https://doi.org/10.20965/jrm.2015.p0469

Paper:

A Study of Effective Prediction Methods of the State-Action Pair for Robot Control Using Online SVR

Masashi Sugimoto and Kentarou Kurashige

Muroran Institute of Technology
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan
E-mail: sm04322@gmail.com, kentarou@csse.muroran-it.ac.jp
[Received April 16, 2015; accepted July 1, 2015]

In order to work effectively, a robot should be able to adapt to different environments by deciding its correct course of action according to the situation, using determinants other than pre-registered commands. For this purpose, the ability to predict the future state of a robot would be effective. On the other hand, the future state of a robot varies infinitely if it depends on its current action. Therefore, it is difficult to predict only the future state. Thus, it is important to simultaneously predict the state and the action that the robot will adopt. The purpose of this study was to investigate the prediction of the advanced future state and action of a robot. In this paper, the results of the study are reported and methods that allow a robot to decide its appropriate behavior quickly, according to the predicted future state are discussed. As an application example for evaluating the proposed method, the inverted pendulum model is used and the prediction results are compared with the robot's actual responses. Then, two methods will be discussed for predicting the robot's state and action. To perform state and action prediction, two methods are used, firstly the Online SVR (Support Vector Regression) and secondly Online SVR and the LQR (Linear Quadratic Regulator).

Keywords: online state prediction, internal state space, learning using combination of state space and action, prediction using combination of state space and action, mobile robot

1. Introduction

Over the years, many studies have been conducted with the objective of facilitating the working of robots in dynamic environments [1–9]. Various robots have been developed to assist humans in workspaces, such as a house or factory [10]. In general, robots are required to work effectively and safely in a dynamic environment to achieve their tasks. However, it is not easy to make a robot behave like a human in dynamic environments [11, 12]. When they are working in a certain environment, humans select an appropriate course of action through subconsciously predicting all the changes in the environment and their next state.

Humans subconsciously use their past experience and memory to predict the posture and force required in certain environments [13]. That is, they would find it difficult to consciously perform these actions. Because it is not easy to deal, consciously with a current situation, sometimes we cannot accomplish our objective, and hence, there are cases where we sustain a loss. For example, human walking is rhythmic and stable, because appropriately according to the sensory input related to their environment body is adjusted. The brain should recognize the act of walking and the environment and accordingly adjust each joint of the body so that their movement to the environment is adopted, without them realizing it. In this process, in their ordinary daily life humans use their predictive abilities to control their body balance appropriately in order to reduce their risk of falling or to avoid colliding with an obstacle [13].

Similarly, in the case of robots, if prediction is not applied, the load of control processing for behavior selection is considered to be large. For example, the manipulated variable will increase the sampling time of the controller. In these problems, most of many studies use the machine learning, such as Reinforcement Learning (RL) that acquire the optimal action to learn the environment by trial and error [14, 15]. Or possibly they use the Model Predictive Control (MPC) that is most suitable input sequentially gained by each time, that is much better for well-generalized to use as control rule [16, 17]. However, these techniques have a problem to be debatable, that is computation delay and the hardware overhead, or whether can respond flexibly to changes in the dynamic environment in the working [18–22]. On the other hand, there are some techniques to be presented that generate the robot motion. In this case, combine with ordinary control rule and EKF (Extended Kalman Filter) [1] or UKF (Unscented Kalman Filter) [1], for avoid to linearize the model of the robot [23–25]. However, in these techniques, some problems are still remaining; the case that applied filter will become often unstable, the case that using non-linear model is not easy, the case that the parameter that should be defined are increasing. Moreover, it is difficult to decide the effective action by using the robot's current sensory input, and therefore, this input may adversely affect robot's task performance. In recent years, a robot was de-

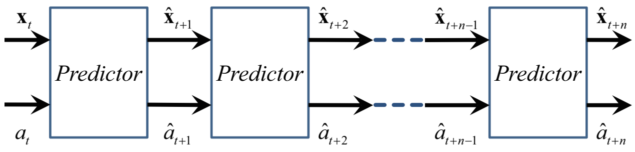


Fig. 1. Overall view of proposed method.

veloped with advanced behavior characteristics; humans control the robot through some control rules. However, it is expected that in the future these control rules will be incorporated in the robot, through machine learning in dynamic environments, and robots that are required to support human labor will not be controlled by a fixed precontrol rule. A robot should decide each action to be taken in a dynamic environment adaptive, in addition to the preregistered commands, like a human. Moreover, hardware and the limited computational resources of a robot pose a physical limitation, and therefore, it needs some time to decide its course of action; for example, whether it should take one or more steps. Thus, state prediction is important, when robots have to achieve a task or support people in a dynamic environment and work more flexibly. In a previous study related to this issue, machine learning was applied, through which a robot autonomously learned the appropriate actions for certain environments.

In this paper, the results of these studies will be presented and methods that allow a robot to decide its appropriate behavior quickly, using the predicted state, are discussed. To realize this, the purpose of this research is to propose an advance prediction method using the Online SVR (Support Vector Regression) as a predictor. To achieve this, an Online SVR will be used and it will be improved that can predict state and action sequentially. Accordingly, the state and the action of the distant future by repeatedly using the predictor of the proposed method can be predicted. An overview of the proposed method is shown in **Fig. 1**. This Online SVR predicts the robot's future state, i.e., the robot's next state, and appropriate future course of action. Furthermore, this predictor facilitates the prediction of the robot's distant future state, using the states and actions that the robot adopts repeatedly. Using this method, the system that allows a robot to decide its course of action, can be realized.

In this section, that was started, to allow robots to achieve their task in a dynamic environment, prediction of their future state and action is required. In Section 2, former future prediction techniques for robots and the problems that they entail will be described. In Section 3, the details of the proposed method will be provided. In Section 4, the experimental setup will be described. Finally, in Section 5, the conclusions of this study will be presented.

2. Problems Related to Former Future Prediction Techniques for Robots

In previous studies, prediction for robots was achieved by indirect means. The research studies that have addressed prediction for the control of a robot can be divided into three types: first, studies on applying predictive control to a flying robot or manipulator using a plant model [16, 17, 26], second, studies on modular reinforcement learning using a multiple state prediction model in combination with a reward predictor model [14] and third, studies on SVR (Support Vector Regression) based obstacle avoidance and the control of a two-wheeled mobile robot [27, 28].

As mentioned above, these studies did not address the future state and action of the robot. In other words, it is difficult to predict the future state by considering the current action. The reason for this is that robots detect the current environment, and accordingly, take one course of action. The current state and action change the environment to a new state, and the robots receive a reward. Through these mutual interactions, robots learn the appropriate action required to perform a given task [15]. For this reason, the robot's action, which is acquired using experiences gained by machine learning, is influenced by determinism and the law of causality of the relationship between the environment and the robot's action.

This means the action taken by a robot when it receives a certain input depends also on the input of the current situation. Here, the selected action pertains are shown to a certain situation obtained through machine learning, including the state of the robot. Let us consider driving a car to a destination. The movement of a car conforms to physical laws. However, the driver must decide whether to go straight or turn at an intersection, and this decision depends on his/her past driving experiences. This means, an action selected by the driver influences his/her next state and environment. The same applies in the case of controlling robots. In other words, an action selected by the robot influences its next state and environment. Hence, the state of the robot varies infinitely depending on the action that the robot selects from multiple choices, subject to given conditions.

Figure 2 shows the relationship between the future state and the action. The future state depends on the current action of the robot. However, in previous studies this relationship was not considered. That is, using the methods proposed in previous studies it is difficult to predict the future state including the action. Because of these characteristics, the selection of the state of the robot depends on the course of action that it takes to achieves its tasks. Thus, a particular problem arises in that the action selected by a robot influences the future. For this reason, an approach will be needed that includes state prediction in which the action that the robot has taken is considered as mentioned earlier.

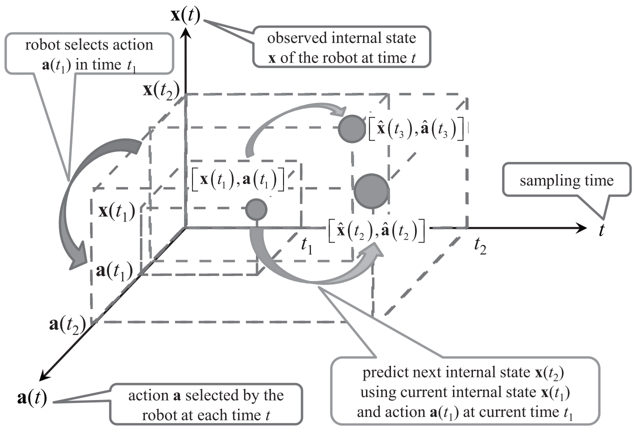


Fig. 2. Prediction of future state and action from current state.

3. Proposed Future Prediction Technique for Robots

A scheme for predicting the future state of a robot, to meet the requirements of action in the unknown environment will be proposed. In this proposed method, in the future the robot uses the state and action that it adopts as it acquires an action in the unknown environment. Here, the appropriate control rule using prediction results is attempted to determine. If a general learning method is employed, robots need to undergo many trial and error iterations to determine the control rule themselves. However, learning using real robots will be assumed, although they have a few degrees of freedom (DOFs). The endurance of the robot is considered, and then, the purpose of this study is to learn the task in fewer trials. This would allow robots to work more effectively.

In this section, two kinds of approach for predicting the future state of a robot will be proposed. In this case, firstly a method that adds sequential prediction to Online SVR [29] will be considered, and predict the state and action using the current action and state of all the past states until the present time. And second, a combination of Online SVR and LQR (Linear Quadratic Regulator) will be considered. In this study, this knowledge is used and virtual learning using a mathematical model is implemented. Then, a method based on an experimental procedure for acquiring the procedure for executing the required task using the prediction results will be proposed.

In this study, the posture of a self-propelled inverted pendulum will be stabilized. The control of the motion, such as stabilizing the inverted posture, is described by a second-order nonlinear differential equation. However, in general, it is difficult to derive or determine the control rule in theory. Therefore, the control rule is needed to acquire using machine learning. Therefore, to predict an orbit of the states and the action of the inverted pendulum using machine learning to acquire the control rule will be needed.

3.1. Online SVR Extension

In this study, the Online SVR is used for constructing the proposed system. However, this technique cannot be applied directly to the method for predicting the state and action. Hence, the Online SVR was extended what is applied to this study. In particular, the input of Online SVR, and additional learning were focused to extend.

3.1.1. Online SVR Input Extension

The Online SVR technique does not recognize the concept of state and action. In other words, this technique does not accommodate a multiple input set. Therefore, in this study, the input of the Online SVR was trying to re-interpret the concept of former research to suit the proposed method, and was re-defined as following:

$$\mathbf{z}_t = \begin{bmatrix} x_{t,1} & x_{t,2} & \cdots & x_{t,n} & a_{t-1,1} & a_{t-1,2} & \cdots & a_{t-1,n} \end{bmatrix} \quad (a_{t-1} = 0 \text{ when } t = 0) \qquad (1)$$

In this paper, the notation is defined as

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{x}_t & | & \mathbf{a}_{t-1} \end{bmatrix}. \qquad (2)$$

Here, $\mathbf{x}_t$ represents the state of a robot at time t and $\mathbf{a}_t$ represents the action that the robot performs at time t. In this study, it is assumed that the action that the robot performs consists of moving forward or backward in a one-dimensional coordinate system.

3.1.2. Sequential Prediction: Prediction of the Next State and Action Using New Samples

The Online SVR method predicts events in that is given in the training sets, and hence, does not predict events for areas that are out of the range of the training sets. In this section, the dual representations of SVR [30] is focused on, and how to implement the future prediction is 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 that are defined by $\mathbf{z}_t = \begin{bmatrix} x_{t,1} & \cdots & x_{t,n} & | & a_t \end{bmatrix}$. Therefore, this vector $\mathbf{z}_t$ is an $(n+1) \times 1$ vector. Next, let's consider the sum-of-squares error function J from training set $\{\mathbf{x}_j, y_j\}$ described by the SVR model $y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$ [30].

$$J(\mathbf{w}) = \frac{1}{2} \sum_{j=1}^{t} \left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\}^2 + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \quad (\lambda \geq 0) \qquad (3)$$

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 to zero (thus, $\partial J(\mathbf{w})/\partial \mathbf{w} = 0$). Hence,

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = 2 \times \frac{1}{2} \sum_{j=1}^{t} \left[\left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \phi(\mathbf{x}_j) \right] + \frac{\lambda}{2} \mathbf{w} + \frac{\lambda}{2} \mathbf{w} = 0$$

$$0 = \sum_{j=1}^{t} \left[\left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \phi(\mathbf{x}_j) \right] + \lambda \mathbf{w}$$

$$\mathbf{w} = -\frac{1}{\lambda} \sum_{j=1}^{t} \left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\} \phi(\mathbf{x}_j)$$

$$= \sum_{j=1}^{t} a_j \phi(\mathbf{x}_j) = \Phi^\top \mathbf{a} \quad (4)$$

where $\mathbf{a} = \begin{bmatrix} a_1 & \cdots & a_t \end{bmatrix}^\top$,

$$a_j = -\frac{1}{\lambda} \left\{ \mathbf{w}^\top \phi(\mathbf{x}_j) + b - y_j \right\}.$$

Now, Φ is called the design matrix [31], and the j-th row is described by $\phi(\mathbf{x}_j)^\top$. Here, the parameter vector $\Phi\mathbf{a}$ replaces $\mathbf{w}$,

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^\top \Phi\Phi^\top \Phi\Phi^\top \mathbf{a} - \mathbf{a}^\top \Phi\Phi^\top \mathbf{y} + \frac{1}{2}\mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2}\mathbf{a}^\top \Phi\Phi^\top \mathbf{a}. \quad (5)$$

Now, the Gramian matrix $\mathbf{K} = \Phi\Phi^\top$ will be defined [32]. Here, the matrix coefficient of $\mathbf{K}$ is given by

$$K_{jm} = \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}_m) = k(\mathbf{x}_j, \mathbf{x}_m). \quad (6)$$

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

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^\top \mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^\top \mathbf{K}\mathbf{y} + \frac{1}{2}\mathbf{y}^\top \mathbf{y} + \frac{\lambda}{2}\mathbf{a}^\top \mathbf{K}\mathbf{a}. \quad (7)$$

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

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_t)^{-1} \mathbf{y}. \quad (8)$$

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

$$\hat{y}(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x}) + b = \mathbf{a}^\top \Phi \phi(\mathbf{x}) + b$$

$$= \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbf{I}_t)^{-1} \mathbf{y} + b \quad (9)$$

where $\mathbf{k}(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}) & \cdots & k(\mathbf{x}_j, \mathbf{x}) \end{bmatrix}^\top$.

Here, b_i is a bias term, and $\mathbf{k}$ represents the mapping function for calculating the inner product into the feature space. Then, this relationship is [30, 33]

$$b = \varepsilon + x_t - \sum_{j=1}^{t} \theta_j k\left(\mathbf{x}_j^\top \mathbf{x}_j\right). \quad (10)$$

Here, ε represents the dead zone of the ε-insensitive loss function [33, 34] as

$$\xi(\mathbf{r}) = \begin{cases} 0 & \text{if } |\mathbf{r}| < \varepsilon \\ |\mathbf{r}| - \varepsilon & \text{otherwise} \end{cases} \quad (11)$$

Here, $\mathbf{r}$ represents the residue. From the above, the equations for the state prediction is derived.

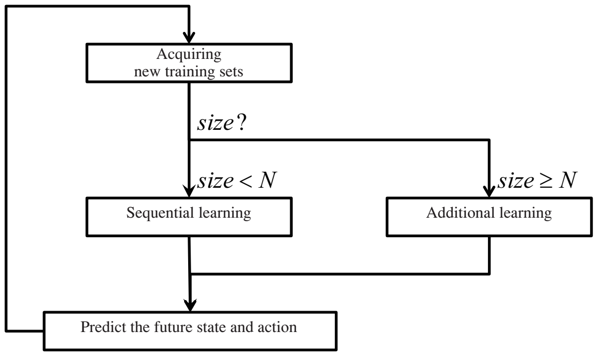


Fig. 3. Overview of the sequential prediction using incremental learning.

$$\hat{x}_{t+1,i} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}({}^{x}\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda \mathbf{I}_l)^{-1} \, {}^{x}\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \quad (12)$$

$$\text{when } i \in \dim \mathbf{x}_t$$

$$\hat{a}_{t+1} = \begin{cases} 0 & \text{if } t = 0 \\ \Delta\theta & \text{if } t = 1 \\ \mathbf{k}_{sv}({}^{a}\mathbf{z}_t)^\top (\mathbf{K}_{sv} + \lambda \mathbf{I}_l)^{-1} \, {}^{a}\mathbf{z}_{sv} + b' & \text{otherwise} \end{cases} \quad (13)$$

From the above, the equations of sequential prediction will be defined. Here, b'_i is a bias term for x_i (the i-th element of $\mathbf{x}$), b' is a bias term in the predictor of a_t, $\Delta\theta$ represents the Lagrange multiplier, l represents the number of the former support vector $\mathbf{z}_{s_k} (k \in l)$, λ represents the regularization parameter, $\mathbf{I}_l$ represents the $l \times l$ identity matrix, $\mathbf{K}_{sv}$ represents the Gramian matrix, and $\mathbf{k}_{sv}$ is the mapping matrix.

Here, ${}^{x}\mathbf{z}_t$ is defined by state $\mathbf{x}_t$ and the pair $\mathbf{z}_t$: ${}^{x}\mathbf{z}_t = \begin{bmatrix} \mathbf{z}_t & | & \mathbf{x}_t \end{bmatrix}$ and ${}^{a}\mathbf{z}_t$ is defined by action a_{t-1} and the pair $\mathbf{z}_t$: ${}^{a}\mathbf{z}_t = \begin{bmatrix} \mathbf{z}_t & | & a_{t-1} \end{bmatrix}$. Hence, this extended method predicts the state and action at each time.

3.1.3. Additional Learning: Learning New Samples Using Incremental Learning

In the previous subsection, the Online SVR is shown what predicts events only the area that is given in the training sets, and this method cannot predict events in an area out of the range of the training sets. In this case, Online SVR does not complete the learning and the prediction until the training sets reach the fixed data length. Hence, it is assumed that the Online SVR cannot predict the future state. In this section, how to extend the fixed data length using incremental learning is focused on.

In this case, the incremental learning will be considered that combine with increasing the training set. For example, the data length N is defined. The learning and prediction are performed using the $(N-1)$-th and N-th training sets (**Fig. 3**).

In Online SVR, the learning parameters are updated

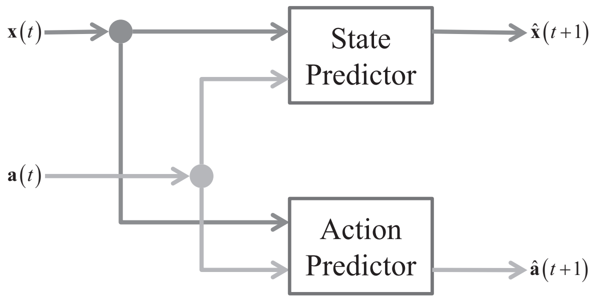


Fig. 4. Outline of the state and the action prediction system using only Online SVR.

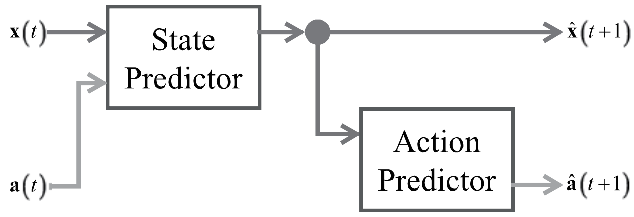


Fig. 5. Outline of state and action prediction system of using Online SVR and LQR.

when the learning is performed in the $(N-1)$-th training data. That is, the Online SVR model abstracts the pattern of the training set. In this case, the weight vector $\mathbf{w}_{N-1}$ and the bias term b were used in Online SVR.

In this study, this point will be focused on. The sequential prediction for the future using the $(N-1)$-th and N-th training sets will be considered. Hence, the incremental learning of batch SVR will be introduced.

Consequently, it is assumed that the learning parameters were updated, and the support vector set S_{N-1}, weight vector $\mathbf{w}_{N-1}$ and bias term b are defined in the Online SVR model. Now, the following equation will be minimized in the case where the N-th data will be given.

$$\min_{\mathbf{w}_N, b} \sum_{j=N-1}^{N} \xi\left(y_{j,i} - f(\mathbf{x}_j)\right) + \frac{\lambda}{2}||\mathbf{w}_N||^2 \quad \text{. . .} \quad (14)$$
$$i \in \dim \mathbf{x}_j$$

As a result, the parameters $\mathbf{w}_N, b$ are obtained as the optimal solution. Hence, these parameters are the pattern of the N-th training set. Thus, additional learning is realized using the learning parameters of the $(N-1)$-th and N-th data in combination. Thus, following equations are obtained

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_{N-1} & \mathbf{w}_N \end{bmatrix} \quad \text{. . . . .} \quad (15)$$
$$b' = b \quad \text{. . . . .} \quad (16)$$
$$S = \{s_1, \ldots, s_{N-1}, s_N\}. \quad \text{. . . . . . .} \quad (17)$$

Thus, it is possible to produce predictions by adding new training sets using Eqs. (12) and (13).

3.2. Application of the Predictor Using Online SVR

In [35], a method for predicting the next state and action using the current state and action with Online SVR was proposed, is shown in **Fig. 4**.

To address future state prediction, a scheme will be proposed for predicting the future internal state of a robot using the internal state and action that it adopts as it acquires an action in the unknown environment, to meet the action requirement in the unknown environment. Here, the appropriate control rule was attempted to be determined using the prediction results. If a general learning method will be employed, robots need to undergo many trial and error iterations to determine the control rule themselves. However, learning using real robots will be assumed, although they have few DOFs. Considering the durability of the target robots, less trial and error is the preferred ways as previously mentioned.

In this case, a method that adds sequential prediction to Online SVR is considered, and predicts the internal state and action using the current action and internal state of all the past states until the present time. In this study, this knowledge and implemented virtual learning using the mathematical model were used. Then, a method for acquiring the task using the result of the future prediction based on these experiments will be proposed.

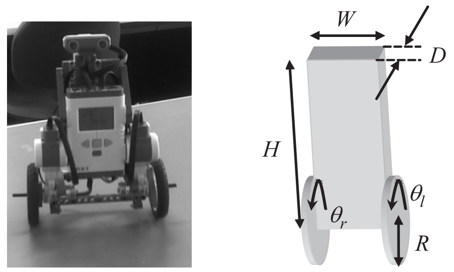
In this study, the posture of a self-propelled inverted pendulum will be stabilized. Control of the motion, such as stabilizing the desired posture, is described by a second-order nonlinear differential equation. It is not easy to derive the control rule in theory or accurately determine the control rule. Therefore, the control rule is needed to be acquired using machine learning. To address future state prediction, Eqs. (12) and (13) are obtained, provided above. Here, the state predictor shown in **Fig. 4** is described by Eq. (12), and the action predictor in the outline is described by Eq. (13). Here, the state and the action can be predicted at each time. By using one of the proposed systems, not only the next state and action but also the advanced future state and action, can be predicted repeatedly.

3.3. Application of the Predictor Using Online SVR and LQR (Linear Quadratic Regulator)

In [36], a scheme for the prediction of the future state and action of a robot using the current state and action with Online SVR and LQR was proposed, as shown in **Fig. 5**. In this case, the control theory is focused on, not only on Online SVR. Using Online SVR for state prediction, and LQR for action prediction is specifically addressed.

Figure 5 shows the prediction of the next state $\hat{\mathbf{x}}_{t+1}$ and action $\hat{a}_{t+1}$ using the current state $\mathbf{x}_t$ and action a_t. For future state prediction, the equation was derived as shown in Eq. (12).

Next, an action predictor is dealt here with using LQR. The future action $\hat{a}_{t+1}$ can be predicted using state feedback gain $\mathbf{k}_f$ if it is possible describe the model of a prediction target as a nonlinear discrete state space model



(a) NXTway-GS (b) Model of NXTway-GS

Fig. 6. Two-wheeled inverted pendulum "NXTway-GS."

correctly:

$$\hat{a}_{t+1} = \mathbf{k}_f \hat{\mathbf{x}}_{t+1}. \quad (18)$$

Here, LQR calculates the feedback gain $\mathbf{k}_f$ in order to minimize the cost function $J[\mathbf{x}(t), \mathbf{a}(t)] \equiv J$ given as

$$J = \int_0^\infty \left(\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{a}^\top(t)\mathbf{R}\mathbf{a}(t) \right) dt. \quad (19)$$

The tuning parameter is the weight matrix for state $\mathbf{Q}$ and for input $\mathbf{R}$. Thus, $\mathbf{k}_f$ represents a state feedback gain that is given by

$$\mathbf{k}_f = -\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}. \quad (20)$$

In this equation, $\mathbf{R}$, $\mathbf{B}$ and $\mathbf{P}$ are the parameters of the Riccati differential equation.

$$\mathbf{PA} + \mathbf{A}^\top\mathbf{P} - \mathbf{PBR}^{-1}\mathbf{B}^\top\mathbf{P} + \mathbf{Q} = 0 \quad (21)$$

Thus, the state and the action can be predicted at each time. By using the proposed system, not only the next state and action but also the future state and action repeatedly, can be predicted.

4. Experiment: Simulation Using Proposed Method for Prediction of the State and Action of the Two-Wheeled Inverted Pendulum

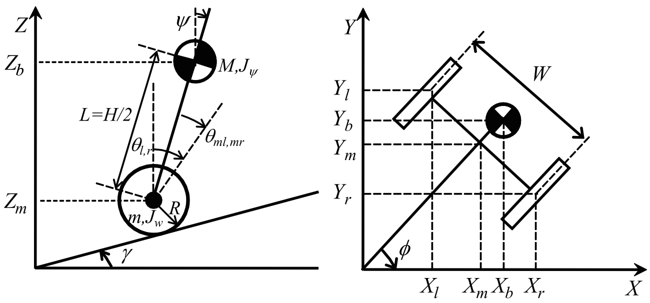
4.1. Outline of Experiment

In this paper, future prediction is focus on and the proposed methods is discussed. As an application example, this simulation used an inverted pendulum "NXTway-GS" (**Fig. 6(a)**) as mathematical model, and parallelly, compared the predicted results with actual response results of simulated results.

4.2. Simulation Setup: Two-Wheeled Inverted Pendulum Model

NXTway-GS can be considered a two-wheeled inverted pendulum model, as shown in **Fig. 6(b)**. **Fig. 7** shows the side view and plane view of the model. The coordinate system referred to in Section 4.3 is described in **Fig. 7**.

In **Fig. 7**, ψ denotes the body pitch angle and $\theta_{ml,mr}$ denotes the DC motor angle (l and r indicate *left* and *right*).



(a) Side view (b) Plane view

Fig. 7. Side view and plane view of NXTway-GS.

Table 1. Physical parameters of NXTway-GS.

Symbol	Value	Unit	Property
g	9.81	[m/s^2]	Gravity acceleration
m	0.03	[kg]	Wheel weight
R	0.04	[m]	Wheel radius
J_w	$\frac{mR^2}{2}$	[kgm^2]	Wheel inertia moment
M	0.635	[kg]	Body weight
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	$\frac{M\left(W^2+D^2\right)}{12}$	[kgm^2]	Body yaw inertia moment
J_m	1×10^{-5}	[kgm^2]	DC motor inertia moment [37]
R_m	6.69	[Ω]	DC motor resistance [a]
K_b	0.468	[V·s/rad]	DC motor back EMF constant [a]
K_t	0.317	[N·m/A]	DC motor torque constant [a]
n	1	[1]	Gear ratio [37]
f_m	0.0022	[1]	Friction coefficient between body and DC motor [37]
f_W	0	[1]	Friction coefficient between wheel and floor [37]

The physical parameters of NXTway-GS are listed in **Table 1**.

4.3. Simulation Setup: Modeling NXTway-GS

The Lagrange equation based on the coordinate system in **Fig. 7** is able to derive the equations of motion of the two-wheeled inverted pendulum "NXTway-GS." If the direction of NXTway-GS is in the x-axis positive direction at $t = 0$, the equations of motion for each coordinate are

$$\left[(2m+M)R^2 + 2J_w + 2n^2J_m\right]\ddot{\theta} + \left(MLR - 2n^2J_m\right)\ddot{\psi} \\ -Rg(M+2m)\sin\gamma = F_\theta \quad (22)$$

$$\left(MLR - 2n^2J_m\right)\ddot{\theta} + \left(ML^2 + J_\psi + 2n^2J_m\right)\ddot{\psi} \\ -MgL\psi = F_\psi \quad (23)$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}\left(J_w + n^2J_m\right)\right]\ddot{\phi} = F_\phi. \quad (24)$$

Table 2. Learning parameters of predictors using Online SVR.

Symbol	Value	Property
C_1	300	Regularization parameter for predictor of x_1
ε_1	0.03	Error tolerance for predictor of x_1
β_1	30	Kernel parameter for predictor of x_1
C_2	300	Regularization parameter for predictor of x_2
ε_2	0.03	Error tolerance for predictor of x_2
β_2	30	Kernel parameter for predictor of x_2
C_3	300	Regularization parameter for predictor of x_3
ε_3	0.03	Error tolerance for predictor of x_3
β_3	30	Kernel parameter for predictor of x_3
C_4	300	Regularization parameter for predictor of x_4
ε_4	0.03	Error tolerance for predictor of x_4
β_4	30	Kernel parameter for predictor of x_4
C_a	300	Regularization parameter for predictor of a
ε_a	0.03	Error tolerance for predictor of a
β_a	30	Kernel parameter for predictor of a

Here, the variables $\mathbf{x}_1$ $\mathbf{x}_2$ as the state variables and $\mathbf{u}$ as the input variable are obtained as following equations:

$$\mathbf{x}_1 = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^\top \qquad (25)$$

$$\mathbf{x}_2 = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top \qquad (26)$$

$$\mathbf{u} = \begin{bmatrix}v_l & v_r\end{bmatrix}^\top \qquad (27)$$

where v_l and v_r indicate the DC motor voltage (l and r indicate *left* and *right*).

Consequently, the state equations of NXTway-GS can be derived from Eqs. (22), (23), and (24).

$$\dot{\mathbf{x}}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \qquad (28)$$

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \qquad (29)$$

Here, $\mathbf{S}$ denotes the slope vector. In this study, only the state variables $\mathbf{x}_1$ is focused. Because $\mathbf{x}_1$ includes the body pitch angle as important variables ψ and $\dot{\psi}$ for the control of self-balancing, the plane motion will not be considered in this experiment.

4.4. Simulation Setup: Using Online SVR

First in this method, this proposed system applied the Online SVR as a learner (see also **Fig. 4**). Moreover, the learner applied the RBF kernel [38] as the kernel function to the Online SVR. The RBF kernel on two samples $\mathbf{x}$ and $\mathbf{x}'$, represented as feature vectors in some input space, is defined as

$$k(\mathbf{x},\mathbf{x}') = \exp\left(-\beta \left|\left|\mathbf{x}-\mathbf{x}'\right|\right|^2\right). \qquad (30)$$

In addition, the learning parameters of Online SVR are listed in **Table 2** (see also Eqs. (4) and (10)).

Here, λ_i denotes following equation using Regularization parameter C_i:

$$\lambda_i = \frac{2}{C_i}. \qquad (31)$$

Table 3. Parameters for Online SVR and LQR.

Symbol	Value	Property
C_1	300	Regularization parameter for predictor of x_1
ε_1	0.03	Error tolerance for predictor of x_1
β_1	30	Kernel parameter for predictor of x_1
C_2	300	Regularization parameter for predictor of x_2
ε_2	0.03	Error tolerance for predictor of x_2
β_2	30	Kernel parameter for predictor of x_2
C_3	300	Regularization parameter for predictor of x_3
ε_3	0.03	Error tolerance for predictor of x_3
β_3	30	Kernel parameter for predictor of x_3
C_4	300	Regularization parameter for predictor of x_4
ε_4	0.03	Error tolerance for predictor of x_4
β_4	30	Kernel parameter for predictor of x_4
$\mathbf{k}_f$	$\begin{bmatrix}-0.870 \\ -32.2 \\ -1.16 \\ -2.81\end{bmatrix}^\top$	Feedback gain for predictor of a

4.5. Simulation Setup: Using Online SVR and LQR

Second, Online SVR was applied as a state predictor, and LQR as an action predictor, as shown in **Fig. 5**. Therefore, the controller for the modern control theory was designed as an action predictor. The tuning parameter is the weight matrix for state for matrix $\mathbf{Q}$ and for input for matrix $\mathbf{R}$. In this study, the weight matrices $\mathbf{Q}$ and $\mathbf{R}$ [37] were defined in this experiment as following.

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6\times10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4\times10^2 \end{bmatrix} \qquad (32)$$

$$\mathbf{R} = \begin{bmatrix} 1\times10^3 & 0 \\ 0 & 1\times10^3 \end{bmatrix} \qquad (33)$$

Then, the feedback gain $\mathbf{k}_f$ is obtained from J, that was minimized. Therefore, $\mathbf{k}_f$ was applied as an action predictor in this experiment. All the parameters for Online SVR and LQR are listed in **Table 3**.

4.6. Simulation Conditions

In this experiment, the unknown periodic disturbance signal as a predictable signal (**Figs. 8** and **9**) was mixed to the action signal. And then, NXTway-GS received this signal.

The properties of disturbance signal provided as the input signal, and the other conditions of the simulations are listed in **Table 4**.

4.7. Simulation Results

4.7.1. Using Online SVR

Figures 10–13 show the prediction of the state of $\mathbf{x}_1 =$

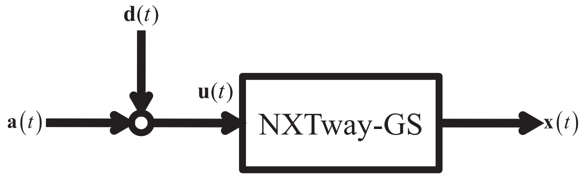


Fig. 8. Control input obtained by mixing the action and disturbance inputs.

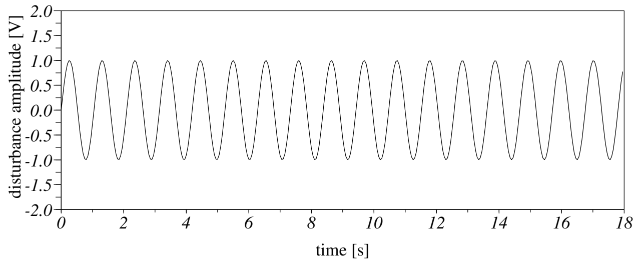


Fig. 9. Disturbance signal in control inputs $\mathbf{d}(t)$.

Table 4. Parameters for simulation conditions.

Symbol	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
$t_{d,\mathrm{finish}}$	18.0	[s]	Finish time of application of predictable disturbance
A_{d1}	1.0	[V]	Amplitude of predictable disturbance
f_{d1}	6.0	[Hz]	Frequency of predictable disturbance
N	60	—	Initial dataset length

$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\top$, and **Fig. 14** shows the prediction of the control input of a. In this section, the part that is given in real training sets will be ignored, because the initial learning part is not arranged to predict next state. Thus, only the part of the graph pertaining to the state predicted part shown in T (at $t = 2.95$ s) of **Figs. 10–14**, will be focused and discussed.

4.7.2. Using LQR

Figures 15–18 show the prediction of the state of $\mathbf{x}_1 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\top$, and **Fig. 19** shows the prediction of the control input of a. In this section, the part that is given in real training sets will not be considered, because the initial learning part, in this area will not predict next state. Thus, only the part of the graph pertaining to the state predicted part shown in T (at $t = 2.95$ s) of **Figs. 15–19**, will be focused and discussed.

4.8. Discussion of Simulated Results of Proposed Method

4.8.1. Using Online SVR as a State Predictor and Action Predictor

Figures 10–14 show that the prediction results almost

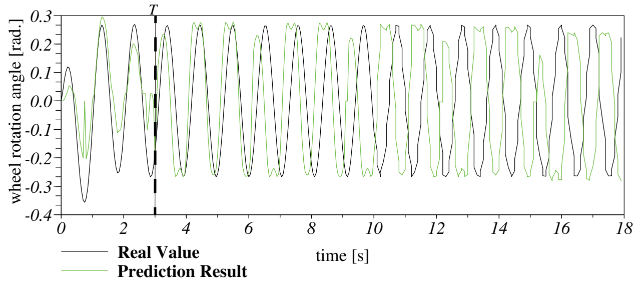


Fig. 10. Prediction result of state x_1 (wheel rotation angle θ) using only Online SVR.

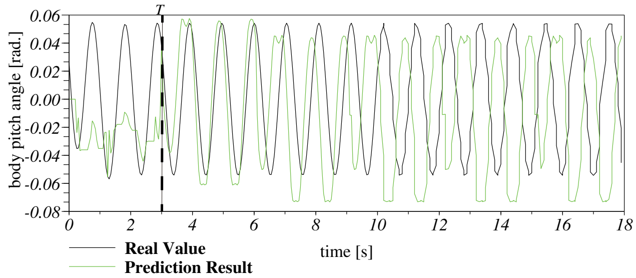


Fig. 11. Prediction result of state x_2 (body pitch angle ψ) using only Online SVR.

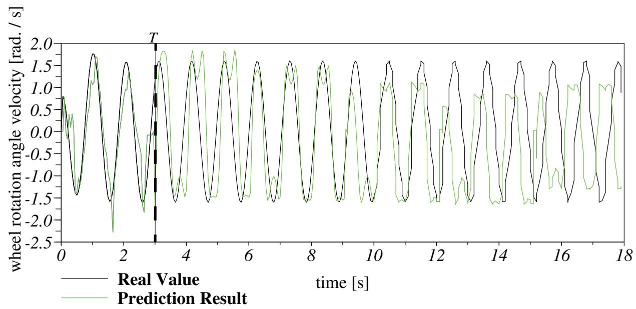


Fig. 12. Prediction result of state x_3 (wheel rotation angle velocity $\dot{\theta}$) using only Online SVR.

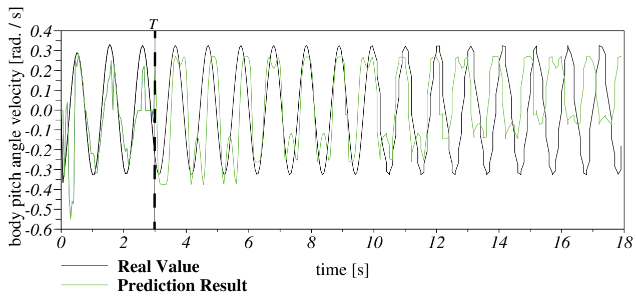


Fig. 13. Prediction result of state x_4 (body pitch angle velocity $\dot{\psi}$) using only Online SVR.

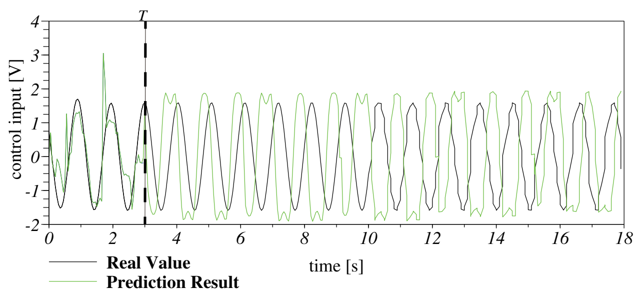


Fig. 14. Prediction result of control input a using only Online SVR.

track the real states and control input. Clearly, **Figs. 10** and **12** show the best prediction results. Because the fluctuations in **Figs. 10** and **12** are smooth, the predicted values are almost the same as the real values. In addition, each predicted value curve tracked the variation of real values correctly. Because sinusoidal voltage is applied to the motor as a disturbance signal through the body of the inverted pendulum, the predictors learn the sinusoid force and predict this tendency as a "predictable disturbance." Therefore, this indicates that this method can be applied in systems for tracking the states and action.

Now let's discuss these results. The increase in the prediction errors can be attributed to the following two reasons. First, trying to consider the parameter settings. The prediction results are determined by an insensitive value of maximal tolerance and a regularization parameter. Thus, the situation was considered wherein a wide insensitive value of tolerance and narrow values of the regularization parameter were used. In this case, there is a possibility that the learner will regard real data as an error value. Second, trying to consider the influence of the disturbance signal input. In this case, there is a possibility that the learner also will regard the real data as an error value. However, if the regularization value and allowable error are set accurately, the prediction will become accurate, at the expense of the generalization. If the disturbance signal with a small amplitude was given to the control target, there is a possibility that the learner will ignore that disturbance signal for learning. This means that the learner regards real data including disturbance as an allowable error.

As a result, except for the nonlinear part, the proposed method was able to conclude that the result can predict, the tendency of inputs except certain errors.

4.8.2. Using Online SVR as a State Prediction and LQR an Action Prediction

Figures 15–19 show that the prediction results almost track the real states and control input. Clearly, **Figs. 15** and **18** show the best prediction results. Because the fluctuations in **Figs. 15** and **18** are smooth, the predicted values are almost the same as the real values. In addition, each predicted value curve tracked the variation of real values correctly. Because a sinusoidal voltage is applied to the motor as a disturbance signal through the body of the inverted pendulum, the predictors learn the sinusoid force and predict this tendency as a "predictable disturbance." Therefore, this indicates that this method can be applied in systems for tracking the states and action.

Now let's discuss these results. These results show the similar tendency of the prediction results using only Online SVR. On the other hand, the prediction results show that fitting the actual control results than using only Online SVR. In this method, the LQR was applied to derive an action that multiplied states to the optimal feedback gain. Therefore, the prediction results show ahead of the tendency at all times, predicting results will be attempting to become a stable state, moreover.

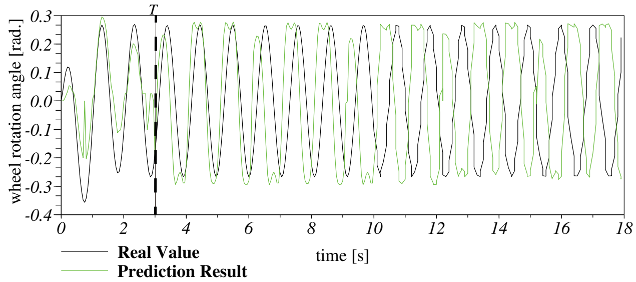


Fig. 15. Prediction result of state x_1 (wheel rotation angle θ) using Online SVR and LQR.

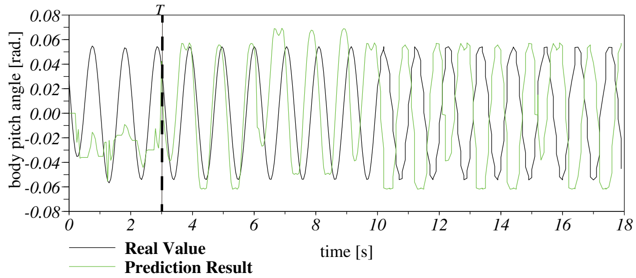


Fig. 16. Prediction result of state x_2 (body pitch angle ψ) using Online SVR and LQR.

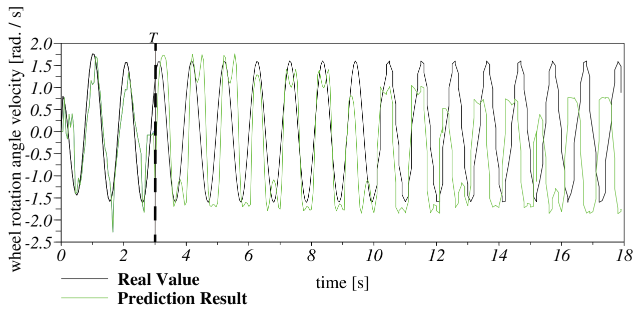


Fig. 17. Prediction result of state x_3 (wheel rotation angle velocity $\dot{\theta}$) using Online SVR and LQR.

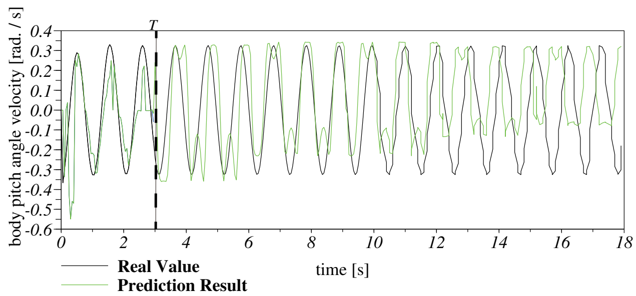


Fig. 18. Prediction result of state x_4 (body pitch angle velocity $\dot{\psi}$) using Online SVR and LQR.

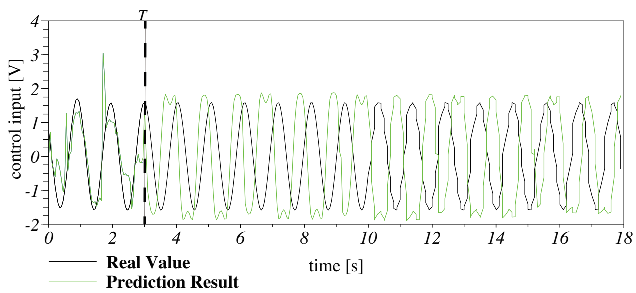


Fig. 19. Prediction result of control input a using only Online SVR and LQR.

Accordingly, the experimental results were able to conclude that the proposed method can achieve predictions, as when only Online SVR is used for predictions, as mentioned above.

4.8.3. Summary

In this experiment, the experimental results showed prediction results using two approaches, i.e., using only Online SVR, or using Online SVR and LQR. As a result, if the fluctuations of the results are smooth, the predicted values tend to be almost the same as the real values. On the other hand, the prediction results changed direction in contradiction to the tendency, since the tendency changes over time. This is attributed to using the prediction values. In the proposed methods, the prediction values are used repeatedly for predicting the future state and action. Hence, the prediction errors will accumulate with elapsed time.

From these reasons, it is concluded that the experimental results are reasonable.

5. Conclusion

In this study, predicting the future state and action of a robot through the current state and action, were focused on. Moreover, the methods that allow a robot to decide its appropriate behavior quickly, using the predicted state were realized. To realize this, the proposed method was considered to predict the advanced future using Online SVR as a learner. Moreover, the only Online SVR, or using Online SVR and LQR were used in proposed two approaches for prediction. The simulation results verify that the proposed method can predict the tendency of inputs. Hence, this paper confirmed the following points.

- The proposed methods can predict adaptively using Online SVR as an online prediction method.
- The proposed methods consider not only the state but also the action, and therefore, these methods can predict future state-action pairs according to the past or current course of action.

As a result, the effectiveness of the proposed method was confirmed, which forms a framework for the state and action. However, in the proposed methods, the prediction values are used repeatedly for predicting the future state and action. Hence, the prediction errors accumulate with elapsed time. And more, the experimental parameters, and the data length that can be used must be considered.

In the future work, applying the knowledge that was learned from this study, to design a system that allows a robot to decide its course of action, will be considered. Specifically, a method that allows a robot to decide an appropriate action at the present time using the prediction results will be considered.

References:

[1] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)," The MIT Press, 2005.

[2] S. Asaka and S. Ishikawa, "Behavior Control of an Autonomous Mobile Robot in Dynamically Changing Environment," J. of the Robotics Society of Japan, Vol.12, No.4, pp. 583-589, 1994.

[3] T. Kanda, H. Ishiguro, T. Ono, M. Imai, T. Maeda, and R. Nakatsu, "Development of "Robovie" as Platform of Everyday-Robot Research," IEICE Trans. on Information and Systems, Pt.1 (Japanese Edition), Vol.J85-D-1, No.4, pp. 380-389, 2002.

[4] D. F. Wolf and G. S. Sukhatme, "Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments," Autonomous Robots, Vol.19, pp. 53-65, Springer, Netherlands, 2005.

[5] D. Fox, W. Burgard, and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," J. of Artificial Intelligence Research, Vol.11, pp. 391-427, 1999.

[6] M. A. K. Jaradata, M. Al-Rousanb, and L. Quadanb, "Reinforcement based Mobile Robot Navigation in Dynamic Environment," Robotics and Computer-Integrated Manufacturing, Vol.27, pp. 135-149, 2011.

[7] E. Masehian and Y. Katebi, "Sensor-Based Motion Planning of Wheeled Mobile Robots in Unknown Dynamic Environments," J. of Intelligent & Robotic Systems, DOI: 10.1007/s10846-013-9837-3, 2013.

[8] M. Faisal, R. Hedjar, M. Al Sulaiman, and K. Al-Mutib, "Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment," Int. J. of Advanced Robotic Systems, DOI: 10.5772/54427, 2012.

[9] F. Abrate, B. Bona, M. Indri, S. Rosa, and F. Tibaldi, "Multi-robot Map Updating in Dynamic Environments," Distributed Autonomous Robotic Systems, Springer Tracts in Advanced Robotics, Vol.83, pp. 147-160, 2013.

[10] International Federation of Robotics, "All-time-high for industrial robots," Substantial increase of industrial robot installations is continuing, 2011.

[11] T. Sogo, K. Kimoto, H. Ishiguro, and T. Ishida, "Mobile Robot Navigation by a Distributed Vision System," J. of the Robotics Society of Japan, Vol.17, No.7, pp. 1-7, 1999.

[12] J. J. Park, C. Johnson, and B. Kuipers, "Robot Navigation with MPEPC in Dynamic and Uncertain Environments: From Theory to Practice," IROS 2012 Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs, 2012.

[13] M. Nishioka, A. Okada, M. Miyano, K. Mori, K. Yamashita, and K. Nakayama, "A Basic Study on the Relationship between Operating Strategy and Body Movements under a Repetitive Operation – Evaluation of Methods of the Learning Process from the Viewpoint of Performance and Physiological Data during Pulling Cart Operation –," J. of Human Life Science, Vol.7, pp. 45-55, 2008.

[14] N. Sugimoto, K. Samejima, K. Doya, and M. Kawato, "Reinforcement Learning and Goal Estimation by Multiple Forward and Reward Models," IEICE Trans. on Information and Systems, Pt.2 (Japanese Edition), Vol.J87-D-2, No.2, pp. 683-694, 2004.

[15] Y. Takahashi and M. Asada, "Incremental State Space Segmentation for Behavior Learning by Real Robot," J. of the Robotics Society of Japan, Vol.17, No.1, pp. 118-124, 1999.

[16] Y. Choi, S.-Y. Cheong, and N. Schweighofer, "Local Online Support Vector Regression for Learning Control," Proc. of the 2007 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation Jacksonville, FL, USA, 2007.

[17] J. Shin, H. J. Kim, S. Park, and Y. Kim, "Model predictive flight control using adaptive support vector regression," Neurocomputing, Vol.73, No.4-6, pp. 1031-1037, 2010.

[18] E. Schuitema, L. Busoniu, R. Babuska, and P. Jonker, "Control Delay in Reinforcement Learning for Real-Time Dynamic systems: A Memoryless Approach," Proc. of Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ Int. Conf., pp. 3226-3231, 2010.

[19] T. J. Walsh, A. Nouri, L. Li, and M. L. Littman, "Planning and Learning in Environments with Delayed Feedback," Machine Learning: ECML 2007, pp. 442-453, 2007.

[20] Y. Su, K. K. Tan, and T. H. Lee, "Computation Delay Compensation for Real Time Implementation of Robust Model Predictive Control," Proc. of Industrial Informatics (INDIN), 2012 10th IEEE Int. Conf., pp. 242-247, 2012.

[21] C. Liu, W.-H. Chen, and J. Andrews, "Model Predictive Control for Autonomous Helicopters with Computational Delay," Proc. of Control 2010, UKACC Int. Conf., pp. 1-6, 2010.

[22] G. Marafioti, S. Olaru, and M. Hovd, "State Estimation in Nonlinear Model Predictive Control, Unscented Kalman Filter Advantages," Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences Vol.384, pp. 305-313, 2009.

[23] N. Sünderhauf, S. Lange, and P. Protzel, "Using the Unscented Kalman Filter in Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control," Proc. of IEEE Int. Workshop on SSRR 2007, pp. 1-6, 2007.

[24] M. A. Badamchizadeh, I. Hassanzadeh, and M. A. Fallah, "Extended and Unscented Kalman Filtering Applied to a Flexible-Joint Robot with Jerk Estimation," Discrete Dynamics in Nature and Society, Vol.2010, Article ID 482972, 2010.

[25] J. G. Iossaqui, J. F. Camino, and D. E. Zampieri, "Slip Estimation Using The Unscented Kalman Filter for The Tracking Control of Mobile Robots," Proceeding of the Int. Congress of Mechanical Engineering (COBEM), pp. 1-10, 2011.

[26] A. F. Foka and P. E. Trahanias, "Predictive Control of Robot Velocity to Avoid Obstacles in Dynamic Environments," Proc. of the 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol.1, pp. 370-375, 2003.

[27] L. Jiang, M. Deng, and A. Inoue, "SVR based Obstacle Avoidance and Control of a Two Wheeled Mobile Robot," Proceeding of Innovative Computing, Information and Control 2007 (ICICIC '07), Second Int. Conf., Okayama, DOI: 10.1109/ICICIC.2007.553, 2007.

[28] Z. Li, K. Yang, and Y. Yang, "Support Vector Machine based Optimal Control for Mobile Wheeled Inverted Pendulums with Dynamics Uncertainties," Proc. of the 48th IEEE Conf. on Decision and Control 2009 held jointly with the 2009 28th Chinese Control Conf., 2009.

[29] F. Parrella, "Online Support Vector Regression," Ph.D. thesis, Department of Information Science, University of Genoa, Italy, 2007.

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

[31] B. S. Everitt, "Cambridge Dictionary of Statistics (2nd edition)," Cambridge, UK: Cambridge University Press, ISBN 0-521-81099-X, 2002.

[32] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the Kernel Matrix with Semidefinite Programming," J. of Machine Learning Research, Vol.5, pp. 27-72, 2004.

[33] V. N. Vapnik, "The Nature of Statistical Learning Theory," Springer, New York, 1995.

[34] F. Girosi, "An Equivalence between Sparse Approximation and Support Vector Machines," Neural Computation, Vol.10, No.6, pp. 1455-1480, 1998.

[35] M. Sugimoto and K. Kurashige, "The Proposal for Prediction of Internal Robot State Based on Internal State and Action," Proc. of IWACIII2013 CD-ROM, SS1-2, Oct. 18-21, Shanghai, China, 2013.

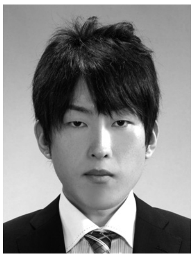
[36] M. Sugimoto and K. Kurashige, "The Proposal for Deciding Effective Action using Prediction of Internal Robot State Based on Internal State and Action," Proc. of 2013 Int. Symposium on Micro-NanoMechatronics and Human Science, pp. 221-226, Nov. 10-13, Nagoya, Japan, 2013.

[37] Y. Yamamoto, "NXTway-GS Model-Based Design – Control of Self-balancing Two-wheeled Robot Built with LEGO Mindstorms NXT –," Cybernet Systems Co., Ltd., 2009.

[38] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and Testing Low-degree Polynomial Data Mappings via Linear SVM," J. Machine Learning Research, Vol.11, pp. 1471-1490, 2010.

Supporting Online Materials:

[a] R. Watanabe, "Ryo's Holiday LEGO Mindstorms NXT," 2008. http://web.archive.org/web/20120617051545/http://web.mac.com/ryo_watanabe/iWeb/Ryo's%20Holiday/NXTway-G.html [Accessed September 28, 2015]



Name:
Masashi Sugimoto

Affiliation:
Muroran Institute of Technology

Address:
27-1 Mizumoto-cho, Muroran-shi, Hokkaido 050-8585, Japan

Brief Biographical History:

2013 Received Master degree from Muroran Institute of Technology

2013- Doctoral Course Student, Muroran Institute of Technology

Main Works:

- M. Sugimoto and K. Kurashige, "The Proposal for Prediction of Internal Robot State Based on Internal State and Action," Proc. of IWACIII2013 CD-ROM, SS1-2, 2013.
- M. Sugimoto and K. Kurashige, "The Proposal for Deciding Effective Action using Prediction of Internal Robot State Based on Internal State and Action," Proc. of 2013 IEEE Int. Symposium on Micro-NanoMechatronics and Human Science, pp. 221-226, 2013.
- M. Sugimoto and K. Kurashige, "Real-time Sequentially Decision for Optimal Action using Prediction of the State-Action Pair," Proc. of 2014 IEEE Int. Symposium on Micro-Nano Mechatronics and Human Science, pp. 199-204, 2014.

Membership in Academic Societies:

- The Japan Society for Precision Engineering (JSPE)
- The Robotics Society of Japan (RSJ)
- The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Kentarou Kurashige

Affiliation:
Muroran Institute of Technology

Address:
27-1 Mizumoto-cho, Muroran-shi, Hokkaido 050-8585, Japan

Brief Biographical History:

2002 Received the Ph.D. degree from Nagoya University

2002-2005 Research Associate, Fukuoka University

2005-2015 Research Associate, Muroran Institute of Technology

2015- Associate Professor, Muroran Institute of Technology

Main Works:

- Y. Kishima, K. Kurashige, and T. Kimura, "Decision Making in Reinforcement Learning Using a Modified Learning Space Based on the Importance of Sensors," J. of Sensors, Vol.2013, Article ID 141353, 2013. DOI: 10.1155/2013/141353.
- K. Kurashige and Y. Miyazaki, "Use of the Knowledge of Perceptual State Transition in Reinforcement Learning," JSCSE, Vol.3, No.2, pp. 1-12, 2013.
- K. Kurashige, N. Kitayama, and M. Kiyohashi, "Proposal of Method "Motion Space" to Express Movement of Robot," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.16, No.6, pp. 704-712, 2012.
- K. Kurashige, Y. Onoue, and T. Fukuda, "From Automation To Autonomy," A. Mellouk and A. Chebira (Eds.), Machine Learning, In-Tech, pp. 39-52, Feb. 2009.

Membership in Academic Societies:

- IEEE Robotics and Automation Society
- IEEE Systems, Man, and Cybernetics Society
- The Robotics Society of Japan (RSJ)
- Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
- The Japanese Society for Artificial Intelligence (JSAI)