Hindawi
Complexity
Volume 2018, Article ID 8953035, 12 pages
https://doi.org/10.1155/2018/8953035

Research Article

Control of Complex Nonlinear Dynamic Rational Systems

Quanmin Zhu,[1,2] **Li Liu**,[1] **Weicun Zhang**,[1] **and Shaoyuan Li**[3]

[1]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2]*Department of Engineering Design and Mathematics, University of the West of England, Frenchay Campus, Coldharbour Lane, Bristol BS16 1QY, UK*
[3]*Department of Automation, Shanghai Jiao Tong University, 800 Dongchuan Rd., Minhang, Shanghai 200240, China*

Correspondence should be addressed to Li Liu; liuli@ustb.edu.cn

Received 11 April 2018; Accepted 6 May 2018; Published 14 June 2018

Academic Editor: Zhile Yang

Nonlinear rational systems/models, also known as total nonlinear dynamic systems/models, in an expression of a ratio of two polynomials, have roots in describing general engineering plants and chemical reaction processes. The major challenge issue in the control of such a system is the control input embedded in its denominator polynomials. With extensive searching, it could not find any systematic approach in designing this class of control systems directly from its model structure. This study expands the U-model-based approach to establish a platform for the first layer of feedback control and the second layer of adaptive control of the nonlinear rational systems, which, in principle, separates control system design (without involving a plant model) and controller output determination (with solving inversion of the plant U-model). This procedure makes it possible to achieve closed-loop control of nonlinear systems with linear performance (transient response and steady-state accuracy). For the conditions using the approach, this study presents the associated stability and convergence analyses. Simulation studies are performed to show off the characteristics of the developed procedure in numerical tests and to give the general guidelines for applications.

1. Introduction

This section justifies the reasons for designing controllers for rational models by introducing model expression and representations, achieved results in model identification, and a critical review of controller-designing approaches.

1.1. Nonlinear Dynamic Rational Systems

Definition 1 [1]. Assign a triplet (X, f, h), where X is an irreducible real affine variety and (f, h) are mapping functions. A system Σ, with input $U \in \mathbb{R}^m$ and output $Y \in \mathbb{R}^r$, is defined as polynomial/rational, while the functions $f = \{f_\alpha \mid \alpha \in U\}$ and $h : X \rightarrow \mathbb{R}^r$ both on X are mappings from input space to state space and from state space to output space polynomial/rational, respectively. That is, for polynomial systems, $h_i \in A$ for all $i = 1, \dots, r$ where A is the algebra of all polynomials on the variety X, and for rational systems, $h_i \in Q$ for all $i = 1, \dots, r$ where Q is the algebra of all rational functions on the variety X.

For a single-input and single-output (SISO) nonlinear dynamic rational system, it can be generally modelled with a ratio of two polynomials [1, 2].

$$
\begin{aligned}
y(k) &= \frac{N_p(k)}{D_p(k)} + e(k) \\
&= \frac{N_p(Y_{k-1}, U_{k-1}, E_{k-1})}{D_p(Y_{k-1}, U_{k-1}, E_{k-1})} + e(k) \\
&= \frac{\sum_{j=1}^{\text{num}} p_{nj}(k)\theta_{nj}}{\sum_{j=1}^{\text{den}} p_{dj}(k)\theta_{dj}} + e(k),
\end{aligned} \tag{1}
$$

where $y(k) \in \mathbb{R}$, $u(k) \in \mathbb{R}$, and $e(k) \in \mathbb{R}$ denote measured output, input, and model error/noise/uncertainties, respectively, at time instant $k(=1,2,\ldots)$. $N_p(k)$ and $D_p(k)$ are real valued and smooth numerator and denominator polynomials, respectively. $Y_{k-1} \in \mathbb{R}^n \supset y(k-1),\ldots,y(k-n)$, $U_{k-1} \in \mathbb{R}^n \supset u(k-1),\ldots,\mathrm{u}(k-n)$, and $E_{k-1} \in \mathbb{R}^n \supset e(k-1),\ldots,\mathrm{e}(k-n)$ denote the delayed outputs, inputs, and model noises, respectively. $p_{nj}(k) \in \mathbb{R}$ and $p_{dj}(k) \in \mathbb{R}$ for regression terms $\theta_{nj} \in \mathbb{R}$ and $\theta_{dj} \in \mathbb{R}$, respectively, are the coefficients and num and den for numbers of total regression terms of the polynomials. The major properties of the rational model (1) are summarised below:

It is also defined as a total nonlinear model [2] as it covers many different linear and nonlinear models as its subsets (such as NARMAX (nonlinear autoregressive moving average with exogenous input) models [3] and intelligent models for neurofuzzy systems [4]). Rational systems have been observed in general engineering, chemical processes, physics, biological reactions, and econometrics; for example, rational models are a class of mechanistic models in describing catalytic reactions in chemical kinetics [5, 6]; metabolic, signal, and genetic networks in systems biology [1]; and movement of satellites in Earth orbit [1]. There have also been reports of rational modelling applications [7–9].

This is more concise in structure than a polynomial; the example below uses a Taylor series expansion to approximate a simple rational model below.

$$\begin{aligned} y(k) &= \frac{\sin\left(u(k-1)\right)}{1+y^2(k-3)} \\ &= \sin\left(u(k-1)\right)\left[1-y^2(k-3)+y^4(k-3)\right]. \end{aligned} \tag{2}$$

The other characteristic of the rational models is the power to quickly change the model output while input has small variations. Consider a simple system output below

$$y(k) = \frac{1}{1+u(k-2)}. \tag{3}$$

Clearly the model output will be dramatically increased, as the input $u(k-2)$ approaches -1. This comes from the function of the denominator.

Introducing a denominator polynomial makes the model concise in describing complexity and adds more functions in describing nonlinearities. On the other side, in contrast to polynomial systems, this makes identification and control system design noticeably different and more difficult with the inherent nonlinear parameters and control inputs [2]. Therefore, comprehensive studies of this class of systems in theoretical and application aspects are required. This study takes the pioneer step towards the control of rational systems.

1.2. Model Identification. Model identification has been relatively mutual to some extent. So far, the identification aspect has gone through data-driven model structure detection, parameter estimation, and model validation from noise-contaminated input and output data. The major work on rational model identification is summarised in the following categories: linear least squares (LLS) algorithms for parameter estimation—extended LLS estimator [10], recursive LLS estimator [11], orthogonal LLS structure detector and estimator [12], fast orthogonal algorithm [13], and implicit least squares algorithm [14], and nonlinear least squares algorithms—prediction error estimator [15] and globally consistent nonlinear least squares estimator [16]. Other algorithms include the following categories: back propagation (BP) algorithm [17] and enhanced linear Kalman filter (EnLKF) [18].

There are two model validation methods: higher order correlation tests [19] and omnidirectional cross-correlation tests [20].

A summary of the representative publications till 2015 can be found in a survey of rational model identification [2].

1.3. Controller Design. As surveyed above, rational models have been increasingly used to represent nonlinear dynamic plants. Consequently, the control system design should have been considered on the agenda in the follow-up studies. However, up to now, there is no reference found for designing such controllers directly referred to the model analytical knowledge. The paramount difficulty is that part of the controller output is embedded in the denominator polynomial $D_p(k)$. For example, $y(k) = (0.5y(k-1) - y(k-2)u(k-1) + 0.1u^5(k-1))/(1+y^2(k-2)+0.2u^2(k-1))$. With extensive investigations through major academic publication searching engines, it can be concluded that this study is the first trial with analytical approaches to design a controller for rational systems.

Regarding controller design approaches possibly referred to the rational systems, these could be the reduction of rational model structure complexity, which are neural network models, linear approximation models, linearization, and iterative learning control and U-model enhanced control. A brief critical review of the approaches is presented.

Reference [21] on neural controllers is probably the first publication relating to control of rational models. However, the design approach has merely used rational models as extreme nonlinear examples; it has not designed controllers by taking the model structure into consideration (even if known in advance), except for taking the models as the representatives of complex nonlinear dynamic systems.

Piecewise linearization [22, 23] around operating points has been widely studied to simplify controller-designed procedures when plants are subject to mild nonlinear dynamics. It should be mentioned that a group of piecewise linear models can be admitted as a linear model, with varying order and parameters in different operating intervals. The promising property is using linear control design strategies directly. However, it could induce inaccuracy and dynamic uncertainty because of ignoring some inherent nonlinearities from their original nonlinear representations. Further, this method may also increase computational burden/complexity while overborrowing piecewise linear intervals to match severe nonlinearities.

Pointwise linearization has been claimed by neural network-based control and/or adaptive control, which uses linear models to approximate predominant dynamics around an operating point or every input-output dynamic gain at each time instance and then employs a neural network to determine the error induced by the linearization [24, 25].

Once again, it uses linear control system design to construct nonlinear control systems. However, this involves online neural network learning or online model iden parameter estimation, and therefore, the constructed nonlinear control system is operated under adaptive principles (the controller parameters are updated with the neural network output), even for deterministic nonlinear plants. The other related issue is the selection of neural network topology, which has no systematic procedure available to find the best fitted neural network representative.

Feedback linearization is a well-developed subject [26]. A general SISO nonlinear system is described as

$$\begin{aligned} \dot{x} &= f(\mathbf{x}) = g(\mathbf{x})u \\ y &= h(\mathbf{x}) \end{aligned} \tag{4}$$

where $\mathbf{x}$ is the state vector and u and y are the input and output, respectively. $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are real valued and smooth mapping functions. With this model structure, a series of analogies with some fundamental features of linear control systems have been established, which provides a very useful concept in the design of nonlinear control systems using linear design methodologies. Obviously, the model has u in an explicit position. The studied nonlinear rational model has no such explicit expression for input u to be designed, and this immediately reveals that the methodologies rooted in the approach, although useful references, are not directly applicable in designing control of nonlinear rational systems. The other input-output linearization techniques [27] have had similar requirements for an explicit u expression and special skills for state variable transformation.

Iterative learning/data-driven control/model-free control is another possible control system design methodology in avoiding model structure complexity. The approaches do not require a clear plant model structure but still need plants with some mild conditions in control [28, 29]. Again, if a rational model is available, it is wasteful without using the model information in the control system design. It is believed, particularly for man-made engineering systems/products (built up by rules/models), that any repetitive process and motion has a model existing in operation even though the model is yet to be identified.

U-Model-based control has claimed to radically relieve the dependence of plant model-oriented design foundation. The use of the plant model is effectively reduced as a reference for converting to U-model and accordingly to work out the control output [30]. U-Model-based control assumes the feasibility of using linear system design procedures to design the control of nonlinear dynamic plants with assigned response performances. The U-model control platform is illustrated in Figure 1.

The U-model systematically converts smooth (polynomial and extended including transcendental functions) models, derived from principles or identified from measurements, into a type of U-based model to equivalently describe plant input-output relationship, so that it establishes a general platform to facilitate control system design and dynamic inversion. It should be mentioned that there is nothing lost with the derived U-models from the original

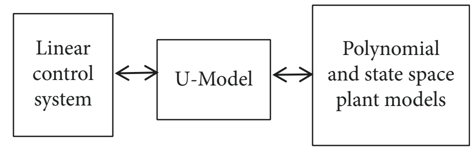


Figure 1: U-Model-based control system design.

nonlinear models. The difference between the two types of model expressions is that those original nonlinear models could be obtained from principles, such as Newton's law, or identified directly from measured data; the U-models are derived from the original models in control design-oriented expressions. Regarding the U-control (U-model-based control) research status, Zhu and Guo [31] have brought forward a fundamental framework in terms of pole placement control for nonlinear systems. More recently, U-control has been expanded to general predictive control [32] and sliding mode control [30]. To accommodate the U-control of state space models, a backstepping algorithm is being expanded to extract the controller output within multiloop U-models. With the nature of separating control system design (specifying closed-loop performance) and controller output calculation (by resolving plant dynamic inversion through U-model), it can be forecast that the other classical control issues could be similarly formulated within a general and concise framework.

1.4. Organisation of the Study. The remaining study is organised into five major sections. Section 2 is used to define a generic framework of control-oriented U-model for representing smooth nonlinear dynamic plants. It is then expanded by including a rational model and transcendental functions as its subsets to lay a basis for applying linear control system design techniques. Section 3 proposes a general pole placement controller for nonlinear rational systems within the U-model framework. Section 4 shows design of an adaptive UPPC for the control of stochastic nonlinear rational systems. Section 5 tests a number of typical rational systems with the developed procedures and shows the exemplary procedures for potential users.

2. U-Model: A Generic Framework of Control-Oriented Nonlinear Plant Models

2.1. U-Model Foundation: Polynomial [30]. Consider a general polynomial description of

$$y(k) = N_p(Y_{k-1}U_{k-1}) = \sum_{i=0}^{L} p_i(k)\theta_i, \tag{5}$$

where $y(k) \in \mathbb{R}$ and $u(k) \in \mathbb{R}$ denote the plant output and input, respectively, at time instant $k(=1, 2, \dots)$. $N_p(\cdot) \in \mathbb{R}$ is a real valued and smooth polynomial function and $Y_{k-1} \in \mathbb{R}^n \supset y(k-1), \dots, y(k-n)$ and $U_{k-1} \in \mathbb{R}^n \supset u(k-1)$, $\dots, \mathrm{u}(k-n)$ denote the delayed outputs and inputs, respectively. $p_i(k) \in \mathbb{R}$ denotes the model structure variables, e.g., $u(k-2)y^3(k-1)$, $u(k-1)u^2(k-3)$, $y(k-2)y(k-3)$, and $\theta_i \in \mathbb{R}$ denote the coefficients. To convert the above polynomial into a U-model, which is a polynomial with an argument of control input $u(k-1)$ (also called controller output while talking about control system design), it gives [30]

$$y(k)=\sum_{j=0}^{M}\boldsymbol{\lambda}_j(k)u^j(k-1), \tag{6}$$

where degree M is of controller output $u(k-1)$ and $\boldsymbol{\lambda}(k)=[\boldsymbol{\lambda}_0(k)\ \dots\ \boldsymbol{\lambda}_M(k)]\in\mathbb{R}^{M+1}$ is the time-varying parameter vector, a function of absorbing past inputs U_{k-2}, outputs Y_{t-1}, and parameters θ_{nj} in the original polynomial. An example illustrating the conversion to U-model from an ordinary polynomial is shown here. Consider a polynomial,

$$\begin{aligned}y(k)&=0.2y(k-1)y(k-3)+0.5u(k-1)u(k-3)\\&\quad-0.9y(k-2)u^2(k-1).\end{aligned} \tag{7}$$

Rearrange polynomial (7) with

$$y(k)=\boldsymbol{\lambda}_0(k)+\boldsymbol{\lambda}_1(k)u(k-1)+\boldsymbol{\lambda}_2(t)u^2(k-1), \tag{8}$$

where $\boldsymbol{\lambda}_0(k)=0.2y(k-1)y(k-3)$, $\boldsymbol{\lambda}_1(k)=0.5u(k-3)$, and $\boldsymbol{\lambda}_2(k)=-0.9y(k-2)$.

Clearly, the time-varying $\boldsymbol{\lambda}_j(k)$ is absorbing the past inputs/outputs and parameters of the original polynomial, associated with $u^j(k-1)$.

Property 1. Assign $\varphi:\mathbb{R}^{L+1}\to\mathbb{R}^{M+1}$ a U-mapping to convert the classical polynomial expression of (5) to its U-expression of (6) and the inverse be φ^{-1}, that is

$$f(p_i,\theta_i)\xrightarrow{\varphi}f\left(u^j,\boldsymbol{\lambda}_j\right) \tag{9}$$

Thus, it has good mapping properties [30].

2.2. U-Mode: Rational. With reference to (1), its deterministic parametric rational expression is given below:

$$y(k)=\frac{N_p(^*)}{D_p(^*)}=\frac{\sum_{j=1}^{\text{num}}p_{nj}(k)\theta_{nj}}{\sum_{j=1}^{\text{den}}p_{dj}(k)\theta_{dj}}. \tag{10}$$

Its U-model realisation can be determined by removing the denominator to the left-hand side of (10); it gives

$$y(k)D_p(^*)=N_p(^*). \tag{11}$$

Convert (11) into U-model form to yield

$$y(k)\sum_{i=0}^{L}\gamma_i(k)u^i(k-1)=\sum_{j=0}^{M}\boldsymbol{\lambda}_j(k)u^j(k-1), \tag{12}$$

where $\boldsymbol{\lambda}_j(k)\in\mathbb{R}$ is a function of past inputs U_{k-2} and outputs Y_{k-1} and parameters θ_{nj} in the numerator polynomial. Similarly, $\gamma_i(k)\in\mathbb{R}$ is a function of past inputs U_{k-2} and outputs Y_{k-1} and parameters θ_{dj} in the denominator polynomial. M and L are the degrees of the model input $u(k-1)$ in the numerator and denominator, respectively. Here is a simple example to show the conversion of

$$y(k)=y(k-1)\frac{1}{u(k-1)}. \tag{13}$$

Inspection of (12) gives

$$y(k)\gamma_1(k)u(t-1)=\boldsymbol{\lambda}_0(k), \tag{14}$$

where $\gamma_1(k)=1$ and $\boldsymbol{\lambda}_0(k)=y(k-1)$.

In the following sections of the controller design, it is required to make a dynamic inversion of (12) to solve for roots.

There are many standard root-solving algorithms for such polynomial equations [30].

Remark 1. Compared with polynomial U-realisation, it can be noted that rational model U-realisation is an implicit expression of $y(k)$ due to the multiplicative item $y(k)D_p(k)$.

2.3. U-Model: Extended. To describe more general nonlinear terms including those transcendental functions, define the extended U-model below:

$$y(k)f_b(u(k-1))=f_a(u(k-1)), \tag{15}$$

where $f_b(u(k-1))\in\mathbb{R}$ and $f_a(u(k-1))\in\mathbb{R}$ are smooth functions. In general, these can be expressed as

$$\begin{aligned}f_b(u(k-1))&=\sum_j f_{bj}(u(k-1)),\\f_a(u(k-1))&=\sum_j f_{aj}(u(k-1)).\end{aligned} \tag{16}$$

Here is a simple example to show its U-model representation; consider

$$y(k)=\frac{y(k-1)\sin(u(k-1))}{1+\cos^2(u(k-1))}. \tag{17}$$

For its U-model of (15), it gives

$$\begin{aligned}f_b(u(k-1))&=\gamma_0(k)+\gamma_1(k)\cos^2(u(k-1)),\\f_a(u(k-1))&=\boldsymbol{\lambda}_1(k)\sin(u(k-1)),\end{aligned} \tag{18}$$

where $\gamma_0(k)=1$, $\gamma_1(k)=1$, and $\boldsymbol{\lambda}_1(k)=y(k-1)$.

3. Pole Placement Controller: A Show Case of the Design Procedure

The control objective is, for a desired trajectory $v(k)$, to determine a control input $u(t)$ to drive the underlying system output $y(k)$ to follow the desired trajectory $v(k)$ with an acceptable performance (such as transient response and steady-state error), while all the inputs and outputs of the control system are bounded within the permitted ranges.

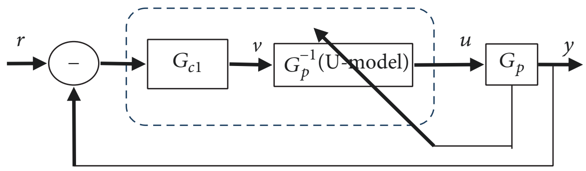


Figure 2: U-Model control system.

3.1. U-Control System Design. In general, there are three steps in the U-control system design routine:

Form a proper linear feedback control system structure, as shown in Figure 2. The controller, in the dashed line block, consists of two functions, the invariant controller G_{c1} and the dynamic inverter G_p^{-1}. The plant model is G_p.

Design the invariant controller G_{c1} by linear control system approach. By letting $G_p = 1$, therefore $G_p^{-1} = 1$, and specifying the desired closed-loop transfer function G, it gives $G_{c1} = G/(1-G)$ and the invariant controller output $v(t)$ is the desired output while the plant model is a unit constant.

Determine dynamic inverter G_p^{-1} to work out the controller output $u(k-1)$. Assuming the plant model is bounded-input/bounded-output (BIBO) stable and the inverse of G_p exists, expressing the plant model G_p in forms of U-model, letting$y(k) = v(k)$ in the U-model, gives model (15) in expression of $v(k) f_b(u(k-1)) = f_a(u(k-1))$. To determine control input $u(t-1)$ is to find the inverse by resolving the equation of $v(k) f_b(u(k-1)) - f_a(u(k-1)) = 0$.

It should be noted that the arrow line from the plant to the dashed line block represents the U-model update from the plant model at each time instance.

Proposition 1. *Generality: the U-model-based control allows a once-off design for all linear and nonlinear polynomial models. This is due to the controller G_{c1} design being independent of model G_p.*

Proposition 2. *Simplicity: the U-model-based control requires no repeated computation if a plant model is changed. Again, this is due to the controller G_{c1} design being once-off and independent of model G_p, and changes to plant model G_p only change the U-model to resolve different roots. In comparison, almost all classical and modern control approaches are plant model-based designs; that is, the controller design is a function of both system performance and plant; accordingly, if the plant model is changed, the controller must be redesigned.*

Proposition 3. *Feasibility for controller design of rational systems: this can be proved directly from Proposition 1 and U-realisation of the rational model in (12).*

In formality, the U-adaptive control is very similar to deterministic U-model control. The difference is that the plant model is required to be estimated or updated online in the adaptive control.

For simplicity, but without losing generality, in formulation of the U-model (polynomial), once the invariant controller is designed, the real controller output can be determined by letting

$$v(k) = \sum_{j=0}^{M} \boldsymbol{\lambda}_j(k) u^j(k-1). \quad (19)$$

Then resolving one of the roots from

$$v(k) - \sum_{j=0}^{M} \boldsymbol{\lambda}_j(k) u^j(k-1) = 0. \quad (20)$$

3.2. Stability and Robust Analysis of U-Model Control Systems. There are two typical situations: ideal case—deterministic systems without modelling error and disturbance, and nonideal case—deterministic systems with modelling errors and/or disturbance.

Theorem 1. (bounded-input, bounded-output (BIBO) stability of deterministic U-model control systems). *Regarding the U-model control system shown in Figure 2, it is BIBO stable and tracks the bounded reference signal r properly while the following conditions are satisfied:*

(i) Invariant controller G_{c1} is closed-loop stable; that is, all poles of the closed loop are located with the unit circle.

(ii) Plant model G_p is a bounded-input/bounded-output (BIBO).

(iii) The inverse of the plant model G_p^{-1} exits.

Proof. With reference to Figure 2, it has $G_p^{-1} G_p = 1$ from the conditions (ii) and (iii). Accordingly, the closed-loop transfer function is given in terms of $G_{c1}/(1+G_{c1})$, which is stable from (i), and thus, the tracking performance is given by $rG_{c1}/(1+G_{c1})$.

Remark 2. This establishes a framework for designing control for both linear and nonlinear dynamic plants. It is feasible, simple, general, and with no repetition of controller design on changes to the plant model, except the computation of the inversion of the changed plant U-model polynomial. In other words, this is a new methodology for minimising the complexity induced by the plant model in control system design, which is particularly important for nonlinear plants. U-model, as a universal dynamic inverter, is the key to achieve the goals.

Theorem 2. (BIBO stability of uncertain U-model control systems). *Regarding the U-model control system structured in Figure 2, modelling error and/or disturbance $\varepsilon_U(t)$ can be treated as an external disturbance as shown in Figure 3. It is BIBO stable and tracking the reference signal with a bounded error while the following conditions are satisfied:*

(i) Invariant controller G_{c1} is closed-loop stable.

(ii) Plant model G_p is a bounded-input/bounded-output (BIBO).

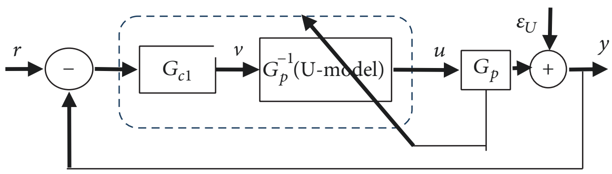


Figure 3: Uncertain U-model control system.

(iii) The inverse of the plant model G_p^{-1} exits.

(iv) The upper bound of modelling error and/or disturbance $\varepsilon_U(t)$ is satisfied with the conditions of small gain robust stability [33].

Proof. In Figure 3, $G_p^{-1}G_p = 1$; this gives $y = rG_{c1}/(1 + G_{c1}) + \varepsilon_U/(1 + G_{c1})$.

Then the stability of Figure 3 is the same as in Figure 2 while the upper bound $\varepsilon_U(t)$ is satisfied with the small gain robust stability criterion.

Remark 3. It should be noted that the tracking error is determined by $\varepsilon_U/(1 + G_{c1})$; therefore, a properly designed G_{c1} will have a degree of robustness against uncertainties/disturbance.

4. Design of Pole Placement Controller

A classical approach [34] has been selected to formulate the U-model-enhanced pole placement controller (UPPC) [30, 31]. Here, a further refined version of UPPC is presented. Within the U-model framework, closed-loop control system performance is independently specified without involving the plant model. Therefore, the classical version involving plant model can be simplified as below.

$$Rv(k) = Tr(k) - Sy(k), \tag{21}$$

and

$$\begin{aligned} R &= z^n + r_1 z^{n-1} + \cdots + r_n, \\ T &= t_0 z^m + t_1 z^{m-1} + \cdots + t_m, \\ S &= s_0 z^l + s_1 z^{l-1} + \cdots + s_l, \end{aligned} \tag{22}$$

with $r(k)$ for reference, $v(k)$ for invariant controller G_{c1} output, and $y(k)$ for plant output. The polynomials R, S, and T, with backward shift operator z^{-1} and proper orders (n, m, and l), are used to specify closed-loop control system performance.

To guarantee that the control system is realistically implementable, specify

$$O(S) < O(R) \Leftrightarrow l < n, \quad O(T) \le O(R) \Leftrightarrow m \le n, \tag{23}$$

where the operator $O(*) = \text{order}(*)$ denotes the order of the concerned linear polynomial.

With reference to (19), two control roles can be assigned with negative feedback $-R/S$ for stabilising the closed-loop system with requested dynamics and feedforward T/R for reducing steady-state errors. The structured control system is shown in Figure 4.

For designing an invariant controller, let $v(t) = y(t)$ in (19); thus, it gives the closed-loop transfer function

$$y(k) = \frac{T}{R+S} r(k) = \frac{T}{A_c} r(k). \tag{24}$$

Accordingly, the required design task is to assign the closed-loop denominator polynomial A_c and the numerator polynomial T.

It should be noted that after A_c is specified (by customers and/or designers), a routine for resolving Diophantine is needed to work out the parameters of polynomials R and S from the following relationship:

$$R + S = A_c. \tag{25}$$

To achieve zero steady state, T can be designed with

$$T = A_c(1). \tag{26}$$

The detailed design procedure and examples can be refereed to [31].

Remark 4. Compared with classical pole placement control design procedures [34], the UPPC is more concise and independent of the plant model, which results in the UPPC being generalised to any plant model structure and once-off designed. For each different plant model, this task is merely the resolving of the U-model to obtain one of the roots as the operational controller output. The relevant comparison details can be referred in [30].

5. U-Model-Based Pole Placement Control with Adaptive Parameter Estimation

The U-model-based adaptive control schematic diagram is shown in Figure 5. Again, this U-model adaptive control is different from those classical adaptive/self-tuning control approaches in terms of control structure. The feedback controller parameters are not tuned and thereafter are fixed: the only adaptation is to update U-model parameters to accommodate the plant model parameter variation and/or external disturbance, which is consistent with Propositions 1, 2, and 3.

In general, an adaptive control system can be considered as a two-layer system, that is:

(i) Layer 1: conventional feedback control

(ii) Layer 2: adaptation loop

In this study, the UPPC presented in Section 3 is selected to form the conventional feedback control. Thus, this section mainly develops this adaptation loop formulation.

In recursive formulation, there are two ways to estimate the U-model parameters in the adaptation loop.

(i) Indirect parameter estimation: estimate the original rational model parameters $(\theta_{nj}(k), \theta_{dj}(k))$ first and then convert into U-model parameters $\boldsymbol{\lambda}_j(k)$. The challenging issue is that classical recursive least

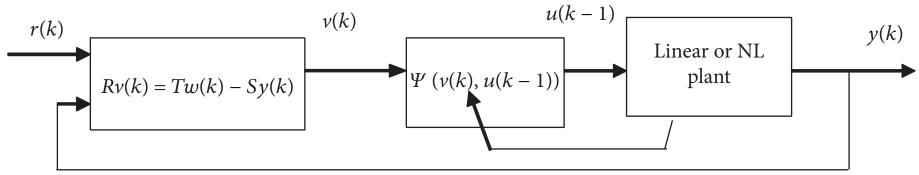


Figure 4: Structured UPPC control system.

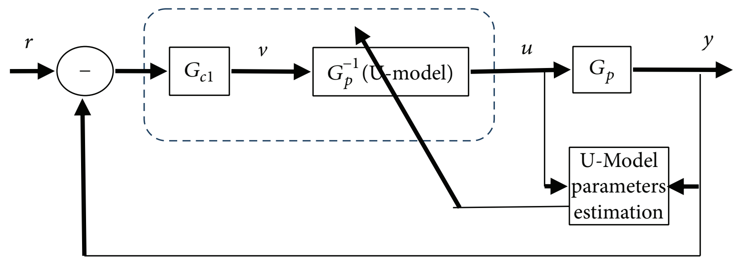


Figure 5: Adaptive U-model control system.

squares estimation algorithms give biased estimates and recursive rational model estimators need noise variance information in advance [11, 18].

(ii) Direct parameter estimation: estimate the U-model parameters $\boldsymbol{\lambda}_j(k)$ directly. The challenging issue is that the parameters $\boldsymbol{\lambda}_j(k)$, while converted from a rational model, are time varying at every sampling time. It has been proved [35] that for time-varying stochastic models, the parameter estimation errors (PEE) with the well-known forgetting factor least squares (FFLS) algorithm are bounded and the FFLS is capable of reducing the squared measurement error (the difference between measured output and model-predicted output); even the time-varying parameter estimates are not converged to their real values.

In this study, a FFLS estimator [36] is selected with the following formulations:

$$\begin{aligned}\varepsilon_U(k) &= y(k) - \boldsymbol{\Psi}^T(k)\widehat{\boldsymbol{\lambda}}(k-1),\\ \mathbf{K}(k) &= \frac{\mathbf{P}(k-1)\boldsymbol{\Psi}(k)}{\rho + \boldsymbol{\Psi}^T(k)\mathbf{P}(k-1)\boldsymbol{\Psi}(k)},\\ \widehat{\boldsymbol{\lambda}}(k) &= \widehat{\boldsymbol{\lambda}}(k-1) + \mathbf{K}(k)\varepsilon_U(k),\\ \mathbf{P}(k) &= \left[\mathrm{I} - \mathbf{K}(k)\Psi^T(k)\right]\mathbf{P}(k-1),\end{aligned} \tag{27}$$

where vector $\widehat{\boldsymbol{\lambda}}(k) = \left[\widehat{\boldsymbol{\lambda}}_0(k)\widehat{\boldsymbol{\lambda}}_1(k)\dots\widehat{\boldsymbol{\lambda}}_M(k)\right]^T \in \mathbb{R}^M$ is the estimate of $\boldsymbol{\lambda}(k)$; $\varepsilon(k) \in \mathbb{R}$ is the error, that is, the difference between the measured output and the model-predicted output; $\mathbf{K}(t) \in (M+1)\times 1$ is the weighting factor vector indicating the effect of $\varepsilon(t)$ to change the parameter vector; $\boldsymbol{\Psi}(k) = [1 \quad u(k-1) \quad \dots \quad u^M(k-1)]^T \in \mathbb{R}^M$ is the input vector at time $k-1$; ρ is the forgetting factor (a number less than 1, e.g., 0.99 or 0.95, represents a trade-off between fast tracking and noisy estimate), the smaller the value of ρ, the quicker the information in previous data will be forgotten; and $\mathbf{P}(k) \in \mathbb{R}^{(M+1)\times(M+1)}$ is the covariance matrix.

In presenting the stability of the proposed adaptive U-control, expand the virtual equivalent system (VES) concept and methodology [37] for the analysis, which is an alternative insight and judgement of the stability/convergence for adaptive control systems. Following the similar arguments as shown before, we assume $G_p^{-1}G_p = 1$, and the invariant controller G_{c1} is well defined to stabilise conventional feedback control systems and track the bounded reference signal in terms of mean squares. Then for a slow time-varying parameter model (because it is converted from its original time-invariant parameter model referred to in (5) and (6)), the U-model parameter estimation errors $\varepsilon_U(t)$ are bounded with FFLS or the other recursive algorithms [35, 38]. In this case, using Figure 3 again, knowing $\varepsilon_U(t)$ includes U-model parameter estimation errors. Hence, in terms of VES, the adaptive control system can be treated as a summation of two subsystems of

$$y = y_1 + y_2 = \frac{rG_{c1}}{(1+G_{c1})} + \frac{\varepsilon_U}{(1+G_{c1})}. \tag{28}$$

As $\varepsilon_U(t)$ is bounded, the adaptive control system is stable and the tracking control error will converge to a bounded compact set around zero, whose size depends on the ultimate bounds of estimation error ε_U.

Remark 5. The U-model provides a platform for simplifying control system design, and VES provides a platform for simplifying the analysis of stability and convergence of general adaptive control systems.

6. Simulation Studies

Four case studies have been conducted to initially validate the new design procedure. It should be made clear that there is no comparison result that can be provided as this is the first study in the control of such nonlinear rational systems.

As described before, the design is split into two stages, design invariant control G_{c1} (thus, $v(k)$ by pole placement) and determination of the controller output $u(k-1)$ by resolving plant U-model equation.

To design the pole placement controller, assign the characteristic equation

$$A_c = z^2 - 1.3205z + 0.4966. \quad (29)$$

Factorisation of (29) gives the closed-loop poles as $0.6603 \pm 0.2463i$; this gives a decayed oscillatory response ($\omega_n = 1$, $\zeta = 0.7$), which is a commonly used dynamic response index. For steady-state error performance, making its error zero gives

$$T = A_c(1) = 1 - 1.3205 + 0.4966 = 0.1761. \quad (30)$$

From the causality condition, specify the structures of R and S with

$$\begin{aligned} R &= z^2 + r_1 z + r_2, \\ S &= s_0 z + s_1. \end{aligned} \quad (31)$$

Form a Diophantine equation with polynomials A_c, R, and S [30] to yield

$$\begin{aligned} r_2 + s_1 &= 0.4966, \\ r_1 + s_0 &= -1.3205. \end{aligned} \quad (32)$$

To make polynomial R stable and having the requested response, assign $r_1 = -0.06$ and $r_2 = 0.0005$, which give two poles ($z - 0.05$ and $z - 0.01$). Then the coefficients of polynomial S are resolved in the Diophantine equation of (32) as follows.

$$\begin{aligned} s_0 &= -1.2605, \\ s_1 &= 0.4961. \end{aligned} \quad (33)$$

Consequently, controller (19) can be recursively implemented to calculate the virtual controller output $v(t)$:

$$\begin{aligned} v(k+1) = {} & 0.06v(k) - 0.0005v(k-1) + 0.1761r(k-1) \\ & + 1.2605y(k) - 0.4961y(k-1). \end{aligned} \quad (34)$$

Case 1 (feasibility test of U-control of nonlinear rational systems). Consider a rational system modelled by

$$y(k) = \frac{0.5y(k-1)u(k-1) + u^3(k-1)}{1 + y^2(k-1) + u^2(k-1)}, \quad (35)$$

where $y(k)$ is the plant output and $u(k)$ is the input of the model or controller output. This is used to test deterministic feedback control. The model structure has been typically investigated in system identification. Accordingly, its U-realisation can be expressed as

$$y(k)\left(1 + y^2(k-1) + u^2(k-1)\right) = 0.5y(k-1)u(k-1) + u^3(k-1). \quad (36)$$

To obtain the dynamic inverter G_p^{-1} output, that is, the controller output $u(t)$, let $y(k) = v(k)$; then it gives rise to

$$v(k)\left(1 + y^2(k-1) + u^2(k-1)\right) = 0.5y(k-1)u(k-1) + u^3(k-1). \quad (37)$$

To determine the control input $u(k-1)$, form a U-model equation from (37) as

$$\boldsymbol{\lambda}_0(k) - \boldsymbol{\lambda}_1(k)u(k-1) + \boldsymbol{\lambda}_2(k)u^2(k-1) - \boldsymbol{\lambda}_3(k)u^3(k-1) = 0, \quad (38)$$

where

$$\begin{aligned} \boldsymbol{\lambda}_0(k) &= v(k)\left(1 + y^2(k-1)\right), \\ \boldsymbol{\lambda}_1(k) &= 0.5y(k-1), \\ \boldsymbol{\lambda}_2(k) &= v(t), \\ \boldsymbol{\lambda}_3(k) &= 1. \end{aligned} \quad (39)$$

In this simulation, the operation time length was configured with 400 sampling points and the reference was a sequence of multiamplitude steps. The achieved output response and controller output are shown in Figures 6(a) and 6(b), respectively.

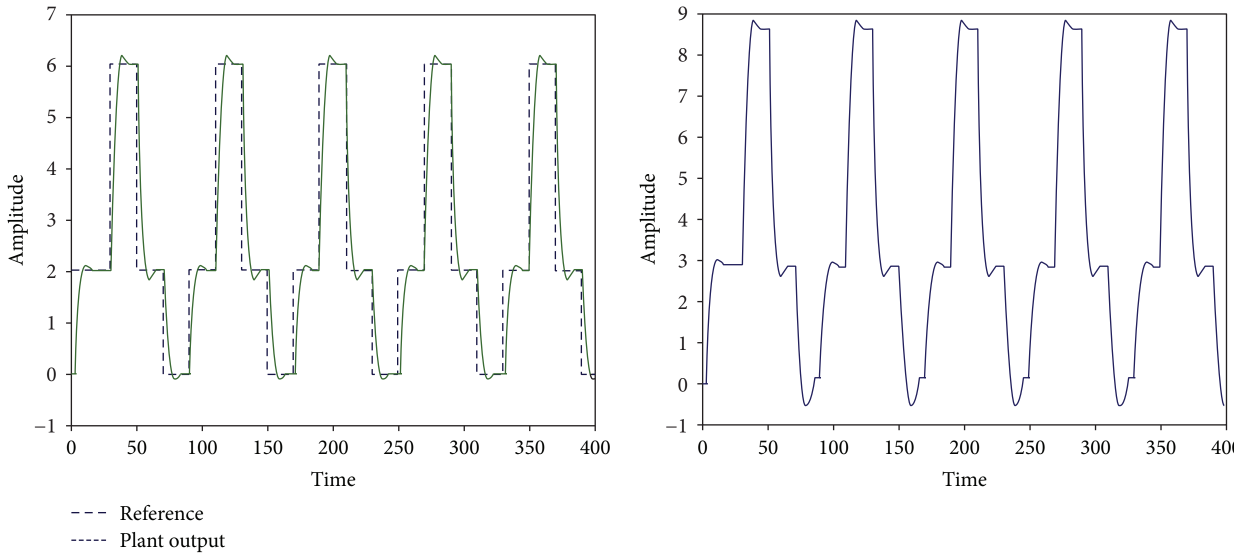
Case 2 (test of external disturbance). Consider a stochastic rational system modelled by

$$y(k) = \frac{0.5y(k-1)u(k-1) + u^3(k-1)}{1 + y^2(k-1) + u^2(k-1)} + e(k), \quad (40)$$

where $y(k)$ is the plant output, $u(k)$ is the input of the model or controller output, and $e(k)$ is Gaussian noise representing an unknown disturbance acting on the controlled plant output.

This case study was used to test adaptive feedback control. The feedback control loop has been designed as in Case 1; that is, all configurations for feedback control were kept as those used in Case 1. For the adaptation loop, the disturbance was configured with $e(k) \sim N(0, 0.01)$, the initial covariance matrix with $\mathbf{P}(k) = 10^6 I_4$, and the forgetting factor with $\rho = 0.95$ to deal with fast time-varying parameter estimation; the initial parameter vector was randomly assigned with $\widehat{\boldsymbol{\lambda}}(0) = \left[\widehat{\boldsymbol{\lambda}}_0(0)\widehat{\boldsymbol{\lambda}}_1(0)\widehat{\boldsymbol{\lambda}}_3(0)\widehat{\boldsymbol{\lambda}}_4(0)\right]^T = [0.3 \quad 0.2 \quad 0.1 \quad 0.1]^T$; and the input vector was specified with $\boldsymbol{\Psi}(k) = [1 \quad u(k-1) \quad u^2(k-1) \quad u^3(k-1)]^T$. The achieved output response and controller output are shown in Figures 7(a) and 7(b), respectively.

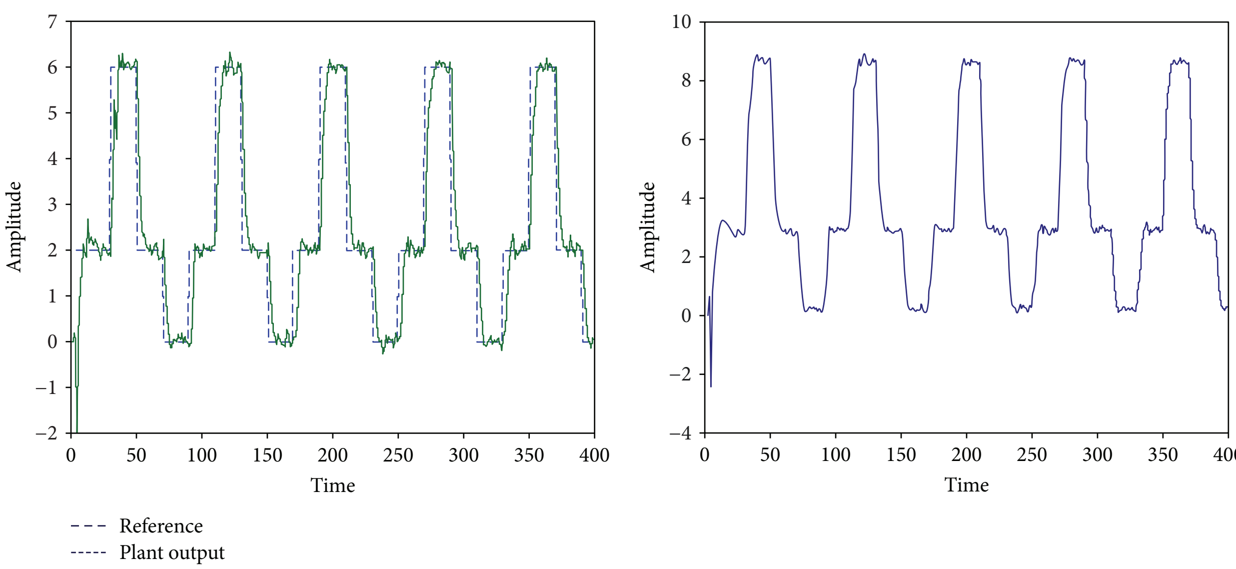
Case 3 (test of internal parameter variation). The same model structure as Case 1 is used, but the parameter associated with



(a) Plant output response

(b) Control input

Figure 6: Plant output and control input.



(a) Plant output

(b) Control input

Figure 7: Plant output and control input.

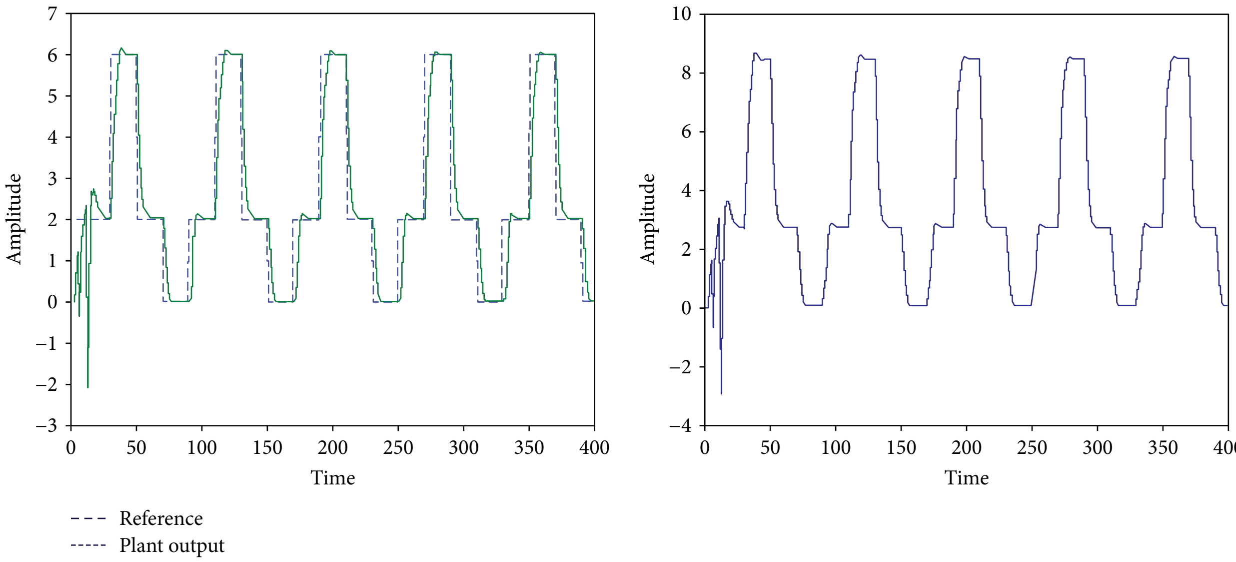
$y(k-1)$ and $u(k-1)$ is time varying representing internal parameter disturbances, such as worn parts in mechanical and electrical systems.

$$y(k) = \frac{a(k)y(k-1)u(k-1) + u^3(k-1)}{1 + y^2(k-1) + u^2(k-1)}. \tag{41}$$

In simulation, all the setups were the same as those used in Case 1. The parameter variation was configured as

$$a(k) = \begin{cases} 0.9, & 120 \le k \le 250, \\ 0.5, & \text{otherwise}. \end{cases} \tag{42}$$

The adaption loop, specified as in Case 2, was used to follow the plant model internal structure variation. The achieved output response and controller output are shown in Figures 8(a) and 8(b), respectively. Inspecting the simulation results, the output of the systems are seen to track the reference signals after a short transient phase. U-model parameter estimation is shown in Figure 9. It should be noted that this estimated parameter vector is to achieve smaller squared error between the measured output and model-predicted output. Therefore, the estimates are not converged to those real time-varying parameters in the U-model. In the future, studies to deal with time-varying parameter estimation will be conducted



(a) Plant output

(b) Control input

Figure 8: Plant output and control input.

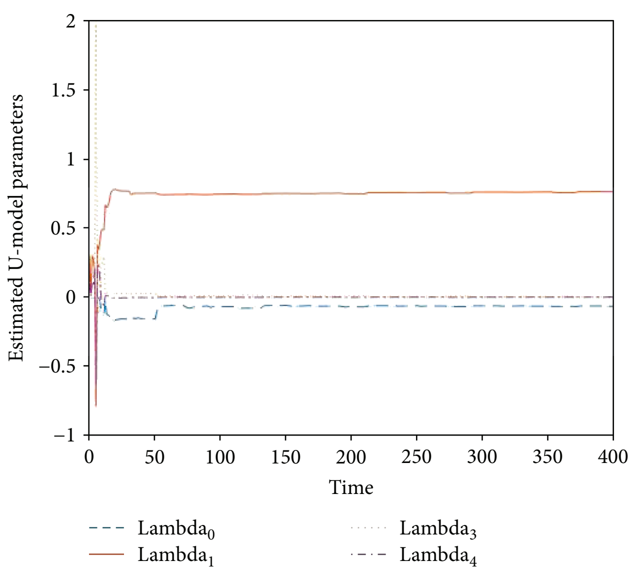


Figure 9: U-Model parameter estimates.

in terms of reducing both squared measurement errors and squared dynamic errors [39].

Case 4 (feasibility test of U-control of extended nonlinear rational systems). This study is used to test the U-control of extended rational systems with transcendental input and delayed output.

$$y(k) = \frac{0.5y(k-1) + \sin(u(k-1)) + u(k-1)}{1 + \exp(-y^2(k-1))}, \tag{43}$$

where $y(k)$ is the plant output and $u(k)$ is the input of the model or controller output. Accordingly, the extended U-model can be expressed as

$$y(k)\left(1 + \exp\left(-y^2(k-1)\right)\right) = 0.5y(k-1) + \sin(u(k-1)) + u(k-1). \tag{44}$$

With the same controller designed in (44) above, assigning the output $y(k)$ of (44) with the desired output $v(k)$ of (34) gives

$$v(k)\left(1 + \exp\left(-y^2(k-1)\right)\right) = 0.5y(k-1) + \sin(u(k-1)) + u(k-1). \tag{45}$$

Therefore, the control input $u(k-1)$ can be solved by

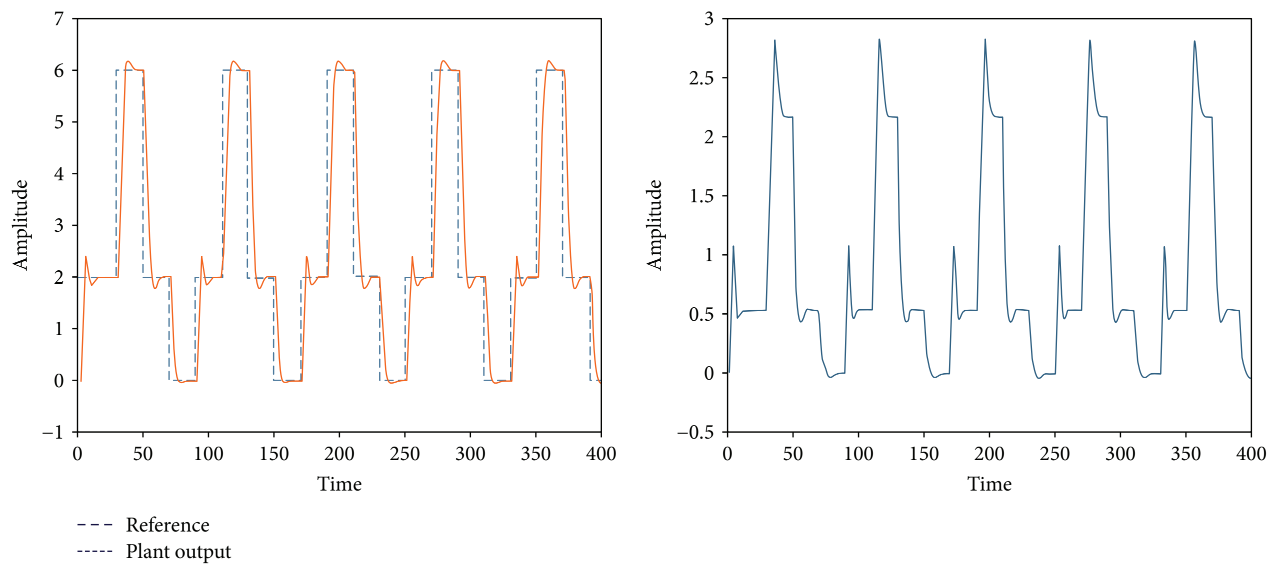
$$v(k)\left(1 + \exp\left(-y^2(k-1)\right)\right) - 0.5y(k-1) - \sin(u(k-1)) - u(k-1) = 0. \tag{46}$$

The achieved output response and controller output are shown in Figures 10(a) and 10(b), respectively. Once again, the computational experiment confirms the feasibility of U-control.

7. Conclusions

A fundamental question is raised in this study and those for the other U-model-enhanced controls: after two generations of plant model- (polynomial and state space) centered control system design research/applications, what is the next generation of development? Should the research for new model structures continue, or should control systems be designed without such plant model requirements (possibly implying separation of control system design and controller output determination)?

One of the feasible choices in the future progression could be the U-control design methodology, which radically



(a) Plant output

(b) Control input

FIGURE 10: Plant output and control input.

reduces the complexity of plant model-oriented design methods. The proposed U-control method provides a platform (1) with a universal control-oriented structure to represent existing models, (2) separating closed control system design from plant model structure (no matter whether linear or nonlinear or polynomial or state space), (3) where all well-developed linear control system design methods can be expanded in parallel to nonlinear plant models, (4) with a supplementary to all existing control design methods. Accordingly, this study is a show case using the U-model framework to design the control of the nonlinear rational systems with classical linear design approaches. Further study on the rational model control could derive concise algorithms for robust and adaptive control with reference to the recent research development [40, 41].

This foundation work has put an emphasis on formulation of structure in a systematic approach. Rigorous mathematical considerations should be followed to establish a comprehensive description and explanation.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge Dr. Steve Wright for English proof reading. Finally, the authors are grateful to the anonymous reviewers for their constructive comments and suggestions with regard to the revision of the paper.

References

[1] J. Nemcova, *Rational Systems in Control and System Theory, [Ph.D. thesis]*, Centrum Wiskunde & Informatica (CWI), Amsterdam, 2009.

[2] Q. M. Zhu, Y. J. Wang, D. Y. Zhao, S. Y. Li, and S. A. Billings, "Review of rational (total) nonlinear dynamic system modelling, identification, and control," *International Journal of Systems Science*, vol. 46, no. 12, pp. 2122–2133, 2015.

[3] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, Wiley, John & Sons, Chichester, West Sussex, 2013.

[4] L. X. Wang, *Adaptive Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 1994.

[5] S. D. Dimitrov and D. I. Kamenski, "A parameter estimation method for rational functions," *Computers and Chemical Engineering*, vol. 15, no. 9, pp. 657–662, 1991.

[6] D. I. Kamenski and S. D. Dimitrov, "Parameter estimation in differential equations by application of rational functions," *Computers & Chemical Engineering*, vol. 17, no. 7, pp. 643–651, 1993.

[7] I. Ford, D. M. Titterington, and C. P. Kitsos, "Recent advances in nonlinear experimental design," *Technometrics*, vol. 31, no. 1, pp. 49–60x, 1989.

[8] J. W. Ponton, "The use of multivariable rational functions for nonlinear data representation and classification," *Computers & Chemical Engineering*, vol. 17, no. 10, pp. 1047–1052, 1993.

[9] C. Kambhampati, J. D. Mason, and K. Warwick, "A stable one-step-ahead predictive control of nonlinear systems," *Automatica*, vol. 36, no. 4, pp. 485–495, 2000.

[10] S. A. Billings and Q. M. Zhu, "Rational model identification using an extended least-squares algorithm," *International Journal of Control*, vol. 54, no. 3, pp. 529–546, 1991.

[11] Q. M. Zhu and S. A. Billings, "Recursive parameter estimation for nonlinear rational models," *Journal of Systems Engineering*, vol. 1, pp. 63–67, 1991.

[12] S. A. Billings and Q. M. Zhu, "A structure detection algorithm for nonlinear dynamic rational models," *International Journal of Control*, vol. 59, pp. 1439–1463, 1994.

[13] Q. M. Zhu and S. A. Billings, "Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks," *International Journal of Control*, vol. 64, no. 5, pp. 871–886, 1996.

[14] Q. M. Zhu, "An implicit least squares algorithm for nonlinear rational model parameter estimation," *Applied Mathematical Modelling*, vol. 29, no. 7, pp. 673–689, 2005.

[15] S. A. Billings and S. Chen, "Identification of non-linear rational systems using a prediction-error estimation algorithm," *International Journal of Systems Science*, vol. 20, no. 3, pp. 467–494, 1989.

[16] B. Q. Mu, E. W. Bai, W. X. Zheng, and Q. M. Zhu, "A globally consistent nonlinear least squares estimator for identification of nonlinear rational systems," *Automatica*, vol. 77, pp. 322–335, 2017.

[17] Q. M. Zhu, "A back propagation algorithm to estimate the parameters of nonlinear dynamic rational models," *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 169–187, 2003.

[18] Q. M. Zhu, D. L. Yu, and D. Y. Zhao, "An enhanced linear Kalman filter (EnLKF) algorithm for parameter estimation of nonlinear rational models," *International Journal of Systems Science*, vol. 48, no. 3, pp. 451–461, 2017.

[19] S. A. Billings and Q. M. Zhu, "Nonlinear model validation using correlation tests," *International Journal of Control*, vol. 60, no. 6, pp. 1107–1120, 1994.

[20] Q. M. Zhu, L. F. Zhang, and A. Longden, "Development of omni-directional correlation functions for nonlinear model validation," *Automatica*, vol. 43, no. 9, pp. 1519–1531, 2007.

[21] K. S. Narendra and K. Parthasapathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[22] B. G. Romanchuk and M. C. Smith, "Incremental gain analysis of piecewise linear systems and application to the antiwindup problem," *Automatica*, vol. 35, no. 7, pp. 1275–1283, 1999.

[23] L. Ozkan, M. V. Kothare, and C. Georgakis, "Model predictive control of nonlinear systems using piecewise linear models," *Computers & Chemical Engineering*, vol. 24, no. 2–7, pp. 793–799, 2000.

[24] T. Tsuji, B. H. Xu, and M. Kaneko, "Adaptive control and identification using one neural network for a class of plants with uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 4, pp. 496–505, 1998.

[25] Q. M. Zhu, Z. Ma, and K. Warwick, "Neural network enhanced generalised minimum variance self-tuning controller for nonlinear discrete-time systems," *IEE Proceedings - Control Theory and Applications*, vol. 146, no. 4, pp. 319–326, 1999.

[26] A. Isidori, L. Marconi, and A. Serrani, "New results on semi-global output regulation of nonminimum-phase nonlinear systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, pp. 1467–1472, Las Vegas, NV, USA, December 2002.

[27] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, London, 1991.

[28] Y. Q. Li, Z. S. Hou, Y. J. Feng, and R. H. Chi, "Data-driven approximate value iteration with optimality error bound analysis," *Automatica*, vol. 78, pp. 79–87, 2017.

[29] M. Fliess and C. Join, "Model-free control," *International Journal of Control*, vol. 86, no. 12, pp. 2228–2252, 2013.

[30] Q. M. Zhu, D. Y. Zhao, and J. H. Zhang, "A general U-block model-based design procedure for nonlinear polynomial control systems," *International Journal of Systems Science*, vol. 47, no. 14, pp. 3465–3475, 2016.

[31] Q. M. Zhu and L. Z. Guo, "A pole placement controller for nonlinear dynamic plant," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 216, no. 6, pp. 467–476, 2002.

[32] W. X. Du, X. L. Wu, and Q. M. Zhu, "Direct design of a U-model-based generalized predictive controller for a class of non-linear (polynomial) dynamic plants," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 226, no. 1, pp. 27–42, 2012.

[33] C. Kravaris and R. A. Wright, "Deadtime compensation for nonlinear processes," *AICHE Journal*, vol. 35, no. 9, pp. 1535–1542, 1989.

[34] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, MA, USA, 2nd edition, 1995.

[35] F. Ding and T. Chen, "Performance bounds of forgetting factor least-squares algorithms for time-varying systems with finite measurement data," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 3, pp. 555–566, 2005.

[36] T. Soderstrom and P. Stoica, *System Identification*, Prentice Hall International, Hemel Hempstead, 1989.

[37] W. C. Zhang, "On the stability and convergence of self-tuning control–virtual equivalent system approach," *International Journal of Control*, vol. 83, no. 5, pp. 879–896, 2010.

[38] F. Ding, L. Xu, and Q. M. Zhu, "Performance analysis of the generalised projection identification for time-varying systems," *IET Control Theory & Application*, vol. 10, no. 18, pp. 2506–2514, 2016.

[39] R. Kalaba and L. Tesfatsion, "Time-varying linear regression via flexible least squares," *Computers and Mathematics with Applications*, vol. 17, no. 8-9, pp. 1215–1245, 1989.

[40] J. Na, G. Herrmann, and K. Q. Zhang, "Improving transient performance of adaptive control via a modified reference model and novel adaptation," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 8, pp. 1351–1372, 2017.

[41] J. Na, M. N. Mahyuddin, G. Herrmann, X. M. Ren, and P. Barber, "Robust adaptive finite-time parameter estimation and control for robotic systems," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3045–3071, 2015.