A Study on Data-Driven Predictive Control

DISSERTATION

A Study on Data-Driven Predictive Control

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

Intelligent Systems and Information Mathematics

Student Number: 1123112105

Herlambang Saputra

Chief advisor :
Shigeru YAMAMOTO, Prof. Dr
July, 2015

Abstract

This study provides a comparison of three methods, i.e., standard locally weighted averaging (LWA), least-norm solutions, and ℓ_1-minimization, for model-free predictive control based on Just-In-Time modeling and database maintenance for an unstable system. In contrast to conventional model predictive control, the model-free predictive control method does not use any mathematical model; rather, it uses the past input/output data stored in a database. Although conventional stabilizing feedback is used to obtain the input/output data of an unstable system, model-free predictive control is assumed to be used without it. Three methods based on standard LWA, least-norm solutions, and ℓ_1-minimization are statistically compared using a simple model. The results show that the methods of least-norm solutions and ℓ_1-minimization are superior to that of LWA. The method by ℓ_1-minimization yields tracking errors smaller than that by least-norm solutions; however, the method by ℓ_1-minimization requires a long computational time. In addition, the effectiveness of a method of database maintenance is illustrated by numerical simulations.

Contents

List of Figures

Acknowledgement

First of all, I would like to say thank you very much for every thing that I got from the lab of Prof. Shigeru YAMAMOTO. There are plenty of experience that I got from the lab. I also want to say thank you with Associate Prof. Osamu KANEKO for his support, knowledge and everything that he gave to me. For thesis committee Prof. Miyoshi, Prof. Yamane, and Prof. Kenji Satou thank you for suggestion, comment and input for the dissertation.

Second, I want to say thank you for my parents, Karsiman and Erna Purwantini for support in my life, and R.I.P for my father Karsiman. I also say thanks to my parents in law Hamdi Som and Ermawati to give me a lot of thing for study. A special thank for my wife Alvi Syahrini Utami and My daughters Clairine Falisha Imtiyaz, Syifa Artanti Hisanah and Callia Natsumi Akhsania. My younger sister Dian Kusumaningrum and younger brother Fajar Wijaya Putra thanks.

Third, for all member MoCCoS laboratory, Mr. Mohd Syakirin and Mrs. Dessy Novita thank you for sharing knowledge. For my tutor Mr. Yuki Hayashi and Mr. Fumiaki Uozumi thanks for helping me in daily life. All of my friends in PPI Ishikawa especially DIKTI 2011 and my friend in Indonesia thank you very much.

I also thank to my government DIKTI for give me time and scholarship to study until finish and also State Polytechnic of Sriwijaya for the support and contribution.

Last but not least, I believe there is a very strong power that control and create everything in this Universe and the power is belong to my God ALLAH S.W.T. We are just a human and we have a lot of limitation, thank you ALLAH S.W.T.

Herlambang Saputra

Kanazawa, June 2015

Chapter 1

Introduction

In the real world, plenty of systems have insufficient information about the way to control. To design a control method, the information is important. When we do not have enough information it is like driving a car in the desert without help of a map. The purpose of the map is a guidance for the driver to go to the point of achievement with a little mistake. It is not easy to get information, one way is to make several preliminary experiment and see the reaction of the system after that make a temporary conclusion to define the next condition. Based on temporary conclusion, the control method built and the duty of controller is to produce the control input. Figure.1.1 shows the mechanism to treat the disturbance d and the desired output r to the measured output y :

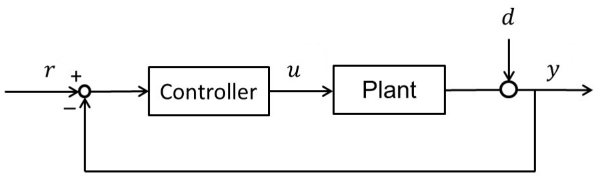


Figure 1.1: Overview control of a system

An error between the reference r and the output y should be minimized when the controller is used to control a system (plant). In this study, the reference signal r is included in calculation to achieve the best output with minimum error. In optimization problem in control world, the cost function to be minimized sometime is not only one. For example, there are two variables that have to be minimized error and computation time. The lowest error probably makes the computation time increase whereas fast computation time can make a big error. The mechanism of the optimize solution choose the best available error not always the lowest error but acceptable for the system and fast solution.

Another approach in control theory is data-driven theory. The data-driven theory [1] is designed to use directly using input/output data with on-line or off-line mechanism, the data is taken as a knowledge without any mathematical models or implicit and explicit information. Figure.1.2 shows the mechanism in general the data flow of the data-driven system.

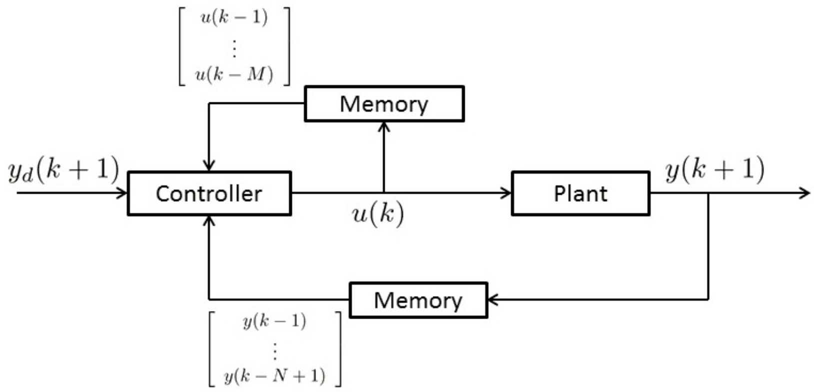


Figure 1.2: General overview of data-driven method

In Fig.1.2 M and N are length of data. Furthermore $y_d(k)$, $y(k)$, $u(k)$ are the desired output r, output, and the input respectively. All I/O data are stored in a database and then can be used again in control.

There are many methods in the frame of the data-driven theory to obtain optimal control input u based on such as k-Nearest Neighborhood, LWA (Locally Weighted Average), Least-norm solution, and ℓ_1-norm optimization. In this research, we investigate three methods LWA, least-norm and the ℓ_1-norm to optimize the control input u. The objective is to obtain the best output y.

Main idea of this research is to understand control performance use model-free predictive control or Just-in-Time (JIT) predictive. Model predictive control commonly is used in chemical industry [2]. On the other hand, and the method proposed by Stenman in 1999 [3] updates the mathematical model constantly based on input and output data, and Just-In-Time modelling. The data store in a database [4], [5] a sufficient data [6] same like model on-demand [7], lazy learning [8], or instance based learning [9]. The Just-In-Time technique is applied for prediction in steel industry [10], [11], [12], [13], PID parameter tuning [14], [15], a soft sensor in industrial chemical processes [16], etc.

Inoue and Yamamoto [17] introduced a model, it is called "model free" for predictive control in just-in-time modelling. The method directly predicted an optimal input without using any local models, but using online measured data and also stored prior data. The Just-In-Time method searches the neighbors of the current data from the stored data or database and weight average is used to predict a future behaviour system. Although for this weighted average, several methods are applicable in the Just-In-Time modeling framework, control performance highly depends on the weights. Similar Just-In-Time control methods are also proposed in [18], [19], [20], etc. They also use the nearest neighbor and LWA (Locally Weighted Average) technique as in the Just-In-Time modeling method.

Recently, two approaches substituting the conventional the nearest neighbor and LWA technique have been introduced [21], [22]. In a previous study [21], weights are calculated as a solution of a linear equation. In another previous study [22], weights are computed as a solution of an ℓ_1-minimization problem which produces a sparse vector with a few nonzero elements. This type of ℓ_1-minimization is currently popular in signal processing community [23], because sparse solutions yield benefit. In control community, sparsity is also utilized in efficient data compression for control signals through rate-limited erasure channels [24]. In [22], sparsity is utilized to find the nearest neighbor and the weights.

The focus of this paper is to compare of three methods ([17], [21], and [22]) by applying them to control of an unstable system. Stabilization by model free predictive control is still an open problem. Asymptotic stabilization seems to be impossible except for an ideal case where there is no noise and nonlinearity. The boundedness of all signals in the control system will only be guaranteed in practical applications. In this paper, we statistically evaluate the effect by model free predictive control through many trials. In the case of an unstable system, it is difficult to construct a rich database containing input/output data without feedback control. Hence, to construct a database, we assume that there exists simple feedback control that stabilizes the unstable system. In [18] and [25], data-driven is used to improve stabilizing feedback control. In contrast to these, since our main aim is to show that model-free predictive control has the ability to stabilized unstable systems, when we use model-free predictive control, we do not use the stabilizing controller, unlike in [18] and [25]. In addition, we investigate the effect of database maintenance. In this paper, as a method of database maintenance, we propose that least accessed data in the database is replaced with the most current data which was obtained online. Replacing is done to prevent the size of the database increasing.

1.1 Motivations and Objectives

This research in concern with a algorithm computing control a small tracking error. The evaluation criteria of the algorithm is low memory usage, fast computation time and a small tracking error. Model free prediction or JIT optimization is one of promising solution and model-free predictive control method does not use any mathematical models but the past input/output from currently time the data must be stored in a database. The JIT contribution is as an estimator to approach an estimate of nonlinear estimator base on observed the data from prior process. The application for JIT can be used in real world, for instance in chemical industrial to process the liquid, etc. JIT is used as the data-driven for the methods to follow the reference.

The aim of this research is to measure the capability of the three methods : LWA, least-norm, and ℓ_1-norm. There are two evaluate criteria error and computation time. The goal of the methods in running process is to get the vector x or weight factors with several methods. One of the method is ℓ_1-norm optimization, the method produces a sparse vector x which contains a few zero non zero element. For example, the vector $\in \Re^{1000}$ with 20 element non-zero elements can be considered as a sparse vector.

In this research, the database can be updated and the process to update the database is called database maintenance. Another objective is to see the influence of maintenance the database to achieve a better result.The database maintenance in this case use a constant value to remove an unimportant value for the next process. The goal is to make a perfect database to get input to the system.

1.2 Overview of Previous and Related Researches

1.2.1 Just-in-Time Models with Application to Dynamical System

The common of method 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 [6], Just-in-time estimator used to make the weighted matrix as the neighborhood data. The weights are made to be optimized to minimize MSE (Mean Square Error). The size of the data to make database is determined by an operator. The order is significant to order the dimension of matrix space in programming language. He said that the convergence rate from JIT estimator give a consistent estimates as a function for same case in kernel method. There are two essential step in the concept. Firstly, consider about nonlinear identification in time domain which can predict the output from nonlinear system based on data sets of prior input and output in dynamical system. Secondly, in frequency domain identification occurs when a linear system is faced for estimating the problem of the frequency response function. The benefit of JIT is the optimized locally, the locally method compared with the global that make the performance increase. The complexity is in the computational programming code, the reason is to search the neighborhood in multidimensional space and estimator derivation in term of computational effort.

1.2.2 Just-In-Time Predictive Control for a Two-wheeled Robot

JIT is also used in Nakpong and Yamamoto [25], to control Two-Wheeled robot. The robot moves with two wheels as an invert pendulum to maintain the standing position. In the method, they made a database as a data-driven to control the robot. The database was collected with preliminary experiment and the data are the angle θ, ϕ and also a random data of input between -0.5 until 0.5. The preliminary experiment took $N = 5000$ sample data and then stack the data into one vector. The vector arrange in time of t to be a single matrix A and compare with information vector to find a similar phenomena in database. The JIT predictive control calculate in computer and then send the controller output to the robot. The communication between JIT predictive control and the robot use WiFi connection to send and receive the data, Fig.1.3 give an illustration about the communication data. The result of the research can make the robot stable by state feedback in advance.

1.2.3 Design of Discrete Predictive Controller Using Approximate Nearest Neighbor Method

In Konaka [19], a method is proposed to the neighbor as a solution of control input u. There are several steps to get the controller. Firstly, a training data is collected randomly from five limited input $u$to get output y and then normalized all of vector.

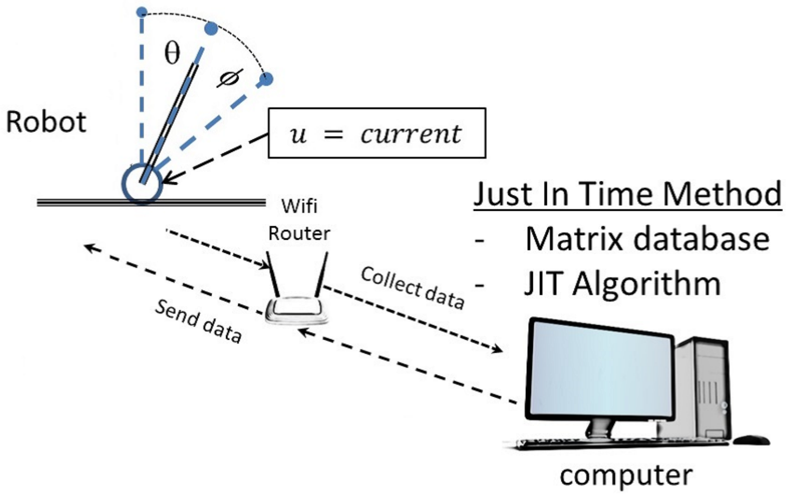


Figure 1.3: Communication data between robot and computer

The *locality sensitive hashing* (LSH) is used to classify the data 1.1.

$$h : \Re^d \rightarrow Z, h_{a,b}(v) = \left\lfloor \frac{a^T v + b}{r} \right\rfloor \tag{1.1}$$

where $r > 0$ is a hash function parameter, $a \in \Re^d$ is chosen independently from normal distribution, b is a scalar from uniform distribution and $\lfloor x \rfloor$ is a floor function. Secondly, an information vector is made by realtime process.

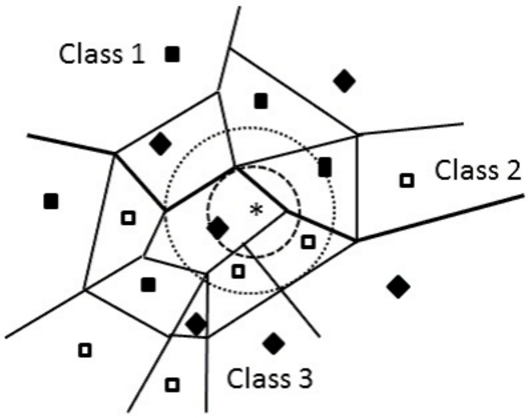


Figure 1.4: Example of *k*-nearest neighbor classification

The initial k in Fig.1.4 indicates how many data that are taken near the sample data and classification means the selection input. In Fig.1.4, there are three class and the nearest data is belong to the third class.

In each time of k the data selected of the neighbor from training data, for instance in $k = 10$ after normalized and calculated LSH value, the next step calculate the distance as an indicator of nearest data to get the nearest data. For this case, the data that have same hash value first from training data are calculated and the goal is to reduce the computation time. The solution is the index i as the nearest data and the apply $u(k)$ to the system after that move to next time $(k + 1)$ and the input/ouput from the process are stored.

1.3 Outline of The Dissertation

This dissertation is organized as follows.

Chapter 2 describes model-free predictive control and a few support system for this research.

Chapter 3 explains three kinds of control algorithms and also the algorithm steps of model free predictive control.

Chapter 4 represents simulation results and discussion. In the simulation there are graphs that indicate the results of the method.

In Chapter 5, we state conclusions that summarizes the research and a plan for future works.

Chapter 2

Model-Free Predictive Control

In real life, predictive control commonly use in petrochemical industry, the influence of the theory has made impact the industry significantly. Most of the system have a constrain and the constrain could be come from internal and external of the system. A good controller method must be consider all of constrain when control a system. The useful to calculate the constraints are for success in cost prediction. In manufacturing process, the low cost production and fast in process are needed for competition in world of industry to give a maximum profit for a company. The constrain [26] could be appear in input, or in variable which could be manipulated. There are many kind of constrains for instance the adjustment of valve, the maximum or minimum input value in control for dynamic system, the limitation of the system performance, etc. The circumstances are about the constrain commonly exist, such as :

(i). A constrain occur at physical system usually called physical constrain. eg. temperature, overshoot, etc.

(ii). In optimization theory, the results are often near the exact solution.

(iii). Addresses of constrain in most of control method make it as a posteriori.

The illustration of constrain in horizon can be seen as Fig.2.1. In the figure, there are a few feature : a reference signal is a goal for the output to follow with small error, k is a discrete time and a constrain value. One of predictive control capability is to handle the constrain, systematically. A predictive control is used to predict the behaviour of the system starting from current time k until future prediction time $(k + l)$ and l step a head could be different for input and output.

2.1 General form of the state space model

A state space model for predictive control has been discussed in many book. In [26], the author assume about a linearized, discrete-time and state space model of

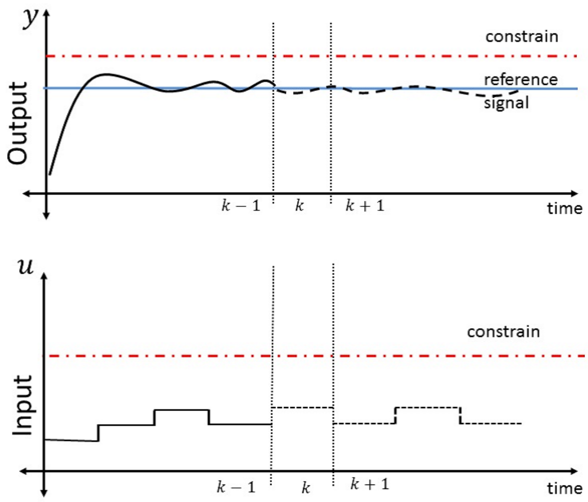


Figure 2.1: Receding horizon of predictive control with constrain

the system can be seen as :

$$\begin{aligned} x(k+1) &= Ax(k) + u(k) \\ y(k) &= C_y x(k) \\ z(k) &= C_z x(k) \end{aligned} \tag{2.1}$$

where x, n, u, l, and y are an n-dimensional state vector, an l-dimensional input vector, an m_y-dimensional vector of measured outputs, and an m_z-dimensional vector of outputs, respectively. The overlap sometime could be occur in large case for variables y and z. The assumption which often use $y \equiv z$, the C is used to denote C_y and C_z, m_y and m_z are denoted by m, and the k index is time steps.

The standard form is used because it connects with the standard theory of linear system and control. There are a few assumption at time k :

(i). To get of $y(k)$ value.

(ii). The input for the plant $u(k)$ is computed.

(iii). Use the input $u(k)$ to the plant, the aim is to get the output predictive $y(k+l)$.

Based on the assumption, the delay of time occur between measuring $y(k)$ and applying $u(k)$. So, the measurement of $y(k)$ from $u(k)$ cannot calculate directly, like (2.1). The structure of predictive control can be seen in Fig.2.2.

The variable of $z(k)$ in principle is depends on variable $u(k)$ to calculate the output of controller and the description of the $z(k)$ is as follow :

$$z(k) = C_z x(k) + D_z u(k) \tag{2.2}$$

in some case the non zero of D_z could be useful. A new vector is designed to avoid losing data at calculation when control the output.

$$\tilde{z}(k) = z(k) - D_z u(k) \tag{2.3}$$

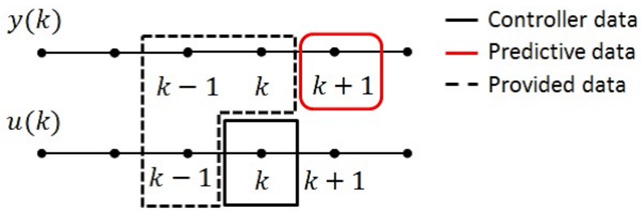


Figure 2.2: Example of predictive control

2.2 The Sparse Vector for ℓ_1-norm Optimization

This paper use a sparseness vector, which has contain element zero more than element non zero inside the vector. This theory is used to reduce the process of calculation and there are two kind of sparseness row sparse and column sparse. In this research, we use column sparse and the goal of this research is to get a simple programming code, low error and fast in computation with sparseness vector. Another benefit of sparseness is to reduce a memory space in computer. Consider there is a problem solution :

$$Ax = b \tag{2.4}$$

where a matrix $A \in \Re^{m \times n}$, vector $b \in \Re^{m}$ and find a vector $x \in \Re^{n}$. The optimize solution is vector x and the vector of x has a form like Fig.2.3.

$$\overset{A}{\begin{bmatrix} a_1 & a_2 & \cdots & \cdots \end{bmatrix}} \overset{x}{\begin{bmatrix} 0 \\ * \\ 0 \\ * \\ 0 \\ 0 \\ * \\ 0 \end{bmatrix}} = b$$

Figure 2.3: Example of sparseness

Commonly, the vector x in Fig.2.3 is called sparse vector where * is a non zero element, a_i and b are a vector. Obviously, the advantage of sparseness can be known by the structure of the vector. The method fast in calculation and lack of memory binary code in store process.

A conventional method to solve $Ax = b$ is *linear least-square* or *linear least-norm* and to solve the problem l_2-norm is used for the optimal solution. In this paper there are a few approach to solve the optimization problem to solve JIT case and of the is ℓ_1-norm solution. The ℓ_1-norm can be used to check if the vector x is sparse and the definition of ℓ_1-norm is :

$$||x||_1 = |x_1| + |x_2| + ... + |x_N| \tag{2.5}$$

where N is the length of the x and the ℓ_1-norm optimization [23] is :

$$\begin{array}{cc} \min & ||x||_1 \\ \text{subject to} & Ax = b \end{array} \tag{2.6}$$

where $C \in \Re^{o \times n}$ and $d \in \Re^{o}$. The vector x gives us an approximation solution and the x has plenty of zero elements than non zero elements.

2.3 Just In Time Matrix Database

Database of JIT is collected by applying a randomly selected input for a system. A matrix was designed to instead of model. To get appropriate control input for the future, we need a data from previous process, the data are input and output. The data are arranged in a matrix and the matrix call a database matrix. The goal is to get a better data that near the past process and better result also.

The range of data, length of data, number of data,etc. from input and output can be defined by the operator before arrange them in a matrix. Fig.2.4 and Fig.2.5 illustrate the graph of input and output for database.

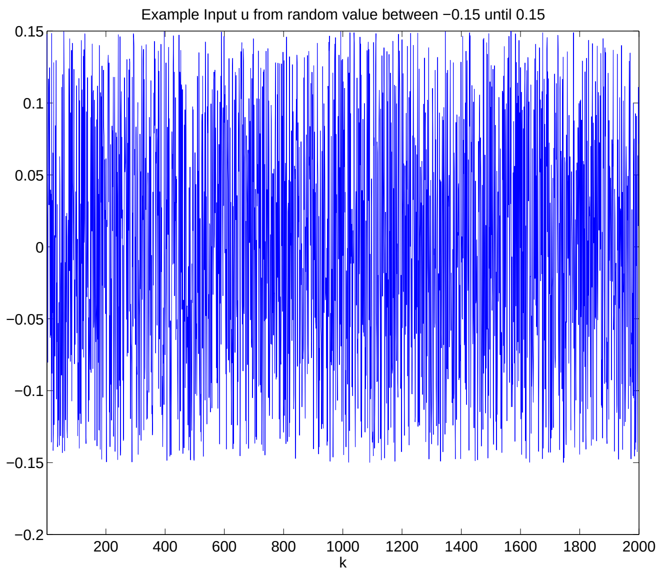


Figure 2.4: Example of input u for database

One way to make database matrix is to collect data from several input and output from random process. To produce the data, we have to assume a few parameter and

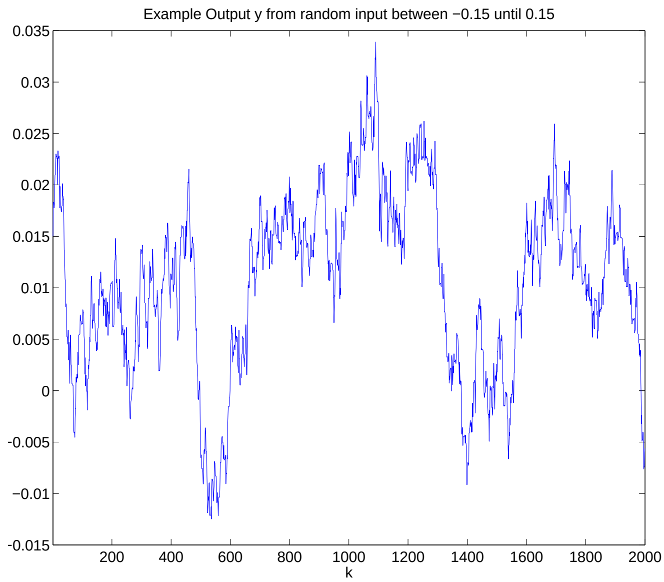


Figure 2.5: Example of output y for database

the parameters are l (length of data), m (integer number of past output), n (integer number of future input) and P (integer number of step ahead). The parameter is defined before design the matrix. The purpose is to measure the size of matrix that is needed in the database to order the space memory in computer.

The aim of this problem is to find the best future input $u(k+1)$ to follow the reference r with the smallest error. For the first step we make a vector consist of u^f is u future, y^f is y future and r is a reference signal. The P-step-ahead is decided by operator. The design of vector can be shown as :

$$u^f = \begin{bmatrix} u(k+1) \\ u(k+2) \\ \vdots \\ u(k+P) \end{bmatrix} \tag{2.7}$$

$$y^f = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+P) \end{bmatrix} \tag{2.8}$$

The reference r for system is :

$$r(k+1) = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+P) \end{bmatrix} \tag{2.9}$$

In this section, the parameter l, m and n produce these kind of vector. The y^p is a past output, u^p is a past input. All of data come from prior experiment to set a matrix database : (where p = past, f=future)

$$y^p(k) = \begin{bmatrix} y(k-(m-1)) \\ \vdots \\ y(k) \end{bmatrix}, \tag{2.10}$$

$$u^p(k) = \begin{bmatrix} u(k-n) \\ \vdots \\ u(k-1) \end{bmatrix} \tag{2.11}$$

After that, we have to get a weight matrix $\psi_1 ... \psi_k$ and the weight matrix comprise of three vector. The matrix ψ_l can be arranged as follow :

$$\psi_l = \begin{bmatrix} y_l^p \\ y_l^f \\ u_l^p \end{bmatrix} \tag{2.12}$$

Matrix ψ_l is an important matrix and ψ_l is matrix A. This matrix is calculated with information vector data $\phi(k)$ to find optimize vector x and to get the information vector discuss in the next section. The last step is to make the a matrix database M and the matrix M consist of two element ψ_l and input u^f. In this matrix we add one parameter u^f, the purpose is to make a memory slot in matrix. But to compare with information vector we do not use u^f. The u^f uses in matrix C to obtain vector d. The following database algorithm is used to predict the $\hat{u}(k+1)$.

$$M = \begin{bmatrix} \psi_1 & \cdots & \psi_l \\ u_1^f & \cdots & u_l^f \end{bmatrix} \tag{2.13}$$

2.4 Just In Time Information Vector

A plant is controlled with assumption by sampling time of T. To solve a predictive control problem, we have to have a way to predict the input for the future and JIT is a controller that we choose.

In Fig.2.6, we can see the overview of the control system with JIT controller. The controller produce u from JIT and then use it to be an input for the system to follow the reference signal and the process repeat time by time until finish.

Almost all JIT method has an information vector ($\phi(k)$). The vector give an information about a few signal to achieve the goal to follow the reference. The information vector data consist of u^p,y^p and r. Therefore, the design of information vector is :

$$u_i^p = \begin{bmatrix} u(\tau_i - n) \\ \vdots \\ u(\tau_i - 1) \end{bmatrix} \tag{2.14}$$

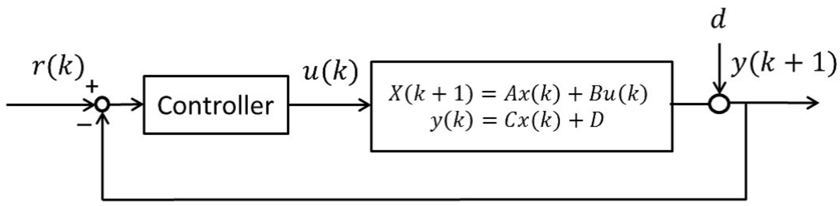


Figure 2.6: System description of discrete controller

$$y^p = \begin{bmatrix} y(\tau_i - (m-1)) \\ \vdots \\ y(\tau_i) \end{bmatrix}, \tag{2.15}$$

$$\phi(k) = \begin{bmatrix} y^p(\tau_i) \\ r(k) \\ u^p(\tau_i) \end{bmatrix} \tag{2.16}$$

In the eq.2.1, there is matrix A. The matrix A in this method is a database ψ and the matrix b is an information vector. The matrix C is a matrix consist of a vector from u^f. The last vector d is a desire future input $\hat{u}(k+1)$ for the system.

2.5 Database Maintenance

An overview of database maintenance :

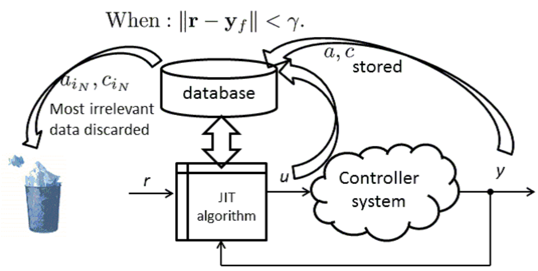


Figure 2.7: Database maintenance in model-free predictive control

When an unstable system is given to be controlled, we first make a database which stores input/output data of the unstable system. Then, we have to stabilize the unstable system to use a standard feedback control method not model-free predictive control. The simplest way of stabilizing is static feedback

$$u(k) = K(r(k) - y(k)) + v(k) \tag{2.17}$$

with a constant gain K and the additional control input v to the stabilized system.

Due to limitation of making the database for model-free predictive control as in the case where the unstable system is given, we often see the degradation in control performance when we control the system at different operation point from that the database was made. To resolve the problem, we store the latest data obtained in real-time control online data into the database.

Chapter 3

Comparison of Model-Free Predictive Control Algorithm

Many kinds of method which can use to solve the optimization problem. In this thesis, there are three methods : Locally Weight Average (LWA), Least-Norm and ℓ_1-norm minimization. The methods are compared each others to understand the capability of methods in model-free predictive control case.

The key to control the system in this paper is weight factor x ($Ax = b$). The weight factor is used to get the input u to the system. The database is made to enrich the information based on previous experiment and the database collected to be a single matrix A. The b is called information vector and designed from a few past output y, reference signal r and a few past input u. In previous chapter the discrete time is indicated by k but in this chapter until the last chapter the discrete time is indicated by t and the symbol k represented for another purpose.

3.1 Locally Weight Average (LWA)

To use model predictive control, we need to apply a system identification technique to obtain a model. Instead, model-free predictive control does not require any mathematical models. Model-free predictive control proposed by [17] utilizes collected past I/O data of the system to build a controller method as N vectors

$$\mathbf{a}_i := \begin{bmatrix} \mathbf{y}_p(t_i) \\ \mathbf{y}_f(t_i) \\ \mathbf{u}_p(t_i) \end{bmatrix} \in \Re^d, i = 1, 2, \ldots, N, \tag{3.1}$$

$$\mathbf{c}_i := \mathbf{u}_f(t_i) \in \Re^{h_u}, i = 1, 2, \ldots, N, \tag{3.2}$$

where

$$d = n + h_y + m, \tag{3.3}$$

$$\mathbf{y}_p(t) = \begin{bmatrix} y(t-n+1) \\ \vdots \\ y(t) \end{bmatrix}, \tag{3.4}$$

$$\mathbf{u}_p(t) = \begin{bmatrix} u(t-m) \\ \vdots \\ u(t-1) \end{bmatrix}. \tag{3.5}$$

An underlying idea of model-free predictive control consists two step:

(i). selecting k nearest vectors $\mathbf{a}_{i_j}$ to a query vector

$$\mathbf{b} = \begin{bmatrix} \mathbf{y}_p(t) \\ \mathbf{r}(t) \\ \mathbf{u}_p(t) \end{bmatrix} \tag{3.6}$$

that contains the current situation $\mathbf{u}_p(t)$, $\mathbf{y}_p(t)$, and the desired trajectory for the future output $\mathbf{r}(t)$;

(ii). generating the expected future input sequence as LWA to use weights x_{i_j} as

$$\hat{\mathbf{u}}_f(t) = \begin{bmatrix} \hat{u}(t|t) \\ \vdots \\ \hat{u}(t+h_u-1|t) \end{bmatrix} \tag{3.7}$$

$$= \sum_{j=1}^{k} x_{i_j} \mathbf{u}_f(t_{i_j}) = \sum_{j=1}^{k} x_{i_j} \mathbf{c}_{i_j}. \tag{3.8}$$

In [17], the so-called Just-In-Time method [6] is utilized. Basically, all vectors $\mathbf{a}_i$ are sorted according to the distance to $\mathbf{b}$ as

$$d(\mathbf{a}_{i_1}, \mathbf{b}) \leq \cdots \leq d(\mathbf{a}_{i_k}, \mathbf{b}) \leq \cdots \leq d(\mathbf{a}_{i_N}, \mathbf{b}). \tag{3.9}$$

In addition, the number k and weights x_{i_j} for $\mathbf{a}_{i_j}$ satisfying

$$x_{i_1} \geq x_{i_2} \geq \cdots \geq x_{i_k} \text{ and } \sum_{j=1}^{k} x_{i_j} = 1. \tag{3.10}$$

are determined, for example by using LWA and the Akaike's Final Prediction Error criterion. In [15], the distance based on the ℓ_1-norm

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{k} |x_i| \tag{3.11}$$

is defined as

$$d(\mathbf{a}, \mathbf{b}) = \left\| W^{-1}(\mathbf{a} - \mathbf{b}) \right\|_1 \tag{3.12}$$

$$W = \operatorname{diag}(w_1, \ldots, w_d) \tag{3.13}$$

where for the ith element of $\mathbf{a}_j$,

$$w_i = \max_{j=1,\ldots N} \mathbf{a}_{ji} - \min_{j=1,\ldots N} \mathbf{a}_{ji}. \quad (3.14)$$

Moreover, the weight is calculated as

$$\begin{aligned} \tilde{x}_i &= \operatorname{tr}\left(I_d - W^{-1}(\mathbf{a}_i - \mathbf{b})(\mathbf{a}_i - \mathbf{b})^T W^{-1}\right) & (3.15) \\ x_i &= \tilde{x}_i / \sum_i^k \tilde{x}_i. & (3.16) \end{aligned}$$

3.2 Linear Norm Solution

In [21], finding the weights x_{i_j} is reformulated as solving the linear equation

$$A\mathbf{x} = \mathbf{b}, \quad (3.17)$$

where

$$\begin{aligned} A &= \begin{bmatrix} \mathbf{a}_{i_1} & \mathbf{a}_{i_2} & \cdots & \mathbf{a}_{i_k} \end{bmatrix} \in \Re^{d \times k}, & (3.18) \\ \mathbf{x} &= \begin{bmatrix} x_{i_1} & x_{i_2} & \cdots & x_{i_k} \end{bmatrix}^T \in \Re^k. & (3.19) \end{aligned}$$

When $d > k$, the solution is given by a least mean square solution as $\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$. When $d < k$, the solution is given by the least-norm (minimum norm) solution $\mathbf{x} = A^T (AA^T)^{-1} \mathbf{b}$ of

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2 . \quad (3.20)$$

The size of the solution $\mathbf{x}$ in (3.20) (i.e., the neighbor size k) can be extended to the size of database N by introducing

$$\begin{aligned} A &= \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_N \end{bmatrix} \in \Re^{d \times N} & (3.21) \\ \mathbf{x} &= \begin{bmatrix} x_1 & x_2 & \ldots & x_N \end{bmatrix}^T \in \Re^N. & (3.22) \end{aligned}$$

as

$$\min_x \quad \|A\mathbf{x} - \mathbf{b}\| \text{ subject to } \|\mathbf{x}\|_0 = k, \quad (3.23)$$

where

$$\|\mathbf{x}\|_0 = \operatorname{card}\{x_i \mid x_i \neq 0\} \quad (3.24)$$

is the l_0 norm is the total number of non-zero elements in $\mathbf{x}$. Because of the l_0 norm constraint, (3.23) is a mixed-integer problem, which is generally difficult to solve in real time.

3.3 ℓ_1-Norm Solution

In [22], (3.23) is reformulated as an ℓ_1-minimization problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } A\mathbf{x} - \mathbf{b} = 0. \tag{3.25}$$

To solve the ℓ_1-minimization problem, it is not necessary to decide the neighbor size k and several methods have been developed. In particular, there are a large number of ℓ_1-minimization algorithms [23] such as gradient projection, homotopy, iterative shrinkage-thresholding, proximal gradient, augmented Lagrange multiplier, and Dual Augmented Lagrange Multiplier (DALM) algorithms[1].

Remark 1 *Just-In-Time algorithms generally cause long feedback delays. Hence, model-free predictive control is limited to slow dynamical systems.*

3.4 Model-free Predictive Control Algorithm

The fundamental procedure is summarized as follows.

Initialization. Determine n, m, N, h_u, and h_y. Let the discrete-time be $t = 0$.

Step 1. Whenever $t \leq \max(n, m)$, repeat this step. Measure $y(t)$ and apply $u(t)$ with an appropriate value to the controlled system. Increment the discrete-time as $t \leftarrow t + 1$.

Step 2. From the given reference trajectory $\mathbf{r}(t)$, define a query vector (3.6).

Step 3. Perform one of the three methods given below.

Step 3a (by LWA). Using the sorted vectors $\mathbf{a}_{i_1} \dots, \mathbf{a}_{i_k}$ satisfying (3.9) and LWA in the Just-In-Time algorithm [6], determine the number k and weights $x_{i_1}, \dots, x_{i_k}$ as 3.10.

Step 3b (by least-norm solution). For the sorted vectors $\mathbf{a}_{i_1} \dots, \mathbf{a}_{i_k}$ satisfying (3.9), determine weights $x_{i_1}, \dots, x_{i_k}$ by solving (3.17).

Step 3c (by ℓ_1-minimization). Using all vectors $\mathbf{a}_1 \dots, \mathbf{a}_N$, solve the ℓ_1-minimization problem 3.25, and determine k and index $i_1, \dots, i_k$ according to

$$|x_{i_1}| \geq \cdots \geq |x_{i_k}| \geq \cdots \geq |x_{i_N}|. \tag{3.26}$$

Step 4. The expected future input sequence is calculated by (3.7).

[1]MATLAB solvers are available at http://www.eecs.berkeley.edu /˜yang/software/l1benchmark/l1benchmark.zip

Step 5. Apply the first element $\hat{u}(t|t)$ of $\hat{\mathbf{u}}_f(t)$ to the system as $u(t)$. Increment the discrete-time as $t \leftarrow t+1$, and return to Step 2.

Remark 2 *In this paper, to compare the ability of model-free predictive control to stabilize unstable systems, only when we first construct a database that stores the input/output (training) data of the given unstable system, we use a standard feedback control, rather than model-free predictive control. Hence, in Section 3, to obtain the the input/output (training) data, we used*

$$u(t) = K(z)(r(t) - y(t)) + v(t) \tag{3.27}$$

with a controller $K(z)$ and a persisting exciting signal v to the system.

3.5 Database Maintenance for system

The stabilization step use (3.27) and the irrelevant data can make the size of database increase, to avoid that for bad data can be deleted. For example in Step 5, at time t the most irrelevant data $\mathbf{a}_{i_N}$ and $\mathbf{c}_{i_N}$ in the database are replaced with

$$\begin{bmatrix} \mathbf{y}_p(t-h) \\ \mathbf{y}_f(t-h) \\ \mathbf{u}_p(t-h) \end{bmatrix} \text{ and } \mathbf{u}_f(t-h) \tag{3.28}$$

where $h = \max(h_y, h_u)$.

However, because this method records u produced unsatisfactory control results (i.e., large difference $r - y$) in the database, it often generates a poor control performance. Hence, we update the database only when (3.28) yields small tracking errors that are less than a prescribed level, i.e.,

$$\|\mathbf{r}(t-h) - \mathbf{y}_f(t-h)\| < \gamma \tag{3.29}$$

where γ is a constant value.

Chapter 4

Simulation and Discussions

4.1 Simulation Setting

In this section, we present several simulation results to evaluate the effect by database updates on model-free predictive control for unstable systems and to compare the three methods in Step 3. We used the system

$$y(t) = 1.2y(t-1) + u(t-1) + \varepsilon(t) \tag{4.1}$$

with the unstable pole 1.2. The training data was created to use stabilizing feedback (3.27) with $K = -0.5$ and $r(k) = 0$. The resulting stabilized system is

$$y(t) = 0.7y(t-1) + v(t-1) + \varepsilon(t). \tag{4.2}$$

To apply 100 sets of random sequences $\varepsilon(t)$ according to Gaussian distribution with zero mean, variance $\sigma^2 = 0.05^2$, and random sequence $v(t)$ generated from a uniform distribution $[-3, 3]$ to the stabilized system, we generated 100 databases containing samples ($N = 600$) of the control input $u(t)$ and output $y(t)$. An example of the input/output data is shown in Fig. 4.1. Throughout the simulations, we set the order of the system and horizons as $n = 1$, $m = 1$, $h_y = 1$, and $h_u = 1$, and used two types of the references r: the sinusoidal signal

$$r(t) = 2\sin\frac{2\pi}{40}t \tag{4.3}$$

and the square signal

$$r(t) = \begin{cases} 0 & 0 \le t < 50 \\ 1 & 50 \le t < 100 \\ 0 & 100 \le t < 150 \\ -1 & 150 \le t < 200 \\ \vdots & \vdots \end{cases} \tag{4.4}$$

We used (3.9) and (3.15) as LWA for Step 3a and fixed the neighbor size $k = 4$. We adopted the distance defined by (3.12) for all methods to sort vectors. In Step 3b, we fixed $k = 10$. Since $d = n + h_y + m = 3 < k$, Step 3b provides the least-norm solution. In Step 3c, we used the DALM method [23] to solve (3.25).

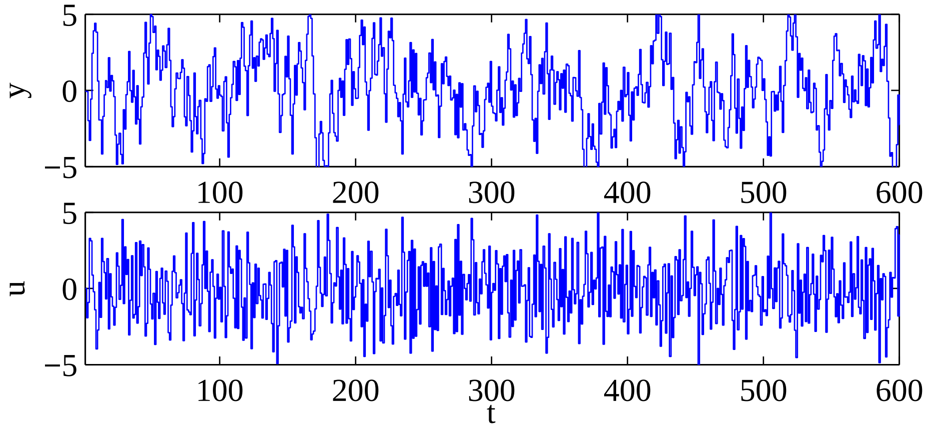


Figure 4.1: Stored measurement data of the stabilized system for model-free predictive control. Top plot: y. Bottom plot: u.

4.2 Results and Discussions

To use the generated 100 databases and another 100 random sequences $\varepsilon(t)$, we simulated the three methods for model-free predictive control. To compare these methods, we calculated the sum of the squares of the tracking error $e(t) = r(t) - y(t)$ for the interval $t \in [a, b]$ as

$$\sum_{t=a}^{b} e(t)^2. \tag{4.5}$$

In Fig. 4.2, To denote the sequence of signal $e(a), ..., e(b)$, we adopt the "colon" notation in Matlab as $e(a : b)$. In fig Fig. 4.2, we show the boxplots of 100 samples of the sum of the squares of $e(101{:}500)$ and we conclude as follows.

- Model-free predictive control by the least-norm solution (Step 3b) and ℓ_1-minimization (Step 3c) yields less tracking errors than the standard LWA method (Step 3a). In the standard LWA method, we have several tunable parameters (the neighbor size k, weight for the distance, etc.). Hence, there is a possibility to obtain better results using more appropriate parameter values.

- Although ℓ_1-minimization (Step 3c) is the best in view of the tracking error, the computational time by ℓ_1-minimization is much longer than that by other methods. The average computational ratios of Step 3b to Step 3a and Step 3c to Step 3a were approximately 0.999 and 14.21, respectively.

- In all methods, the tracking error for the square reference signal is smaller than that for the sinusoidal one because the former is a piecewise constant.

Furthermore, to evaluate the effectiveness of database maintenance for model-free predictive control based on the least-norm solution, we used 100 small-sized

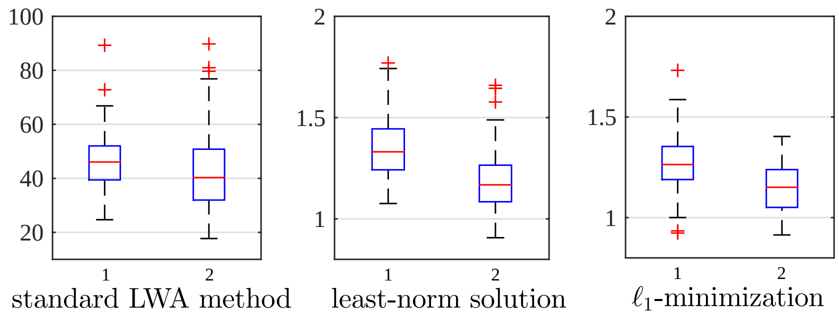


Figure 4.2: Boxplot of the sum of squares of the tracking error $e(t) = r(t) - y(t)$ for the sinusoidal (label 1) and square references (label 2): (a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.

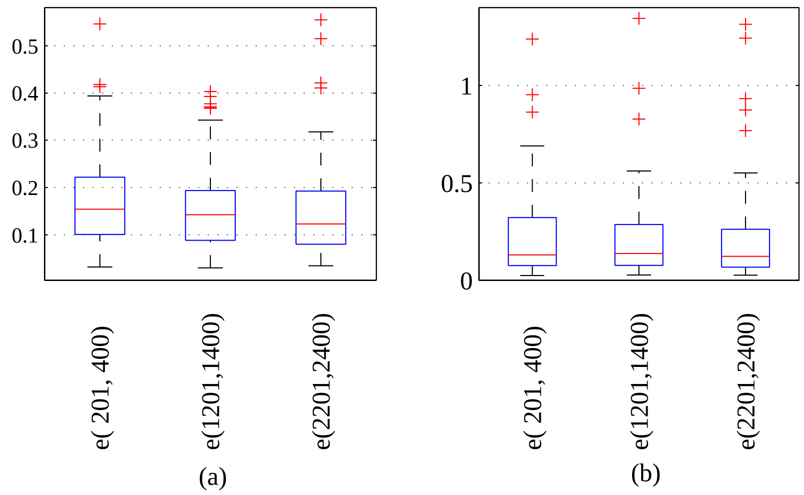
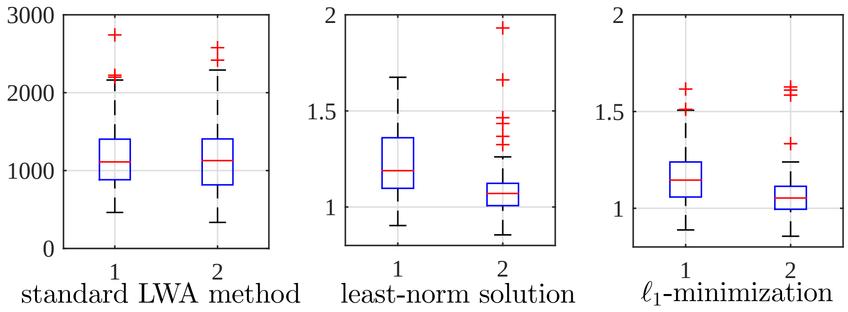


Figure 4.3: Boxplot of the sum of squares of the tracking error $e(t) = r(t) - y(t)$ when we used the method by the least-norm solution to evaluate the effect of database maintenance: (a) the sinusoidal reference and (b) the square reference.

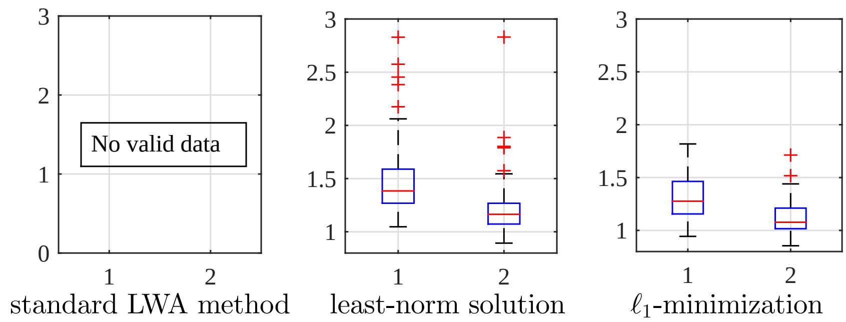
databases ($N = 200$) and compared the sum of squares of the tracking error over three intervals: $e(201{:}400)$, $e(1201{:}1400)$, and $e(2201{:}2400)$.

We show a typical result in Fig. 4.3, which we obtained when we used 100 sets of random sequences $\varepsilon(t)$ according to Gaussian distribution with zero mean and variance $\sigma^2 = 0.01^2$. The variance was smaller than that ($\sigma^2 = 0.05^2$) in the first simulation results. To obtain the results, we used the level of database maintenance $\gamma = 6 \times 10^{-4}$ for the sinusoidal reference and $\gamma = 5 \times 10^{-4}$ for the square reference. The results were sensitive to γ. From Fig. 4.3, we conclude as follows.

- The interquartile range indicated by the boxes became smaller through database maintenance.
- The maximum of data points indicated by the end of the upper whiskers also became smaller through database maintenance.
- There are outliers indicated by "+". In particular, there exist large valued outliers in the results for the square reference.



(a) the system (4.6)



(b) the system (4.7)

Figure 4.4: Boxplot of the sum of squares of the tracking error $e(t) = r(t) - y(t)$ for the sinusoidal (label 1) and square references (label 2).

- The distribution of the tracking errors for the square reference is poorer than that for the sinusoidal reference, unlike the distribution shown in Fig. 4.2; this is because of the piecewise constant reference. The procedure of database maintenance sweeps away important data for another setpoint r. The growth of the ratio of the current setpoint data causes the degradation of the control results when the setpoint r changes.

To investigate more, we also used several addition system for comparison.

$$y(t) = y(t-1) + y(t-2) + 4u(t-1) + \varepsilon(t), \tag{4.6}$$

$$y(t) = 2y(t-1) - 3y(t-2) + 2u(t-1) + \varepsilon(t). \tag{4.7}$$

The training data was also created to use stabilizing feedback

$$u(t) = -0.35y(t-1) + v(t), \tag{4.8}$$

$$u(t) = -0.5y(t) + 1.3y(t-1) + v(t), \tag{4.9}$$

respectively, so that the resulting stabilized systems are both

$$y(t) = y(t-1) - 0.4y(t-2) + v(t-1) + \varepsilon(t). \tag{4.10}$$

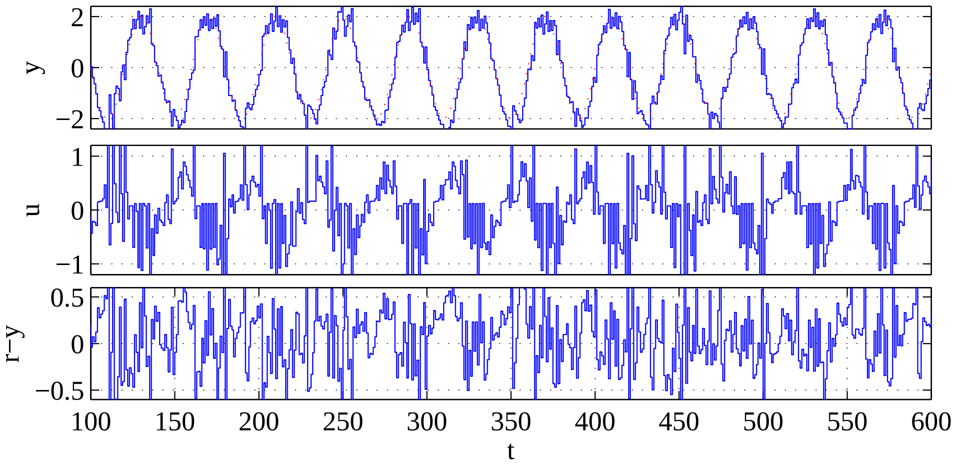
Obtained boxplots are shown in Fig. 4.4. In particular, the standard LWA method (Step 3a) shows worse tracking errors and cannot stabilize (4.7).

4.3 Plots of Simulations

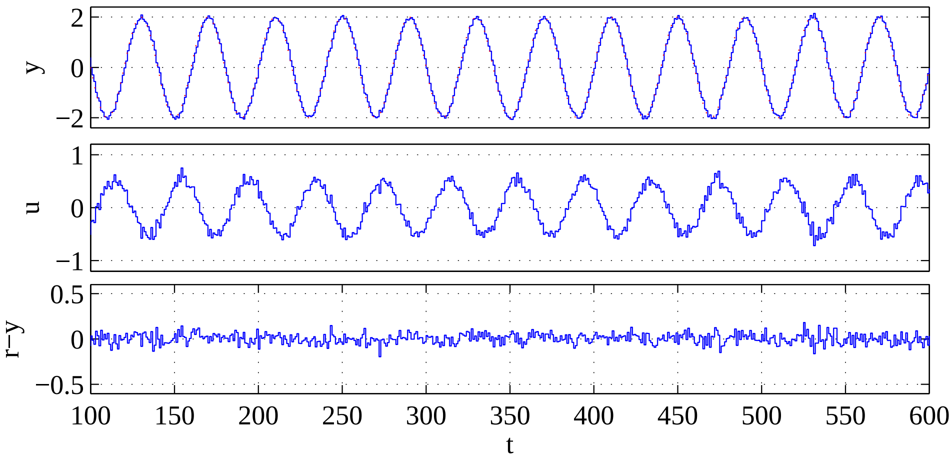
Finally, we show a few of simulation results in Figs. 4.5, 4.6, 4.7 and 4.8. In the figures, the red dashed line indicate the reference signal r; the blue solid line is the output y; and the top, middle, and bottom are output y, input u, and error e, respectively. In the simulation, there are two reference signal. For sinusoidal reference signal are represented by Figs. 4.5 and 4.7, and then square reference signal can be seen in Figs. 4.6 and 4.8.

The fixed database graphs in Figs. 4.5 and 4.6 indicate that for blue line as an output can follow the red line reference very well but in Fig.4.5a which used LWA method, the graph shows the worst result than others graph.

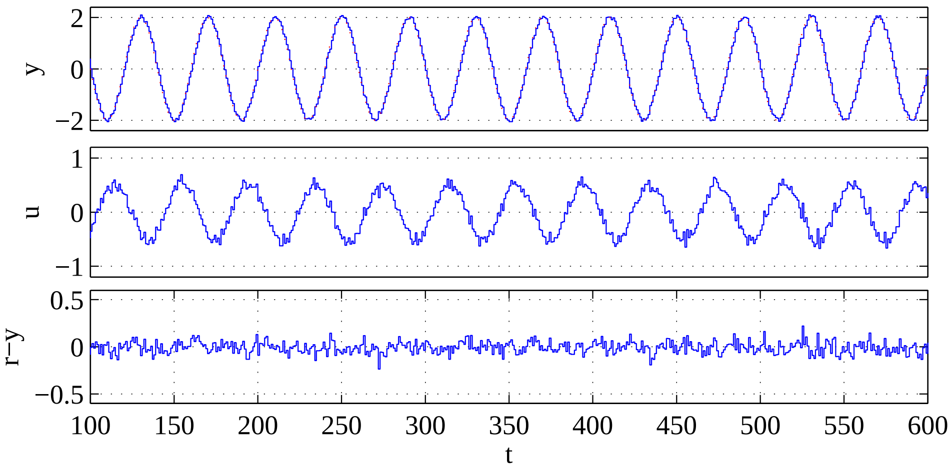
The next simulation used a database maintenance and the simulation can be seen in Figs. 4.7 and 4.8. Both graph give us information that database maintenance give us better results than a fixed database.



(a)



(b)



(c)

Figure 4.5: Simulation results of model-free predictive control for the sinusoidal reference signal using a fixed database and the (a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.

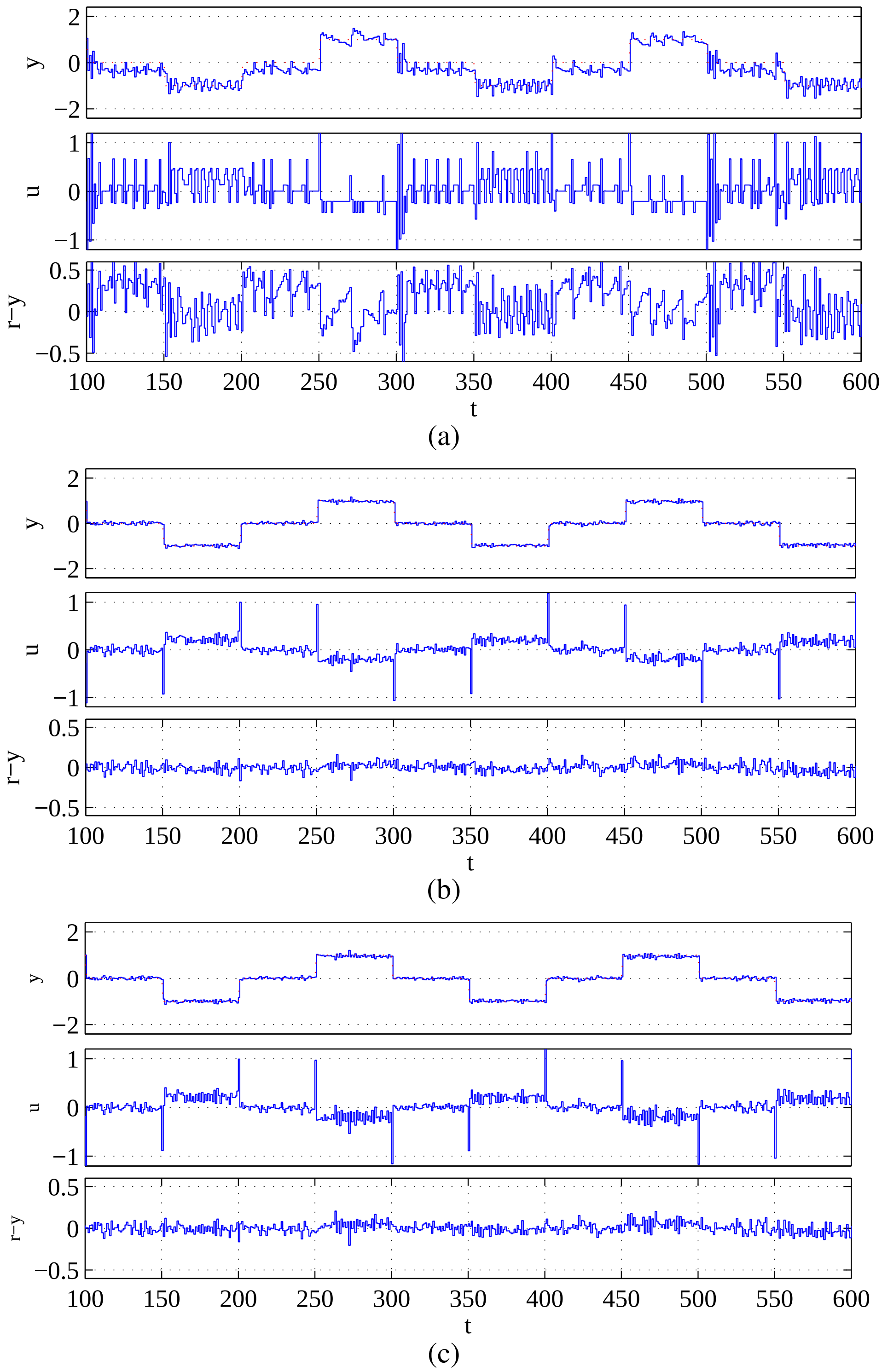


Figure 4.6: Simulation results of model-free predictive control for the square reference signal using a fixed database and the (a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.

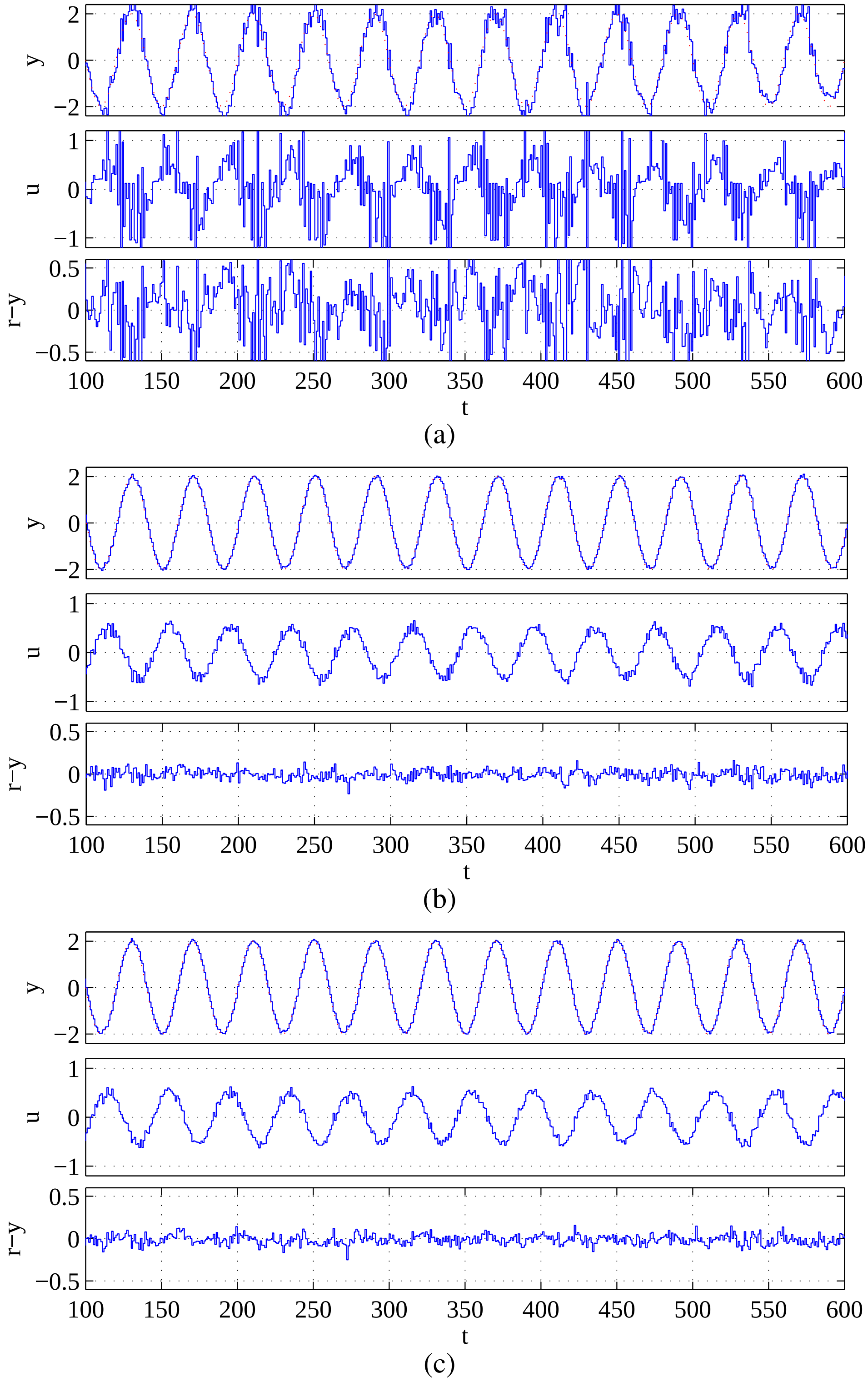


Figure 4.7: Simulation results of model-free predictive control for the sinusoida reference signal using an update database and the (a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.

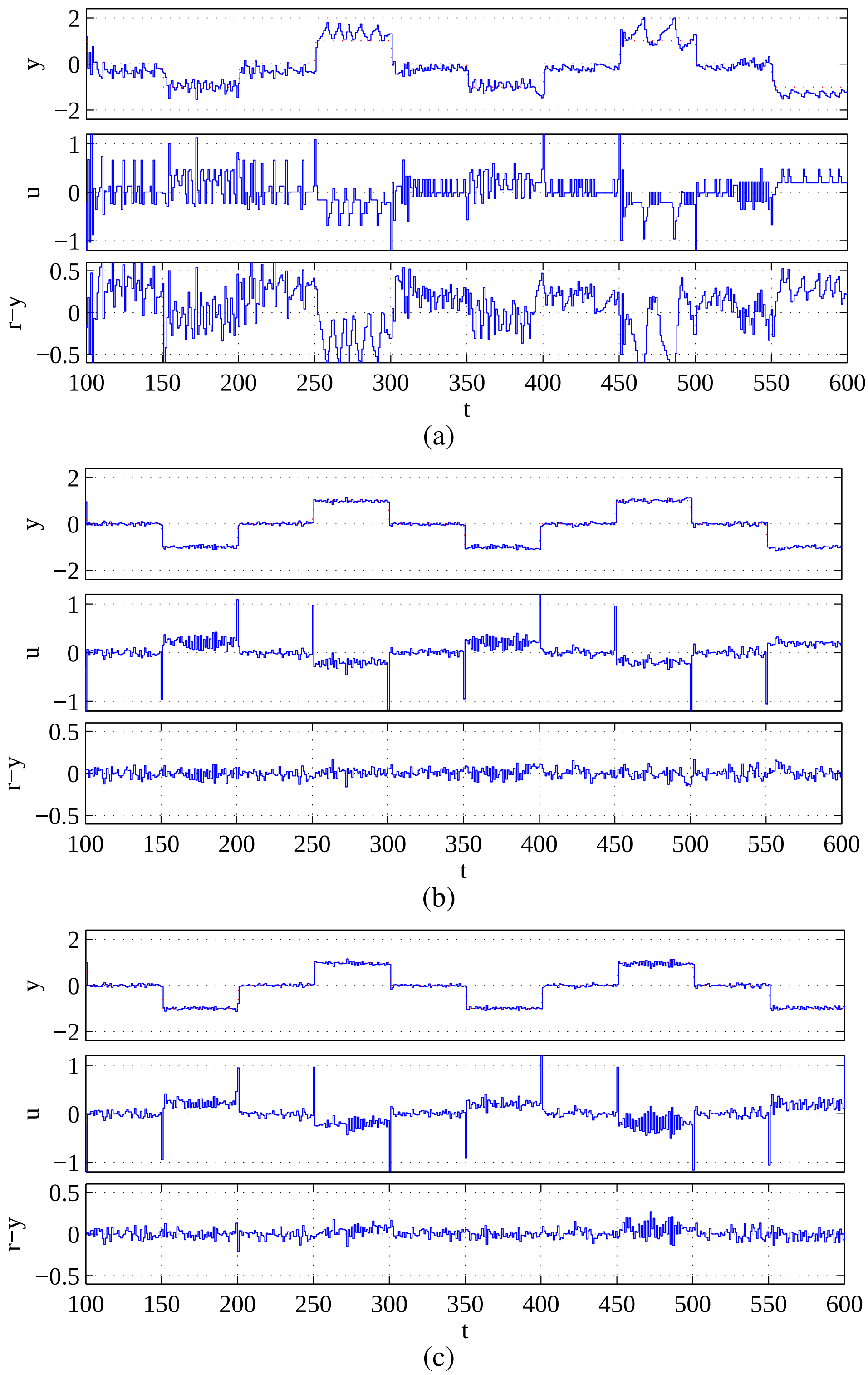


Figure 4.8: Simulation results of model-free predictive control for the square reference signal using an update database and the (a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this study, we compared the three methods based on LWA, least-norm solutions, and ℓ_1-minimization in model-free predictive control using Just-In-Time modeling for an unstable system.

- The least-norm solutions and ℓ_1-norm solutions give much smaller tracking errors than the LWA.
- ℓ_1-minimization requires much longer computational time.
- we concluded that the method using least-norm solutions is the best for practical usage.
- we determined that database maintenance yields better results when working with a small-sized database.
- We concluded for system that has similar characteristics with our system (4.1) can produce similar results in general.

5.2 Future Work

The results in this research are simulation based. Then, we consider that a further theoretical research is needed. The next step is to apply the proposed methods to practical systems and to investigate the difficulties when we use it to the real plants or robot.

Publications

Journal papers

[1] Herlambang Saputra and Shigeru Yamamoto: Comparative Study od Model-Free Predictive Control and its Database Maintenance for Unstable System, *SICE Journal of Control, Measurement, and System Integration* (Accepted)

Bibliography

[1] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, June 2013.

[2] C. E. Garcia, D. M. Prett, and M. Morari, "Model Predictive Control : Theory and Practice a Survey," *Automatica*, vol. Vol.25, no. Iss.3, pp. 335–348, 1989.

[3] A. Stenman, "Model-free Predictive Control," in *38th IEEE Conference on Decision and Control*, vol. 5, pp. 3712–3717, 1999.

[4] G. Cybenko, "Just-in-time learning and estimation," *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES*, vol. 153, pp. 423–434, 1996.

[5] A. Stenman, N. A. V, and G. F, "Asypmtotic properties of Just-In-Time models.," in *11th IFAC Symposium on System Identification*, pp. 1249–1254, 1997.

[6] A. Stenman, *Model on Demand : Algorithms , Analysis and Applications, PhD thesis*. No. 571, Department of Electric Engineering Linkoping University, 1999.

[7] 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.

[8] 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.

[9] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[10] 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.

[11] Q. Zheng and H. Kimura, "Just-in-time Modeling for function prediction and its aplication," *Asian Journal of Control*, vol. 3, no. 1, pp. 35–44, 2001.

[12] M. Kishi, K. Kimura, J. Ohta, and S. Yamamoto, "Shrinkage prediction of a steel production via model-on-demand," *Automation in Mining, Mineral and Metal Processing 2004*, p. 447, 2006.

[13] 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.

[14] 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, June 2004.

[15] 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.

[16] 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.

[17] 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, Sept. 2006.

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

[19] E. Konaka, "Design of Discrete Predictive Controller Using Approximate Nearest Neighbor Method," in *Proceedings of the 18th IFAC World Congress*, (Milano (Italy)), pp. 10213–10218, Aug. 2011.

[20] 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.

[21] S. Yamamoto, "A new model-free Predictive Control Method using Input and Output Data.," in *3rd International Conference on Key Engineering Materials and Computer Science (KEMCS 2014)* (Advanced Materials Research Vol.1042, ed.), (Singapore, August 5, 2014), pp. 182–187, Trans Tech Publication, 2014.

[22] S. Yamamoto, "A Model-Free Predictive Control Method by L_1-minimization," in *Proceedings of the 10th Asian Control Conference 2015 (ASCC 2015)*, 2015.

[23] A. Y. Yang, A. Ganesh, Z. Zhou, S. Sastry, and M. Y, "A review of fast l1-minimization algorithms for robust face recognition, http://arxiv.org/abs/1007.3753," 2010.

[24] M. Nagahara, D. E. Quevedo, and J. Ostergaard, "Sparse Packetized Predictive Control for Networked Control Over Erasure Channels," in *IEEE Transaction On Automatic Control*, vol. 59, No.7, pp. 1899–1905, 2014.

[25] N. Nakpong and S. Yamamoto, “Just-In-Time predictive control for a two-wheeled robot,” in *2012 Tenth International Conference on ICT and Knowledge Engineering*, no. 3, (Thailand), pp. 95–98, Nov. 2012.

[26] J. Maciejowski, “Predictive Control with Constraints,” London: Prentice Hall, 2002.