A Study on Data-Driven Predictive Control

Abstract for Disertation

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

1 Research Motivation and Objective

Main idea of this research is to understand control performance use model-free predictive control or Just-in-Time (JIT) predictive control, the model as a data-driven for controller to obtain optimize solutions. Model predictive control commonly use in chemical industry [1] and proposed by Stenman in 1999 [2], the method use an update of mathematical model constantly refer to input and output data by Just-In-Time modelling. The data store in a database [3], [4] a sufficient data [5].

Inoue and Yamamoto [6] proposed another ”model free” predictive control in the just-in-time modelling framework. In the method, an optimal control input is directly predicted not using any local models, but by online current measured data and stored past data.

More recently, two approaches substituting the conventional the nearest neighbor and LWA technique have been introduced [7], [8]. In [7], weights are calculated as a solution of a linear equation. In [8], weights are computed as a solution of an ℓ_1-minimization problem which produces a sparse vector with a few nonzero elements. This kind of ℓ_1-minimization is now popular in signal processing community [9].

The focus of this paper is to compare of three methods ([6], [7], and [8]) 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 and so on. Boundedness of all signals in the control system only will be guaranteed in practical applications. In this paper, we statistically evaluate the effect by model free predictive control through many trials. When an unstable system is given, it is difficult to make a rich database containing input/output data without feedback control. Hence, we assume that there exists simple feedback control stabilizing the unstable system to make a database. However, when we use model free predictive control, we do not use the stabilizing controller unlike [10] and [11]. 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.

2 Model Free Predictive Control

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 :

$$u^f = \begin{bmatrix} u(k+1) \\ \vdots \\ u(k+P) \end{bmatrix}, \quad y^f = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+P) \end{bmatrix} \tag{1}$$

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

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. (where p = past, f=future)

$$y^p(k) = \begin{bmatrix} y(k-(m-1)) \\ \vdots \\ y(k) \end{bmatrix}, \quad u^p(k) = \begin{bmatrix} u(k-n) \\ \vdots \\ u(k-1) \end{bmatrix} \tag{3}$$

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 :

$$\psi_l = \begin{bmatrix} y_l^p \\ y_l^f \\ u_l^p \end{bmatrix} \tag{4}$$

2.1 Just In Time Information Vector

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. Therefore, the design of information vector is :

$$u_i^p = \begin{bmatrix} u(\tau_i - n) \\ \vdots \\ u(\tau_i - 1) \end{bmatrix}, \quad y^p = \begin{bmatrix} y(\tau_i - (m-1)) \\ \vdots \\ y(\tau_i) \end{bmatrix}, \tag{5}$$

$$\phi(k) = \begin{bmatrix} y^p(\tau_i) \\ r(k) \\ u^p(\tau_i) \end{bmatrix} \tag{6}$$

The vector ψ in this method is a database and the vector ϕ is an information vector.

2.2 Database Maintenance

An overview of database maintenance :

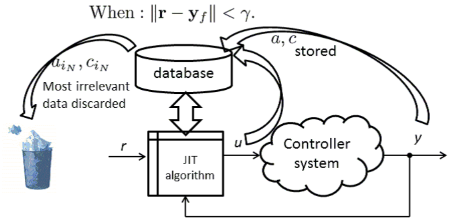


Figure 1: Database maintenance in model-free predictive control

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

$$u(k) = K(r(k) - y(k)) + v(k) \tag{7}$$

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

3 Comparison of Model-Free Predictive Control Algorithm

3.1 Locally Weight Average (LWA)

Model-free predictive control proposed by [6] utilizes collected past input/output data of the controlled system as N vectors

$$\mathbf{a}_i := \begin{bmatrix} \mathbf{y}_p(t_i) \\ \mathbf{y}_f(t_i) \\ \mathbf{u}_p(t_i) \end{bmatrix} \in \Re^d, i = 1, 2, \ldots, N, \tag{8}$$

$$\mathbf{c}_i := \mathbf{u}_f(t_i) \in \Re^{h_u}, i = 1, 2, \ldots, N, \tag{9}$$

where $d = n + h_y + m$,

$$\mathbf{y}_p(t) = \begin{bmatrix} y(t-n+1) \\ \vdots \\ y(t) \end{bmatrix}, \quad \text{and}\, \mathbf{u}_p(t) = \begin{bmatrix} u(t-m) \\ \vdots \\ u(t-1) \end{bmatrix}. \tag{10}$$

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

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

$$\mathbf{b} \ = \begin{bmatrix} \mathbf{y}_p(t) \\ \mathbf{r}(t) \\ \mathbf{u}_p(t) \end{bmatrix} \tag{11}$$

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

$$\hat{\mathbf{u}}_f(t) = \begin{bmatrix} \hat{u}(t|t) \\ \vdots \\ \hat{u}(t+h_u-1|t) \end{bmatrix} \tag{12}$$

$$= \sum_{j=1}^{k} x_{i_j} \mathbf{u}_f(t_{i_j}) = \sum_{j=1}^{k} x_{i_j} \mathbf{c}_{i_j}. \tag{13}$$

In [6], the so-called Just-In-Time method [5] is utilized. Basically, all vectors $\mathbf{a}_i$ are sorted according to the distance to $\mathbf{b}$ as

$$d(\mathbf{a}_{i_1}, \mathbf{b}) \leq \cdots \leq d(\mathbf{a}_{i_k}, \mathbf{b}) \leq \cdots \leq d(\mathbf{a}_{i_N}, \mathbf{b}). \tag{14}$$

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

$$x_{i_1} \geq x_{i_2} \geq \cdots \geq x_{i_k} \text{ and } \sum_{j=1}^{k} x_{i_j} = 1. \tag{15}$$

are determined, for example by using LWA and the Akaike's Final Prediction Error criterion. In [12], the distance based on the ℓ_1-norm

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{k} |x_i| \tag{16}$$

is defined as

$$\begin{aligned} d(\mathbf{a}, \mathbf{b}) &= \left\| W^{-1}(\mathbf{a} - \mathbf{b}) \right\|_1 && (17) \\ W &= \operatorname{diag}(w_1, \ldots, w_d) && (18) \end{aligned}$$

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

$$w_i = \max_{j=1,\ldots N} \mathbf{a}_{ji} - \min_{j=1,\ldots N} \mathbf{a}_{ji}. \tag{19}$$

Moreover, the weight is calculated as

$$\begin{aligned} \tilde{x}_i &= \operatorname{tr} \left(I_d - W^{-1}(\mathbf{a}_i - \mathbf{b})(\mathbf{a}_i - \mathbf{b})^T W^{-1} \right) && (20) \\ x_i &= \tilde{x}_i / \sum_{i}^{k} \tilde{x}_i. && (21) \end{aligned}$$

3.2 Linear Norm Solution

In [7], finding the weights x_{i_j} is reformulated as solving the linear equation

$$A\mathbf{x} = \mathbf{b}, \tag{22}$$

where

$$\begin{aligned} A &= \begin{bmatrix} \mathbf{a}_{i_1} & \mathbf{a}_{i_2} & \cdots & \mathbf{a}_{i_k} \end{bmatrix} \in \Re^{d \times k}, && (23) \\ \mathbf{x} &= \begin{bmatrix} x_{i_1} & x_{i_2} & \cdots & x_{i_k} \end{bmatrix}^T \in \Re^{k}. && (24) \end{aligned}$$

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

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2 . \tag{25}$$

The size of the solution $\mathbf{x}$ in (25) (i.e., the neighbor size k) can be extended to the size of database N by introducing

$$A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_N \end{bmatrix} \in \Re^{d \times N} \tag{26}$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T \in \Re^N. \tag{27}$$

as

$$\min_x \quad ||A\mathbf{x} - \mathbf{b}|| \text{ subject to } ||\mathbf{x}||_0 = k, \tag{28}$$

where

$$||\mathbf{x}||_0 = \text{card}\{x_i \mid x_i \neq 0\} \tag{29}$$

is the l_0 norm is the total number of non-zero elements in $\mathbf{x}$. Because of the l_0 norm constraint, (28) is a mixed-integer problem, which is generally difficult to solve in real time.

3.3 l_1 Norm Solution

In [8], (28) is reformulated as an ℓ_1-minimization problem:

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 \text{ subject to } A\mathbf{x} - \mathbf{b} = 0. \tag{30}$$

To solve the ℓ_1-minimization problem, several methods have been developed. In particular, there are a large number of ℓ_1-minimization algorithms [9] such as gradient projection, homotopy, 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.*

4 Model-free Predictive Control Algorithm

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 (11).

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

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

Step 3a (by LWA), determine the number k and weights $x_{i_1}, \dots, x_{i_k}$ as (15).

Step 3b (by least-norm solution), determine weights $x_{i_1}, \dots, x_{i_k}$ by (22).

Step 3c (by ℓ_1-minimization), solve by the ℓ_1-minimization problem (30).

Step 4. The expected future input sequence is calculated by (12).

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.

5 Database Maintenance for system

The stabilization step use (7) 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

$$\begin{bmatrix} \mathbf{y}_p(t-h) \\ \mathbf{y}_f(t-h) \\ \mathbf{u}_p(t-h) \end{bmatrix} \text{ and } \mathbf{u}_f(t-h) \tag{31}$$

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 (31) yields small tracking errors that are less than a prescribed level, i.e.

$$||\mathbf{r}(t-h) - \mathbf{y}_f(t-h)|| < \gamma. \tag{32}$$

where γ is a constant value.

6 Simulation and Discussions

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

$$y(t) = 1.2y(t-1) + u(t-1) + \varepsilon(t) \tag{33}$$

with the unstable pole 1.2. The training data was created to use stabilizing feedback (7) with $K = -0.5$ and $r(k) = 0$. The resulting stabilized system is

$$y(t) = 0.7y(t-1) + v(t-1) + \varepsilon(t). \tag{34}$$

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 $y(t)$. 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 signal r:

$$\text{sinusoidal} : r(t) = 2\sin\frac{2\pi}{40}t, \quad \text{square} : r(t) = \begin{cases} 0 & 0 \le t < 50 \\ 1 & 50 \le t < 100 \\ 0 & 100 \le t < 150 \\ -1 & 150 \le t < 200 \\ \vdots & \vdots \end{cases} \tag{35}$$

We used (14) and (20) as LWA for Step 3a and fixed the neighbor size $k = 4$. We adopted the distance defined by (17) 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 [9] to solve (30).

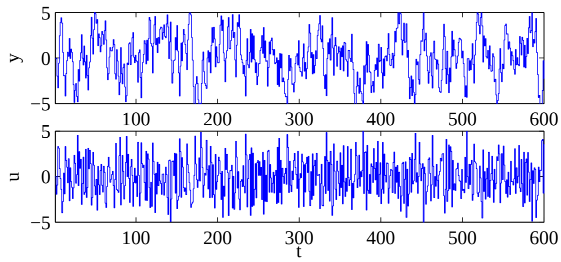


Figure 2: Stored measurement data. Top plot: y. Bottom plot: u.

To use the generated 100 databases and another 100 random sequences $\varepsilon(t)$, we simulated the three methods for model-free predictive control. We calculated the sum of the squares of the tracking error to compare these methods $e(a{:}b) = r(a{:}b) - y(a{:}b)$, we adopt the "colon" notation in Matlab, as

$$\sum_{t=a}^{b} e(t)^2. \tag{36}$$

In Fig. 3, From Fig. 3, 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). 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.

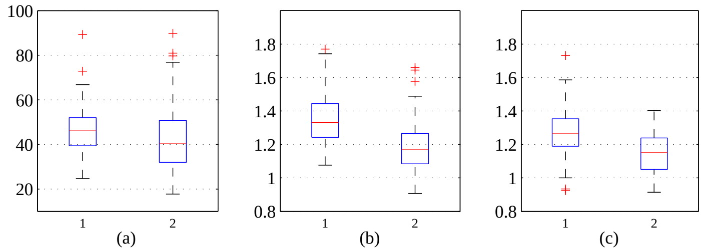


Figure 3: Boxplot of the sum of squares of the tracking error $e(t) = r(t) - y(t)$ for the sinusoidal (label 1) and square references (label 2): (a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.

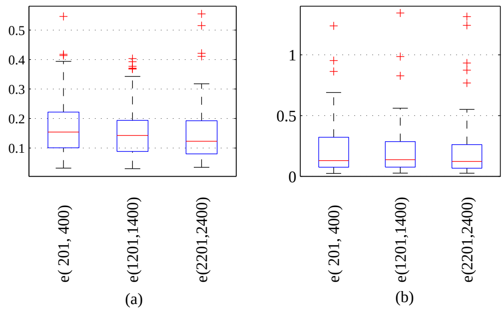
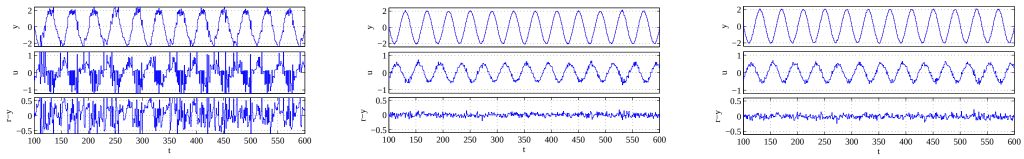


Figure 4: Boxplot of the sum of tracking error $e(t) = r(t) - y(t)$ by the least-norm solution to evaluate the effect of database maintenance: (a) the sinusoidal reference and (b) the square reference.

We show a typical result in Fig. 4, 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, we conclude as follows.

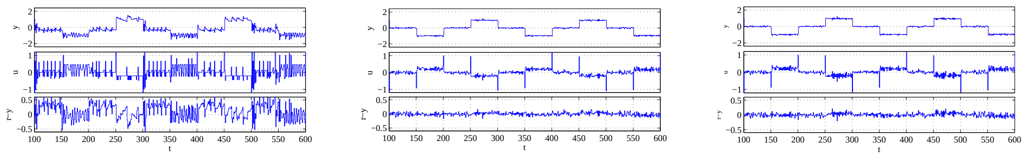
- The interquartile range indicated by the boxes became smaller through database maintenance.
- The maximum of data points indicated by the end of the upper whiskers also became smaller through database maintenance.
- There are outliers indicated by "+". In particular, there exist large valued outliers in the results for the square reference.
- The distribution of the tracking errors for the square reference is poorer than that for the sinusoidal reference, unlike the distribution shown in Fig. 3; this is because of the piecewise constant reference.

Finally, we show examples of simulation results in Figs. 5, 6, 7 and 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. For Figs.5-8:(a) standard LWA method, (b) least-norm solution, and (c) ℓ_1-minimization.



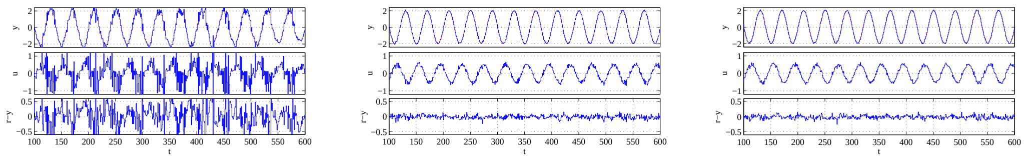
(a) (b) (c)

Figure 5: Simulation results using a fixed database.



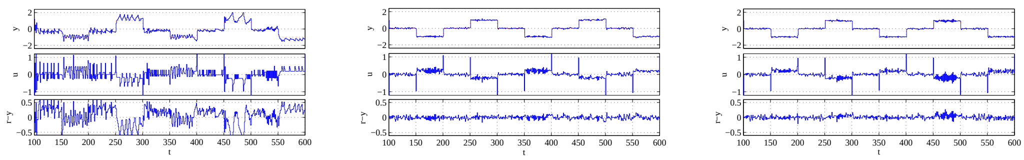
(a) (b) (c)

Figure 6: Simulation results using a fixed database.



(a) (b) (c)

Figure 7: Simulation results using an update database.



(a) (b) (c)

Figure 8: Simulation results using an update database.

7 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 gave much smaller tracking errors than the LWA. Since ℓ_1-minimization requires much longer computational time, we concluded that the method using least-norm solutions is the best for practical usage. Furthermore, we determined that database maintenance yields better results when working with a small-sized database.

References

[1] 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.

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] S. Yamamoto, “A Model-Free Predictive Control Method by l_1-minimization,” in *Proceedings of the 10th Asian Control Conference 2015 (ASCC 2015)*, 2015.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

平成２７年　８月　３日

学位論文審査報告書（甲）

１．学位論文題目（外国語の場合は和訳を付けること。）

A Study on Data-Driven Predictive Control

（データ駆動型予測制御に関する研究）

２．論文提出者　(1) 所　　属　電子情報科学　　専攻

(2) 氏　　名（ふりがな）　Herlambang Saputra（はーらむばん　さぷとら）

３．審査結果の要旨（600～650字）

当該学位論文に関し，平成２７年８月３日に第１回学位論文審査委員会を開催した．同日に口頭発表を実施し，その後に第２回審査委員会を開催した．慎重審議の結果，以下の通り判定した．なお口頭発表における質疑を最終試験に代えるものとした．

本論文は，制御対象のモデルの代わりに制御対象の観測値を直接利用する予測制御に関する研究をまとめたものである．特に，モデルを用いない予測制御ではこれまで適用が困難であると考えられていた不安定な制御対象の安定化に成功した結果を示している．本論文が対象とするデータ駆動型予測制御は，制御対象の観測値を事前に蓄えておき，その中から予測に必要なデータを抽出することと最適な制御入力の算出を同時に行う．その代表的な方法の比較と蓄積データのメンテナンス法の考察を数値シミュレーションを駆使して行っている．特に，重み付き線形平均法により最適入力を求める場合は安定化が難しいことと，最小ノルム解とL1ノルム最小解を用いて最適入力を求めると良好な制御性能が得られることを示めしている．

以上の研究成果は，データ駆動型予測制御の研究に新しい知見を与えるだけでなく，産業応用を考える上での貢献も大きい．従って，本論文は博士（学術）に値すると判定した．

４．審査結果　(1) 判　　定（いずれかに○印）　合　格　・　不合格

(2) 授与学位　博　士（　学　術　）