The Proposal for Compensation to the Action of Motion Control based on the Prediction of State-action Pair

Masashi Sugimoto Kentarou Kurashige
Muroran Institute of Technology
Email: sm04322@gmail.com Email: kentarou@csse.muroran-it.ac.jp

ABSTRACT

For a robot that works in a dynamic environment, the ability to autonomously cope with the changes in the environment, is important. In this paper, an approach to predict the changes of the state and action of the robot is proposed. Further, to extend this approach, the action to be taken in the future will be attempted to apply, to the current action. This method predicts the robot state and action for the distant future using the state that the robot adopts repeatedly. Using this method, the actions that the robot will take in the future can be predicted. In this paper, a method that predicts the state and the action each time the robot decides to perform an action will be proposed. In particular, in this paper, how to define the weight coefficients will be focused on, using the characteristics of the future prediction results. Using this method, the compensatory current action will be obtained. This paper presents the results of our study and discusses methods that allow the robot to decide its desirable behavior quickly, using state prediction and optimal control methods.

KEYWORDS

Online State Prediction, Prediction using Combination of State Space and Action, Online SVR, Control using State-action Prediction

1 INTRODUCTION

Over the years, many studies have been conducted with the objective of developing working robots that can facilitate or are suitable for dynamic environments [1]-[9]. Further, various robots have been developed to assist humans in workspaces such as a house and factory [10]. However, it is virtually impossible to predict all possible situations and to preprogram a robot with all suitable reaction patterns for each of the possible situations, because robots are required to act differently in different situations [11, 12]. Further, it is difficult to calculate the inverse problem of the robot for generating a suitable reaction in real-time. Considering these problems, it can be said that for robots that work in a dynamic environment, the ability to cope with changes in the environment is important. Especially, the mobile robot will be focused on. In this case, it can be said that the behaviour and the posture have crucially important meaning during executing the task.

Some of the conventional studies related to robot control focus on the states of the target robot or the parameters of the robot. Based on these viewpoints, these studies have presented certain methods to control the robots [13, 14]. Thus, it is important that to know the states of the robot and to know that the robot will take an action in the next scene, when the robot in dynamic environments is controlled. Moreover, the model of the robot to obtain the states of the robot and to predict the action that will be taken by the robot have to be known.

For these problems, many studies have used the machine learning, such as reinforcement learning (RL), that acquire the optimal action to learn the environment by trial and error [15, 16]. Alternatively, model predictive control (MPC) has been used to input to the control sequentially, that is gained suitable input by each time. In this point, this MPC is better as the typically control rule [17, 18]. However, these techniques suffer from problems such as computation delay, hardware overhead, and whether the robot can respond flexibly to changes in the dynamic environment [19]-[23]. Some techniques for generating robot motion have also been presented. In this case, the ordinary control rule with the extended Kalman

ISBN: 978-1-941968-17-8 ©2015 SDIWC

filter (EKF) [1] or the unscented Kalman filter (UKF) [1] were combined, to avoid linearizing the robot model was combined [24]-[26]. However, these techniques still have some problems; the case of applied filter will often become unstable, the case of using a nonlinear model is not easy, and the case in which the parameter should be defined are increasing [27].

By contrast, some researches are applying Support Vector Regression (SVR). These works provides appropriate results, generally [17, 18, 28]. However, these works are avoiding the disturbances. Moreover, these works don't consider the relationship between robot's state transition and the action that the robot taken. Considering the dynamic environment, for example, there are unforeseen loads or backlash, complex friction and stiction. In this case, it is essential that predicting the state and the action. Furthermore, it also allows the adaptation of the model to changes in the robot dynamics.

Thus, the prediction of the state-action pair to apply the prediction for the robot control has already been proposed [29, 30], on the basis of Online SVR [31]. This method predicts the robot state and action, using them as a "pair" for the distant future, applying the state that the robot adopts repeatedly. Using this method, the suitable actions that the robot can take in the future can be obtained [32]. Most of the earlier conventional approaches that based on parametric techniques were mentioned before. However, this research is based on nonparametric techniques. In this respect, the proposed method does not need a strict parameter or model of the target robot. On the basis of this characteristic, it can be said that the proposed method is different from the other related methods.

In this paper, the results of these studies and discuss methods that allow the robot to decide its desirable behavior quickly will be presented, using the state prediction and ordinary optimal control methods. First, we will attempt to apply the action to be taken in the future to the current action, by extending the former approach. In particular, in this study, we have tried to determine the future action and apply the current action; further, the weight coefficients for the future actions that were obtained through the prediction had been designed. In [32], the fixed-value weight coefficients has already been proposed. Hence, in this paper, the weight coefficients will be proposed that focus on the "variation of the predicted results. " Using this method, the compensatory current action more flexibly will be obtained.

In this study, the stabilize the posture using the system that combining the optimal control and the proposed method will be realized. In particular, for realization and considering the proposed method, the inverted pendulum that can be linearized around these operating point was focused on. From this example, the ordinary LQR can be applied. Using this method, in this paper, to converge the posture to stable state immediately using future prediction result will be aimed. In particular, as an application example, the two-wheeled mobile inverted pendulum robot "NXTway-GS" model was used that drives a stabilize control task. Moreover, the move instruction was sent from the command input within stabilize control task. And hence, the results of the proposed method with the ordinary control method of the control response were compared, using a computer simulation. As a result of the computer simulation, the proposed method is well adapted to the disturbance input and obtained the action that decreases the pitch angle and achieves the desirable state of the NXTway-GS, as compared to the ordinary control method was confirmed, with time.

This paper is organized as follows: In Section 2, how to obtain or decide the optimal action for the robot using future prediction techniques is explained. Further, details about the proposed method is provided. In Section 3, the experimental setting is explained. Finally, in Section 4, the conclusions of this study is presented.

ISBN: 978-1-941968-17-8 ©2015 SDIWC

2 AN APPROACH FOR DECIDING THE OPTIMAL ACTION FOR A ROBOT

2.1 Basic Idea

We mentioned in Section 1 that for controlling a robot in a dynamic environment, an action that adopted the current result by predicting the future state using previous actions and states can be chosen. In this paper, we will try to consider that obtain the optimal action to minimize the body pitch angle of the inverted pendulum, in case of continuing input the predictable disturbance. To realize this, we will try to use the prediction the state-action pair that had proposed in the former our studies [29], [30]. Therefore, in this study, a system that decides the action to optimize using the proposed method illustrated in fig. 1 based on previous studies including [32] is considered.

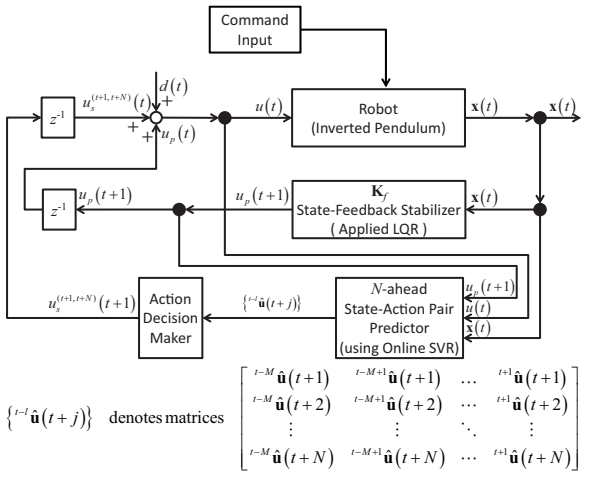


Figure 1. Outline for Deciding the Optimal Action for the Robot using the Prediction of State-Action Pair

2.1.1 N-ahead state-action pair predictor

As shown in fig. 1, ${}^{t-l}\hat{\mathbf{u}}(t+j)$ describes the prediction result of the control input $\mathbf{u}(t+j)$, when this input is predicted in time $(t-l)$. Hence, this proposed method is trying to revise the current action, using combination of the optimal control and the prediction result of state-action pair. The structure of the prediction of a state-action pair is named "N-ahead state-action pair predictor," and the internal structure is described in fig. 2 [30]. Here, the equations

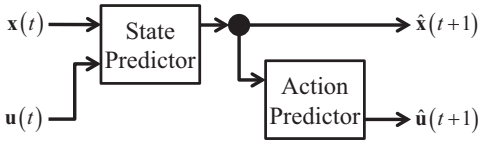


Figure 2. Outline of the Prediction System of State and Action [30]

for the state prediction is as follows.

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

$$\text{when}\, i \in \dim \mathbf{x}(t)$$

In this paper, the notation is defined as

$$\mathbf{z}(t) = \left[\mathbf{x}(t) \quad | \quad \mathbf{u}(t-1)\right] \tag{2}$$

Here, b_i' is a bias term for $x_i(t)$ (the i-th element of $\mathbf{x}(t)$ in time t), $\Delta\theta$ represents the Lagrange multiplier, l represents the number of the former support vector $\mathbf{z}_{s_k}(t)(k \in l)$, 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.

Moreover, ${}^{x}\mathbf{z}(t)$ is defined by state $\mathbf{x}(t)$ and the pair $\mathbf{z}(t)$: ${}^{x}\mathbf{z}(t) = \left[\mathbf{z}(t) \quad | \quad \mathbf{x}(t)\right]$. Next, as an action predictor to be dealt here with using the linear-quadratic regulator (LQR). The future action $\hat{\mathbf{u}}(t+1)$ 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 correctly:

$$\hat{\mathbf{u}}(t+1) = \mathbf{k}_f\hat{\mathbf{x}}(t+1) \tag{3}$$

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

$$J = \int_0^{\infty} \left(\mathbf{x}^{\top}(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^{\top}(t)\mathbf{R}\mathbf{u}(t)\right) dt \tag{4}$$

In this equation, $\mathbf{x}^{\top}$ indicates the transpose of $\mathbf{x}$. The tuning parameter is the weight matrix

ISBN: 978-1-941968-17-8 ©2015 SDIWC

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

$$\mathbf{k}_f = -\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P} \quad (5)$$

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

$$\mathbf{PA} + \mathbf{A}^\top\mathbf{P} - \mathbf{PBR}^{-1}\mathbf{B}^\top\mathbf{P} + \mathbf{Q} = 0 \quad (6)$$

Thus, the state and the action can predict 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.

2.1.2 State-Feedback Stabilizer

As shown in fig. 1, this system will be applied to an optimal control by using a feedback gain $\mathbf{K}_f$ based on the linear-quadratic regulator (LQR), and in parallel, decide the action that the robot will have to take in the future using the prediction of a state-action pair.

In this system, the LQR as deriving a optimal feedback gain is applied. Therefore, the controller is based on modern control theory is designed. This LQR calculates the feedback gain $\mathbf{K}_f$ so as to minimize the cost function J given as eq. (4).

2.2 How to Obtain and Use the Optimal Action

However, the prediction error must be considered because the proposed method uses the action obtained from the N-ahead state-action pair predictor. In case if N is larger or more distant than current time t, then the prediction error will be proportional to N, and the piling prediction error cannot be ignored in the prediction (fig. 3).

Considering this prediction problem, in [32], fixed-value weighting coefficients for the prediction series of the action was defined, defined the measure of importance of each prediction series, and tried decreasing the influence of the prediction error of these prediction series of the action. In this paper, the variations of the predicted results at any time $(t + j)$ is focused on, and try to decide the "*dependability*" of the predicted value at that time. In other words, variable weight coefficients based on this dependability is defined.

The prediction of a state-action pair prediction that can predict future values in each sampling time is illustrated in fig. 3. In fig. 3, the

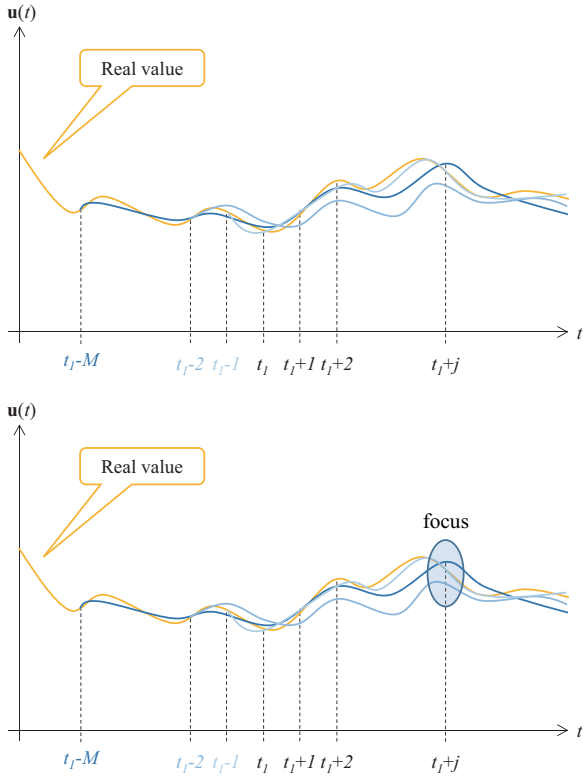


Figure 3. Prediction Using a State-Action Pair

prediction values is focused on. In here, the prediction values at time $(t_1 + j)$ can be obtained, from each past time $(t_1 - i)$ (the range of i is $(0 \ll M \leq i \leq N)$) as illustrated in fig. 4. In this area, the predicted values are distributed. To quantitatively measure these variations, the standard deviation is focused on. In brief, the standard deviation σ_j of the control input in time $(t_1 + j)$ using the prediction values that are predicted in each past time $(t_1 - i)$ is derived as follows:

$$\begin{aligned}\sigma_j \quad &= \sigma\left[{}^{t_1-M}\hat{\mathbf{u}}(t_1+j), {}^{t_1-M+1}\hat{\mathbf{u}}(t_1+j),\right. \\ &\left. \ldots, {}^{t_1-i}\hat{\mathbf{u}}(t_1+j), \ldots, {}^{t_1+1}\hat{\mathbf{u}}(t_1+j)\right]\end{aligned} \quad (7)$$

Based on the above σ_j, we try to consider the weight coefficients. In eq. (7), $\sigma\,[\cdot]$ denotes the standard deviation of $(\cdot)$. Using the characteristics of the standard deviation, a weight coefficient according to the variations of the pre-

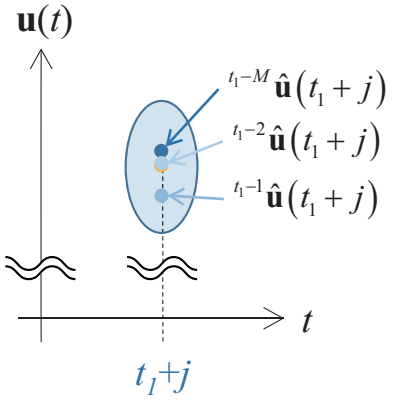


Figure 4. Focus on the Variations of Predicted Values in Past Time (From Fig. 3)

dicted results is defined, flexibly. Therefore, weight coefficients α_j can be expressed as follows:

$$\alpha_j = C_\sigma \cdot \sigma_j \tag{8}$$

Here, C_σ is a coefficient that is a small positive real value.

Here, that the influence increases for an action in the near future and inversely decreases for an action in the distant future can be considered. That is, to the "decide the optimal action" section from [32] can be refered as follows:

$$\mathbf{u}_s^{(t+1,\,t+N)}(t) = \sum_{j=1}^{N} \alpha_j \hat{\mathbf{u}}(t+j) \tag{9}$$

Here, α_j describes the weight coefficients for each j in each t. This $\mathbf{u}_s^{(t+1,\,t+N)}(t)$ is generated from "Action Decision Maker" and is the revised action for considering a future action. Now, the prediction results $\mathbf{u}(t), \hat{\mathbf{u}}(t+j)$, from the N-ahead state-action pair predictor's outputs include the disturbance input $\mathbf{d}(t)$, explicitly [32]. Hereby, if an action that can correspond to against the disturbance input before ahead is created, the optimal action that is considered in the future can be obtained. In other words, the future action $\mathbf{u}_s^{(t+1,\,t+N)}(t)$ and the optimal control action $\mathbf{u}_p(t)$ is used as follows:

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1,\,t+N)}(t) + \mathbf{d}(t) \tag{10}$$

Here, the compensate control input $\mathbf{u}(t)$ is given. The proposed method can obtain a series of actions in time $(t+N)$ in the distant future from current time t using the N-ahead state-action pair predictor. Now, on the basis of this prediction series, the current action, combining "*the action that will be taken in the future*" and using the prediction series of the action can be revised. Namely, as the compensate control input, combining current action and the action that takes future action in advance. Hereby, the compensate control input can compensate the disturbance in the future using the N-ahead state-action pair predictor. Moreover, the N-ahead state-action pair predictor can work in every sampling time. Therefore, the compensate control input can treat the changing disturbance.

3 EXPERIMENT – SIMULATION USING THE PROPOSED METHOD FOR CONTROL OF TWO-WHEELED INVERTED PENDULUM

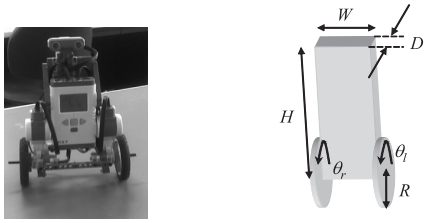
3.1 Outline of Experiment

In this study, as an application, the posture of a two-wheeled self-propelled inverted pendulum, using the computer simulation was stabilized. Moreover, the command input will be sent to the inverted pendulum. In this experiment, it will be confirmed that the proposed method adapting the changing environment. Namely, it will be confirmed that using learning and predicting continuously, the posture will be keep continuing the stable posture on the flat floor or the undulate floor. Furthermore, the environment will be changed due to the command input. From this simulation, states and an action as training samples was obtained while stabilizing postural control. In this paper, stabilizing the inverted pendulum by using a future prediction based on the prediction of a state-action pair and considering the proposed system is focused on. In this verification experiment, as an application example, an inverted pendulum "NXTway-GS" (fig. 5) is used and compare the control response of the proposed method with that of the ordinary method. Moreover, more than 300 steps (actual predictive control range was 3.00 [s] to 15.00 [s]). Furthermore, in the proposed method, the

predictor only used the proximate predicted result repeatedly with 0.05 [s] for each sampling time, for postural control.

3.2 Simulation Setup - NXTway-GS Model

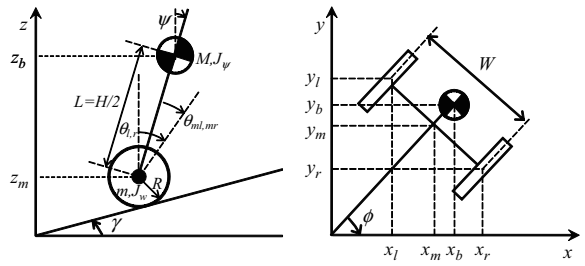
NXTway-GS (fig. 5) can be considered a two-wheeled inverted pendulum model as shown in fig. 6. Figure 6 shows the side view and the plane view of the model. The coordinate system used in 3.3 is described in fig. 6. Further, in fig. 6, ψ denotes the body pitch angle and $\theta_{l,r}$denotes the wheel angle (l and r indicate *left* and *right*, moreover $\theta = \frac{1}{2}(\theta_l + \theta_r)$), $\theta_{ml,mr}$ 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. Side View and Plane View of NXTway-GS [33]-[35]

3.3 Simulation Setup - Modeling of NXTway-GS

The equations of motion of the two-wheeled inverted pendulum model using the Lagrange equation based on the coordinate system shown in fig. 6 can be derived. If the direction of the model is the x-axis positive direction at $t = 0$, the equations of motion for each coordinate are given as follows ([33]-[35]):

$$\left[(2m+M)R^2 + 2J_w + 2n^2J_m\right]\ddot{\theta} + \left(MLR - 2n^2J_m\right)\ddot{\psi} - Rg(M+2m)\sin\gamma = F_\theta \tag{11}$$

$$\left(MLR - 2n^2J_m\right)\ddot{\theta} + \left(ML^2 + J_\psi + 2n^2J_m\right)\ddot{\psi} - MgL\psi = F_\psi \tag{12}$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}\left(J_w + n^2J_m\right)\right]\ddot{\phi} = F_\phi \tag{13}$$

Here, the following variables $\mathbf{x}_1, \mathbf{x}_2$ as the state variables and $\mathbf{u}$ as the input variable is considered.

$$\mathbf{x}_1 = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^\top \tag{14}$$

$$\mathbf{x}_2 = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top \tag{15}$$

$$\mathbf{u} = \begin{bmatrix}v_l & v_r\end{bmatrix}^\top \tag{16}$$

Consequently, the state equations of the inverted pendulum model using eqs. (11), (12), and (13) can be derived.

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \tag{17}$$

$$\frac{d}{dt}\mathbf{x}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \tag{18}$$

In this study, the state variable $\mathbf{x}_1$ is only used. Because $\mathbf{x}_1$ includes the body pitch angles as important variables ψ and $\dot{\psi}$ for the control of self-balancing, the plane motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$) will be not considered.

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

In this method, online SVR [31] as a learner is used in state predictor, as shown in fig. 2.

Moreover, the RBF kernel [37] as the kernel function to the online SVR of the learner is applied. The RBF kernel on two samples $\mathbf{x}$ and $\mathbf{x}'$, represented as feature vectors in some input space, is defined as

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\beta ||\mathbf{x} - \mathbf{x}'||^2\right) \quad (19)$$

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

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

In this experiment, the LQR as an action predictor is applied, as shown in fig. 2. Therefore, the controller as an action predictor based on modern control theory is designed. This LQR calculates the feedback gain $\mathbf{k}_f$ so as to minimize the cost function J given as eq. (4). In this study, the following weight matrix $\mathbf{Q}$ and $\mathbf{R}$ is chosen:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6 \times 10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \times 10^2 \end{bmatrix} \quad (20)$$

$$\mathbf{R} = 1 \times 10^3 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (21)$$

Then, the feedback gain $\mathbf{k}_f$ by minimizing J is obtained. Therefore, $\mathbf{k}_f$ as an action predictor is applied [30]. And also, the feedback gain $\mathbf{K}_f$ of state-feedback stabilizer was applied. Hence, in this experiment, the plane move of the two-wheeled inverted pendulum is not considered. In other words, $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$ were considered.

3.6 Conditions of Simulation - Acquiring Training Sets

In this experiment, the command input will be sent to the inverted pendulum. Furthermore, the environment will be changed due to the command input. In detail, shape of floor will be changed from flat to undulating. Figure 7

Table 1. Physical Parameters of NXTway-GS

Symbol	Value	Unit	Physical property
g	9.81	[m/s^2]	Gravity acceleration
m	0.03	[kg]	Wheel weight [33, 34]
R	0.04	[m]	Wheel radius
J_w	$\frac{mR^2}{2}$	[kgm^2]	Wheel inertia moment
M	0.635	[kg]	Body weight [33, 34]
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 [35]
R_m	6.69	[Ω]	DC motor resistance [36]
K_b	0.468	[V·s/rad.]	DC motor back EMF constant [36]
K_t	0.317	[N·m/A]	DC motor torque constant [36]
n	1	[1]	Gear ratio [35]
f_m	0.0022	[1]	Friction coefficient between body and DC motor [35]
f_W	0	[1]	Friction coefficient between wheel and floor [35]

Table 2. Learning Parameters of 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

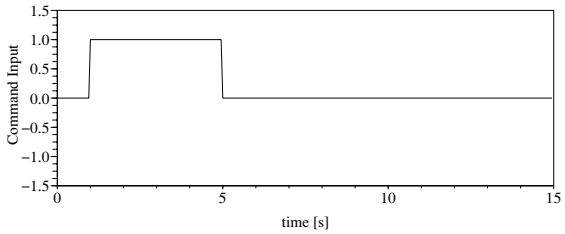


Figure 7. Signal of the Command Input (+1 Indicates Forward Run, 0 Indicates Stationary Balancing)

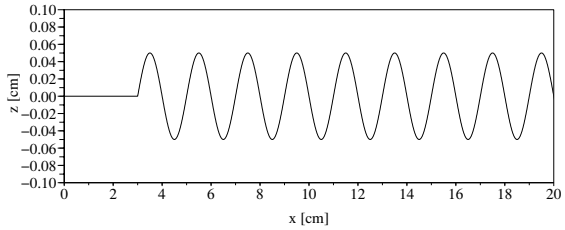


Figure 8. Simulation Environment (the Shape of the Floor)

shows the command input and fig. 8 shows the shape of the floor. Here, the shape of the floor is given by the equation below.

$$z = 0.05U(x-3)\sin\left[\pi\cdot(x-3)\right] \quad \text{[cm]} \tag{22}$$

In this equation, $U(x)$ indicates the unit step function, x [cm] indicates length of the floor, z [cm] indicates undulating height of the floor.

From the settings mentioned above, we try to drive the inverted pendulum model on the floor with the command input (fig. 9). Thus,

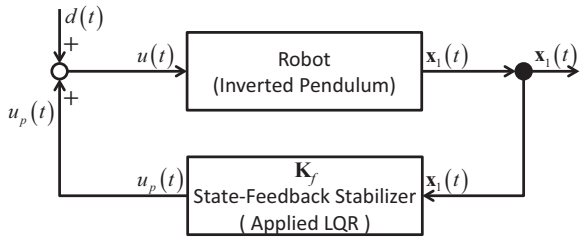


Figure 9. Control Input Obtained by Mixing the Action and Command Inputs

the training sets from the two-wheeled inverted pendulum can be acquired. Figures 10 to 14 show training sets that were obtained from the computer simulation of the stabilizing control of the two-wheeled inverted pendulum, and sent move forward instruction from the command input as fig.7 within the control.

Moreover, figs. 15 and 16 show the position of the mass of the inverted pendulum. Here, movement distance for displaying the position is given as below equation.

$$x = 100\cdot R\cdot\int\dot{\theta}(t)dt \quad \text{[cm]} \tag{23}$$

Here, position of the mass of the inverted pendulum z_m is given as below equation.

$$z_m = 100\cdot\left[R + R\sin\gamma\cdot\int\dot{\theta}(t)dt + L\cos\left\{\int\dot{\psi}(t)dt\right\}\right] \quad \text{[cm]} \tag{24}$$

In this experiment, the two-wheeled inverted pendulum model takes stationary balancing or moves forward from the command input. Moreover, the properties of disturbance that we provide as input and other conditions of a simulation are listed in table 3.

Table 3. Parameters for a Simulation

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
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
C_σ	0.05	—	The coefficient for the standard deviation of the predicted values
N_σ	10	—	The calculate range of standard deviation the predicted values

3.7 Simulation Results

In this simulation, the input of NXTway-GS is using the predicted result that is based on the proposed method. Also NXTway-GS model takes stationary balancing based on that input, and thereby moving a backward or forward.

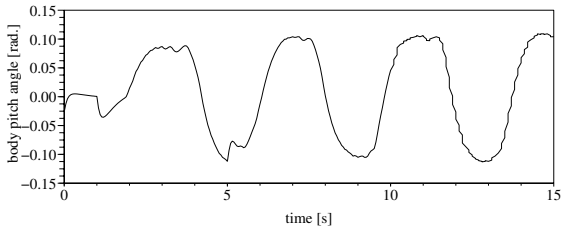


Figure 10. Training Set of Control Response of Body Pitch Angle ψ using Only LQR

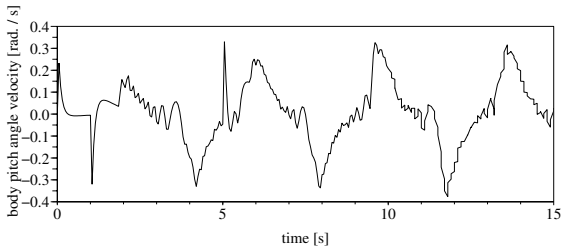


Figure 11. Training Set of Control Response of Body Pitch Angle Velocity $\dot{\psi}$ using Only LQR

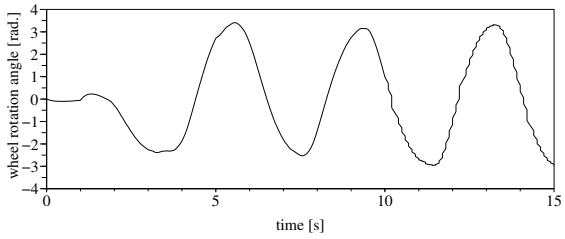


Figure 12. Training Set of Control Response of Wheel Rotation Angle θ using Only LQR

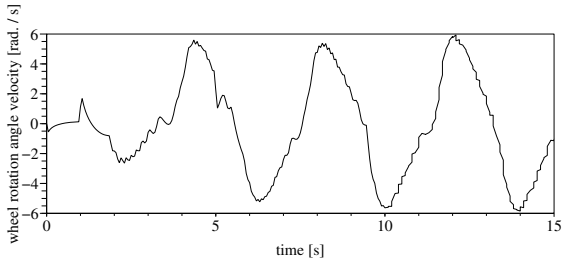


Figure 13. Training Set of Control Response of Wheel Rotation Angle Velocity $\dot{\theta}$ using Only LQR

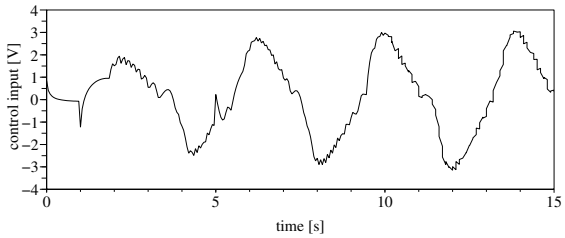


Figure 14. Training Set of the Control Response of the Control Input u using Only LQR

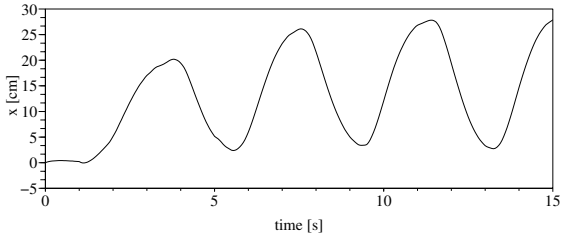


Figure 15. Position of the Inverted Pendulum on x-axis using Only LQR

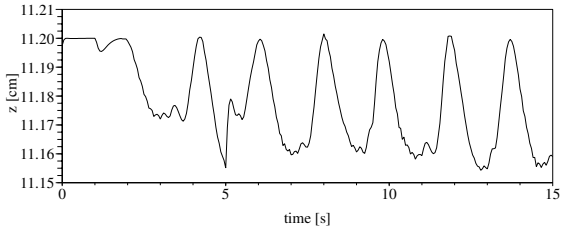


Figure 16. Position of Mass of the Inverted Pendulum on z-axis using Only LQR

Figures 17 to 20 show compensation results of the state of $\mathbf{x}_1$, and fig. 21 shows the compensation result of the control input $\mathbf{u}$. In this section, the part that is given in real training sets will be not considered. Thus, the part of the graph pertaining to the state prediction part shown in T (at $t = 3.00$ [s]) of figs. 17 to 20 will be only argued and focused on. Moreover, figs. 22 and 23 show the position of mass of the inverted pendulum. Figure 24 is enlarging around z position of mass of the inverted pendulum of fig. 23. Here, movement distance for displaying the trajectory was obtained by eqs. (23) and (24).

3.8 Discussion on Simulated Results of the Proposed Method

Here, starting and predicting the state prediction point is shown at $t = 3.00$ [s]. Therefore, we will only argue and focus on the part of the graph pertaining to state prediction part shown in T.

According to these results (figs. 17, 18 and 20), compensation results obtained using the proposed method (shown as the red solid line) approach or converge to near zero with time. Moreover, the wheel rotation angle $\theta(t)$ in fig. 19, a compensation results obtained using the proposed method is driving backward

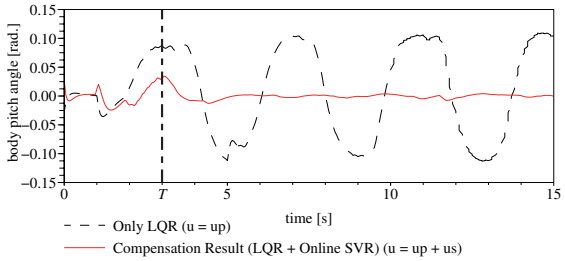


Figure 17. Control Response of Body Pitch Angle ψ using the Proposed Method

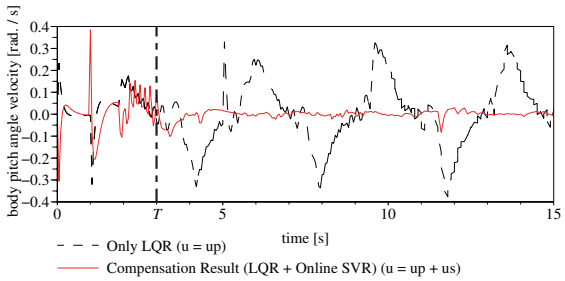


Figure 18. Control Response of Wheel Rotation Angle Velocity $\dot{\psi}$ using the Proposed Method

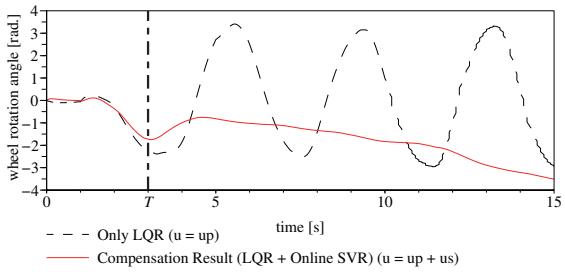


Figure 19. Control Response of Wheel Rotation Angle θ using the Proposed Method

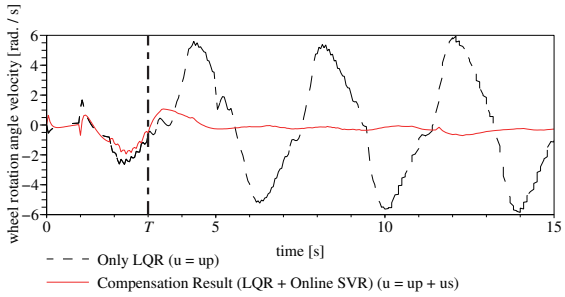


Figure 20. Control Response of Wheel Rotation Angle Velocity $\dot{\theta}$ using the Proposed Method

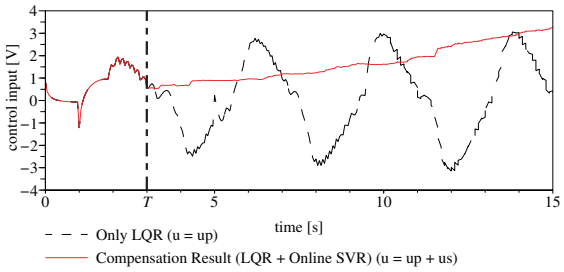


Figure 21. Control Response of the Control Input u using the Proposed Method

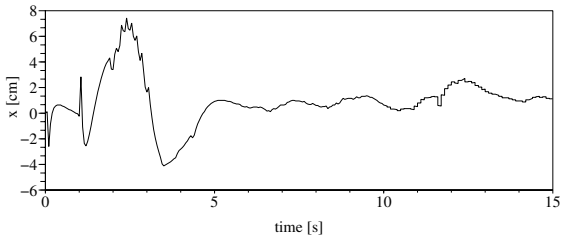


Figure 22. Position of the Inverted Pendulum on x-axis using the Proposed Method

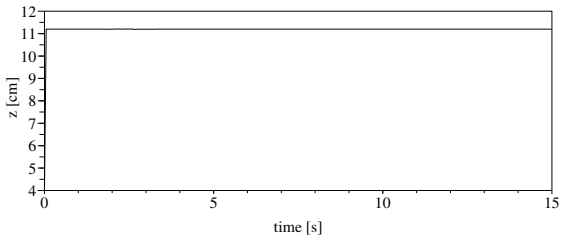


Figure 23. Position of Mass of the Inverted Pendulum on z-axis using the Proposed Method

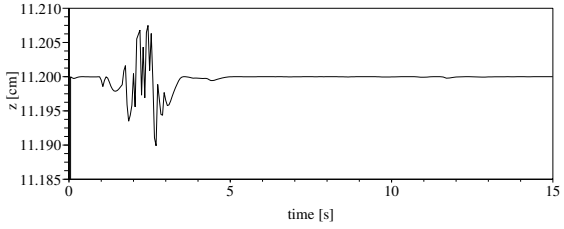


Figure 24. Position of Mass of the Inverted Pendulum on z-axis using the Proposed Method focused around $z = R + L$ [cm] in Figure 23

ISBN: 978-1-941968-17-8 ©2015 SDIWC

or forward, continuously. And also, the control input $u(t)$ in fig. 21, a compensation results obtained using the proposed method is generating a control input, continuously. Next, in fig. 17, the result of the proposed method almost converges to zero. Next, each result will be focused on. From fig. 19, $\theta(t)$ swings considerably and can be confirmed. Therefore, it can be said that the wheel is moving while trying to decrease the body pitch angle $\psi(t)$. And in figs. 18 and 20, these results are moving smaller than other state in advance of the conventional method using only LQR. Finally, in fig. 21, also this result is taken in advance states than the conventional method that using only LQR. As the reason, the compensate control input is combining current action and the action that takes future in advance. In this case, as the action that takes future, compensation control input will be using prediction result of the state-action pair prediction, directly. Therefore, the compensate control input generates the action that considered future disturbance. As the result, the effect of the disturbance will be reduced and the system will be converging in the desirable state.

Next, the movement distance will be focused on. In figure 22, it can be confirmed that the inverted pendulum is moving forward because of it received the forward run from the command input. From this command input, the inverted pendulum is approaching undulating floor. From this result, it can be said that the inverted pendulum moved autonomously based on prediction results, for balancing itself. Therefore, body pitch angle is near zero. Subsequently, the trajectory of mass of the inverted pendulum will be focused on. In figure 23, it cannot be confirmed that the body pitching around balance point. Accordingly, figure 24 will be focused on. In figure 24, it can be confirmed that the mass of the inverted pendulum is slightly pitching around balance point as 0 [deg.] than the method that using only LQR (fig. 16). From this result, also the inverted pendulum moved autonomously based on prediction results, for balancing itself. All the same, it can be confirmed that the body pitch angle is near zero.

Therefore, along with the action predictor, the LQR feedback controller was applied, and this controller maintained the "desired" stable state. In other words, this system stabilizes the inverted pendulum using the current outside data, previous states, and an action. As a result, it can be said that the robot's states will converge to stable state, according to the time course. Furthermore, this system acquires data at each of the sampling times. Using these results, the proposed system derives an action that multiplies states with the optimal feedback gain for obtaining the future state. Thus, the predictors will predict in the direction of stable states, even if there are some disturbances in the environment such as flat floor or undulate floor. In other words, this proposed method is robust with a disturbance that directly affects the application. From these viewpoints, it will be concluded that the experimental results are reasonable.

4 CONCLUSION

In this study, we focused on the relationship between the state and the action of robots. Therefore, on the basis of our former study, we proposed a method that decides an action that the robot will take using the recent tendency of the prediction results every time. Moreover, we applied the LQR to derive an action using the optimal feedback gain, and the weight coefficient was defined by the standard deviation as in the proposed method. Using the proposed method, we obtained the compensated current action for greater convergence in the desirable state. Further, the command input was sent to the inverted pendulum. From this command input, shape of floor was changed from flat to undulating. As a result, we confirmed that the inverted pendulum moved autonomously based on prediction results, for balancing itself.

On the basis of the experimental results, the proposed method could be converged to a desirable state as the optimal solution of this problem using prediction and selection. In other words, the proposed methods can adjust the transition of a robot's state or outside en-

ISBN: 978-1-941968-17-8 ©2015 SDIWC

vironment. Accordingly, we can conclude that the proposed method can predict, using online SVR and LQR and decide an action for the future.

As a future work, we will try to introduce a various types of shape of the floor, and we will confirm that behavior of the proposed method.

REFERENCES

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*, The MIT Press, 2005.

[2] Shun'ichi Asaka and Shigeki Ishikawa, "Behavior Control of an Autonomous Mobile Robot in Dynamically Changing Environment," Journal of the Robotics Society of Japan, vol.12, no.4, pp.583-589, 1994.

[3] Takayuki Kanda, Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, and Ryohei Nakatsu, "Development of "Robovie" as Platform of Everyday-Robot Research," IEICE Transactions on Information and Systems, Pt.1 (Japanese Edition), vol.J85-D-1, no.4, pp.380-389, 2002.

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

[5] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," Journal of Artificial Intelligence Research vol.11, pp.391-427, 1999.

[6] Mohammad Abdel Kareem Jaradata, Mohammad Al-Rousanb, and Lara Quadanb, "Reinforcement based Mobile Robot Navigation in Dynamic Environment," Robotics and Computer-Integrated Manufacturing vol.27, pp.135-149, 2011.

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

[8] Mohammed Faisal, Ramdane Hedjar, Mansour Al Sulaiman, and Khalid Al-Mutib, "Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment," International Journal of Advanced Robotic Systems, DOI: 10.5772/54427, 2012.

[9] Fabrizio Abrate, Basilio Bona, Marina Indri, Stefano Rosa, and Federico 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] Takushi Sogo, Katsumi Kimoto, Hiroshi Ishiguro, and Toru Ishida, "Mobile Robot Navigation by a Distributed Vision System," Journal of the Robotics Society of Japan, vol.17, no.7, pp.1-7, 1999.

[12] Jong Jin Park, Collin Johnson, and Benjamin 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] Edvard Naerum, H. Hawkeye King, and Blake Hannaford, "Robustness of the Unscented Kalman Filter for State and Parameter Estimation in an Elastic Transmission," In Proceedings of the Robotics: Science and Systems, 2009.

[14] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda, "Purposive Behavior Acquisition for a Robot by Vision-Based Reinforcement Learning, " Journal of the Robotics Society of Japan, vol.13, no.1, pp.68-74, 1995.

[15] Norikazu Sugimoto, Kazuyuki Samejima, Kenji Doya, and Mitsuo Kawato, "Reinforcement Learning and Goal Estimation by Multiple Forward and Reward Models," IEICE Transactions on Information and Systems, Pt.2 (Japanese Edition), vol.J87-D-2, no.2, pp.683-694, 2004.

[16] Yasutake Takahashi and Minoru Asada, "Incremental State Space Segmentation for Behavior Learning by Real Robot," Journal of the Robotics Society of Japan, vol.17, no.1, pp.118-124, 1999.

[17] Jongho Shin, H. Jin Kim, Sewook Park, and Youdan Kim, "Model Predictive Flight Control using Adaptive Support Vector Regression," Neurocomputing, vol.73, no.4-6, pp.1031-1037, 2010.

[18] Younggeun Choi, Shin-Young Cheong, and Nicolas Schweighofer, "Local Online Support Vector Regression for Learning Control," In Proceedings

of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation Jacksonville, FL, USA, pp.13-18, 2007.

[19] Erik Schuitema, Lucian Busoniu, Robert Babuska, and Pieter Jonker, "Control Delay in Reinforcement Learning for Real-Time Dynamic Systems: A Memoryless Approach," In Proceedings of Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference, pp.3226-3231, IEEE, 2010.

[20] Thomas J. Walsh, Ali Nouri, Lihong Li, and Micheal L. Littman, "Planning and Learning in Environments with Delayed Feedback," Machine Learning: ECML 2007, pp.442-453, 2007.

[21] Yang Su, Kok Kiong Tan, and Tong Heng Lee, "Computation Delay Compensation for Real Time Implementation of Robust Model Predictive Control," In Proceedings of Industrial Informatics (INDIN), 2012 10th IEEE International Conference, pp.242-247, IEEE, 2012.

[22] Cunjia Liu, Wen-Hua Chen, and John Andrews, "Model Predictive Control for Autonomous Helicopters with Computational Delay," In Proceedings of Control 2010, UKACC International Conference, pp.1-6, IET, 2010.

[23] Giancarlo Marafioti, Sorin Olaru, and Morten 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.

[24] Niko Sünderhauf, Sven Lange, and Peter Protzel, "Using the Unscented Kalman Filter in MonoSLAM with Inverse Depth Parametrization for Autonomous Airship Control," In Proceedings of IEEE International Workshop on SSRR 2007, pp.1-6, 2007.

[25] Mohammad Ali Badamchizadeh, Iraj Hassanzadeh, and Mehdi Abedinpour 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.

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

[27] Ramazan Havangi, Mohammad Ali Nekoui, and Mohammad Teshnehlab, "Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization," IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, pp.15-23, 2010.

[28] Foudil Abdessemed, "SVM-Based Control System for a Robot Manipulator, " Int J Adv Robotic Sy, 2012, vol. 9, 247, DOI: 10.5772/51192, 2012.

[29] Masashi Sugimoto and Kentarou Kurashige, "The Proposal for Prediction of Internal Robot State Based on Internal State and Action," In Proceedings of IWACIII2013 CD-ROM, SS1-2, Oct.18-21, Shanghai, China, 2013.

[30] Masashi Sugimoto and Kentarou Kurashige, "The Proposal for Deciding Effective Action using Prediction of Internal Robot State Based on Internal State and Action," In Proceedings of 2013 International Symposium on Micro-NanoMechatronics and Human Science, pp.221-226, Nov.10-13, Nagoya, Japan, 2013.

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

[32] Masashi Sugimoto and Kentarou 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.

[33] Masashi Sugimoto, Hitoshi Yoshimura, Tsukasa Abe, and Isao 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.

[34] Masashi Sugimoto, Hitoshi Yoshimura, Tsukasa Abe, and Isao Ohmura, "A Study on Model-Based Development of Embedded System using Scilab/Scicos – Development of Auto-Code Generator –," In Proceedings of the 2010 JSME Conference on Robotics and Mechatronics (ROBOMEC' 10), Asahikawa, vol.10, no.4, 2A2-C10, 2010.

[35] Yorihisa Yamamoto, NXTway-GS Model-Based Design –Control of Self-Balancing Two-Wheeled Robot Built with LEGO Mindstorms NXT–. Cybernet Systems Co., Ltd., 2009.

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

[37] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin, "Training and Testing Low-Degree Polynomial Data Mappings Via Linear SVM," J. Machine Learning Research, vol.11, pp.1471-1490, 2010.

[38] John Devcic, "Weighted Moving Averages: The Basics," 2006. [Online]. Available: http://www.investopedia.com/articles/technical/060401.asp