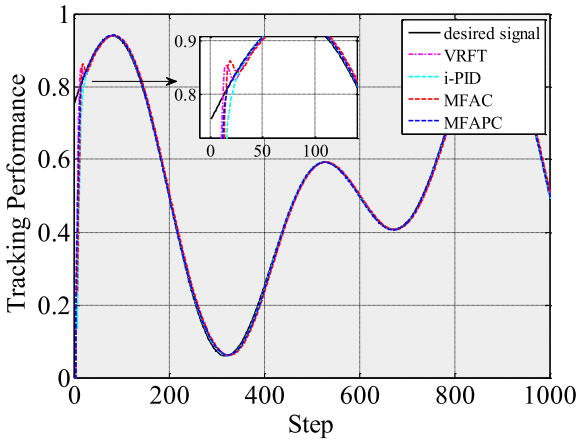

FIGURE 6. Tracking performance of y_1 with a time-varying desired signal.

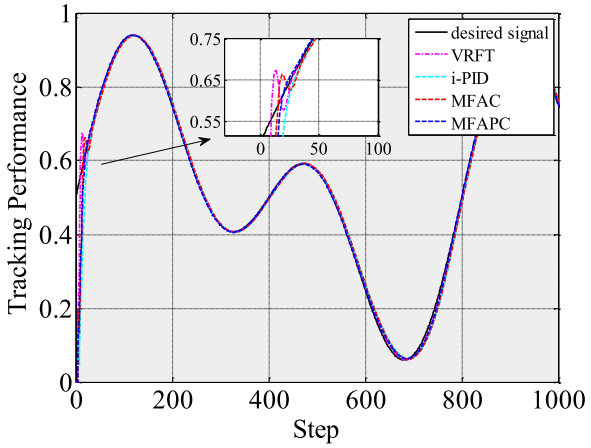

FIGURE 7. Tracking performance of y_2 with a time-varying desired signal.

TABLE 6. Numerical indicators of y_1.

	MFAPC	MFAC	i-PID	VRFT
Δu_{TSS}	0.3295	0.3509	0.3114	0.4909
e_{MKS}	0.0375	0.0475	0.0615	0.0470

The tracking performance results are shown in Figure 6-7 and Table 6-7, which show that the four algorithms above are effective, and all algorithms have good tracking effect

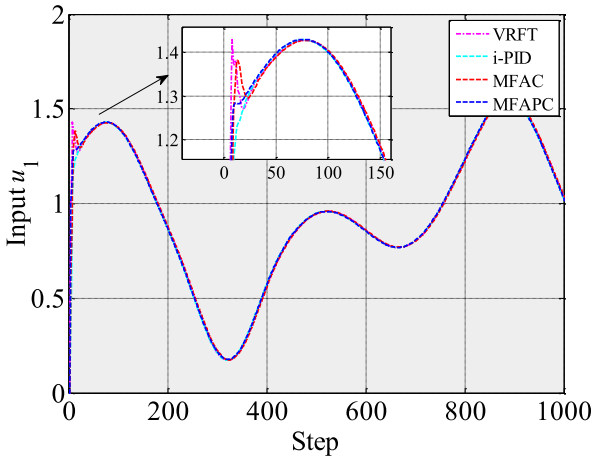

FIGURE 8. Control input u_1 with a time-varying desired signal.

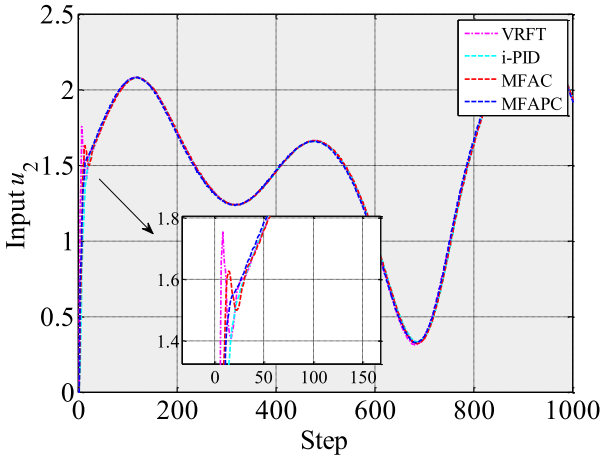

FIGURE 9. Control input u_2 with a time-varying desired signal.

TABLE 7. Numerical indicators of y_2.

	MFAPC	MFAC	i-PID	VRFT
Δu_{TSS}	0.3419	0.3708	0.3151	0.7607
e_{MKS}	0.0339	0.0349	0.0432	0.0351

after 100 steps. For MIMO nonlinear time-varying system, the performance of MFAPC algorithm is superior to other three algorithms.

The figures 6-7 show clearly that the VRFT and MFAC algorithms have obvious oscillations at the beginning of tracking, while i-PID algorithm and MFAPC algorithm has no overshoot, but convergence speed of i-PID algorithm is slower than MFAPC algorithm. Compared with the other three algorithms, MFAPC algorithm has faster convergence speed, and the tracking performance has non-overshoot and a smoother transient process. Therefore, the MFAPC algorithm has better tracking performance. Moreover, containing future output information, the MFAPC algorithm makes good control decisions in a period of time.

Considering the numerical index, the Δu_{TSS} reflects the change rate of the control input in a certain extent. The smaller the value of Δu_{TSS}, the smoother the control input curve. In addition, Δu_{TSS} also reflects to some extent the energy consumed by the application of the algorithm in practice. The smaller the Δu_{TSS}, the less energy is consumed by applying the control algorithm. As can be seen from Table 6-7, the i-PID algorithm has the lowest Δu_{TSS} value, which corresponds to the low overshoot of i-PID algorithm. However, the convergence speed of the algorithm is relatively slow. The Δu_{TSS} values of the MFAC algorithm and the VRFT algorithm are relatively large. Therefore, it can be said that the MFAPC algorithm combines the advantages of good convergence speed and low control input rate. e_{RMS} reflects the overall tracking effect of the control algorithm in a certain extent. The smaller the e_{RMS} value, the better the overall tracking effect of the algorithm. As can be seen from Table 6-7, the tracking effect of MFAPC algorithm is significantly better than the other three algorithms.

B. ACTUAL SYSTEM SIMULATION

Simulation example 2 is based on a two-degree-of-freedom manipulator system [36]. The applicability of this method in practical nonlinear systems is proved.

Example 2: [36]: The discrete nonlinear system model of the two degree of freedom manipulator is expressed as follows.

$$\begin{cases} y_1(k+1) = y_1(k) + T_s * \Big[a_{24}{x_4}^2(k) + a_{22}{x_2}^2(k) \\ \qquad +a_{224}x_2(k)x_4(k) + b_{21}u_1(k) + b_{22}u_2(k)\Big], \\ y_2(k+1) = y_2(k) + T_s * \Big[a_{42}{x_2}^2(k) + a_{44}{x_4}^2(k) \\ \qquad +a_{424}x_2(k)x_4(k) + b_{41}u_1(k) + b_{42}u_2(k)\Big]. \end{cases} \tag{37}$$

where

$$\begin{cases} H_{11} = m_1 l_{c1} + I_1 + m_2[l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos\theta_2] + I_2, \\ H_{22} = m_2 l_{c2}^2 + I_2, \\ H_{12} = m_2 l_1 l_{c2}\cos\theta_2 + m_2 l_{c2}^2 + I_2, \\ h = m_2 l_{c2}\sin\theta_2, \\ a_{22} = \dfrac{H_{12}h}{H_{11}H_{22} - H_{12}^2}, a_{24} = \dfrac{H_{22}h}{H_{11}H_{22} - H_{12}^2}, \\ a_{42} = \dfrac{-H_{11}h}{H_{11}H_{22} - H_{12}^2}, a_{44} = \dfrac{-H_{12}h}{H_{11}H_{22} - H_{12}^2}, \\ a_{224} = \dfrac{2H_{22}h}{H_{11}H_{22} - H_{12}^2}, a_{424} = \dfrac{-2H_{12}h}{H_{11}H_{22} - H_{12}^2}, \\ b_{21} = \dfrac{H_{22}h}{H_{11}H_{22} - H_{12}^2}, b_{22} = \dfrac{-H_{12}}{H_{11}H_{22} - H_{12}^2}, \\ b_{41} = \dfrac{-H_{12}}{H_{11}H_{22} - H_{12}^2}, b_{42} = \dfrac{H_{11}}{H_{11}H_{22} - H_{12}^2}. \end{cases} \tag{38}$$

In this simulation experiment, according to the physical structure of the two-degree-of-freedom manipulator, the parameters are set as follows.

$$\begin{cases} l_1 = l_2 = 0.5, \quad l_{c1} = l_{c2} = 0.25, \\ I_1 = I_2 = 0.1, \quad m_1 = m_2 = 1, \end{cases} \tag{39}$$

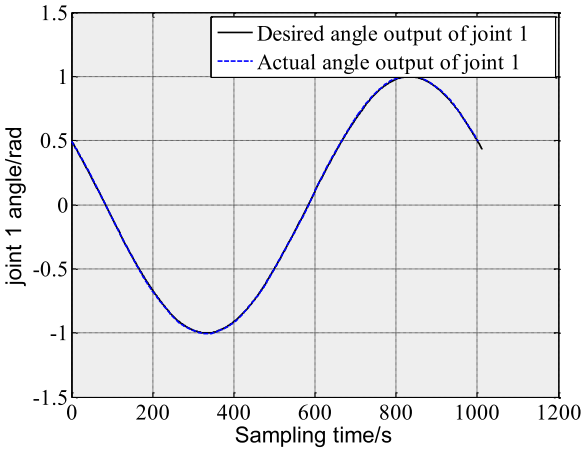

FIGURE 10. Tracking performance of y_1 using MFAPC algorithm.

where l_1, l_2 represents the length of arms 1, 2 of the manipulator, respectively. l_{c1}, l_{c2} represents the distance from the center of mass 1 and 2 to the start of the arms 1, 2, respectively. I_1, I_2 are the moment of inertia of the manipulator arms around the center of mass. m_1, m_2 are the mass of the arms 1, 2, respectively. The joint angles of the manipulator are denoted as θ_1, θ_2, respectively.

The desired output angular velocity is set to

$$\begin{cases} y_1^*(k) = \pi \cdot \cos(\pi k T_s + \pi/3), \\ y_2^*(k) = -\pi \cdot \sin(\pi k T_s + \pi/3). \end{cases} \tag{40}$$

The system initial condition is

$$y_1(i) = 3.14, y_2(i) = 0.1, \quad i = 1, 2, 3, 4.$$
$$u_1(j) = u_2(j) = 0.1, \quad j = 1, 2, 3, 4.$$

The initial values of system parameters are set as below,

$$\eta = 1, \quad \rho = 1, \ \mu = 1.5, \ \lambda = 4.5,$$
$$\boldsymbol{\Phi}_c(1) = \boldsymbol{\Phi}_c(2) = \boldsymbol{\Phi}_c(3) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T.$$

This simulation uses MFAPC control algorithm to control the two-degree-of-freedom manipulator, and the results prove the applicability of this algorithm in the actual system as presented in Figure 10-13.

The simulation proves that the proposed algorithm has an excellent control effect on the two degree of freedom manipulator system. From Figure 10 and Figure 11, it appears that MFAPC algorithm achieves a good tracking effect for the system output of each degree of freedom. The motion trajectory of the manipulator is highly consistent with the desired trajectory as presented in Figure 12. The tracking error in Figure 13 is maintained below 0.015, which illustrates the effectiveness of the MFAPC algorithm in actual systems. In addition, this simulation also proves that MFAPC algorithm has strong decoupling ability for MIMO nonlinear systems.

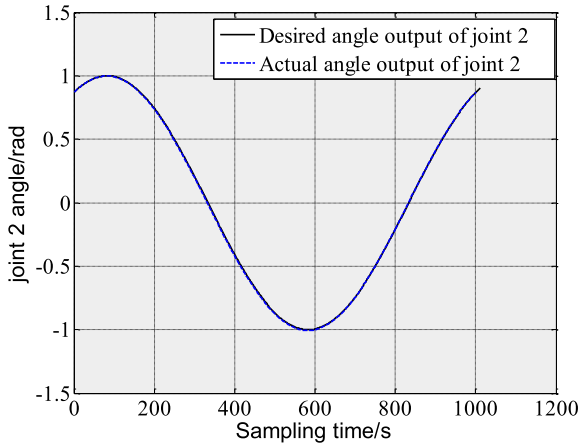

FIGURE 11. Tracking performance of y_2 using MFAPC algorithm.

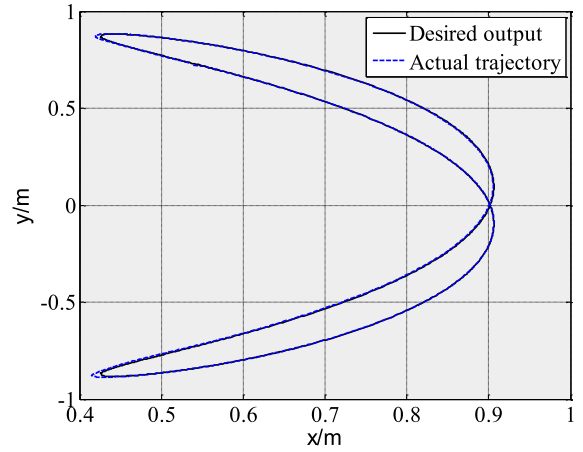

FIGURE 12. The actual tracking trajectory of the two-degree-of-freedom manipulator.

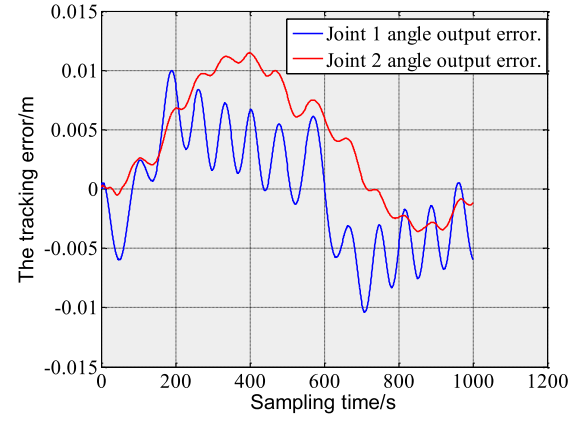

FIGURE 13. Tracking errors of the two-degree-of-freedom manipulator.

V. CONCLUSION

In this work, a novel model-free adaptive predictive control method for a MIMO nonlinear system is proposed. This algorithm is based on the equivalent dynamic linearization model and the Pseudo-Jacobian matrix. The characteristic of this algorithm is that it merely needs the system I/O data for the design of the controller. Moreover, due to the addition of the idea of predictive control, the future input and output information are introduced, so that this algorithm has excellent tracking performance and strong robustness. Meanwhile, this algorithm can also realize the elimination of coupling of MIMO nonlinear system. Through a series of simulations, the effectiveness of the algorithm is verified. At the same time, through a series of reasonable assumptions, the convergence and stability of the algorithm are proved by strict mathematical analysis.

APPENDIX A
PROOF OF THEOREM 2

The proof of this part is divided into two steps. One is the proof of the boundedness of PJM estimation, followed by the proof of the convergence of the tracking error and the BIBO stability of the MFAPC system.

Step 1: We will prove the boundedness of $\hat{\boldsymbol{\Phi}}(k)$ and the PJM predictions $\hat{\boldsymbol{\Phi}}(k+1), \cdots, \hat{\boldsymbol{\Phi}}(k+N_u-1)$.

Let $\hat{\boldsymbol{\Phi}}(k) = [\hat{\boldsymbol{\phi}}_1^T(k), \cdots, \hat{\boldsymbol{\phi}}_m^T(k)]^T$, $\hat{\boldsymbol{\phi}}_i(k) = [\hat{\phi}_{i1}(k), \cdots, \hat{\phi}_{im}(k)]$, $i = 1, 2, \cdots, m$. The PJM $\hat{\boldsymbol{\Phi}}(k)$ estimation algorithm (18) can be rewritten as:

$$\hat{\boldsymbol{\phi}}_i(k) = \hat{\boldsymbol{\phi}}_i(k-1) + \frac{\eta(\Delta \boldsymbol{y}_i(k) - \hat{\boldsymbol{\phi}}_i(k-1)\Delta \boldsymbol{u}(k-1))\Delta \boldsymbol{u}^T(k-1)}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}, \tag{41}$$

where $\Delta \boldsymbol{y}_i(k) = \boldsymbol{\phi}_i(k-1)\Delta \boldsymbol{u}(k-1)$, $i = 1, 2, \cdots m$.

Defining the estimation error $\tilde{\boldsymbol{\phi}}_i(k) = \hat{\boldsymbol{\phi}}_i(k) - \boldsymbol{\phi}_i(k)$. Subtracting $\boldsymbol{\phi}_i(k)$ from both sides of equation (41):

$$\tilde{\boldsymbol{\phi}}_i(k) = \tilde{\boldsymbol{\phi}}_i(k-1)[\boldsymbol{I} - \frac{\eta \Delta \boldsymbol{u}(k-1)\Delta \boldsymbol{u}^T(k-1)}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}] + \boldsymbol{\phi}_i(k-1) - \boldsymbol{\phi}_i(k). \tag{42}$$

According to theorem 1, $\|\boldsymbol{\Phi}(k)\|$ is bounded. Suppose there is a constant $\bar{b}$, such that $\|\boldsymbol{\Phi}(k)\| \leq \bar{b}$.

Therefore, $\left\|\boldsymbol{\phi}_i(k-1) - \boldsymbol{\phi}_i(k)\right\| \leq 2\bar{b}$.

Taking the norm on both sides of (42) leads to:

$$\begin{aligned} \left\|\tilde{\boldsymbol{\phi}}_i(k)\right\| &\leq \left\|\tilde{\boldsymbol{\phi}}_i(k-1)[\boldsymbol{I} - \frac{\eta \Delta \boldsymbol{u}(k-1)\Delta \boldsymbol{u}^T(k-1)}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}]\right\| \\ &\quad + \left\|\boldsymbol{\phi}_i(k-1) - \boldsymbol{\phi}_i(k)\right\| \\ &\leq \left\|\tilde{\boldsymbol{\phi}}_i(k-1)[\boldsymbol{I} - \frac{\eta \Delta \boldsymbol{u}(k-1)\Delta \boldsymbol{u}^T(k-1)}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}]\right\| + 2\bar{b}. \end{aligned} \tag{43}$$

Extract the first item on the right of (43) and square it:

$$\begin{aligned} &\left\|\tilde{\boldsymbol{\phi}}_i(k-1)[\boldsymbol{I} - \frac{\eta \Delta \boldsymbol{u}(k-1)\Delta \boldsymbol{u}^T(k-1)}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}]\right\|^2 \\ &= \left\|\tilde{\boldsymbol{\phi}}_i(k-1)\right\|^2 + \left[-2 + \frac{\eta \|\Delta \boldsymbol{u}(k-1)\|^2}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}\right] \\ &\quad \times \frac{\eta \left\|\tilde{\boldsymbol{\phi}}_i(k-1)\Delta \boldsymbol{u}(k-1)\right\|^2}{\mu + \|\Delta \boldsymbol{u}(k-1)\|^2}. \end{aligned} \tag{44}$$

Since $0 < \eta \leq 2$ and $\mu > 0$, the following inequalities can be obtained:

$$-2 + \frac{\eta \left\|\Delta \boldsymbol{u}(k-1)\right\|^2}{\mu + \left\|\Delta \boldsymbol{u}(k-1)\right\|^2} < 0, \tag{45}$$

According to (44) and (45), a constant d_1 must exist, which satisfies $0 < d_1 < 1$, such that the inequality holds as below:

$$\left\|\tilde{\boldsymbol{\phi}}_i(k-1)[\boldsymbol{I} - \frac{\eta \Delta \boldsymbol{u}(k-1)\Delta \boldsymbol{u}^T(k-1)}{\mu + \left\|\Delta \boldsymbol{u}(k-1)\right\|^2}]\right\| \leq d_1 \left\|\tilde{\boldsymbol{\phi}}_i(k-1)\right\|. \tag{46}$$

Then, substituting (46) into (43) yields:

$$\begin{aligned}\left\|\tilde{\boldsymbol{\phi}}_i(k)\right\| &\leq d_1 \left\|\tilde{\boldsymbol{\phi}}_i(k-1)\right\| + 2\bar{b} \leq d_1^2 \left\|\tilde{\boldsymbol{\phi}}_i(k-2)\right\| + 2d_1\bar{b} + 2\bar{b} \\ &\leq \cdots \leq d_1^{k-1} \left\|\tilde{\boldsymbol{\phi}}_i(1)\right\| + \frac{2\bar{b}}{1-d_1}.\end{aligned} \tag{47}$$

The inequality (47) demonstrate that $\tilde{\boldsymbol{\phi}}_i(k)$ is bounded. As $\boldsymbol{\phi}_i(k)$ is bounded according to theorem 1, so $\hat{\boldsymbol{\phi}}_i(k)$ is bounded, $\hat{\boldsymbol{\Phi}}(k)$ is bounded too. The boundedness of the PJM predictions values $\hat{\boldsymbol{\Phi}}(k+1), \cdots, \hat{\boldsymbol{\Phi}}(k+N_u-1)$ is the direct result of algorithms (21)-(25).

Step 2: The convergence of the tracking error and the BIBO stability will be demonstrated respectively.

Firstly, define the tracking error as $\boldsymbol{e}(k+1) = \boldsymbol{y}^* - \boldsymbol{y}(k+1)$.

Substituting (3) into the tracking error equation and using (26)-(27), we have

$$\begin{aligned}\boldsymbol{e}(k+1) &= \boldsymbol{y}^* - \boldsymbol{y}(k+1) = \boldsymbol{y}^* - \boldsymbol{y}(k) - \boldsymbol{\Phi}(k)\Delta \boldsymbol{u}(k) \\ &= \boldsymbol{y}^* - \boldsymbol{y}(k) - \boldsymbol{\Phi}(k) \\ &\quad \times \left[\frac{\boldsymbol{g}^T \hat{\boldsymbol{A}}_1^T(k)\left[\boldsymbol{Y}_{Nm}^*(k+1) - \boldsymbol{E}(k)\boldsymbol{y}(k)\right]}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right] \\ &= \left[\boldsymbol{I} - \boldsymbol{\Phi}(k)\frac{\left(\boldsymbol{g}^T \hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k)\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right]\boldsymbol{e}(k).\end{aligned} \tag{48}$$

Taking the absolute value on the two sides of (48) yields:

$$\left\|\boldsymbol{e}(k+1)\right\| \leq \left\|\boldsymbol{I} - \boldsymbol{\Phi}(k)\frac{\left(\boldsymbol{g}^T \hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k)\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right\| \times \left\|\boldsymbol{e}(k)\right\|.$$

$$\hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k) = \begin{bmatrix} N \cdot \hat{\boldsymbol{\Phi}}^T(k) \\ (N-1) \cdot \hat{\boldsymbol{\Phi}}^T(k+1) \\ \vdots \\ (N-N_u+1) \cdot \hat{\boldsymbol{\Phi}}^T(k+N_u-1) \end{bmatrix}_{N_u m \times m}, \tag{49}$$

thus,

$$\boldsymbol{g}^T \hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k) = N \cdot \hat{\boldsymbol{\Phi}}^T(k),$$

Then, (49) can be rewritten as:

$$\left\|\boldsymbol{e}(k+1)\right\| \leq \left\|\boldsymbol{I} - \frac{\boldsymbol{\Phi}(k) \cdot N \cdot \hat{\boldsymbol{\Phi}}^T(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right\| \times \left\|\boldsymbol{e}(k)\right\|.$$

From lemma 1, we assumption z is the eigenvalue of the matrix: $I - \frac{\boldsymbol{\Phi}(k) \cdot N \cdot \hat{\boldsymbol{\Phi}}^T(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}$. so, we have

$$\begin{aligned}D_i &= \left\{ z \left| z - \left| 1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \right| \\ &\quad \leq \sum_{l=1, l\neq i}^{m} \left| \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \Bigg\},\end{aligned} \tag{50}$$

Using triangle inequality, (50) can be rewritten as (51).

$$\begin{aligned}D_j = \{z \,||z| \;&\leq \left| 1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \\ &+ \sum_{l=1, l\neq i}^{m} \left| \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \Bigg\}.\end{aligned} \tag{51}$$

From resetting algorithms (19) and (20), we have $b_2 \leq \left|\hat{\phi}_{ii}(k)\right| \leq \alpha b_2$, and $\left|\hat{\phi}_{ij}(k)\right| \leq b_1, i = 1, 2, \cdots m, j = 1, 2, \cdots m, \; i \neq j$, assumption 3 gives that $b_2 \leq |\phi_{ii}(k)| \leq \alpha b_2$, $|\phi_{ij}(k)| \leq b_1, i = 1, 2, \cdots m, j = 1, 2, \cdots m, i \neq j$.

Thus, the following two inequalities hold:

$$\begin{aligned}&1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\ &\quad \leq 1 - \frac{N \cdot |\phi_{ii}(k)| \left|\hat{\phi}_{ii}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\ &\quad \leq 1 - \frac{N \cdot b_2^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}.\end{aligned} \tag{52}$$

$$\sum_{l=1, l\neq i}^{m} \left| \frac{N \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right|$$

$$
\begin{aligned}
&\le N \sum_{l=1,l\neq i}^{m} \frac{\sum_{j=1}^{m} |\phi_{ij}(k)| \left|\hat{\phi}_{lj}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&= N \sum_{l=1,l\neq i}^{m} \frac{|\phi_{ii}(k)| \left|\hat{\phi}_{li}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} + N \sum_{l=1,l\neq i}^{m} \frac{|\phi_{il}(k)| \left|\hat{\phi}_{ll}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&\quad + N \sum_{l=1,l\neq i}^{m} \frac{\sum_{j=1,j\neq i,l}^{m} |\phi_{ij}(k)| \left|\hat{\phi}_{lj}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&\le \frac{N(m-1)\cdot \alpha b_2 \cdot b_1}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} + \frac{N(m-1)\cdot b_1 \cdot \alpha b_2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&\quad + \frac{N(m-1)(m-2)\cdot b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&= N \frac{2(m-1)\cdot \alpha b_2 \cdot b_1 + (m-1)(m-2)\cdot b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \qquad (53)
\end{aligned}
$$

From assumption 1, we have $b_2 > b_1(2\alpha+1)(m-1)$. Summing (52) and (53) yields, we can get (54).

$$
\begin{aligned}
&1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} + \sum_{l=1,l\neq i}^{m} \left| \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \\
&\le 1 - N \cdot \frac{b_2^2 - 2(m-1)\cdot \alpha b_2 \cdot b_1 + (m-1)(m-2)\cdot b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&\le 1 - N \cdot \frac{b_2(b_2 - 2(m-1)\cdot \alpha b_1) + (m-1)(m-1)\cdot b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&= 1 - N \cdot \frac{b_2 b_1 (m-1) + (m-1)^2 b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&\le 1 - N \cdot \frac{2\alpha(m-1)^2 b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \qquad (54)
\end{aligned}
$$

By resetting algorithm and assumption, $\phi_{ij}(k)\hat{\phi}_{ij}(k) > 0, i = 1, 2, \cdots, m; \quad j = 1, 2, \cdots, m$. Thus, there exists a constant $\lambda_{\min} > 0$, such that the inequality (55) holds for any $\lambda > \lambda_{\min}$:

$$
\begin{aligned}
&\frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&= N \cdot \frac{\sum_{j=1}^{m} |\phi_{ij}(k)| \left|\hat{\phi}_{ij}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&= \frac{N \cdot |\phi_{ii}(k)| \left|\hat{\phi}_{ii}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} + N \cdot \frac{\sum_{j=1,j\neq i}^{m} |\phi_{ij}(k)| \left|\hat{\phi}_{ij}(k)\right|}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&\le \frac{N \cdot (\alpha^2 b_2^2 + (m-1)b_1^2)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} < \frac{N \cdot (\alpha^2 b_2^2 + (m-1)b_1^2)}{\lambda_{\min} + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} < 1. \qquad (55)
\end{aligned}
$$

Therefore, the following inequality can be obtained.

$$
1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} > 0 \qquad (56)
$$

Thus, according to inequality (54) and (56), the following inequality holds for any $\lambda > \lambda_{\min}$:

$$
\begin{aligned}
&\left| 1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| + \sum_{l=1,l\neq i}^{m} \left| \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \\
&= 1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} + \sum_{l=1,l\neq i}^{m} \left| \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \\
&< 1 - N \cdot \frac{2\alpha(m-1)^2 b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}. \qquad (57)
\end{aligned}
$$

So, there exists inequality as below,

$$
\begin{aligned}
0 < M_1 &\le \frac{N \cdot 2\alpha(m-1)^2 b_1^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} < \frac{N \cdot b_2^2}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&< \frac{N \cdot \left(\alpha^2 b_2^2 + (m-1)b_1^2\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\
&< \frac{N \cdot \left(\alpha^2 b_2^2 + (m-1)b_1^2\right)}{\lambda_{\min} + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} < 1. \qquad (58)
\end{aligned}
$$

Then,

$$
\begin{aligned}
&\left| 1 - \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{ij}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| + \sum_{l=1,l\neq i}^{m} \left| \frac{N \cdot \sum_{j=1}^{m} \phi_{ij}(k)\hat{\phi}_{lj}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right| \\
&< 1 - M_1 < 1. \qquad (59)
\end{aligned}
$$

According to the Gerschgorin disk and inequality (59), one obtains:

$$
s\left[\boldsymbol{I} - \boldsymbol{\Phi}(k) \frac{\left(\boldsymbol{g}^T \hat{\boldsymbol{A}}_1^T(k) \boldsymbol{E}(k)\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right] < 1 - M_1 < 1.
$$

where $s[\boldsymbol{A}]$ denotes the spectral radius of matrix $\boldsymbol{A}$, that is, $s(\boldsymbol{A}) = \max_{i\in\{1,2,\cdots,m\}} |z_i|$, and $z_i, i = 1, 2, \cdots, m$ is the eigenvalue of $\boldsymbol{A}$.

Using the conclusion on spectral radius in Reference [37], there must exist a constant $\varepsilon_1 > 0$ which can be made arbitrarily small such that:

$$\begin{aligned}\left\| \boldsymbol{I} - \boldsymbol{\Phi}(k)\frac{\left(\boldsymbol{g}^T\hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k)\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right\|_v &< s\left[\boldsymbol{I} - \boldsymbol{\Phi}(k)\frac{\left(\boldsymbol{g}^T\hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k)\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right] + \varepsilon_1 \\ &\le 1 - M_1 + \varepsilon_1 < 1. \end{aligned} \tag{60}$$

where $\|\boldsymbol{A}\|_v$ describes the consistent norm of matrix $\boldsymbol{A}$.

Taking norms on each side of (60) yields, and defining $d_2 = 1 - M_1 + \varepsilon_1$.

$$\begin{aligned}\|\boldsymbol{e}(k+1)\|_v &\le \left\| \boldsymbol{I} - \boldsymbol{\Phi}(k)\frac{\left(\boldsymbol{g}^T\hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k)\right)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\right\|_v \times \|\boldsymbol{e}(k)\|_v \\ &\le d_2 \|\boldsymbol{e}(k)\|_v \le \cdots \le d_2^k \|\boldsymbol{e}(1)\|_v . \end{aligned} \tag{61}$$

Conclusion (1) of theorem 2 can be proved directly through (61).

Because of the boundedness of $\boldsymbol{y}^*$ and $\boldsymbol{e}(k)$, $\boldsymbol{y}(k)$ is also bounded.

$$\begin{aligned}\Delta\boldsymbol{u}(k) &= \boldsymbol{g}^T\Delta\boldsymbol{U}_{N_u m}(k) \\ &= \frac{\boldsymbol{g}^T\hat{\boldsymbol{A}}_1^T(k)\left[\boldsymbol{Y}_{Nm}^*(k+1) - \boldsymbol{E}(k)\boldsymbol{y}(k)\right]}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \\ &= \frac{\boldsymbol{g}^T\hat{\boldsymbol{A}}_1^T(k)\boldsymbol{E}(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\boldsymbol{e}(k) = \frac{N\cdot\hat{\boldsymbol{\Phi}}^T(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2}\boldsymbol{e}(k). \end{aligned} \tag{62}$$

Since $\hat{\boldsymbol{\Phi}}(k)$ is bounded, $\hat{\boldsymbol{A}}_1(k)$ is bounded as well, a positive constant M_2 exists so that (63) holds:

$$\left\| \frac{N\cdot\hat{\boldsymbol{\Phi}}^T(k)}{\lambda + \left\|\hat{\boldsymbol{A}}_1(k)\right\|^2} \right\| < M_2. \tag{63}$$

Therefore, the inequalities can be obtains as:

$$\|\Delta\boldsymbol{u}(k)\| \le M_2 \|\boldsymbol{e}(k)\| .$$

So, we have the following inequality that is:

$$\begin{aligned}\|\boldsymbol{u}(k)\|_v &\le \|\Delta\boldsymbol{u}(k)\|_v + \|\Delta\boldsymbol{u}(k-1)\|_v \\ &\quad + \cdots + \|\Delta\boldsymbol{u}(1)\|_v + \|\boldsymbol{u}(0)\|_v \\ &= M_2(\|\boldsymbol{e}(k)\|_v + \|\boldsymbol{e}(k-1)\|_v \\ &\quad + \cdots + \|\boldsymbol{e}(1)\|_v) + \|\boldsymbol{u}(0)\|_v \\ &< M_2\frac{\|\boldsymbol{e}(1)\|_v}{1-d_2} + \|\boldsymbol{u}(0)\|_v . \end{aligned} \tag{64}$$

Thus, conclusion (2) of theorem 2 is obtained.

REFERENCES

[1] Z. Hou and S. Jin, *Model Free Adaptive Control: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2013.

[2] C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan, and S. Xie, "Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics," *IEEE Trans. Autom. Control*, to be published. doi: 10.1109/TAC.2019.2905215.

[3] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, Jun. 2003.

[4] G. O. Guardabassi and S. M. Savaresi, "Virtual reference direct design method: An off-line approach to data-based control system design," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 954–959, May 2000.

[5] Y. Jiang, Y. Zhu, K. Yang, C. Hu, and D. Yu, "A data-driven iterative decoupling feedforward control strategy with application to an ultraprecision motion stage," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 620–627, Jan. 2015.

[6] M.-B. Radac, R.-E. Precup, and R.-C. Roman, "Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-Learning," *ISA Trans.*, vol. 73, pp. 227–238, Feb. 2018.

[7] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[8] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Inf. Sci.*, vol. 235, pp. 3–35, Jun. 2013.

[9] H. Ji, Z. Hou, L. Fan, and F. L. Lewis, "Adaptive iterative learning reliable control for a class of non-linearly parameterised systems with unknown state delays and input saturation," *IET Control Theory Appl.*, vol. 10, no. 17, pp. 2160–2174, Nov. 2016.

[10] M. Yu and C. Li, "Robust adaptive iterative learning control for discrete-time nonlinear systems with time-iteration-varying parameters," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1737–1745, Jul. 2017.

[11] A. Karimi and G. Galdos, "Fixed-order H-infinity controller design for nonparametric models by convex optimization," *Automatica*, vol. 46, no. 8, pp. 1388–1394, Aug. 2010.

[12] H. Hjalmarsson, "Iterative feedback tuning—An overview," *Int. J. Adapt. Control Signal Process.*, vol. 16, no. 5, pp. 373–395, Jun. 2002.

[13] R. Chi, Z. Hou, S. Jin, D. Wang, and J. Hao, "A data-driven iterative feedback tuning approach of ALINEA for freeway traffic ramp metering with PARAMICS simulations," *IEEE Trans Ind. Informat.*, vol. 9, no. 4, pp. 2310–2317, Nov. 2013.

[14] R.-E. Precup, M.-B. Radac, R.-C. Roman, and E. M. Petriu, "Model-free sliding mode control of nonlinear systems: Algorithms and experiments," *Inf. Sci.*, vol. 381, pp. 176–192, Mar. 2017.

[15] Q. Wu, H. Li, F. Meng, and K. N. Ngan, "Generic proposal evaluator: A lazy learning strategy toward blind proposal quality assessment," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 306–319, Jan. 2018.

[16] Z. Hou, "The parameter identification, adaptive control and model free learning adaptive control for nonlinear systems," Ph.D. dissertation, Northeastern Univ., Shenyang, China, 1994.

[17] Y. Zhu and Z. S. Hou, "Controller dynamic linearisation-based model-free adaptive control framework for a class of non-linear system," *IET Control Theory Appl.*, vol. 9, no. 7, pp. 1162–1172, Apr. 2015.

[18] S. Liu, Z. Hou, and J. Zheng, "Attitude adjustment of quadrotor aircraft platform via a data-driven model free adaptive control cascaded with intelligent PID," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2016, pp. 4971–4976.

[19] R.-C. Roman, M.-B. Radac, and R.-E. Precup, "Multi-input—Multi-output system experimental validation of model-free control and virtual reference feedback tuning techniques," *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1395–1403, Aug. 2016.

[20] D. Xu, Y. Shi, and Z. Ji, "Model free adaptive discrete-time integral sliding mode constrained control for autonomous 4WMV parking systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 834–843, Jan. 2018.

[21] C. Lu, Y. Zhao, K. Men, L. Tu, and Y. Han, "Wide-area power system stabiliser based on model-free adaptive control," *IET Control Theory Appl.*, vol. 9, no. 13, pp. 1996–2007, Aug. 2015.

[22] D. Xu, B. Jiang, and P. Shi, "A novel model-free adaptive control design for multivariable industrial processes," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6391–6398, Nov. 2014.

[23] X. Wang, X. Li, J. Wang, X. Fang, and X. Zhu, "Data-driven model-free adaptive sliding mode control for the multi degree-of-freedom robotic exoskeleton," *Inf. Sci.*, vol. 327, pp. 246–257, Jan. 2016.
[24] Z. Cheng, Z. Hou, and S. Jin, "MFAC-based balance control for freeway and auxiliary road system with multi-intersections," in *Proc. 10th Asian Control Conf. (ASCC)*, Kota Kinabalu, Malaysia, Jun. 2015, pp. 1–6.
[25] L. Duan, Z. Hou, X. Yu, S. Jin, and K. Lu, "Data-driven model-free adaptive attitude control approach for launch vehicle with virtual reference feedback parameters tuning method," *IEEE Access*, vol. 7, pp. 54106–54116, 2019.
[26] S. Jin, Z. Hou, R. Chi, and X. Bu, "Model free adaptive predictive control approach for phase splits of urban traffic network," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2016, pp. 5750–5754.
[27] Z. Hou, S. Liu, and T. Tian, "Lazy-learning-based data-driven model-free adaptive predictive control for a class of discrete-time nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 28, no. 8, pp. 1914–1928, Aug. 2017.
[28] Z. S. Hou, S. Liu, and C. Yin, "Local learning-based model-free adaptive predictive control for adjustment of oxygen concentration in syngas manufacturing industry," *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1384–1394, Aug. 2016.
[29] C. Chen, F. L. Lewis, S. Xie, H. Modares, Z. Liu, S. Zuo, and A. Davoudi, "Resilient adaptive and H∞ controls of multi-agent systems under sensor and actuator faults," *Automatica*, vol. 102, pp. 19–26, Jan. 2019.
[30] C. Chen, K. Xie, F. L. Lewis, S. Xie, and A. Davoudi, "Fully distributed resilience for adaptive exponential synchronization of heterogeneous multi-agent systems against actuator faults," *IEEE Trans. Autom. Control*, to be published. doi: 10.1109/TAC.2018.2881148.
[31] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2173–2188, Dec. 2011.
[32] Z. Hou and S. Xiong, "On model free adaptive control and its stability analysis," *IEEE Trans. Autom. Control*. to be published. doi: 10.1109/TAC.2019.2894586.
[33] Z. Han, "On the identification of time-varying parameters in dynamic systems," *Acta Autom. Sinica*, vol. 10, no. 4, pp. 330–337, 1984.
[34] S. Gerschgorin, "Uber die Abgrenzung der Eigenwerte einer matrix," *Izvestija Akademii Nauk SSSR, Serija Matematika*, vol. 7, no. 3, pp. 749–754, 1931.
[35] M. Ticherfatine and Q. Zhu, "Fast ferry smoothing motion via intelligent PD controller," *J. Mar. Sci. Appl.*, vol. 2, pp. 273–279, Jun. 2018.
[36] Z. Zen, R. Cao, and Z. Hou, "MIMO model free adaptive control of two degree of freedom manipulator," in *Proc. IEEE 7th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2018, pp. 693–697.
[37] G. Strang, *Linear Algebra Its Applications*. Boston, MA, USA: Thomson Learning, 2006.

YUAN GUO received the bachelor's degree from Beijing Jiaotong University, Beijing, China, in 2014, where she is currently pursuing the master's degree. Her research interests include data-driven control and quad-rotor aircraft control.

ZHONGSHENG HOU (SM'13) received the bachelor's and master's degrees from the Jilin University of Technology, China, in 1983 and 1988, respectively, and the Ph.D. degree from Northeastern University, China, in 1994. He was a Postdoctoral Fellow with the Harbin Institute of Technology, China, from 1995 to 1997, and a Visiting Scholar with Yale University, CT, USA, from 2002 to 2003. In 1997, he joined Beijing Jiaotong University, China, where he had been a Distinguished Professor and the Founding Director of the Advanced Control Systems Laboratory, and the Head of the Department of Automatic Control. He is currently a Chair Professor with the School of Automation, Qingdao University, Qingdao, China. He is the Founding Director of the Technical Committee on Data Driven Control, Learning and Optimization (DDCLO), Chinese Association of Automation (CAA), Beijing, China, and is a Fellow of CAA. He is an IFAC Technical Committee Member on both Adaptive and Learning Systems and Transportation Systems. His research interests include data-driven control, model-free adaptive control, learning control, and intelligent transportation systems.

His original work on model-free adaptive control has been cited by ten monographs and one textbook as a whole Chapter or Section, and has been successfully applied to more than 160 different systems in industries, laboratories, and simulations with practical background, including wide-area power systems, lateral control of autonomous vehicle, and temperature control of silicon rod. His pioneering contributions in Data Driven Control and Learning Control have been recognized by multiple projects supported by the National Natural Science Foundation of China (NSFC), including three key projects of NSFC, 2009, 2015, and 2019, respectively, and a major international cooperation project of NSFC, in 2012, and also by his leading role as a Guest Editor in two Special Sections on the topic of data-driven control in the IEEE TRANSACTIONS ON NEURAL NETWORKS, 2011, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 2017.

Up to now, he has more than 180 peer-reviewed journal papers published and more than 140 papers in prestigious conference proceedings. He has authored two monographs *Nonparametric Model and its Adaptive Control Theory*, Science Press, 1999 (in Chinese) and *Model Free Adaptive Control: Theory and Applications*, CRC Press, 2013.

SHIDA LIU received the bachelor's degree from Inner Mongolia University, Hohhot, China, in 2011, and the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2017. He currently holds a postdoctoral position with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His research interests include learning control, data-driven control, complex industrial process, and autonomous car.

SHANGTAI JIN received the bachelor's, master's, and Ph.D. degrees from Beijing Jiaotong University, Beijing, China, in 1999, 2004, and 2009, respectively.

He is currently an Associate Professor with Beijing Jiaotong University. His current research interests include model-free adaptive control, datadriven control, learning control, and intelligent transportation systems.