

Neural network based iterative learning predictive control design for mechatronic systems with isolated nonlinearity

Ridong Zhang^{a,*}, Anke Xue^a, Jianzhong Wang^a, Shuqing Wang^b, Zhengyun Ren^c

^aInformation and Control Institute, Hangzhou Dianzi University, Hangzhou 310018, PR China

^bNational Key Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Hangzhou 310027, PR China

^cDepartment of Automation, Donghua University, Shanghai 200051, PR China

Received 22 October 2007; received in revised form 28 January 2008; accepted 31 January 2008

Abstract

The paper presents a new nonlinear predictive control design for a kind of nonlinear mechatronic drive systems, which leads to the improvement of regulatory capacity for both reference input tracking and load disturbance rejection. The nonlinear system is first treated into an equal linear time-variant system plus a nonlinear part using a neural network, then an iterative learning linear predictive controller is developed with a similar structure of PI optimal regulator and with setpoint feed forward control. Because the overall control law is a linear one, this design gives a direct and also effective multi-step prediction method and avoids the complicated nonlinear optimization. The control law is also an accurate one compared with traditional linearized method. Besides, changes of the system state variables are considered in the objective function with control performance superior to conventional state space predictive control designs which only consider the predicted output errors. The proposed method is compared with conventional state space predictive control method and classical PI optimal control method. Tracking performance, robustness and disturbance rejection are enlightened.

© 2008 Elsevier Ltd. All rights reserved.

Keywords: Neural network; Iterative learning; Predictive control; Mechatronic drive systems

1. Introduction

Mechatronics is the synergetic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of industrial products and processes [1–3]. In many cases, the mechanical part of the system is coupled with the electrical, thermodynamical, chemical or information processing part. Therefore, mechatronic systems are actually coupled processes exhibiting nonlinear characteristics.

Different control methods, e.g., for the control of position, speed or force for various mechatronic systems are provided [5–8] based on different control design theories [4]. The methods are generally designed to compensate for the system's nonlinearities. However, they are rather com-

putationally demanding and require high processing capability of CPUs and the control performance greatly relies on the accuracy of the compensators [4]. Typical examples focusing on compensating for nonlinearities are: (1) Compensation of nonlinear static characteristics [5], however, this method needs to design an inverse function compensator and suppose that the nonlinear function has an inverse function. (2) Friction compensation [8], however, the control performance greatly depends on the accuracy of the linearization and generally this linearization is not an accurate one. Therefore, the development of effective and reliable control methods with relatively simple structure is a challenge for the designers.

The paper is to focus on the mechatronic drive systems [8], which are typically controlled by conventional state space predictive control (CSSPC) [8] and classical PI control (PI) [10] and recently studied by Rau et al. [8]. The paper proposes a nonlinear iterative learning predictive

* Corresponding author. Tel.: +86 0571 86878566.

E-mail address: zrd-el@163.com (R. Zhang).

control method (NILPC), which provides improvement of regulatory capacity for both reference input tracking and load disturbance rejection compared with CSSPC and PI. General description of model predictive control (MPC) can be found in [9].

Generally, the procedure consists of two steps: (i) unlike traditional methods [8], the nonlinear part of the process is first transformed into a linear time-variant part and a nonlinear part, and the linear part of the system is used to design an overall convergent linear predictive control law (ii) secondly, different from CSSPC [8], NILPC is designed for the obtained equal linear time-variant system plus a nonlinear part.

2. Process description and its presentation

2.1. Process description

Single-input single-output (SISO) mechatronic systems will be considered throughout the paper. An extension to multi-input multi-output systems is possible without any complexity. The system under consideration is described by the following nonlinear state space model [8]

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= \bar{\mathbf{A}}\bar{\mathbf{x}}(k) + \bar{\mathbf{b}}u(k-d) + \mathbf{k} \cdot NL(y(k)) \\ y(k) &= \bar{\mathbf{c}}^T \bar{\mathbf{x}}(k) + hu(k-d) \end{aligned} \quad (1)$$

where $\bar{\mathbf{x}}(k) = [\bar{x}_1(k), \bar{x}_2(k), \dots, \bar{x}_n(k)]^T$ is the state vector with dimension $n \times 1$, $\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{\mathbf{c}}^T, h, \mathbf{k}$ are matrices or constants of appropriate dimensions, \mathbf{k} describes the coupling of the nonlinearity into the system. NL is a nonlinear function.

2.2. Equalization of the nonlinear process

The nonlinear part $NL(y(k))$ is first described by a three-layer BP neural network as follows:

$$N\hat{L}(y(k)) = NN(y(k)) = g \left\{ \sum_{i=1}^I w_3(i)g[w_2(i)y(k)] \right\} \quad (2)$$

where NN is a BP neural network, $w_2(i), w_3(i)$ are the linking weights between layers, I is the number of hidden nodes, g is the activation function:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Consider the nonlinear function NN , let $\gamma(k)$ be the center of it, through Taylor expansion:

$$\begin{aligned} NN(y(k)) &= NN(\gamma(k)) + \left. \frac{dNN}{dy(k)} \right|_{y(k)=\gamma(k)} \cdot [y(k) - \gamma(k)] \\ &\quad + \varepsilon[y(k) - \gamma(k)] \\ &= a(k)\bar{\mathbf{c}}^T \bar{\mathbf{x}}(k) + a(k)hu(k-d) \\ &\quad + NN(\gamma(k)) - a(k)\gamma(k) + \varepsilon[y(k) - \gamma(k)] \\ &= a(k)\bar{\mathbf{c}}^T \bar{\mathbf{x}}(k) + a(k)hu(k-d) + NN1(y(k)) + C \end{aligned} \quad (4)$$

where

$$a(k) = \left. \frac{dNN}{dy(k)} \right|_{y(k)=\gamma(k)},$$

ε is a nonlinear function, $NN1(y(k)) = \varepsilon[y(k) - \gamma(k)]$, $C = NN(\gamma(k)) - a(k)\gamma(k)$, and it is a constant. Substituting Eq. (4) into Eq. (1), a new model is generated:

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= \bar{\mathbf{A}}_k \bar{\mathbf{x}}(k) + \bar{\mathbf{b}}_k u(k-d) + \mathbf{k} NN1(y(k)) + \mathbf{k} C \\ y(k) &= \bar{\mathbf{c}}^T \bar{\mathbf{x}}(k) + hu(k-d) \end{aligned} \quad (5)$$

with

$$\begin{aligned} \bar{\mathbf{A}}_k &= \bar{\mathbf{A}} + \mathbf{k} a(k) \bar{\mathbf{c}}^T \\ \bar{\mathbf{b}}_k &= \bar{\mathbf{b}} + \mathbf{k} a(k) h \end{aligned} \quad (6)$$

The treatment of the process this way results in a linear part plus a nonlinear part. Note that Eq. (5) is different from Eq. (1) in that its linear part contains more information of the original nonlinear process than that of Eq. (1). The aim of the paper is to use the linear part to design an overall convergent nonlinear predictive controller. Thus it is more suitable to use Eq. (5) than Eq. (1). The next step is to design a linear predictive control method regardless of the form of the nonlinear part. By doing so, the nonlinear function observer [8] is not needed any longer when designing the controller and the structure of the control system is simpler because of omitting the nonlinear observer.

2.3. Further treatment of the derived model

Take $u(k-1), \dots, u(k-d)$ as the system's state variables and $u(k)$ as the input only

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_k \mathbf{x}(k) + \mathbf{b}u(k) + \mathbf{k}' NN1(y(k)) + \mathbf{k}' C \\ y(k) &= \mathbf{c} \mathbf{x}(k) \end{aligned} \quad (7a)$$

where $\mathbf{x}(k) = [\bar{\mathbf{x}}(k)^T, u(k-d), u(k-d+1), \dots, u(k-1)]^T$ is the new state variable,

$$\mathbf{A}_k = \begin{bmatrix} \bar{\mathbf{A}}_k & \bar{\mathbf{b}}_k & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \bar{\mathbf{0}} & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ \bar{\mathbf{0}} & 0 & & 0 & & 1 \\ \bar{\mathbf{0}} & 0 & \dots & & & 0 \end{bmatrix}$$

$$\mathbf{b} = [\mathbf{0}^T, 0, L, 0, 1]^T \quad \mathbf{c} = [\bar{\mathbf{c}}^T, h, 0, L, 0] \quad \mathbf{k}' = [\mathbf{k}^T, 0, 0, L, 0]^T \quad (7b)$$

$\bar{\mathbf{0}}$ and $\mathbf{0}$ are matrices with appropriate dimensions. Add the back shift operator to Eq. (7) and note that C is a constant, so $\Delta C = 0$, then

$$\Delta \mathbf{x}(k+1) = \mathbf{A}_k \Delta \mathbf{x}(k) + \mathbf{b} \Delta u(k) + \mathbf{k}' \Delta NN1(y(k)) \quad (8)$$

Define the expected output as $r(k)$, and the output tracking error is

$$e(k) = y(k) - r(k) \quad (9)$$

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات