Hindawi
Complexity
Volume 2020, Article ID 1050254, 13 pages
https://doi.org/10.1155/2020/1050254

Research Article

U-Model and *U*-Control Methodology for Nonlinear Dynamic Systems

Weicun Zhang[1] **Quanmin Zhu**[2] **Saleh Mobayen,**[3,4] **Hao Yan**[5] **Ji Qiu,**[2] **and Pritesh Narayan**[2]

[1]*Key Laboratory of Knowledge Automation for Industrial Processes of Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2]*University of the West of England, Coldharbour Lane, Bristol BS16 1QY, UK*
[3]*Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan*
[4]*Department of Electrical Engineering, University of Zanjan, Zanjan 45371-38791, Iran*
[5]*School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Weicun Zhang; weicunzhang@263.net

Received 26 April 2020; Accepted 9 June 2020; Published 30 June 2020

Guest Editor: Jing Na

This study presents the fundamental concepts and technical details of a *U*-model-based control (*U*-control for short) system design framework, including *U*-model realisation from classic model sets, control system design procedures, and simulated showcase examples. Consequently, the framework provides readers with clear understandings and practical skills for further research expansion and applications. In contrast to the classic model-based design and model-free design methodologies, this model-independent design takes two parallel formations: (1) it designs an invariant virtual controller with a specified closed-loop transfer function in a feedback control loop and (2) it determines the real controller output by resolving the inverse of the plant *U*-model. It should be noted that (1) this *U*-control provides a universal control system design platform for many existing linear/nonlinear and polynomial/state-space models and (2) it complements many existing design approaches. Simulation studies are used as examples to demonstrate the analytically developed formulations and guideline for potential applications.

1. Introduction

In general, there are three frameworks for control system design. The two popular frameworks are (1) the model-based approach and (2) the model-free/data-driven approach. The third is relatively new and that is (3) the model-independent/*U*-model-based approach. Here is a brief introduction to the three frameworks.

1.1. Model-Based Control System Design. To show this framework, consider the general cascade feedback control system shown in Figure 1, consisting of the following elements: G_p: plant, which could be modelled as a linear transfer function or a nonlinear dynamic equation in either the polynomial or state-space expression; G_c: classic controller; and G: closed-loop performance function, specified in advance by designers and/or users.

For a linear plant G_p, the controller G_c could be designed by means of

$$G_c = G_p^{-1}\frac{G}{1-G}. \tag{1}$$

For a nonlinear plant G_p, the controller could be designed as follows:

$$G_c = f\left(G_p, G\right), \tag{2}$$

where $f(*)$ is a function that links the plant and closed-loop performance to determine the control through a certain type of inversion.

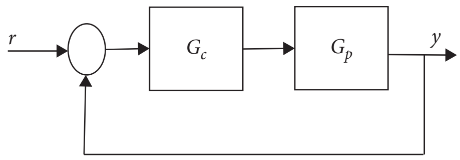

Figure 1: Model-based control.

Here are some remarks on the control-design framework:

(i) The model of the plant G_p is requested in advance, where the model sets include the linear/nonlinear polynomial and state-space expressions.

(ii) Advantages: there are many mature approaches available for this design framework [1–3]. It has been the predominant approach in academic research and industrial applications.

(iii) Disadvantage 1: the framework features unnecessary repetition in design. Taking a linear plant model as an example, it unnecessarily repeats the calculation of $(G/1-G)$ if the plant model changed in (1).

(iv) Disadvantage 2: it is difficult to design nonlinear plant-based control systems and it is also difficult to specify the transient responses of nonlinear control systems with this framework.

(v) Disadvantage 3: the model structure affects the approach needed for the linear/nonlinear and polynomial/state-space models, which is a common feature of model-based design frameworks.

1.2. Model-Free/Data-Driven Control System Design. There are various approaches to model-free control system design. A few well-known designs are described below.

(1) PID control by the Ziegler–Nichols approach [4]: this heuristic method of tuning a PID controller G_c (see Figure 1) has the following features:

No need for a model of the plant G_p, even when mild conditions are required for the controlled plants.

Advantage: it is the most commonly and easily used trial-and-error approach.

Disadvantages: this approach wastes experimental work to obtain plant models. Almost all engineering plants/processes and input/output measurements are possible to model in principle, although it is sometimes a difficult task.

(2) Iterative learning control (ILC) [5]: this framework (see Figure 2) has the following features:

No need for a model of the plant G_p in design, even when mild conditions are required for the controlled plant.

Requires iterative learning to improve the controller G_c with repeated reference stimulation; we finally achieve $G_cG_p = G_p^{-1}G_p = 1$.

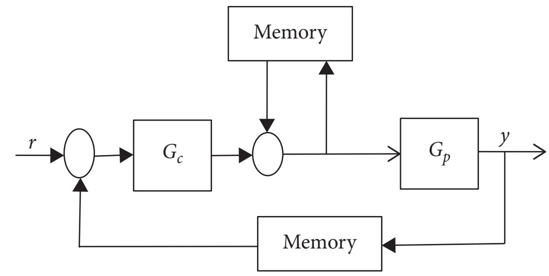

Figure 2: ILC.

Advantages: this approach considers every possibility for integrating past control information into the next round of control design. There is no need for a clear model structure.

Disadvantage 1: this approach wastes experimental work to obtain plant models, which is an issue with almost every engineering process.

Disadvantage 2: this approach is only available in a repeatable control environment under strict conditions

Disadvantage 3: it is challenging to control nonlinear dynamic plants with this approach.

(3) Model-free control (MFC) [6]: this framework (see Figure 1) has the following features:

No need for a model of the plant G_p, even under mild constraints (e.g., an ultralocal model $y^{(v)} = F + \alpha u$ where α is a coefficient and u is the controller output) on the controlled plants.

This approach is an enhanced PID controller $(u = -(F - y^{(v)} + k_p e + k_i \int e + k_d \dot{e}/\alpha))$ in which F needs to be estimated each time.

Advantages: the ultralocal model can be used to approximate complex dynamic plants and improve control performance in this approach.

Disadvantages: they are similar to those of PID controllers.

1.3. Model-Independent Control System Design [7–11]. This framework (see Figure 3) consists of the following: G_{c1}, linear invariant controller, and G_p^{-1}, dynamic inversion of plant.

$$G_{c1} = \left.\frac{G}{1-G}\right|_{G_p=1},$$
$$G_c = G_{c1}G_p^{-1}. \tag{3}$$

Some remarks are given on the control framework.

(i) It features model-independent controller design.

(ii) Advantage 1: the parallel design controller and dynamic inversion make the design procedure applicable to linear/nonlinear polynomial/state-space model structures. Transient responses can be specified for nonlinear systems. It is neat in design without waste/repetition if the plant model changes.

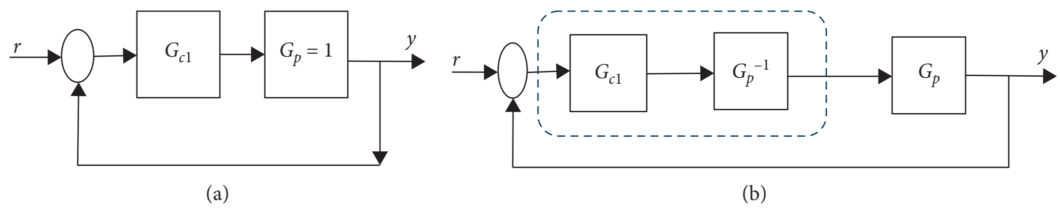

Figure 3: Model-independent control system design.

(iii) Advantage 2: this approach complements most existing design approaches.

(iv) Disadvantages: this approach is sensitive to model uncertainty; robustness is the paramount issue in designing control systems.

2. Discrete Time *U*-Model Set

The U-model expresses an explicit input-output relationship $U(y(t), u(t-1), \Psi(t))$ at time t with time-varying parameters $\Psi(t)$ to absorb dynamics implicitly. This is a control-oriented model and is derived from existing principle models or data-fitting models. This section explains (1) the definition of the U-model and the principles of converting classic models into U-models, (2) the dynamic inversion of U-polynomial models, and (3) the dynamic inversion of U-state-space models.

2.1. U-Models. Definition: for a single input ($u \in \mathbb{R}$ single) and single output ($y \in \mathbb{R}$) dynamic system $\sum$, assign a triplet (U, Ψ, f_U), where $U = \{U_\alpha \mid \alpha \in u(t-1)\}$ is a vector of appropriate dimension and $\Psi = \{\Psi_\alpha \mid \alpha \notin u(t-1)\}$ is a dynamic absorbing vector of appropriate dimension that is associated with U. Accordingly, system U-model $\sum_{U\text{-model}}$ is defined as a polynomial/rational system, where the polynomial/rational function $f_U = \{f_U(\alpha) \mid \alpha \in \Psi, U\}$ is a mapping f_U: $u(t-1) \xrightarrow{\Psi \cup U} y(t) \in \mathbb{R}$ from the input space to the output space.

2.1.1. U-Model Realisation from Classic Polynomials. Consider a general classical SISO polynomial model in the form of

$$y(t) = f_P(\Phi(*), \Theta),\quad \Phi(*) = \Phi(Y_{t-1}, U_{t-1}), \tag{4}$$

where $\begin{cases} y(t) \in \mathbb{R} \\ u(t-1) \in \mathbb{R} \end{cases}$ are the output/input, respectively, at $t \in \mathbb{Z}^+$ and $\Phi(*) = [\,\phi_0(*) \ \cdots \ \phi_L(*)\,] \in \mathbb{R}^{L+1}$, where Y_{t-1}, U_{t-1} are expanded from the output and input, respectively, in the proper dimensions. Let $\phi_i(*) = \begin{cases} \phi_i y(t-1), \ldots, y(t-n) \\ u(t-1), \ldots u(t-n) \\ \forall i = 0 \cdots n \end{cases}$, where n is the plant dynamic order and $\Theta = [\,\theta_0 \ \cdots \ \theta_L\,] \in \mathbb{R}^{L+1}$ is the associated parametric vector. Let the function f_P: $u \longrightarrow y$ be a polynomial mapping of the input space to the output space.

The vector form of the expanded equation (4) is given as follows:

$$y(t) = \Phi^T \Theta = \sum_{l=0}^{L} \phi_l(*)\theta_l, \tag{5}$$

where the bases $\phi_l(*)$ are the smooth functions in the space expanded from the past inputs/outputs, for example, $y^3(t-2)u(t-1)$, $u^3(t-1)$, $y(t-1)y(t-5)$, and the associated coefficients θ_l are real constants.

In other terms, this is a general expression of a nonlinear autoregressive moving average with exogenous input model (NARMAX) [12].

To realise a U-model from this classical polynomial, set up an absorbing rule.

Absorbing rule: let μ: $R^{L+1} \longrightarrow R^{M+1}$ be a map from a polynomial f_P to its U-polynomial f_U and suppose that its inverse μ^{-1} exists; therefore, it has

$$f_P(P(*), \Theta) \xrightarrow{\mu} f_U(\Psi(*), U(u(t-1))). \tag{6}$$

The mapping has some proper algebra properties as [8].

$$\begin{aligned}
&(a)\ \forall (P(*), \Theta), (\Psi(*), U(u(t-1))),\\
&f_P(*) = f_U(*) \Longrightarrow (P(*), \Theta) = (\Psi(*), U(u(t-1))\\
&(b)\ \forall (P(*), \Theta) \in f_P, \exists (\Psi(*), U(u(t-1))) \in f_U,\\
&\mu(P(*), \Theta) = (\Psi(*), U(u(t-1)))\\
&(c)\ \mu^{-1} \cdot \mu = I.
\end{aligned} \tag{7}$$

Accordingly, with reference to (7), the mapping is (a) injective (one to one), (b) surjective (onto) and bijective as both (a) and (b), and (c) invertible (I is an identity function). In system aspect, the map, except making the structure expression changed, does not change any characteristics of both models, such as output response, stability, dynamics, and statics.

The absorbing rule is a formation of $\Psi(*)$ from the polynomial f_P with reference to $u(t-1)$: first identify a control basis function $U(u(t-1))$ and then absorb all the other associated functions as a coefficient that varies with time.

Therefore, using the absorbing rule, realising f_U mapped from polynomial f_P (5) gives the following:

$$y(t) = \Psi^T U = \sum_{j=0}^{M} \psi_j(t) U_j(u(t-1)). \tag{8}$$

This function is expanded from the above nonlinear function f_P as a polynomial in terms of $u(t-1)$. M is the number of items associated with input $u(t-1)$ and the time-varying parameter vector $\Psi(*) = [\psi_0(t) \cdots \psi_M(t)] \in \mathbb{R}^{M+1}$ is a function derived from absorbing the other regression terms and the coefficients.

Example 1. Consider the polynomial model as shown below:

$$y(t) = 0.2 * \sin(y(t-1)) + u(t-1)\exp\left(-y^2(t-1)\right) - 0.8y(t-2)u(t-2)u^3(t-1). \quad (9)$$

Absorbing the terms associated with $u(t-1)$ into the vector $\Psi(*)$ gives the corresponding U-model realisation as follows:

$$y(t) = \psi_0(t) + \psi_1(t)U_1(u(t-1)) + \psi_2(t)U_2(u(t-1)), \quad (10)$$

where

$$\begin{aligned} \psi_0(t) &= 0.2 * \sin(y(t-1)), \\ \psi_1(t) &= \exp\left(-y^2(t-1)\right), \\ \psi_2(t) &= -0.8y(t-2)u(t-2), \\ U_1(u(t-1)) &= u(t-1), \\ U_2(u(t-1)) &= u^3(t-1). \end{aligned} \quad (11)$$

2.1.2. U-Model Realisation from Classic Rational Models. Rational model, also known as total nonlinear model [13], is a ratio of two polynomials as follows:

$$y(t) = f_r(\Phi(*), \Theta) = \frac{f_{\text{pn}}(\Phi_n(*), \Theta_n)}{f_{\text{pd}}(\Phi_d(*), \Theta_d)}. \quad (12)$$

Here f_r is a rational function, the ratio of the f_{pn}/numerator polynomial and f_{pd}/denominator polynomial, which are maps of the input space into the output space. The other definitions follow from the polynomial model above. Note that this rational model is totally nonlinear in terms of parameter estimation and control input design [13].

Continuing with the U-polynomial model conversion, formulate the U-rational model expression as follows:

$$y(t) = \frac{\Psi_n^T U_n}{\Psi_d^T U_d} = \frac{\sum_{j=0}^{M_n} \psi_{\text{jn}}(t) U_{\text{jn}}(u(t-1))}{\sum_{j=0}^{M_d} \psi_{\text{jd}}(t) U_{\text{jd}}(u(t-1))}. \quad (13)$$

To obtain the model inversion for solving the roots, expand the model as follows:

$$y(t)\left(\sum_{j=0}^{M_d} \psi_{\text{jd}}(t) U_{\text{jd}}(u(t-1))\right) = \sum_{j=0}^{M_n} \psi_{\text{jn}}(t) U_{\text{jn}}(u(t-1)). \quad (14)$$

Example 2. Consider the rational model as follows:

$$y(t) = \frac{0.1y^3(t-2) + \sin(u(t-1)) + 0.5u^3(t-1)}{1 + \cos^2(y(t-2)) + u^2(t-1)}. \quad (15)$$

Absorbing the terms associated with $u(t-1)$ into the vectors $\Psi_n(*)$, $\Psi_d(*)$ gives the corresponding U-model realisation as follows:

$$\begin{aligned} f_{\text{pn}}(u(t-1)) &= \psi_{0n}(t) + \psi_{1n}(t)U_{1n}(u(t-1)) + \psi_{2n}(t)U_{2n}(u(t-1)), \\ f_{\text{pd}}(u(t-1)) &= \psi_{0d}(t) + \psi_{1d}(t)U_{1d}(u(t-1)), \end{aligned} \quad (16)$$

where

$$\begin{aligned} \psi_{0n}(t) &= 0.1y^3(t-2), \\ \psi_{0d}(t) &= 1 + \cos^2(y(t-2)), \\ \psi_{1n}(t) &= 1, \\ \psi_{2n} &= 0.5, \\ \psi_{1d}(t) &= 1, \end{aligned} \quad (17)$$

$$\begin{aligned} U_{1n}(u(t-1)) &= \sin(u(t-1)), \\ U_{2n}(u(t-1)) &= u^3(t-1), \\ U_{1d}(u(t-1)) &= u^2(t-1). \end{aligned} \quad (18)$$

2.1.3. U Realisation from a Classical State-Space Mode-Multilayer U-Model. For a general SISO state-space system model, it has

$$\begin{aligned} X(t+1) &= F(X(t), u(t)), \\ y(t) &= h(X(t)), \end{aligned} \quad (19)$$

where $\begin{cases} X \in \mathbb{R}^n \\ u \in \mathbb{R} \\ y \in \mathbb{R} \end{cases}$ denotes the state, the control, and the output at time $t \in \mathbb{Z}^+$, respectively. $F \in \mathbb{R}^n$ is a smooth mapping to represent the input to the state output, and $h \in \mathbb{R}$ is a smooth mapping to drive the states to the outputs. In this study, assume that the system relative degree r equals the system order n and has no unstable zero dynamics (i.e., the model reversible) and that the state X can be obtained through measurement or observation.

Convert state-space model (19) into a multilayer U-model expression as follows:

$$\begin{cases} x_1(t+1) = \sum_{j=0}^{M_1} \psi_{1j}(t) U_{1j}(x_2(t)), \\ \vdots, \\ x_{n-1}(t+1) = \sum_{j=0}^{M_{n-1}} \psi_{(n-1)j}(t) U_{(n-1)j}(x_n(t)), \\ x_n(t+1) = \sum_{j=0}^{M_n} \psi_{\text{nj}}(t) U_{\text{nj}}(u(t)), \\ y(t) = h(X(t)). \end{cases} \quad (20)$$

For each line, M_j is the number of terms associated with the next line state variable $x_{j+1}(t)$ and $\psi_{ij}(t) = [\psi_{j0}(t) \cdots \psi_{jM_j}(t)] \in \mathbb{R}^{M_j+1}$ $i \in 1 \ldots n$ are time-varying parameter vector functions absorbing the other state variables. In the penultimate line, M_n consists of the terms associated with control $u(t)$ and the time-varying vectors $\left[\psi_{n0}(t) \ \cdots \ \psi_{nM_n}(t)\right] \in \mathbb{R}^{M_n+1}$ absorb all the states associated with the control vector $\left[U_{n0}(t) \ \cdots \ U_{nM_n}(t)\right] \in \mathbb{R}^{M_n+1}$. Therefore, each line of the state-space equation is a U-polynomial model, consisting of a multilayer U-model expression.

To illustrate the realisation, consider a nonlinear system represented in terms of state-space model:

$$\begin{cases} x_1(t+1) = x_2(t) + 0.1x_1(t)x_2(t), \\ x_2(t+1) = -0.1x_1(t) - 0.7x_2(t) + u(t), \\ y(t) = x_1(t). \end{cases} \quad (21)$$

Take realisation of the corresponding multilayer U-model by using the absorbing rule as below:

$$\begin{cases} x_1(t+1) = \psi_{11}(t)U_{11}(x_2(t)), \\ x_2(t+1) = \psi_{20}(t) + \psi_{21}U_{12}(u(t)), \\ y(t) = x_1(t), \end{cases} \quad (22)$$

where

$$\begin{aligned} \psi_{11}(t) &= 1 + 0.1x_1(t), \\ \psi_{20}(t) &= -0.1x_1(t) - 0.7x_2(t), \\ \psi_{21}(t) &= 1, \\ U_{11}(x_2(t)) &= x_2(t), \\ U_{12}(u(t)) &= u(t). \end{aligned} \quad (23)$$

2.2. Inversion of U-Polynomial Models. For simplicity, consider the SISO polynomial U-model (28). Newton–Raphson algorithm [14] is a choice to determine the roots of U-models; that is, the roots are the candidates of controller output $u(t-1)$.

Iteratively, the root searching computation gives rise to the following formulation:

$$u_{k+1}(t-1) = u_k(t-1) - \left.\frac{y(t) - \sum_{j=0}^{M} \lambda_j(t)u_k^j(t-1)}{\left(d\left[\sum_{j=0}^{M} \lambda_j(t)u_j(t-1)\right]/du(t-1)\right)}\right|_{u_k(t-1)=u_k^j(t-1)}. \quad (24)$$

Here, index k is the iteration handle: generate the $(k+1)$th results from the kth iteration, $k>0$. There are also various root solving algorithms available [15]. In parallel, these algorithms are also applicable for U-rational model root solving based on (14).

It should be noted that, in simulation studies, MATLAB codes, such as *roots,* can be used to find accurate roots of the U-model equations.

2.3. Inversion of U-State Space Models. For simplicity, consider the SISO U-state space model (20). Inversion is a multilayer root solving procedure involving a back-stepping routine whenever $x_1(t+1)$ is known; each line of the equation iteratively uses the Newton–Raphson algorithm to obtain $x_1(t+1) \quad x_2(t+1) \quad \cdots$ in back-stepping order.

3. *U*-Model-Based Control System Design

A Chinese survey paper [16] has covered the major publications till 2012. Later, representative studies include "U-Block model technique" [8], "control of total nonlinear systems" [9], "U-model enhanced Smith predict control for time delayed nonlinear processes" [11], and "U-neural networks enhanced control system design" [10]. This section further expands/formulates the U-control framework with updated results, including newly introduced two parallel dynamic inversions in design, robust analysis, and a step-by-step procedure for U-control implementation.

3.1. U-Control Framework. Let G_p be general dynamic in any expression of linear/nonlinear and polynomial/state-space models. Assumingly, the plant has the mostly claimed properties as those claimed in the other representative works [17]. Accordingly

(1) The model inverse G_p^{-1} exists

(2) Lipschitz continuity is satisfied, and G_p and its inverse G_p^{-1} are diffeomorphic and globally uniformly Lipschitz in $\mathbb{R}^n$; that is,

$$\begin{aligned} &\|G(x_1) - G(x_2)\| \le \gamma_1 G\|x_1 - x_2\|, \ \forall x_1, x_2 \in \mathbb{R}^n \\ &\|G^{-1}(x_1) - G^{-1}(x_2)\| \le \gamma_2 G^{-1}\|x_1 - x_2\|, \ \forall x_1, x_2 \in \mathbb{R}^n, \end{aligned}$$

where x_1, x_2 are the states with G_p in expression of state space equation and γ_1, γ_2 are the Lipschitz coefficients.

For simplicity, but not losing generality, take consideration of a SISO (input $u \in \mathbb{R}^1$ and output $y \in \mathbb{R}^1$) U-model based control system, U-control system in short, which is constructed within an autonomous linear feedback control framework with a bracketed triplet of

$$\sum = \left(F_{\text{lfbc}} \ \ G_{c1} \ \ G_{ip}\right), \quad (25)$$

where F_{lfbc} is a linear feedback loop with functions, linear virtual controller $G_{c1}: y \longrightarrow u$, and virtual unit plant $G_{ip} = 1: u \longrightarrow y$.

This U-control system structure proposes a model-independent control procedure, because the designs of $G_{c1}: y \longrightarrow u$ and $G_{ip} = 1: u \longrightarrow y$ are independent. These two independent designs are explained below.

For design of the virtual linear controller $G_{c1}: y \longrightarrow u$, referring to Figure 3(a), it gives

$$G_{c1} = \frac{G}{1-G} = (1-G)^{-1}G, \quad (26)$$

where G is a specified closed-loop transfer function with proper dynamic/static responses.

For design of the virtual unit plant $G_{ip} = 1: u \longrightarrow y$, designing/formulating the plant inverse G_P^{-1} gives

$$G_{ip} = G_P^{-1}G_p = 1: u \longrightarrow y. \quad (27)$$

Remark 1. Regarding the merit of the design prototype, the established U-control system framework (25) has two independent inversion designs: (1) linear controller $G_{c1}: y \longrightarrow u$ without involving any plant model structures; therefore, it is also named as linear invariant controller [9]; (2) virtual plant unitisation $G_{ip} = G_P^{-1}G_p = 1: u \longrightarrow y$ applicable to almost all smooth dynamics models (note: hard nonlinear dynamic models could be sorted out along similar route in the subsequent studies). Therefore, the two designs are separately independent and connected within a linear feedback control loop.

Remark 2. Regarding the efficiency of the U-control system design, linear controller $G_{c1}: y \longrightarrow u$ is once-off design irrespective of plant model types and parameters. Plant inverter G_P^{-1} is formable for polynomial and state-space equations in U-model and numerically solvable for the roots to achieve $G_{ip} = G_P^{-1}G_p = 1: u \longrightarrow y$. Consequently, the control system design is reduced to the determination of the plant inverse G_P^{-1} once the linear controller designed. Consequently, the design procedure is that once-off $G_{c1}: y \longrightarrow u$ design and $G_P^{-1} \Longrightarrow G_{ip} = G_P^{-1}G_p = 1: u \longrightarrow y$ follow-up design to keep the same closed-loop performance while plant model is changed.

Remark 3. Regarding the inversion involved in control system design, this is a must for any type of control system design. U-control provides concise structure and less computational effort for its two inversions (one is the inversion of specified linear closed-loop transfer function and the other is the inversion of plant U-model). This aspect can be explained through an inverse function Ψ^{-1}; for U-control systems, it is split into two separate functions of $G_{c1} = \Psi_1^{-1}(G)$ (linear dynamic inversion) and $G_p^{-1} = \Psi_2^{-1}(G_p)$($U$-model-based root solving). For the other popular control system design approaches, it is at least a function of $\Psi^{-1}(G_{c1}, G_p)$, which is a common formulation in classical linear feedback control system design. It should be noted that it is more complex in designing control systems with nonlinear plant models.

Remark 4. With regard to the relationship in control system design between the U-control and the other major approaches, U-control is a supplement to the approaches and takes away the need for the plant structures in controller design and clearly specifies the closed-loop dynamic/static performances. It should be noted that taking the transient performance into consideration when designing nonlinear control systems has received significant attention, and analysing their performance through linear system approaches is a key research domain [18]. U-control is therefore a promising procedure.

In some sense, those, using the other approaches, well-designed control systems could take U-control as a plug-in box to expand to control different types of plants.

Remark 5. As U-control is fundamentally based on the assumption $G_P^{-1}G_p = 1$, it is critical to consider the robustness of the resulting control system in the case of uncertainty, which is very common in practical systems. Surely two types of approaches are the candidates by adding additional robust control loop and/or adaptive loop.

3.2. Design Procedure. With reference to the aforementioned description and the block diagram in Figure 4, here we list a step-by-step design procedure.

(1) Establish a stable linear feedback control system structured in Figure 4. Assign G for the whole system transfer function in the closed-loop setup. Specify G by means of damping ratio, undamped natural frequency, and steady-state error and/or the other performance indices (such as poles and zeros and frequency response).
(2) Let the plant model be a constant unit or the virtual plant $G_{ip} = 1: u \longrightarrow y$ that has been achieved. determine a linear invariant controller G_{c1} by taking inverse of the closed-loop transfer function G using (26). Accordingly, the desired system output is equivalently determined by the output v of the controller G_{c1}.
(3) Convert plant model into U-model realisation $\sum_{U-\text{model}}$ with reference to the formulations presented in Section 2.
(4) To achieve $G_{ip} = 1: u \longrightarrow y$ to guarantee the desired output $y_{\text{desired}}(t) = v(t)$, determine the controller output by solving an equation $v(t) - \sum_{U-\text{model}} = 0$; that is, $u(t) \in v(t) - \sum_{U-\text{model}} = 0$.
(5) Locate/connect the blocks in Figure 5.

3.3. U-Model-Based Adaptive Control. This was first studied in recent publications [9, 19]. Figure 6 shows a double-looped (feedback control and adaptation) diagram that adds an adaptation role in dealing with uncertainties and disturbance by online updating model parameters. Interested readers can find the details in the aforementioned reference. Compared with the classic adaptive control scheme, adaptive U-control does not request controller design in each updating step; it only updates the plant model, while the controller is fixed. Here only the framework is explained briefly and the detailed expansions will be reported in the future publications.

3.4. Robustness Analysis of U-Control. This section presents the robustness analysis of U-control based on discrete-time H_∞ using linear matrix inequalities (LMI) technique. Consider the state-space equation in terms of multilayer U-realisation (20) with an external disturbance vector $W(t) = [w_1(t) \cdots w_n(t)]^T$ as

$$\begin{cases} x_1(t+1) = \displaystyle\sum_{j=0}^{M_1} \Psi_{1j}(t) U_{1j}(x_2(t)) + w_1(t), \\ \vdots, \\ x_{n-1}(t+1) = \displaystyle\sum_{j=0}^{M_{n-1}} \Psi_{(n-1)j}(t) U_{(n-1)j}(x_n(t)) + w_{n-1}(t), \\ x_1(t+1) = \displaystyle\sum_{j=0}^{M_n} \Psi_{nj}(t) U_{nj}(u(t)) + w_n(t), \\ y(t) = h(X(t)). \end{cases} \tag{28}$$

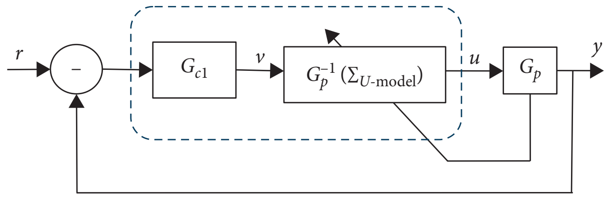

Figure 4: U-control framework.

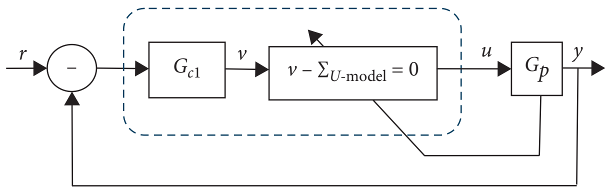

Figure 5: U-control implementation.

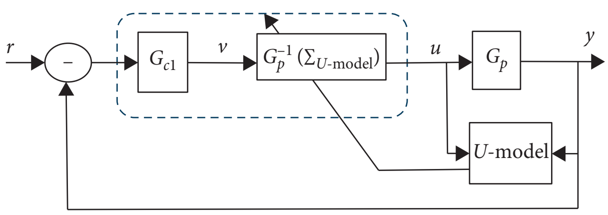

Figure 6: U-model-based adaptive control.

Remark 1. Assume that the elements of the external disturbance vector are bounded; that is, $|w_i(t)| < d_i$, $\forall i = 1 \cdots n$, where d_i is a positive constant.

To provide the robustness analysis, take one single line state of $x_n = (t+1)$ from state-space equation (28) at first, and then extend the analysis to the other state variables $x_i(t+1)$. Accordingly, take out

$$T_{\mathrm{wx}}: \begin{cases} x_n(t+1) = \sum_{j=0}^{M_n} \left\{ \Psi_{nj}(t) U_{nj}(X_n(t)) + w_{nj}(t) \right\}, \\ y(t) = h(x(t)). \end{cases} \tag{29}$$

The control objective is to minimize the effect of the external disturbance w_n on the state vector x_n. This study takes the discrete-time H_∞ robust control technique into consideration, where the robust control condition is

$$\|T_{\mathrm{wx}}\| < \beta \Rightarrow \sup \frac{\|x_n\|_{L_2}}{\|w_n\|_{L_2}} < \beta, \tag{30}$$

where β is a known constant defining the upper boundary of H_∞ performance index. Equation (30) can be rewritten as

$$\|x_n\|_{L_2}^2 < \beta^2 \|w_n\|_{L_2}^2, \tag{31}$$

or equivalently

$$\beta^{-1} \|x_n\|_{L_2}^2 - \beta \|w_n\|_{L_2}^2 < 0, \tag{32}$$

with

$$\begin{cases} \|x_n\|_{L_2}^2 = \sum_{t=0}^{\infty} x_n^T(t) x_n t, \\ \|w_n\|_{L_2}^2 = \sum_{t=0}^{\infty} w_n^T(t) w_n t. \end{cases} \tag{33}$$

From the above formulations, we have

$$\Longrightarrow \sum_{t=0}^{\infty} \left\{ \beta^{-1} x_n^T(t) x_n(t) - \beta w_n^T(t) w_n(t) \right\} < 0. \tag{34}$$

Construct the positive-definite Lyapunov function with

$$V(x_n(t)) = x_n^T(t) Q x_n(t) > 0, \tag{35}$$

where $Q > 0$. Suppose that the gradient of the Lyapunov function $(\nabla V(x_n(t)))$ is satisfied in the following inequality:

$$\nabla V(x_n(t)) + \beta^{-1} x_n^T(t) x_n(t) - \beta w_n^T(t) w_n(t) < 0. \tag{36}$$

In order to prove condition of (36), take summation (Σ) of all terms as

$$\sum_{t=0}^{\infty} \nabla V(x_n(t)) + \sum_{t=0}^{\infty} \left\{ \beta^{-1} x_n^T(t) x_n(t) - \beta w_n^T(t) w_n(t) \right\} < 0. \tag{37}$$

Since the first term of (37) is positive, the second term is always negative; that is,

$$\sum_{t=0}^{\infty} \left\{ \beta^{-1} x_n^T(t) x_n(t) - \beta w_n^T(t) w_n(t) \right\} < 0. \tag{38}$$

which is the same as condition (34). Then, inequality (36) is a correct assumption.

Determine the gradient of the Lyapunov function by

$$\nabla V(x_n(t)) = x_n^T(t+1) Q x_n(t+1) - x_n^T(t+1) Q x_n^T(t). \tag{39}$$

By substituting (39) into (36), we have

$$x_n^T(t+1) Q x_n(t+1) - x_n^T(t) Q x_n(t) + \beta^{-1} x_n^T(t) Q x_n(t) - \beta w_n^T(t) w_n(t) < 0. \tag{40}$$

Now, substituting $x_n(t+1)$ from (28) into (40), we have

$$\begin{aligned} &\sum_{j=0}^{M_n} \left\{ \Psi_{nj}(t) U_{nj}(x_n(t)) + w_{nj}(t)^T Q \left(\Psi_{nj}(t) U_{nj}(x_n(t)) + w_{nj}(t) \right) \right\} \\ &- \sum_{j=0}^{M_n} \left\{ \begin{array}{c} \left(\Psi_{nj}(t-1) U_{nj}(x_n(t-1)) + w_{nj}(t-1) \right)^T Q \\ \left(\Psi_{nj}(t-1) U_{nj}(x_n(t-1)) + w_{nj}(t-1) \right) \end{array} \right\} \\ &+ \sum_{j=0}^{M_n} \left\{ \begin{array}{c} \beta^{-1} \left(\Psi_{nj}(t-1) U_{nj}(x_n(t-1)) + w_{nj}(t-1) \right)^T \\ \left(\Psi_{nj}(t-1) U_{nj}(x_n(t-1)) + w_{nj}(t-1) \right) \end{array} \right\} \\ &- \sum_{j=0}^{M_n} \left\{ \beta \left(w_{nj}^T(t) w_{nj}(t) \right) \right\} < 0. \end{aligned} \tag{41}$$

In what follows, for simplicity, shorten the following notations as

$$\begin{cases} \Psi_{nj}(t) \triangleq \Psi_t, \\ \Psi_{nj}(t-1) \triangleq \Psi_{t-1}, \\ U_{nj}(x_n(t)) \triangleq U_t, \\ U_{nj}(x_n(t-1)) \triangleq U_{t-1}, \\ w_{nj}(t) \triangleq w_t, \\ w_{nj}(t-1) \triangleq w_{t-1}. \end{cases} \quad (42)$$

Then (41) is expressed as

$$\begin{aligned} &\sum \left\{ \begin{array}{c} U_t^T \psi_t^T Q \psi_t U_t + U_t^T \psi_t^T Q w_t \\ + w_t^T Q \psi_t U_t + w_t^T Q w_t \end{array} \right\} \\ &\quad - \sum \left\{ \begin{array}{c} U_{t-1}^T \psi_{t-1}^T Q \psi_{t-1} U_{t-1} + U_{t-1}^T \psi_{t-1}^T Q w_{t-1} \\ + w_{t-1}^T Q \psi_{t-1} U_{t-1} + w_{t-1}^T Q w_{t-1} \end{array} \right\} \\ &\quad + \sum \beta^{-1} \left\{ \begin{array}{c} U_{t-1}^T \psi_{t-1}^T \psi_{t-1} U_{t-1} + U_{t-1}^T \psi_{t-1}^T w_{t-1} \\ + w_{t-1}^T \psi_{t-1} U_{t-1} + w_{t-1}^T w_{t-1} \end{array} \right\} \\ &\quad - \sum \left\{ \beta \left(w_t^T w_t \right) \right\} < 0. \end{aligned} \quad (43)$$

Furthermore, it can be expressed in terms of quadratic form

$$\begin{aligned} &\sum \left(U_t^T \psi_t^T \{Q\} \psi_t U_t \right) + \sum \left(U_t^T \psi_t^T \{Q\} w_t \right) + \sum \left(w_t^T \{Q\} \psi_t U_t \right) \\ &\quad + \sum \left(w_t^T \{Q - \beta I\} w_t \right) + \sum \left(U_{t-1}^T \psi_{t-1}^T \left\{ \beta^{-1} I - Q \right\} \psi_{t-1} U_{t-1} \right) \\ &\quad + \sum \left(U_{t-1}^T \psi_{t-1}^T \left\{ \beta^{-1} I - Q \right\} w_{t-1} \right) \\ &\quad + \sum \left(w_{t-1}^T \left\{ \beta^{-1} I - Q \right\} \psi_{t-1} U_{t-1} \right) \\ &\quad - \sum \left(w_{t-1}^T \left\{ \beta^{-1} I - Q \right\} w_{t-1} \right) < 0, \end{aligned} \quad (44)$$

and then matrix form of

$$\begin{bmatrix} \sum U_t \psi_t \\ \sum w_t \\ \sum U_{t-1} \psi_{t-1} \\ \sum w_{t-1} \end{bmatrix}^H H \begin{bmatrix} \sum U_t \psi_t \\ \sum w_t \\ \sum U_{t-1} \psi_{t-1} \\ \sum w_{t-1} \end{bmatrix} < 0, \quad (45)$$

with

$$H = \begin{bmatrix} Q & Q & 0 & 0 \\ Q & Q - \beta I & 0 & 0 \\ 0 & 0 & \beta^{-1} I - Q & \beta^{-1} I - Q \\ 0 & 0 & \beta^{-1} I - Q & \beta^{-1} I - Q \end{bmatrix} < 0. \quad (46)$$

Now, applying the Schur complement [20] on (46), we get

$$\begin{cases} F_{22} < 0, \\ F_{11} - F_{12} (F_{22})^{-1} F_{21} < 0, \end{cases} \quad (47)$$

where

$$\begin{aligned} F_{11} &= \begin{bmatrix} Q & Q \\ Q & Q - \beta I \end{bmatrix}, \\ F_{12} = F_{21} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\ F_{22} &\begin{bmatrix} \beta^{-1} I - Q & \beta^{-1} I - Q \\ \beta^{-1} I - Q & \beta^{-1} I - Q \end{bmatrix} < 0. \end{aligned} \quad (48)$$

Then condition of (47) can be simplified as

$$\begin{cases} \begin{bmatrix} \beta^{-1} I - Q & \beta^{-1} I - Q \\ \beta^{-1} I - Q & \beta^{-1} I - Q \end{bmatrix} < 0, \\ \begin{bmatrix} Q & Q \\ Q & Q - \beta I \end{bmatrix} < 0. \end{cases} \quad (49)$$

Defining a new variable $\gamma: = \beta^{-1}$ in the first inequality of (49), it changes to

$$LMI(Q, \gamma): \begin{bmatrix} \gamma I - Q & \gamma I - Q \\ \gamma I - Q & \gamma I - Q \end{bmatrix} < 0 \Longrightarrow \gamma I - Q < 0, \quad (50)$$

where the optimal values Q^* and γ^* can be calculated via Matlab LMI toolbox. The optimal value of β is correspondingly given by $\beta^* = (\gamma^*)^{-1}$. Applying Schur complement on the second inequality of (49), we have

$$\begin{cases} Q < 0, \\ Q - \beta I - Q(Q)^{-1} Q < 0 \longrightarrow -\beta I < 0, \end{cases} \quad (51)$$

which yields $-\gamma I < 0$. Then, from (51), it gives

$$LMI(Q, \gamma): \begin{bmatrix} Q & 0 \\ 0 & -\gamma I \end{bmatrix} < 0 \Longrightarrow Q^*, \gamma^*, \quad (52)$$

where the existence of optimal solutions for Q^* and γ^* means the robustness of the U-model system versus external disturbances.

The robustness analysis for the remainder of equations of state-space model (28) can be proved similar to the above-presented procedure. In addition, Virtual Equivalent System (VES) methods [21–23] can also be used for robustness analysis of U-model control systems.

4. Simulation Examples

This simulation demonstration selected three plant models: SISO Hammerstein model, SISO nonlinear state-space model, and an extended total nonlinear model. In the control system design, it formulated a commonly used pole placement controller for the three examples. The main purposes for designing the simulation tests of the U-control are as follows:

(1) To demonstrate the principle of model-independent design in U-control.

(2) To demonstrate the once-off design of the invariant controller which specifies the whole closed loop

performance. In analysis, as always the product of the dynamic inverter and plant is a unit constant, therefore, a uniquely specified invariant controller will achieve the same output performance with different dynamic plants in the selected examples.

(3) To demonstrate the workability and conciseness/simplicity of U-control, particularly in the design of nonlinear control systems.

(4) To demonstrate that U-control can supplement/enhance classic pole placement control.

From previous sections, the design is divided into two parallel blocks: (1) designing the linear invariant control G_{c1} (thus $v(t)$) by reversing the specified closed-loop transfer function and (2) determining the control input $u(t-1)$ by reversing the plant U-model equation.

For familiarisation of different notations used in U-control, this simulation section takes in $y_{t-j} = y(t-j)$, $j \in \mathbb{Z}^+$, $u_{t-j} = u(t-j)$, $j \in \mathbb{Z}^+$, $v_{t-j} = v(t-j)$, $j \in \mathbb{Z}^+$, $x_j(t+1) = x_{j(t+1)}$, $j \in \mathbb{Z}^+$, and $\psi_j(t) = \psi_j$, $j \in \mathbb{Z}^+$.

4.1. Design Invariant Control G_{c1}.

In a popular approach, the conventional pole placement control [24] assigns the closed-loop characteristic equation in terms of Z transform:

$$\begin{aligned} A(z) &= z^2 + a_1 z + a_2 \\ &= z^2 - 1.3205z + 0.4966. \end{aligned} \tag{53}$$

Equivalently the poles are located at $0.6603 \pm i0.2463$ within the unit circle (stable), a typical decayed oscillatory response with damping ratio of 0.7, and unit undamped natural frequency; this is a commonly used dynamic response index set.

Assign the numerator polynomial in the desired closed-loop transfer function as

$$B(z) = bz, \tag{54}$$

where the constant b is determined by steady-state error requirement to a given reference input. Accordingly, in this case study, to make the steady state follow a given step reference input without error, it sets up

$$b = A(z)|_{z=1} = 1 - 1.3205 + 0.4966 = 0.1761. \tag{55}$$

Thereby, the resultant transfer function is specified as

$$\frac{Y(z)}{R(z)} = G(z) = \frac{0.1761z^{-1}}{1 - 1.3205z^{-1} + 0.4966z^{-2}}. \tag{56}$$

It should be noted that when the condition $G_{\text{ip}} = 1: u \longrightarrow y$ is satisfied, it gives

$$\frac{V(z)}{R(z)} = G(z) = \frac{0.1761z^{-1}}{1 - 1.3205z^{-1} + 0.4966z^{-2}}, \tag{57}$$

where $V(z)$ is the Z transform of the controller G_{c1} output as shown in Figure 3.

To determine the linear invariant controller G_{c1}, temporarily, let the plant $G_p = 1$ or $G_{\text{ip}} = G_p^{-1} G_p = 1: u \longrightarrow y$. Then take inverse of the transfer function G to yield

$$\begin{aligned} G_{c1} &= \frac{G(z)}{1 - G(z)} \\ &= \frac{bz^{-1}}{1 + (a_1 - c)z^{-1} + a_2 z^{-2}} \\ &= \frac{0.1761z^{-1}}{1 - 1.4966z^{-1} + 0.4966z^{-2}}. \end{aligned} \tag{58}$$

The rest of the control system design will formulate the specific plant inverse G_p^{-1} in form of U-model for each selected example, which will be implemented in each related subsection.

4.2. Hammerstein Model: A SISO Nonlinear Polynomial [7].

The Hammerstein style model, a static (memoryless) nonlinear block, is cascaded with a linear differential equation (dynamic) and is a good representative of various nonlinear dynamic plants/processes. Its control has been widely studied with model-based approaches [25]. The simulation example selected [7] is as follows:

$$\begin{aligned} y_t &= 0.5y_{t-1} + x_{t-1} + 0.1x_{t-2}, \\ x_t &= 1 + u_{t-1} - u_{t-1}^2 + 0.2u_{t-1}^3, \end{aligned} \tag{59}$$

where $\{y_t, u_t, x_t\}$ are the plant output, input, and intermediate variable for the static nonlinear component output, respectively.

As explained above, the first step in U-control system design is generic to determine the linear invariant controller G_{c1}, that is, independent of the plant model and universally designed (as was done in the beginning of this section). The second step of the design is specifically working out the controller output u_{t-1} by inverting the plant model to find its U-model roots. Accordingly, to realise a U-model for the controller output, it uses the absorbing rule to convert the Hammerstein model into the following U-expression:

$$y_t = \psi_0 + \psi_1 u_{t-1} + \psi_2 u_{t-1}^2 + \psi_3 u_{t-1}^3, \tag{60}$$

where

$$\begin{aligned} \psi_0 &= 0.5y_{t-1} + 1 + 0.3x_{t-2}, \\ \psi_1 &= 1, \\ \psi_2 &= -1, \\ \psi_3 &= 0.2. \end{aligned} \tag{61}$$

Then replace the output y_t with the virtual controller output v_t (i.e., the desired output). Subsequently, it determines one of the roots by solving (60) as the controller output. This gives the following formula:

$$u_{t-1} \in \text{roots}\left(v_t - \psi_0 + \psi_1 u_{t-1} + \psi_2 u_{t-1}^2 + \psi_3 u_{t-1}^3 = 0\right). \tag{62}$$

Figure 7 illustrates the simulation results.

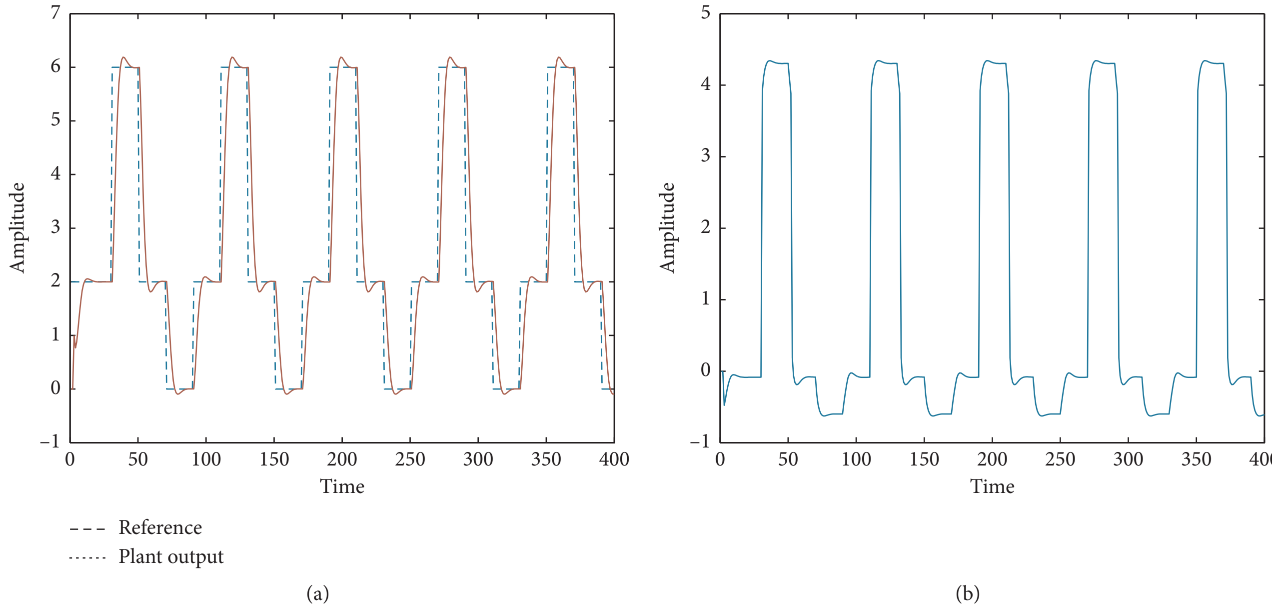

Figure 7: (a) Plant output. (b) Control input.

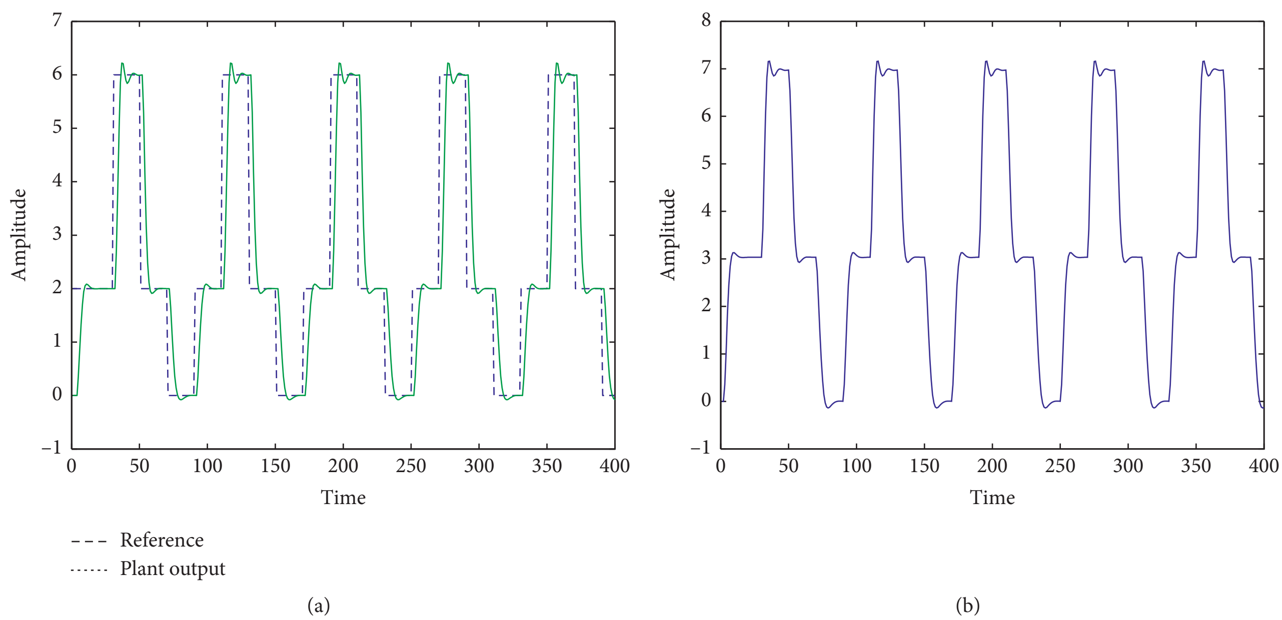

Figure 8: (a) Plant output. (b) Control input.

4.3. Nonlinear State-Space Model. The control of strict-feedback nonlinear systems is a widely studied, challenging topic [26]. Many leading publications have used neural network model-based approaches to approximate the model set as a pointwise linear model set to alternatively design equivalent linear control systems [27]. The simulation example for the state-space model is as follows:

$$\begin{cases} x_{1t} = x_{2(t-1)} + 0.1x_{1(t-1)}x_{2(t-1)}, \\ x_{2t} = -0.1x_{1(t-1)} - 0.7x_{2(t-1)} + u_{t-1}, \\ y_t = x_{1t}, \end{cases} \quad (63)$$

where $\{y_t, u_t, x_t\}$ denote the plant output and input and $x(t)$ is a state vector, respectively. This represents a second-order nonlinear dynamic plant.

Again, in the second step of U-control system design, it requires to work out the specific controller output u_{t-1} by inverting the plant U-model. The realised multilayer U-model is expressed as follows:

$$\begin{aligned} x_{1t} &= \psi_{11}x_{2(t-1)}, \\ x_{2t} &= \psi_{20} + \psi_{21}u_{t-1}, \\ y_t &= x_{1t}, \end{aligned} \quad (64)$$

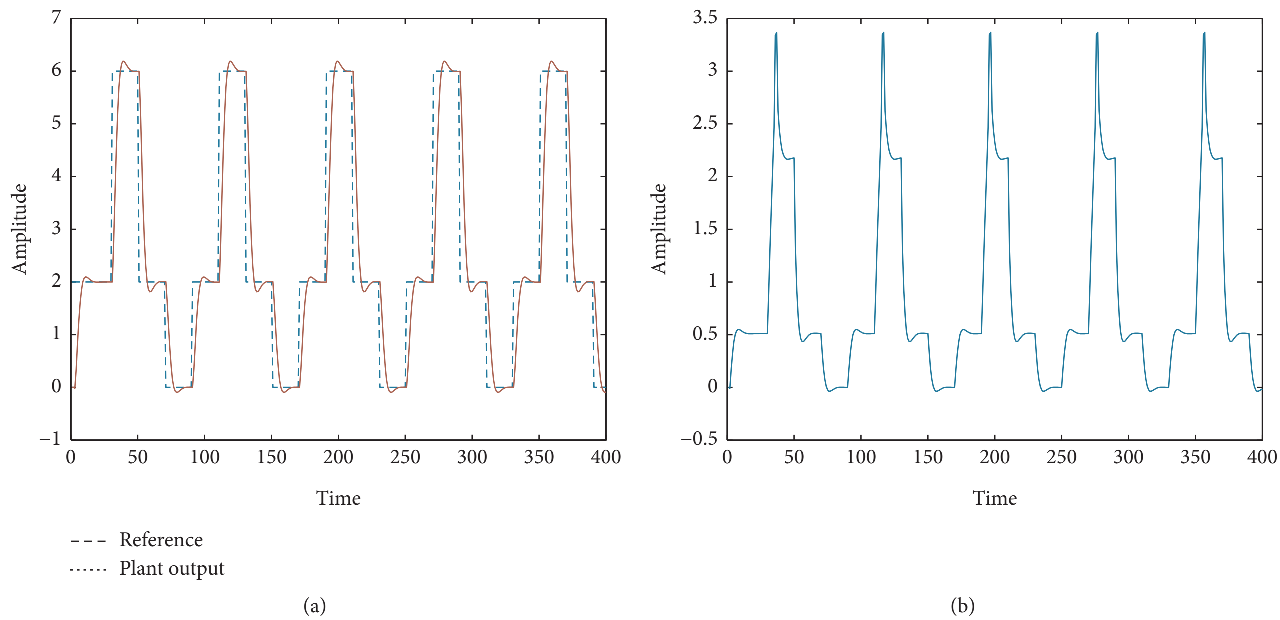

Figure 9: (a) Plant output. (b) Control input.

where $\psi_{11} = 1 + 0.1x_{1t}$, $\psi_{20} = -0.1x_{1(t-1)} - 0.7x_{2(t-1)}$, and $\psi_{21} = 1$.

This is a two-layer U-model structure. Accordingly, using back-stepping routine with initial $x_{1t} = v_t$ works out the controller output u_{t-1} by inverting each line of the equations, as specified in step 4 in U-control design procedure. The simulation results are shown in Figure 8.

4.4. Extended Total Nonlinear Model [9]. The control of nonlinear rational systems, which are modelled as ratios of two nonlinear polynomials, is even more challenging. Until a recent analytical U-model-based approach [9], these models were previously taken as examples of complex systems in neurocontrol system design. The difficulty is that rational model sets are subject to total nonlinearity (both in the parameters/identification and in input/control) [13]. The selected simulation example [9] with dynamics (time delay) and transcendental nonlinearities was as follows:

$$y(t) = \frac{0.5y(t-1) + \sin(u(t-1)) + u(t-1)}{1 + \exp(-y^2(t-1))}, \qquad (65)$$

where $\{y_t, u_t\}$ are the plant output and input, respectively. Once again, by applying the absorbing rule, it yields the following U-rational model:

$$y_t\left(1 + \exp\left(-y_{t-1}^2\right)\right) = 0.5y_{t-1} + \sin(u_{t-1}) + u_{t-1} \Longrightarrow,$$
$$\psi_{0d}\, y(t) = \psi_{0n} + \psi_{1n}\sin(u_{t-1}) + \psi_{2n}u_{t-1}. \qquad (66)$$

With the same linear invariant controller G_{c1} used as before, replacing the output y_t of (66) with the desired output v_t of (58) gives the following:

$$\psi_{0d}v_t = \psi_{0n} + \psi_{1n}\sin(u_{t-1}) + \psi_{2n}u_{t-1}. \qquad (67)$$

Subsequently, the control input u_{t-1} is obtained by the following:

$$\psi_{0d}v_t - \psi_{0n} - \psi_{1n}\sin(u_{t-1}) - \psi_{2n}u_{t-1} = 0. \qquad (68)$$

Figure 9 illustrates the simulation results. Again, the bench test confirms the performance of the U-control.

5. Conclusion

U-control has been featured in several publications. This tutorial has been presented to summarise and expand on the essential insights, formulations, and simulated case studies. We hope that this self-contained study can achieve the following purposes:

(1) Explain/demonstrate the principle of model-independent design in U-control
(2) Explain/demonstrate a universal design for multiple plant model structures
(3) Explain/demonstrate U-control workability and effectiveness/efficiency, particularly dealing with nonlinear plant control
(4) Explain/demonstrate U-control as a supplement to classic control system design frameworks

In terms of research techniques, compared with the two most popular control system design frameworks, model-based and model-free, this model-independent design effectively relieves the complexity involved in inverting the controller and plant together. The problem of inversion is reduced to inverting the plant model only, which means this framework results in an invariant controller that is universally applicable to the classic model sets and features no repetition if the plant model changes. The most critical issue with this design framework is its robustness because it relies on having $G_p^{-1}G_p = 1$. Accordingly, robust *U*-control is a central topic for research and applications. Additional demonstrations of its use in real cases will help to prepare it for wider application.

In research methodology, *U*-control is simple/concise and uses basic tools such as poles and zeros for analysing/designing linear system stability, transient responses (damping ratios and undamped natural frequencies), and the small gain theorem for robustness analysis. All of these are fundamental in postgraduate courses. However, *U*-control effectively combines them to provide solutions for challenging research problems. It is hoped that this technique will be user-friendly for industrial engineers working with ad hoc applications and easy to use for academics developing further enhancements of the method. As future work, *U*-model and *U*-control methodology can be integrated with other concepts in modelling and control of nonlinear dynamic systems, such as multiple model approaches [28–32].

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

This work was supported by the National Natural Science Foundation of China (nos. 61520106010, 61741302, and 51775032).

References

[1] K. Kim, M. Kim, D. Kim, and D. Lee, "Modeling and velocity-field control of autonomous excavator with main control valve," *Automatica*, vol. 104, pp. 67–81, 2019.

[2] S. Mobayen and J. Ma, "Robust finite-time composite nonlinear feedback control for synchronization of uncertain chaotic systems with nonlinearity and time-delay," *Chaos, Solitons & Fractals*, vol. 114, pp. 46–54, 2018.

[3] T. R. Oliveira, V. H. P. Rodrigues, and L. Fridman, "Generalized model reference adaptive control by means of global HOSM differentiators," *IEEE Transactions on Automatic Control*, vol. 64, no. 5, pp. 2053–2060, 2019.

[4] K. Ogata and Y. Yang, *Modern Control Engineering*, Prentice-Hall, Upper Saddle River, NJ, USA, 2002.

[5] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*, Springer, Berlin, Germany, 2003.

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

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

[8] Q. M. Zhu, D. Y. Zhao, and J. 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.

[9] Q. Zhu, L. Liu, W. Zhang, and S. Li, "Control of complex nonlinear dynamic rational systems," *Complexity*, vol. 2018, Article ID 8953035, 12 pages, 2018.

[10] Q. Zhu, W. Zhang, J. Zhang, and B. Sun, "*U*-neural network-enhanced control of nonlinear dynamic systems," *Neurocomputing*, vol. 352, pp. 12–21, 2019.

[11] X. Geng, Q. Zhu, T. Liu, and J. Na, "*U*-model based predictive control for nonlinear processes with input delay," *Journal of Process Control*, vol. 75, pp. 156–170, 2019.

[12] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, John Wiley & Sons, Hoboken, NJ, USA, 2013.

[13] Q. Zhu, Y. Wang, D. Zhao, S. 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.

[14] C. F. Gerald, *Applied Numerical Analysis*, Pearson Education India, Bengaluru, Karnataka, India, 2004.

[15] E. K. Chong and S. H. Zak, *An Introduction to Optimization*, John Wiley & Sons, Hoboken, NJ, USA, 2013.

[16] F.-X. Xu, Q.-M. Zhu, D. Zhao, and S.-Y. Li, "U-model based design methods for nonlinear control systems a survey of the development in the 1st decade," *Control and Decision*, vol. 28, no. 7, pp. 961–971, 2013.

[17] A. Isidori, *Nonlinear Control Systems*, Springer Science & Business Media, Berlin, Germany, 2013.

[18] B. M. Chen, T. H. Lee, K. Kemao Peng, and V. Venkataramanan, "Composite nonlinear feedback control for linear systems with input saturation: theory and an application," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 427–439, 2003.

[19] W. Du, X. Wu, and Q. Zhu, "Direct design of a U-model-based generalized predictive controller for a class of nonlinear (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.

[20] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory:," SIAM Studies in Applied Mathematics, Philadelphia, PA, USA, 1994.

[21] Z. Weicun, "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.

[22] W. Zhang and W. Wei, "Virtual equivalent system theory for adaptive control and simulation verification," *Scientia Sinica Informationis*, vol. 48, no. 7, pp. 947–962, 2018.

[23] Li Zhao, W. Zhang, and T. Chu, "Stability, convergence, and robustness of deterministic multivariable self-tuning control," *Chinese Journal of Engineering*, vol. 41, no. 9, pp. 1215–1221, 2019.

[24] K. J. Åström and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Boston, MA, USA, 2nd edition, 1995.

[25] K. P. Fruzzetti, A. Palazoğlu, and K. A. McDonald, "Nolinear model predictive control using Hammerstein models," *Journal of Process Control*, vol. 7, no. 1, pp. 31–41, 1997.

[26] M. Lv, Y. Wang, S. Baldi, Z. Liu, and Z. Wang, "A DSC method for strict-feedback nonlinear systems with possibly unbounded control gain functions," *Neurocomputing*, vol. 275, pp. 1383–1392, 2018.

[27] S. S. Ge, C. Yang, S.-L. Dai, Z. Jiao, and T. H. Lee, "Robust adaptive control of a class of nonlinear strict-feedback discrete-time systems with exact output tracking," *Automatica*, vol. 45, no. 11, pp. 2537–2545, 2009.

[28] W. Zhang and L. Zhao, "Survey and tutorial on multiple model methodologies in modelling, identification and

control," *International Journal of Modelling, Identification and Control*, vol. 32, no. 1, pp. 1–9, 2019.

[29] W. Zhang, Z. Zhang, Q. Li, and Z. Xue, "Multiple U-model control of uncertain discrete-time nonlinear systems," *International Journal of Modelling, Identification and Control*, vol. 34, no. 4, pp. 1–12, 2020.

[30] J. Li, W. Zhang, and Q. Zhu, "Weighted multiple-model neural network adaptive control for robotic manipulators with jumping parameters," *Complexity*, vol. 2020, Article ID 3172431, 12 pages, 2020.

[31] W. Zhang, S. Wang, and Y. Zhang, "Multiple-model adaptive estimation with A new weighting algorithm," *Complexity*, vol. 2018, Article ID 4789142, 11 pages, 2018.

[32] W. Zhang and J. Li, "Stable weighted multiple model adaptive control of continuous-time plant with large parameter uncertainties," *IEEE ACCESS*, vol. 7, pp. 144125–144133, 2019.