DISSERTATION

Model-Free Predictive Control for Nonlinear Systems

Division of Electrical Engineering and Computer Science
Graduate School of Natural Science & Technology
Kanazawa University

Intelligent Systems and Information Mathematics

Student Number: 1424042011

Hongran LI

Supervisor: Prof. **Shigeru YAMAMOTO**

Date of submission: January, 2017

Abstract

In this thesis, we introduce a model-free predictive control method for linear and nonlinear system based on polynomial regressors that according to Volterra series. It is not only a class of polynomial representation of nonlinear system, but also natural extension of the classical linear system representation. Volterra series includes a series of nonlinear terms that contain product of increasing order Volterra kernel and input/output signal space, and the Volterra kernel and input/output signal space are not interdependent. Therefore, input/output signal space of polynomial regression can be used individually. The polynomial regression is a form that can be extended by the linear regression; it describe one relationship between the independent variable and dependent variable, which was modeled as a pth degree polynomial. Therefore, polynomial regression vectors can fit a nonlinear relationship between the independent variable vectors and dependent variable vectors that can describe nonlinear phenomena.

Model-free predictive control that directly computes the control input from massive input/output datasets and does not use a mathematical model. In contrast, conventional model predictive control relies on mathematical models. Although the underlying principle of model-free predictive control utilizes linear regression vectors comprising input/output data, it can also be applied to control nonlinear systems.

In this study, the linear regression vectors are extended to polynomial regression vectors that contain the control input and measurement output. It has recently been shown that the control offered by model-free predictive control can be improved. There is an indicator for us to discuss the conclusion that is an error with a reference trajectory based on the order of polynomial regressors. Using numerical simulations, we demonstrate the effectiveness of this approach, and then we extend these findings to multi-input multi-output nonlinear systems investigate the effectiveness of the approach through application of a wastewater treatment process.

Contents

List of Figures

Acknowledgement

First and foremost I want to thank my advisor Pro. Shigeru YAMAMOTO. It has been an honor to be his Ph.D. student. He has taught me, both consciously and unconsciously, how good research new algorithm. I appreciate all his contributions of time, ideas, experience productive and stimulating. His joy and enthusiasm for his research was contagious and motivational for me. I also want to say thank you with Pro. Satoshi YAMANE, Associate Prof. Yuki TODO, Ichiro JIKUYA and Akihiro HIRANO for their support, knowledge and everything that they gave to me.

Second, I want to say thank you for my parents, Qun LI and Benying SUN , receive my deepest gratitude and love for their dedication and the many years of support during my study abroad. A special thank for my fiancee, thank you for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible.

Third, the friendship of MoCCoS member are much appreciated and have led to many interesting and good spirited discussions relating to this research. I am also grateful to my friends in China for helping considerably.

Last but not least, I also thank to MEXT and Kanazawa University for give me scholarship to study.

Hongran LI

Kanazawa University

November 2016

Chapter 1

Introduction

The model-free predictive control is applicable to nonlinear systems assuming that the controlled system can be locally linearized. This allows short-length vectors to be constructed that are compatible with the regressor vector used in the autoregressive model to identify a locally linear model. In this thesis, the considered problem is how to to extend the short-length vectors to make them compatible with a polynomial regression model and improve control performance

1.1 Motivations and Objectives

Over last three decades, the idea of predictive control have been developed based on mathematical model. Generally, the objective of predictive control is compute a future manipulated variable input u to optimize the future output y of plant that can track a reference trajectory r. In other words, predictive control can predict the a few steps ahead outputs at each instant t with a determined horizon P, it calls predictive horizon. Predicted outputs $y(t+k|t)$ $k = 1, \cdots N$ depend on past inputs and outputs and the future inputs $u(t-k|t)$, $k = 0 \cdots N-1$ that can be sent to system and calculated. Through the calculation, the future output as close as possible to the reference trajectory $r(t+k)$, as shown in Fig.1.1

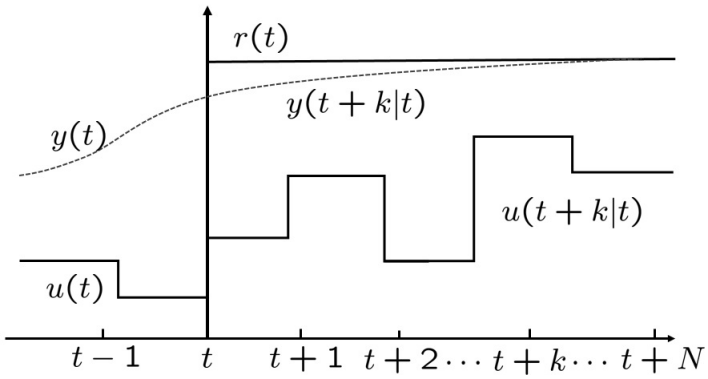


Figure 1.1: The basic idea of predictive control

Model predictive control (MPC) is designed based on a mathematical model of

plant. The model can be used in the system. Generally, the model can be designed as a discrete state-space equation:

$$\begin{aligned} L(t+1) &= AL(t) + Bu(t) \\ y_m(t) &= CL(t) \end{aligned} \tag{1.1}$$

In the practical, most often systems are described with mathematical model by a feedback loop. A simple structure shown in Fig.1.2. Although model predictive

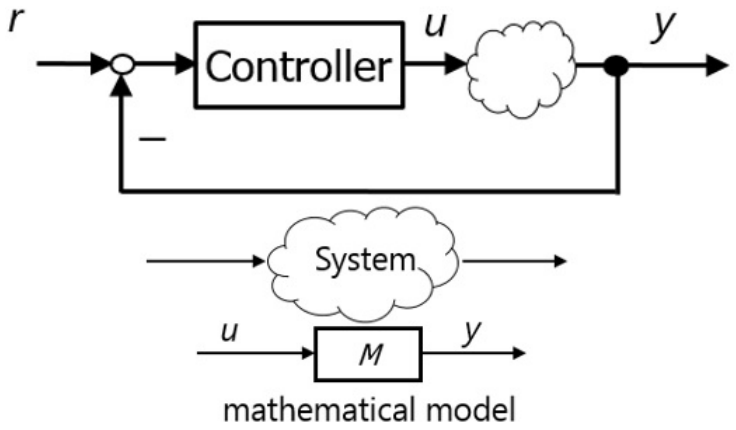


Figure 1.2: The simple structure of model predictive control

control is a good method for predictive control. However, most often to control the linear systems. When a difficult nonlinear system is involved, a corresponding perfect model will be required. Therefore, the computational burden will be increased.

Model-free predictive control was proposed as a data-driven control method that does not explicitly require a mathematical model [1]-[10]. In contrast to the standard model predictive control, which uses mathematical modeling, the model-free predictive control method uses the records of past input and output datasets and the current inputs and outputs to predict future inputs and outputs, as shown in Fig.1.3 The underlying principle of the method is just-in-time modeling, which

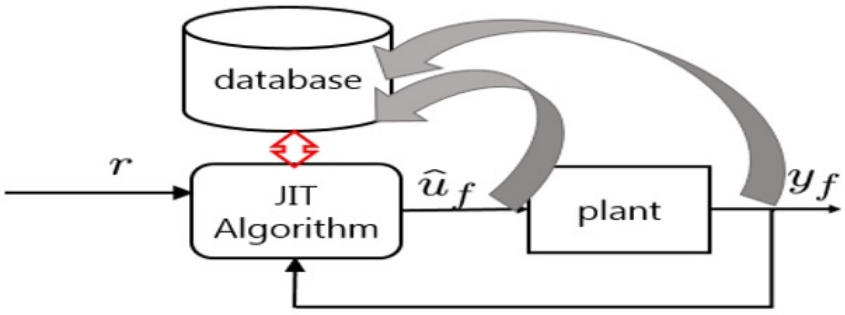


Figure 1.3: General Overview of Data Driven method

was originally proposed in [11]-[14]. This aims to adaptively derive a local linear

model using recorded data in the neighborhood of the query point [12, 13]. Just-in-time modeling is also referred to as model-on-demand [14], [15], lazy learning [16], or instance-based learning [17]. Just-in-time modeling is used in numerous applications, e.g., for predicting the production processes in the steel industry [18]-[21], PID parameter tuning [22], [23], and for soft sensors in industrial chemical processes [24].

The model-free predictive control proposed in [1]-[3] used massive short-length vectors cut from recorded past input and output sequences of the controlled system. Optimal control was predicted from the set of nearest short-length vectors to the most recent input and output sequences and the desired output. Although these vectors can be used to identify an auto-regressive model, in [25], just-in-time modeling was utilized to identify a local linear model for the standard model predictive control, which was then directly used to predict the control input as the linearly weighted average. A similar approach was followed in [4] and can be used to treat discretized input systems [5]. It has also been applied to an inverted pendulum system [6] and to a parallel mechanism with pneumatic drives [7]. To predict an optimal control input, a local weighted average method is frequently adopted. Recently, in [8] and [9], it has been suggested that the local linearly weighted average can be replaced by solving a linear algebraic equation using a least-norm and an ℓ_1-norm approach, yielding a mathematically much simpler model-free predictive control algorithm. In [10], three methods for model-free predictive control were compared in terms of their control performance.

1.2 Model-Free Predictive Control

The main contribution in the model-free predictive control is just-in-time modeling and how to calculate the optimal input for systems. The basic idea is via a local linear model that can be constructed successively, through expansion, it also can be summarized as optimization problems.

1.2.1 Basic Idea of Just-In-Time Modeling

For unknown nonlinear function $f(\cdot)$: $R^r \rightarrow R$

$$y_i = f(x_i) + \epsilon_i, \tag{1.2}$$

it generate a large amount of observation value $(x_i,\ y_i)$ with observation noise ϵ_i. The methdo of just in time collected the data from prior process to understand the characteristic of the system, behavior of the movement and as a training data to make the data driven in the future time. In Stenman paper [14], just-in-time modeling is also referred to as model-on-demand that make the weighted matrix as the nearest neighborhood data. He use the weights to optimize to get minimize point to measure MSE (Mean Square Error). In the method, a JIT estimator was considered that can give a consistent estimates as a function for same case in kernel method. when the nonlinear system was considered in time domain that can predict the output from the data sets of nonlinear system of prior input and output in dynamical system, which via a local linear model to construct successively.

Purely model-free predictive control with no model usage was proposed in [1]-[3]. It basically uses input and output sequences that are cut out into short-length vectors. Although the vectors can be used to identify an auto-regressive model, they are instead used to estimate a short-length vector corresponding to future input sequences by using locally weighted averaging. The idea can be used to treat discretized input systems [5]. It has also been applied to an inverted pendulum system [6] and a parallel mechanism with pneumatic drives [7].

1.2.2 Optimization of Just-In-Time Method

Instead of the local linear model, an optimal control input can be directly predicted by online current measured data and stored past data, do not using any local models. Recently, in [3], [8], [9] it has been pointed out that locally weighted averaging (LWA) can be replaced with optimization under a linear algebraic equation that relates to least-norm solutions and ℓ_1 minimization. This yields us a mathematically much simpler model-free predictive control algorithm. The effectiveness of the simplified algorithms is investigated in [10].

- For the LWA method, the information vector $\mathbf{a}_i$ are sorted according to the distance to query vector $\mathbf{b}$

$$d(\mathbf{a}_{i_1},\ \mathbf{b}) \leq \cdots \leq d(\mathbf{a}_{i_k},\ \mathbf{b}) \leq \cdots \leq d(\mathbf{a}_{i_N},\ \mathbf{b}). \tag{1.3}$$

  Furthermore, the weights $\mathbf{w}_{i_j}$ for $\mathbf{a}_{i_j}$ that satisfy

$$\mathbf{w}_{i_1} \geq \mathbf{w}_{i_2} \geq \cdots \geq \mathbf{w}_{i_k} \ and \ \sum_{j=1}^{k} = 1, \tag{1.4}$$

  the distance based on the ℓ_1-norm

$$\|\mathbf{w}\|_1 = \sum_{i=1}^{k} |\mathbf{w}_i| \tag{1.5}$$

  that can be defined as

$$d(\mathbf{a},\ \mathbf{b}) = \|\Omega^{-1}(\mathbf{a} - \mathbf{b})\|_1, \tag{1.6}$$

  where $\Omega = diag(\omega_1, \cdots, \omega_d)$. Then, for the $\mathbf{a}_j$,

$$\omega_i = \max_{j=1\cdots N} \mathbf{a}_{ji} - \min_{j=1\cdots N} \mathbf{a}_{ji}. \tag{1.7}$$

  The weight can be calculated as

$$\hat{\mathbf{w}}_i = tr(I_d - \Omega^{-1}(\mathbf{a}_i - \mathbf{b})(\mathbf{a}_i - \mathbf{b})^T\Omega^{-1}) \tag{1.8}$$

$$\mathbf{w}_i = \hat{\mathbf{w}}_i / \sum_{i}^{k} \hat{\mathbf{w}}_i. \tag{1.9}$$

Moreover, There are two methods to calculate the optimal input by linear equation $\mathbf{Aw} = \mathbf{b}$, in here, $\mathbf{A} \in R^{d\times k}$, $\mathbf{w} \in R^k$

- By least norm solution. when $d > k$, the solution can be given by a least mean square solution as $\mathbf{w} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. when $d < k$, the solution can be given by a minimum norm solution $\mathbf{w} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$ of

$$\min_{\mathbf{w}} \|\mathbf{Aw} - \mathbf{b}\|_2 \tag{1.10}$$

- By ℓ_1-minimization. Using all vectors in matrix $\mathbf{A}$, to solve the optimization problem as

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ subject to } \mathbf{Aw} - \mathbf{b} = 0 \tag{1.11}$$

From [10], we saw that the effectiveness of ℓ_1-minimization method is better than least-norm solutions, therefore, we choose ℓ_1-minimization to calculate the optimal input in this study.

In this thesis, the truncated Volterra series was used that is also a polynomial regression model [26], [27]. Using it, the short-length vectors can be extended to regressor vectors, thus the data in the vectors can become abundance. The effectiveness of the polynomial regression in the model-free predictive control is illustrated via simulations using linear and nonlinear system. In the simulations, a smoothing filter was adopted for the reference signal to prevent instability arising in the closed loop system because of abrupt changes in the signal.

1.3 Extension Via Polynomial Regressors

Volterra series can describe the nonlinear systems via the extended standard convolution description of linear systems by a series of polynomial integral operators with increasing order. It similar to the Taylor series, the difference is the ability to capture memory. Taylor series only can describe instantaneous input and output of nonlinear systems, in contrast, Volterra series describe output of nonlinear systems that depends on the input at all past times. This provides the ability to capture memory of systems. So it is very suitable for predictive control. On the other hand, Volterra series consist of product of operator and input signal that was built up with weighted sum of products or integral of products. There is not interdependence relationship between the operator and input signal. Thus, we can use the input signal alone with weighted sum of products. It is defined as polynomial regression models that apply to extend the short-length vectors.

1.4 Applications

To verify the performance of model-free predictive control method based on polynomial regression. We need different systems to illustrate the effectiveness of the polynomial regression in the model-free predictive control

- For system identification, auto- regressive eXogeneous often can be used that structure is the simple linear difference equation

$$y(t) = \theta^T \phi(t) + \epsilon(t) \tag{1.12}$$

where $\theta(t)$ is the regression vector, $\phi(t)$ is a vector that include past input/output, $\epsilon(t)$ is a noise.

- For the neural networks system, the output of the plant depends linearly on the control input. It makes the computation of the latter relatively straightforward

$$y(t) = f[\phi(t)] + \epsilon(t) \tag{1.13}$$

where $f[\cdot]$ is nonlinear function.

- For practical application, the activated sludge process was considered that is a typical biodegradation wastewater treatment process (WWTP).

1.5 Outline of The Dissertation

The thesis is divided into three chapters, excluding the introduction chapter and conclusion chapter. The first two chapter give a theory of Volterra series, and applicate to the model-free predictive control. the last two chapter is about application for linear/nonlinear system, and wastewater treatment process.

Chapter 2, we describe the model-free predictive control in detail, and how to optimize the input.

Chapter 3 is the theory of Volterra series that can describe the nonlinear systems via the extended standard convolution description of linear systems by a series of polynomial integral operators with increasing order. And then we express how to achieve performance of model-free predictive control by the polynomial regression, moreover some simulations will be discussed.

Chapter 4, we use the wastewater treatment process to illustrate the performance in practical application.

Chapter 2

Model-Free Predictive Control

For the model-free predictive control, there are a lot of method to solve the optimization problem. The methods are Locally Weight Average (LWA), Least Norm and ℓ_1-norm minimization.

In this thesis, we use ℓ_1-norm minimization to solve the optimization problem.The key is weight factor $\mathbf{w}$ ($\mathbf{Aw} = \mathbf{b}$). The weight factor was used into the system to get the optimal input u to the system. The database is made to enrich the data that based on previous experiment and the database collected to be a single matrix $\mathbf{A}$. The $\mathbf{b}$ is called query vector that designed from a few past output (y), reference signal (r) and a few past input (u).

2.1 Linear Regression Case

Consider the following discrete-time system:

$$y(t) \quad = f(\mathbf{x}_1(t)) + \varepsilon(t), \tag{2.1}$$

where $u \in \mathbf{R}$ is the control input, $y \in \mathbf{R}$ is the controlled output, ε is independent and identically distributed noise, and

$$\mathbf{x}_1(t) \quad = \begin{bmatrix} x_1(t) \\ \vdots \\ x_{\hat{m}}(t) \\ x_{\hat{m}+1}(t) \\ \vdots \\ x_{\hat{m}+\hat{n}}(t) \end{bmatrix} = \begin{bmatrix} u(t-\hat{m}) \\ \vdots \\ u(t-1) \\ y(t-\hat{n}) \\ \vdots \\ y(t-1) \end{bmatrix} \in \mathbf{R}^{\hat{n}+\hat{m}} \tag{2.2}$$

is the regression vector. We assume that $\hat{n}$ and $\hat{m}$ are unknown together with the nonlinear function f.

The control objective is to let the h-step output trajectory

$$\hat{\mathbf{y}}_f(t) = \begin{bmatrix} \hat{y}(t+1) \\ \vdots \\ \hat{y}(t+h) \end{bmatrix} \in \mathbf{R}^h \tag{2.3}$$

track the given reference trajectory

$$\mathbf{r}(t) = \begin{bmatrix} r(t+1) \\ \vdots \\ r(t+h) \end{bmatrix} \in \mathbf{R}^h. \tag{2.4}$$

The parameter h is the prediction horizon. To achieve this control objective, we predict an h-step future input sequence

$$\hat{\mathbf{u}}_f(t) = \begin{bmatrix} \hat{u}(t|t) \\ \vdots \\ \hat{u}(t+h-1|t) \end{bmatrix} \in \mathbf{R}^h. \tag{2.5}$$

Assumption 1 *When $\varepsilon(k) \equiv 0$, there exists a $\hat{\mathbf{u}}_f(t)$ such that*

$$\hat{\mathbf{y}}_f(t) = \mathbf{r}(t). \tag{2.6}$$

The model-free predictive control [1],[2] utilizes recorded past data $\{u(t), y(t)\}$. For N pairs of the recorded data $\{u(t_j), y(t_j)\}$ $(j = 1, 2, \ldots, N)$, we define the following vectors:

$$\mathbf{a}_j = \begin{cases} \begin{bmatrix} \mathbf{u}_p(t_j) \\ \mathbf{y}_p(t_j) \\ \mathbf{y}_f(t_j) \end{bmatrix} \in \mathbf{R}^{m+n+h-1}, & m \geq 2, \\ \begin{bmatrix} \mathbf{y}_p(t_j) \\ \mathbf{y}_f(t_j) \end{bmatrix} \in \mathbf{R}^{n+h}, & m = 1, \end{cases} \tag{2.7}$$

$$\mathbf{c}_j = \mathbf{u}_f(t_j) \in \mathbf{R}^h, \tag{2.8}$$

where

$$\mathbf{u}_p(t) = \begin{bmatrix} u(t-m+1) \\ \vdots \\ u(t-1) \end{bmatrix} \in \mathbf{R}^{m-1},\ m \geq 2, \tag{2.9}$$

$$\mathbf{y}_p(t) = \begin{bmatrix} y(t-n+1) \\ \vdots \\ y(t) \end{bmatrix} \in \mathbf{R}^n, \tag{2.10}$$

$$\mathbf{y}_f(t) = \begin{bmatrix} y(t+1) \\ \vdots \\ y(t+h) \end{bmatrix} \in \mathbf{R}^h, \tag{2.11}$$

$$\mathbf{u}_f(t) = \begin{bmatrix} u(t) \\ \vdots \\ u(t+h-1) \end{bmatrix} \in \mathbf{R}^h. \tag{2.12}$$

Remark 1 *The recorded data $\{u(t_j), y(t_j)\}$ $(j = 1, 2, \ldots, N)$ must sufficiently cover the operating points to track r, as discussed in Chapter 4.*

Remark 2 *The sizes m and n are parameters in model-free predictive control. Unless the exact values of $\hat{m}$ and $\hat{n}$ are available, m and n must be estimated to obtain the best control performance, as discussed in Chapter 4.*

2.2 Optimization of Input

To find a future input sequence $\hat{\mathbf{u}}_f(t)$, using the most recent input and output trajectories $\mathbf{u}_p(t), \mathbf{y}_p(t)$ and the reference trajectory $\mathbf{r}(t)$, a query vector is defined as follows:

$$\mathbf{b} = \begin{cases} \begin{bmatrix} \mathbf{u}_p(t) \\ \mathbf{y}_p(t) \\ \mathbf{r}(t) \end{bmatrix} \in \mathbf{R}^{m+n+h-1}, & m \geq 2, \\ \begin{bmatrix} \mathbf{y}_p(t) \\ \mathbf{r}(t) \end{bmatrix} \in \mathbf{R}^{n+h}, & m = 1, \end{cases} \tag{2.13}$$

we solve

$$\mathbf{A}\mathbf{w} = \mathbf{b}, \tag{2.14}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_N \end{bmatrix} \in \mathbf{R}^{(n+m+h)\times N}, \tag{2.15}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} \in \mathbf{R}^N, \tag{2.16}$$

Then, the future input sequence is given as

$$\hat{\mathbf{u}}_f(t) = \mathbf{C}\mathbf{w}, \tag{2.17}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_N \end{bmatrix} \in \mathbf{R}^{h\times N}. \tag{2.18}$$

The first element $\hat{u}(t|t)$ of $\hat{u}_f(t)$ is only applied to the system as $u(t)$.

Remark 3 *Matrices* $\mathbf{A}$ *and* $\mathbf{C}$ *comprise only recorded data. Query vector* $\mathbf{b}$ *is updated according to the measured data, and vector* $\mathbf{w}$ *is determined in every sampling interval. In [10], a method to update* $\mathbf{A}$ *and* $\mathbf{C}$ *in real-time was proposed.*

In earlier studies [1]-[3], the linear equation was not used. Instead, only a few vectors $\mathbf{a}_i$ close to $\mathbf{b}$ were chosen. The vector $\mathbf{w}$ was determined using Akaike's Final Prediction Error as the criterion. In [8], a least-norm solution was proposed for $\mathbf{w}$. More recently, an ℓ_1-minimization

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ subject to } \mathbf{A}\mathbf{w} - \mathbf{b} = 0 \tag{2.19}$$

was presented [9]. where $\|\cdot\|_1$ is the ℓ_1 norm of the vector, defined as follows:

$$\|\mathbf{w}\|_1 = \sum_{i=1}^{N} |w_i|. \tag{2.20}$$

Using ℓ_1-minimization, we can avoid explicitly choosing a nearest vector $\mathbf{a}_i$. Multiple algorithms have been proposed to solve the ℓ_1-minimization problem [30], and a useful tool is now available [31].

Chapter 3

Extension Via Polynomial Regressors

Volterra series can describe the nonlinear systems via the extended standard convolution description of linear systems by a series of polynomial integral operators with increasing order. Volterra series describe output of nonlinear systems that depends on the input at all past times. This provides the ability to capture memory of systems.

3.1 Volterra Series

The Volterra model is a general nonlinear model with an output $y(t)$ and an input $u(t)$ that earliest approach to a systematic characterization of nonlinear systems dates back to V. Volterra (Volterra, 1887). For a system can be defined mathematically as an output $y(t)$ with the product of input $x(t)$ and operator H which can be expressed by

$$y(t) = H\mathbf{x}(t) + e(t) \tag{3.1}$$

where e is independent and identically distributed noise. In traditional systems theory, H is a linear operator vector, thus the system response can be described by a convolution as

$$y(t) = H\mathbf{x}(t) + e(t) = \int_0^t h_1(\tau_1)x(t-\tau_1)d\tau_1 + e(t) \tag{3.2}$$

Volterra extended this linear system to nonlinear systems by adding a series of nonlinear integral operators as

$$\begin{aligned} y(t) = h_0 \quad &+ \quad \int_a^b h_1(\tau_1)x(t-\tau_1)d\tau_1 \\ &+ \quad \int_a^b h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 \\ &+ \quad \int_a^b h_2(\tau_1,\tau_2,\tau_3)x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)d\tau_1 d\tau_2 d\tau_3 \\ &+ \quad \cdots + e(t) \end{aligned} \tag{3.3}$$

Comparing with continuous time systems, discrete system is simply described by a function. Based on theory of convolution, the discretized Volterra series can be defined as

$$y(t) = h_0 + \sum_{p=1}^{P} \sum_{\tau_1=a}^{b} \cdots \sum_{\tau_p=a}^{b} h_p(\tau_1, \ldots, \tau_p) \prod_{j=1}^{p} x(t - \tau_j) \tag{3.4}$$

$h_p(\tau_1, \ldots, \tau_p)$ are called discrete time Volterra kernels. For the number P, a and b, when P is finite, the series can be truncated. If a, b and P are finite, we can call Volterra series is doubly finite Volterra series.

3.2 Polynomial Regression Models

In this section, we review a polynomial regression model that based on a Volterra model [26], [27].

The so-called Volterra model is a general nonlinear model with an output $y(t)$ and an input $u(t)$ that can be expressed as follows:

$$y(t) = \sum_{p=0}^{P} H_p(\mathbf{x}_1(t)) + e(t) \tag{3.5}$$

where e is independent and identically distributed noise

$$\mathbf{x}_1(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_L(t) \end{bmatrix} \in \mathbf{R}^L, \tag{3.6}$$

$$x_i(t) = u(t - i), \quad i = 1, \ldots, L, \tag{3.7}$$

and

$$\begin{aligned} H_p(\mathbf{x}_1(t)) = \; & \textstyle\sum_{i_1=1}^{L} \cdots \sum_{i_p=1}^{L} h_p(i_1, \ldots, i_p) \\ & \times \textstyle\prod_{k=1}^{p} u(t - i_k) \end{aligned} \tag{3.8}$$

where each $h_p(i_1 \cdots i_p)$ is called a Volterra kernel of the system. In general, the expansion order P is infinity. Here, we consider the truncated model, that is, for $p > P$, $|H_p(\mathbf{x}_1(t))|$ is sufficiently small. For $p > 1$, by defining the pth order monomials (homogeneous) regressor vector

$$\mathbf{x}_p(t) = \mathbf{x}_{p-1}(t) \otimes \mathbf{x}_1(t) \in \mathbf{R}^{L^p}, \tag{3.9}$$

where $\otimes$ denotes the tensor (Kronecker) product; we can rewrite (3.8) as

$$H_p(\mathbf{x}_1(t)) = \mathbf{x}_p^\top(t)\mathbf{h}_p, \tag{3.10}$$

where $\mathbf{h}_p$ is a vector containing Volterra kernels $h_p(i_1 \cdots i_p)$. By using (3.8), another expression of (3.5) is given as

$$y(t) = \phi^\top(t)\mathbf{h} + e(t), \tag{3.11}$$

where

$$\phi^\top(t) = \begin{bmatrix} 1 & \mathbf{x}_1^\top(t) & \cdots & \mathbf{x}_P^\top(t) \end{bmatrix} \tag{3.12}$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_P \end{bmatrix} \in \mathbf{R}^{\sum_{p=0}^{P} L^p} \tag{3.13}$$

By changing the meaning of the index t in (3.6) so as to neither limit the time nor restrict $x_i(t) = u(t-i)$, we obtain a polynomial regression model. Since we can set $x_i(t) = u(t-i)$, the Volterra model is a special polynomial regression model.

To use polynomial regression, we define another form of the Volterra model. That is, we define

$$\mathbf{x}_1(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_m(t) \\ x_{m+1}(t) \\ \vdots \\ x_L(t) \end{bmatrix} = \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-m) \\ y(t-1) \\ \vdots \\ y(t-n) \end{bmatrix} \tag{3.14}$$

and $L = m + n$. By adopting this model, we can reduce the truncated order P.

Since the tensor product is a particularly effective method to establish the topological vector space, it yields several duplicate terms. By eliminating these duplicate terms, we define the pseudo-tensor product $\tilde{\otimes}$, for instance,

$$\begin{bmatrix} a & b \end{bmatrix} \tilde{\otimes} \begin{bmatrix} a & b \end{bmatrix} = \begin{bmatrix} a^2 & ab & b^2 \end{bmatrix}, \tag{3.15}$$

by unifying the duplicated term ab in $\begin{bmatrix} a & b \end{bmatrix} \otimes \begin{bmatrix} a & b \end{bmatrix}$.

When we use the pseudo-tensor product in (3.9) as

$$\mathbf{x}_p(t) = \mathbf{x}_{p-1}(t) \tilde{\otimes} \mathbf{x}_1(t) \in \mathbf{R}^{\binom{L+p-1}{L-1}}. \tag{3.16}$$

the size of ϕ and $\mathbf{h}$ can be reduced to

$$1 + \sum_{p=0}^{P} \binom{L+p-1}{L-1}. \tag{3.17}$$

3.3 Extension to a Polynomial Regressor

It is known that the regression vector $\mathbf{x}_1$ in the nonlinear system (2.1) can be extended to a polynomial regressor as follows.

First, we define the pseudo-tensor $\tilde{\otimes}$ as removing duplicated terms in the usual tensor (Kronecker) product $\otimes$. For example,

$$\begin{bmatrix} a & b \end{bmatrix}^\top \tilde{\otimes} \begin{bmatrix} a & b \end{bmatrix}^\top = \begin{bmatrix} a^2 & ab & b^2 \end{bmatrix}^\top, \tag{3.18}$$

$$\begin{bmatrix} a & b \end{bmatrix}^\top \otimes \begin{bmatrix} a & b \end{bmatrix}^\top = \begin{bmatrix} a^2 & ab & ab & b^2 \end{bmatrix}^\top. \tag{3.19}$$

Algorithm 1 Model-free predictive control algorithm

Determine m, n, N, h, and the order P. Construct $\mathbf{A}$ and $\mathbf{C}$. $t \leftarrow 0$.
while $t \leq \max(n, m)$ **do**
  Measure $y(t)$ and apply $u(t)$ with an appropriate value. Increment the time as $t \leftarrow t + 1$.
end while
repeat
  Measure $y(t)$ and define a query vector $\mathbf{b}$.
  Compute $\mathbf{w}$ by solving ℓ_1 minimization (2.19).
  Apply $u(t) := \hat{u}(t|t)$ to the system.
  $t \leftarrow t + 1$
until a terminate condition is met.

For $p > 1$, we define the pth order monomials regressor vector

$$\mathbf{x}_p(t) = \mathbf{x}_{p-1}(t)\tilde{\otimes}\mathbf{x}_1(t) \in \mathbf{R}^{n_p}, \tag{3.20}$$

where

$$n_p = \binom{p + \hat{n} + \hat{m} - 1}{\hat{n} + \hat{m} - 1} = \frac{(p + \hat{n} + \hat{m} - 1)!}{p!(\hat{n} + \hat{m} - 1)!}. \tag{3.21}$$

Using the monomial regressor vectors $\mathbf{x}_p$, a polynomial regression model is defined as follows:

$$y(t) \quad = \phi^\top(t)\mathbf{h} + \varepsilon(t), \tag{3.22}$$

where

$$\phi^\top(t) \quad = \quad \begin{bmatrix} 1 & \mathbf{x}_1^\top(t) & \cdots & \mathbf{x}_P^\top(t) \end{bmatrix} \in \mathbf{R}^{n_h}, \tag{3.23}$$

$$\mathbf{h} \quad = \quad \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1^\top & \cdots & \mathbf{h}_P^\top \end{bmatrix}^\top \in \mathbf{R}^{n_h}. \tag{3.24}$$

The size of ϕ and $\mathbf{h}$ is

$$n_h = 1 + \sum_{p=1}^{P} n_p. \tag{3.25}$$

When $\mathbf{x}_i$ does not contain y, (3.22) is the truncated Volterra model.

Using the polynomial regression model expression, we can reformulate the model-free predictive control as follows:

$$\mathbf{a}_{p,j} \quad := \quad \mathbf{a}_{p-1,j}\tilde{\otimes}\mathbf{a}_{1,j}, \quad \mathbf{a}_{1,j} := \mathbf{a}_j, \tag{3.26}$$

$$\mathbf{b}_p \quad := \quad \mathbf{b}_{p-1}\tilde{\otimes}\mathbf{b}_1, \qquad \mathbf{b}_1 := \mathbf{b}. \tag{3.27}$$

Using (3.26) and (3.27), we define the following equations:

$$\mathbf{A} \quad = \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{a}_{1,1} & \cdots & \mathbf{a}_{1,N} \\ \vdots & & \vdots \\ \mathbf{a}_{P,1} & \cdots & \mathbf{a}_{P,N} \end{bmatrix} \in \mathbf{R}^{L \times N}, \ \mathbf{b} = \begin{bmatrix} 1 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_P \end{bmatrix} \in \mathbf{R}^L, \tag{3.28}$$

where

$$L = 1 + \sum_{p=1}^{P} L_p, \tag{3.29}$$

$$L_p = \binom{p+m+n+h-2}{m+n+h-2} = \frac{(p+m+n+h-2)!}{p!(m+n+h-2)!}. \tag{3.30}$$

The procedure for the model-free predictive control is summarized in Algorithm 1.

Remark 4 *The computational complexity of the Dual Augmented Lagrangian Method (DALM) in solving ℓ_1-minimization is $O(L^2+LN)$ [30]. Because size L is dominated by $P^{m+n+h-1}$ when P becomes larger, the complexity of Algorithm 1 is $O(P^{2(m+n+h-1)} + P^{(m+n+h-1)}N)$. Theoretically, when P is large, computational burden is not avoidable. Thus, the potential practical application of model-free predictive control using DALM is restricted to systems with slow dynamics.*

3.4 Simulation Result and Discussions

In this section, we illustrate several simulation results to show the effectiveness of the proposed method. Throughout the simulations, we used the square signal reference

$$r(t) = \begin{cases} 0 & 200k \le t < 50 + 200k \\ 1 & 50 + 200k \le t < 100 + 200k \\ 0 & 100 + 200k \le t < 150 + 200k \\ -1 & 150 + 200k \le t < 200 + 200k \end{cases}$$
$$k = 0, 1, \ldots \tag{3.31}$$

3.4.1 Linear System

We first used the linear system that is a ARX model.

$$y(t) - 1.7y(t-1) + 0.72y(t-2) = 0.1u(t-1) + 0.2u(t-2) + e(t) \tag{3.32}$$

with stable poles 0.9 and 0.8 and an unstable zero -2 [33]. To apply a random sequence $e(t)$ according to a Gaussian distribution with zero mean, variance 0.001^2, and a random sequence $u(t)$ generated from a uniform distribution $[-2, 2]$, we prepared a dataset containing samples ($N = 300$) of $u(t)$ and $y(t)$, as shown in Fig. 3.1.

By using parameters for model-free predictive control $m = 3$, $n = 2$, $P = 3$, and $h = 2$ under the noisy condition $e(t) \sim \mathcal{N}(0, 0.001^2)$, we obtained the control result shown in Fig. 3.2. It shows that the output y can track the reference r.

Next, we used an overestimated order $m = 3$, with other parameters being the same as before, i.e., $n = 2$, $P = 3$, and $h = 2$, and obtained the control result shown in Fig. 3.3; this is similar to that shown in Fig. 3.2.

From the two results (Fig. 3.2 and 3.3), we see that the proposed method can achieve the desired control performance even when the order of the system is overestimated.

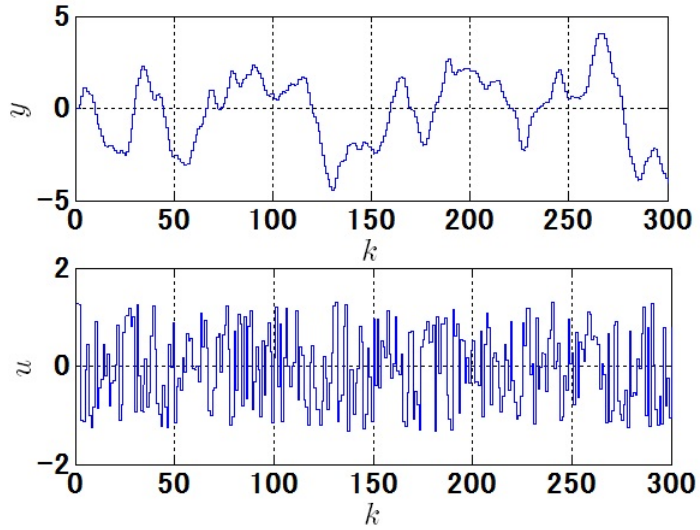


Figure 3.1: Stored measurement data of the linear system (3.32) for model-free predictive control

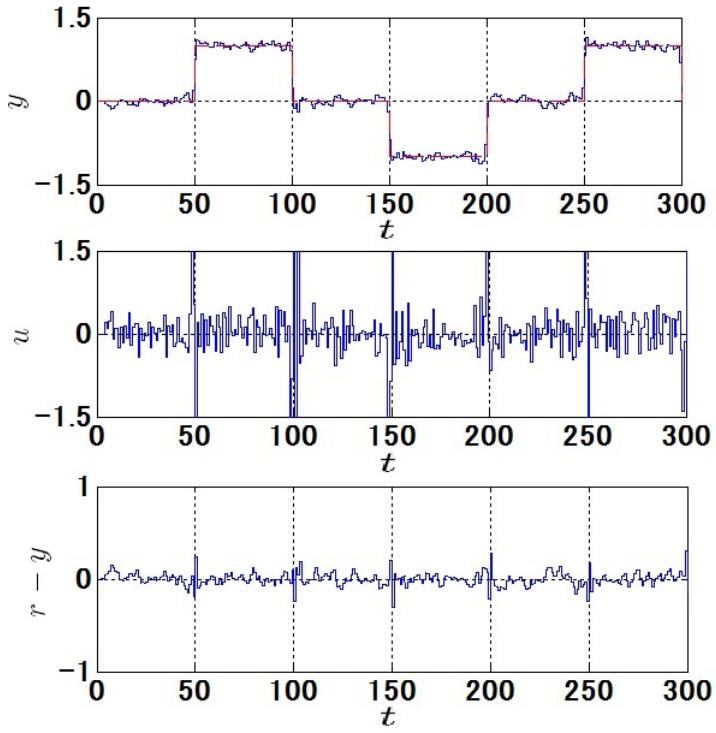


Figure 3.2: Simulation result of model-free predictive control for the linear system (3.32)

3.4.2 Nonlinear Systems

In this section, we present simulation results to demonstrate the application of our proposed method to a nonlinear system.

$$y(t+1) = \frac{y(t)}{1+y(t)^2} + u(t)^3 + \varepsilon(t). \tag{3.33}$$

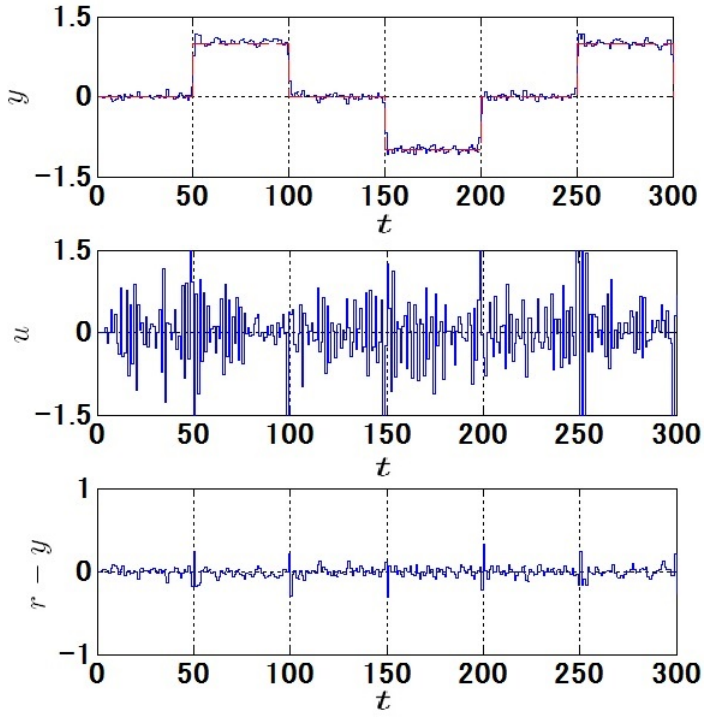


Figure 3.3: Simulation result of model-free predictive control with the overestimated order for the linear system (3.32)

This system was also used in [10] to compare the model-free predictive control methods. Clearly, when $\varepsilon(k) \equiv 0$, (3.33) has three fixed points $(u, y) = (0, 0), (2^{-1/3}, 1)$, and $(-2^{-1/3}, -1)$, and only the first is unstable. In all simulations, we applied a random sequence $\varepsilon(t)$ using a Gaussian distribution $\varepsilon(t) \sim \mathcal{N}(0, 0.1^2)$.

In the simulations, to prevent instability caused by abrupt changes in r, we applied a smoothing filter

$$\tilde{r}(t) = \frac{1}{\alpha} \sum_{i=0}^{\alpha-1} r(t-i) \tag{3.34}$$

for an integer $\alpha > 0$ and redefined (2.4) as follows:

$$\mathbf{r}(t) = \begin{bmatrix} \tilde{r}(t+1) \\ \vdots \\ \tilde{r}(t+h) \end{bmatrix} \in \mathbf{R}^h. \tag{3.35}$$

In all simulations, we set $\alpha = 5$.

3.4.3 Comparison of datasets

We first compared two datasets containing $N = 500$ input and output samples, as shown in Figs. 3.4 and 3.5. Using a uniform distribution $[-\sqrt{7}/2, \sqrt{7}/2]$, $u(t)$ was generated and applied to (3.33) to obtain the first dataset. The second dataset was

generated to use PI control:

$$u(t) = 0.6e(t) + 0.4 \sum_{\tau=-\infty}^{t} e(\tau) + v(t) \tag{3.36}$$

$$e(t) = r(t) - y(t) \tag{3.37}$$

Using a uniform distribution $[-0.1, 0.1]$, $v(t)$ was generated. In the second dataset more y values were presented close to the references $r = -1, 0$, and 1 than in the first dataset, as can be observed in the histograms in Figs. 3.4 and 3.5. To use parameters $(n, m, h) = (1, 2, 1)$ and $P = 3, 2$, and 1, we compared the performance of the model-free predictive control when using two datasets. In the simulation results shown in Figs. 3.6 and 3.7, the broken line represents the reference r. It can be observed that the tracking error $e = r - y$ in Fig. 3.7 is smaller than that in Fig. 3.6. Therefore, we can conclude the following:

- Datasets containing many y around r are required to reduce the tracking error.

3.4.4 Comparison of Parameters

The output $y(t)$ of (3.33) solely depends on $y(t-1)$ and $u^3(t-1)$. Next, we investigated whether the size $n = 1$, $m = 1$, and $P = 3$ yielded the best control performance. To select the 54 combinations of n, m, h, and P, we compared the tracking error at a ratio of

$$20 \log_{10} \left(\sum_{t=1}^{300} e(t)^2 \right) / \left(\sum_{t=1}^{300} \varepsilon(t)^2 \right). \tag{3.38}$$

We generated 100 databases and another 100 random $\varepsilon(t)$ values to simulate the model-free predictive control. We conducted 100 trails of each combination of n, m, h, and P. These are shown in the boxplots in Fig. 3.8, where the bottom of the box represents the first quartile, the top of the box represents the third quartile, the horizontal line near the middle of the box indicates the median, a vertical line extends to the maximum value and another vertical line extends to the minimum value, and the potential outliers are represented by "+". From Fig. 3.8, it can be observed that a smaller tracking error was observed when $P = 3$ than when $P = 1, 2$, for any combination of parameters. When we tried higher orders $P = 4, 5, 6$, control failed several times. Therefore, we have omitted the results from Fig. 3.8 and conclude that $P = 3$ is the best. Noting that the order $P = 1$ corresponds to the model-free predictive control using linear regression vectors, we can conclude the following:

- Polynomial regressors are more effective for the model-free predictive control of a nonlinear system than the linear regressor adopted in the existing model-free predictive control.

Next, for $P = 3$, we compared the boxplots obtained with the 10 best combinations of parameters (n, m, h). This is shown in Table 4.1, where $(n, m, h) = (2, 1, 3)$;

$(n, m, h) = (1, 1, 3)$; and $(n, m, h) = (1, 2, 6)$ show the lowest third quartile. Moreover, $(n, m, h) = (2, 1, 3)$ showed a lower first quartile than $(n, m, h) = (1, 1, 3)$ and $(n, m, h) = (1, 2, 6)$; its interquartile range (IQR), which measures the difference between the third and first quartiles was smaller, indicating tighter distribution of performance. Based on the results in Fig. 3.8 and Table 4.1, we conclude that the optimal parameters are $(n, m, h, P) = (2, 1, 3, 3)$ for the nonlinear system given by (3.33) and

- There exists a best combination of parameters, the order P, sizes of n and m, and horizon h.
- An appropriate combination of parameters must be selected to improve control performance.
- An unnecessarily large order P causes instability.
- There is no explicit criterion for selecting an appropriate prediction horizon h as in standard model predictive control.

Remark 5 *All simulations were performed in MATLAB on a Thirdwave Diginnos PC with a dual-core 3.40GHz Intel Core i3-4130 processor and 4 GB of memory. The execution time of DALM did not exceed 0.064 s for any combination of parameters* (n, m, h, P).

Table 3.1: Best 10 quartiles when order $P = 3$, as in Fig. 3.8.

$n,\ m,\ h$	Third quartile	First quartile	IQR
2, 1, 3	13.74	12.22	1.52
1, 1, 3	13.82	12.26	1.56
1, 2, 6	13.83	12.41	1.42
1, 2, 2	14.02	12.53	1.49
1, 2, 3	14.05	12.40	1.65
1, 1, 4	14.07	12.61	1.46
1, 1, 2	14.13	12.56	1.57
2, 1, 2	14.17	12.62	1.55
1, 2, 4	14.19	12.53	1.66
1, 2, 5	14.23	12.24	1.99

3.4.5 Supplemental Results

Using another nonlinear system in [32]

$$\begin{aligned} y(t) &= \frac{z(t)(y(t-3)-1)u(t-2)+u(t-1)}{1+y(t-3)^2+y(t-2)^2}+\varepsilon(t), \\ z(t) &= y(t-1)y(t-2)y(t-3), \end{aligned} \tag{3.39}$$

Table 3.2: Best 10 quartiles when order $P = 2$, as in Fig. 3.9.

$n,\ m,\ h$	Third quartile	First quartile	IQR
2, 1, 3	10.28	8.80	1.48
1, 1, 3	10.36	8.90	1.46
2, 1, 2	10.56	9.29	1.27
2, 1, 4	10.87	9.24	1.63
2, 1, 5	11.06	9.29	1.81
1, 2, 3	11.60	10.20	1.4
1, 2, 4	11.78	10.35	1.43
2, 1, 6	11.80	9.86	1.94
1, 1, 4	12.10	9.64	2.46
1, 2, 5	12.17	10.35	1.82

we investigated the efficiency of utilization of polynomial regressor vectors in model-free predictive control under the same conditions as in previous subsections. For (3.39), we obtained boxplots, see Fig. 3.9, showing a smaller tracking error when $P = 2$ than that when $P = 1, 3$, for any combination of parameters. In particular, when $P = 3$ and $h = 4, 5, 6$, we observed several unstable results in 100 trials. Thus, we omitted the boxplots when $P = 3$ and $h = 4, 5, 6$. Next, for $P = 2$, a comparison of various combinations of parameters (n, m, h) is shown in Table 3.2. The results shown in Fig. 3.9 and Table 3.2 support the conclusions in the previous subsections.

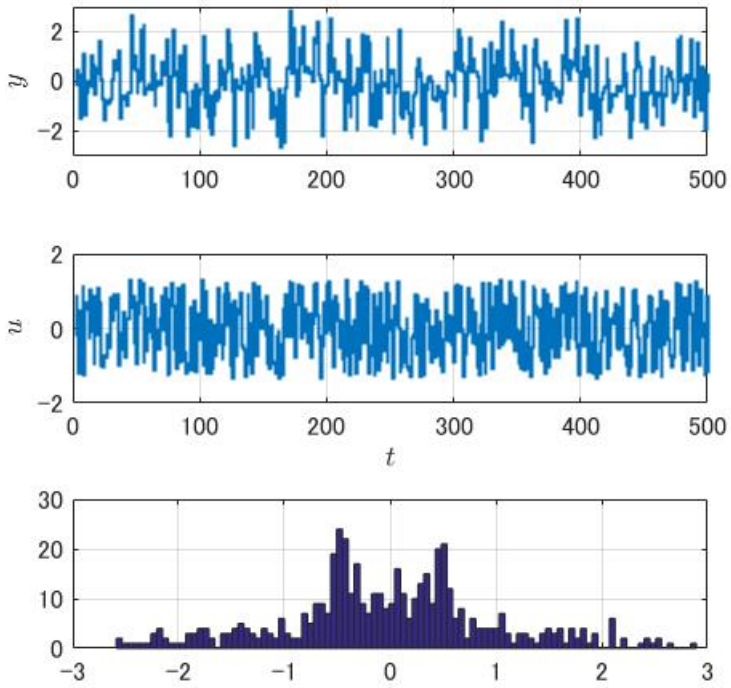


Figure 3.4: Stored datasets for model-free predictive control of nonlinear system (3.33) to obtain control results shown in Fig. 3.6. Histogram of values of output y in the dataset used to obtain control results in Fig. 3.6.

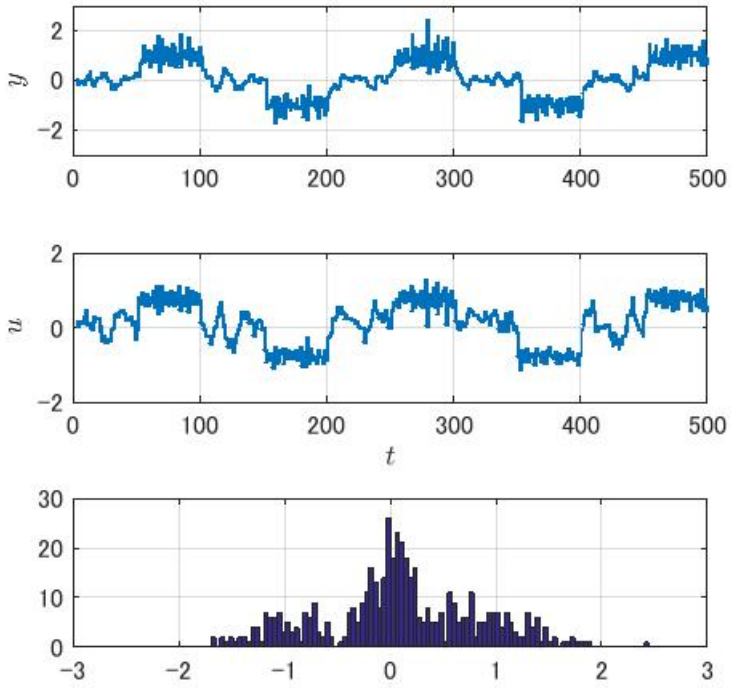
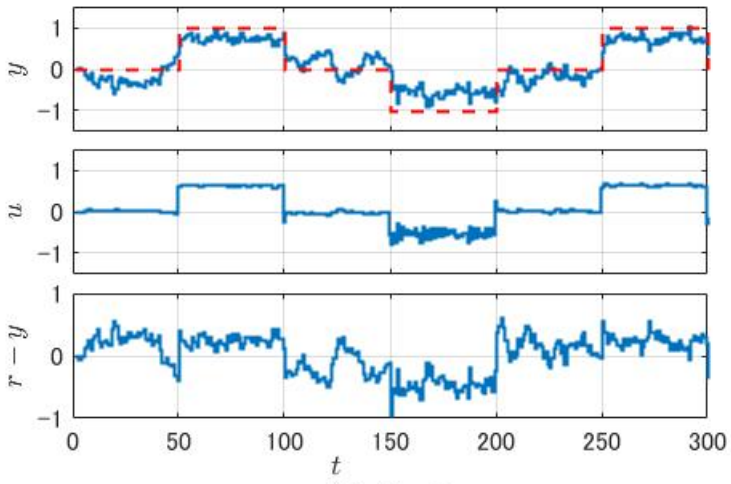
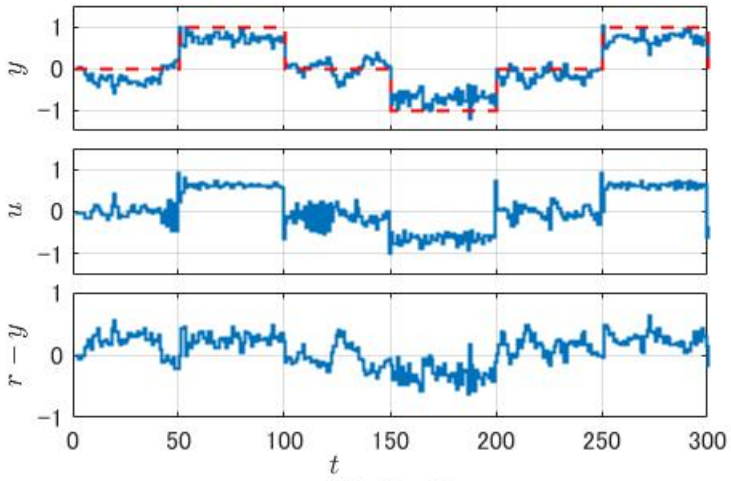


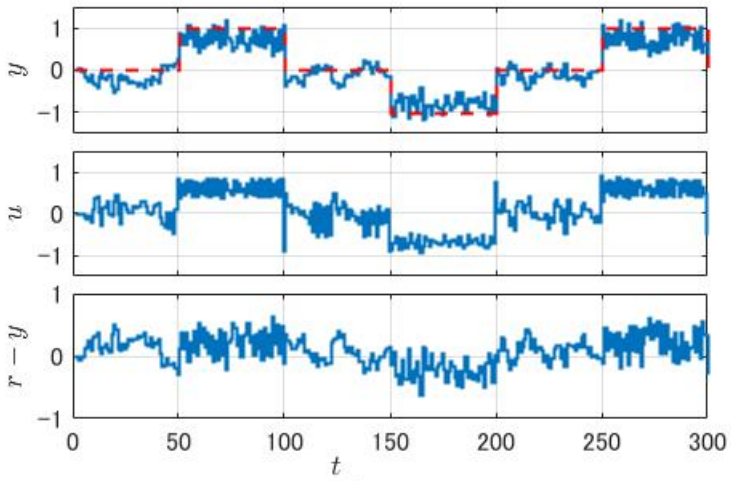
Figure 3.5: Stored datasets for model-free predictive control of nonlinear system (3.33) to obtain control results in Fig. 3.7. Histogram of values of output y in the dataset used to obtain control results in Fig. 3.7.



(a) $P = 1$

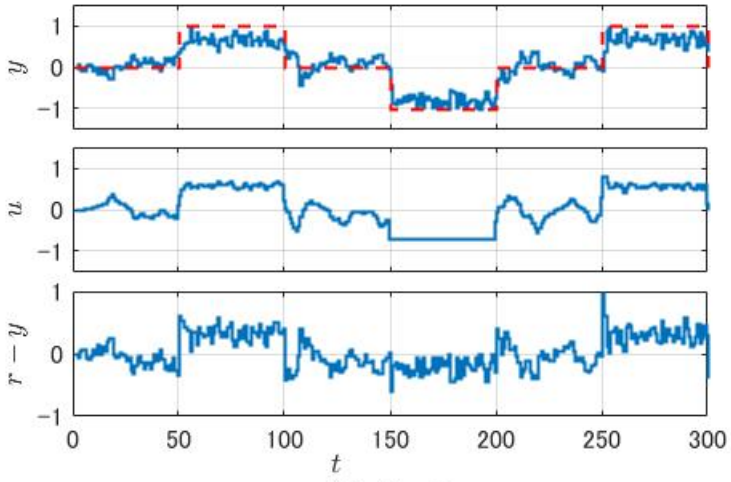


(b) $P = 2$

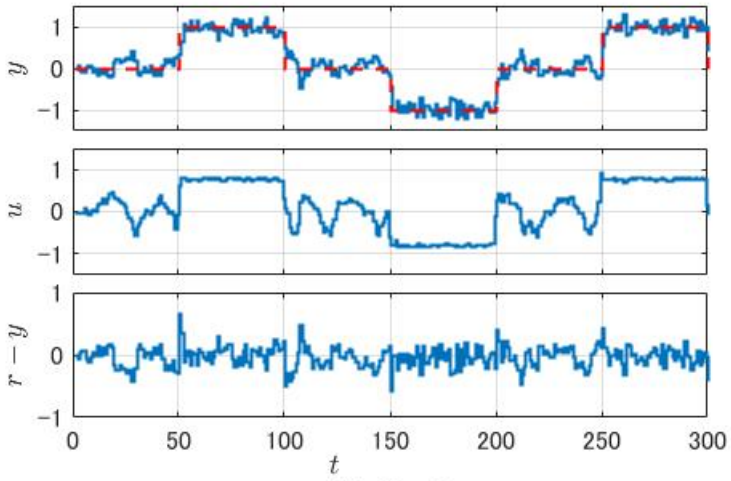


(c) $P = 3$

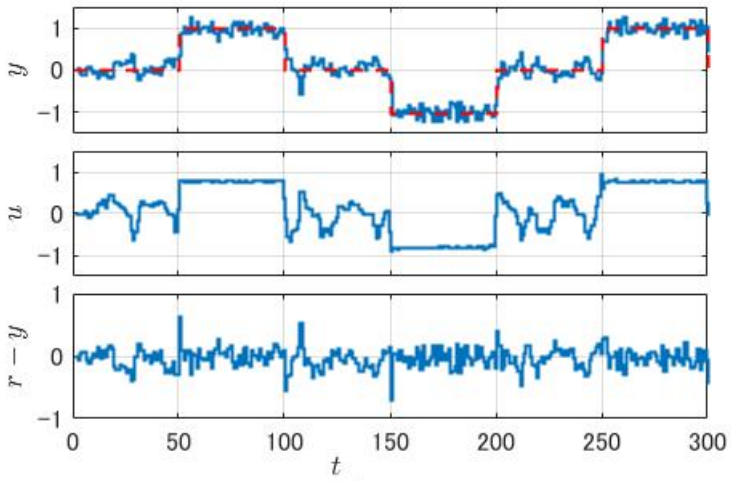
Figure 3.6: Simulation results of model-free predictive control for the nonlinear system (3.33) based on the datasets in Fig. 3.4 with $(n, m, h) = (1, 2, 1)$. (a) Order $P = 1$, (b) order $P = 2$, and (c) order $P = 3$.



(a) $P = 1$



(b) $P = 2$



(c) $P = 3$

Figure 3.7: Simulation results of model-free predictive control for nonlinear system (3.33) based on the datasets in Fig. 3.5 with $(n, m, h) = (1, 2, 1)$. (a) Order $P = 1$, (b) order $P = 2$, and (c) order $P = 3$.

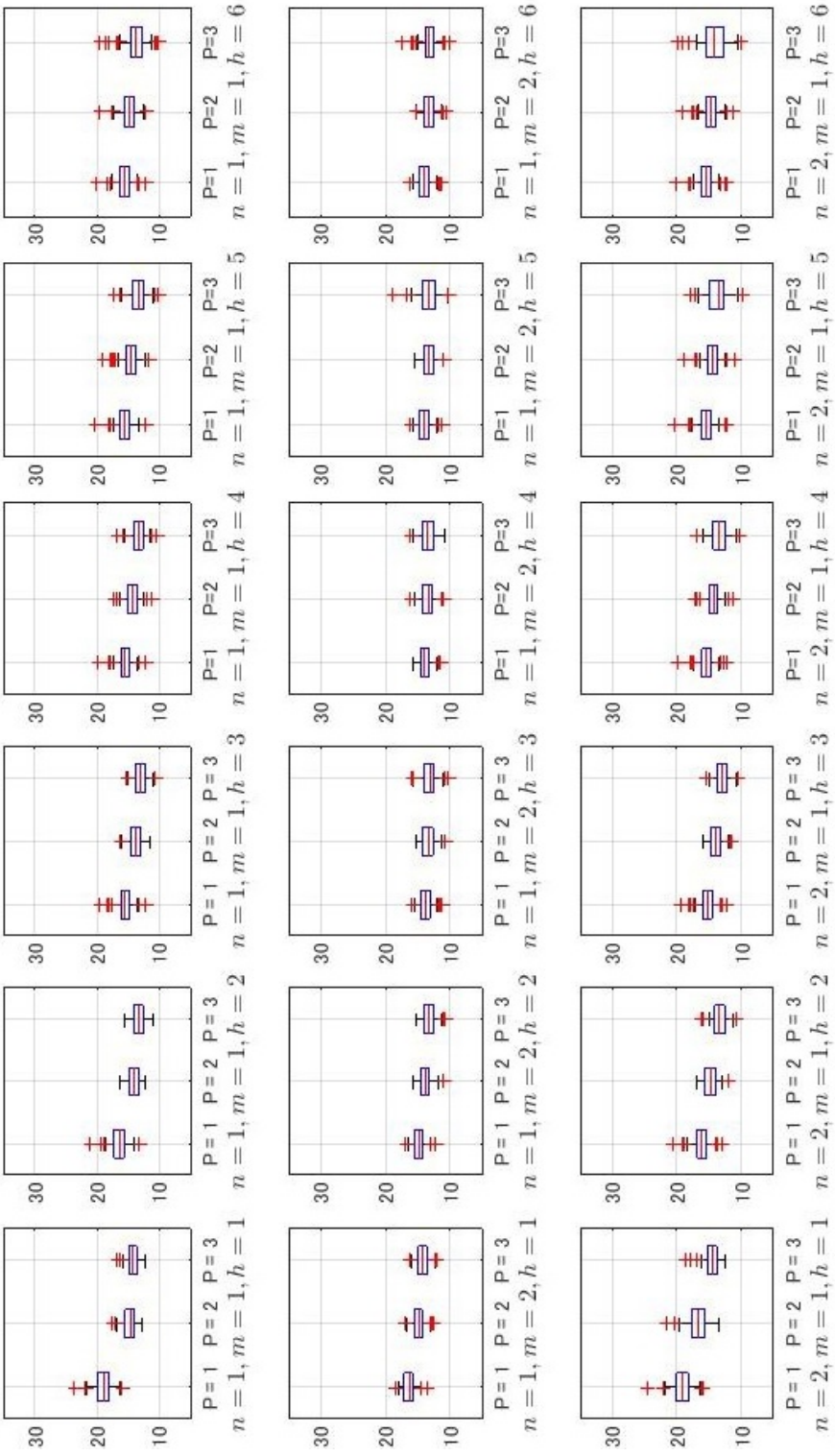


Figure 3.8: Boxplots when nonlinear system is given by (3.38).

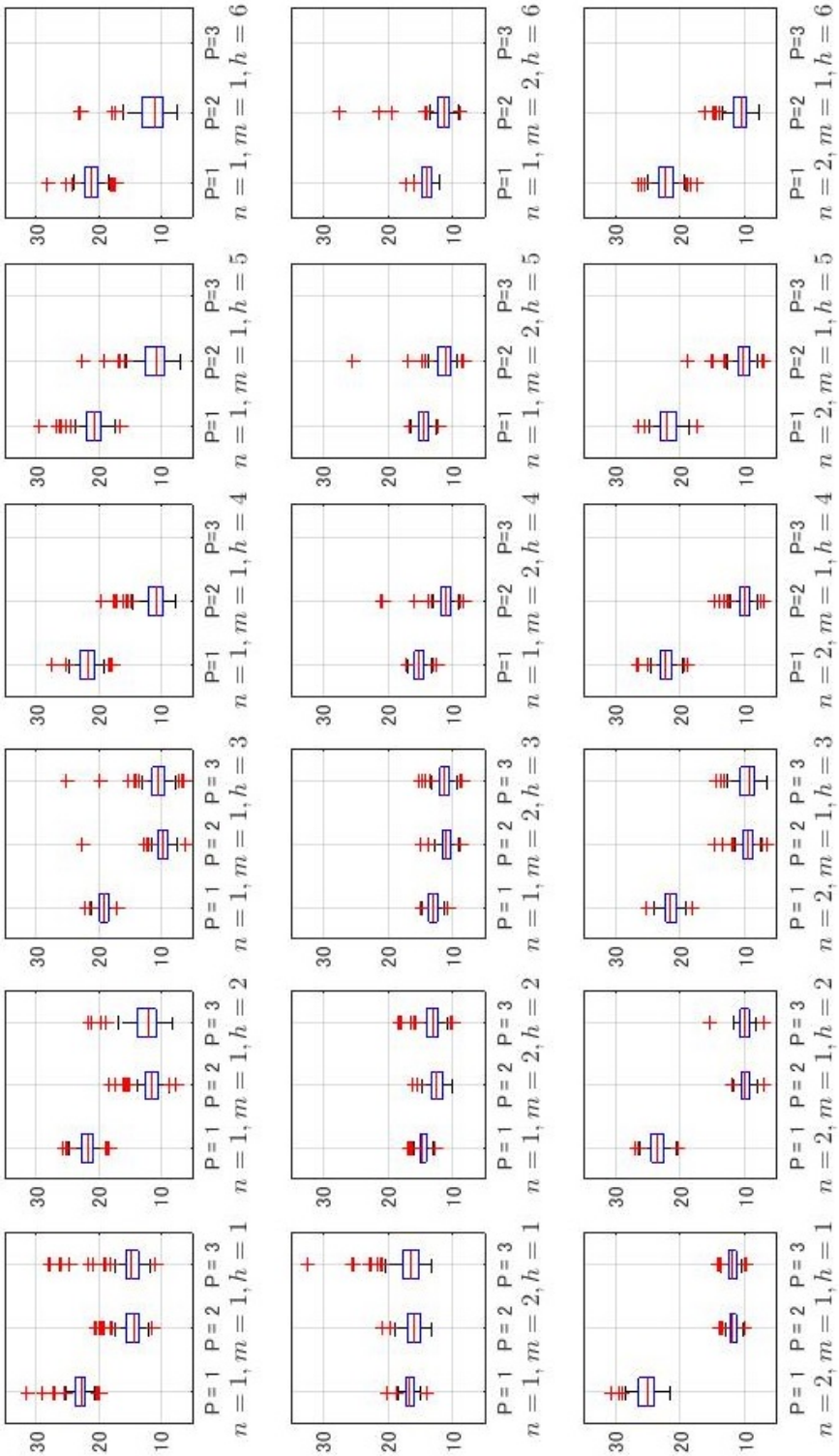


Figure 3.9: Boxplots when nonlinear system is (3.39).

Chapter 4

Application of Wastewater Treatment Process

The activated sludge process (ASP) is a typical biodegradation wastewater treatment process (WWTP). The ASP continuously maintains and circulates a stock of biological organisms. The WWTP principally comprises septic tanks, aerobic treatment systems, and sedimentation tanks. The aerobic treatment systems provide oxygen, controlling the substrate concentrations of aerobic organisms.

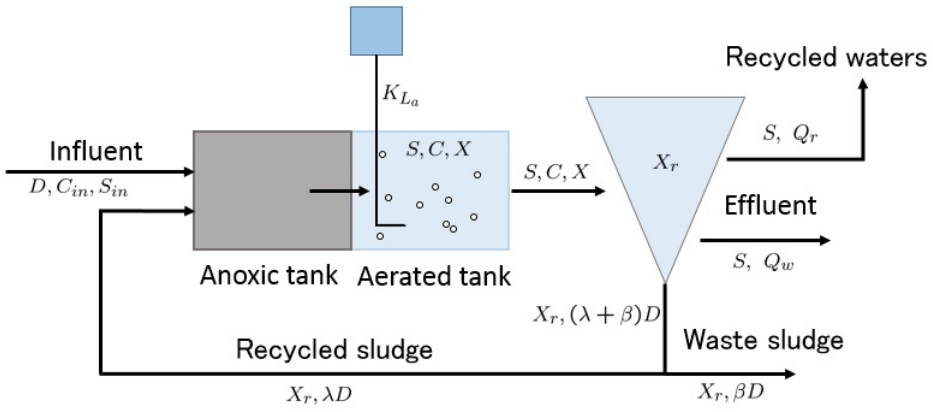


Figure 4.1: Wastewater Treatment Process

4.1 The Model

The continuous-time activated sludge model No. 1 (ASM1)[34]-[36]is expressed by

$$\begin{cases} \dfrac{\mathrm{d}S}{\mathrm{d}t} = & -\dfrac{1}{Y}\mu(t)X(t) + \{S_{\mathrm{in}} - (1+\lambda(t))S(t)\}D(t) \\ \dfrac{\mathrm{d}C}{\mathrm{d}t} = & -\dfrac{K_0}{Y}\mu(t)X(t) - (1+\lambda(t))C(t)D(t) \\ & +K_{L_a}(t)(C_{\mathrm{S}} - C(t)) + C_{\mathrm{in}}D(t) \\ \dfrac{\mathrm{d}X}{\mathrm{d}t} = & \mu(t)X(t) - \{(1+\lambda(t))X(t) - \lambda(t)X_r(t)\}D(t) \\ \dfrac{\mathrm{d}X_r}{\mathrm{d}t} = & \{(1+\lambda(t))X(t) - (\beta(t)+\lambda(t))X_r(t)\}D(t) \\ \mu(t) = & \mu_{\max}\dfrac{S(t)}{K_{\mathrm{S}}+S(t)}\dfrac{C(t)}{K_{\mathrm{C}}+C(t)} \\ \beta(t) = & \dfrac{Q_w}{Q(t)},\ \lambda(t) = \dfrac{Q_r}{Q(t)}, \end{cases} \tag{4.1}$$

where $t \in \mathbf{R}$ and the symbols are listed in Table 4.1. The control input and output are given by

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} D \\ K_{L_a} \end{bmatrix},\ y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} S \\ C \end{bmatrix}, \tag{4.2}$$

where the dilution rate D and the oxygen mass transfer coefficient K_{L_a} are

$$D(t) = \frac{Q(t)}{V},\ K_{L_a}(t) = \alpha W(t). \tag{4.3}$$

4.2 Simulation Results and Discussion

We applied MIMO model-free predictive control to the WWTP (4.1), discretized by Euler's method with the parameters and initial values given in Table 4.1. We set the sampling time at $T_s = 1$ hour in the simulations.

We prepared datasets containing the samples ($N = 300$) of inputs/outputs shown in Fig. 4.2, using random inputs u_1 and u_2 generated from uniform distributions $\mathbf{U}(0.01, 0.36)$ and $\mathbf{U}(0.05, 1.36]$). We used the datasets shown in Fig. 4.2 for model-free predictive control throughout the simulations. In addition, we set the reference trajectory to r_1 for the ammonium concentration $y_1 = S$ and to r_2 for the dissolved oxygen concentration $y_2 = C$ as shown in Fig. 4.3.

We set the parameters for model-free predictive control as $(m, n, h, P) = (1, 2, 1, 1)$ and $(m, n, h, P) = (1, 2, 1, 3)$. In the simulations, we used two measurement noises for y_1 and y_2. These were generated according to $\mathcal{N}(0, \sigma^2)$, where $\sigma^2 = 5 \times 10^{-4}$. We show the simulation results in Figs. 4.4 and 4.5. It can be seen that the output y tracks the reference trajectory r without a large overshoot, limiting the tracking error $r - y$ to ± 0.5mg/l. The desired concentration of ammonium is therefore achieved accurately in the effluent. The efficient removal of ammonium is also linked to the reconstructed dissolved oxygen concentration. The tracking error $r - y$ is lower when $P = 3$, as in Fig. 4.5, than when $P = 1$, as in Fig. 4.4. This suggests that the

Table 4.1: List of symbols and parameters ("$*$" denotes the initial value).

Symbol	Term	Value	Unit
S	substrate concentration of ammonium	19^*	[mg/l]
C	dissolved oxygen concentration	4.75^*	[mg/l]
X	biomass concentration	251.4^*	[mg/l]
X_r	recycled biomass concentration	503.98^*	[mg/l]
Q	influent volume	1.3^*	[m^3/h]
V	aerator volume	35	[m^3/h]
Q_r	recycled waste volume	0.78	[m^3/h]
Q_w	flowed waste volume	0.26	[m^3/h]
D	dilution rate Q/V	0.0389^*	[1/h]
W	aeration rate	48.2^*	[m^3/h]
β	flowed waste rate Q_w/Q	0.19^*	-
λ	recycled waste rate Q_r/Q	0.58^*	-
K_{L_a}	oxygen mass transfer coefficient αW	0.8684^*	-
Y	biomass yield factor	0.65	-
α	oxygen transfer rate	0.018	-
S_{in}	concentration of ammonium in the influent	200	[mg/l]
C_{S}	maximum concentration of dissolved oxygen	10	[mg/l]
C_{in}	concentration of dissolved oxygen	0.5	[mg/l]
K_{S}	saturation coefficient of ammonium S	100	[mg/l]
K_{C}	saturation coefficient of dissolved oxygen C	2	[mg/l]
K_0	model constant	0.64	-
μ	biomass growth rate	0.0173	[1/h]
$\mu_{\max}$	specific growth rate for autotrophs	0.15	[1/h]

introduction of polynomial regressor vectors can improve the control performance for ASM1 (4.1), even when the linear regressor vectors gives a satisfactory control result.

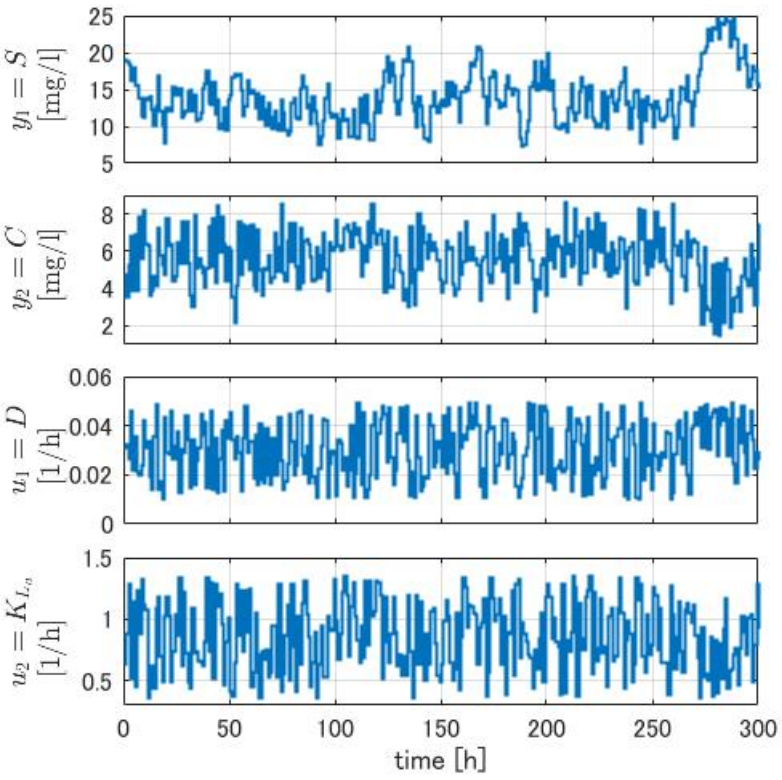


Figure 4.2: Stored measurement input/output dataset for model-free predictive control

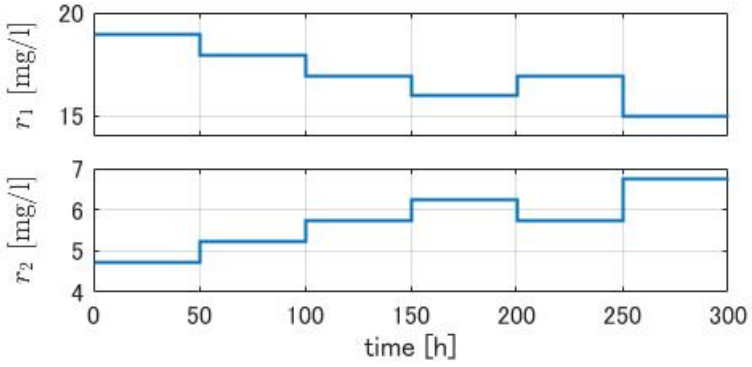


Figure 4.3: Reference inputs

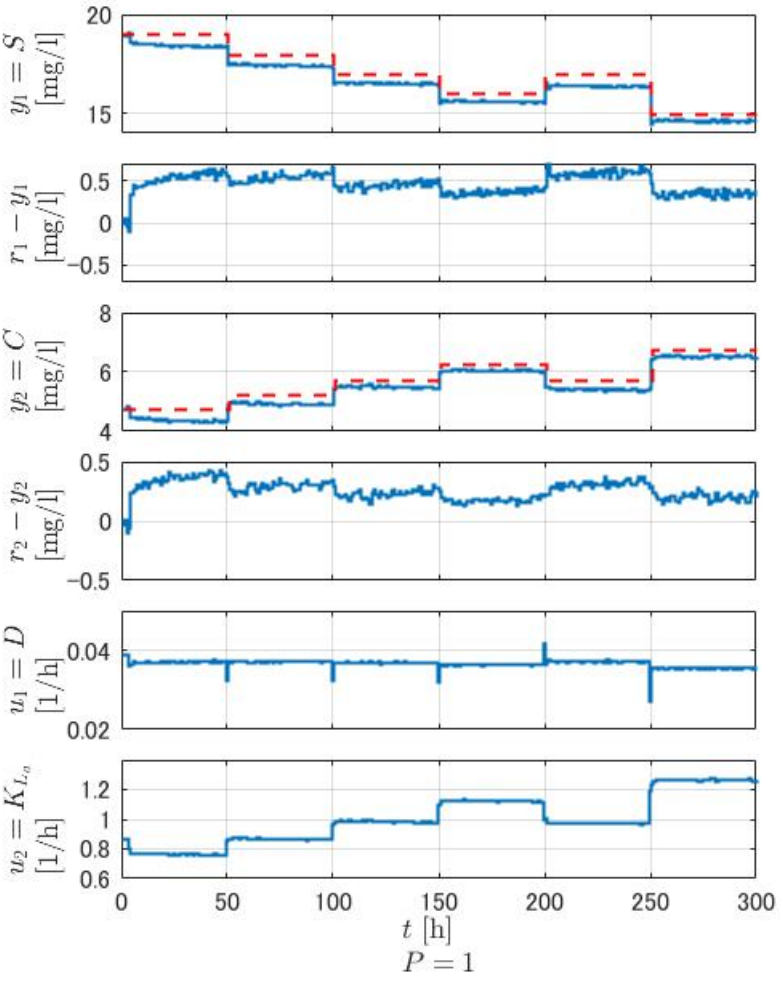


Figure 4.4: Simulation results for model-free predictive control when $P = 1$. Control starts at $t = 3$.

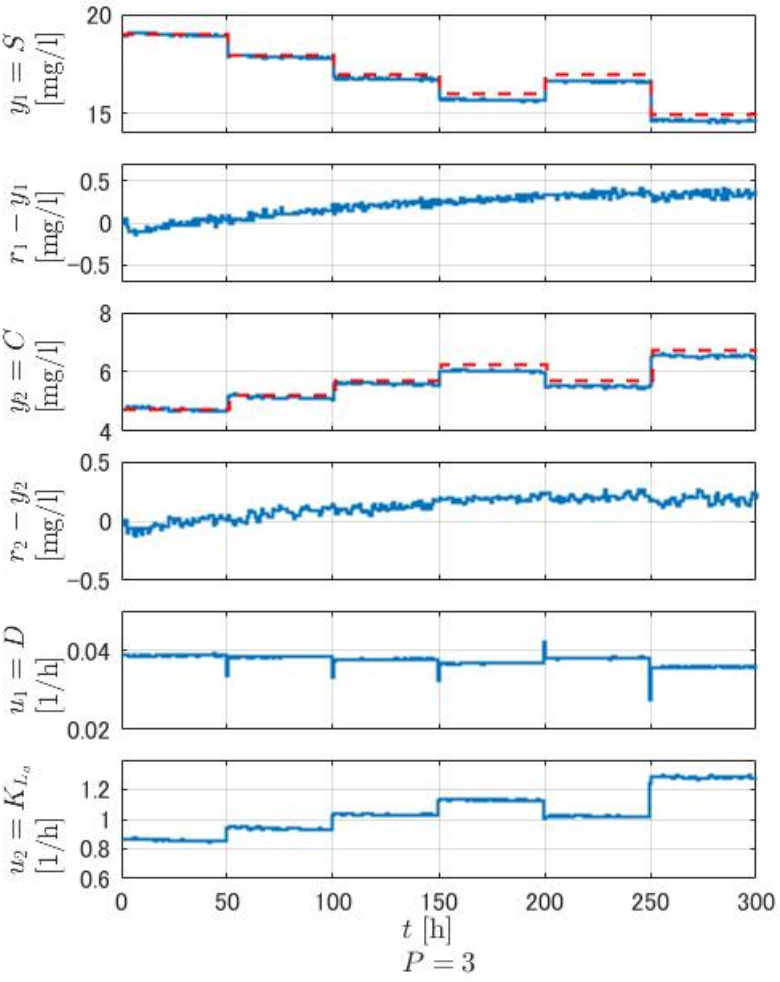


Figure 4.5: Simulation results for model-free predictive control when $P = 3$. Control starts at $t = 3$.

Chapter 5

Conclusions

In this thesis, we introduced polynomial regressors into the model-free predictive control and investigated their effectiveness. Using numerical simulations of nonlinear system we demonstrated that with an appropriate selection of the order of the polynomial regressors and the size of the polynomial regression vectors, a better control performance can be obtained compared to that obtained using the linear regression of existing model-free predictive control approaches. We also showed that rich datasets containing many output data y around the desired trajectory r must be recorded in advance to reduce the tracking error. We introduced a smoothing filter for the reference signal to prevent the instability arising in the closed loop system due to abrupt changes in the reference signal. In our future study, we will investigate the ways of selecting an appropriate combination of the parameters to improve the control performance.

In this study, we investigated the effectiveness of MIMO model-free predictive control using polynomial regressor vectors. Numerical simulations of a wastewater treatment process confirmed that satisfactory control can be achieved by applying MIMO model-free predictive control. In particular, the control performance can be improved by replacing linear regressor vectors with polynomial regressor vectors. In general, ℓ_1-minimization in model-free predictive control imposes a heavy computational burden. Hence, nonlinear systems with slow dynamics, such as the wastewater treatment process, are promising applications for model-free predictive control.

Publications

Journal papers

[1] Hongran Li and Shigeru Yamamoto: Model-Free Predictive Control Using Polynomial Regressors, Accepted in JCMSI

Conference papers

[1] Hongran Li and Shigeru Yamamoto: A Model-Free Predictive Control Method based on Polynomial Regression; SICE International Symposium on Control Systems 2016, 3A2-4, Nagoya, March 9, 2016.

[2]Hongran Li and Shigeru Yamamoto: Polynomial Regression Based Model-Free Predictive Control for Nonlinear Systems; SICE 2016, pp. 578-582, Tsukuba, September 21, 2016.

Bibliography

[1] 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.

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

[3] D. Inoue and S. Yamamoto, "Support for Drivers via Just-in-Time Predictive Control and Fault Detection Based on a Nearest Neighbor Method during Braking to Stop Trains," *Transactions of the Japan Society of Mechanical Engineers. C*, Vol. 72, pp. 2756–2761, 2006.

[4] K. Fukuda, S. Ushida, and K. Deguchi, "Just-in-Time Control of Image-Based Inverted Pendulum Systems with a Time-Delay," *Proc. SICE-ICASE 2006*, pp. 4016–4021, 2006.

[5] E. Konaka, "Design of Discrete Predictive Controller Using Approximate Nearest Neighbor Method," *Proc. the 18th IFAC World Congress*, pp. 10213–10218, 2011.

[6] N. Nakpong and S. Yamamoto, "Just-in-Time Predictive Control for a Two-Wheeled Robot," *Proc. 2012 Tenth Int. Conf. on ICT and Knowledge Engineering*, No. 3, pp. 95–98, 2012.

[7] T. Kosaki and M. Sano, "Networked Just-in-Time Control of a Parallel Mechanism with Pneumatic Linear Drives," *Communication in Control Science and Engineering (CCSE)*, Vol. 2, pp. 19–25, 2014.

[8] S. Yamamoto, "A New Model-Free Predictive Control Method Using Input and Output Data," *3rd International Conference on Key Engineering Materials and Computer Science (KEMCS 2014)* (Advanced Materials Research Vol.1042, ed.), pp. 182–187, Trans Tech Publication, 2014.

[9] S. Yamamoto, "A Model-Free Predictive Control Method by l_1-minimization," *Proc. of the 10th Asian Control Conference 2015*, 1570064495, 2015.

[10] H. Saputra and S. Yamamoto, "Comparative Study of Model-Free Predictive Control and its Database Maintenance for Unstable Systems," *SICE Journal of Control, Measurement, and System Integration*, Vol. 8, No. 6, pp. 390–395, 2015.

[11] G. Cybenko, "Just-in-Time Learning and Estimation," *NATO ASI Series of Computer and Systems Sciences*, Vol. 153, pp. 423–434, 1996.

[12] A. Stenman, F. Gustafsson, and L. Ljung, "Just in Time Models for Dynamical Systems," *Proc. of the 35th IEEE Conference* on Decision and Control, Vol. 1, pp. 1115–1120, 1996.

[13] A. Stenman, A. V. Nazin, and F. Gustafsson, "Asymptotic Properties of Just-in-Time Models," *Proc. of the 11th IFAC Symposium on System Identification*, pp. 1249–1254, 1997.

[14] A. Stenman, "Model on Demand: Algorithms, Analysis and Applications," PhD thesis, Department of Electrical Engineering Linkoping University, 1999.

[15] M. W. Braun, D. E. Rivera, and A. Stenman, "A Model-on-Demand Identification Methodology for Nonlinear Process Systems," *International Journal of Control*, Vol. 74, No. 18, pp. 1708–1717, 2001.

[16] G. Bontempi, M. Birattari, and H. Bersini, "Lazy Learning for Local Modelling and Control Design," *International Journal of Control*, Vol. 72, No. 7-8, pp. 643–658, 1999.

[17] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-Based Learning Algorithms," *Machine Learning*, Vol. 6, No. 1, pp. 37–66, 1991.

[18] Q. Zheng and H. Kimura, "A New Just-in-Time Modeling Method and its Application to Rolling Set-Up Modeling," *Trans. of the Society of Instrument and Control Engineers*, Vol. 37, No. 2, pp. 640–646, 2001 (in Japanese).

[19] Q. Zheng and H. Kimura, "Just-in-Time Modeling for Function Prediction and its Application," *Asian Journal of Control*, Vol. 3, No. 1, pp. 35–44, 2001.

[20] M. Kishi, K. Kimura, J. Ohta and S. Yamamoto, "Shrinkage Prediction of a Steel Production via Model-on-Demand," *Proc. of the 11th IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, pp. 447–450, 2004.

[21] H. Shigemori, M. Kano, and S. Hasebe, "Optimum Quality Design System for Steel Products Through Locally Weighted Regression Model," *Journal of Process Control*, Vol. 21, No. 2, pp. 293–301, 2011.

[22] J. Ohta and S. Yamamoto, "Database-Driven Tuning of PID Controllers," *Trans. of the Society of Instrument and Control Engineers*, Vol. 40, pp. 664–669, 2004 (in Japanese).

[23] T. Yamamoto, K. Takao, and T. Yamada, "Design of a Data-Driven PID Controller," *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 1, pp. 29–39, 2009.

[24] K. Fujiwara, M. Kano, S. Hasebe and A. Takinami, "Soft-Sensor Development Using Correlation-Based Just-in-Time Modeling," *AIChE Journal*, Vol. 55, No. 7, pp. 1754–1765, 2009.

[25] A. Stenman, "Model-free Predictive Control," *Proc. of the 38th IEEE Conference on Decision and Control*, Vol. 4, pp. 3712–3717, 1999.

[26] M. O. Franz and B. Scholkopf, "A Unifying View of Wiener and Volterra Theory and Polynomial Kernel Regression," *Neural Computation*, Vol. 18, No. 12, pp. 3097–3118, 2006.

[27] P. Alper, "A Consideration of the Discrete Volterra Series," *IEEE Transactions on Automatic Control*, Vol. 10, No. 3, pp. 322–327, 1965.

[28] H. Li and S. Yamamoto, "A Model-Free Predictive Control Method Based on Polynomial Regression," *Proc. SICE International Symposium on Control Systems 2016*, 3A2-4, 2016.

[29] H. Li and S. Yamamoto, "Polynomial Regression Based Model-Free Predictive Control for Nonlinear Systems," *Proc. SICE 2016*, pp. 578–582, September, 2016.

[30] A. Y. Yang, Z. Zhou, A. Ganesh, S. S. Sastry, and Y. Ma, "Fast ℓ_1-Minimization Algorithms For Robust Face Recognition," *IEEE Transactions on Image Processing*, Vol. 22, No. 8, pp. 3234–3246, 2013.

[31] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab Software for Disciplined Convex Programming," cvxr.com, 2008.

[32] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, pp. 4–27, 1990.

[33] D. W. Clarke, "Self-Tuning Control of Nonminimum-Phase Systems," *Automatica*, Vol. 20, No. 5, pp. 501–517, 1984.

[34] C. Vlad, M. Sbarciog, M. Barbu, S. Caramanet, and A. V. Wouweral, "Indirect Control of Substrate Concentration for a Wastewater Treatment Process by Dissolved Oxygen Tracking," *Journal of Control Engineering and Applied Informatics*, Vol. 14, No. 1, pp. 38–47, 2012.

[35] M. Barbu and S. Caraman, "QFT Multivariable Control of A Biological Wastewater Treatment Process Using ASM1 Model," *IFAC Proceedings Volumes (10th IFAC Symposium on Computer Applications in Biotechnology)*, Vol. 40, No. 4, pp. 295–300, 2007.

[36] F. Nejjari, B. Dahhou, and A. Benhammou, "Nonlinear Multivariable Adaptive Control of an Activated Sludge Wastewater Treatment Process," *International Journal of Adaptive Control and Signal Processing*, Vol. 13, No. 4, pp. 347–365, 1999.