Mechatronics 52 (2018) 135–149

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Robust predictive tracking control for a class of nonlinear systems

T.Q. Dinh[a,*], J. Marco[a], J.I. Yoon[b], K.K. Ahn[c]

[a] Warwick Manufacturing Group (WMG), University of Warwick, Coventry CV4 7AL, UK
[b] Korea Construction Equipment Technology Institute, Jeollabuk-do Gunsan 573-540 Sandan-ro 36, Republic of Korea
[c] School of Mechanical Engineering, University of Ulsan, Namgu Muger2dong, Ulsan 680-749, Republic of Korea

ARTICLE INFO

Keywords:
Fuzzy
PID
Neural network
Grey predictor
Fuzzy cognitive map
Nonlinear system

ABSTRACT

A robust predictive tracking control (RPTC) approach is developed in this paper to deal with a class of nonlinear SISO systems. To improve the control performance, the RPTC architecture mainly consists of a robust fuzzy PID (RFPID)-based control module and a robust PI grey model (RPIGM)-based prediction module. The RFPID functions as the main control unit to drive the system to desired goals. The control gains are online optimized by neural network-based fuzzy tuners. Meanwhile using grey and neural network theories, the RPIGM is designed with two tasks: to forecast the future system output which is fed to the RFPID to optimize the controller parameters ahead of time; and to estimate the impacts of noises and disturbances on the system performance in order to create properly a compensating control signal. Furthermore, a fuzzy grey cognitive map (FGCM)-based decision tool is built to regulate the RPIGM prediction step size to maximize the control efforts. Convergences of both the predictor and controller are theoretically guaranteed by Lyapunov stability conditions. The effectiveness of the proposed RPTC approach has been proved through real-time experiments on a nonlinear SISO system.

1. Introduction

Nowadays, automation in control has been applied more and more in the modern life. However, most of industrial machines are nonlinear systems with large uncertainties which cause challenges to design the controllers. Conventional PID controllers are commonly used in industry due to their simplicity, clear functionality and ease of implementation. However, this type of controllers may not perform well for nonlinear, complex and vague systems with uncertainties. And it has been found that fuzzy-logic-based PID controllers is one of potential solutions with better capabilities of handling the aforesaid systems [2–17].

Although fuzzy logic has a reputation of handling complicated control problems, typical fuzzy designs depend largely on experiences of experts [1–5]. Hence, these controllers cannot adapt for highly uncertain systems working in environments with large perturbations [9,12]. There is no systematic method to design and examine the number of rules, as well as input space partitions and membership functions (MFs). As a result, other control techniques, such as robust control, intelligent theory and estimation methods [6–14], are needed to combine with the fuzzy PID to overcome this weakness. Nevertheless, most of the traditional control strategies adopted the previous state information as the input signal of the controllers to make the decisions. Subsequently, this type of control reflects only the current status and lacks adaptability.

As a recent trend to overcome this drawback, fuzzy PID combined with prediction theories could produce in advance the control action for the following step according to the predicted value of control error before it occurs [15,16]. And the combination with neural technique and grey prediction is a feasible solution. Neural network is a universal algorithm which is able to approximate almost nonlinear functions [17–24] while the grey theory [25] is distinguished by its ability to deal with systems that have partially unknown parameters [15,16,26–35]. However, there are the shortcomings of the typical grey models such as grey sequence conditions and background series calculation which limit their applicability as well as prediction accuracy [34,35]. Additionally, there is no constraint to guarantee the prediction stability of these developed models.

The aim of this paper is to develop a robust predictive tracking control (RPTC) approach to improve performances of SISO systems with large nonlinearities and uncertainties. The RPTC architecture mainly consists of two modules: robust PI grey model (RPIGM) -based prediction module and robust fuzzy PID (RFPID) -based control module with the following contributions:

(1) To deal with any signal with random distribution, the RPIGM is newly developed using a closed-loop control form in which the robust prediction performance is ensured by a PI-based neural

* Corresponding author.
E-mail address: q.dinh@warwick.ac.uk (T.Q. Dinh).

https://doi.org/10.1016/j.mechatronics.2018.04.010
Received 11 February 2017; Received in revised form 11 January 2018; Accepted 25 April 2018
Available online 29 May 2018

network controller.
(2) Outputs from the RPIGM module are fed to the RFPID control module to optimize its parameters and, used to compensate for the impacts of noises and disturbances on the overall system response.
(3) The RFPID of which the control gains are regulated by fuzzy tuners is designed to drive the system to a desired goal. Based on the RPIGM outputs, the control parameters are optimized in advance by a neural network-based learning mechanism.
(4) A fuzzy grey cognitive map (FGCM) –based decision tool is built and integrated to the RPIGM to regulate online the RPIGM prediction step size in order to maximize the control capability.
(5) The robust performances of both the RFPID and RPIGM are guaranteed by the Lyapunov stability conditions.

As the result, the overall control performance with high accuracy, fast response and stability can be achieved. This paper is organized as: Section II shows the system description and the RPTC architecture. Section III presents the design of the RFPID control module while the design of the RPIGM prediction module is described in Section IV. Illustrative examples via real-time experiments are provided and discussed in Section V to verify the effectiveness of the proposed control methodology. Finally, concluding remarks are given in Section VI.

2. System description and RPTC design architecture

Without loss of generality, the RPTC control scheme is designed for an uncertain nonlinear system (*P*) with single-input-single-output (SISO) [12] as in Fig. 1. The proposed RPTC architecture with the two modules, RFPID and RPIGM, is employed to drive the system to follow a given reference (*R*) (the system response $y(t) \equiv y_t$ needs to reach the desired level $y_r(t) \equiv y_{rt}$).

At step $(t+1)^{th}$ with the tracking error, $e(t) = y_r(t) – y(t)$, the RFPID generates a proper control action based on the PID algorithm, $u_{RFPID}(t+1) \equiv u_m(t+1) \equiv u_{m(t+1)}$. Meanwhile using the information of $y_r(t)$ and $y(t)$, the RPIGM estimates the system actuation *p*-step ahead of time, $\hat{y}(t+p) \equiv \hat{y}_{t+p}$. This estimated response is then employed with the *p*-step ahead desired set point, $y_r(t+p)$, to optimize robustly the RFPID parameters. Moreover, the RPIGM produces an additive control correction, $u_{RPIGP}(t+1) \equiv u_c(t+1) \equiv u_{c(t+1)}$, which is added to the main control signal $u_{m(t+1)}$, to compensate for system noises (*N*) and disturbances (*D*). Therefore, the system control input generated by the RPTC scheme is computed as

$$u_{RPTC}(t+1) \equiv u(t+1) = u_m(t+1) + u_c(t+1) \tag{1}$$

$$u_m(t+1) = K_P(t+1)e(t) + K_I(t+1)\int_0^t e(t)dt + K_D(t+1)\frac{de(t)}{dt} \tag{2}$$

$$u_c(t+1) = K_c \times \hat{e}_{ND}^{(0)}(t+1) \tag{3}$$

where: $e(t)$ is the control error; $de(t)$ is the derivation of error $e(t)$; $\hat{e}_{ND}^{(0)}(t+1) = \hat{y}(t+1) - y(t+1)$ is the impact of noise and disturbance on the system response, $\hat{y}(t+1)$ is estimated by the RPIGM; $K_P(t+1)$, $K_I(t+1)$, and $K_D(t+1)$ are the dynamic proportional, integral, and derivative gains of the PID algorithm, respectively, regulated by fuzzy inferences; K_c is the fixed conversion factor.

The detailed designs of the RFPID and RPIGM modules are introduced in the following sections.

3. Robust fuzzy PID-based control module

Structure of the RFPID control module (shown in Fig. 1) is described in Fig. 2a. This module includes two main blocks: a fuzzy PID mechanism, which is the combination of the PID algorithm and three fuzzy tuners to regulate the PID gains via a robust updating rule (RUR), to produce the control output, and a robust learning mechanism (RLM) to optimize parameters of the fuzzy tuners.

3.1. Fuzzy PID mechanism

3.1.1. Fuzzy tuners

To minimize the tracking error, the PID gains, K_P, K_I and K_D, are online regulated using the three separate fuzzy tuners: fuzzy P, fuzzy I and fuzzy D, respectively, as the following:

$$\begin{aligned} &K_A^{Fuzzy}(t+1) = K_{A0} + U_A^{Fuzzy}(t+1)\Delta K_A \\ &U_A^{Fuzzy}(t+1) \in (0, 1), \quad (A \text{ is } P,\ I \text{ or } D) \end{aligned} \tag{4}$$

where $\Delta K_A = K_{A1} - K_{A0}$ is the allowable deviation of K_A; K_{A0}, K_{A1} are the minimum, maximum values of K_A, respectively; $U_A^{Fuzzy}(t+1)$ is the bounded parameter and derived from the fuzzy tuner P or I or D. Thus, one has $K_A(t+1) \subset [K_A^{\min}, K_A^{\max}]$.

Remark 1. For all the fuzzy tuners, triangle and singleton MFs are used to represent for partitions of fuzzy inputs and outputs, respectively. Fuzzy control is applied using local inferences. That means each rule is inferred and the inferring results of individual rules are then aggregated. Here, the most common inference using max-min method, which offers a computationally nice and expressive setting for constraint propagation, is selected. Finally, a defuzzification is needed to obtain a crisp output from the aggregated fuzzy result. The centroid defuzzification, which is widely used for fuzzy control problems needing crisp outputs, is chosen to construct the fuzzy tuners.

From (2), (4) and using Remark 1, each fuzzy tuner are designed with two inputs (as the most practical fuzzy PID type [12]) and one output as depicted in Fig. 2b. For the optimisation purpose, each tuner is structured in the network form with five layers. In the layer 1, the two inputs x_1 and x_2 are the same for both the tuners and derived as normal scales or absolute scales of the control error and its derivative, which are depended on the symmetric behaviour of the system. The range for each fuzzy input is correspondingly forced into range from − 1 to 1 or from 0 to 1 by proper scaling factors (k_1 and k_2) chosen from the system specifications. These inputs are then converted into fuzzy values via the layer 2 using triangle MFs. Each MF of each input variable can be expressed in a general form as follows:

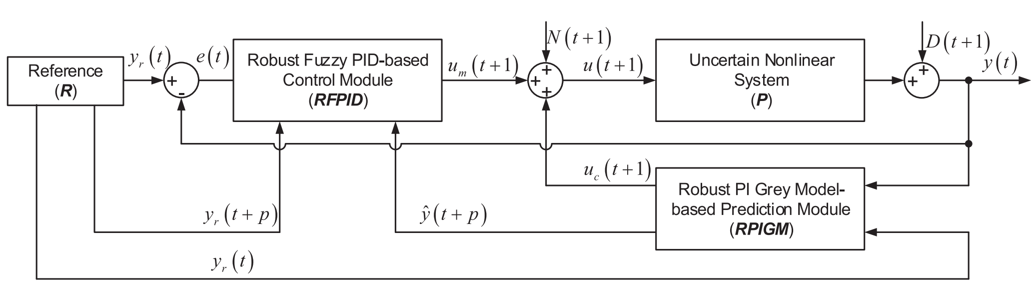

Fig. 1. Overall RPTC control architecture for a generic nonlinear system.