

## Research Article

# Algorithms for U-Model-Based Dynamic Inversion (UM-Dynamic Inversion) for Continuous Time Control Systems

Ruobing Li <sup>1</sup>, Quanmin Zhu <sup>1</sup>, Janice Kiely,<sup>1</sup> and Weicun Zhang <sup>2</sup>

<sup>1</sup>Department of Engineering Design and Mathematics, University of the West of England, Frenchy Campus, Coldharbour Lane, Bristol BS16 1QY, UK

<sup>2</sup>School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

Correspondence should be addressed to Ruobing Li; 195938486@qq.com

Received 28 April 2020; Accepted 2 June 2020; Published 16 July 2020

Guest Editor: Shubo Wang

Copyright © 2020 Ruobing Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To setup a universal proper user toolbox from previous individual research publications, this study generalises the algorithms for the U-model dynamic inversion based on the realisation of U-model from polynomial and state-space described continuous-time (CT) systems and presents the corresponding U-control system design in a systematic procedure. Then, it selects four CT dynamic plants plus a wind energy conversion system for simulation case studies in Matlab/Simulink to test/demonstrate the proposed U-model-based design procedure and dynamic inversion algorithms. This work can be treated as a U-control system design user manual in some sense.

## 1. Introduction

Linear control system design approaches can be divided into state-space model-based [1] and polynomial model-based, which have been well studied and are widely used. However, in an actual production process, nonlinear control is ubiquitous and more difficult as the superposition principle no longer holds, in contrast to linear systems; therefore, how to design a standard-compliant nonlinear control system to match desired performance properly is a hot issue. For nonlinear systems, a variety of analysis and design approaches already exist, and the most commonly used method to design a nonlinear control system is still linearization. However, the linearization method has certain limitations, and most linear control methods cannot be applied to the design of nonlinear systems directly. For example, compared with the linear polynomial model, nonlinear polynomial models, such as the Nonlinear Autoregressive Moving Average with exogenous inputs (NARMAX) [2] model, have been used widely in applications and academic research publications [3]; however, there is no systematic routine to convert it into an equivalent state-space model.

Generally, there are three methodologies for the nonlinear plant-model-based control system design, two widely used and one less-attended. The first approach is using linear expressions to describe the nonlinear state-space models by feedback linearization approach [4, 5] and then designing this linear-expression corresponding control systems by linear state-space approaches, which has been well studied in [1]. However, this case-by-case method requires certain skills in selecting the appropriate coordinate system and solving the equations requires extraeffort. Furthermore, this state-space linearization approach cannot be directly applied for nonlinear polynomial models. The second method is to use a time-varying linear model to fit the polynomial model, for example, state-dependent parameter (SDP) transformation [6, 7] method can use specified (desired) poles to transform a nonlinear closed-loop control system model to a linear transfer function expression. In summary, it is clear that these approaches for designing a nonlinear control system are trying to convert the original nonlinear system into a quasi-linear domain system firstly and then to choose an opportune linear control approach for the system. In the model structure, the other variables in the system can also determine this quasi-linear SDP transformation's

parameters [8]. The nonlinear polynomial model can be converted into a time-varying linear state-space model by the second method; however, there are obstacles for using this method because this design and transform procedure is not unique, that is, personal and subjective for selection of SDP models.

The third approach is the U-model-based design, which is relatively new and less-attended. The U-model is defined as a polynomial or state space function, with time-varying parameters, representing a class of smooth and analytic systems. Zhu et al. [9] proposed the use of a Newton–Raphson iterative algorithm for the root solving of the controller output function, which provided a basic procedure for the design of controllers in the U-model. Zhu and Guo [10] formally proposed the concept of U-model and established the U-model-based control, U-control in short, and system design framework, which provides a general routine to convert smooth nonlinear plant models into U-model. The U-model-based design method can be recognised as converting nonlinear models to time-varying parameter models associated with controller output  $u$ , that is, linear control-oriented model structure [10–13].

Regarding the research status of U-model-based control, the discrete-time systems has been studied with more attention, especially the representative approaches including pole placement control design method [13, 14], U-Smith predictor with input time delay [15], adaptive U-control of total nonlinear dynamic systems [16], U-neural network enhanced control [17], and underactuated coupled nonlinear adaptive control synthesis, using U-model for multivariable unmanned marine robotics [18]. However, the majority of U-control approaches have assumed that the plant model can be detected without errors or inaccuracies. Therefore, designing an adaptive U-model controller when the plant model is inaccurate, especially the robustness control, will be a hotspot and difficult study area for intensive research. At the same time, there is very little research on U-model-based control system design for continuous-time systems so far [19]. Consequently, the main purpose of this study is to provide a pack of dynamic inversion routines for U-model-based continuous-time control systems.

Compared with methods 1 (linear model approximation) and 2 (time-varying linear model approximation), aforementioned, the main contributions of this U-design method are as follows:

- (1) In dealing with nonlinearity, the U-model-based design method does not require linearization of the nonlinear models in advance. Instead, this nonlinear plant model-based system is designed directly using linear design methods.
- (2) In methodology, using those well-studied linear methods to design nonlinear control systems greatly reduces the complexity of the design procedure.
- (3) In design, once the closed loop system output is specified, the only remaining work is to calculate the output of the U model controller.
- (4) U-model-based design procedure is more general and effective for designing a linearly behaved control

system, which provides new insight and solutions to design the controller.

- (5) U-control can be applied together with the other well-developed control system design methods, such as pole placement control, sliding mode control, general predictive control, adaptive, and Smith predictive control [20, 21].
- (6) It should be noted that unless the plant model is accurately known, U-model dynamic inversion is very sensitive to internal uncertainties, so the whole control system performance.

Accordingly, the main contributions of this study are as follows:

- (1) Generalise dynamic inversion algorithms for continuous-time U-model
- (2) Generalise U-model-based design procedure for continuous-time dynamic plants in forms of linear/nonlinear and polynomial/state space
- (3) Showcases for bench tests and illustration of applications
- (4) An industrial backgrounded study: U-control of a wind energy conversion system

For the rest of the study, Section 2 generalises U-polynomial and U-state space model sets and its associated step-by-step U-control design procedure. Section 3 generalises the dynamic inversion algorithms. Section 4 presents a series of computational case studies to test/demonstrate the analytical results numerically and provides an effective procedure for testing designed U-control systems with computational experiments. Section 5 presents an industrial backgrounded case study from modelling, dynamic inversion, and U-control system design to simulation. Section 6 concludes the study.

## 2. U-Model and U-Control System Design

### 2.1. Polynomial U-Model: Single Layer Realisation

**2.1.1. Basic Polynomial U-Model.** Consider a general continuous-time U-model [19] for Single-Input and Single-Output (SISO) polynomial dynamic systems with a triplet of  $(y(t), u(t), \lambda(t))$ ,  $y(t) \in \mathbb{R}$ , and  $u(t) \in \mathbb{R}$  for the output, input, and parameter, respectively, at time  $t \in \mathbb{R}^+$ :

$$y^{(M)} = \sum_{j=0}^J \lambda_j(Y_{M-1}, U_{N-1}, \Theta) \begin{pmatrix} (N) \\ u \end{pmatrix}^j, \quad M \geq N, \quad (1)$$

where  $y^{(M)}$  and  $u^{(N)}$  are the  $M$ th and  $N$ th order derivatives of the plant output  $y$  and the plant input  $u$ , respectively, the time-varying parameter  $\lambda_j(*) \in \mathbb{R}^+$ , associated with the

input  $\begin{pmatrix} (N) \\ u \end{pmatrix}^j$ , absorbs all the other terms in  $Y_{M-1} = \begin{bmatrix} (m-1) & (m-2) & \dots & y \end{bmatrix} \in \mathbb{R}^M$ ,  $U_{N-1} = \begin{bmatrix} (n-1) & (n-2) & \dots & u \end{bmatrix} \in \mathbb{R}^N$ , and coefficients  $\Theta$ .

Here is an example for understanding, consider a classical NAMAX polynomial model:

$$\ddot{y} = (y - 1)\dot{y} + (1 + y^2)u + (1 + \dot{y}^2)u^2 + y + (y + \dot{y}^2)u^3. \quad (2)$$

Its U-model realisation can be determined with

$$\begin{cases} \dot{y} = \lambda_0 + \lambda_1 u + \lambda_2 u^2 + \lambda_3 u^3, \\ \lambda_0 = (y - 1)\dot{y} + y, \\ \lambda_1 = 1 + y^2, \\ \lambda_2 = 1 + \dot{y}^2, \\ \lambda_3 = y + \dot{y}^2. \end{cases} \quad (3)$$

Inspection of (2) and (3), the U-realisation is straightforward generally.

It should be remarked that the U-polynomial is the same as its presented classical polynomials in the model properties, but oriented expression for control system design [14].

**2.1.2. Extended U-Model (Rational Models).** Rational model is totally nonlinear [3], and its polynomial expression is a ratio of two polynomials, that is,

$$(M) \quad y = \frac{f_{pn}(\psi_n(*)\Theta_n)}{f_{pd}(\psi_d(*)\Theta_d)}, \quad (4)$$

where  $f_{pn}$  is the input-output mapping function of the polynomial in the numerator and  $f_{pd}$  is the input-output mapping function of the polynomial in the denominator. Its U-realisation is accordingly described by

$$(M) \quad y = \frac{\sum_{j=0}^n \lambda_{nj}(Y_{M-1}, U_{N-1}, \Theta) f_{nj} \begin{pmatrix} (N) \\ u \end{pmatrix}}{\sum_{j=0}^d \lambda_{dj}(Y_{M-1}, U_{N-1}, \Theta) f_{dj} \begin{pmatrix} (N) \\ u \end{pmatrix}}, \quad M \geq N, \quad (5)$$

where  $f_{nj}$  and  $f_{dj}$  are vector functions of the control vector  $(N)$  in numerator and denominator, respectively, and  $\lambda_{nj}$  and  $\lambda_{dj}$  are the associated parameters vectors absorbing all the other terms in the model. Here is a simple example of the rational model:

$$\dot{y} = \frac{0.1y^3 + \sin(u) + 0.5u^3}{1 + \cos(y) + u^2}. \quad (6)$$

Its U-model realisation can be determined with

$$\dot{y} = \frac{\lambda_{n0} + \lambda_{n1} \sin(u) + \lambda_{n2} u^2 + \lambda_{n3} u^3}{\lambda_{d0} + \lambda_{d1} u + \lambda_{d2} u^2}, \quad (7)$$

where

$$\begin{cases} \lambda_{n0} = 0.1y^3, \\ \lambda_{n1} = 1, \\ \lambda_{n2} = 0, \\ \lambda_{n3} = 0.5, \\ \lambda_{d0} = 1 + \cos(y), \\ \lambda_{d1} = 0, \\ \lambda_{d2} = 1. \end{cases} \quad (8)$$

**2.2. State Space U-Model: Multilayer Realisation.** For a general SISO CT state space model,

$$\begin{cases} \dot{X} = F(X, u), \\ y = H(X), \end{cases} \quad (9)$$

where  $u, y \in \mathbb{R}$ ,  $X \in \mathbb{R}^n$ , and  $F \in \mathbb{R}^n$  are smooth mapping to represent the input to the state and  $H \in \mathbb{R}$  is a smooth mapping to drive the states to the outputs. In this study, assume that there is no unstable zero dynamics (i.e., the model reversible) and that the state  $X$  can be obtained through measurement or observation.

Expand state-space model (9) into a multilayer polynomial expression as follows:

$$\begin{cases} \dot{x}_1 = F_1(x_1, x_2, \dots, x_n), \\ \dot{x}_2 = F_2(x_1, x_2, \dots, x_n), \\ \vdots \\ \dot{x}_n = F_n(x_1, x_2, \dots, x_n, u), \\ y = H(x_1, x_2, \dots, x_n). \end{cases} \quad (10)$$

Convert state-space model (10) into a multilayer U-model expression as follows:

$$\begin{cases} \dot{x}_1 = \sum_{i=0}^n \lambda_{1i} f_{1i}(x_2), \\ \dot{x}_2 = \sum_{i=0}^n \lambda_{2i} f_{2i}(x_3), \\ \vdots \\ \dot{x}_n = \sum_{i=0}^n \lambda_{ni} f_{ni}(u), \\ y = \sum_{i=0}^n h_i(x_1, x_2, \dots, x_n). \end{cases} \quad (11)$$

For each line of (11),  $\lambda_{**}$  and  $f_{**}$  are time-varying parameter absorbing all the other variables and the U-basis function, respectively.

For illustration, consider a nonlinear SISO system state-space model of

$$\begin{cases} \dot{x}_1 = x_1 + 0.5 \sin(x_2), \\ \dot{x}_2 = -x_1 + u, \\ y = x_1. \end{cases} \quad (12)$$

Using the absorbing rule to convert (12) into multilayer U-model as

$$\begin{cases} \dot{x}_1 = \lambda_{10} + \lambda_{11}f_{11}(x_2), \\ \dot{x}_2 = \lambda_{20} + \lambda_{21}f_{21}(u), \\ y = x_1, \end{cases} \quad (13)$$

where

$$\begin{cases} \lambda_{10} = x_1, \\ \lambda_{11} = 0.5, \\ f_{11}(x_2) = \sin(x_2), \\ \lambda_{20} = -x_1, \\ \lambda_{21} = 1, \\ f_{21}(u) = u. \end{cases} \quad (14)$$

**2.3. U-Control System Design.** Here is a systematic summary of the U-control framework.

Figure 1 shows the classical control system framework, where  $G_p$  is the plant model, which could be linear or nonlinear dynamics and can be described by polynomial and state space models [14]. Let  $G$  (not shown in the figure) be the closed-loop performance function, specified with ad hoc applications in advance by designers and/or customers,  $r$  is the reference, which is the desired output of the control system,  $e$  is the difference (error) between the output  $y$ , and the reference  $r$ .  $G_c$  is the designed controller.

The main principles of this kind of control system design framework are to generate a suitable control input signal  $u$  to drive the system output trajectory  $y$  following a set of specified closed loop performances (both transient and steady state).

Figure 2 shows the U-control system framework [3, 5], in which  $G_{c1}$  is a linear invariant controller, can be designed by  $G_{c1} = G/(1 - G)$  while  $G_p^{-1}G_p = 1$ , where  $G_p^{-1}$  is the controlled plant's dynamic inverse. U-control framework is applicable to both linear and nonlinear structures as long as dynamic inverse  $G_p^{-1}$  exist.

To explain the control system design procedure, consider a CT SISO linear closed-loop feedback control system framework with a set of  $(F, G_{c1}, G_{ip})$ :

$$\Sigma = (F, G_{c1}, G_{ip}), \quad (15)$$

where  $G_{ip} = G_p^{-1}G_p$ .

In general, the U-control system design procedure has two separate steps:

- (1) Assume the plant model  $G_p$  stable and bounded, and its inverse  $G_p^{-1}$  exists. From Figure 2, the controller which is shown in the dashed line block has two parts: the invariant controller  $G_{c1}$  and plant's dynamic inverter  $G_p^{-1}$ . To facilitate the design of  $G_p^{-1}$ , the convert plant model  $G_p$  (1) into its U-model. Alternatively, (15) can be expressed as

$$\Sigma = (F, U(G_{c1}, G_p^{-1}), G_p), \quad (16)$$

where  $(G_{c1}, G_p^{-1})$  is defined as U-controller. Determine  $G_p^{-1}$  to work out the controller output  $u$ , which

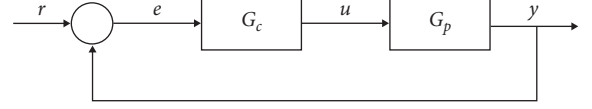


FIGURE 1: Classical control system framework.

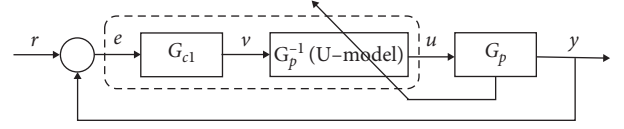


FIGURE 2: U-control system framework.

is the control input to the plant  $G_p$ . The eventual goal is to make plant output equal to the invariant (IV) controller output:  $v = y$ , that is, to reach  $G_p^{-1}G_p = 1$  under the proper dynamic inversion.

- (2) Design invariant controller  $G_{c1}$ . Figure 3 shows the U-control structure while  $G_p^{-1}G_p = 1$  achieved, that is,

$$\Sigma = (F, G_{c1}). \quad (17)$$

This is a type of linear control systems. Therefore, the desired closed-loop transfer function  $G$  can be expressed as  $G = G_{c1}/(1 + G_{c1})$ , where  $G$  can be effectively designed with two significant factors shaping linear system response, damping ratio  $\zeta$ , and undamped natural frequency  $\omega_n$ . Therefore, the invariant controller  $G_{c1}$  can be obtained by inverting  $G$  into  $G_{c1} = G/(1 - G)$  while  $G_p^{-1}G_p = 1$ .

As the invariant controller  $G_{c1}$  design is independent of plant  $G_p$ , the U-control allows once-off design for all stable-nonminimum phase plants, except designing the inverter  $G_p^{-1}$  of the considered plant.

### 3. UM-Dynamic Inversion

Nonlinear dynamic inversion (NDI) is a generic control technique in nature, that is, improving control performance through control system design. Currently, NDI has been a challenging research issue and practical significance in mechanical motion control systems, such as turbines, robots, and flying vehicles [22]. Basic NDI calculation procedure is differentiating the plant output equation results  $N$  times to find the direct relationship between the input  $u$  and  $N$ th order derivative of the output  $y$  under the Lie derivative formulation [22]. However, NDI is very sensitive and unstable in case of model inaccuracies and mismatch. In order to combat the uncertainty of the plant and improve the system robustness, Incremental Nonlinear Dynamic Inversion (INDI) [22] and adaptive INDI [23] have been introduced in another complicated formulation.

Different from the computational complexity of basic NDI under the Lie derivative expression, this study converts the plant model into U-model realisation in a systematic concise formulation, which is generically applicable to both polynomial and state space equations. This also establishes a

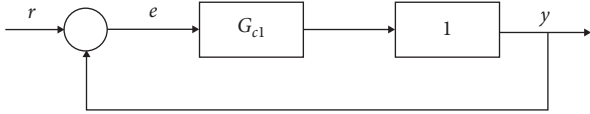


FIGURE 3: U-model-based simplified control system.

foundation for future development of robust UM-dynamic inversion.

The U-model-based dynamic inversion (UM-dynamic inversion) algorithm is to obtain the input  $u$  by solving the root from (1), that is,

$$G_p^{-1} \Leftrightarrow \begin{matrix} (N) \\ u \end{matrix} \in \begin{matrix} (M) \\ y \end{matrix} - \sum_{j=0}^J \lambda_j(Y_{M-1}, U_{N-1}, \Theta) \begin{pmatrix} (N) \\ u \end{pmatrix}^j = 0. \quad (18)$$

For the solution which exists, the systems must be Bounded Input and Bounded Output (BIBO) stable and no unstable zero dynamic (nonminimum phase).

### 3.1. Algorithms for Polynomial Models

**3.1.1. Linear Plant.** Use Laplace transform ( $S$  operator,  $(j) = (d^j y/dt^j) \Leftrightarrow s^j Y$ ,  $\int \dots \int y dt \Leftrightarrow (1/s^j)Y$ ) to express a set of general linear dynamic plants as

$$Y = \frac{\sum_{j=0}^{j=N} \beta_j s^{N-j}}{\sum_{j=0}^{j=M} \alpha_j s^{M-j}} u = \frac{\beta_0 s^N + \beta_1 s^{N-1} + \dots + \beta_{N-1} s + \beta_N}{\alpha_0 s^M + \alpha_1 s^{M-1} + \dots + \alpha_{M-1} s + \alpha_M} U, \quad (19)$$

$M \geq N,$

$$\lambda_0(Y_{M-1}, U_{N-1}, \Theta) = \frac{\sum_{j=1}^{j=M-1} \alpha_j (s^{M-j}/s^N)Y + \sum_{j=1}^{j=N-1} \beta_j (s^{N-j}/s^N)U}{\alpha_0}, \quad (23)$$

$$\lambda_1(Y_{M-1}, U_{N-1}, \Theta) = \frac{\beta_0}{\alpha_0},$$

Therefore, the alternative U-model is

$$\begin{matrix} (M-N) \\ y \end{matrix} = \sum_{j=0}^J \lambda_j(Y_{M-1}, U_{N-1}, \Theta) u^j. \quad (24)$$

**3.1.2. Nonlinear Plant.** For UM-dynamic inversion of (18), replace all the derivatives with integrals through division by the output derivative order, which is formulated as

$$\begin{matrix} (j) \\ y \end{matrix} = \begin{matrix} (M) \\ y \end{matrix} \left(\frac{1}{s}\right)^{M-j}, \quad (25)$$

where  $\begin{matrix} (j) \\ y \end{matrix} = (d^j y/dt^j)$

Here is the practical implementation of (18):

where  $Y$  and  $U$  are Laplace transform of the output and input, respectively, and  $M$  and  $N$  are the orders (highest power) of the denominator and numerator functions, respectively.

Accordingly, its U-realisation is given as

$$\begin{matrix} (M) \\ y \end{matrix} = \sum_{j=0}^1 \lambda_j(Y_{M-1}, U_{N-1}, \Theta) \begin{pmatrix} (N) \\ u \end{pmatrix}^j, \quad (20)$$

where

$$\lambda_0(Y_{M-1}, U_{N-1}, \Theta) = \frac{-\sum_{j=1}^{j=M-1} \alpha_j s^{M-j} Y + \sum_{j=1}^{j=N-1} \beta_j s^{N-j} U}{\alpha_0},$$

$$\lambda_1(Y_{M-1}, U_{N-1}, \Theta) = \frac{\beta_0}{\alpha_0}. \quad (21)$$

As the drives are sensitive to noise signals in applications, convert the operations into integral implementations by multiplying  $1/s^N$  on both sides of (20), and this gives

$$\begin{matrix} (M-N) \\ y \end{matrix} = \frac{\sum_{j=0}^1 \lambda_j(Y_{M-1}, U_{N-1}, \Theta) u}{s^N}, \quad (22)$$

where

$$G_p^{-1} \Leftrightarrow u \in y - \frac{\sum_{j=0}^J \lambda_j(Y_{M-1}, U_{N-1}, \Theta) \begin{pmatrix} (N) \\ u \end{pmatrix}^j}{s^M} = 0. \quad (26)$$

To illustrate the conversion to U-model from a nonlinear polynomial, consider an example of

$$\ddot{y} = (1 - y^2)\dot{y} - u + (1 + y)\dot{u} + (1 + y^2)\dot{u}^2 + \dot{u}^3. \quad (27)$$

In U-realisation, its derivative-based operation becomes

$$\dot{y} = \lambda_0 + \lambda_1 \dot{u} + \lambda_2 \dot{u}^2 + \lambda_3 \dot{u}^3, \quad (28)$$

where



$$\begin{cases} \lambda_0 = (1 - y^2)\dot{y} - u, \\ \lambda_1 = (1 + y^2 + \dot{y}^2), \\ \lambda_2 = 1 + \dot{y}^2, \\ \lambda_3 = 1. \end{cases} \quad (29)$$

To convert it into integration operation, the corresponding U-realisation has the form of

$$y = \frac{\lambda_0}{s^2} + \frac{\lambda_1}{s}u + \frac{\lambda_2}{s}u^2 + \frac{\lambda_3}{s}u^3. \quad (30)$$

### 3.2. Algorithms for State Space Models

3.2.1. *Linear Plant.* For a general SISO linear CT state-space system model, it has

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx + Du, \end{cases} \quad (31)$$

where  $u, y \in \mathbb{R}$ ,  $x \in \mathbb{R}^n$ , let  $A = \begin{bmatrix} 0 & 1 & 0 & \cdots \\ \cdots & \ddots & \ddots & \cdots \\ -\alpha_1 & \cdots & \cdots & -\alpha_n \end{bmatrix}$ ,  $B = [0 \ 0 \ \cdots \ 1]^T$ ,  $C = [\beta_1 \ \cdots \ \beta_n]$ , and  $D = 0$ . Expanding

(14) gives rise to

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = x_3, \\ \vdots \\ \dot{x}_n = \sum_{j=1}^n \alpha_j x_j + u, \\ y = \sum_{j=1}^n \beta_j x_j, \end{cases} \quad (32)$$

where  $x$  is the state vector and  $u$  and  $y$  are the input and output of the model, respectively.

For taking up such UM-dynamic inversion, first use a systematic approach [24] to convert the linear state space model into input/output transfer function by

$$\frac{Y(s)}{U(s)} = C[sI_n - A]^{-1}B, \quad (33)$$

where  $Y(s)$  and  $U(s)$  are the Laplace transforms of the output and input, respectively, and  $I_n$  is an identity matrix of dimension  $n$ .

Then, use the linear polynomial dynamic inversion procedure presented in Section 3.1.1.

3.2.2. *Nonlinear Plant.* For the model of (10), here is the step by step procedure for UM-dynamic inversion.

- (1) Generate direct mapping between the output  $y$  and the input  $u$ , by differentiating the state variables in the output equation till it is directly related to  $u$  in the state equation
- (2) Use the procedure for nonlinear polynomial dynamic inversion in Section 3.1.2 to determine the solutions

It should be noted that the above computations require the full state variable necessarily available/measurable.

To illustrate the realisation, consider a nonlinear state space model of

$$\begin{cases} \dot{x}_1 = x_1 - x_2 - x_3^2, \\ \dot{x}_2 = -x_1 + x_2^2 - x_3, \\ \dot{x}_3 = -x_1^2 - x_2 + x_3 + u, \\ y = x_1. \end{cases} \quad (34)$$

Differentiating  $y$  twice against  $x_1$  with the output equation gives

$$\begin{cases} \dot{y} = \dot{x}_1 = x_1 - x_2 - x_3^2, \\ \ddot{y} = \dot{x}_1 - \dot{x}_2 - 2x_3(x_1^2 - x_2 + x_3 + u). \end{cases} \quad (35)$$

As the second line of (35) directly relates the output and the input, it can be used for the dynamic inversion. The corresponding polynomial U-model is given by

$$\begin{cases} \ddot{y} = \lambda_0 + \lambda_1 u, \\ \lambda_0 = \dot{x}_1 - \dot{x}_2 - 2x_3(x_1^2 - x_2 + x_3), \\ \lambda_1 = -2x_3. \end{cases} \quad (36)$$

Convert to integral expression as

$$y = \frac{\lambda_0}{s^2} + \frac{\lambda_1}{s}u. \quad (37)$$

3.3. *Iterative Algorithms.* There are systematic routines to find roots for the 1st order (linear) and 2nd order (non-linear) polynomials. However, it is difficult analytically to determine the roots for the 3rd and up order polynomials. Commonly iterative root-solving algorithms are considered.

Newton-Raphson algorithm [25] is a classical iterative algorithm, which can be used to determine the roots of polynomial U-models, that is, the solution of UM-dynamic inversion. In formulation, this algorithm is given by

$$u \Big|_{K+1}^{(N)} = u \Big|_K^{(N)} - \frac{y \Big|_{K+1}^{(M)} - \sum_{j=0}^J \lambda_j(Y_{M-1}, U_{M-1}, \Theta) \left( \begin{matrix} (N) \\ u \end{matrix} \right)^j}{d \left( \begin{matrix} (M) \\ y \end{matrix} \right) - \sum_{j=0}^J \lambda_j(Y_{M-1}, U_{M-1}, \Theta) \left( \begin{matrix} (N) \\ u \end{matrix} \right)^j} / d \left( \begin{matrix} (N) \\ u \end{matrix} \right). \quad (38)$$

In addition, to test U-control systems in Simulink/Matlab simulation, Matlab functions can be used to find the roots directly.

#### 4. Simulation Demonstrations

This simulation demonstration selected four plant models to tests the UM-dynamic inversion and their associated U-control systems with the following bullet points:

- (i) To demonstrate the generality and effectiveness of UM dynamic inversion
- (ii) To demonstrate the principle of model-independent design in U-control, supported by the dynamic inversions
- (iii) To demonstrate a once-off design with the linear invariant controller in accordance with a closed-loop performance specification irrespective of the plant model structures
- (iv) To validate the applicability, conciseness, and efficiency of the U-control and UM-dynamic inversion, particularly in designing nonlinear control systems

*4.1. U-Control System Design.* With reference to the previous introduction in Section 2, for these simulations, design a unique U-control system with a desired system output response in terms of damping ratio  $\zeta = 0.7$ , undamped natural frequency  $\omega_n = 1$ , and zero steady state error to a step reference input [15]. Accordingly, this closed-loop transfer function was specified as

$$\frac{Y(S)}{R(S)} = G(S) = \frac{1}{s^2 + 1.4s + 1}. \quad (39)$$

The invariant controller  $G_{c1}$  was determined by taking the inverse of (39):

$$G_{c1} = \frac{G}{1 - G} = \frac{1}{s^2 + 1.4s}. \quad (40)$$

#### 4.2. Case 1: Linear Polynomial and State Space Models

##### 4.2.1. Plant 1: Polynomial Model.

$$G_P(s) = \frac{3s^2 + 6s + 4}{s^2 + 2s + 1}. \quad (41)$$

The corresponding U-model was

$$\begin{cases} y = \frac{\lambda_0}{s^2} + \lambda_1 u, \\ \lambda_0 = -2\dot{y} - y + 6\dot{u} + 4u, \\ \lambda_1 = 3. \end{cases} \quad (42)$$

##### 4.2.2. Plant 2: State Space Model.

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1.5 \\ -0.5 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \\ y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]. \end{cases} \quad (43)$$

The corresponding U-model was

$$\begin{cases} \ddot{y} = \lambda_0 + \lambda_1 u, \\ \lambda_0 = 1.5(-0.5x_1 - 2x_2), \\ \lambda_1 = 1.5. \end{cases} \quad (44)$$

#### 4.3. Case 2: Nonlinear Polynomial and State Space Models

##### 4.3.1. Plant 3: Polynomial Model.

$$\dot{y} = \dot{u}^3 + \dot{u}^2 - \dot{u} - 0.5y + \sin(u). \quad (45)$$

The corresponding U-model was

$$\begin{cases} \dot{y} = \lambda_0 + \lambda_1 \dot{u} + \lambda_2 \dot{u}^2 + \lambda_3 \dot{u}^3, \\ \lambda_0 = -0.5y + \sin(u), \\ \lambda_1 = -1, \\ \lambda_2 = 1, \\ \lambda_3 = 1. \end{cases} \quad (46)$$

##### 4.3.2. Plant 4: State Space Model.

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \sin(x_1) + u, \\ y = x_1. \end{cases} \quad (47)$$

The corresponding U-model model was

$$\begin{cases} \ddot{y} = \lambda_0 + \lambda_1 u, \\ \lambda_0 = \sin(x_1), \\ \lambda_1 = 1. \end{cases} \quad (48)$$

*4.4. Simulation.* Figures 4 and 5 show the U-control system design simulation structures. Figure 6 and 7 show the simulation results. All of these have demonstrated that the purposes outlined at the beginning of the section have been achieved.

## 5. Test of U-Control of a Wind Energy Conversion System

*5.1. Brief Review of Wind Energy Conversion Control Systems.* Wind power is a clean natural resource to supplement the other power resources from fossil fuels, coal, solar, and so on. This rich power source is widely distributed, renewable, has no greenhouse gas emissions, and uses little land [26]. In conversion of wind power, in which air flows drive wind turbines to generate electrical power, the need of effective control strategies for cost reduction and power acquisition

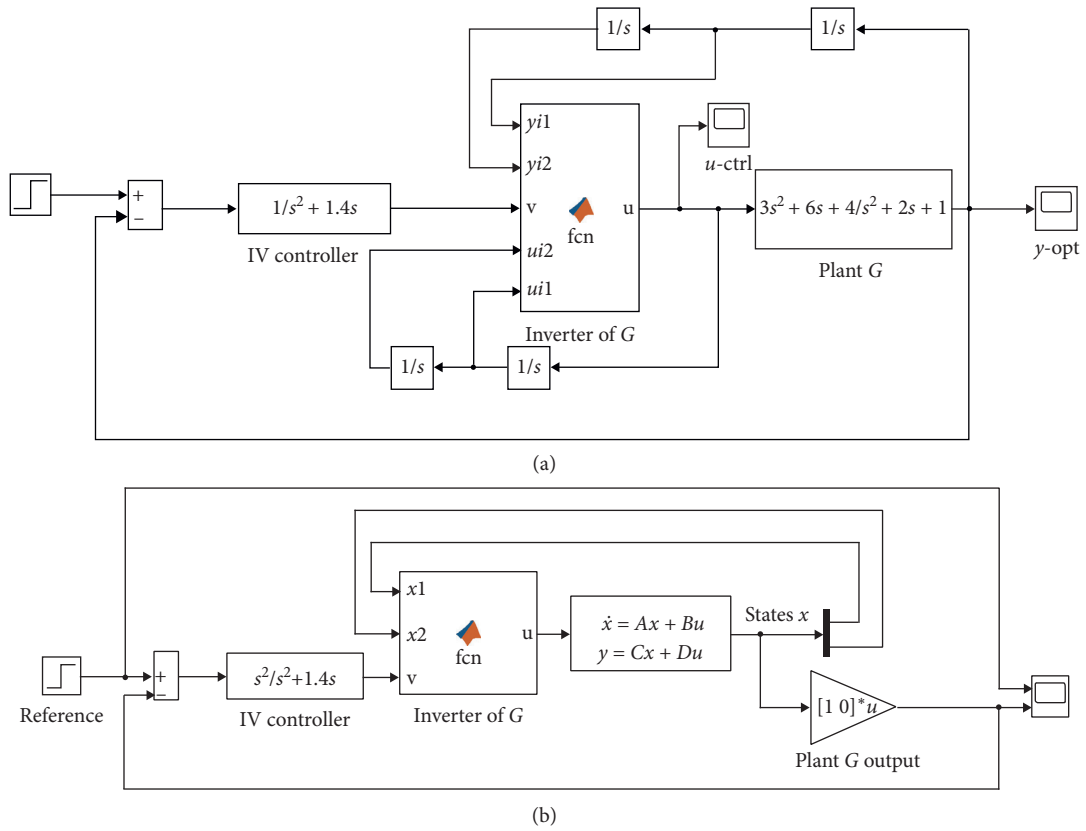


FIGURE 4: (a) U-control of plant 1. (b) U-control of plant 2.

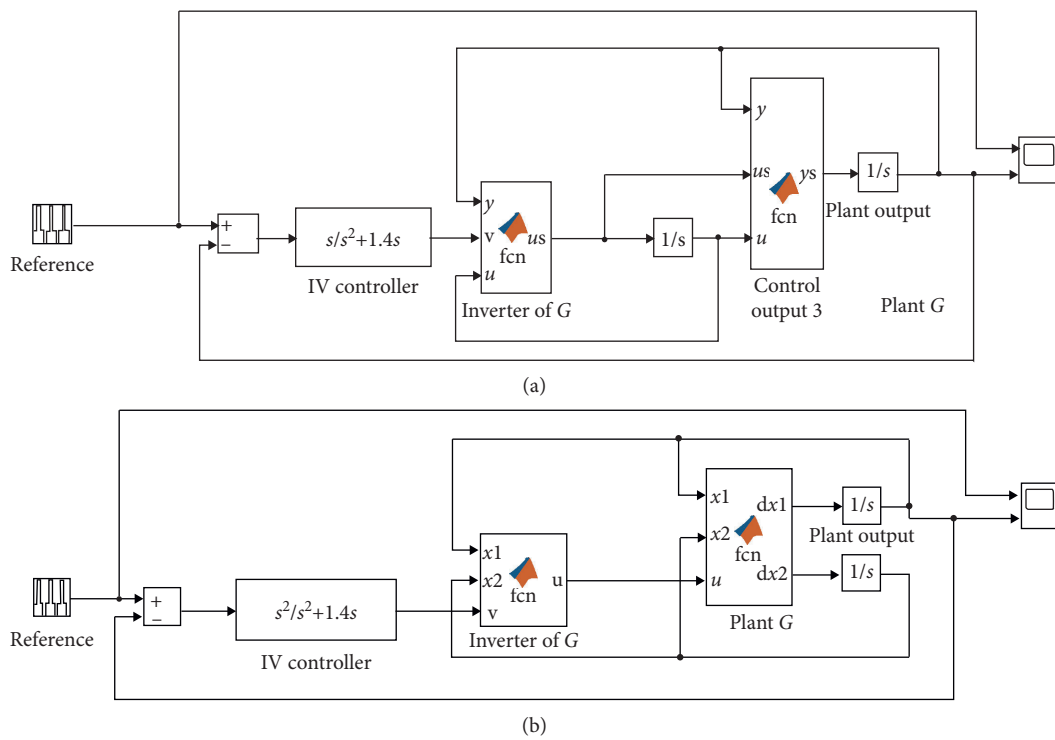


FIGURE 5: (a) U-control of plant 3. (b) U-control of plant 4.



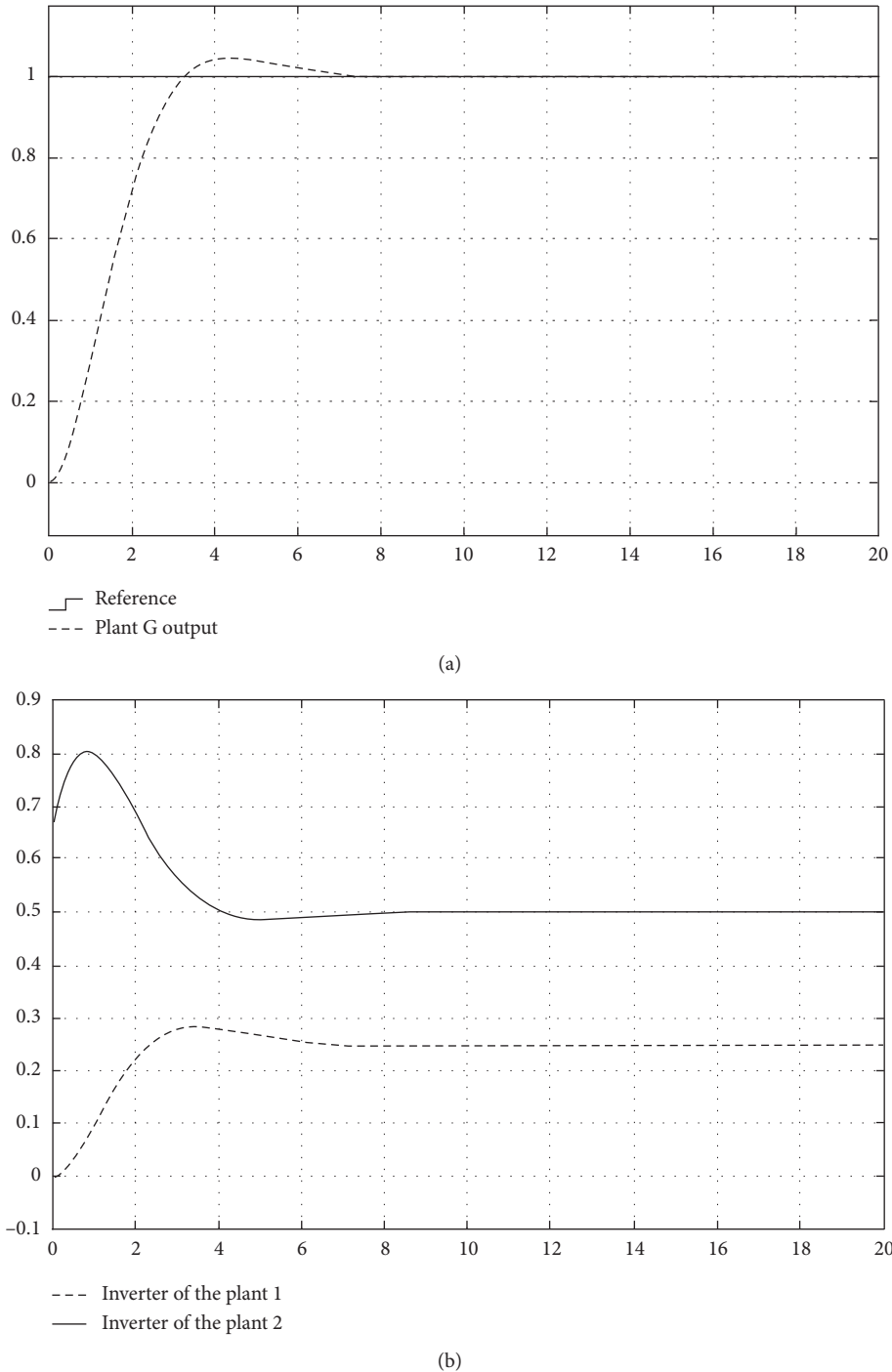


FIGURE 6: (a) Plant outputs and reference for plant1 and plant2. (b) Controller outputs for plant1 and plant2.

performance is commonly recognised. Particularly, such control system design is very critical for Variable Speed Wind Turbines (VSWT). The other unavoidable issue in wind energy conversion is that the turbine performs according to linear dynamics, but the power conversion is nonlinear from the multiplication of two dynamic variables (the wind power is obtained as a product of torque/input and rotor angular speed). Designing such control systems is

challenging in formulation and implementation. Precup et al. [27] gives a collection of the up to date research on advanced control and optimization paradigms for wind energy systems. Regarding U-control of the wind energy conversion systems, Zhu et al. [28] presents the 1st U-model-based control system formulation and design for wind energy conversion systems. In contrast, this study removes the demand on solving Diophantine equation for

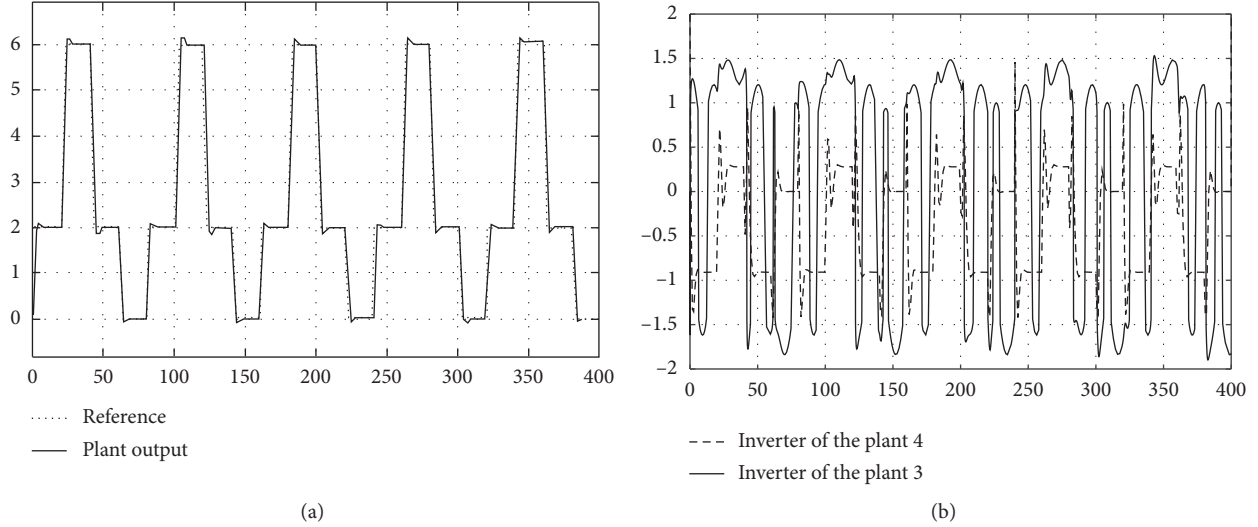


FIGURE 7: (a) Plant outputs and reference for plant 3 and plant 4. (b) Controller outputs for plant 3 and plant 4.

pole placement assignment and directly uses closed loop inversion to design the invariant controller. Furthermore, this study is a continuous time control and gives emphasis on illustration of the general platform for industrial applications using Simulink block diagram connections, rather than Matlab-coded programs.

**5.2. Plant Models [29].** Modelling of the wind turbine has played a significant role in understanding of the behaviour of the wind turbine over its region of operation because it allows for the development of comprehensive control systems that aid in optimal operation of a wind turbine. Such mathematical models are the foundation to quantify control performance of the energy systems. Furthermore, these models are essential reference for the design of the turbines and minimise generation costs leading to cost reduction in wind energy, consequently making it an economically viable alternative source of energy [30]. This section characterises these wind turbine section models into an integrated nonlinear dynamic plant operational model to describe input/output relationships.

**5.2.1. Drive Train in Lumped Mass Model (as Shown in Figure 8).** Here is the nomenclature list.

- $\omega_r$ : rotor angular speed aerodynamic torque
- $J_r$  and  $J_g$ : rotor and generator inertias, respectively.
- $K_r$  and  $K_g$ : rotor and generator external damping, respectively
- $J_t$ : integrated inertia
- $K_t$ : integrated damping
- $T_{em}$ : converted electromagnetic torque
- $T_a$ : rotor torque, from external wind power in practical systems

$T_g$ : generator torque, regulate the system operation to generate power

The rotor speed  $\omega_r$  is driven by the rotor torque  $T_a$  and the low-speed torque  $T_{ls}$ . The generator speed  $\omega_g$  is driven by the high-speed torque  $T_{hs}$  and the electromagnetic torque  $T_{em}$ . It should be noted that using the gear box can change the generator speed. The dynamics of the rotor and the generator can be described by Newton's law in forms of

$$\begin{cases} J_r \dot{\omega}_r = T_a - K_r \omega_r - T_{ls}, \\ J_g \dot{\omega}_g = T_{hs} - K_g \omega_g - T_{em}. \end{cases} \quad (49)$$

Define the gearbox ratio  $n_g$  as

$$n_g = \frac{\omega_g}{\omega_r} = \frac{T_{ls}}{T_{hs}}. \quad (50)$$

Invoking (50), the generator dynamic in (49) can be rewritten as

$$n_g^2 J_g \dot{\omega}_r = T_{ls} - n_g^2 K_g \omega_r - n_g T_{em}. \quad (51)$$

Thus, the drive train model can be described by combining (49) with (51) as

$$J_t \dot{\omega}_r = T_a - K_t \omega_r - T_g, \quad (52)$$

where

$$\begin{cases} J_t = J_r + n_g^2 J_g, \\ K_t = K_r + n_g^2 K_g, \\ T_g = n_g T_{em}. \end{cases} \quad (53)$$

**5.2.2. Power Output  $P_g$ .**  $P_g$ , the power output from the generator, is given by

$$P_g = T_g \omega_r. \quad (54)$$

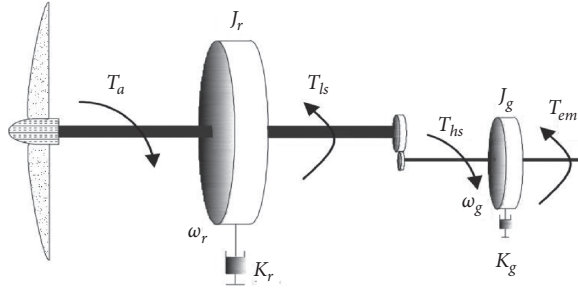


FIGURE 8: Schematic diagram of the drive train.

5.2.3. *Wind Speed Torque  $T_a$ .* In system (1), external wind is the source of the driven force. The wind speed torque is given by

$$T_a = \frac{1}{2} \rho \pi R^3 C_q(\lambda, \beta) (\hat{v} - \xi)^2, \quad (55)$$

where  $\rho$ ,  $v$ , and  $R$  are the air density, wind speed, and rotor radius, respectively,  $\beta$  is the blade pitch angle,  $\lambda = R\omega_r/v$  is the tip-speed ratio;  $\hat{v}$  is the estimation of the effective wind speed  $v$ , which can be measured via anemometer, and  $\xi$  denotes the measurement noise;  $C_q(\lambda, \beta)$ , the efficiency for the wind turbine power conversion, is given by

$$C_q(\lambda, \beta) = \frac{0.22}{\lambda} \left( \frac{116}{m} - 0.4\beta - 5 \right) \exp\left(\frac{-12.5}{m}\right), \quad (56)$$

with  $1/m = (1/(\lambda + 0.08\beta)) - (0.035/(\beta^3 + 1))$

5.2.4. *Energy Conversion Input-Output Model for Control System Design.* From the above physical principle models, the energy conversion input-output model for control system design can be expressed as

$$\dot{P}_g T_g - \dot{T}_g P_g = \frac{1}{J_t} T_a T_g^2 - \frac{K_t}{J_t} P_g T_g - \frac{1}{J_t} T_g^3, \quad (57)$$

where the control input is the generator torque  $T_g$  and the plant output is the power output  $P_g$ .

5.2.5. *Power Conversion in Low Speed Region.* For the drive train (49) to collect the maximum quantity of energy embedded in the low speed wind region, it requires

$$P_d = n_p P_{a \max}, \quad (58)$$

with

$$P_{a \max} = \frac{1}{2} \rho \pi R^2 C_{p \max} v^3, \quad (59)$$

where  $C_{p \max}$  is the maximum power coefficient and  $n_p$  is the ratio between the desired generator power  $P_d$  and the maximised available power  $P_{a \max}$ .

5.3. *U-Control System.* The U-control system was the same as designed in Section 4. The U-realisation of the input-output plant model was derived as

TABLE 1: Wind turbine characteristics.

Rated power	1.5 MW
Rotor radius	$R = 38.5$ m
Rotor inertia	$J_r = 4456761$ kg·m <sup>2</sup>
Generator inertia	$J_g = 123$ kg·m <sup>2</sup>
Rotor friction coefficient	$K_r = 45.52$ N·m/rad/s
Generator friction coefficient	$K_g = 0.4$ N·m/rad/s
Gearbox ratio	$n_g = 104.494$

$$\dot{P}_g = \lambda_0 + \lambda_1 \dot{T}_g,$$

$$\lambda_0 = \frac{(1/J_t)T_a T_g^2 - (K_t/J_t)P_g T_g - (1/J_t)T_g^3}{T_g}, \quad (60)$$

$$\lambda_1 = \frac{P_g}{T_g}.$$

5.4. *Simulation Results.* The selected generator, equipped with three blade, horizontal axis, and up wind variable speed wind turbine, generates 1.5 MW electrical output, made by WINDEY Co. This category of generators has been used worldwide [27]. The major parameters are listed in Table 1.

In consequence, the parameters of the turbine dynamic model are determined with

$$J_t = J_r + n_g^2 J_g = 5.7998 \times 10^6,$$

$$K_t = K_r + n_g^2 K_g = 4.4131 \times 10^3, \quad (61)$$

$$\frac{K_t}{J_t} = 7.609 \times 10^{-4}.$$

For U-model (60), its time varying parameters are assigned with

$$\begin{cases} \dot{P}_g = \lambda_0 + \lambda_1 \dot{T}_g, \\ \lambda_0 = \frac{1}{5.7998 \times 10^6} T_a T_g - 7.609 \times 10^{-4} P_g - \frac{1}{5.7998 \times 10^6} P_g^2, \\ \lambda_1 = \frac{P_g}{T_g}. \end{cases} \quad (62)$$

The rest of the simulation conditions/parameters include the desired power  $P_d$ , wind torque  $T_a$  from (55) and (56), the specified wind speed  $v$  with a mean of 9 m/s and turbulence intensity of 10%, sensor noise represented by a uniformly distributed random sequence of  $[-0.3, 0.3]$ ,  $n_p = 0.8$  for the ratio between the desired generator power  $P_d$  and the maximum available power  $P_{a \max}$ ,  $C_{p \max} = 0.4382$  for the maximum power ratio, and  $\rho = 1.12$  for the air density.

Figure 9 shows the constructed U-control system in Simulink block diagrams. Figure 10 shows the simulation results which are the same as those obtained from [27]. However, the differences are (1) this study is continuous time control system against the discrete time control systems

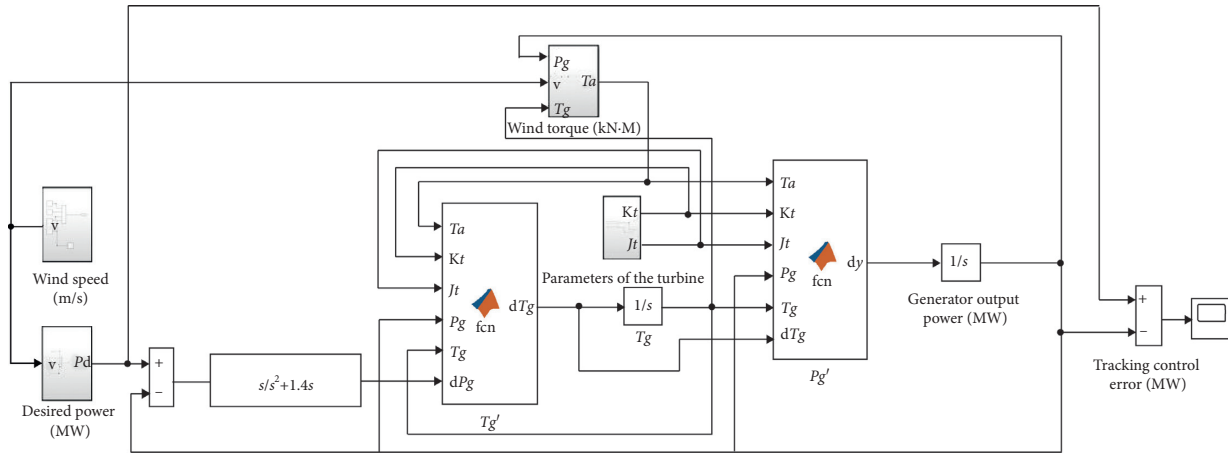


FIGURE 9: U-control of the wind energy conversion system.

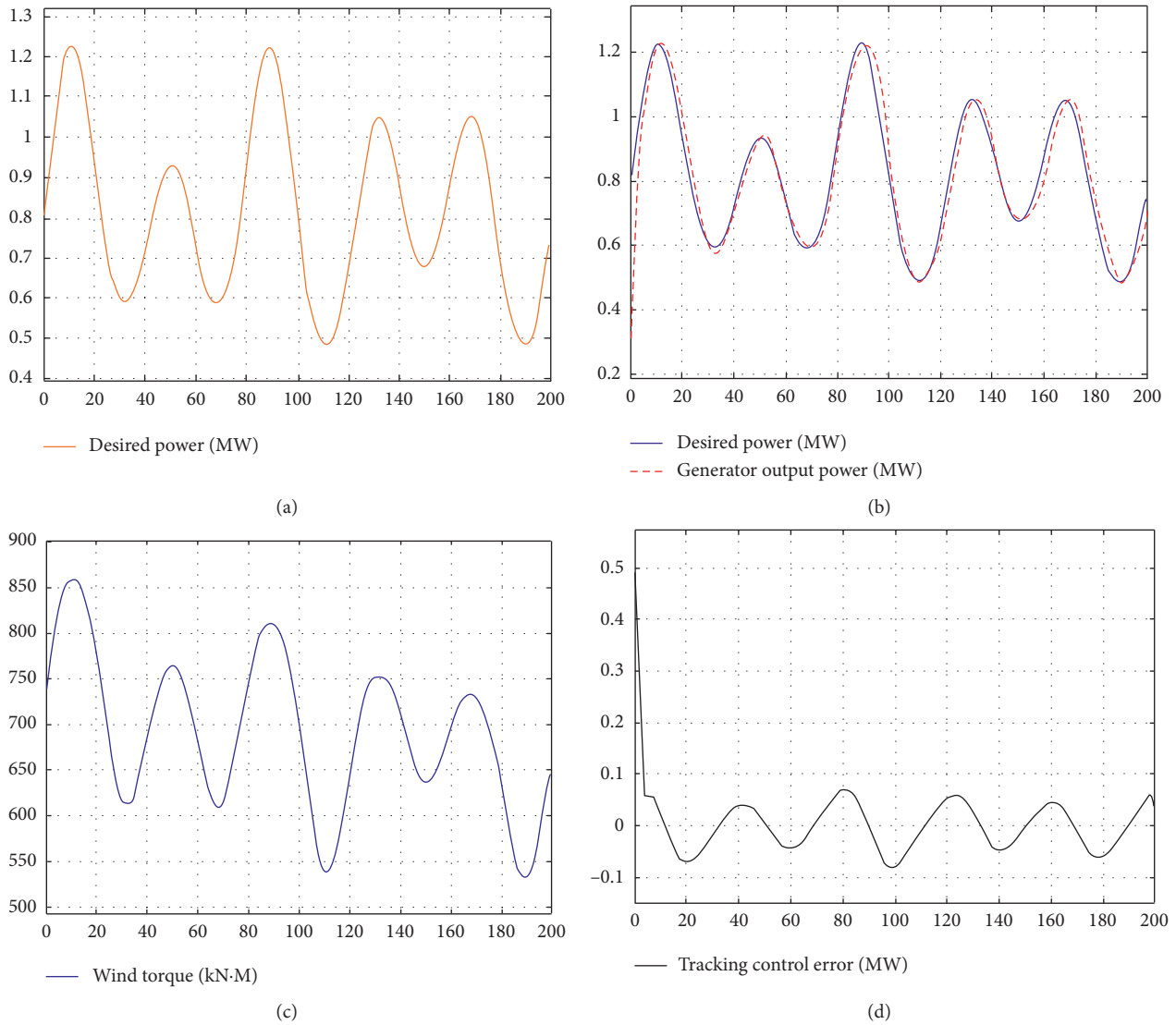


FIGURE 10: (a) Effective wind speed  $v(t)$ . (b) Desired power  $P_d(t)$  and generator output power  $P_g(t)$ . (c) Wind torque  $T_a(t)$ . (d) Tracking control error between  $P_d(t)$  and  $P_g(t)$ .

[27]; (2) this study uses concise closed loop inversion to determine the invariant controller against solving the complicated Diophantine equation; and (3) this study uses the Simulink block diagram to build up the control system against using Matlab functions to develop coded programs, which block diagram-based simulation is much more engineering meaningful, transparent, and cost-effective than coded programming. Inspection of the generated plots: Figure 10(a) shows the simulated wind speed profile, Figure 10(b) shows that the generated output power properly follows the specified power trajectory and is similar to wind speed profile even though with wind measurement noise, Figure 10(c) shows the variation of the rotor torque following the wind force, and Figure 10(d) shows the tracking error amplitude converged to zero mean and small variance, and this indicates that the U-control system has converted into the maximum available power from the wind power.

## 6. Conclusions

In U-control system design/operation, the condition of  $G_p^{-1}G_p = 1$  is the backbone. Therefore, this requires an accurate model of  $G_p$  and an effective routine/algorithm for the UM-dynamic inversion  $G_p^{-1}$ . This study provides a generalised methodology, with the assumption of an accurate model of  $G_p$ , a set of algorithms for the UM-dynamic inversion  $G_p^{-1}$ . Simulated bench examples have demonstrated the analytical results and provided an effective procedure in testing designed control systems with computational experiments.

The remaining challenging issues with UM-dynamic inversion are robust UM-dynamic inversion dealing with uncertainties in model  $G_p$  and data driven dynamic inversion (DD-dynamic inversion) for an unknown model of  $G_p$ . These solutions, no doubt will significantly make U-control realistically feasible and supplementary to the other exiting approaches for a wide range of applications.

## Data Availability

The Simulink block diagrams are included within the article to generate the plot data for supporting the findings of this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] J. M. Maciejowski, "Robustness of multivariable smith predictors," *Journal of Process Control*, vol. 4, no. 1, pp. 29–32, 1994.
- [2] S. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, Wiley, Hoboken, NJ, USA, 2013.
- [3] 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.
- [4] J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, London, UK, 1991.
- [5] A. Isidori, *Nonlinear Control Systems*, Springer-Verlag, Berlin, Germany, 3rd edition, 1995.
- [6] T. Çimen, "Systematic and effective design of nonlinear feedback controllers via the state-dependent Riccati equation (SDRE) method," *Annual Reviews in Control*, vol. 34, no. 1, pp. 32–51, 2010.
- [7] C. J. Taylor, A. Chotai, and P. C. Young, "Non-linear control by input-output state variable feedback pole assignment," *International Journal of Control*, vol. 82, no. 6, pp. 1029–1044, 2009.
- [8] P. Young, "Stochastic, dynamic modelling and signal processing: time variable and state dependent parameter estimation," in *Nonlinear and Nonstationary Signal Processing*, W. J. Fitzgerald, Ed., pp. 74–115, Cambridge University Press, Cambridge, UK, 2000.
- [9] Q. M. Zhu, K. Warwick, and J. L. Douce, "Adaptive general predictive controller for nonlinear systems," *IEE Proceedings D Control Theory and Applications*, vol. 138, no. 1, pp. 33–40, 1991.
- [10] 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.
- [11] W. Du, X. Wu, and Q. 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.
- [12] S. Muhammad and M. Haseebiddon, "U-model-based internal model control for non-linear dynamic plants," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 219, no. 6, pp. 449–458, 2005.
- [13] Q. M. Zhu, *Identification and control of nonlinear systems*, Ph.D. thesis, University of Warwick, Coventry, UK, 1989.
- [14] 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.
- [15] 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.
- [16] Q. M. Zhu, L. Liu, W. C. Zhang, and S. Y. Li, "Control of complex nonlinear dynamic rational systems," *Complexity*, vol. 2018, Article ID 8953035, 12 pages, 2018.
- [17] 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.
- [18] N. A. A. Hussain, S. S. A. Ali, M. Ovinis, M. Arshad, and U. M. Al-saggaf, "Underactuated coupled nonlinear adaptive control synthesis using U-model for multivariable unmanned marine robotics," *IEEE Access*, vol. 8, pp. 1851–1965, 2020.
- [19] Q. M. Zhu, W. C. Zhang, J. Na, and B. Sun, "U-model based control design framework for continuous-time systems," in *Proceedings of the Chinese Control Conference (CCC)*, pp. 106–111, IEEE Explore, Guangzhou, China, July 2019.
- [20] S. B. Wang, J. Na, and Y. Xing, "Adaptive optimal parameter estimation and control of servo mechanisms: theory and experiments," *IEEE Transactions on Industrial Electronics*, 2019.

- [21] S. B. Wang, L. Tao, Q. Chen, J. Na, and X. M. Ren, "USDE-based sliding mode control for servo mechanisms with unknown system dynamics," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 1066–1056, 2020.
- [22] S. Sieberling, Q. P. Chu, and J. A. Mulder, "Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1732–1742, 2010.
- [23] A. W. Manyonge, R. M. Ochieng, F. N. Onyango, and J. M. Shichikha, "Mathematical modelling of wind turbine in a wind energy conversion system: power coefficient analysis," *Applied Mathematical Sciences*, vol. 6, no. 91, pp. 4527–4536, 2012.
- [24] K. Ogata, *Modern Control Engineering*, Prentice-Hall, Boston, MA, USA, 5th edition, 2010.
- [25] C. F. Gerald, *Applied Numerical Analysis*, Pearson Education India, Bengaluru, India, 2004.
- [26] V. Fthenakis and H. C. Kim, "Land use and electricity generation: a life-cycle analysis," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 6-7, pp. 1465–1474, 2009.
- [27] R. E. Precup, T. Kamal, and H. S. Zulqadar, "Advanced control and optimization paradigms for wind energy systems," in *Power Systems*, Springer, Berlin, Germany, 2019.
- [28] Q. M. Zhu, J. Na, and S. Ghauri, "U-model based control system formulation and design for wind energy conversion systems," in *Proceedings of the 2016 UKACC 11th International Conference on Control (CONTROL)*, Belfast, UK, August 2016.
- [29] W. C. Meng, Q. M. Yang, Y. Ying, Y. Sun, and Y. X. Sun, "Adaptive power acquisition control of variable speed wind energy conversion systems under inaccurate wind speed measurement," in *Proceedings of the American Control Conference*, pp. 4271–4276, Washington, DC, USA, June 2013.
- [30] E. J. J. Smeur, Q. Chu, and G. C. H. E. de Croon, "Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 450–461, 2016.