A Study for effectiveness of Dimensionality Reduction for State-action Pair Prediction –Training set reduction using Tendency–

Masashi Sugimoto[1*] Naoya Iwamoto[1] Robert W. Johnston[1]
Keizo Kanazawa[1] Yukinori Misaki[1] Kentarou Kurashige[2]

[1] National Institute of Technology, Kagawa College

[2] Muroran Institute of Technology

[*]Department of Electronic Systems Engineering,
551 Kohda, Takuma, Mitoyo, Kagawa, 769-1192 Japan.

Email(corresponding author): sugimoto-m@es.kagawa-nct.ac.jp

ABSTRACT

This paper investigates the effectiveness of reduction of training sets and kernel space for action-decision using future prediction. Considering a working in a real environment based on future prediction, it's necessary to know the property of its state and disturbance that will be given by the outside environment. On the other hand, obtaining the property of the disturbance depends on specification for target processor, especially, sensor resolution or processing ability of the processor. Therefore, sampling rate settings will be limited by hardware specification. In contrast, in case of a future prediction using a machine learning, it predicts that based on the tendency that obtained by past training or learning. In this kind of situation, the learning time will be proportionally larger to training data. At worst, the prediction algorithm will be hard to work in real time due to time-complexity.

In the proposed method, the possibility of carefully analyzing the algorithm and applying dimensionality reduction techniques in order to accelerate the algorithm has been considered. In particular, we will consider that to reduce the training sets and kernel space based on the recent tendency of disturbance or state using FFT and pattern matching will be focused on. From this standpoint, we will propose the method that to dimensionality reduction dynamically based on the tendency of disturbance.

KEYWORDS

Online SVR, Predict and Control using State-action Pair Prediction, Dimensionality Reduction

1 INTRODUCTION

Considering an action decision based on future prediction, it's necessary to know the property of disturbance that will be given by outside environment [1]. On the other hand, obtaining the property of the disturbance is depend on specification for target processor, especially, sensor resolution or processing ability of the processor. Therefore, sampling rate settings will be limited by hardware specification. In contrast, in case of a future prediction using a machine learning, it predicts that based on the tendency that obtained by past training or learning. In this kind of situation, the learning time will be proportionally larger to training data [1, 2].

A State-action Pair Prediction had been proposed. In this method, the prediction performance [3] and action decision methods [4, 5], had been considered based on some prediction results. In above-mentioned methods, the behavior of the robot has been considered when an unknown periodic disturbance signal will be given the robot, continuously. On the other hand, in these studies, the learning space had not been considered in action decision or future prediction. In general, non-linear clustering (or regression, such as this work), Kernel Function was used, that allows growth of the SVM (also SVR) solution, which starts invading other space, and this "other space" is called the Features Space. This allows us to change the information from one linear space

ISBN: 978-1-941968-42-0 ©2017 SDIWC

to another one. This permits us to better classify (or regression) the examples. However, the speed of learning depends mostly on the number of support vectors, that can influence significatively performances. Therefore, in simply, the complexity of learning will be proportional as the size of training sets. If the recent tendency of disturbance or state, or these period will be obtained, the training sets will be reduced. Moreover, the length of training sets will be fixed in spite of a new training set will be added.

Therefore, in the proposed method, the possibility of carefully analyzing the algorithm and applying dimensionality reduction techniques in order to accelerate the algorithm has been considered. In particular, we will consider that to reduce the training sets and kernel space based on the recent tendency of disturbance or state using FFT and pattern matching will be focused on. From this standpoint, we will propose the method that to dimensionality reduction dynamically based on the tendency of disturbance.

This paper is organized as follows: In section II, how to reduce a learning space (feature space) dynamically, will be motivated. Further, details about the decide a learning space based on the Nearest-neighbor one-step-ahead forecasts and Nyquist-Shannon Sampling Theorem will be stated. In Section III, a verification experiment configuration will be described. In Section IV, the summary of this work is concluded.

2 AN APPROACH OF THE REDUCING THE LEARNING SPACE BASED ON FREQUENCY PROPERTY OF THE DISTURBANCE SIGNAL

2.1 About Former Our Works

We mentioned in citations that for controlling the robot in a dynamic environment, it can realize choosing the action that adopted the current result by predicting the future state using previous actions and states. In this paper, we will try to consider that obtain the optimal action that is minimizing the body pitch angle of the inverted pendulum, in case of continuing the predictive disturbance using the prediction the State-action Pair that had proposed in the former our study [3]. Therefore, in this paper, we considering the system that decides the action to optimize as the proposed method in fig. 1 based on the former study and the study [4].

In fig. 1, this system will be applied optimal

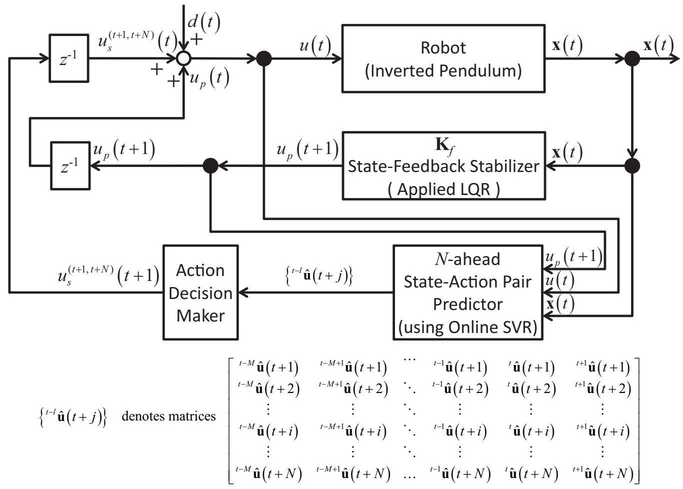


Figure 1. A Outline of the Deciding the Optimal Action for the Robot using the Prediction of State-action Pair [5]

control using a gain $\mathbf{K}_f$ as an optimal feedback gain, and in parallel, deciding the action that will have to take in the future using the prediction of state-action pair. In fig. 1, ${}^{t-l}\hat{\mathbf{u}}(t+j)$ is describing the prediction result of the control input $\mathbf{u}(t+j)$, when that input predicted in time $(t-l)$. And hence, this proposed method is revised the current action using the action that combining the optimal control and the prediction result of State-action Pair Prediction. The structure of the prediction of State-action Pair is named "N-ahead State-action Pair Predictor," that the internal structure is described in fig. 2 [3]. The proposed method obtain a series of action in time $(t+N)$ in the distant future from current time t using N-ahead State-

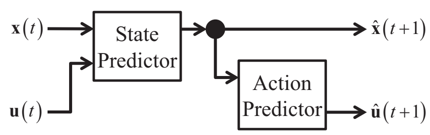


Figure 2. Outline of the Prediction System of State and Action [3]

action Pair Predictor. Now from this prediction series, the current action, combining "*the action will be taken in the future*" and using the prediction series of the action, will be able to revise.

Hereby, in this system, the optimal compensation control input $\mathbf{u}(t)$ will be given as follow:

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1,t+N)}(t) + \mathbf{d}(t) \qquad (1)$$

In this equation, $\mathbf{u}_p(t)$ denotes an optimal control action with optimal feedback control gain, $\mathbf{u}_s^{(t+1,t+N)}(t)$ is generated from "Action Decision Maker," and $\mathbf{d}(t)$ denotes unknown periodic disturbance input signal. Then, $\mathbf{u}_s^{(t+1,t+N)}(t)$ can be defined as follows.

$$\mathbf{u}_s^{(t+1,t+N)}(t) = \sum_{i=1}^{N} \alpha_i \hat{\mathbf{u}}(t+i) \qquad (2)$$

Moreover, in this study, the coefficient α_i will be defined as:

$$\alpha_i = \frac{N+i-1}{100 \cdot N} \qquad (3)$$

From this technique, we create an action that can correspond in ahead a time, can obtain the optimal action that will be considered in the future.

2.2 Basic Idea

In Section I, we stated about the relationship dimensionality reduction and size of training sets for the learning and predicting. From this viewpoint, we can reduce the training time and the size of the learning space for predicting future states and action if we can reduce the training sets.

In SVR, it defines "support vector," and a number of them are less than numbers of other training sets. In detail, support vector denotes the property of unknown function. From this viewpoint, it's not needed a precision training set for unknown function. Thus, removing any training samples that are not relevant to support vectors might have no effect on building the proper decision function [6]. In other words, in this study, the training sets can reduce if the unknown periodic disturbance will be applied to plant model. In addition, the model can be built a prediction model for an almost-periodic disturbance signal if the support vectors can denote the property of an almost-periodic; that is, in the proposed method, the training sets will be reduced based on the recent tendency of disturbance or state, or these period. Moreover, the length of training sets will be fixed in spite of a new training set will be added. Moreover, the support vectors of "one-period" of almost-periodic disturbance will be used repeatedly. Therefore, in the proposed method, the prediction model can be predicted and adapted the plant if an unknown periodic disturbance will be applied as same as former works, in spite of dimensionality reduction dynamically based on the tendency of disturbance.

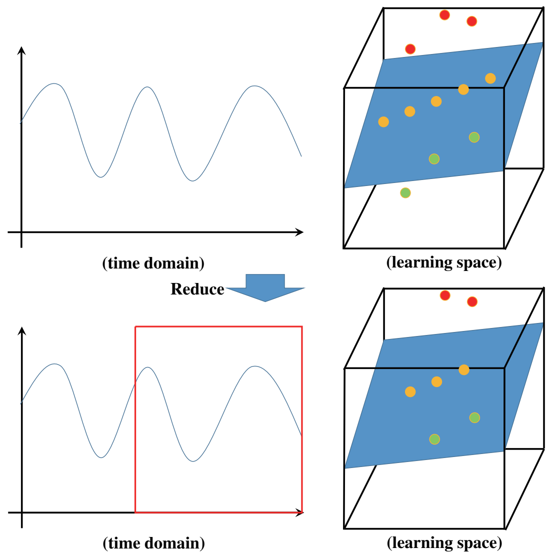


Figure 3. Outline of reduce the learning space according to a Period of Disturbance Signal

2.3 How to Estimate the Frequency of A Disturbance Signal and Reduce Learning Space

As mentioned above, the unknown periodic signal will be used as a disturbance signal. Therefore, we will try to analyze the property of disturbance signal, to reduce the learning space (in fig. 3). In this case, the disturbance signal will be represented as a similar tendency as a pattern.

From this property, the Nearest-neighbor One-step-ahead forecasts [7] will be applied, to detect a cycle period of the disturbance signal. Let illustrate about the Nearest-neighbor One-step-ahead forecasts. As shown in fig. 4, target function $f(t)$ will be repeated similar tendency. In this case, we want to predict at time $t+1$ the next value of the series f. The pattern $f(t-12), f(t-6)$ is the most similar to the pattern. Then, the prediction will be calculated. As a result, the Nearest-neighbor One-step-ahead forecasts provide not only a one-step prediction result, but also the cycle period.

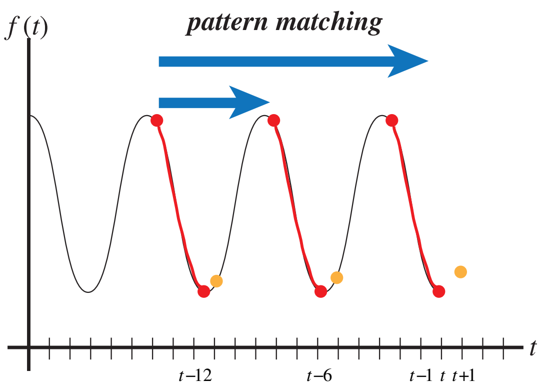


Figure 4. Outline of Nearest-neighbor One-step-ahead Forecasts

From these results, the cycle period $T_{\text{disturbance}}$ will be calculated as follows:

$$T_{\text{disturbance}} = |t_{\text{max, disturbance}} - t_{\text{min, disturbance}}| \times 2 \tag{4}$$

Here, $t_{\text{max, disturbance}}$ denotes the time when the maximum values of disturbance signal $d(t)$ has been reached, moreover, $t_{\text{min, disturbance}}$ denotes the time when the minimum values of disturbance signal $d(t)$ has been reached.

On the other hand, we have stated that we will try to analyze the property of disturbance signal, to reduce the training sets, as mentioned above. Therefore, Nyquist-Shannon Sampling Theorem will be focused on, to reduce the training sets, and to keep the property of the disturbance signal. Now, the Theorem states: *A sufficient sample-rate is therefore $2B$ samples/second, or anything larger. Equivalently, for a given sample rate f_s, perfect reconstruction is guaranteed possible for a bandlimit $B < f_s/2$.* From this theorem, the sampling rate t'_s will be defined as follows:

$$t'_s \leq \frac{T_{\text{disturbance}}}{2} \tag{5}$$

In here, the sampling rate t'_s ts will be integral multiple of original sampling rate t_s. Then, a result obtained by divide t'_s by t_s will be training set that build a prediction model. Therefore, new training sets will be defined:

$$N = \frac{t'_s}{t_s} \tag{6}$$

$$S = \{s_{t-N}, s_{t-N+1}, \ldots, s_t\} \tag{7}$$

In above equation, S is a list of support sets.

The proposed method predicts events in that is given in the training sets, however, does not reduce former training sets. In this section, how to implement the future prediction will be stated.

In this case, a next state $\hat{x}_{t+1,i}, i \in \dim \hat{\mathbf{x}}_{t+1}$ (i denotes an element of all the robot's state) is estimated by using the state and action are defined by $\mathbf{z}_t = \begin{bmatrix} x_{t,1} & \ldots & x_{t,n} & | & a_t \end{bmatrix}$. Therefore, this vector $\mathbf{z}_t$ is an $(n+1)\times 1$ vector. Next, let's consider the sum-of-squares error function J_S from training set $\{\mathbf{x}_j, y_j\}$ described by the SVR model $y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$ [8].

$$J_S(\mathbf{w}) = \frac{1}{2} \sum_{j=t-N}^{t} \left\{\mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j\right\}^2 + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} \quad (\lambda \geq 0) \tag{8}$$

where $\mathbf{w}^\top$ indicates the transpose of $\mathbf{w}$. Here, λ represents the regularization parameter, and $\mathbf{w}$ represents the weight matrix of the SVR model. The weight matrix $\mathbf{w}$ is found by setting the gradient for minimizing the sum-of-squares error function J_S to zero (thus,

$\partial J_S(\mathbf{w})/\partial \mathbf{w} = 0$). Hence,

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} J_S(\mathbf{w}) =& 2 \times \frac{1}{2} \sum_{j=t-N}^{t} \left[\left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \right. \\ & \left. \phi(\mathbf{x}_j) \right] + \frac{\lambda}{2}\mathbf{w} + \frac{\lambda}{2}\mathbf{w} = 0 \\ 0 =& \sum_{j=t-N}^{t} \left[\left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \right. \\ & \left. \phi(\mathbf{x}_j) \right] + \lambda \mathbf{w} \\ \mathbf{w} =& -\frac{1}{\lambda} \sum_{j=t-N}^{t} \left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \phi(\mathbf{x}_j) \\ =& \sum_{j=t-N}^{t} a_j \phi(\mathbf{x}_j) = \Phi^\top \mathbf{a}\end{aligned} \tag{9}$$

$$\begin{aligned} \textit{where } \mathbf{a} &= \begin{bmatrix} a_{t-N} & \dots & a_t \end{bmatrix}^\top, \\ a_j &= -\frac{1}{\lambda} \left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \end{aligned}$$

Now, Φ is called the design matrix, and the j-th row is described by $\phi(\mathbf{x}_j)^\top$. Here, the parameter vector $\Phi\mathbf{a}$ replaces $\mathbf{w}$,

$$\begin{aligned} J(\mathbf{a}) =& \frac{1}{2}\mathbf{a}^\top \Phi\Phi^\top\Phi\Phi^\top \mathbf{a} - \mathbf{a}^\top \Phi\Phi^\top \mathbf{y} \\ & + \frac{1}{2}\mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2}\mathbf{a}^\top \Phi\Phi^\top \mathbf{a} \end{aligned} \tag{10}$$

Now, the Gramian matrix $\mathbf{K} = \Phi\Phi^\top$ will be defined. Here, the matrix coefficient of $\mathbf{K}$ is given by

$$K_{jm} = \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}_m) = k(\mathbf{x}_j, \mathbf{x}_m) = Q_{jm} \tag{11}$$

This matrix coefficient is the symmetric matrix as a kernel matrix. Now, let's rearrange the sum-of-squares error function J_S by using the Gramian matrix:

$$\begin{aligned} J_S(\mathbf{a}) =& \frac{1}{2}\mathbf{a}^\top \mathbf{KKa} - \mathbf{a}^\top \mathbf{Ky} \\ & + \frac{1}{2}\mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2}\mathbf{a}^\top \mathbf{Ka} \end{aligned} \tag{12}$$

The equation is rearranged by isolating $\mathbf{a}$:

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \tag{13}$$

Here, $\mathbf{I}_N$ represents the $N \times N$ identity matrix. Therefore, the prediction result $\hat{y}(\mathbf{x})$ for the SVR model to input $\mathbf{x}$ can be derived the equation anew as

$$\begin{aligned} \hat{y}(\mathbf{x}) &= \mathbf{w}\phi(\mathbf{x}) + b = \mathbf{a}^\top \Phi\phi(\mathbf{x}) + b \\ &= \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{y} + b \end{aligned} \tag{14}$$

where $\mathbf{k}(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_{t-N}, \mathbf{x}) & \dots & k(\mathbf{x}_j, \mathbf{x}) \end{bmatrix}^\top$

In this time, prediction result and Kernel matrix will be updated as below:

$$\begin{aligned} \hat{y}(\mathbf{x}) &= \mathbf{w}\phi(\mathbf{x}) + b = \mathbf{a}^\top \Phi\phi(\mathbf{x}) + b \\ &= \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{y} + b \end{aligned} \tag{15}$$

where $\mathbf{k}(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_{t-N}, \mathbf{x}) & \dots & k(\mathbf{x}_t, \mathbf{x}) \end{bmatrix}^\top$

$$Q = \begin{bmatrix} Q_{s_{t-N}, s_{t-N}} & \cdots & Q_{s_{t-N}, s_t} \\ \vdots & \ddots & \vdots \\ Q_{s_t, s_{t-N}} & \cdots & Q_{s_t, s_t} \end{bmatrix} \tag{16}$$

The matrix Q contains the values of kernel function and it is called kernel matrix. In these equations, learning space Q and training sets $\mathbf{x}$ will be re-construct each sampling time and adding new training data. Therefore, the learning space and training sets will be reduced each sampling time. As mentioned before, the speed of learning depends mostly on the number of support vectors, that can influence significatively performances. As a result, the speed of learning will be improved than former works.

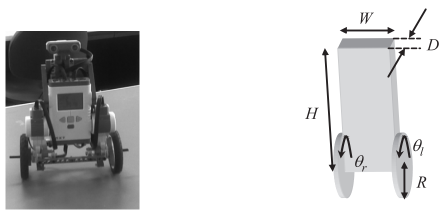
3 THE VERIFICATION EXPERIMENT – COMPUTATIONAL SIMULATION USING THE PROPOSED METHOD

3.1 Outline of the Experiment

In this experiment, we stabilize the posture of a two-wheeled self-propelled inverted pendulum "NXTway-GS" (fig. 5) as an application, using the computer simulation. In this verification experiment, we compared the control response of the proposed method with the ordinary method. Furthermore, in proposed method, the predictor only used the proximate predicted result repeatedly training data from 0 [s] (don't reduce) or training sets that reduced in each sampling time, for postural control.

3.2 Simulation Setup - the NXTway-GS Model

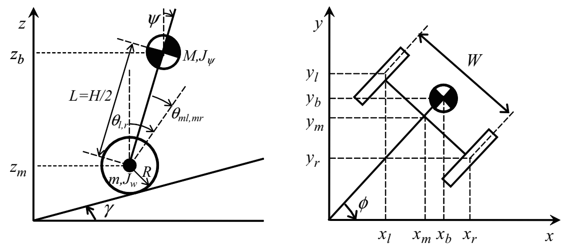
NXTway-GS (fig. 5) can be considered as inverted pendulum model shown in fig. 6. Figure 6 shows the side view and the plane view of the model. The coordinate system used in 3.3 is described in fig. 6. In figure 6, ψ denotes the body pitch angle and $\theta_{ml,mr}$ denotes the DC motor angle (l and r indicate *left* and *right*). The physical parameters of NXTway-GS are listed in table 1.



(a) NXTway-GS

(b) Model of NXTway-GS

Figure 5. Two-wheeled Inverted Pendulum "NXTway-GS"



(a) Side view

(b) Plane view

Figure 6. The Side View and the Plane View of NXTway-GS [10]-[11]

3.3 Simulation Setup - Modeling of the NXTway-GS

We can derive the equations of motion of the inverted pendulum model using the Lagrange equation based on the coordinate system in fig. 6. If the direction of model is in the x-axis positive direction at $t = 0$, the equations motion for each coordinate are given as ([10]-[11]) ;

$$\left[(2m+M)\,R^2 + 2J_w + 2n^2J_m\right]\ddot{\theta} + (MLR \\ -2n^2J_m)\,\ddot{\psi} - Rg\,(M+2m)\sin\gamma = F_\theta \tag{17}$$

$$\left(MLR - 2n^2J_m\right)\ddot{\theta} + \left(ML^2 + J_\psi \right. \\ \left. +2n^2J_m\right)\ddot{\psi} - MgL\psi = F_\psi \tag{18}$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}\left(J_w + n^2J_m\right)\right]\ddot{\phi} = F_\phi \tag{19}$$

Here, we consider the following variables $\mathbf{x}_1, \mathbf{x}_2$ as the state variables and $\mathbf{u}$ as the input variable ($\mathbf{x}^\top$ indicates the transpose of $\mathbf{x}$.).

$$\mathbf{x}_1 = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^\top \tag{20}$$

$$\mathbf{x}_2 = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top \tag{21}$$

$$\mathbf{u} = \begin{bmatrix}v_l & v_r\end{bmatrix}^\top \tag{22}$$

Consequently, we can derive the state equations of the inverted pendulum model from eq. (17), (18) and (19).

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \tag{23}$$

$$\frac{d}{dt}\mathbf{x}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \tag{24}$$

In this paper, we only use the state variables $\mathbf{x}_1$. Because $\mathbf{x}_1$ is including body pitch angle as important variables ψ and $\dot{\psi}$ for control of self-balancing, and we will not consider plane motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$).

3.4 Simulation Setup - How to Apply the Online SVR to the State Predictor

In this method, we use Online SVR [9] as a learner. Moreover, we applied RBF kernel [13] as the kernel function to the Online SVR of the learner. The RBF kernel on two samples $\mathbf{x}$ and $\mathbf{x}'$, represented as feature vectors in some input space, is defined as

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\beta\,||\mathbf{x} - \mathbf{x}'||^2\right) \tag{25}$$

And the learning parameters of Online SVR are listed in table 2. In table 2, $i \in \{1, 2, 3, 4\}$.

ISBN: 978-1-941968-42-0 ©2017 SDIWC

Table 1. Physical Parameters of the NXTway-GS

Symbol	Value	Unit	Physical property
g	9.81	[m/s^2]	Gravity acceleration
m	0.03	[kg]	Wheel weight [10]
R	0.04	[m]	Wheel radius
J_w	$\frac{mR^2}{2}$	[kgm^2]	Wheel inertia moment
M	0.635	[kg]	Body weight [10]
W	0.14	[m]	Body width
D	0.04	[m]	Body depth
H	0.144	[m]	Body height
L	$\frac{H}{2}$	[m]	Distance of center of mass from wheel axle
J_ψ	$\frac{ML^2}{3}$	[kgm^2]	Body pitch inertia moment
J_ϕ	$\frac{M(W^2+D^2)}{12}$	[kgm^2]	Body yaw inertia moment
J_m	1×10^{-5}	[kgm^2]	DC motor inertia moment [11]
R_m	6.69	[Ω]	DC motor resistance [12]
K_b	0.468	[V·s/rad.]	DC motor back EMF constant [12]
K_t	0.317	[N·m/A]	DC motor torque constant [12]
n	1	[1]	Gear ratio [11]
f_m	0.0022	[1]	Friction coefficient between body and DC motor [11]
f_W	0	[1]	Friction coefficient between wheel and floor [11]

Table 2. Learning Parameters of the Online SVR

Symbol	Value	Property
C_i	300	Regularization parameter or predictor of x_i
ϵ_i	0.02	Error tolerance for predictor of x_i
β_i	30	Kernel parameter for predictor of x_i

3.5 Simulation Setup - How to Apply the Linear-quadratic Regulator to the Action Predictor

In this experiment, we apply LQR (Linear-quadratic Regulator) as an action prediction (And a predictor). So we design the controller as an action predictor based on modern control theory. This LQR calculates the feedback gain $\mathbf{k}_f$ so as to minimize the cost function J_C given as the following;

$$J_C = \int_0^\infty \left[\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)\right] dt \tag{26}$$

The tuning parameter is the weight matrix for state $\mathbf{Q}$ and for input $\mathbf{R}$. In this paper, we choose the following weight matrix $\mathbf{Q}$ and $\mathbf{R}$;

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^2 \end{bmatrix} \tag{27}$$

$$\mathbf{R} = 1 \times 10^3 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{28}$$

Then, we obtain the feedback gain $\mathbf{k}_f$ from minimizing J_C. Therefore, we apply $\mathbf{k}_f$ as an action predictor [3]. And hence, in this experiment, we do not consider the plane move of the two-wheeled inverted pendulum. In other words, we consider that $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$.

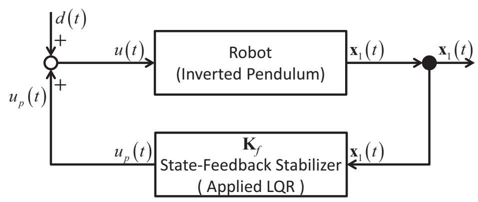


Figure 7. Control Input Obtained by Mixing the Action and Disturbance Inputs

3.6 Conditions of Simulation - Acquiring the Training Sets

In this experiment, we mix the action signal with a known disturbance signal $\mathbf{d}(t)$ (figs. 7, 8, and 9), and $\mathbf{d}(t)$ is given as

$$d(t) = A_{d1} \sin(2\pi f_{d1} t) \tag{29}$$

Then, the signal $\mathbf{d}(t)$ will be mixed to the

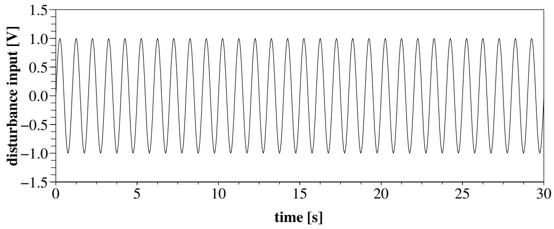


Figure 8. Disturbance Signal in Control Inputs $\mathbf{d}(t)$ (overall)

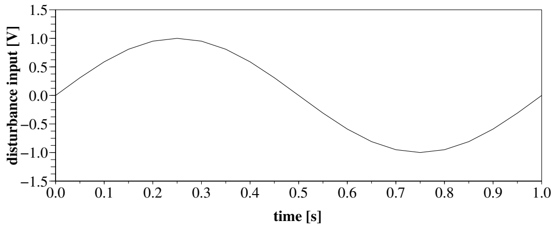


Figure 9. Disturbance Signal in Control Inputs $\mathbf{d}(t)$ (focused on 0 [s] to 1[s])

model. Herewith, we can acquire the training sets from the two-wheeled inverted pendulum. In figures 10 to 12 shows training sets that were obtained from the computer simulation of the stabilize control of the two-wheel inverted pendulum. Moreover, the properties of disturbance that we provide as input and other conditions of a simulation are listed in table 3.

3.7 Simulation Results

Figures 10 and 11 show compensation results of the state of $\mathbf{x}_1$, and fig. 12 shows the prediction of the control input and compensation input using prediction result of $\mathbf{u}$.

In this section, we will not consider the part that is given in real training sets. Thus we will only argue and focus on the part of the graph pertaining to the state predicted part shown in T (at $t = 3.00$ [s]) of figs. 10 and 11.

3.8 Discussion on Simulated Results

In here, starting and predicting the state predicted point is shown at $t = 3.00$ [s] as shown in T.

According to these results (figs. 10 through 12), compensation results using the proposed

Table 3. Parameters for Condition of a Simulation

Sybmol	Value	Unit	Physical property
ψ_0	0.0262	[rad.]	Initial value of body pitch angle
γ_0	0.0	[rad.]	Slope angle of movement direction
t_s	0.05	[s]	Sampling rate
$t_{d,\mathrm{start}}$	0.0	[s]	Start time of application of predictable disturbance
$t_{d,\mathrm{finish}}$	45.0	[s]	Finish time of application of predictable disturbance
A_{d1}	1.0	[V]	Amplitude of predictable disturbance
f_{d1}	1.0	[Hz]	Frequency of predictable disturbance
N_s	60	—	Initial dataset length
$N_{\max}$	241	—	Maximum dataset length for the prediction
N	20	—	Step size of outputs for N-ahead State-action Pair Predictor's outputs
α_i $i \in N$	$\frac{N-i+1}{100N}$	—	Weight coefficients for $\hat{u}(t+i), i \in N$ (for the contrast experiment [4])

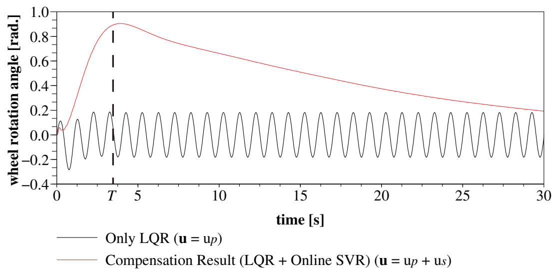


Figure 10. Control Response of the Wheel Rotation Angle θ (1)

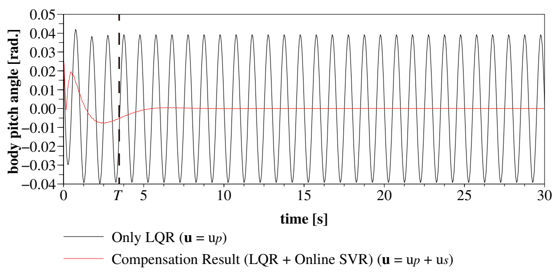


Figure 11. Control Response of the Body Pitch Angle ψ (1)

ISBN: 978-1-941968-42-0 ©2017 SDIWC

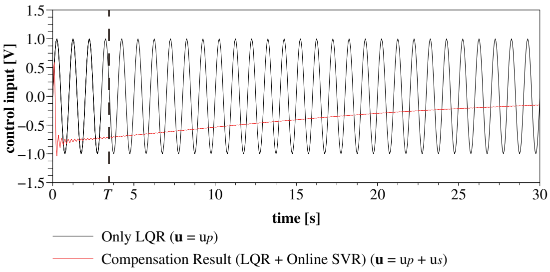


Figure 12. Control Response of the Control Input u (1)

method (are described in red solid line) are approaching to zero, with time. Next, we will focus on each result.

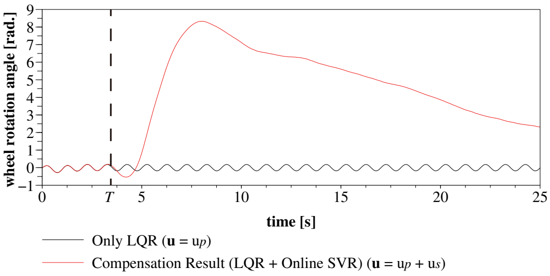


Figure 13. Control Response of the Wheel Rotation Angle θ (2) [4]

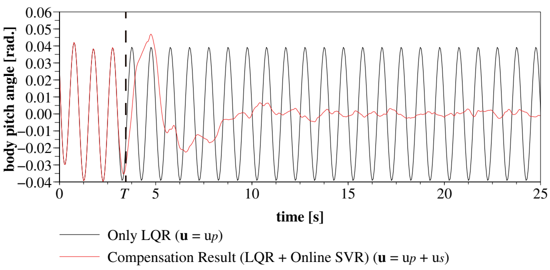


Figure 14. Control Response of the Body Pitch Angle ψ (2) [4]

Next, let's compare the results between proposed method and the results that showed in study [4]. Figure 13 through 15 show compensation results of the state of $\mathbf{x}_1$, and fig. 12 shows the compensation input using prediction result of $\mathbf{u}$, that used the method we proposed in a study [4]. Comparing each state and control inputs, it can be said that the convergence speed of experimental results is earlier than the former study.

Now let's focus on these results. In this study, the learning space and training sets will be changed "suddenly," when the period of the disturbance signal will be estimated. In this time, the current action and states are also suddenly changed by the compensation control input based on the former learning space and training sets. In a former study, training sets and learning space are including much prediction error, in phase of early training. Therefore, in the mechanism of State-action Pair Prediction, it will use former training sets, and will add new training sets to current learning space. Namely, results of prediction and revised action will be influenced by past prediction error. On the other hand, in the proposed method, the kernel matrix will ignore early learning results and early compensation result. In other words, it use results including less affection of prediction error, according to the time elapsed by the training set of out of the disturbance tendency period will not be used. Further, this system acquires the data each at the sampling times. Using these changed results, the proposed system derives an action that multiplied states to the optimal feedback gain for the future state. As a result, this system is stabilizing the inverted pendulum using current outside data and previous states and an action. As a result, we can be said that internal states and an action will converge to zero, according time course. From these viewpoints, we conclude the experimental results are reasonable.

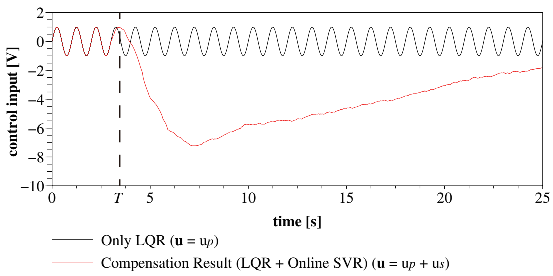


Figure 15. Control Response of the Control Input u (2) [4]

4 CONCLUSION

In this paper, the relationship between learning space and training sets for prediction and frequency of the disturbance signal that given by outside environment will be focused on. To achieve this problem, on the basis of former our works, we proposed the method that reducing training sets and learning space, for prediction, based on prediction results that obtained by recent tendency of disturbance frequency, dynamically by using Nyquist-Shannon sampling theorem. Applying this proposed method, it was obtained that the body pitch angle of NXTway-GS was converged to zero, with time. In other words, the compensated action for rapid convergence was obtained, similarly as a normal training sets and learning space method.

From the verification experimental results, the proposed method could be converged to a desirable state as similar as fixed training sets. To be more specific, the slope of the body pitch angle of NXTway-GS will be converged to zero, based on state and action prediction and decision. Accordingly, the proposed method can be adapted to the situation of frequency property of disturbance will be concluded. From results of verification experiments, it can be concluded that the proposed system can predict what be defined by training sets and learning space that can be obtained the property of a disturbance signal, based on the Nyquist-Shannon Sampling Theorem. In addition, as a future work, we will confirm the response of the proposed system on actual robot.

REFERENCES

[1] P. Pivoňka, V. Veleba, M. Šeda, P. Ošmera, and R. Matoušek, "The Short Sampling Period in Adaptive Control," In Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II WCECS, pp.724-729, 2009.

[2] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," Artificial Intelligence, Vol. 97 No. 1-2, pp.245-271, 1997.

[3] M. Sugimoto and K. Kurashige, "A Study of Effective Prediction Methods of the State-action Pair for Robot Control using Online SVR," Journal of Robotics and Mechatronics, Vol. 27 No. 5, pp.469-479, 2015.

[4] M. Sugimoto and K. Kurashige, "Real-time Sequentially Decision for Optimal Action using Prediction of the State-Action Pair," In Proceedings of 2014 International Symposium on Micro-NanoMechatronics and Human Science, pp.199-204, Nov.9-12, Nagoya, Japan, 2014.

[5] M. Sugimoto and K. Kurashige, "Future Motion Decisions using State-action Pair Predictions," International Journal of New Computer Architectures and their Applications, Vol. 5 No. 2, pp.79-93, 2015.

[6] R. Koggalage and S. Halgamuge, "Reducing the Number of Training Samples for Fast Support Vector Machine Classification," Neural Information Processing – Letters and Reviews, Vol. 2, No. 3, pp.57-65, 2004.

[7] G. Bontempi, Machine Learning Strategies for Time Series Prediction, Machine Learning Summer School, 2013. [Online]. Available: http://www.ulb.ac.be/di/map/gbonte/ftp/time_ser.pdf

[8] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 2006.

[9] F. Parrella, Online Support Vector Regression. PhD thesis, Department of Information Science, University of Genoa, Italy, 2007.

[10] M. Sugimoto, H. Yoshimura, T. Abe, and I. Ohmura, "A Study on Model-Based Development of Embedded System using Scilab/Scicos," In Proceedings of the Japan Society for Precision Engineering 2010 Spring Meeting, Saitama, D82, pp.343-344, 2010.

[11] Y. Yamamoto, NXTway-GS Model-Based Design –Control of self-balancing two-wheeled robot built with LEGO Mindstorms NXT–. CYBERNET SYSTEMS CO.,LTD., 2009.

[12] R. Watanabe, Ryo's Holiday LEGO Mindstorms NXT, 2008.

[13] Y. Chang, C. Hsieh, K. Chang, M. Ringgaard, and C. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," J. Machine Learning Research, vol.11, pp.1471-1490, 2010.