International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 122-131
The Society of Digital Information and Wireless Communications (SDIWC), 2016 ISSN: 2220-9085 (Online); ISSN: 2412-3587 (Print)

# Adaptability to Periodic Variable Disturbance using Probabilistic State-Action Pair Prediction

Masashi Sugimoto[1*] Naoya Iwamoto[1] Robert W. Johnston[1]
Keizo Kanazawa[1] Yukinori Misaki[2] Kentarou Kurashige[3]
[1]National Institute of Technology, Kagawa College
[2]Toyohashi University of Technology
[3]Muroran Institute of Technology
Email(*: corresponding author): sugimoto-m@es.kagawa-nct.ac.jp

## ABSTRACT

When operating a robot in a real environment, its behavior is probabilistic because of slight transition of the robot's state or error in the action taken at a given time. In this case, it is difficult to operate the robot using rule-based-like action decision methods. Therefore, ad-hoc-like action decision methods are needed. A method is proposed for deciding on future actions based on a robot's present information. The state-action pair prediction method has been reported; it links the state and future actions of a robot using internal information. A statistical approach to state-action pair prediction has been introduced previously, in which the existence probability of a state and action in the future is calculated according to the normal distribution. This paper considers the situation where a command input is sent to an inverted pendulum. Based on this command input, the shape of the floor is changed from flat to undulating. The results of verification experiments confirm that the proposed method can adjust the shape of the floor autonomously.

## KEYWORDS

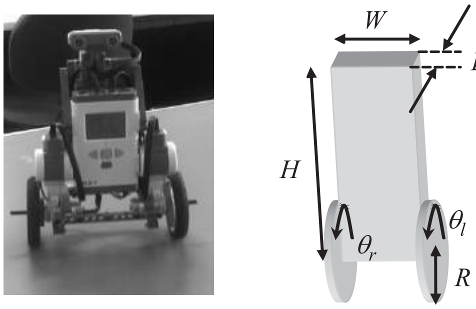
State-Action Pair Prediction, Decision for Optimal Action, Prediction Distribution, Online SVR, Normal Distribution, Online State Prediction

## 1 INTRODUCTION

In controlling robots in dynamic environments, an action may be chosen and adopted based on current state and action results by predicting a future state from previous actions and states [1]. In operating a robot in a real environment, its behavior changes probabilistically due to slight fluctuations in the robot's state or errors in the actions taken at a given time. In this case, a stochastic technique is necessary for handling problems with unknown disturbances [2, 3, 4, 5, 6, 7].

From this standpoint, a stochastic technique that features a state and action pair prediction method is effective and necessary. In a previous study [8], the statistical approach was introduced to state-action pair prediction. In particular, a method that decides which future action to take was proposed given the current action. In the proposed method, the existence probability of a state and action in the future is calculated according to the normal distribution. In a previous study [8], an inverted pendulum was kept balanced while given a predictable unknown disturbance using the proposed method. This study considers the situation where a command input is sent to an inverted pendulum; from this command input, the shape of the floor is changed from flat to undulating.

This paper is organized as follows: in section 2, motivation is provided for using the state-action pair prediction technique for deciding the optimal action for a robot to apply, introducing a stochastic technique. Furthermore, details of stochastic state-action pair prediction are stated. In section 3, a verification experiment configuration using variable floor shape is described. Finally, section 4 presents the conclusions of this study.




(a) Structure (b) 3D Model

**Figure 1.** The Structure of "NXTway-GS."

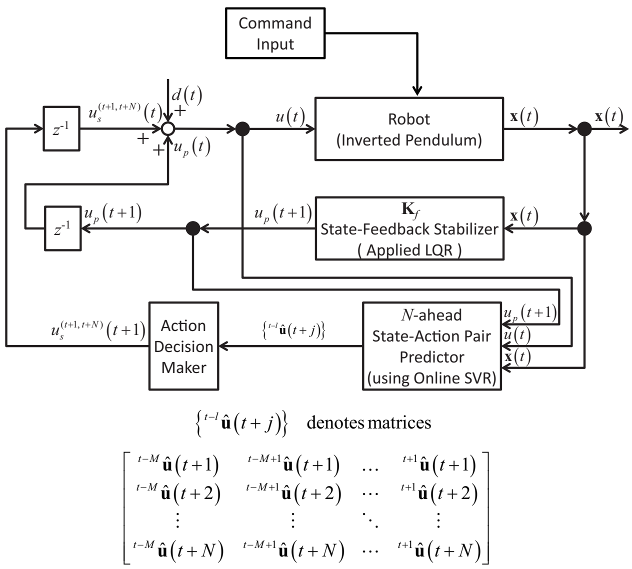



**Figure 2.** Outline of deciding the optimal action for a robot using the prediction of a stochastic state-action pair.

## 2 AN APPROACH FOR INTRODUCING THE NORMAL DISTRIBUTION FUNCTION INTO STATE-ACTION PAIR PREDICTION

In this study, we consider determining the optimal action to minimize the body pitch angle of an inverted pendulum (shown in Fig. 1) under a predictable unknown disturbance given continuously using the state-action pair prediction proposed by our former studies [9, 10]. Therefore, in this paper, the system in Fig. 2 based on the former studies is considered. As shown in Fig. 2, ${}^{t-l}\hat{\mathbf{u}}(t+j)$ describes the prediction result of the control input $\mathbf{u}(t+j)$ when this input is predicted at time $(t-l)$. Therefore,

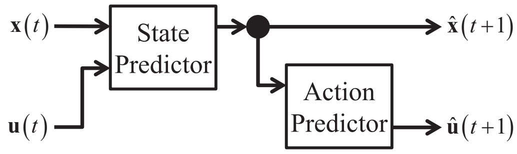


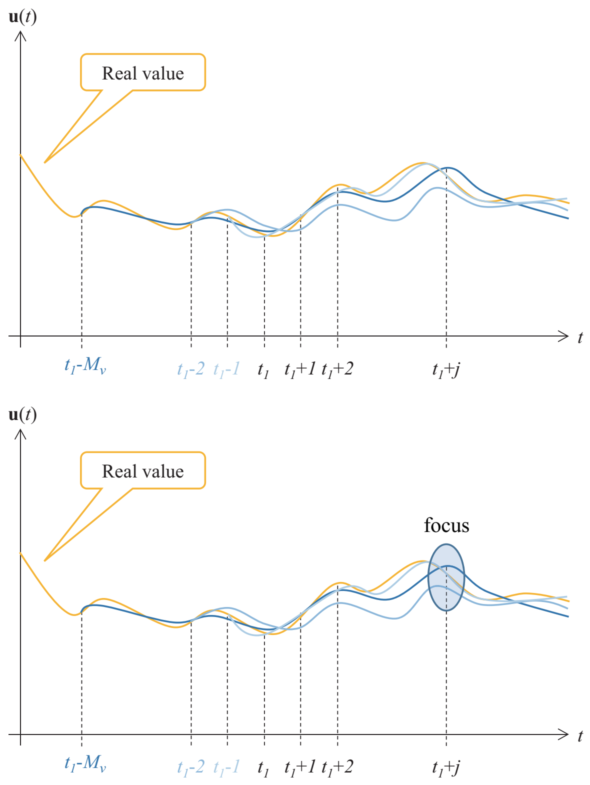

**Figure 3.** Structure of the $N$-ahead state-action pair predictor [1].

this proposed method attempts to compensate for a current action in progress using a combination of optimal control and state-action pair prediction. A structure of prediction of a state-action pair called an "$N$-ahead state-action pair predictor" is the internal structure described in Fig. 3. In this case, if $N$ is larger than the current time $t$, then the prediction error rate is proportional to $N$; moreover, the cumulative prediction error cannot be ignored in a prediction depending on time $t$. In case of stochastic probability for ordinary robot control, probabilistic curves have certain peaks. However, in this study, the maximum peak of probability for a robot action is the focus. From this viewpoint, the normal distribution is applied in this study, and the existence probability of a state and action in the future, according to the normal distribution, is calculated. Moreover, future actions are revised based on the state and action that have the highest existence probability. In addition, this study considers the increasing influence of an action at the latest time $t$, focused on the time at which prediction starts [11].

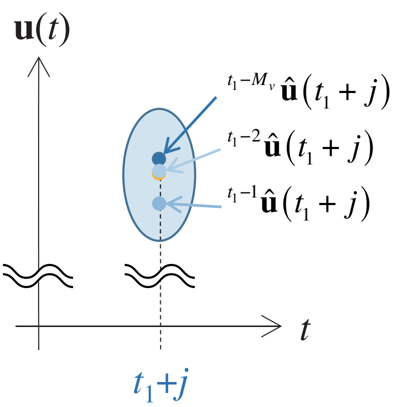
On the basis of this idea, the distribution of existence probability of a prediction series is calculated based on the time at which a prediction is started. In addition, the focus is on the average and standard deviation. The predictor makes predictions continuously at each point in time. Therefore, from this mechanism, the values of prediction results are obtained at each sampling time. Future values at each sampling time are predicted and obtained by a stochastic state-action pair prediction. These results are illustrated in Fig. 4(a). Note the "focus" area in fig. 4(a).

Here, the prediction results at time $(t_1+j)$ can be obtained from each previous time at $(t_1-i)$ (the range of $i$ is given by $(0 \ll M_v \leq i \leq N)$)

International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 122-131
The Society of Digital Information and Wireless Communications (SDIWC), 2016 ISSN: 2220-9085 (Online); ISSN: 2412-3587 (Print)




(a) Series of state-action pair predictions




(b) Focus on variations of predicted values at past times (from Fig. 4(a)).

**Figure 4.** Variations of predicted series using a state-action pair.

as illustrated in Fig. 4(b). In this zone at $(t_1 + j)$, the prediction results are distributed. To quantitatively measure these variations, the average and the standard deviation are emphasized. In brief, the average $\mu_j^{\mathbf{u}}$ and the standard deviation $\sigma_j^{\mathbf{u}}$ of the control input at time $(t_1+j)$ using the values predicted at each $(t_1 - i)$ (as past times) can be derived as follows:

$$\mu_j^{\mathbf{u}} = \mu\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{\mathbf{u}}(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{\mathbf{u}}(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{\mathbf{u}}(t_1+j)\right) \tag{1}$$

$$\sigma_j^{\mathbf{u}} = \sigma\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{\mathbf{u}}(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{\mathbf{u}}(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{\mathbf{u}}(t_1+j)\right) \tag{2}$$

Similarly, for the state $\mathbf{x}(t_1 + j)$, the average $\mu_j^{x_k}$ and standard deviation $\sigma_j^{x_k}$ are derived. Here, $k \in \dim \mathbf{x}$.

$$\mu_j^{x_k} = \mu\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{x}_k(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{x}_k(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{x}_k(t_1+j)\right) \tag{3}$$

$$\sigma_j^{x_k} = \sigma\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{x}_k(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{x}_k(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{x}_k(t_1+j)\right) \tag{4}$$

Hence, $\mu(\cdot)$ denotes the average of $(\cdot)$ and $\sigma(\cdot)$ denotes the standard deviation of $(\cdot)$. Moreover, $w_M, w_{M+1}, \ldots, w_1$ denote weights of the weighted moving average, and $w_\Sigma$ denotes the sum of these weights. From the above expression, the normal distribution function $N_j^{x_k}$ at time $t_1 + j$ and the function $N_j^{\mathbf{u}}$ for the control input $\mathbf{u}$ are derived, as shown in the equations below.

$$N_j^{x_k}(X_k) = \frac{1}{\sqrt{2\pi\left(\sigma_j^{x_k}\right)^2}} \exp\left\{-\frac{\left(X_k - \mu_j^{x_k}\right)^2}{2\left(\sigma_j^{x_k}\right)^2}\right\} \tag{5}$$

$$N_j^{\mathbf{u}}(U) = \frac{1}{\sqrt{2\pi\left(\sigma_j^{\mathbf{u}}\right)^2}} \exp\left\{-\frac{\left(U - \mu_j^{\mathbf{u}}\right)^2}{2\left(\sigma_j^{\mathbf{u}}\right)^2}\right\} \tag{6}$$

Here, $X_k, U$ are random variables:

$$X_k = {}^{t-M}\hat{x}_k\left(t_1 + j\right), \ldots {}^{t-2}\hat{x}_k\left(t_1 + j\right), {}^{t-1}\hat{x}_k\left(t_1 + j\right) \tag{7}$$

$$U = {}^{t-M}\hat{\mathbf{u}}\left(t_1 + j\right), \ldots {}^{t-2}\hat{\mathbf{u}}\left(t_1 + j\right), {}^{t-1}\hat{\mathbf{u}}\left(t_1 + j\right) \tag{8}$$

From this normal distribution function, values larger than the threshold $\vartheta$ are treated as follows:

$$\Pr\left(N_j^{x_k} > \vartheta\right) = N_{j,l}^{x_k} \tag{9}$$

$$\Pr\left(N_j^{\mathbf{u}} > \vartheta\right) = N_{j,l}^{\mathbf{u}} \tag{10}$$

In these equations, $l$ denotes the number of states or actions larger than the threshold $\vartheta$. Therefore, the compensation action regarded in the future can be obtained if an action corresponds to the disturbance input before a future action is created based on probability. In other words, the compensation action for the normal distribution function $N_{j+1,l}^{\mathbf{u}}$ is as follows:

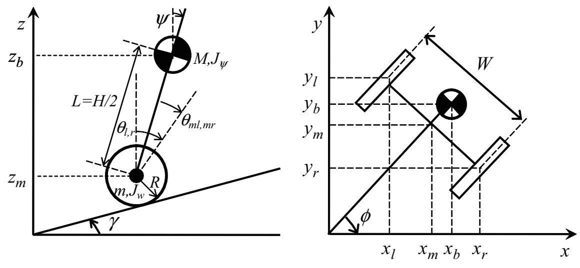
$$N_{j+1,l}^{\mathbf{u}} = -\mathbf{k}_f \cdot N_{j,l}^{\mathbf{x}} \tag{11}$$

In the above equation, $x_k \in \mathbf{x}, 1 \leq l \leq k$. From this equation, a future action is compensated continuously using the largest existence probability as the compensation action at each given time.

## 3 VERIFICATION EXPERIMENT - COMPUTATIONAL SIMULATION USING THE PROPOSED METHOD

### 3.1 Experimental Outline

In this verification experiment, the posture of a two-wheeled inverted pendulum "NXTway-GS" is considered as an application. A computer simulation indicated its stabilization in a verification experiment. Here, we obtained training sets for postural control. The training sets contain states and actions of NXTway-GS. In this experiment, stabilizing the inverted pendulum is considered between using stochastic state-action prediction and an ordinary linear quadratic regulator (LQR) controller. The response of the proposed method using stochastic state-action prediction was compared with the control response of the conventional method using LQR. The experiment included 270 steps (the actual stochastic predictable range was 3.00 [s] to 14.00 [s]).




(a) Side model view (b) Plane model view

**Figure 5.** Model side and plane view for NXTway-GS [12]

### 3.2 Configuration of Simulation 1 for the NXTway-GS model.

As shown in Fig. 5, "NXTway-GS" can be described as a two-wheeled inverted pendulum model. The coordinate system used in section 3.3 is described in Fig. 5. Moreover, in Fig. 5, $\psi$ denotes the body pitch angle and $\theta_{l,r}$ denotes the wheel angle ($l$ and $r$ indicate *left* and *right*, respectively. Furthermore, $\theta = 1/2 \cdot (\theta_l + \theta_r)$, and $\theta_{ml,mr}$ denotes the DC motor angle ($l$ and $r$ indicate *left* and *right*, respectively). The NXTway-GS's physical parameters are listed in Table 1.

### 3.3 Configuration of Simulation 2 – for NXTway-GS modeling.

NXTway-GS's motion equations can be derived as shown in Fig. 5. If the direction of the model is the positive $x$-axis direction at

$t = 0$, the equations of motion for each coordinate can be given as follows [12]:

$$\left[(2m + M) R^2 + 2J_w + 2n^2 J_m\right] \ddot{\theta} + \left(MLR - 2n^2 J_m\right) \ddot{\psi} - Rg (M + 2m) \sin\gamma = F_\theta \quad (12)$$

$$\left(MLR - 2n^2 J_m\right) \ddot{\theta} + \left(ML^2 + J_\psi + 2n^2 J_m\right) \ddot{\psi} - MgL\psi = F_\psi \quad (13)$$

$$\left[\frac{1}{2} m W^2 + J_\phi + \frac{W^2}{2R^2} \left(J_w + n^2 J_m\right)\right] \ddot{\phi} = F_\phi \quad (14)$$

Here, $\mathbf{x}_1$ and $\mathbf{x}_2$ represent state variables. In addition, $\mathbf{u}$ denotes input:

$$\mathbf{x}_1 = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^\top \quad (15)$$

$$\mathbf{x}_2 = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top \quad (16)$$

$$\mathbf{u} = \begin{bmatrix}v_l & v_r\end{bmatrix}^\top \quad (17)$$

From the above equations, the state equations of NXTway-GS can be derived using eqs. (12), (13), and (14).

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \quad (18)$$

$$\frac{d}{dt}\mathbf{x}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \quad (19)$$

In this verification experiment, only the state variable $\mathbf{x}_1$ is used because $\mathbf{x}_1$ contains the body pitch angles as variables $\psi$ and $\dot{\psi}$, which are important for self-balancing. This why planar motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$) is not considered in this experiment:

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} \quad (20)$$

### 3.4 Configuration of Simulation 3 – Applying the Online SVR as a Predictor

In this study, an online SVR is used as a predictor (see Fig. 3). In addition, in this experiment, a radial basis function (RBF) kernel is applied to this predictor. The RBF kernel of samples is notated by $\mathbf{x}$, $\mathbf{x}'$, which represents the feature vectors in any input space and are defined as follows:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\beta \,||\mathbf{x} - \mathbf{x}'||^2\right) \quad (21)$$

Moreover, the predictor's parameters are listed in Table 2. In the table, $i \in \{1, 2, 3, 4\}$.

### 3.5 Configuration of Simulation 4 – Applying the LQR as a Predictor

In this study, a linear quadratic regulator (LQR) is applied as shown in Fig. 3. The feedback gain $\mathbf{k}_f$ is applied so as to minimize the quadratic cost function $J$; this is calculated by the LQR as in eq. (22).

$$J = \int_0^\infty \left[\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)\right] dt \quad (22)$$

In this verification experiment, matrices $\mathbf{Q}$ and $\mathbf{R}$ are defined as follows:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6\times10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4\times10^2 \end{bmatrix} \quad (23)$$

$$\mathbf{R} = 1\times10^3 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (24)$$

In the above equation, $\mathbf{k}_f$ is the gain for optimal feedback obtained by minimizing $J$. From these results, $\mathbf{k}_f$ is applied as the action predictor [1]. Furthermore, the feedback gain $\mathbf{K}_f$ of the state-feedback stabilizer is applied. However, in the verification experiment, planar movement of the two-wheeled inverted pendulum is not considered. Hence, $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$ are considered.

### 3.6 Simulation Conditions – Training Set Acquisition

In this experiment, the command input is sent to the inverted pendulum, and the environment is changed based on the command input. The shape of the floor is changed from flat to undulating. Figure 6 shows the command input, and Fig. 7 shows the shape of the floor. The shape of the floor is given by the following equation:

$$z = 0.01U(x_f)\sin\left[2\times2\cdot\pi\cdot(x_f)\right] \quad \text{[cm]} \quad (25)$$

$$x_f = x - 2 \quad \text{[cm]}$$

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

**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 [12] |
| $R$ | 0.04 | [m] | Wheel radius |
| $J_w$ | $\frac{mR^2}{2}$ | [kgm$^2$] | Wheel inertia moment |
| $M$ | 0.635 | [kg] | Body weight [12] |
| $W$ | 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 |
| $J_\psi$ | $\frac{ML^2}{3}$ | [kgm$^2$] | Body pitch inertia moment |
| $J_\phi$ | $\frac{M(W^2+D^2)}{12}$ | [kgm$^2$] | Body yaw inertia moment |
| $J_m$ | $1 \times 10^{-5}$ | [kgm$^2$] | DC motor inertia moment [12] |
| $R_m$ | 6.69 | [Ω] | DC motor resistance [12] |
| $K_b$ | 0.468 | [V·s/rad.] | DC motor back EMF constant [12] |
| $K_t$ | 0.317 | [N·m/A] | DC motor torque constant [12] |
| $n$ | 1 | [1] | Gear ratio [12] |
| $f_m$ | 0.0022 | [1] | Friction coefficient between body and DC motor [12] |
| $f_W$ | 0 | [1] | Friction coefficient between wheel and floor [12] |

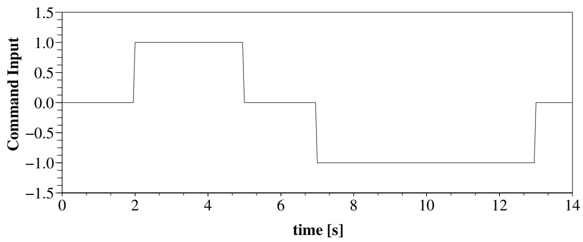



**Figure 6.** Command input signals (+1 indicates running forward; 0 indicates stationary balancing).

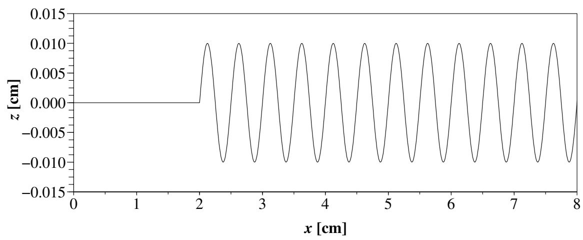



**Figure 7.** Simulation environment (the shape of the floor).

height of the floor. Using the settings mentioned above, we attempted to drive the inverted pendulum model on the floor by using the command input (Fig. 8). Thus, the train-

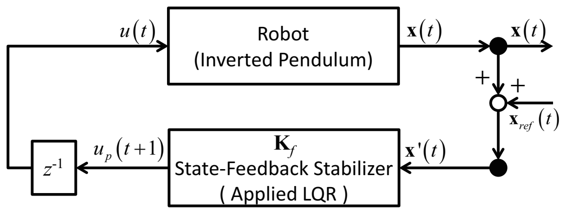



**Figure 8.** Control input obtained by mixing the action and disturbance inputs.

ing sets for the two-wheeled inverted pendulum can be acquired. Here, the movement distance for displaying the position is given by the following equation:

$$x = 100 \cdot R \cdot \int \dot{\theta}(t)dt \quad \text{[cm]} \tag{26}$$

The position of the mass of the inverted pendulum $z_m$ is given in the following equation:

$$z_m = 100 \cdot \left[ R + R\sin\gamma \cdot \int \dot{\theta}(t)dt + L\cos\left\{ \int \dot{\psi}(t)dt \right\} \right] \quad \text{[cm]} \tag{27}$$

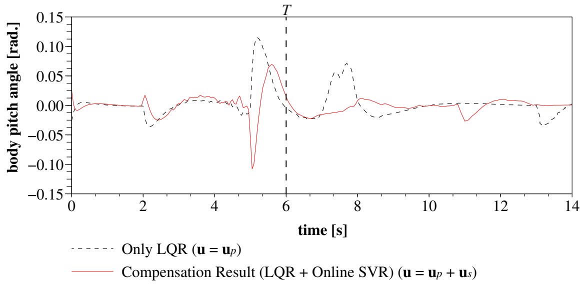



**Figure 9.** Control response for body pitch angle $\psi$ using LQR as the training set vs. control response of the proposed method.

In this verification experiment, the "NXTway-GS" model achieves self-balancing. In addition, the properties of the disturbance signal that we provide as input and other conditions of the simulation are listed in Table 3.

**Table 2.** Learning parameters of the online SVR.

| Symbol | Value | Property |
|---|---|---|
| $C_i$ | 300 | Regularization parameter or predictor of $x_i$ |
| $\epsilon_i$ | 0.02 | Error tolerance for predictor of $x_i$ |
| $\beta_i$ | 30 | Kernel parameter for predictor of $x_i$ |

### 3.7 Simulation Results

In this experiment, the training sets provided to the NXTway-GS are based on predicted results generated by the proposed method. In addition, the NXTway-GS model performs stationary balancing on the basis of these sets, and thereby moves backward or forward. Figures 9 and 10 show the compensation results and ordinary results for comparing state $\mathbf{x}_1$, and Fig. 11 shows the compensation results for control input $\mathbf{u}$. In this section, the part of the graph obtained from the actual training sets is not considered. Thus, only those parts of the graph pertaining to the state prediction part shown in $T$ (at $t = 6.00$ [s]) of Figs. 9 through 11 are analyzed.

Moreover, Figs. 12 and 13 show the position of the mass of the inverted pendulum. The graph in Fig. 14 shows enlargement around the $z$ position of the mass of the inverted pendulum in Fig. 13. Here, the movement distance used for displaying the trajectory was obtained by eqs. (26) and (27).

**Table 3.** Parameters for the simulation using the proposed method.

| Symbol | Value | Unit | Physical property |
|---|---|---|---|
| $\psi_0$ | 0.0262 | [rad.] | Initial value of body pitch angle |
| $\gamma_0$ | 0.0 | [rad.] | Slope angle of movement direction |
| $t_s$ | 0.05 | [s] | Sampling rate |
| $N_s$ | 120 | — | Initial dataset length |
| $N_{\max}$ | 270 | — | Maximum dataset length for the prediction |
| $N$ | 20 | — | Step size of outputs for $N$-ahead state-action pair predictor's outputs |
| $N_\sigma$ | 10 | — | Calculation range of the standard deviation of the predicted values |

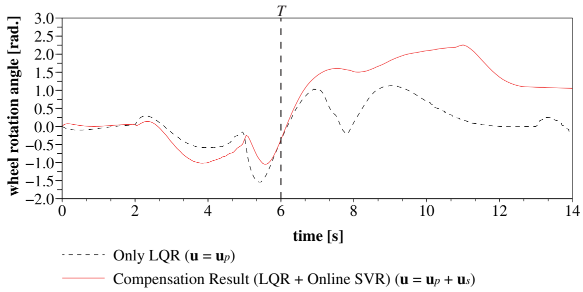



**Figure 10.** Control response of wheel rotation angle $\theta$ using LQR as the training set vs. control response of the proposed method.

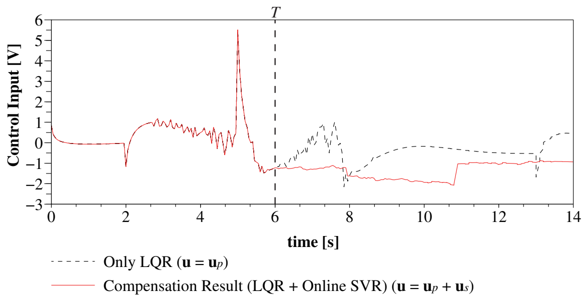



**Figure 11.** Control response of the control input $u$ using LQR as the training set vs. control response of the proposed method.

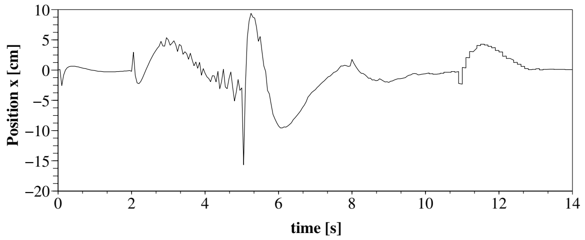



**Figure 12.** Position of the inverted pendulum on the $x$-axis using the proposed method.

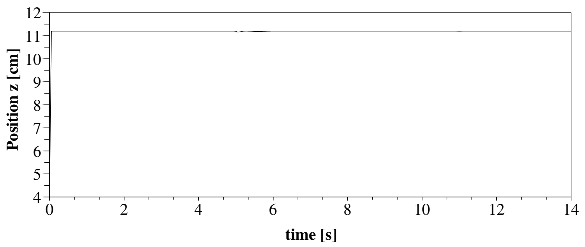



**Figure 13.** Position of the center of mass of the inverted pendulum on the $z$-axis using the proposed method.

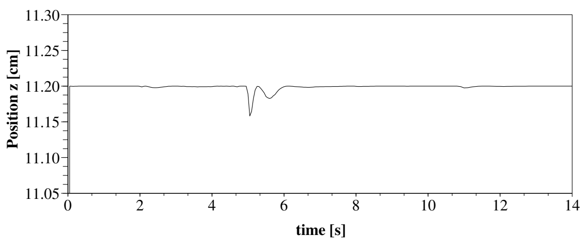



**Figure 14.** Position of the center of mass of the inverted pendulum on the $z$-axis using the proposed method focused around $z = R + L$ [cm] in Figure 13.

### 3.8 Discussion of Simulated Results using the Proposed Method

Here, the state prediction point is shown after $t = 6.00$ [s]. Therefore, we focus only on the part of the graph pertaining to the state prediction part shown in $T$. According to Fig. 9, the compensation result obtained using the proposed method (shown as the solid line) approaches and oscillates around zero with time. From Fig. 10, $\theta(t)$ is rotated backward continuously and can be confirmed. Therefore, the wheel is moving while trying to decrease the body pitch angle $\psi(t)$. However, the wheel rotation angle rotates in the backward direction continuously from 12 [s]. On the other hand, it is confirmed that the slope of the wheel rotation angle is not steep.

Additionally, with the control input $u(t)$ shown in Fig. 11, compensation results obtained using the proposed method generate control input to drive the wheel backward continuously. Furthermore, this result is taken for more advanced states than the conventional method that uses only LQR. This is because the compensated control input combines consideration of the current action and future anticipated action. In this case, as the future action, compensation control input uses the result of state-action pair prediction directly. In other words, $\theta(t)$ and $u(t)$ are influenced by the sum of the compensation input and disturbance input $d(t)$. Therefore, the compensated control input generates an action considering a future disturbance. As a result, the effectiveness of the disturbance signal is reduced. From this system characteristic, the control response for body pitch angle converges to the desired state.

Next, the movement distance is considered. In Fig. 12, it can be confirmed that the inverted pendulum is moving forward because it received a forward-running command input. In parallel, the inverted pendulum is also moving backward. From this command input, the inverted pendulum approaches the undulating floor. From this result it is clear that the inverted pendulum moved autonomously based on predicted results and was able to balance itself. Therefore, the resulting body pitch angle was near zero. Subsequently, the trajectory of the mass of the inverted pendulum is analyzed. In Fig. 13, it cannot be confirmed that the body is pitching around the balance point. Accordingly, Fig. 14 is considered next. In Fig. 14, it can be confirmed that the mass of the inverted pendulum is only slightly pitched around the 0 [deg.] balance point. These results also show that the inverted pendulum moved autonomously based on predicted results, and was able to balance itself. Based on these results, the body pitch angle was confirmed to be near zero.

In this experiment, we only focused on the body pitch angle. From this viewpoint, it can be concluded that the simulation results attain a stable state despite the continuously changing shape of the road. Moreover, this response of the wheel rotation angle can be further improved if the distribution function is reconsidered. Therefore, it is concluded that the results of the verification experiment are reasonable.

## 4 CONCLUSION

In this paper, the relationship between the states and actions of a robot with stochastic properties was examined. To achieve this, on the basis of our previous work, we proposed a method that decides a compensation action at each sampling time based on predictions obtained from recent stochastic tendencies. Moreover, in the proposed method, LQR was applied to derive the action predictor's gain, and the normal probabilistic function was applied to define the weight coefficients. Applying this proposed method, compensated actions for rapid convergence were obtained. In other words, it was determined that the body pitch angle of the NXTway-GS model converged to zero with time.

Using the proposed method, the command input was sent to the inverted pendulum. From this command input, the shape of the floor was changed from flat to undulating. As a result, we confirmed that the inverted pendulum moved autonomously based on the predictions and could balance itself. From the results of the verification experiment, the proposed method could converge to the desired state. In particular, the slope of the body pitch angle of the NXTway-GS model converged to zero based on state and action predictions and decisions. Accordingly, the proposed method which considers a stochastic technique can be adapted to situations of stochastic transitions in external environments.

## REFERENCES

[1] Masashi Sugimoto, Kentarou Kurashige : A Study of Effective Prediction Methods of the State-action Pair for Robot Control using Online SVR, *Journal of Robotics and Mechatronics*, Vol. 27 No. 5, pp. 469-479 (2015)

[2] Mark Cannon, Basil Kouvaritakis, Saša V. Raković, and Qifeng Cheng : Stochastic Tubes in Model Predictive Control with Probabilistic Constraints, Proc. 2010 American Control Conference, pp. 6274-6279 (2010)

[3] Tomoaki Hashimoto : Probabilistic Constrained Model Predictive Control for Linear Discrete-time Systems with Additive Stochastic Disturbances, Proc. IEEE 52nd Annual Conference on Decision and Control, pp. 6434-6439 (2013)

[4] Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C. Williams : A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control, *IEEE Trans. on Robotics*, Vol.26 No. 3, pp. 502-517 (2010)

[5] Andrew Gray, Yiqi Gao, Theresa Lin, J. Karl Hedrick, and Francesco Borrelli : Stochastic Predictive Control for Semi-Autonomous Vehicles with an Uncertain Driver Model, Proc. IEEE 16th Annual Conference on Intelligent Transportation Systems (ITSC 2013), pp. 2329-2334 (2013)

[6] Shridhar K. Shah, Chetan D. Pahlajani, Nicholaus A. Lacock, and Herbert G. Tanner : Stochastic receding horizon control for robots with probabilistic state constraints, Proc. 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 2893-2898 (2012)

[7] Florian Weissel, Thomas Schreiter, Marco F. Huber, and Uwe D. Hanebeck : Stochastic model predictive control of time-variant nonlinear systems with imperfect state information, Proc. 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008), pp. 40-46 (2008)

[8] Masashi Sugimoto, Kentarou Kurashige : A Study on the Deciding an Action Based on the Future Probabilistic Distribution, Intelligent Robotics and Applications, 9th International Conference, ICIRA 2016, Tokyo, Japan, August 22-24, 2016, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9835, pp. 383-394, Springer (2016)

[9] Masashi Sugimoto, Kentarou Kurashge : The Proposal for Real-time Sequential-decision for Optimal Action using Flexible-weight Coefficient based on the State-Action Pair, Proc. 2015 IEEE Congress on Evolutionary Computation, pp. 544-551 (2015)

[10] Masashi Sugimoto, et al. : A Study of Effectiveness of Dynamically Varying Sampling Rate for State-action Pair Prediction, Proc. the Second International Conference on Electronics and Software Science, pp.79-87 (2016)

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

[12] Masashi Sugimoto, Hitoshi Yoshimura, Tsukasa Abe, and Isao Ohmura : A Study on Model-Based Development of Embedded System using Scilab/Scicos, Proc. the Japan Society for Precision Engineering 2010 Spring Meeting, Saitama, D82, pp. 343-344 (2010)