Data-Driven Model-Free Adaptive Predictive Control and its Stability Analysis
Feilong Zhang, Bi Zhang, Xingang Zhao

State Key Laboratory of Robotics, Chinese Academy of Science, P. R. China

Abstract—In this paper, a novel full form dynamic linearization (FFDL) data-driven model-free adaptive predictive control (MFAPC) method is proposed for a class of discrete-time single-input single-output nonlinear systems. The novelty of MFAPC is that preliminary physical model and the Lyapunov stability theory are not required for the controller design and theoretical analysis. Instead, the proposed MFAPC only uses a new dynamic linearization method called pseudo-gradient (PG) vector, which is merely related to the input/output (I/O) measurement data. The main contributions of this paper are: First, a novel MFAPC with adjustable parameters is proposed; Second, we have proved the bounded-input bounded-output stability, the monotonic convergence of the tracking error, and the internal stability of the proposed method. Third, the proposed MFAPC can be considered as an elegant extension of the current MFAC. The simulations have been carried out to verify the effectiveness of the proposed MFAPC.

Index Terms—model-free adaptive predictive control, discrete-time single-input single-output nonlinear systems, pseudo-gradient, input/output (I/O) measurement data

I. INTRODUCTION

A variety of control design methods have been proposed and fully realized so far, such as robust control, zero-pole assignment, optimal control, and so on. Most of them are typical model-based control methods in which a priori quantitative or qualitative knowledge of the systems is necessary for the controller design [1][2][3][4][5]. However, in most industrial control methods, it is hard to identify the accurate model of a nonlinear system. Hence, the alternative methods such as PID control and generalized predictive control (GPC) have become the most widely used methods in industry settings. In most cases, accurate models are not required for these methods [6]. Nonetheless, traditional PID control with fixed parameters can hardly meet the control demand of some unknown systems with strong nonlinearities, time-varying parameters and structures.

To address this issue, researchers proposed the self-tuning control as a kind of adaptive control, which can adjust the parameters of the PID controller according to the required control performance [7][8][9][10][11][12]. However, the unmodeled dynamics is unfortunately inevitable in the modeling process of the adaptive control design approach, which causes an inherent unsafety in the closed loop control system [13].

In recent years, the data-driven model-free adaptive control (MFAC) has drawn much attention. It has the advantage that the controller design depends on nothing but the measured closed-loop I/O data of the controlled objects. It is not necessary to build the off-line model of the system. Instead, MFAC control law is designed through the so-called equivalent dynamic linearization data models at each operating point using a novel concept called pseudo-gradient (PG). The time-varying PG is based on the deterministic estimation algorithms, merely using the I/O measurement data of the controlled system [14][15], whose data contains all the information of system dynamics. Moreover, the data-driven model-free adaptive control method does not require the model of the system in controller design. Subsequently, the system modeling, the unmodeled dynamics, and the theoretical assumptions on the dynamics of the system do not exist [14].

In addition, the model-free adaptive control (MFAC) of the discrete-time nonlinear systems is more suitable for the computer control with wide applications in industry settings. For example, MFAC has been successfully implemented in many practical fields, such as: chemical industry [16][17], linear motor control, injection molding process [18], PH value control [19], and robotic welding process [20]. This is all possible because of the advantage of a simplified discrete control structure which makes it easier to be implemented through computers. However, because the increments of the parameter estimation do not appear in linear form in the increments of Lyapunov functions, the Lyapunov design method performs poorly in discrete-time nonlinear system. This may be the reason why there exist fewer adaptive control methods based on discrete-time nonlinear systems [21] than on the continuous-time systems [22][23][24][25].

Fortunately, the stability analysis method of MFAC firstly proposed by Hou [26] is not based on the Lyapunov stability theory. Rather, it is analyzed by the contraction mapping principle [14][26][28], which is a new proof method in adaptive control research community. Minification inequations are used as the key proof methods to solve the main problems which are caused by the time varying of the parameters of the system model. This proof of convergence is more straightforward compared to the Lyapunov stability theory which is based on more strict assumptions and lemmas.

In order to further improve the stability and robustness of the current MFAC method for nonlinear unknown systems, we propose the model-free adaptive predictive control (MFAPC). It outperforms the MFAC for two main reasons. Firstly, the MFAPC method can use more future information of the reference trajectory. The system input can be adjusted appropriately before the reference trajectory changes, especially when the operation conditions vary severely in the control systems. Secondly, the index function of the MFAC is
only optimal for the error at the current time step, while the
index function of the MFAPC takes multiple prediction errors
into consideration. To this end, the MFAPC uses more
information in the past time to predict the output of the system.
These modifications bring several advantages into the MFAPC-
controlled system. For some nonlinear unknown systems even
with large time delay, the system has better respond speed.
Meanwhile, the tracking trajectory is more smoothly and
resistant to disturbances. Besides, the key parameter λ of the
MFAPC can be chosen from a wider range in practice, which
shares the same advantage of the GPC in industrial settings.

In regard to the relationship between MFAPC and MFAC,
some interesting findings are shown in this paper: one is that
the proposed MFAPC is an elegant extension of the current
MFAC, sharing its general structure, which hasn’t been
discussed so far, to the author’s best knowledge. Along with
this, MFAPC has all the characteristics of the MFAC, whose
characteristics are detailed in [14][26]. Another finding is that
there is a relationship between the well-known PID in MFAC
structure and the predictive PID in MFAPC.

The main contributions of this work are summarized as follows.
1) A novel MFAPC with adjustable parameters is proposed, and
the relationships between MFAPC, MFAC, MFAPC-PID and
MF-PID are analyzed.
2) The bounded-input bounded-output stability and the
monotonic convergence of the tracking error dynamics of the
MFAPC method are analyzed.
3) The classical and practical Self-tuning-PID, GPC, and the
MFAC have been compared with the proposed MFAPC method
in simulations.

The rest of the paper is organized as follows. In Section II, the
equivalent FFDL data predictive model is presented for a class
of discrete time nonlinear systems. In Section III, the MFAPC
method design and its stability analysis results are presented.
In Section IV, the comparison results of simulations are presented
to validate the effectiveness and advantages of the proposed
MFAPC method. Conclusions are given in Section V. At last,
Appendix presents the detailed stability analysis of the
proposed method.

II. DYNAMIC LINEARIZATION DATA PREDICTIVE
MODELS FOR DISCRETE-TIME NONLINEAR SYSTEMS

A. System Model

In this section, the dynamic linearization data modeling
method are given as a fundamental tool for the MFAPC
controller design, and its basic assumptions, theorem, and
insights are given as follows.

The discrete-time SISO nonlinear system is considered as follows:
y(k + 1) = f (y(k),...,y(k – n_y),u(k),...,u(k – n_u)) (1)
where f (·) ∈ R is an unknown nonlinear function, n_y, n_u ∈
Z are the unknown orders of input u(k) and the output y(k)
of the system at time k, respectively.

Assume that the nonlinear system (1) conforms with the
following assumptions:

**Assumption 1:** The partial derivatives of f (·) with respect to
all variables are continuous.

**Assumption 2:** System (1) satisfies generalized Lipschitz
condition shown as follows.

\[ |y(k + 1) – y(k + 1)| \leq b |H(k)| - |H(k)| \] (2)

Where \( H(k) = \begin{bmatrix} Y(k) \\ U(k) \end{bmatrix} = [y(k),...,y(k – L_y + 1)] \), is a vector
that contains control input within a time window \([k – L_y + 1, k]\)
and output within a moving time window \([k – L_y + 1, k]\).

Two positive integers \( L_y (1 \leq L_y \leq n_y) \) and \( L_u (1 \leq L_u \leq n_u) \) are
called pseudo orders of the system. For more detailed
explanations about Assumption 1 and Assumption 2 please refer
to [14][26].

**Theorem 1:** For the non-linear system (1) satisfying
Assumptions 1 and 2, there must exist a time-varying vector
\( \phi(k) \) called PG vector; if \( \Delta H(k) \neq 0 \), \( 1 \leq L_y \leq n_y \),
and \( 1 \leq L_u \leq n_u \) system (1) can be transformed into the full-form
dynamic -linearization data model shown as follows
\[ \Delta y(k + 1) = \dot{\phi}(k) \Delta H(k) \] (3)

For any time \( k \), we have \( \phi(k) \leq b \), where
\[ \dot{\phi}(k) = \begin{bmatrix} \phi_1(k) \\ \phi_2(k) \end{bmatrix} = [\phi_1(k),...,\phi_{L_y}(k),\phi_{1+1}(k),...,\phi_{L_u+1}(k)] \],
\[ \Delta H(k) = \begin{bmatrix} \Delta Y(k) \\ \Delta U(k) \end{bmatrix} = [\Delta y(k),...,\Delta y(k – L_y + 1)] \]
\[ \Delta u(k),...,\Delta u(k – L_u + 1) \]

**Proof:** Refer to [14][26] for details.

**Remark 1:** Please refer to [14][26] for the detailed comments
and significances about this dynamic linearization data
modeling method. [14][26] also present the relationships
between LTI DARMA model and the dynamic linearization
data model, and give the suggestions of how to choose the
pseudo-orders \( L_y \) and \( L_u \) of the model.

B. Predictive System Model

Rewrite Equation (3) into the finite N step forward prediction
equation:

\[ y(k + 1) = y(k) + \phi^T(k) \Delta H(k) \] (4)

Here, we define
\[ A = \begin{bmatrix} 0 & & & \\ 1 & 0 & & & \\ & \ddots & \ddots & & \\ & & 1 & 0 & & \\ & & & \ddots & \ddots & \end{bmatrix}_{L_y \times L_y} \]
\[ C = \begin{bmatrix} 0 & & & \\ 1 & 0 & & & \\ & \ddots & \ddots & & \\ & & 1 & 0 & & \\ & & & \ddots & \ddots & \end{bmatrix}_{L_u \times L_u} \]
\[ B^T = [1 \ 0 \ \cdots \ \cdots \ 0]_{L_u \times L_u} \]
\[ D^T = [1 \ 0 \ \cdots \ \cdots \ 0]_{L_u \times L_u} \]

In addition, we define \( A' = 0 \) and \( C' = 0 \), \( i = -1,-2, \ldots \), for
the convenience of the following expression. Then, we have
\[\Delta y(k+1) = \phi^*_y(k)A\Delta H(k) + \phi^*_y(k)\Delta Y(k) + \phi^*_y(k)\Delta U_L(k)\]
= \phi^*_y(k)\Delta Y(k) + \phi^*_y(k)A\Delta U_L(k) - 1 + \phi^*_y(k)B\Delta u(k)

\[\Delta y(k+2) = \phi^*_y(k+1)\Delta Y(k+1) + \phi^*_y(k+1)A\Delta U_L(k)\]
+ \phi^*_y(k+1)B\Delta u(k+1)

\[\Delta y(k+3) = \phi^*_y(k+2)\Delta Y(k+2) + \phi^*_y(k+2)A\Delta U_L(k+1)\]
+ \phi^*_y(k+2)B\Delta u(k+2) + \ldots

\[\Delta y(k+N) = \phi^*_y(k+N-1)\Delta Y(k)\]
+ \phi^*_y(k+N-1)A^{N-1}\Delta U_L(k-1) + \ldots

\[+ \phi^*_y(k+N-1)A^{N-2}B\Delta u(k)\]
+ \phi^*_y(k+N-1)A^{N-1}B\Delta u(k+1) + \ldots

\[+ \phi^*_y(k+N-1)B\Delta u(k+N-1)\]  

(5)

Where, \( N \) is the predictive step length, \( \Delta y(k+i) \) and \( \Delta u(k+i) \) are the increment values of the predictive output and the predictive input in the future time \( k+i \) \((i=1,2,\ldots,N)\), respectively. Here, we define \( Y_N(k) , \Delta Y_N(k+1), \Delta U_N(k), \Delta U_N(k) , \Psi^*_y(k), \Psi^*_U(k), \Psi^*_y(k), \Psi^*_N(k) \) and \( \Psi^*_N(k) \) as follows:

\[Y_N(k+1) = [y(k+1) \ldots y(k+N)]^T \quad E = \begin{bmatrix} 1 & \ldots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \ldots & 1 \end{bmatrix}_{N+1} \quad A_N = \begin{bmatrix} 1 & \ldots & \ldots \end{bmatrix} \]

\[\Delta U_N(k) = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix} \quad \Delta U_N(k) = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix} \]

\[\Psi^*_y(k) = A_N\Psi^*_y(k) = \begin{bmatrix} \phi^*_y(k+1)[C + D\phi^*_y(k)] \\ \vdots \\ \phi^*_y(k+N)[C + D\phi^*_y(k)] \end{bmatrix} + \begin{bmatrix} \phi^*_y(k+1)(k+1) \ldots \phi^*_y(k+N)(k+N) \end{bmatrix} \]

\[\Psi^*_U(k) = A_N\Psi^*_U(k) = \begin{bmatrix} \phi^*_U(k+1)A \phi^*_U(k+1)(k+1)A \ldots \phi^*_U(k+N)A \phi^*_U(k+N)(k+N) \end{bmatrix} + \begin{bmatrix} \phi^*_U(k+1)(k+1)A \phi^*_U(k+1)(k+1)A \ldots \phi^*_U(k+N)A \phi^*_U(k+N)(k+N) \end{bmatrix} \]

\[\Psi^*_N(k)_{N+1} = \begin{bmatrix} \phi^*_N(k)B \\ \phi^*_N(k+1)AB + \phi^*_N(k+1)(k+1)D\phi^*_N(k)B \\ \phi^*_N(k+2)AB + \phi^*_N(k+2)(k+2)D\phi^*_N(k)B \\ \vdots \\ \phi^*_N(k+N-1)A^{N-1}B \phi^*_N(k+N-1)(k+N-1)A^{N-2}B \phi^*_N(k+N-1)(k+N-1)A^{N-3}B \ldots \phi^*_N(k+N-1)(k+N-1)B \end{bmatrix} \]
\[ \Psi_t(k)_{N_N} = A \Psi_t(k)_{N_N} \]

\[
\begin{bmatrix}
\phi^T_m(k)B \\
\phi^T_m(k+1)AB + \phi^T_m(k)B \\
+ \phi^T_m(k+1)D\phi^T_m(k)B \\
\end{bmatrix} = \begin{bmatrix}
0 & 0 & \cdots & 0 \\
\phi^T_m(k+1)AB + \phi^T_m(k+1)B \\
\phi^T_m(k+1)B & \cdots & \phi^T_m(k+1)B \\
\end{bmatrix}
\]

where, \( \phi_{j1} \) is the i-th row of the \( \Psi_t(k) \); \( \phi_{j1} \) is the i-th row of the \( \Psi_t(k) \); \( \phi_{ji} \) is the i-th row of the \( \Psi_t(k) \); \( \Psi_{ji} \) is the i-th row of the \( \Psi_t(k) \). In addition, we define \( \phi_{ji} = 0 \), \( \phi_{ji} = 0 \) and \( \phi_{ji} = 0 \), \( i = 1, 2, \ldots \).

Then, the prediction equation (5) can be written as (6):

\[
\Delta Y_n(k + 1) = \sum_{j=1}^{N_N} [\phi^T_m(k + 1)A^{j-1}B + \phi^T_m(k + 1)B + \phi^T_m(k + 1)D\phi^T_m(k + 1)B] + \sum_{j=1}^{N_N} [\phi^T_m(k + 1)A^{j-1}B + \phi^T_m(k + 1)B + \phi^T_m(k + 1)D\phi^T_m(k + 1)B]
\]

Furthermore, both sides of equation (6) are left multiplied by \( \Lambda_y \), then (6) can be rewritten as:

\[
Y_n(k + 1) = E\gamma(k) + A_y \Psi_t(k)\Delta Y_n(k) + A_u \Psi_t(k)\Delta X_n(k)
\]

Define \( N_u \) as control step length. If \( \Delta u(k + j - 1) = 0 \), \( N_u < j \leq N \), the equation (7) can be rewritten into

\[
Y_n(k + 1) = E\gamma(k) + \Psi_{y1}(k)\Delta Y_n(k) + \Psi_{y2}(k)\Delta U_n(k)
\]

where \( \Psi_{y1}(k) \) is defined as follow.

III. MODEL-FREE ADAPTIVE PREDICTIVE CONTROL DESIGN AND STABILITY ANALYSIS

In this section, the design of MFPC method will firstly be presented. Based on the finding in [14] that introduced the relationships between the MFAC, the traditional adaptive control, and the well-known PID as well as the controller parameters choosing suggestions, we present some possible relationships among the MFAPC, MFAC, MFAPC-PID and the MFAC-PID. Then, we present the stability analysis with some necessary Theorems and Lemma.

A. Design of Model Free Adaptive Predictive Control

A weighted control input cost function is shown below:

\[
J = \left[ Y_n(k + 1) - Y_n(k + 1) \right] \left[ Y_n(k + 1) - Y_n(k + 1) \right] + \lambda \Delta U_n(k) \Delta U_n(k)
\]

(9)
Where, \( \lambda \) is a positive weighted constant; 
\[ \hat{Y}_r^{(k)}(k+1) = \left[ y'(k+1), \ldots, y'(k+N) \right]^T \] is the desired system output signal vector, where \( y'(k+i) \) is the prediction of the future output at the time \( (k+i) \) \((i=1,2,\ldots,N)\). Substitute Equation (8) into Equation (9) and solve the optimization condition \( \partial / \partial \hat{U}_n(k) = 0 \) we have:

\[
\Delta U_n(k) = \left[ \hat{Y}_r^{(k)}(k) \right] \left( k \right) + \lambda I^{-1} \hat{Y}_r^{(k)}(k) \right|_{\rho_{L_3}}(Y_n(k+1) - \hat{Y}_r^{(k)}(k)A_3 \Delta Y_{L_3}(k) - \hat{Y}_d^{(k)}(k)A_4 \Delta U_{L_4}(k-l-1) - E_y(k)) - \hat{Y}_r^{(k)}(k)A_4 \Delta U_{L_4}(k-l-1) \right] \hat{Y}_r^{(k)}(k) \right|_{\rho_{L_3}}(Y_n(k+1) - \hat{Y}_r^{(k)}(k)A_3 \Delta Y_{L_3}(k) - \hat{Y}_d^{(k)}(k)A_4 \Delta U_{L_4}(k-l-1) - E_y(k)) \tag{10}
\]

where the step factors \( A_4 = \text{diag} \left[ \rho_{L_3}, \ldots, \rho_{L_6} \right] \), and \( A_7 = \text{diag} \left[ \rho_{L_7}, \ldots, \rho_{L_{10}} \right] \) are introduced to make the controller algorithm more flexible and analysis the stability of the system, \( \rho_{L_3} < 1 \) \((i=1,2,\ldots,L_5+L_6)\). The current input is given by

\[
u(k) = u(k-1) + g^T \Delta U_n(k) \tag{11}
\]

where \( g = [1,0,\ldots,0]^T \).

**Remark 2:** \( \hat{Y}_r^{(k)}(k) \), \( \hat{Y}_r^{(k)}(k) \) and \( \hat{Y}_d^{(k)}(k) \) in the Equation (10) contain the unknown \( \hat{Y}_r^{(k)}(k+i) \) and \( \hat{Y}_d^{(k)}(k+i) \) \((i=0,1,2,\ldots,N-1)\) which need to be replaced by their estimated and predicted values \( \hat{Y}_r^{(k+i)} \) and \( \hat{Y}_d^{(k+i)} \). The \( \hat{Y}_r^{(k)}(k) \) and \( \hat{Y}_d^{(k)}(k) \) are estimated by the projection algorithm in [14][26]. The \( \hat{Y}_r^{(k)}(k+i) \) and \( \hat{Y}_d^{(k)}(k+i) \) \((i=1,2,\ldots,N-1)\) are predicted by the data-driven multi-level hierarchical forecasting method proposed in [26][29][30][31][32]. From these references, we know that the \( \hat{Y}_r^{(k+i)} \) \((i=0,1,2,\ldots,N-1)\), which is the linear combination of the \( \hat{Y}_r^{(k)} \), \( \hat{Y}_r^{(k-1)} \), \ldots, \( \hat{Y}_r^{(k-n_p+1)} \), are bounded. Let us define \( \hat{Y}_r^{(k)}(k) \), \( \hat{Y}_r^{(k)}(k) \) and \( \hat{Y}_d^{(k)}(k) \) as the estimated matrixes of the \( \hat{Y}_r^{(k)}(k) \), \( \hat{Y}_r^{(k)}(k) \) and \( \hat{Y}_d^{(k)}(k) \), respectively. Then, according to the definition of the matrix, the norms of \( \hat{Y}_r^{(k)}(k) \), \( \hat{Y}_r^{(k)}(k) \) and \( \hat{Y}_d^{(k)}(k) \) are bounded.

Then we get the proposed MFAPC control input (12)

\[
\Delta U_n(k) = \left[ \hat{Y}_r^{(k)}(k) \right] \left( k \right) + \lambda I^{-1} \hat{Y}_r^{(k)}(k) \right|_{\rho_{L_3}}(Y_n(k+1) - \hat{Y}_r^{(k)}(k)A_3 \Delta Y_{L_3}(k) - \hat{Y}_d^{(k)}(k)A_4 \Delta U_{L_4}(k-l-1) - E_y(k)) \tag{12}
\]

The current control law is

\[
u(k) = u(k-1) + g^T \Delta U_n(k) \tag{13}
\]

**Remark 3:** The method of how to choose pseudo orders \( L_5, L_6 \) of the data model are detailed in [14][26]. In practical experiments, we'd better first try the relatively small values of the pseudo orders \( L_5, L_6 \), then tune the other parameters \( \lambda, \mu, \eta, \rho_i \) \((i=1,\ldots,L_5+L_6)\), among which the \( \lambda \) plays a major role in stability analysis and should be tuned firstly to guarantee the stability of the system. If it does not converge well, the higher pseudo orders \( L_5, L_6 \) need be adopted. Then we repeat the above process.

We typically choose the sufficiently large predictive step length \( N \) which should be larger than the time-delay or make the dynamics of the system to be covered. The larger \( N \) may improve the robustness of the system. However, this may degenerate the transient and tracking performance and increase the online computational burden. For \( N_u \), while the larger control horizon \( N_u \) may increase the sensitivity and tracking ability of the system, it may cause the degradation of stability and robustness of the system, and online computational cost will be increased inevitably for matrix dimension extended. \( N_u \) can be chosen to be 1 for simple systems (e.g. low-order linear system), whereas for complex systems, a larger value of \( N_u \) can improve the transient and tracking performance. When \( N_u = 1 \), \( \hat{Y}_r^{(k)}(k) \) will become a column vector, and it will reduce the computational cost and computational time.

**Remark 4:** The following cases are given as the special cases of the proposed MFAPC method.

Case 1: When \( N_u = 1 \), we have the following simplified control output (14), which does not have the inverse calculation of matrix

\[
\Delta U_n(k) = \frac{1}{\left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \left[ \hat{Y}_r^{(k)}(k) \right]_{1,1} + \lambda} \left( \rho_{L_3} \right)_{1,2} \left[ \rho_{L_3} \right] \left[ \Delta y(k) \right] \tag{14}
\]

\[
\Delta u(k) = K_e \Delta e(k) + K_t \begin{bmatrix} y'(k+1) - y(k) \\ y'(k+N) - y(k) \end{bmatrix} + K_d \Delta e(k) - \Delta e(k-1) \tag{15}
\]

where

\[
K_e = \rho_{L_3}^2 \left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \left[ \hat{Y}_r^{(k)}(k) \right]_{1,1} \tag{16}
\]

\[
K_t = \rho_{L_3}^2 \left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \left[ \hat{Y}_r^{(k)}(k) \right]_{1,1} \tag{17}
\]

\[
K_d = -\rho_{L_3}^2 \left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \left[ \hat{Y}_r^{(k)}(k) \right]_{1,1} - \rho_{L_3} \left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \left( y(k) - y(k) \right), \text{ and we rewrite}
\]

\[
\left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} = -\left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \left[ \hat{Y}_r^{(k)}(k) \right]_{1,2} \tag{18}
\]

(15) obviously represents the PID form of MFAPC. Furthermore, when \( N = 1 \) and the corresponding \( N_u = 1 \), we have
\[
\Delta u(k) = -\sum_{i=r}^{\infty} a_i [\hat{\phi}_i(k)]^T \rho_i \Delta y(k) - [\hat{\phi}_i(k)]^T \rho_i \Delta y(k-1)
\]
where
\[
K_p = \frac{\rho_1 \hat{\phi}_1(k) + \rho_2 \hat{\phi}_2(k)}{\lambda + \hat{\phi}_1(k)} \quad \text{and} \quad K_i = \frac{\rho_1 \hat{\phi}_1(k)}{\lambda + \hat{\phi}_1(k)}
\]

This is the well-known PID controller structure, which belongs to MFAC structure presented in [14].

Case 3: When \( N = 1 \) and the corresponding \( N_u = 1 \), the MFAPC degenerates into the MFAC shown as (17)

\[
\Delta u(k) = \frac{\rho_1 \hat{\phi}_1(k) + \rho_2 \hat{\phi}_2(k)}{\lambda + \hat{\phi}_1(k)} \Delta y(k) - \frac{\rho_1 \hat{\phi}_1(k)}{\lambda + \hat{\phi}_1(k)} \Delta y(k-1)
\]

From Case 3, we can conclude that the proposed MFAPC can be considered as an elegant extension of the current MFAC, whose meaning and analysis are shown in [14][15][26]. In addition, Fig. 1 shows the relationships among MFAPC, MFAC, the well-known PID in MFAC (MFAC-PID) structure, and the predictive PID in MFAPC (MFAPC-PID).

**IV. SIMULATIONS**

A number of examples are given to show the effectiveness and the advantages of MFAPC-PID methods by comparing with other adaptive control methods: self-tuning PID, predictive PID and MFAC respectively.

Example 1: In this example, the following discrete-time SISO nonlinear structure-varying system is considered.

\[
\begin{align*}
2.5 y(k-1) y(k-2) + 0.7 \sin(0.5(y(k-1)) + y(k-2)) \cos(0.5(y(k-1)) + y(k-2)) + & 1.2 u(k-1) + 1.4 u(k-2) \\
1 + y^2(k-1) + y^2(k-2) + y(k-2) + & u(k-2) \\
0.1 y(k-1) - 0.2 y(k-2) - 0.3 y(k-3) + & 0.2 u(k-3) \\
& 0 < k \leq 250 \\
& 250 < k \leq 500 \\
& 500 < k \leq 750 \\
& 750 < k \leq 1000
\end{align*}
\]

The system is structure-varying, nonlinear, discontinuous, and we suppose that the system is unknown to the controller design process. The desired output trajectory is

\[ y^*(k+1) = 5(x(-1)^{\nu(1/2a)}) \quad 1 \leq k \leq 1000 \]

The controller parameters and initial setting for the GPC-PID, self-tuning PID and MFAPC-PID are listed in Table I.

All the parameters of GPC, Self-tuning PID, and MFAPC should be the same at the beginning, especially the initial value, aiming to create a relatively fair condition to make comparisons. If there is a better parameter for GPC and Self-tuning PID under this condition, we will choose this parameter to show the superiority of MFAPC. The orders of these three methods are set as \( n=2, m=1 \), which is the typical PID mode. And the reset values of PG for MFAPC choose \( \phi_i(k) = [1,1,1] \). The control performances of the Clarke’s GPC, Goodwin’s Self-tuning PID
and the proposed MFAPC in PID form are given in Fig. 2-a and Fig. 2-b.1 to Fig. 2-b.5. The comparisons among these three methods can be separated into the following four stages:

1) In Fig. 2-a and Fig. 2-b.1, we can see that the sequence of the rapidity and the precision of the systems controlled by these methods are: MFAPC>GPC>Self-Tuning PID. In addition, there exists large fluctuations when using Self-Tuning PID, and more details are shown in this paper.

2) In Fig. 2-a, from the time of [100, 500], the outputs of the system controlled by these three methods almost coincide, despite the fact that there is a structure-varying, discontinuous, and nonlinear unknown model switching at the time 250. From Fig. 2-b.2, when the desired trajectory of step signal descents at the time 300, we can see that Self-tuning PID is slightly better than MFAPC, and that MFAC is slightly better than GPC. As shown in Fig. 2-b.3, when the step signal rises, we can see that the output of the system step signal rises, we can see that the output of the system controlled by Self-tuning PID is closer to MFAPC than the result in Fig. 2-b.2, and that these two methods are slightly better than GPC. In this stage, the control performances of these three methods are very close and can almost be considered as the same in [100, 500].

3) In Fig. 2-a, a system model, which differs more than other three stages, is switched at the time 500, the control performances of these methods cannot be guaranteed in [500, 750]. It requires a much longer time for the system to track the desired trajectory. The outputs of the system model controlled by these three methods fail to track the desired trajectory of step signal before the time 650 when the step signal rises. This stage is regarded as an initial parameter setting process of the next stage from [750, 1000], which resets the new initial parameters of these methods and prepares for the comparison of these methods.

3) In Fig. 2-a, the system model, with a 7-sample time delay from input \( u(k-7) \) to output \( y(k) \), is switched at the time 750. After the time 750, the outputs of the model controlled by these methods have a large fluctuation. In Fig. 2-b.4, when the desired trajectory of step signal drops at somewhere between [750,800], the tracking performances of the system controlled by these methods are: MFAPC>GPC>Self-Tuning PID. The system keeps being stabilized in the desired trajectory before the time 900 when the step signal rises. As shown in Fig. 2-b.5, the tracking performances of the system controlled by these methods are still consistent compared with the results in Fig. 2-b.4: MFAPC>GPC>Self-Tuning PID.

The control inputs of these methods are shown in Fig. 3. Fig. 4 shows the components of PG estimation. In this example, Least Square estimate method is applied to estimate the parameters in both GPC and Self-Tuning PID, and the estimated parameters of GPC are shown in Fig. 5. We can see that the convergence of estimated parameters is fast at the beginning, then each parameter is adjusted according to the changes of the model.
(b. 5) From 900 to 1000
(b) Partial enlarged system output
Figure. 2 Tracking performance

Fig. 3 Control input

Fig. 4 Estimated value of PG

Fig. 5 Estimated parameters of GPC

Various parameters of the Self-Tuning PID controller are tested. However, all the tracking performances at the beginning of the simulation are not acceptable. Some of the choices are shown in Table II, and corresponding simulations are shown in Fig. 6. Besides, various initial values of the $\theta$ in [0.001, 5] are tested in simulations, which shows that the large fluctuations at the beginning stage of simulation in [0,50] cannot be eliminated effectively compared with GPC and the MFAPC in Fig. 2-a. The simulation for this case is not shown in this paper.

TABLE II Parameter Settings for the Self-Tuning PID

| PID  | $\xi$, $\omega_n$  | $A_{m}$ |
|------|------------------|---------|
| PID1 | $\xi = 0.707, \omega_n = 1$ | [1, 1.21, 0.3678] |
| PID2 | $\xi = 0.8, \omega_n = 1$ | [1, 1.2376, 0.3829] |
| PID3 | $\xi = 0.9, \omega_n = 1$ | [1, 1.341, 0.4493] |
| PID4 | $\xi = 0.9, \omega_n = 0.5$ | [1, 0.7137, 0.67] |
| PID5 | ** | [1, 1.2, 0.3] |
| PID6 | ** | [1, 6, 0.1] |

Example 2: A number of examples are given in [14][37] to show the effectiveness and the advantages of MFAC methods by comparing with other typical DDC methods, data-driven PID (DD-PID), iterative feedback tuning (IFT), and virtual reference feedback tuning (VRFT), respectively. The conclusion in [14] is that the tracking performance of Hou’s MFAC-PI is better than the above DDC method in its simulation. In this example, comparisons of the simulation results between MFAPC-PI and MFAC-PI are given under the same model which is unknown to the controller design process and is from [14][37]:

$$y(k+1) = 0.6y(k) - 0.1y(k-1) + 1.8u(k) - 1.8u^2(k) + 0.6u^3(k) - 0.15u(k-1) + 0.15u^2(k-1) - 0.05u^3(k-1)$$

(18)

The model is merely applied to generate output data for MFAPC and MFAC. The desired trajectory is the same as example 1, and it is more difficult to be tracked compared with Hou’s [14][37]. The initial values of MFAPC and MFAC are identical. The parameter settings for MFAPC-PI and MFAC-PI methods are given in Table III. The control output of MFAPC-PI is written in the following form:

$$\Delta u(k) = g^T \hat{\Psi}_d(k) \hat{\Psi}_{e0}(k) + \lambda I \hat{\Psi}_{e0}(k) \bullet$$

(19)

$$[\rho_1(\hat{Y}_d(k+1) - E\hat{y}(k)) - \rho_2\hat{\Psi}_y(k)\Delta y(k)]$$

TABLE III Parameter Settings for MFAC-PI and MFAPC-PI
Moreover, its tracking performance is better than that of MFAC with $\lambda=10$. Thus, it validates the claim that MFAPC is not sensitive and has stronger robustness to the change of the key parameter $\lambda$ compared with MFAC in this example, and we can have a wider range for choosing the $\lambda$ in practice.

![Figure 8 Tracking performance](image)

**Figure 8 Tracking performance**

**Table 2**

| Parameter       | MFAC-PI in [14] | MFAC-PI | MFAPC-PI | MFAPC2-PI |
|-----------------|-----------------|---------|----------|-----------|
| Order $L_y$     | $L_y=1, L_u=1$  | $L_y=1, L_u=1$ | $L_y=1, L_u=1$ | $L_y=1, L_u=1$ |
| $\lambda$      | 1               | 1       | 2        | 2         |
| $\rho$         | [0.4,0.4]       | [0.4,0.4] | [0.4,0.4] | [0.4,0.4] |
| $\mu, \eta$    | 1, 1            | 10.05   | 10.05    | 10.05     |
| Initial value   | (By all means)  | [0.1,0.1] | [0.1,0.1] | [0.1,0.1] |
| Reset value     | Null (By all means) | Null | Null | Null |
| $(u(0), u(1), u(2))$ | (0,0)          | (0,0)   | (0,0)    | (0,0)     |
| $(y(0), y(1), y(2))$ | (0,0)          | (0,0)   | (0,0)    | (0,0)     |
| Predictive step | No choice       | No choice | N=3      | N=2       |
| Control step    | No choice       | No choice | N=3      | N=2       |

In [14], it was concluded that the penalty factor $\lambda$ plays a major role in the stability analysis and in applications. In Fig. 7, the MFAC-PI is applied with different values of $\lambda$ in order to make comparisons with MFAPC-PI applied with $\lambda=2$. In addition, Fig. 8 shows the tracking performance of the system when the MFAPC-PI is applied with different values of $\lambda$.  

![Figure 9 Estimated value of PG](image)

**Figure 9 Estimated value of PG**

From Fig. 9, we can see that there is a modest difference in the estimation of PG parameters between MFAPC and MFAC, although all the controller’s parameters are set identically. Since the controller design of MFAPC and MFAC merely use the system I/O data, which contains all the information of the system dynamics. The different outputs of both control methods may further lead to a modest difference of the estimated PG results.

![Figure 10 Tracking performance](image)

**V. CONCLUSION**

A novel data-driven model-free adaptive predictive control (MFAPC) method with adjustable parameters is proposed for a class of discrete-time single-input and single-output nonlinear systems. Then, we show some special cases of MFAPC. The bounded-input bounded-output (BIBO) stability analysis and the monotonic convergence of the tracking error of the MFAPC...
method are analyzed by the contraction mapping technique with some other lemmas and theorems. The simulations were carried out to verify the effectiveness of the proposed MFAPC.

APPENDIX

Proof of Theorem 2. This section shows the proof of convergence of the tracing error and the BIBO stability of the system controlled by proposed MFAPC.

We define $P = g^T [\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I]^{-1} \hat{\Psi}_N (k)$, and from Section II, we know that $\hat{\Psi}_U (k), \hat{\Psi}_Y (k)$ can be expressed as

$\hat{\Psi}_U (k) = \begin{bmatrix} \hat{\Psi}_{U1}(k), \hat{\Psi}_{U2}(k), \ldots, \hat{\Psi}_{ULu}(k), 0 \end{bmatrix}_{N \times Lu}$

$\hat{\Psi}_Y (k) = \begin{bmatrix} \hat{\Psi}_{Y1}(k), \hat{\Psi}_{Y2}(k), \ldots, \hat{\Psi}_{YN}(k) \end{bmatrix}_{N \times Ly}$

We define $\Delta G(k) = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k-\lambda_t+1) \\ \Delta y(k) \\ \vdots \\ \Delta y(k-\lambda_t+1) \end{bmatrix}$ and $F = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Then we have

$\Delta G(k) = \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k-\lambda_t+1) \\ \Delta y(k) \\ \vdots \\ \Delta y(k-\lambda_t+1) \end{bmatrix}^T$

\begin{align*}
&= A(k) \Delta G(k) \\
&+ \rho_{\lambda_t+1} g^T [\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I]^{-1} \hat{\Psi}_N (k) E \Phi F(k)
\end{align*}

(20) can be written as

$$
\Delta G(k) = A(k)CD(k-1)\Delta G(k-1) + \rho_{\lambda_t+1} g^T [\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I]^{-1} \hat{\Psi}_N (k) E \Phi F(k)
$$

Considering the sum of the first row of $A(k)$ and the matrix norm inequalities between $\|\cdot\|_e$ and $\|\cdot\|_c$ we have

$$
\sum_{i=1}^{Lu+1} \rho_{\lambda_t+1} \|\hat{\Psi}_{Ui}(k)\|_e + \sum_{i=1}^{Lu} \rho_{\lambda_t} \|\hat{\Psi}_{U1}(k)\|_e
\leq (\max_{i=1,\ldots,Lu+1} \rho_i) \|\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I\|_e \|\hat{\Psi}_N (k)\|_e
$$

$$
\sum_{i=1}^{Ly+1} \rho_{\lambda_t+1} \|\hat{\Psi}_{Yi}(k)\|_e + \sum_{i=1}^{Ly} \rho_{\lambda_t} \|\hat{\Psi}_{Y1}(k)\|_e
\leq (\max_{i=1,\ldots,Ly+1} \rho_i) \|\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I\|_e \|\hat{\Psi}_N (k)\|_e
$$

(21)

$\hat{\Psi}_N (k)$ is a symmetric semi-positive matrix, which means that $\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I$ will be a symmetric positive matrix, then we have

$$
\|\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I\|_e
$$

is the maximum row sum matrix norm (max norm). $\|\cdot\|_e$ is the spectral norm of matrix. We suppose the eigenvalues of $\hat{\Psi}_N (k) \hat{\Psi}_N (k)$ are $b_i \geq 0$, $i = 1, \ldots, N_u$, so the eigenvalues of $\hat{\Psi}_N (k) \hat{\Psi}_N (k) + \lambda I$ are $\lambda + b_i > 0$, $i = 1, \ldots, N_u$, which means
that the eigenvalues of \([\hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I]^{-1}\) are \(\frac{1}{\lambda + b_i} > 0\), \(i = 1, \ldots, N_u\). Therefore, we get
\[
\left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e = \min_{i = 1, \ldots, N_u} \left\{ \lambda + b_i \right\} > 0
\]

Combining (22) and (23), we have
\[
\begin{align*}
\sum_{i=1}^{L_u} \left( p_{\hat{\Psi}^T_{N_0}(k)} \right) + \sum_{i=1}^{L_u} \left( p_{\hat{\Psi}_{N_0}(k)} \right) \\
\leq \sqrt{N_u} \left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e \left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e \\
\leq \sqrt{N_u} \left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e = \min_{i = 1, \ldots, N_u} \left\{ \lambda + b_i \right\} 
\end{align*}
\]

Assume \(s\) is the number of the maximum sum of the row of \(\hat{\Psi}^T_{N_0}(k)\), then we can see that
\[
\left\| \hat{\Psi}^T_{N_0}(k) \right\|_e \leq \sum_{j=1}^{s} \sum_{i=1}^{L_u} \left( \hat{\psi}^T_{N_0}(k + j - 1)A^T B \hat{\psi}_{N_0}(k + j - 1) + \sum_{j=1}^{L_u} C^T \hat{\phi}_{j-1,i} \right)
\]

is bounded. We suppose that \(s_1\) is the number of the maximum sum of the row of the matrix \(\hat{\Psi}(k)\), then we can see that
\[
\left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) \right\|_e \leq \sum_{j=1}^{s} \sum_{i=1}^{L_u} \left( \hat{\psi}^T_{N_0}(k + j - 1)C^T \hat{\psi}_{N_0}(k + j - 1) + \sum_{j=1}^{L_u} C^T \hat{\phi}_{j-1,i} \right)
\]

is bounded. Therefore, there exists a positive \(\lambda_{min1}\), such that \(\lambda > \lambda_{min1}\), we can obtain the following inequation:
\[
\sum_{i=1}^{L_u} \left( p_{\hat{\Psi}^T_{N_0}(k)} \right) + \sum_{i=1}^{L_u} \left( p_{\hat{\Psi}_{N_0}(k)} \right) \\
\leq \sqrt{N_u} \left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e \left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e \\
\leq \sqrt{N_u} \left\| \hat{\Psi}^T_{N_0}(k)\hat{\Psi}_{N_0}(k) + \lambda I \right\|_e = \min_{i = 1, \ldots, N_u} \left\{ \lambda + b_i \right\} 
\]

Given \(0 < \rho_3 < 1\), \(\cdots\), \(0 < \rho_{L_u+1} < 1\), we have
\[
\left( \max_{i=1, \ldots, L_u + 1} \rho_i \right) < 1.
\]

Then, according to the definition of spectral radius and Lemma 2, [14] has deduced the following inequation
\[
\left\| A(k) \right\|_e = \left\| C \right\|_e \left\| D(k-1) \right\|_e \leq \left( d_1 + \epsilon \right) \left( 1 + \epsilon \right) \max \left\{ 1, \lambda \right\} \leq d_2 \leq 1
\]
where, \( X = \begin{bmatrix} I_{d_i} \end{bmatrix} \)

Taking the norm of (20) and combining (30) and (32), we have
\[
\| \Delta G(k) \| = \| A(k) \|, \| C \|, \| D(k-1) \|, \| \Delta G(k-1) \| \\
+ \left[ \rho_{t_0} g \left( \hat{\Psi}_n^T(k) \hat{\Psi}_n(k) + \lambda I \right)^{-1} \hat{\Psi}_n^T(k) E_x \| e(k) \| \right] \\
= d_2 \| \Delta G(k-1) \| + M_2 \| e(k) \| \\
\vdots \\
= d_2^n \| \Delta G(0) \| + \rho_{t_0+1} M_2 \sum_{i=1}^k d_2^{i-1} \| e(i) \| 
\]

Combining (3), (12), (13) and (20) together, we have
\[
e(k+1) = y - y(k+1) = y - y(k) + \hat{\Psi}_n^T(k) \Delta H(k) \\
= e(k) - X \hat{\phi}_{t_0+1}^T(k) [A(k)CD(k-1) \Delta G(k-1) \\
+ \rho_{t_0+1} \hat{\phi}_{t_0+1} \left( \hat{\Psi}_n^T(k) \hat{\Psi}_n(k) + \lambda I \right)^{-1} \hat{\Psi}_n^T(k) E_x \} e(k) \}
\]

\[
\cdot \hat{\Psi}_n^T(k) E_x \} e(k) - X \hat{\phi}_{t_0+1}^T(k) A(k)CD(k-1) \Delta G(k-1) \\
= (1 - \rho_{t_0+1} \hat{\phi}_{t_0+1} \left( \hat{\Psi}_n^T(k) \hat{\Psi}_n(k) + \lambda I \right)^{-1} \hat{\Psi}_n^T(k) E_x \} e(k) \\
- X \hat{\phi}_{t_0+1}^T(k) A(k)CD(k-1) \Delta G(k-1)
\]

Similarly, there exists a positive \( \lambda_{\text{min}} \) and a positive \( M_3 \), such that \( \lambda > \lambda_{\text{min}} \), then we have the below inequation

\[
0 < M_3 \leq \left| \rho_{t_0+1} \hat{\phi}_{t_0+1}^T(k) P \right| \\
= \left| \rho_{t_0+1} \hat{\phi}_{t_0+1} \left( \hat{\Psi}_n^T(k) \hat{\Psi}_n(k) + \lambda I \right)^{-1} \hat{\Psi}_n^T(k) E_x \} e(k) \right| \\
\leq \rho_{t_0+1} \left| \hat{\phi}_{t_0+1} \left( \hat{\Psi}_n^T(k) \hat{\Psi}_n(k) + \lambda I \right)^{-1} \hat{\Psi}_n^T(k) E_x \} e(k) \right| < 0.5
\]

According to (36), we have

\[
0.5 < \| X \hat{\phi}_{t_0+1}^T(k) \| P \leq \| \rho_{t_0+1} \| \left| \hat{\phi}_{t_0+1}^T(k) P \right| < 1 - M_3 < 1
\]

Let \( d_4 = 1 - M_3 \) and take the norm of (34), then we yield
\[
\| e(k+1) \| = \| X \hat{\phi}_{t_0+1}^T(k) \| P \| e(k) \| \\
+ \left[ \rho_{t_0} \| A(k)CD(k-1) \|, \| \Delta G(k-1) \| \\
< d_2 \| e(k) \| + d_2 \| X \hat{\phi}_{t_0+1}^T(k) \|, \| \Delta G(k-1) \| < \cdots < d_2^{k-1} \| e(2) \| + d_2 \sum_{i=1}^{k-1} d_2^{i-1} \| X \hat{\phi}_{t_0+1}^T(i+1) \|, \| \Delta G(k-1) \| \\
< d_2^{k-1} \| e(2) \| + d_2 \| \Delta G(0) \| \\
+ d_2 \sum_{j=1}^{k-1} d_2^{j-1} \| X \hat{\phi}_{t_0+1}^T(i+1) \|, \| \rho_{t_0+1} M_2 \| d_2^{j-1} \| e(j) \| 
\]

Then (38) becomes
\[
\| e(k+1) \| < d_2^{k-1} \| e(2) \| + d_2 \sum_{j=1}^{k-1} d_2^{j-1} \| e(j) \| \\
+ \| \rho_{t_0+1} M_2 \| d_2^{j-1} \| e(j) \| 
\]

Let
\[
g(k+1) = d_2^{k-1} \| e(2) \| + d_2 \sum_{j=1}^{k-1} d_2^{j-1} \| e(j) \| \\
+ (d_2^{k-2} + d_2^{k-3} + \cdots + d_2^{k-2}) \| e(0) \|
\]

Inequality (39) can be rewritten as follow
\[
|e(k+1)| < g(k+1), \quad k = 1, 2, \ldots
\]

Where, \( g(2) = d_4 \| e(1) \|, \quad d_4 = 1 - M_3 > 0.5 > d_3
\]

According to (40) and (41), we have
\[
g(k+2) = d_4 \| e(2) \| + d_2 d_4 \sum_{j=1}^{k-1} d_2^{j-1} \| e(j) \| \\
+ (d_2^{k-1} + d_2^{k-2} + \cdots + d_2^{k-3}) \| e(0) \| \| G(0) \|
\]

Thus, Theorem 2 is the direct result of (48) and (41) if \( \lambda > \lambda_{\text{min}} = \max \{ \lambda_{\text{min}1}, \lambda_{\text{min}2}, \lambda_{\text{min}3} \} \).

Since \( G(k) \) is the information vector that consists of the inputs and outputs, we can prove the BIBO stability of the closed loop system by proving the boundedness of \( G(k) \).

From (34), (40), (41) and (46), we have
\[
\|G(k)\| \leq \sum_{j=1}^{\infty} \|\Delta G(i)\| \\
\leq \sum_{j=1}^{\infty} \|\Delta G(0)\| + \rho_{\Delta s} M \sum_{j=1}^{\infty} d_j^{-1} \|\varepsilon(j)\| \\
< \|\Delta G(0)\| + \rho_{\Delta s} M \sum_{j=1}^{\infty} d_j^{-1} \|\varepsilon(j)\| \\
= \|\Delta G(0)\| + \rho_{\Delta s} M \left(\|\varepsilon(1)\| + (d_1^{-1} \|\varepsilon(1)\| + \|\varepsilon(2)\|) + (d_2^{-1} \|\varepsilon(1)\| + d_3^{-1} \|\varepsilon(2)\| + \|\varepsilon(3)\| + \cdots + \|\varepsilon(k)\|) + \cdots \right) \\
< \|\Delta G(0)\| + \rho_{\Delta s} M \left(\|\varepsilon(1)\| + \|\varepsilon(2)\| + \cdots + \|\varepsilon(k)\|\right) \\
< \|\Delta G(0)\| + \rho_{\Delta s} M \left\| \|g(1)\| + \|g(2)\| + \cdots + \|g(k)\| \right\| \\
< \|\Delta G(0)\| + \rho_{\Delta s} M \left\| \|g(2)\| \right\| \\
< \|\Delta G(0)\| + \rho_{\Delta s} M \left\| \|g(2)\| \right\| \\
= \|\Delta G(0)\| + \rho_{\Delta s} M \left\| \|g(2)\| \right\| \\
< \|\Delta G(0)\| + \rho_{\Delta s} M \left\| \|g(2)\| \right\| \\
\]

(49)

Therefore, the boundedness of \( \|G(k)\| \) is proved by (49). In other words, the closed-loop system is BIBO stable.

We finished the proof of Theorem 2.

REFERENCES

[1] Y. J. Liu, S. Tong, C. L. P. Chen, D. J. Li. Neural controller design-based adaptive control for nonlinear MIMO systems with unknown hysteresis inputs[J]. IEEE transactions on cybernetics, 2015, 46(1): 9-19.
[2] Liu Y J, Tong S. Barrier Lyapunov functions for Nussbaum gain adaptive control of full state constrained nonlinear systems[J]. Automatica, 2017, 76: 143-152.
[3] Pekař L, Matuš A. A suboptimal shifting based zero-pole placement method for systems with delays[J]. International Journal of Control, Automation and Systems, 2018, 16(2): 594-608.
[4] Kugelmann B, Pulch R. Robust Optimal Control of Fishing in a Three Competing Species Model[J]. IFAC-PapersOnLine, 2018, 51(2): 7-12.
[5] Li Y X, Yang G H. Model-based adaptive event-triggered control of strict-feedback nonlinear systems[J]. IEEE transactions on neural networks and learning systems, 2018, 29(4): 1033-1045.
[6] Liu D, Yang G H. Data-driven adaptive sliding mode control of nonlinear discrete-time systems with prescribed performance[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017 (99): 1-7.
[7] Kalman R E. Design of self-optimizing control system[J]. Trans. ASME, 1958, 80: 468-478.
[8] Åström K J, Wittenmark B. On self-tuning regulators[J]. Automatica, 1973, 9(2): 185-199.
[9] Åström K J, Wittenmark B. Adaptive control[M]. Courier Corporation, 2013.
[10] Goodwin, Graham C., Peter J. Ramadge, and Peter E. Caines. "Recent results in stochastic adaptive control." Conference on Information Sciences and Systems, The Johns Hopkins University, Baltimore MA. 1979.
[11] Goodwin G C, Long R S, McInnis B C. Adaptive control of bilinear systems[M]. 1980.
[12] Goodwin G, Johnson C, Sin K. Global convergence for adaptive one-step-ahead optimal controllers based on input matching[J]. IEEE Transactions on Automatic Control, 1981, 26(6): 1269-1273.
[13] B. D. O. Anderson and A. Dehghani, "Challenges of adaptive control: Past, permanent and future," Annual reviews in control, vol. 32, no. 2, pp. 123–135, Dec. 2008.
[14] Hou Z, Xiong S. On Model Free Adaptive Control and its Stability Analysis[J]. IEEE Transactions on Automatic Control, 2019.
[15] Hou Z, Jin S. A novel data-driven control approach for a class of discrete-time nonlinear systems[J]. IEEE Transactions on Control Systems Technology, 2011, 19(6): 1549-1558.
[16] Z. S. Wang, L. Liu, and H. G. Zhang, “Neural network-based model free adaptive fault-tolerant control for discrete-time nonlinear systems with sensor fault,” IEEE Transaction on Systems Man and Cybernetics: Systems, vol. 47, no. 8, pp. 2351-2362, 2017.
[17] Z. H. Pang, G. P. Liu, D. H. Zhou, and D. H. Sun, “Data-driven control with input design-based data dropout compensation for networked nonlinear systems,” IEEE Transaction on Control Systems Technology, vol. 25, no. 2, pp. 628-636, 2017.
[18] Y. Zhao, Z. C. Yuan, C. Lu, G. L. Zhang, X. Li, and Y. Chen, “Improved model-free adaptive wide-area coordination damping controller for multiple-input multiple-output power systems,” IET Generation, Transmission and Distribution, vol. 10, no. 3, pp. 3264-3275, 2016.
[19] H. G. Zhang, J. G. Zhou, Q. Y. Sun, J. M. Guerrero, and D. Z. Ma, “Data-driven control for interlinked AC/DC microgrids via model-free adaptive control and dual-droop control,” IEEE Transactions on Smart Grid, vol. 8, no. 2, pp. 557-571, 2017.
[20] R. C. Roman, M. B. Radae, and R. E. Precup, “Multi-input-multi-output system experimental validation of model-free control and virtual reference feedback tuning techniques,” IET Control Theory and Applications, vol. 10, no. 12, pp. 1395-1402, 2016.
[21] C. Yeh and P. V. Kokotovic, “Adaptive control of a class of nonlinear discrete-time systems,” International Journal of Control, vol. 62, no. 2, pp. 303-324, 1995.
[22] Kokotović, Petar V. Foundations of adaptive control[M]. Springer Berlin Heidelberg, 1991.
[23] I. Kanellopoulos, P. V. Kokotovic, and S. A. Morse, “Systematic design of adaptive controllers for feedback linearizable systems,” IEEE Transactions on Automatic Control, vol. 36, no. 11, pp. 1241-1251, 1991.
[24] J. B. Pomet and L. Praly, “Adaptive nonlinear regulation: estimation from the Lyapunov equation,” IEEE Transaction on Automatic Control, vol. 37, no. 6, pp. 729-740, 1992
[25] M. Krstić, P. V. Kokotovic, and I. Kanellopoulos, Nonlinear and Adaptive Control Design, John Wiley and Sons, 1995.
[26] Z. S. Hou and S. T. Jin, Model Free Adaptive Control: Theory and Applications, CRC Press, Taylor and Francis Group, 2013.
[27] Z. S. Hou and S. T. Jin, “A novel data-driven control approach for a class of discrete-time nonlinear systems,” IEEE Transaction on Control Systems Technology, vol. 19, no. 6, pp. 1549-1558, 2011.
[28] Z. S. Hou. On model-free adaptive control: the state of the art and perspective[J]. Control Theory & Applications, 2006, 23(4): 586-592.
[29] Yu Q, Hou Z, Bu X, et al. RBFNN-Based Data-Driven Predictive Iterative Learning Control for Nonaffine Nonlinear Systems[J]. IEEE transactions on neural networks and learning systems, 2019, 30(12): 4201-4215.
[30] Hou Z, Liu S, Tian T. Lazy-learning-based data-driven model-free adaptive predictive control for a class of discrete-time nonlinear systems[J]. IEEE transactions on neural networks and learning systems, 2016, 28(8): 1914-1928.
[31] Hou Z, Liu S, Yin C. Local learning-based model-free adaptive predictive control for adjustment of oxygen concentration in syngas manufacturing industry[J]. IET Control Theory & Applications, 2016, 10(12): 1384-1394.
[32] Dong N, Feng Y, Han X, et al. An Improved Model-Free Adaptive Predictive Control Algorithm For Nonlinear Systems With Large Time Delay[C]//2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS). IEEE, 2018: 60-64.
[33] E. I. Jury, Theory and Application of the z-Transform Method. New York: Wiley, 1964.
[34] Abadir K M, Magnus J R. Matrix algebra[M]. Cambridge University Press, 2007.
[35] Hou Z, Jin S. Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems[J]. IEEE Transactions on Neural Networks, 2011, 22(12): 2173-2188.
[36] Chongzhi F, Deyun X. Process identification[J]. Beijing: Tsing, 1988.
[37] Hou Z, Zhu Y. Controller-dynamic-linearization-based model free adaptive control for discrete-time nonlinear systems[J]. IEEE Transactions on Industrial Informatics, 2013, 9(4): 2301-2309.