Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Sampling variance update method in Monte Carlo Model Predictive Control [⋆]

Shintaro Nakatani [*] **Hisashi Date** [**]

[*] *Graduate School of Systems and Information Engineering,University of Tsukuba, Ibaraki, Japan (e-mail: nakatani-s@roboken.iit.tsukuba.ac.jp).*
[**] *Faculty of Engineering,Information and Systems,University of Tsukuba, Ibaraki, Japan (e-mail: hdate@iit.tsukuba.ac.jp)*

Abstract: This study describes the influence of user parameters on control performance in a Monte-Carlo model predictive control (MCMPC). MCMPC based on Monte-Carlo sampling depends significantly on the characteristics of sampling distribution. We quantified the effect of user determinable parameters on control performance uisng the relatonship between the algorithm of MCMPC and convergence to the optimal solution. In particular, we investigated the limitations associated with the variance of sampling distribution causing a trade-off relationship with the convergence speed and accuracy of estimation. To overcome this limitation, we proposed two variance updating methods and new MCMPC algorithm. Furthermore, the effectiveness of the numeriacl simulation was verified.

Keywords: Optimal control theory, Monte-Carlo methods, Randomized methods, Model predictive and optimization-based control

1. INTRODUCTION

In recent years, model predictive control (MPC) has attracted considerable attention in various fields owing to its ability of explicitly handling the required constraints Carlos E. Garcia and Morari (1989),Ohtsuka (2004). In MPC, an algorithm is used to determine the optimal control inputs by repeatedly solving the optimization problem with constraint up to a finite time in the future. From the view point of implementation, MPC can be separated into two categories, i.e., gradient and sample-based MPC.

The former method is currently being researched to be applied in various real-world systems. The C/GMRES proposed by Ohtsuka (2004) is a quite efficient methods among gradient-based MPC. The C/GMRES is known to be an efficient algorithm Cairano and Kolmanovsky (2019) for nonlinear systems and has been considered for application in various systems such as smart grid systems Toru (2012) and vehicle collision avoidance control Masashi Nanno (2010).

In gradient-based MPC, the optimal input is determined by solving the optimal control problem using the gradient information of the cost function. Therefore, if the optimal control problem is simple, the optimal solution can be derived quickly and accurately. Alternatively, the target system is limited to systems with differentiable cost function.

In another method, i.e., sample-based MPC, the optimal input is determined using Monte-Carlo approximation. In general, Monte-Carlo method requires a significant number of computational resources; therefore, real-time implementation of sample-based MPC is difficult. However, in literature Williams et al. (2016); Ohyama and Date (2017), it has been reported that the efficient approach is to take advantage of the parallel nature of sampling and use graphical processing unit to implement it in real time. In addition, as sample-based MPC does not require gradient information of the cost function, there are many significant advanteges. The literature Nakatani and Date (2019) describes the feature of the Monte-Carlo model predictive control (MCMPC), which is a type of sample-based MPC. It also explains its capability of handling discontinuous events, based on the result of experiments of collision of pendulum on a cart.

From theoretical point of view, the most successful method is the path integral optimal control framework Kappen (2007); Satoh et al. (2017). The key idea in this framework is that the solution of the optimal control problem is transformed into the expectation over all possible trajectories and corresponding trajectory costs. This transformation allows stochastic optimal control problems to be solved by using a Monte-Carlo approximation with guaranteed convergence. However, in these studies, effect of the variance of sampling distribution on convergence was not considered. Williams et al. (2015) mentions this problem and proposes a framework that allows users to freely determine the variance of the sampling distribution. These previous studies are common in that the theory of path integration is applied to stochastic optimal control problems.

Alternatively, the MCMPC investigated herein aims to overcome the optimal control problem for deterministic systems. Therefore, herein we discuss the convergence of MCMPC by considering the optimal control problem for

[⋆] This work was not supported by any organization discrete-time linear systems, wherein the only optimal solution can be derived analytically.

This study aims to mainly describe the trade-off relationship between the variance of sampling distribution and the convergence, i.e., if we choose large sampling variance, the convergence can be fastened while a large noise remains on the solution. This problem requires that the variance must be properly controlled to perfectly match the sub-optimal input to the optimal solution. This also means that we need to adjust the sampling variance properly to achieve fast convergence and precision at the same time. Two types of variance update methods are proposed: The one is inspired by cooling principle in simulated annealing method and the other is based on the use of the most recent sample variance. These methods are compared in simulation of a linear system. Besides the variance update methods, we also introduce two types of optimization among the Monte Carlo samples: Top-1 sample and weighted mean. Taking the best sample among all samples tends to achieve fast convergence but suffered from large estimation noise compared with weighted mean. These are compared in simulation.

Based on these results, we show that the newly proposed method is one of the effective methods for the problem discussed in this paper.

2. FINITE-TIME OPTIMAL CONTROL PROBLEM FOR DISCRETE-TIME LINEAR SYSTEMS

We considered an optimal control problem for discrete-time linear systems on the k-th control cycle with prediction for I-th steps, indicated by $\{k|0\}, \dots, \{k|i\}, \dots, \{k|I\}$. Consider a class of linear discrete-time systems described by the following equation:

$$x_{\{k|i+1\}} = Ax_{\{k|i\}} + Bu_{\{k|i\}}, \tag{1}$$

where the state is denoted by $x_{\{k|i\}} \in \mathbb{R}^n$, control input by $u_{\{k|i\}} \in \mathbb{R}^1$, and system matrices are denoted by $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times 1}$. In addition, it is assumed that the initial state $x_{\{k|0\}}$ of the system at each control cycle k is known and there are no constraint about input or state for simplicity. For the system (1), the cost function used in the finite-time optimal control problem from the current control cycle to $I-$steps future is described by following equation:

$$J(x_k, u_k, k) = \frac{1}{2}\sum_{i=0}^{N-1}\left(x^T_{\{k|i+1\}}Qx_{\{k|i+1\}} + u^T_{\{k|i\}}Ru_{\{k|i\}}\right), \tag{2}$$

where the $Q \in \mathbb{R}^{n\times n}$ is the positive definite weight for the state, $R \in \mathbb{R}^1$ is the positive definite weight for the input. In the rest of this study, we use J as the cost value unless otherwise noted. Then, the solution of this optimal control problem is defined as

$$u^*_{\{k|i\}} = \arg\min_{u_{\{k|i\}}} J(x_k, u_k, k). \tag{3}$$

At this moment, by using the fact that the time evolution of the system (1) can be expressed using only the initial state $x_{\{k|0\}}$ and input sequences $u_{\{k|0\}}, \cdots, u_{\{k|N-1\}}$, we can rewrite the equation (2) as following equation:

$$J(x_k, u_k, k) = \frac{1}{2}\hat{\mathbf{u}}^T\hat{Q}\hat{\mathbf{u}} + x^T_{\{k|0\}}\hat{B}\hat{\mathbf{u}} + \frac{1}{2}x^T_{\{k|0\}}\hat{A}x_{\{k|0\}}, \tag{4}$$

where the matrices $\hat{A} \in \mathbb{R}^{n\times n}$, $\hat{B} \in \mathbb{R}^{n\times N}$, and $\hat{Q} \in \mathbb{R}^{N\times N}$ and the vector $\hat{u} \in \mathbb{R}^I$, are shown in from (5) to (8).

$$\hat{A} = A^TQA + (A^2)^TQA^2 + \cdots + (A^N)^TQA^N \tag{5}$$

$$\hat{B} = \left[\sum_{k=1}^{N}(A^k)^TQA^{k-1}B,\dots,\sum_{k=j}^{N}(A^k)^TQA^{k-j}B,\dots,(A^N)^TQB\right] \tag{6}$$

$$\hat{Q} = \begin{bmatrix} \hat{q}_{11} & \cdots & \hat{q}_{1j} & \cdots & \hat{q}_{1I} \\ \vdots & \ddots & & & \vdots \\ \hat{q}_{1i} & & \hat{q}_{ij} & & \hat{q}_{iI} \\ \vdots & & & \ddots & \vdots \\ \hat{q}_{1I} & \cdots & \hat{q}_{jI} & \cdots & \hat{q}_{II} \end{bmatrix} \tag{7}$$

$$\hat{\mathbf{u}} = \left[u_{\{k|0\}}, \dots, u_{\{k|I-1\}}\right] \tag{8}$$

The matrix $\hat{Q}$, whose element in the $i-$th row and $j-$th column of the upper triangle, is a symmetric matrix $\hat{Q}$ and is given by

$$\hat{q}_{ij} = \begin{cases} \displaystyle\sum_{k=0}^{N-i} B^T(A^k)^TQA^kB + R, & (i = j) \\ \displaystyle\sum_{k=j-i}^{N-i} B^T(A^k)^TQA^{k+i-j}B. & (i < j) \end{cases} \tag{9}$$

If the matrix $\hat{Q}$ is positive definite symmetric matrix, the unique solution $\mathbf{u}^*$ can be obtained as

$$\mathbf{u}^* = -\hat{Q}^{-1}\hat{B}^Tx_{\{k|0\}}. \tag{10}$$

These discussions so far are a general theory when considering a finite-time optimal control problem using cost function (2) for discrete-time linear systems (1). In the next section, we discuss the relationship between algorithm of MCMPC which takes expectation over all possible trajectories as sub-optimal input and convergence. We also propose an alternative method: Top-1 sample algorithm for MCMPC.

3. ALGORITHM OF TWO TYPES MCMPC

In this section, we describe two different MCMPC algorithms. First, we describe the relationship between convergence and the normal type MCMPC algorithm that uses the expectation over all possible trajectories as sub-optimal inputs. Next, we describe the TOP1 sample MCMPC algorithm that uses the best trajectories from all sample trajectories as a sub-optimal input.

3.1 Relation between algorithm of normal type MCMPC and convergence

Normal type MCMPC consits of three main phases.

Phase 1
Generating input sequenses

Phase 2
Running forward simulation in parallel

Phase 3
Estimating the sub-optimal input sequences $\tilde{\mathbf{u}}$

At the Phase 1, input sequences are generated by random sampling from normal distribution as following equation:

$$\hat{\mathbf{u}} \sim \mathcal{N}(\bar{\mathbf{u}}, \Sigma), \tag{11}$$

where the mean values $\bar{\mathbf{u}}$ is initialized and updated by using following equation:

$$\bar{\mathbf{u}} = \begin{cases} \mathbf{0}, & (k = 0) \\ \left[\tilde{u}_{\{k|0\}}, \ldots, \tilde{u}_{\{k|I-1\}}\right]^T, & (k \neq 0) \end{cases} \tag{12}$$

where $\tilde{u}$ means sub-optimal input estimated in the previous estimation. $\Sigma \in \mathbb{R}^{I \times I}$ is the variance-covariance matrix and satisfies the following two assumptions.

Assumption 1. The standard deviation σ used in all prediction steps is constant.

Assumption 2. For all $u_{\{\cdot|i\}} \in \mathbb{R}^1$, each element are independent from each other:

$$E\left(u_{\{\cdot|i\}} u_{\{\cdot|j\}}\right) = 0, (i \neq j) \tag{13}$$

where $E(\cdot)$ means expected value.

Then, we can describe Σ as following equation (14) using these two assumptions.

$$\Sigma = \begin{bmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{bmatrix} \tag{14}$$

Therefore, $\hat{u}$ can be regarded as a random variable with probability density function (PDF) as shown in the following equation:

$$\begin{aligned} f(\hat{\mathbf{u}}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}(\hat{\mathbf{u}} - \bar{\mathbf{u}})^T \Sigma^{-1} (\hat{\mathbf{u}} - \bar{\mathbf{u}})\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\hat{\mathbf{u}} - \bar{\mathbf{u}})^T (\hat{\mathbf{u}} - \bar{\mathbf{u}})\right). \end{aligned} \tag{15}$$

In Phase 2, the system state for the number of samples used for predicting and estimating is updated using the system model (1) and input sequences sampled randomly as shown in (11). The updated system state and randomly sampled inputs are also used to calculate the cost values $J(x_k, u_k, k)$.

In Phase 3, sub-optimal input sequences $\tilde{\mathbf{u}}$ are derived as the sample mean using the randomly sampled inputs $\hat{\mathbf{u}}$ and the weights $w(\hat{\mathbf{u}})$ for each input sequence:

$$\tilde{\mathbf{u}} = \frac{\sum^{M} w(\hat{\mathbf{u}})\hat{\mathbf{u}}}{\sum^{M} w(\hat{\mathbf{u}})}, \tag{16}$$

where $w(\hat{\mathbf{u}})$ can be derived as the following equation if $\hat{Q}$ is positive definite:

$$\begin{aligned} w(\hat{\mathbf{u}}) &= \exp\left(-\frac{J}{\lambda^2}\right) \\ &= \exp\left(-\frac{1}{2\lambda^2}\hat{\mathbf{u}}^T\hat{Q}\hat{\mathbf{u}} - \frac{1}{\lambda^2}x_0^T\hat{B}\hat{\mathbf{u}} - \frac{1}{2\lambda^2}x^T_{\{k|0\}}\hat{A}x_{\{k|0\}}\right) \\ &= \exp\left(-\frac{1}{2\lambda^2}(\hat{\mathbf{u}} - \mathbf{u}^*)^T\hat{Q}(\hat{\mathbf{u}} - \mathbf{u}^*) + \mathit{const}\right), \end{aligned} \tag{17}$$

where λ is positive constant. Then, $E(\tilde{\mathbf{u}})$, the expected value of the sample mean (16), can be described by following equation:

$$E(\tilde{\mathbf{u}}) = \int w(\hat{\mathbf{u}})\hat{\mathbf{u}}d\hat{\mathbf{u}}. \tag{18}$$

Note that we are interested in the expected value of the function (17) approximated by using a random variable $\hat{\mathbf{u}}$ with the PDF (15). Then, equation (18) can be rewritten as the following equation from the definition of the expectation of the function of random variables:

$$\begin{aligned} E(\tilde{\mathbf{u}}) &= \int w(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) d\hat{\mathbf{u}} \\ &= (\sigma^2\hat{Q} + \lambda^2 I)^{-1}(\sigma^2\hat{Q}\mathbf{u}^* + \lambda^2\bar{\mathbf{u}}), \end{aligned} \tag{19}$$

where $I \in \mathbb{R}^{N \times N}$ is the identity matrix. The derivation of (19) is shown in Appendix A. Then, the variance of the sample mean Σ_S can be expressed by the following equation:

$$\Sigma_S = \frac{\sigma^2\lambda^2}{M}(\sigma^2\hat{Q} + \lambda^2 I)^{-1}, \tag{20}$$

where M is the total number of samples used for the prediction and estimation, (See Appendix A for derivation). Next, we consider the relationship between iteration of prediction and estimation and the convergence of sub-optimal input sequences $\hat{\mathbf{u}}$. Considering about updating the expected value in (11) by repeating the estimation shown in (18), and the sub-optimal input value by the $d-$th estimation is $\bar{\mathbf{u}}_d$, $\bar{\mathbf{u}}_{d+1}$ can be described as

$$\bar{\mathbf{u}}_{d+1} = E(\tilde{\mathbf{u}}) = (\sigma^2\hat{Q} + \lambda^2 I_N)^{-1}(\sigma^2\hat{Q}\mathbf{u}^* + \lambda^2\bar{\mathbf{u}}_d). \tag{21}$$

If we define the error between the optimal input sequences $\mathbf{u}^*$ and the sub-optimal input $\bar{\mathbf{u}}_d$ estimated by the $d-$th estimation as $\mathbf{e}_d = \bar{\mathbf{u}}_d - \mathbf{u}^*$, we can describe the $d+1$-th estimation error as

$$\mathbf{e}_{d+1} = \left(\frac{\sigma^2}{\lambda^2}\hat{Q} + I\right)^{-1} \mathbf{e}_d \tag{22}$$

As a result of the above considerations, we obtain the theorem on the relationship between convergence and parameters specific to MCMPC as shown below.

Theorem 1. In (4), it is assumed that the matrix $\hat{Q}$ is a real positive definite symmetric matrix and the unique optimal inputs sequences exists as shown in (10).

Then, the sub-optimal input $\bar{\mathbf{u}}_d$ converges to $\mathbf{u}^*$ when $d \to \infty$.

Proof. The necessary and sufficient condition for the error $\mathbf{e}_d$ to asymptotically converge to 0 is that the absolute value of all eigenvalues of matrix Ω shown in (23) is less than 1.

$$\Omega = \left(\frac{\sigma^2}{\lambda^2}\hat{Q} + I\right)^{-1} \tag{23}$$

Assuming that for any real positive definite symmetric matrices M_A, M_B, the following inequality holds:

$$\lambda_i(M_A + M_B) > \lambda_i(M_A), \tag{24}$$

where $\lambda_i(Z)$ means the i-th eigenvalue of a matrix Z (Proof omitted.). Based on the assumption that $\hat{Q}$ is a real positive definite symmetric matrix, the following equation holds:

$$\lambda_i\left(\frac{\sigma^2}{\lambda^2}\hat{Q} + I\right) > \lambda_i(I) = 1. \tag{25}$$

Since $\lambda_i(Z^{-1}) = \frac{1}{\lambda_i(Z)}$ holds for any non-singular matrix, the following inequality holds:

$$\lambda_i(\Omega) = \lambda_i\left(\left(\frac{\sigma^2}{\lambda^2}\hat{Q} + I\right)^{-1}\right) < \lambda_i(I). \tag{26}$$

As the eigenvalues of all real positive definite symmetric matrices are positive real numbers, the absolute value of all eigenvalues of the matrix Ω is less than 1.
Then, the error $\mathbf{e}_d$ satisfies the following equation:

$$\lim_{d\to\infty} \mathbf{e}_d = 0. \tag{27}$$

This means:

$$\lim_{d\to\infty} (\bar{\mathbf{u}}_d - \mathbf{u}^*) = 0. \tag{28}$$

Thus, the sub-optimal input sequences $\bar{\mathbf{u}}_d$ converges asymptotically to $\mathbf{u}^*$ when $d \to \infty$. □

Corollary 1. When $\sigma \to \infty$, Eq. (26) satisfies the following equation:

$$\lim_{\sigma\to\infty} \lambda_i\left(\left(\frac{\sigma^2}{\lambda^2}\hat{Q} + I\right)^{-1}\right) = 0, \forall i. \tag{29}$$

Eq. (29) shows that if $\sigma \to \infty$, the first estimation result $\bar{\mathbf{u}}^{(1)}$ satisfies $\bar{\mathbf{u}}^{(1)} = \mathbf{u}^*$. Therefore, if σ is larger, the sub-optimal input sequences $\bar{\mathbf{u}}_d$ converges to the optimal values faster.
Then, the variance-covariance matrix of the sample mean Σ_S shown in Eq. (20) can be described as the following equation:

$$\lim_{\sigma\to\infty} \Sigma_S = \frac{\lambda^2\hat{Q}^{-1}}{M}. \tag{30}$$

Eq. (30) means that if λ is sufficiently small, the variance of the sub-optimal input sequences $\bar{\mathbf{u}}_d$ is small. This consideration is consistent with the results of path integral analysis. Therefore, this means that there is a tradeoff between convergence and variance. Moreover, equation (30) shows that if sample number M is large, the error of the expected value $E(\tilde{\mathbf{u}})$ by the Monte-Carlo approximation is $\mathcal{O}(\frac{1}{\sqrt{M}})$.

Corollary 2. When $\sigma \to 0$, equation (20) satisfies the following equation:

$$\lim_{\sigma\to 0} \Sigma_S = 0, \tag{31}$$

However, the eigenvalue of the coefficient matrix Ω in equation (22) is as shown below:

$$\lim_{\sigma\to 0} \lambda_i\left(\left(\frac{\sigma^2}{\lambda^2}\hat{Q} + I\right)^{-1}\right) = 1, \forall i. \tag{32}$$

These equations show that there is a tradeoff between the convergence and variance of sample mean Σ_S. Equation (31) and (32) show that if the user choses the variance σ^2 as small as possible to eliminate the variance of sample mean Σ_S, the error e_d at the previous estimation will remain. Moreover, if σ is too small, the sub-optimal input sequences $\bar{\mathbf{u}}_d$ slowly converges to the optimal values.

From Corollary 1 and Corollary 2, it is understood that the variance needs to be controlled appropriately to improve the estimation accuracy and convergence speed.

3.2 Algorithm of TOP1 sample MCMPC

In Top1 sample MCMPC, the optimization problem is solved by iterating the following three processes within the same control cycle.

Phase 1
Generating input sequences
Phase 2
Running forward simulation in parallel
Phase 3
Estimating the sub-optimal input sequences $\tilde{\mathbf{u}}$ and updating standard deviation σ.

Phase 1 and phase 2 are the same as the MCMPC algorithm described above.

In phase 3, sub-optimal input sequences $\tilde{\mathbf{u}}$ is described by the following equation:

$$\tilde{\mathbf{u}} = \arg\min_{u_{\{k|i\}}\in\mathbf{U}} J(x_k, u_k, k), \tag{33}$$

where $\mathbf{U}$ means a set of all inputs sequences $\hat{\mathbf{u}}$ randomly sampled in phase 1. In addition, the standard deviation σ updated as described in section 4

3.3 Model predictive control algorithm

So far we have described how to repeat the prediction in one control cycle. In the model predictive control we propose, the prediction is repeated every control cycle, and the sub-optimal input predicted in the previous control cycle is re-optimized. So, sub-optimal input in k-th control cycle correspond to the result of iteration of $k \times d$ times predictions.

4. SAMPLING VARIANCE UPDATE METHODS

In this section, we describe two types of update methods that are used each time of the iteration of precision. The first variance update method used in this study can be described as following equation:

$$\sigma_d = \gamma^d \sigma_0, \tag{34}$$

where γ is a positive constant $\gamma \in [0.8, 1.0)$, and d is the number of iteration, and σ_0 is a parameter that represents the initial standard deviation that should be designed by the user. Equation (34) is inspired by the cooling schedule used in the simulated annealing (SA) method. In SA, it is guaranteed that the estimated value can reach the optimal solution when γ is chosen appropriately and cooled enough times. For example, if we chose $\gamma = 1/log(1+d)$, estimated value reliably converges to optimal value. But, the cooling rate $\gamma = 1/log(1 + d)$ is too slow, so, in practically, the cooling rate $\gamma \in [0.8, 1.0)$ is generally used Rosen and Nakano (1994).

The second method can be described by the following equation:

$$\sigma_d = \sqrt{\frac{1}{\sum^{M} w_{d-1}(\hat{\mathbf{u}})}}. \tag{35}$$

Equation (35) corresponds to the error variance of equation (16) that can be calculated based on the error propagation law. Note that equation (35) is a variance update method that reflects the quality of the estimation results. In the rest of this study, we will refer to the method shown earlier as the geometric cooling method and the method shown later as latest sample variance method.

5. NUMERICAL SIMULATION

In this section, we first show the models used in two different numerical simulations. Next, we show the simulation results when using normal type MCMPC, which shows the effect of variance σ on convergence. Furthermore, we show the results of applying the two types of variance update methods shown in the subsection 4 to normal type MCMPC and Top1 sample MCMPC. Finally, we show the results of the application to the problem of swing-up stabilization of a double inverted pendulum, which is a type of nonlinear system.

5.1 Simulation models

Example 1. As the first example, we consider the optimal control problem when MCMPC is applied to a three-dimensional unstable discrete-time linear system that can be described by the following equation:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ x_k &\in \mathbb{R}^3, u_k \in \mathbb{R}^1 \end{aligned} \tag{36}$$

where we denote coefficient matrices A and B as show in the following equations:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1.1364 & 0.273 \\ 0 & -0.1339 & -0.1071 \end{bmatrix} \tag{37}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0.0893 \end{bmatrix}, \tag{38}$$

then, the eigenvalues of A are as $\Lambda = [0, -1.1059, -0.1376]^T$. Since one of eigenvalues of A exists outside of the unit circle, system (36) is an unstable system. Then we consider an optimal control problem for system (36) that takes a prediction horizon $N = 15$, initial state $x_0 = [2.98, 0.7, 0.0]^T$, state weight matrix Q and an input weight R as follow:

$$Q = \text{diag}(2.0, 1.0, 0.1), \quad R = 1. \tag{39}$$

Then, the optimal input sequences $\mathbf{u}^*$ can be easily calculated using equation (3). In this study, we show only the analytical solution $u_0^* = -2.69$ used in the following discussion.

Example 2. As the second example, we consider the swing-up stabilization of an arm type double inverted pendulum.

Table 1. Parameters of arm type double pendulum

Name	Symbol $(\cdot)$	Value
Angle of the first link	θ_1 (rad)	Variable
Angle of the second link	θ_2 (rad)	Variable
First link drive torque	τ_1 $(\mathrm{N \cdot m})$	Variable
Mass of first link	m_1 (kg)	–
Mass of second link	m_2 (kg)	9.60×10^{-2}
Coefficient of friction	μ_2 $(\mathrm{kg \cdot m^2 s^{-1}})$	1.26×10^{-4}
Gravity acceleration	g $(\mathrm{ms^{-2}})$	9.81
Length of first link	L_1 (m)	2.27×10^{-1}
Length of second link	l_2 (m)	1.95×10^{-1}
Moment of inertia	J_2 $(\mathrm{kg \cdot m^2})$	1.10×10^{-3}
Positive constant	a_1	6.29
Positive constant	b_1	1.64×10^1

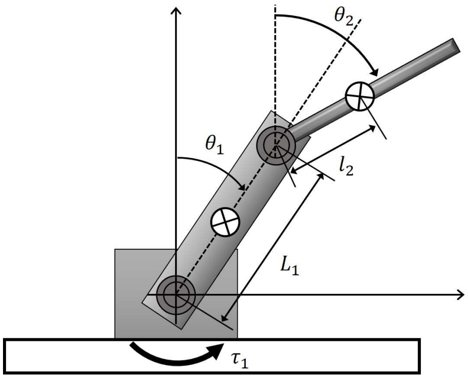

Fig. 1. Model of arm type double pendulum

The state equation of the arm type double inverted pendulum shown in Fig. 1 can be described by the following two equations:

$$\ddot{\theta}_1(t) = -a_1\dot{\theta}_1(t) + b_1u(t) \tag{40}$$

$$\begin{aligned} \alpha_1 \cos\theta_{12}(t)\cdot\ddot{\theta}_1(t) + \alpha_2\ddot{\theta}_2(t) = \alpha_1\dot{\theta}^2(t)\sin\theta_{12}(t) + \alpha_3\sin\theta_2(t) \\ + \mu_2\dot{\theta}_1(t) - \mu_2\dot{\theta}_2(t) \end{aligned} \tag{41}$$

The time-invariant parameters α_1, α_2, and α_3 and the variable θ_{12} in Equation (40) and Equation (41) are as follows:

$$\begin{aligned} \alpha_1 &= m_2L_1l_2, \quad \alpha_2 = J_2 + m_2l_2^2 \\ \alpha_3 &= m_2l_2g, \quad \theta_{12}(t) = \theta_1(t) - \theta_2(t). \end{aligned} \tag{42}$$

The parameters of equations (40) to (42) and Fig. 1 are listed in Table 2. Then we consider an optimal control problem for this example that takes a prediction horizon $N = 80$, initial state shown in equation (43), state weight matrix Q and an input weight R shown in equation (44).

$$\left[\, \theta_1(0),\ \dot{\theta}_1(0),\ \theta_2(0),\ \dot{\theta}_2(0) \,\right] = \left[\, \pi,\ 0,\ \pi,\ 0, \right]. \tag{43}$$

$$Q = \text{diag}(5.0, 0.01, 5.0, 0.01), \quad R = 1. \tag{44}$$

5.2 Trade-off between precision and convergence

In this subsection, we consider the relationship between the variance σ of the sampling distribution and convergence using the result of applying normal type MCMPC to Example 1. Fig. 3 shows the average and standard deviation 3σ of the simulation results of 30 independent trials under each condition.

Table 2. Parameters (for Example 1)

Name	Symbol	Value
Num of predictive steps	N	15 step
Num of samples	M	5,000
Num of iterations	d	100
Variance	σ^2	Variable value
Variance	λ	6.3

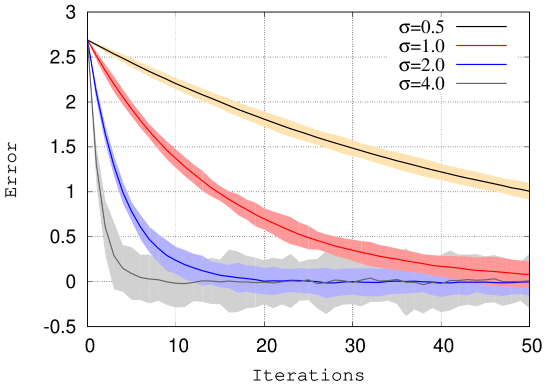

Fig. 2. Effect of σ on estimation error $e_0 = \tilde{u}_0 - u_0^*$ in the Example 1

Table 2 lists the specific parameters of MCMPC used in this simulation to confirm the relationship between variance σ and convergence. In Fig. 3, we compare the result when σ gradually increase to $0.5, 1.0, 2.0, 4.0$. As σ increases, error e_0 converges to 0 with fewer iterations. However, it can e confirmed that the variation in error e_0 as the variance σ increases. This result is a good example showing that the variance σ of sampling distribution results in a trade-off relationship between the speed of convergence and the accuracy of the estimated sub-optimal inputs at the time of convergence.

From the results shown in Fig.3, it is necessary to update the variance σ appropriately to obtain the optimal inputs faster and more accurately.

5.3 *Comparison of sampling variance update methods*

Fig. 3 shows the result obtained by using geometric cooling method, as shown in (34). Then, we plotted the result of the average of 30 independent trails and range of the standard deviation 3σ in Fig. 3. The upper figure shows the result obtained using normal type MCMPC, whereas the lower figure shows the results obtained using Top1 sample MCMPC. We determined γ in equation (34) using the following equation:

$$\gamma = \exp\left(\frac{1}{D}\log\left(\frac{\delta}{\sigma_0}\right)\right) \tag{45}$$

where D number of iterations, σ_0 is initial variance σ of sampling distribution, and δ is variance σ of sampling distribution used in the $D-$th iterations. In this simulation, the conditions of $D = 100$,$\delta = 10^{-5}$ remained, and the value of σ_0 was changed from 0.5 to 4.0. In the upper figure in Fig. 3, it can be confirmed that the error e_0 may or may not converge to 0 depending on the initial variance σ_0. On the contrary, in the lower figure in Fig. 3, the error e_0 converges to 0 at any initial variance. In either case, the variation with respect to the estimated sub-optimal input can be reduced. When normal type MCMPC was applied, the error e_0 in result did not converge to 0 when the initial variance σ_0 was set considerably small because σ_d converged earlier than error e_0.

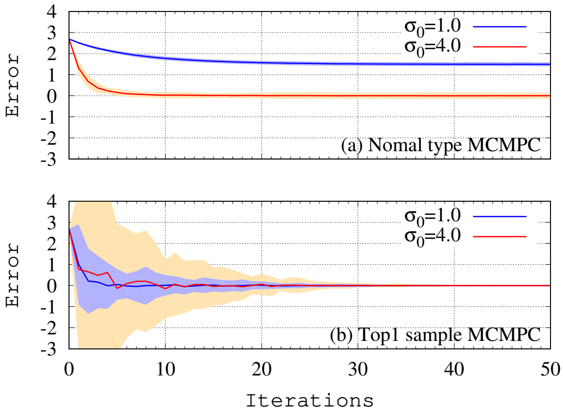

Fig. 3. Effect of σ on estimation error $e_0 = \tilde{u}_0 - u_0^*$ in the Example 1 when using geometric cooling method. (This figure shows results of mean and variance 3σ of 30 trials.)

Fig. 4 shows the result obtained by applying latest sample variance method, as shown in (35). In the upper figure, which shows the result obtained by applying the normal type MCMPC, it can be confirmed that the error e_0 did not converge because σ converged earlier than error e_0. Alternatively, when the TOP1 sample MCMPC, as shown in the lower figure in Fig. 4, is applied, the erro e_0 and variation in the error e_0 of results converged near 0.

These results shown in Fig. 3 and Fig. 4 indicate that the two variance update methods proposed in this study cannot improve the trade off relationship between the convergence speed and the estimation accuracy when the normal type MCMPC is applied. However, when the update method shown in (34) is applied, choosing the appropriate (i.e., sufficiently large) initial variance can improve the trade-off relationship. On the other hand, in the case of TOP1 sample MCMPC, any of the updating methods can reliably converge to the optimal solution if sufficient iteration is taken. This means that TOP1 sample MCMPC has high affinity with any distribution update method.

5.4 *Application to a nonlinear system*

In this section, we show the results of applying what we have analogized so far to nonlinear systems. The discussion of convergence for the linear system can be applied to a nonlinear system that can be linearly approximated around the optimal solution. The system model and cost function are shown in Example 2. The parameters of the controller used for this simulation are as shown in Table 3. We set the initial variance to the lower bound given by:

$$\sigma_0 \geq \frac{u_{max} - u_{min}}{6}. \tag{46}$$

The method of determining the variance σ_0 as in equation (46) is also used in Nakatani and Date (2019). Fig. 5 shows time responses of θ_1, θ_2, $\dot{\theta}_1$, $\dot{\theta}_2$, respectively, and shows a plot of the average value of 30 trails and a stan-

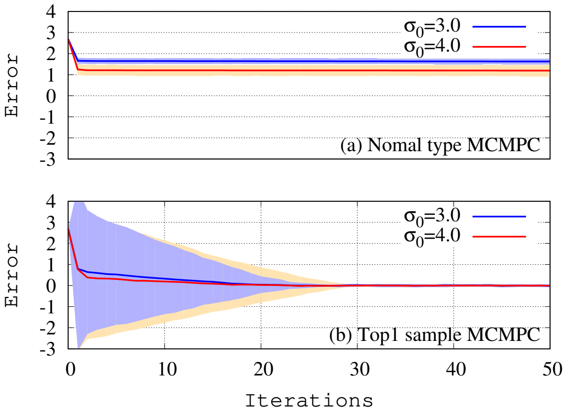

Fig. 4. Effect of σ on estimation error $e_0 = \tilde{u}_0 - u_0^*$ in the Example 1 when using latest sample variance method. (This figure shows results of mean and variance 3σ of 30 trials.)

dard deviation 3σ. In addition, (a) shows in the figure corresponds to the result of applying the TOP1 sample MCMPC, and (b) is the result of applying the normal type MCMPC. When the variance update method considered in this study was applied to normal type MCMPC, none of the methods achieved swing-up stabilization. For this reason, the result shown in Fig. 5 is a result of applying normal type MCMPC without variance updating. Moreover, the result of TOP1 sample MCMPC is the result of using the variance update method shown in equation (34). In addition, the variance σ used in this simulation was one with the best performance among the five different simulations using variance $\sigma_0^2 = 0.5, 1.0, 2.0, 3.0, 4.0$ in normal type MCMPC. Both controllers stabilized the swing up in approximately 2.0 s after the start of control.

The upper figure in Fig. 6 and Fig. 7 shows the input sequences. Immediately after the start of control, TOP1 sample MCMPC selects the smallest input that satisfies the input constraints. On the contrary, the normal type MCMPC selects the conservative input. The lower figure in Fig. 6 and Fig. 7 shows the value of the cost function calculated based on the input sequences predicted in each control cycle. The smaller the value shown in Fig. 6 in each control cycle, the better the control performance. According to the results shown in this study, the TOP1 sample MCMPC demonstrates superior control performance. Moreover, this result was the same when the initial variance σ_0 and the variance update method were changed.

In normal type MCMPC, when the variance σ or the distributed update method was changed, the control performance deteriorated or the swing-up stability could not be stabilized due to the trade-off relationship described in subsection 3.1.

6. CONCLUSION

Herein, we examined the relationship between the convergence of MCMPC and user determinable parameters. Additionally, it was analytically verified that the variance σ of sampling distribution has a trade off relationship with the convergence speed and the accuracy of estimation. Next, we proposed two types of variance update methods and TOP1 sample MCMPC to overcome this trade-off problem. Finally, we completed numerical simulations and discussed the effects of applying the variance update method and TOP1 sample MCMPC. We also showed an example of numerical simulation applied to a nonlinear system and examined the possibility of applying the proposed analogy for controlling nonlinear systems.

Table 3. Parameters (for Example 2)

Name	Value
Simulation time	5.0 (s)
Control cycle	100 (Hz)
Prediction horizon	0.8 (s)
Num of predictive steps	80 step
Num of samples	5,000
Num of iterations	100
σ_0^2 or σ^2	1.0
λ^2	40
γ	0.9
Input constraint	$-3.0 \leq u(t) \leq 3.0$ (V)

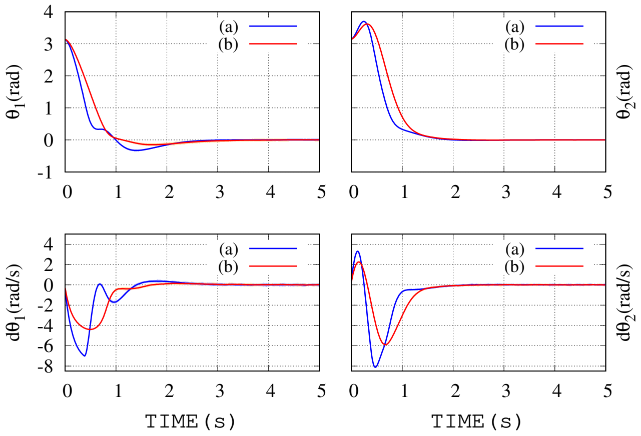

Fig. 5. Simulation result ((a) TOP1 sample MCMPC vs (b) Normal type MCMPC). Left side top: time response of θ_1. Right side top: time response of θ_2. Left side bottom: time response of $\dot{\theta}_1$. Right side bottom: time response of $\dot{\theta}_2$.

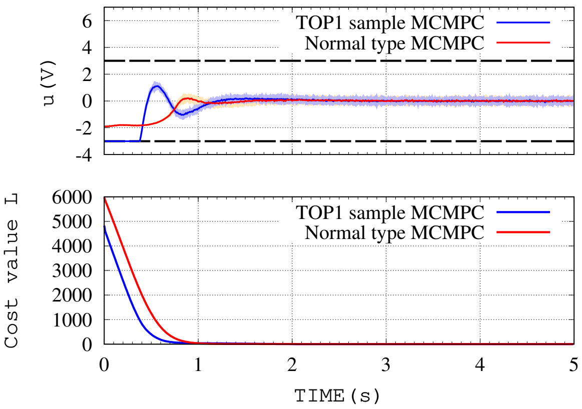

Fig. 6. Top: Simulation result of input sequences. Bottom: Cost value caluclated in each control cycle.(This figure shows results of mean and variance 3σ of 30 trials.)

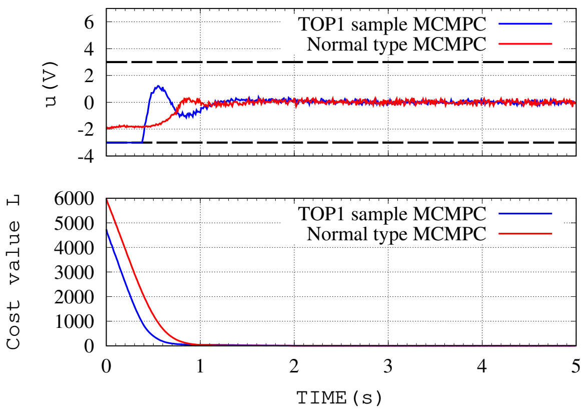

Fig. 7. Top: Simulation result of input sequences. Bottom: Cost value caluclated in each control cycle.(This figure shows results of one trail out of 30 trials.)

REFERENCES

Cairano, S.D. and Kolmanovsky, I.V. (2019). Automotive applications of model predictive control. In *Handbook of Model Predictive Control*, 493–527. Springer International Publishing, Cham.

Carlos E. Garcia, D.M.P. and Morari, M. (1989). Model predictive control: Theory and practice–a survey. *Automatica*, 25, 335–348.

Kappen, H.J. (2007). An introduction to stochastic control theory, path integrals and reinforcement learning. *Proc. 9th Granada seminor on computational physics: Cooperative behavior in nearal systems*, 149–181.

Masashi Nanno, T.O. (2010). Nonlinear model predictive control for vehicle collsion avoidance using c/gmres algorithm. *presented at the 2010 IEEE International Conference on Control Applications*, Yokohama, Japan, September 8–10.

Nakatani, S. and Date, H. (2019). Swing up control of inverted pendulum on a cart with collision by monte carlo model predictive control. *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 1050–1055.

Namekawa, T. (2012). Distributed and predictive control for smart grids. *Journal of the Society of Instrument and Control Engineers*, 51, 62–68.

Ohtsuka, T. (2004). A continuation/gmres method for fast computation of nonlinear receding horizon control. *Automatica*, 40, 563–574.

Ohyama, S. and Date, H. (2017). Parallelized nonlinear model predictive control on gpu. *2017 11th Asian Control Conference (ASCC)*, Gold Coast, QLD, 1620–1625.

Rosen, E.B. and Nakano, R. (1994). Simulated annealing : Basics and recent topics on simulated annealing [in japanese]. *Journal of Japanese Society for Artificial Intelligence*, 365–372.

Satoh, S., Kappen, H.J., and Saeki, M. (2017). An iterative method for nonlinear stochastic optimal control based on path integrals. *IEEE Transactions on Auomatic Control*, 62, 262–276.

Williams, G., Aldrich, A., and Theodorou, E. (2015). Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*.

Williams, G., Paul Drews, B.G., Rehg, J.M., and Theodorou, E.A. (2016). Aggressive driving with model predictive path integral control. *IEEE internatinal Conference on Robotics and Automation(ICRA)*, Stockholm, Sweden, 1433–1440.

Appendix A. DERIVATION OF SAMPLE MEAN EXPECTATION AND VARIANCE OF SAMPLE MEAN

In this section, we describe how to derive the analytical solution (21) from Eq. (18). Substituting the results of Eq. (15) and Eq. (17) for Eq. (18) can be transformed as:

$$
\begin{aligned}
&E(\tilde{u})\\
&=\frac{1}{\sqrt{2\pi\sigma^2}}\int \exp\left(-\frac{1}{2\lambda^2}(\hat{\mathbf{u}}-\mathbf{u}^*)^T\hat{Q}(\hat{\mathbf{u}}-\mathbf{u}^*)-(\hat{\mathbf{u}}-\bar{\mathbf{u}})^T\Sigma^{-1}(\hat{\mathbf{u}}-\bar{\mathbf{u}})\right)d\hat{\mathbf{u}}\\
&=\bar{C}_1\int \exp\left(-\frac{1}{2\lambda^2}(\hat{\mathbf{u}}-\mathbf{u}^*)^T\hat{Q}(\hat{\mathbf{u}}-\mathbf{u}^*)-\frac{1}{2\sigma^2}(\hat{\mathbf{u}}-\bar{\mathbf{u}})^T(\hat{\mathbf{u}}-\bar{\mathbf{u}})\right)d\hat{\mathbf{u}}\\
&=\bar{C}_2\int \exp\left(-\hat{\mathbf{u}}^T\left(\frac{1}{2\lambda^2}\hat{Q}+\frac{1}{2\sigma^2}I\right)\hat{\mathbf{u}}+\frac{1}{\lambda^2}(\mathbf{u}^*)^T\hat{Q}\hat{\mathbf{u}}+\frac{1}{\sigma^2}\bar{\mathbf{u}}^T\hat{\mathbf{u}}\right)d\hat{\mathbf{u}}\\
&=\bar{C}_3\int \exp\left(-\frac{1}{2\lambda^2\sigma^2}\hat{\mathbf{u}}^T\left(\sigma^2\hat{Q}+\lambda^2 I\right)\hat{\mathbf{u}}+\frac{\sigma^2}{\lambda^2\sigma^2}(\mathbf{u}^*)^T\hat{Q}\hat{\mathbf{u}}+\frac{\lambda^2}{\lambda^2\sigma^2}\bar{\mathbf{u}}^T\hat{\mathbf{u}}\right)d\hat{\mathbf{u}}\\
&=\bar{C}_4\int \exp\left(-\frac{1}{2\lambda^2\sigma^2}(\hat{\mathbf{u}}-\tilde{\mathbf{u}})^T\left(\sigma^2\hat{Q}+\lambda^2 I\right)(\hat{\mathbf{u}}-\tilde{\mathbf{u}})\right)d\hat{\mathbf{u}} \qquad \text{(A.1)}
\end{aligned}
$$

where $\bar{C}_1, \bar{C}_2, \bar{C}_3$ and $\bar{C}_4$ are equivalent to terms that are listed as constants to arrange them into terms of the quadratic form and other terms related to $\hat{\mathbf{u}}$, respectively. Then, we define the contents of the exponential function on the fourth line in Eq. (A.1) as g, and obtain a stationary point by partial differentiation of g with $\hat{\mathbf{u}}$ to obtain the following result:

$$
\left.\frac{\partial g}{\partial \hat{\mathbf{u}}}\right|_{\hat{\mathbf{u}}=\tilde{\mathbf{u}}} = (\sigma^2\hat{Q}+\lambda^2 I)\tilde{\mathbf{u}}-(\sigma^2\hat{Q}\mathbf{u}^*+\lambda^2\bar{\mathbf{u}})=0. \qquad \text{(A.2)}
$$

Here, solving the Eq. (A.2) for $\tilde{\mathbf{u}}$ agrees with the result of Eq. (21).

Next, we find the variance of sample mean $\tilde{\mathbf{u}}$ using Eq. (A.1). Let random variable $\hat{\mathbf{u}}$ be a random variable that follows a multidimensional normal distribution with expected value $\tilde{\mathbf{u}}$ and variance Σ_S. From the PDF of this distribution and the result of the coefficient comparison of the integrand on the fifth line in Eq. (A.1), the variance of Σ_S can be shown as:

$$
\frac{1}{2\lambda^2\sigma^2}\left(\sigma^2\hat{Q}+\lambda^2 I\right)=\frac{1}{2}\Sigma_S^{-1} \qquad \text{(A.3)}
$$