A New Model-Free Predictive Control Method Using Input and Output Data

A New Model-Free Predictive Control Method using Input and Output Data

Shigeru Yamamoto[1, a]

[11]Faculty of Electrical and Computer Engineering, Kanazawa University,

Kakuma, Kanazawa, Ishikawa, 920-1192 Japan

[a]shigeru@se.kanazawa-u.ac.jp

Keywords: Predictive Control, Adaptive Control, Data-driven approach.

Abstract. The purpose of this paper is to present a new predictive control utilizing online data and stored data of input/output of the controlled system. The conventional predictive control methods utilize the mathematical model of the control system to predict an optimal future input to control the system. The model is usually obtained by a standard system identification method from the measured input/output data. The proposed method does not require the mathematical model to predict the optimal future control input to achieve the desired output. This control strategy, called just-in-time, was originally proposed by Inoue and Yamamoto in 2004. In this paper, we proposed a simplified version of the original just-in-time predictive control method.

Introduction

Predictive control, usually referred to as model predictive control, is wildly used in industrial (especially chemical) systems nowadays [1]. This model predictive control postulates to use a mathematical model of the controlled system to predict future behavior of the system. The mathematical model is required to appropriately represent the dynamics of the controlled system. To obtain the mathematical model, many system identification methods are available. Normally, the mathematical model is not updated until a great change occurs in the controlled system.

On the other hand, adaptive control constantly update the mathematical model and/or control parameters based on the online measured input and output data [2]. To combine adaptive control and predictive control, the so-called just-in-time model predictive control was proposed by Stenman in 1999 [3]. This method includes a mathematical model which is obtained based on a Just-In-Time modeling [4,5] (also referred to as model-on-demand [6], lazzy learning [7], instance based learning [8]) utilizing both the online measured input and output data and stored past measured input and output data. In the method, the mathematical model is constantly maintained to use online data and to refine the model the stored past measured input and output data are also used. The just-in-time method has been originally developed for nonlinear system modelling which adaptively identifies a local model (not global) around the current operating point to use a large amount of past stored data.

Instead of the just-in-time "model" predictive control, Ota and Yamamoto [9,10] proposed a "model-free" predictive control method in the just-in-time modelling framework. In the method, an optimal control input is directly predicted not to use the local model but online current measured data and stored past. As in the just-in-time modelling method, the neighbors of the current data are searched in the stored data and the predicted control input is derived as weighted average of the neighbors. For this weighted average, several methods are considered in the just-in-time modelling framework. In this paper, we propose a new simplified weighted average.

Notations. Let $\boldsymbol{R}$, $\boldsymbol{R}^n$, $\boldsymbol{R}^{n\times m}$ be the set of real numbers, real valued vector with n elements, real valued matrices with n rows and m columns, respectively. Let $x^T$and A^T be the transpose of $x \in \boldsymbol{R}^n$ and $A \in \boldsymbol{R}^{n\times m}$, respectively. When a square matrix $A \in \boldsymbol{R}^{n\times n}$ is nonsingular, we denote its transpose as A^{-1}.

Model-Free Predictive Control

Consider a nonlinear auto regressive model with exogeneous (NARX) model

$$y(t+1)=f(x(t))+\varepsilon(t)\ ,\ t=0,\ 1,2,\dots. \tag{1}$$

$$x(t)=[y(t),\dots,y(t-n+1),\ u(t),\dots,\ u(t-m+1)] \tag{2}$$

where t is the discrete-time, $u\in\mathbf{R}$ is the input, $y\in\mathbf{R}$ is the output, $x\in\mathbf{R}^p$ is the regression vector of the size $p=n+m$, f is an unknown nonlinear map, and ε is i.i.d. random noise, respectively. We assume that n and m are unknown but we can determine them with uncertainty.

When we are given (1), the goal of control is to make the h steps ahead future output $y(t+1), y(t+2), \dots y(t+h)$ follow the desired future (given) reference $r(t+1)$, $r(t+2)$, ..., $r(t+h)$. To reach the goal, an appropriate future input sequence $u(t), u(t+1), \dots, u(t+h-1)$ should be applied. To determine the future input sequence, we assume that we can use all past input and output stored in a memory as a form of vectors

$$a_i = \begin{pmatrix} \boldsymbol{y}_\mathrm{p}(t_i) \\ \boldsymbol{y}_\mathrm{f}(t_i) \\ \boldsymbol{u}_\mathrm{p}(t_i) \end{pmatrix},\ i = 1,2,\dots,N, \tag{3}$$

$$c_i = \boldsymbol{u}_\mathrm{f}(t_i), \qquad i = 1,2,\dots,N, \tag{4}$$

where

$$\boldsymbol{y}_\mathrm{p}(t) = \begin{pmatrix} y(t-n+1) \\ \vdots \\ y(t) \end{pmatrix}, \boldsymbol{y}_\mathrm{f}(t) = \begin{pmatrix} y(t+1) \\ \vdots \\ y(t+h) \end{pmatrix}, \tag{5}$$

$$\boldsymbol{u}_\mathrm{p}(t) = \begin{pmatrix} u(t-m) \\ \vdots \\ u(t-1) \end{pmatrix}, \boldsymbol{u}_\mathrm{f}(t) = \begin{pmatrix} u(t) \\ \vdots \\ u(t+h-1) \end{pmatrix}. \tag{6}$$

In model-free predictive control [9,10], to use the just-in-time algorithm [4,5], a set of k data a_i is searched which are similar to the current (called a query) vector

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{y}_\mathrm{p}(t) \\ \boldsymbol{r}(t) \\ \boldsymbol{u}_\mathrm{p}(t) \end{pmatrix} \tag{7}$$

where

$$\boldsymbol{r}(t) = \begin{pmatrix} r(t+1) \\ \vdots \\ r(t+h) \end{pmatrix}. \tag{8}$$

The searched k data a_i form k-nearest neighbors of $\boldsymbol{b}$ together with c_i which have the same index i with a_i as

$$\boldsymbol{\Omega}(\boldsymbol{b}) = \{(a_{i_k}, c_{i_k}) | j = 1,\dots,k\}, \tag{9}$$

where we assume that all vectors are sorted by the distance to $\boldsymbol{b}$ as

$$\|a_{i_1} - \boldsymbol{b}\| \le \|a_{i_2} - \boldsymbol{b}\| \le \cdots \le \|a_{i_k} - \boldsymbol{b}\|. \tag{10}$$

In model-free predictive control [9,10], from the k-nearest neighbors $\boldsymbol{\Omega}(\boldsymbol{b})$, weights w_{i_j} are determined such that

$$w_{i_1} \ge w_{i_2} \ge \cdots \ge w_{i_k} > 0 \ \text{ and } \ w_{i_1} + w_{i_2} + \cdots + w_{i_k} = 1. \tag{11}$$

Finally, the expected future input sequence are calculated as

$$\hat{\boldsymbol{u}}_{\mathrm{f}}(t) = \begin{pmatrix} \hat{u}(t) \\ \vdots \\ \hat{u}(t+h-1) \end{pmatrix} = [c_{i_1} \quad \cdots \quad c_{i_k}] \begin{bmatrix} w_{i_1} \\ \vdots \\ w_{i_k} \end{bmatrix}. \tag{12}$$

The control input $\hat{u}(t)$ is actually applied to the controlled system at time k. As time k increases the above steps are repeated.

There are several methods to select k-nearest neighbors and appropriate weights [11,12]. They depend on what system generates the data. Without exact information on the controlled system, it is difficult to determine the suitable k-nearest neighbors and weights.

In this paper, instead of (11) and (12), we propose a method to use a solution $\boldsymbol{w}$ of

$$\boldsymbol{Aw}=\boldsymbol{b} \tag{13}$$

where

$$\boldsymbol{A} = [a_{i_1} \quad \cdots \quad a_{i_k}] \in \boldsymbol{R}^{(n+h+m)\times k} \tag{14}$$

and

$$\boldsymbol{w} = \begin{bmatrix} w_{i_1} \\ \vdots \\ w_{i_k} \end{bmatrix} \in \boldsymbol{R}^k. \tag{15}$$

As is known, when $(n+h+m) > k$ the solution is given by a least mean square solution as $\boldsymbol{w} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$, when $(n+h+m) < k$ the solution is given by a minimum norm solution as $\boldsymbol{w} = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{b}$.

Simulations

In this section, we show several simulation results to illustrate the performance of the proposed model-frees predictive control method. In all simulations, we used

$$\|\mathrm{a} - b\|_\infty = \max_i |a_i - b_i| \tag{16}$$

as the distance between a and b. In addition, we assume that ε is zero mean Gaussian noise with variance 0.05. As the desired reference signal, we used $r(t) = 2\sin(2\pi\, t/40)$.

Linear Case. We used the linear system

$$y(t+1)= y(t)-0.16y(t-1)-1.5u(t)+\varepsilon(t). \tag{17}$$

To generate the input and output data as in Fig. 1, we applied i.i.d. random noise generated from a uniform distribution [-5, 5] to the input *u(t)*. We stored 300 pairs of *u(t) and y(t)*.

Figure 2 shows the output *y*, the control input *u*, and the tracking error *e=r-y* in a simulation result when *n*=2, *m*=1, *h*=2 and *k*=10.

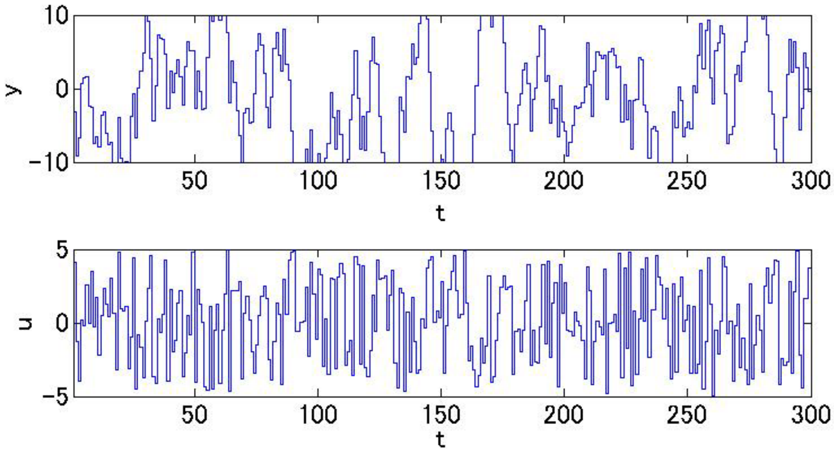

Figure 1 Stored past measurement data

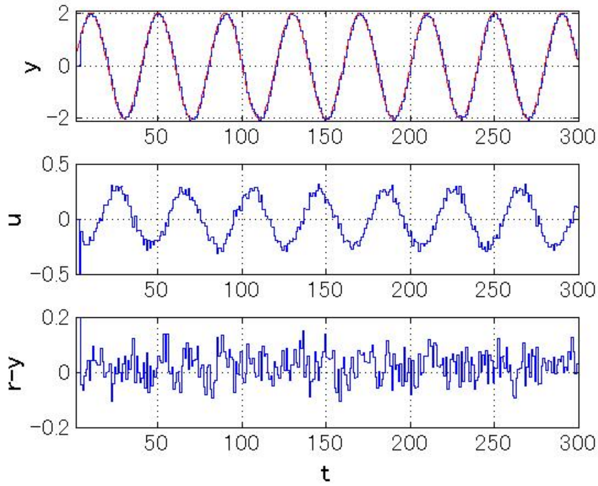

Figure 2 Simulation result of the linear system (17) when *n*=2, *m*=1, *h*=2 and *k*=10.

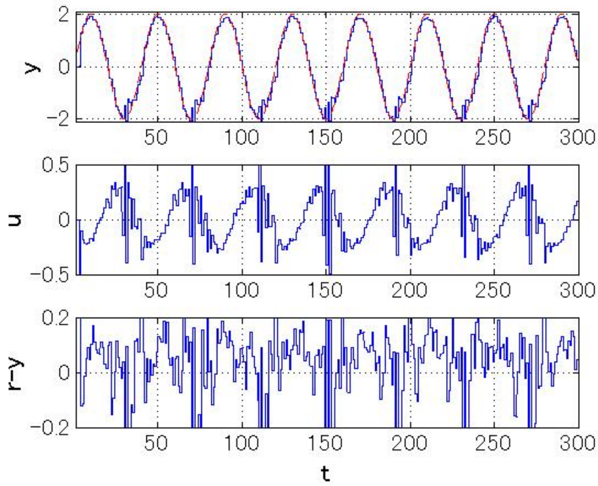

Figure 1 Simulation result of the linear system (17) when *n*=2, *m*=1, *h*=2 and *k*=4

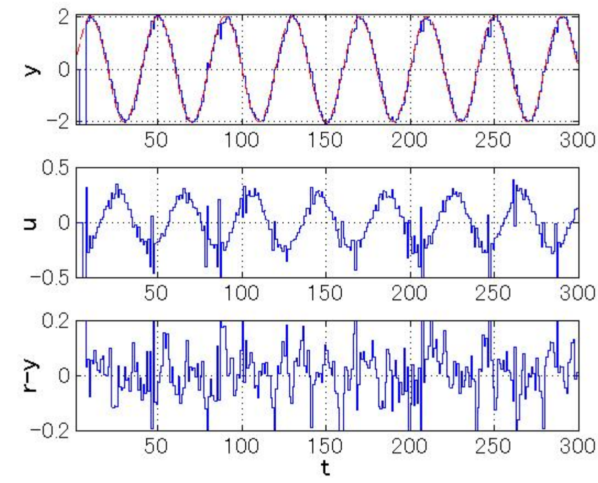

Figure 4 Simulation result of the linear system (17) when *n*=4, *m*=2, *h*=2 and *k*=10

When we used k=4, as shown in Fig. 3, the tracking error $e=r-y$ became worse. Furthermore, Fig 4. shows the result when we changed parameters in (5) and (6) as n=4 and m=2.

These results show that the control performance is sensitive to k and robust for the model parameters n and m.

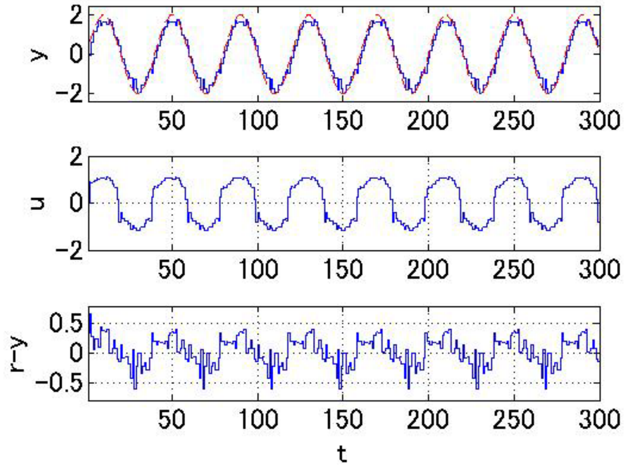

Figure 5 Simulation result of the nonlinear system (18) when *n*=1, *m*=1, *h*=1 and *k*=10

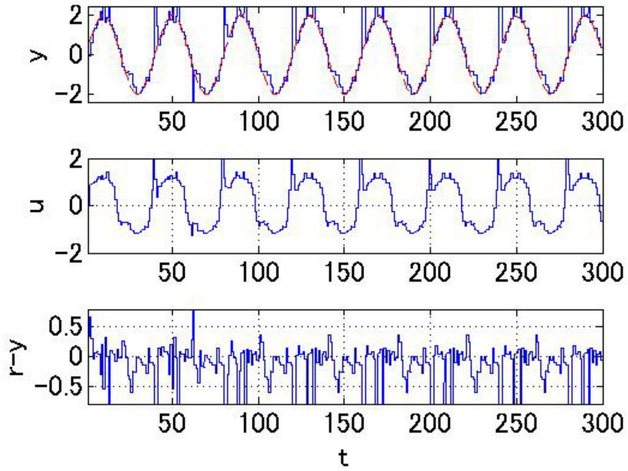

Figure 6 Simulation result of the nonlinear system (18) when *n*=1, *m*=1, *h*=1 and *k*=4

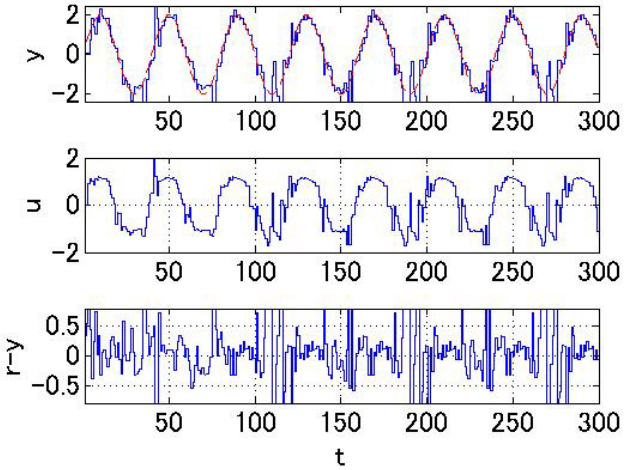

Figure 7 Simulation result of the nonlinear system (18) when *n*=2, *m*=2, *h*=1 and *k*=10

Nonlinear Case. We used the nonlinear system

$$y(t+1) = \frac{y(t)}{1+y(t)^2} + u(t)^3 + \varepsilon(t). \tag{17}$$

To generate the input and output data, we applied i.i.d. random noise generated from a uniform distribution [-3, 3] to the input $u(t)$. We stored 3000 pairs of $u(t)$ and $y(t)$.

Figure 5 shows the output y, the control input u, and the tracking error $e=r-y$ in a simulation result when n=1, m=1, h=1 and k=10. When we used k=4, as shown in Fig. 6, the tracking error $e=r-y$ became worse as in linear system (15). Furthermore, Fig 7. shows the result when we changed parameters in (5) and (6) as n=2 and m=2. These results also show that the control performance is sensitive to k and robust for the model parameters n and m.

Summary

In this paper, a new method for model-free predictive control in the just-in-time modelling framework in which online data and stored data of input/output of the controlled system are utilized. In the original model-free predictive control proposed by Inoue and Yamamoto [9,10], as in just-in-time modelling, we need to determine several tuning parameters for obtaining k nearest neighbors. Our proposed method can avoid such oppressiveness for the users.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number 23360183.

References

[1] Carlos E. Garcia, David M. Prett, and Manfred Morari: Model predictive control: Theory and practice? A survey, Automatica, Vol. 25, Issue 3, pp. 335–348 (1989)

[2] Karl J. Åström and Björn Wittenmark: Adaptive Control: Second Edition, Dover Publication (2008)

[3] A. Stenman: Model-free Predictive Control, Proc. of the 38th IEEE Conference on Decision and Control, pp. 3712-3717 (1999）

[4] A. Stenman, F. Gustafsson and H. Ljung: Just in time models for dynamical systems, Proc. of the 35th IEEE Conference on Decision and Control, pp. 1115-1120 (1996)

[5] A. Stenman, A. V. Nazin, and F. Gustafsson: Asymptotic properties of Just-In-Time models, Preprints of the 11th IFAC Symposium on System Identification, pp. 1249‐1254 (1997）

[6] M. W. Braun, D. E. Rivera, and A. Stenman: A ’Model-on-Demand’ identification methodology for non-linear process systems, International Journal of Control, Vol. 74, No. 18, pp. 1708–1717, (2001)

[7] G. Bontempi, M. Birattari and H. Bersini: Lazy Learning for local modeling and design; International Journal of Control, Vol. 72, No. 7/8, pp. 634–658 (1999)

[8] David W. Aha, Dennis Kibler, and Marc K. Albert, Instance based learning algorithms, Machine Learning, Volume 6, Issue 1, pp. 37–66 (1991)

[9] D. Inoue and S. Yamamoto: An Operation Support System based on Database-Driven On-Demand Predictive Control, Proc. of 2004 SICE Annual Conference, pp. 2024–2027 (2004)

[10] D. Inoue and S. Yamamoto: A Memory-Based Predictive Control Approach to a Braking Assist Problem, Proc. of SICE Annual Conference 2005, pp. 3186–3189 (2005)

[11] Q. Zheng and H. Kimura: Locally Weighted Regression Based on k Biparticle Neighbors, Proc. of the 42nd Japan Joint Automatic Control Conference, pp. 143‐144 (1999）

[12] Q. Zheng and H. Kimura: Just-In-Time Modeling for Function Prediction and its Applications, Asian. J. Control, Vol. 3, No. 1, pp. 35‐44 (2001）