Time-Correlated Model Predictive Path Integral: Smooth Action Generation for Sampling-Based Control

Minhyeong Lee and Dongjun Lee

***Abstract*— In this paper, we introduce time-correlated model predictive path integral (TC-MPPI), a novel approach to mitigate action noise in sampling-based control methods. Unlike conventional smoothing techniques that rely on post-processing or additional state variables, TC-MPPI directly incorporates temporal correlation of actions into stochastic optimal control, effectively enforcing quadratic costs on action derivatives. This reformulation enables us to generate smooth action sequences without extra modifications, using a time-correlated and conditional Gaussian sampling distribution. We demonstrate the effectiveness of our approach through simulations on various robotic platforms, including a pendulum, cart-pole, 2D bicopter, 3D quadcopter, and autonomous vehicle. Simulation videos are available at https://youtu.be/nWfJ2MAV2JI.**

I. Introduction

Model predictive control (MPC) is a powerful framework for managing robotic systems. It optimizes a future trajectory while adhering to state and action feasibility, dynamics, and safety constraints. Traditionally, gradient-based optimization techniques [1], [2] have been widely employed due to their flexibility and robustness. Despite their success, these methods often rely on differentiable objectives, dynamics, and constraints, limiting their applicability to non-differentiable problems. Moreover, their computational complexity often necessitates simplified or linearized dynamics, potentially compromising the accuracy of the solution.

In contrast, sampling-based methods do not require the control problem to be differentiable and can effectively accommodate nonlinear dynamics. They also tend to explore the solution space more broadly, which helps to avoid poor local optima. Nonetheless, the sampling-based methods face considerable challenges, such as low sample efficiency and inherent noise, which can result in jittery actions. In this work, our primary focus is mitigating such noisy actions while ensuring optimality and sample efficiency.

The proposed method, *time-correlated model predictive path integral* (TC-MPPI), is inspired by stochastic optimal control with dynamic extension, where action derivatives are treated as extended state variables to ensure action smoothness. However, dynamic extension also increases the number of state variables, reducing sample efficiency. To address this, we reformulate the control problem by incorporating the temporal correlation of actions, as illustrated in Fig. 1. This reformulation adjusts the sampling mean and covariance by introducing quadratic costs on extended variables and a conditional distribution to maintain causality in time-correlated sampling. With this adjustment, the sampling distribution generates smooth action sequences without requiring post-processing or additional state variables. Notably, TC-MPPI integrates action smoothing within the stochastic optimal control rather than merely penalizing action derivatives.

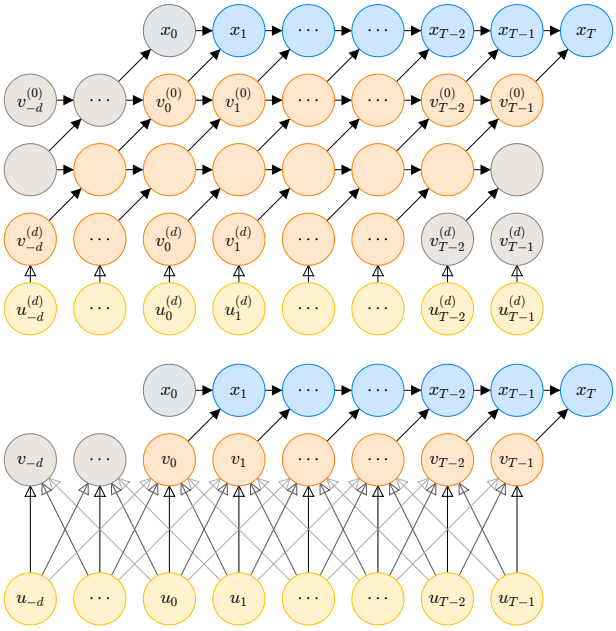

Fig. 1. Markov chains of the (top) extended and (bottom) time-correlated system. Blue nodes represent original system states, orange nodes denote original system actions and their derivatives, and yellow nodes indicate the mean of action nodes. Gray nodes represent the initial state, previous actions, and extended states that do not influence the future states.

Model predictive path integral (MPPI) [3], [4] is a leading approach for sampling-based real-time trajectory generation. Multiple MPPI variants have been developed to enhance performance. For example, linearized dynamics and control laws improve robustness and sample efficiency [5]–[7]. In safety-critical applications, conditional value-at-risk [8], [9] and control barrier functions [10] have been incorporated. To improve sampling distributions, techniques such as conditional variational autoencoders [11], normal log-normal distributions [12], adaptive importance sampling [13], and Stein variational gradient descent [14] have been employed. Yet, these advancements have not fully addressed the action noise. A notable exception is smooth-MPPI (S-MPPI) [15],

This work is supported by the Technology Innovation Program (20024355 and 1415187329, Development of autonomous driving connectivity technology based on sensor-infrastructure cooperation) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea). Corresponding author: Dongjun Lee.

The authors are with the Department of Mechanical Engineering, Institute of Advanced Machines and Design (IAMD) and Institute of Engineering Research (IOER), Seoul National University, Seoul 08826, South Korea (e-mail: minhyeong@snu.ac.kr; djlee@snu.ac.kr).

which introduces a lifting strategy to separate the action from the sampling space, producing smooth action sequences. While effective, the performance of S-MPPI can degrade with fewer samples or longer prediction horizons due to the increased complexity. In contrast, our work achieves action smoothing without relying on post-processing or additional state variables, ensuring smooth action sequences while preserving both optimality and sample efficiency.

The rest of the paper is organized as follows: Section II introduces key components of the stochastic optimal control problem and the MPPI algorithm; Section III details the proposed TC-MPPI algorithm; Section IV demonstrates the algorithm across various robotic tasks; and Section V concludes the paper.

II. Preliminary

This section outlines the theoretical background of model predictive path integral (MPPI) and offers intuitive insights into the MPPI algorithm to enhance understanding. To explore the details, let us consider a discrete-time system:

$$x_{t+1} = f_t(x_t, v_t) \tag{1}$$

where $x_t \in \mathcal{X}_t$ is the state, $v_t \in \mathcal{U}_t \subseteq \mathbb{R}^{n_u}$ is the action, and $f_t : \mathcal{X}_t \times \mathcal{U}_t \to \mathcal{X}_{t+1}$ is the transition model at time step $t \in \mathbb{Z}$. The system exhibits stochastic behavior, with the action being normally distributed as

$$V \sim \mathcal{Q}_{U,\Sigma} := \mathcal{N}(U, \Sigma) \tag{2}$$

where $V := v_{0:T-1}$ is the action sequence, $U := u_{0:T-1}$ is the mean input sequence, $u_t \in \mathcal{U}_t$ is the mean input, $\Sigma := \Sigma_{0:T-1}$ is the block diagonal covariance matrix, $\Sigma_t \in \mathbf{S}^{n_u}_{++}$ is the positive definite covariance matrix, and $T \in \mathbb{Z}_{++}$ is the horizon length. For simplicity, a sequence of vectors $\star_t$ is denoted as $\star_{a:b} := (\star_a, \star_{a+1}, \dots, \star_b)$, and a block diagonal matrix of matrices $\star_t$ is also denoted as $\star_{a:b} := \mathrm{diag}(\star_a, \star_{a+1}, \dots, \star_b)$. To maintain consistency with subsequent notations, we denote the distribution of V as $\mathcal{Q}_{U,\Sigma}$ and its probability density function as $q(V \mid U, \Sigma)$.

Now, an optimal control problem is formulated as

$$U^* = \arg\min_U \mathbb{E}_{V \sim \mathcal{Q}_{U,\Sigma}} \left[c_T(x_T) + \sum_{t=0}^{T-1} \ell_t(x_t, u_t) \right] \tag{3}$$

where $\ell_t : \mathcal{X}_t \times \mathcal{U}_t \to \mathbb{R}$ is the running cost function, and $c_T : \mathcal{X}_T \to \mathbb{R}$ is the terminal state-dependent cost function. Assuming that the running cost can be decoupled into a state-dependent cost $c_t : \mathcal{X}_t \to \mathbb{R}$ and a quadratic action-dependent cost, the control objective can be rewritten as

$$c_T(x_T) + \sum_{t=0}^{T-1} \ell_t(x_t, u_t) = S(V) + \frac{\lambda}{2} \|U - U_{\mathrm{ref}}\|^2_{\Sigma^{-1}} \tag{4}$$

where $U_{\mathrm{ref}} := u_{\mathrm{ref},0:T-1}$ is the reference action sequence, $u_{\mathrm{ref},t} \in \mathcal{U}_t$ is the reference action, $\lambda \in \mathbb{R}_{++}$ is the temperature, and $S(V) := c_T(x_T) + \sum_{t=0}^{T-1} c_t(x_t)$ is the state-dependent cost associated with the state trajectory $X := x_{0:T}$ following the transition model and the action sequence V.

As noted in [4], the optimal distribution $\mathcal{Q}^*$ that provides a lower bound for the control objective (4) is observed as

$$q^*(V) \propto \exp\left(-\frac{1}{\lambda} S(V) \right) p(V) \tag{5}$$

where $p(V) := q(V \mid U_{\mathrm{ref}}, \Sigma)$ is the base probability density function. It implies that the optimal control problem (3) can be solved by minimizing the Kullback-Leibler (KL) divergence between the controlled distribution $\mathcal{Q}_{U,\Sigma}$ and the optimal distribution $\mathcal{Q}^*$ so that

$$U^* = \arg\min_U D_{\mathrm{KL}} \left(\mathcal{Q}^* \,\|\, \mathcal{Q}_{U,\Sigma} \right) \tag{6}$$

which leads to a sampling-based optimization:

$$U^* = \mathbb{E}_{V \sim \mathcal{Q}^*}[V] = \mathbb{E}_{V \sim \hat{\mathcal{Q}}}[w(V) V] \tag{7}$$

where $\hat{\mathcal{Q}}$ is the sampling distribution, $\hat{q}(V) := q(V \mid \hat{U}, \Sigma)$ is the sampling probability density function, and $w(V) := \frac{q^*(V)}{\hat{q}(V)}$ is the importance sampling weight. Importance sampling is employed because directly sampling V from the optimal distribution $\mathcal{Q}^*$ is typically intractable. Notably, this sampling-based optimization does not impose restrictions on the structure of the sampling mean $\hat{U}$ and covariance Σ.

The optimal distribution $\mathcal{Q}^*$ derived from MPPI can also be characterized using Bayesian inference:

$$q^*(V) := p(V \mid o_\tau) = \frac{p(o_\tau \mid V)\, p(V)}{p(o_\tau)} \tag{8}$$

where $o_\tau \in \{0, 1\}$ is the optimality indicator of the trajectory $\tau := (X, V)$. In the context of the optimal distribution (5), the base probability serves as the prior probability $p(V)$, while the negative exponential of the state-dependent cost $S(V)$ represents the likelihood $p(o_\tau \mid V)$.

III. Time-Correlated MPPI

A notable challenge in sampling-based optimal control is managing its noisy actions. While filtering can reduce the noise, weak filters might be ineffective, and strong filters can compromise performance or optimality. Adjusting the sampling distribution offers an alternative, but incorrect choices can lead to optimization failures. To address this, we incorporate temporal correlation of actions into stochastic optimal control, ensuring smooth action sequences and sample efficiency. The following subsections detail this approach.

A. Optimal control with dynamic extension

Before introducing the time-correlated approach, we first define an optimal control with dynamic extension to obtain smooth action sequences. In this context, an extended system is defined as

$$x_{t+1} = f_t(x_t, v_t^{(0)}) \quad \text{and} \quad v_{t+1}^{(i)} = v_t^{(i)} + h_t v_t^{(i+1)} \tag{9}$$

where $v_t^{(i)} \in \mathbb{R}^{n_u}$ is the i-th derivative of the action, and h_t is the step size. Given a dynamic extension depth $d \in \mathbb{Z}_+$, the new action $v_t^{(d)}$ follows a normal distribution:

$$v^{(d)}_{-d:T-1} \sim \mathcal{Q}_{\mathrm{de}} := \mathcal{N}(u^{(d)}_{-d:T-1}, \mathrm{inv}(R^{(d)}_{-d:T-1})) \tag{10}$$

where $u_t^{(d)} \in \mathbb{R}^{n_u}$ is the mean input, and $R_t^{(d)} \in \mathbf{S}_{++}^{n_u}$ is the inverse covariance matrix. It is important to note that the prediction horizon begins from $-d$ rather than zero, since $v_{-d}^{(d)}$ influences the initial action $v_0^{(0)}$, as shown in Fig. 1.

The control objective for the extended system is to find the mean input $u_{-d:T-1}^{(d)}$ that minimizes the expected value:

$$\mathbb{E}_{\substack{v_{-d:T-1}^{(d)} \\ \sim \mathcal{Q}_{\mathrm{de}}}} \left[S_{\mathrm{de}}(v_{-d:T-1}^{(d)}) + \frac{\lambda}{2} \sum_{t=-d}^{T-1} \| u_t^{(d)} - u_{\mathrm{ref},t}^{(d)} \|_{R_t^{(d)}}^2 \right]$$

where $S_{\mathrm{de}} : \mathbb{R}^{n_u(d+T)} \to \mathbb{R}$ is the extended state-dependent cost, and $u_{\mathrm{ref},t}^{(d)} \in \mathbb{R}^{n_u}$ is the reference for the d-th action derivative, both of which will be detailed below. The optimal probability density is then given by

$$\begin{aligned} q_{\mathrm{de}}^*(v_{-d:T-1}^{(d)}) &\propto \exp\left(-\frac{1}{\lambda} S_{\mathrm{de}}(v_{-d:T-1}^{(d)}) \right) \\ &\times \exp\left(-\frac{1}{2} \sum_{t=-d}^{T-1} \| v_t^{(d)} - u_{\mathrm{ref},t}^{(d)} \|_{R_t^{(d)}}^2 \right) \end{aligned} \tag{11}$$

where the second exponential term corresponds to the Gaussian prior distribution of $v_{-d:T-1}^{(d)}$.

B. Time-correlated optimal control

While the optimal control with the extended system in Section III-A can yield smooth action sequences, solving it using sampling-based methods often demands more samples or iterations due to the increased number of state variables. To address this, we aim to revert the optimal distribution (11) back into a distribution of the original action V. For the transcription, we design the cost of extended states $v_t^{(i)}$ to be quadratic, resulting in an extended state-dependent cost:

$$\begin{aligned} &S_{\mathrm{de}}(v_{-d:T-1}^{(d)}) \\ &\quad := S(v_{-d:T-1}^{(0)}) + \frac{\lambda}{2} \sum_{i=0}^{d-1} \sum_{t=-d}^{T-1} \| v_t^{(i)} - u_{\mathrm{ref},t}^{(i)} \|_{R_t^{(i)}}^2 \end{aligned} \tag{12}$$

where $u_{\mathrm{ref},t}^{(i)}$ is the reference, and $R_t^{(i)} \in \mathbf{S}_{+}^{n_u}$ is the positive semi-definite gain matrix for $v_t^{(i)}$ with $i \in \{0, 1, \dots, d-1\}$. Combining (11) and (12), we obtain a quadratic function of $v_t^{(i)}$, with depths $0 \le i < d$ from the extended state-dependent cost and $i = d$ from the prior distribution.

To leverage the fact that action derivatives $v_t^{(i)}$ can be represented as finite differences of the action v_t, we define differentiation operators:

$$\begin{aligned} D_{a:b}^{(1)} &:= \mathrm{diag}(h_a^{-1}, h_{a+1}^{-1}, \dots, h_{b-1}^{-1}) \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & \ddots & \\ & & & -1 \ \ 1 \end{bmatrix}, \\ D_{a:b}^{(c)} &:= D_{a:b-c+1}^{(1)} D_{a:b}^{(c-1)}, \quad \text{and} \quad D_{a:b}^{(0)} := I_{b-a+1} \end{aligned} \tag{13}$$

which maps a scalar signal of length $b-a+1$ to its c-th derivative signal of length $b-a-c+1$. Action derivatives are then represented as $v_{-d:T-1-i}^{(i)} = \mathbf{D}_{-d:T-1}^{(i)} v_{-d:T-1}$ with $\mathbf{D}_{-d:T-1}^{(i)} := D_{-d:T-1}^{(i)} \otimes I_{n_u}$ where $I_a \in \mathbb{R}^{a \times a}$ is the identity matrix, and $\otimes$ is the Kronecker product operator.

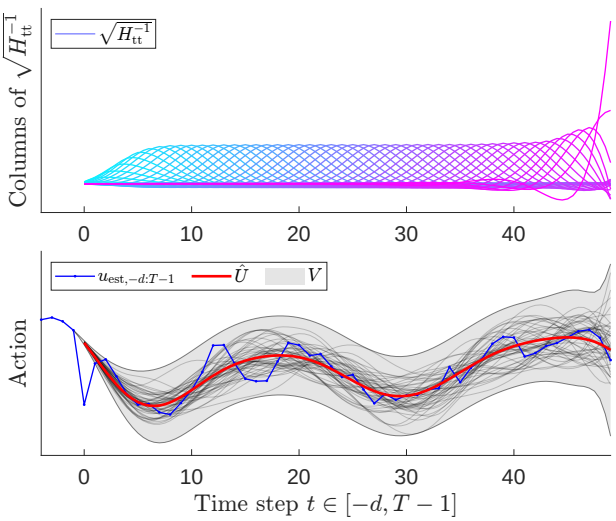

Fig. 2. Example of the sampling distribution. Top: Columns of the covariance square root matrix, forming the action sequence basis. Bottom: Mean (red) and sampled (gray) action sequences. An artificial noise is added to the estimated optimal sequence (blue) to show the smoothing effect of the gradient operator.

Using the differentiation operator (13), quadratic functions of $v_t^{(i)}$, expressed as a sum of squares, can be written as

$$\sum_{i=0}^{d} \sum_{t=-d}^{T-1-i} \frac{\| v_t^{(i)} - u_{\star,t}^{(i)} \|_{R_t^{(i)}}^2}{2} = \frac{\| v_{-d:T-1} + H^{-1} g_\star \|_H^2}{2} + C \tag{14}$$

$$H := + \sum_{i=0}^{d} (\mathbf{D}_{-d:T-1}^{(i)})^{\mathrm{T}} R_{-d:T-1-i}^{(i)} \mathbf{D}_{-d:T-1}^{(i)} \tag{15}$$

$$g_\star := - \sum_{i=0}^{d} (\mathbf{D}_{-d:T-1}^{(i)})^{\mathrm{T}} R_{-d:T-1-i}^{(i)} u_{\star,-d:T-1-i}^{(i)} \tag{16}$$

where $H \in \mathbb{R}^{n_u(T+d) \times n_u(T+d)}$ is the Hessian, $g_\star \in \mathbb{R}^{n_u(T+d)}$ is the gradient, and $C \in \mathbb{R}$ is the scalar term. The index $\star \in \{\mathrm{ref}, \mathrm{est}\}$ denotes either the prior or sampling distribution, which will be elaborated on later. Although some terminal actions $v_{T-i:T-1}^{(i)}$ are missing, these omissions do not affect the original state and action sequences, as shown in Fig. 1. Consequently, arbitrary values can be assigned, such as $v_{T-i:T-1}^{(i)} = u_{\star,T-i:T-1}^{(i)}$.

It is worth noting that the reference $u_{\mathrm{ref},t}^{(i)}$ and the estimated optimal $u_{\mathrm{est},t}^{(i)}$ of extended variables do not need to adhere the extended dynamics (9). These sequences can be arbitrary, but typically $u_{\star,t}^{(i)}$ is chosen as the i-th derivative of $u_{\star,t}$ or zero. This choice results in gradient operators:

$$G_\star := - \sum_{i=\mathcal{I}_\star} (\mathbf{D}_{-d:T-1}^{(i)})^{\mathrm{T}} R_{-d:T-1-i}^{(i)} \mathbf{D}_{-d:T-1}^{(i)} \tag{17}$$

where $\mathcal{I}_\star$ is the index set. Then, the gradient can be obtained as $g_\star = G_\star u_{\star,-d:T-1}$. We select $\mathcal{I}_{\mathrm{ref}} = \{0, 1, \dots, d-1\}$ for the prior distribution and $\mathcal{I}_{\mathrm{est}} = \{0\}$ for the sampling distribution, though other choices are also possible.

By employing the Hessian (15) and gradient (16), we can rewrite the extended optimal distribution (11) with the ex-

tended state cost (12) in terms of the original action sequence $v_{-d:T-1}$. However, the transcription causes a causality issue because the action sequence includes previous actions $v_{-d:-1}$, which are beyond our control at the current time step $t = 0$. To address this, we introduce a conditional distribution given the previous action sequence $u_{-d:-1}$. Consider normal distributions $v_{-d:T-1} \sim \mathcal{N}(-H^{-1}g_\star, H^{-1})$ which corresponds to quadratic functions (14). Their conditional distributions are expressed as $V \sim \mathcal{N}(\bar{U}, H_{\mathrm{tt}}^{-1})$ and $V \sim \mathcal{N}(\hat{U}, H_{\mathrm{tt}}^{-1})$ where

$$\bar{U} := -H_{\mathrm{tt}}^{-1} H_{\mathrm{ht}}^{\mathrm{T}} u_{-d:-1} - H_{\mathrm{tt}}^{-1} G_{\mathrm{ref,t}} u_{\mathrm{ref},-d:T-1} \tag{18}$$

$$\hat{U} := -H_{\mathrm{tt}}^{-1} H_{\mathrm{ht}}^{\mathrm{T}} u_{-d:-1} - H_{\mathrm{tt}}^{-1} G_{\mathrm{est,t}} u_{\mathrm{est},-d:T-1} \tag{19}$$

are the mean of prior and sampling distributions. For the sampling distribution, the estimated optimal action $u_{\mathrm{est},t}$ is obtained from the previous optimization result. The subscripts $\star_{\mathrm{h}}$ and $\star_{\mathrm{t}}$ denotes the head (i.e., $\star_{-d:-1}$) and tail (i.e., $\star_{0:T-1}$) components. Accordingly, we have

$$H = \begin{bmatrix} H_{\mathrm{hh}} & H_{\mathrm{ht}} \\ H_{\mathrm{th}} & H_{\mathrm{tt}} \end{bmatrix} \quad \text{and} \quad G_\star = \begin{bmatrix} G_{\star,\mathrm{h}} \\ G_{\star,\mathrm{t}} \end{bmatrix} \tag{20}$$

where $H_{\mathrm{hh}} \in \mathbb{R}^{n_u d \times n_u d}$, $H_{\mathrm{tt}} \in \mathbb{R}^{n_u T \times n_u T}$, $H_{\mathrm{ht}} = H_{\mathrm{th}}^{\mathrm{T}} \in \mathbb{R}^{n_u d \times n_u T}$ are the inverse covariance components, and $G_{\star,\mathrm{h}} \in \mathbb{R}^{n_u d \times n_u (d+T)}$, $G_{\star,\mathrm{t}} \in \mathbb{R}^{n_u T \times n_u (d+T)}$ are the gradient operator components. Since $V \sim \mathcal{N}(\hat{U}, H_{\mathrm{tt}}^{-1})$ is equivalent to $V = \hat{U} + H_{\mathrm{tt}}^{-1/2}\mathcal{E}$ with $\mathcal{E} \sim \mathcal{N}(0, I)$, the columns of the covariance square root matrix serves as a basis for the random action sequence, capturing the temporal correlation. Fig. 2 illustrates an example of the basis and sampled actions.

Substituting the extended state-dependent cost (12) and conditional distributions (18) and (19) to the optimal distribution (11), we obtain a time-correlated optimal control problem and its optimal probability density:

$$q_{\mathrm{tc}}^*(V) \propto \exp\left(-\frac{1}{\lambda} S(V)\right) \exp\left(-\frac{1}{2}\|V - \bar{U}\|_{H_{\mathrm{tt}}}^2\right) \tag{21}$$

where $\bar{U} \in \mathbb{R}^{n_u T}$ and $H_{\mathrm{tt}} \in \mathbb{R}^{n_u T \times n_u T}$ are the mean and inverse covariance of the prior distribution, and $\mathcal{Q}_{\mathrm{tc}}^*$ is the distribution of the optimal probability $q_{\mathrm{tc}}^*(V)$. In the context of Bayesian inference (8), this reformulation transfers the cost associated with extended variables to the prior distribution, thereby improving sample efficiency. The optimal U, which aligns $\mathcal{N}(U, H_{\mathrm{tt}}^{-1})$ with the optimal distribution $\mathcal{Q}_{\mathrm{tc}}^*$, can be obtained by using a sampling-based optimization:

$$U^* = \mathbb{E}_{V \sim \mathcal{Q}_{\mathrm{tc}}^*}[V] = \mathbb{E}_{V \sim \hat{\mathcal{Q}}_{\mathrm{tc}}}[w_{\mathrm{tc}}(V) V] \tag{22}$$

where $\hat{\mathcal{Q}}_{\mathrm{tc}} := \mathcal{N}(\hat{U}, H_{\mathrm{tt}}^{-1})$ is the sampling distribution, $\hat{q}_{\mathrm{tc}}(V)$ is the corresponding probability density function, and

$$w_{\mathrm{tc}}(V) := \frac{q_{\mathrm{tc}}^*(V)}{\hat{q}_{\mathrm{tc}}(V)} \propto \exp\left(-\frac{1}{\lambda} S(V) + (\bar{U} - \hat{U})^{\mathrm{T}} H_{\mathrm{tt}} V\right) \tag{23}$$

is the importance sampling weight.

The proposed method, referred to as time-correlated model predictive path integral (TC-MPPI), is summarized in Algorithm 1. It is important to note that once the time steps and action cost parameters are specified, the Hessian and gradient operators can be computed offline, reducing online computational efforts. Furthermore, advanced techniques commonly used in MPPI-like algorithms (e.g., incorporating exploration samples or decoupling the action cost from the temperature [4]), though not addressed in Algorithm 1, can also be seamlessly integrated into our framework.

Algorithm 1 Time-Correlated MPPI

Require: f_t transition model, K number of samples, T horizon length, d correlation depth, h_t time step size, $R_t^{(i)}$ action cost parameters, λ temperature parameter

Compute $H_{\mathrm{tt}}, H_{\mathrm{ht}}, G_{\mathrm{ref,t}}, G_{\mathrm{nom,t}}$ ▷ (15), (17) *and* (20)
Initialize $\bar{U}$, $\hat{U}$
while task not done **do**
  $x_{\mathrm{init}} \leftarrow$ `getCurrentState()` ▷ *Estimate state*
  for $k = 0$ **to** $K - 1$ **do**
    $x_0 \leftarrow x_{\mathrm{init}}$
    $V_k \sim \mathcal{N}(\hat{U}, H_{\mathrm{tt}}^{-1})$
    $S_k \leftarrow 0$
    for $t = 0$ **to** $T - 1$ **do**
      $x_{t+1} \leftarrow f_t(x_t, v_t)$
      $S_k \leftarrow S_k + c_t(x_t)$
    end for
    $S_k \leftarrow S_k + c_T(x_T)$
    $S'_k \leftarrow S_k + \lambda(\hat{U} - \bar{U})^{\mathrm{T}} H_{\mathrm{tt}} V_k$
  end for
  $w_{0:K-1} \leftarrow$ `getWeights`$(S'_{0:K-1})$ ▷ (23)
  $U^* \leftarrow \sum_{k=0}^{K-1} V_k w_k$ ▷ (22)
  `setCurrentAction`(u_0^*) ▷ *Apply action*
  $u_{\mathrm{ref},-d:-1} \leftarrow (u_{\mathrm{ref},-d+1:-1}, u_{\mathrm{ref},0})$
  $u_{\mathrm{est},-d:-1} \leftarrow (u_{\mathrm{est},-d+1:-1}, u_0^*)$
  $U_{\mathrm{ref}} \leftarrow$ `getNextReferenceAction()`
  $U_{\mathrm{est}} \leftarrow$ `getNextOptimalActionEstimate`(U^*, h_{-1})
  Update $\bar{U}$, $\hat{U}$ ▷ (18) *and* (19)
end while

IV. Result

We evaluate the proposed method across various control tasks to demonstrate its ability to generate smooth actions while maintaining high performance. TC-MPPI is compared with MPPI variants: the original MPPI [3], MPPI with a Savitzky-Golay filter (SGF) [4], and smooth-MPPI (S-MPPI) [15]. MPPI with dynamic extension (DE), presented in Section III-A, is also tested to validate the influence of extended dynamics. Refer to the simulation videos available at https://youtu.be/nWfJ2MAV2JI. All algorithms are implemented in MATLAB and tested on a Windows 11 machine with AMD Ryzen 5 3600X CPU and 16 GB RAM. The control frequency is set to 100 Hz, with a prediction step size of 0.05 s. Our method adapts to the difference between control frequency and prediction step size using $h_t = 0.01\,\mathrm{s}$ for $-d \le t < 0$ and $h_t = 0.05\,\mathrm{s}$ for $0 \le t < T$. Advanced techniques such as exploration samples and temperature decoupling are not employed for simplicity.

To ensure fairness, the state-dependent cost, temperature, horizon length, and sample sizes are kept the same across

TABLE I

SUCCESS RATE AND TERMINAL ERROR OF SWING-UP TASKS

Method	Pendulum swing-up		Cart-pole swing-up	
	Success	Error (deg)	Success	Error (deg)
MPPI	50/50	0.46 ± 0.32	50/50	0.36 ± 0.28
MPPI w/ SGF	50/50	0.33 ± 0.25	50/50	0.27 ± 0.21
S-MPPI	30/50	5.62 ± 3.96	50/50	0.22 ± 0.18
MPPI w/ DE	20/50	6.33 ± 4.47	0/50	73.88 ± 52.62
TC-MPPI	50/50	0.31 ± 0.22	50/50	0.18 ± 0.13

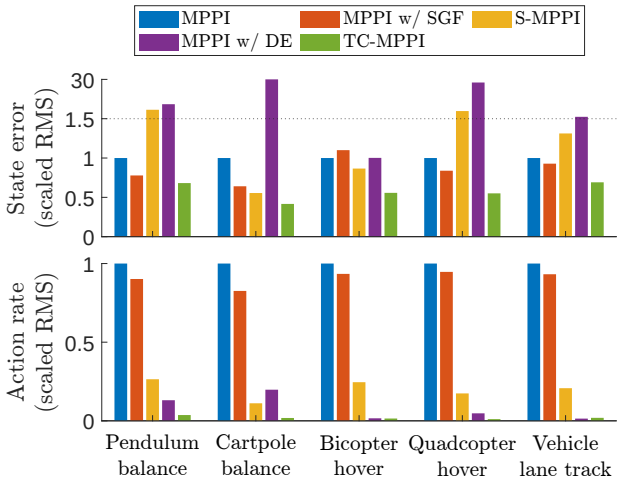

Fig. 3. State errors and action change rates for stabilizing tasks. High action rates indicate high action noise. To account for varying scales across tasks, the values are normalized so that MPPI values are set to 1. Scaled errors exceeding 30 are not shown.

all algorithms. Specifically, we use horizon lengths of 40, 20, 20, 20, and 40 and sample sizes of 50, 200, 100, 100, and 100 for the pendulum, cart-pole, 2D bicopter, 3D quadcopter, and autonomous vehicle, respectively. To isolate the effect of filters, MPPI and MPPI with SGF use the same action covariance. Action derivative costs are set to $R_t^{(1)}, R_t^{(2)}, R_t^{(3)} = 0$ and $R_t^{(4)} = \varepsilon R_t^{(0)}$, where ε is tuned within $[10^{-7}, 10^{-3}]$ for MPPI with DE and $[10^{-12}, 10^{-8}]$ for TC-MPPI. Notably, increasing action derivative costs results in smoother action samples but may restrict action exploration and responsiveness to unexpected disturbances.

Table I summarizes success rates and errors for the pendulum and cart-pole swing-up tasks. Success is defined as achieving a root-mean-squared error (RMSE) within a specified threshold during the final 1 s of scenarios that last between 10 s to 20 s. Fig. 3 shows RMS state errors and action change rates for stabilizing tasks. With small sample sizes, MPPI with SGF often outperforms MPPI without filters, as the filter can mitigate action variance. However, SGF can also degrade performance by violating the solution optimality. S-MPPI effectively reduces action noise and improves performance, but it may result in higher errors if the sample size is insufficient relative to problem complexity. MPPI with DE produces smooth actions yet sacrifices control performance, often leading to failures. In contrast, TC-MPPI remarkably reduces action noise and achieves lower errors, demonstrating its ability to smooth actions

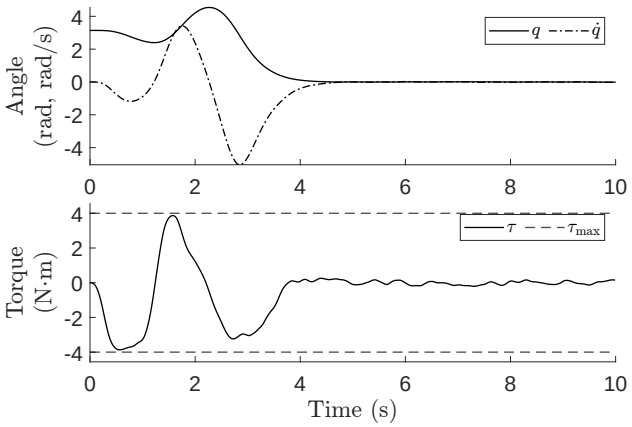

Fig. 4. State and action for the pendulum swing-up task using TC-MPPI. The resulting actions are smooth and effectively accomplish the task.

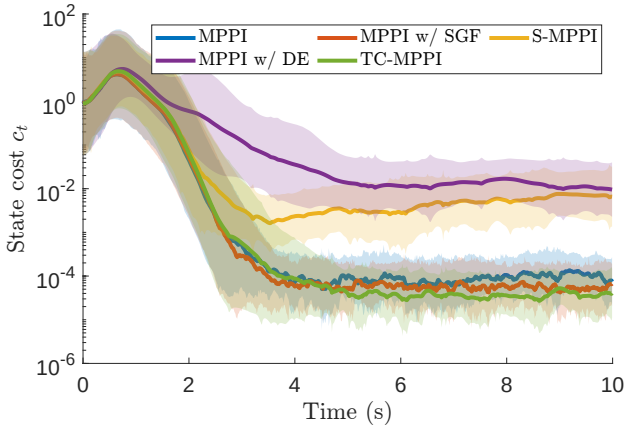

Fig. 5. Average state cost $c_t(x_t)$ for the pendulum swing-up task. TC-MPPI achieves action smoothing without compromising performance.

without compromising performance. While performance is not our primary objective, TC-MPPI yields better results by improving action variance and sample efficiency. In practice, action smoothness would also be beneficial with unmodeled high-frequency dynamics, though it is not considered in simulations. The following subsections detail each platform.

Pendulum: An actuated pendulum, or single-link manipulator, is a good test case for evaluating algorithms. The state includes the joint angle and angular rate $x_t := (q_t, \dot{q}_t) \in \mathbb{R}^2$, and the action is the joint torque $u_t := \tau_t \in \mathbb{R}$. The pendulum has a mass of 1 kg, a length of 1 m, and a maximum torque of 4 N m, which is insufficient to swing-up the pendulum in a single action. In the swing-up task, initial joint angles are randomly selected, and success is defined as achieving a joint angle RMSE below 5°. Fig. 4 illustrates an example of the state and action during swing-up, while Fig. 5 shows the state cost. TC-MPPI achieves a joint angle RMSE of 0.37° in the steady-state using a state cost $c_t := \|x_t\|^2_{\mathrm{diag}(1,0.1)}$. Refer to Table I and Fig. 3 for a comparison of control performance and action noise against other algorithms.

Cart-pole: A cart-pole, which is under-actuated and unstable, is another valuable test case. The state includes the cart position, pole angle, cart velocity, and pole angular rate

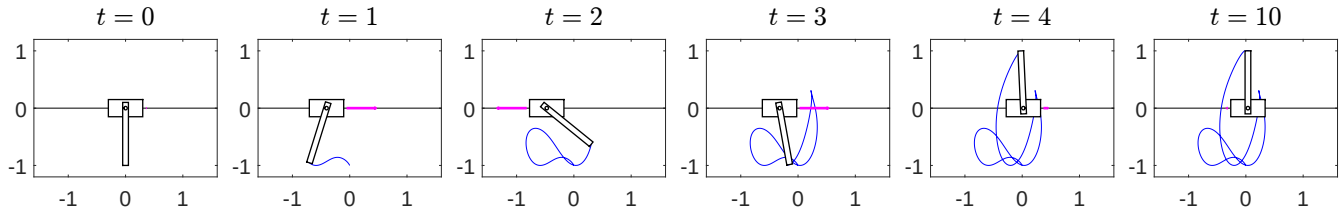

Fig. 6. Snapshots of the cart-pole swing-up task using TC-MPPI. The cart force (magenta) and the pole tip trajectory (blue) are shown. As the cart force is constrained, it adjusts the force direction to swing-up the pole, ultimately balancing at the origin.

$x_t := (p_t^x, \theta_t, \dot{p}_t^x, \dot{\theta}_t) \in \mathbb{R}^4$, and the action is the horizontal cart force $u_t := f_t \in \mathbb{R}$. The system parameters are: cart mass $0.5\,\mathrm{kg}$, pole mass $1\,\mathrm{kg}$, pole length $1\,\mathrm{m}$, maximum force $5\,\mathrm{N}$, and rail length $4\,\mathrm{m}$. In the swing-up task, initial cart positions and pole angles are randomly selected from uniform distributions. Success is defined as achieving a position RMSE within $10\,\mathrm{cm}$ and an angle RMSE within $5°$. Fig. 6 provides snapshots of the swing-up process. TC-MPPI achieves a cart position RMSE of $2.74\,\mathrm{cm}$ and pole angle RMSE of $0.21°$ in the steady-state using a state-dependent cost $c_t := \|x_t\|^2_{\mathrm{diag}(5,10,0.1,0.1)}$.

2D bicopter: The state of a 2D bicopter includes the body pose and spatial velocity $x_t := (p_t, \theta_t, \dot{p}_t, \dot{\theta}_t) \in \mathbb{R}^6$, with thrust forces $u_t := f_t \in \mathbb{R}^2$ as the action. The model parameters are based on [16]. For hovering and trajectory tracking, TC-MPPI achieves position RMSEs of $2.09\,\mathrm{cm}$ and $2.81\,\mathrm{cm}$ using a state cost $c_t := \|x_t - x_{\mathrm{ref},t}\|^2_{\mathrm{diag}(5,5,1,0.5,0.5,0.1)}$. The tracking reference is a vertical circular trajectory at $1\,\mathrm{m/s}$.

3D quadcopter: The state of a 3D quadcopter includes the body pose and spatial velocity $x_t := (p_t, R_t, v_t, \omega_t) \in \mathrm{SE}(3) \times \mathbb{R}^6$, and the action is the thrust forces $u_t := f_t \in \mathbb{R}^4$, based on the dynamics from [16]. The sampling covariance is set to a low value to account for the small yaw moment coefficient. For hovering and trajectory tracking tasks, TC-MPPI achieves position RMSEs of $8.24\,\mathrm{cm}$ and $9.37\,\mathrm{cm}$ where the tracking reference is a horizontal circular trajectory at $1\,\mathrm{m/s}$. The state cost is defined as $c_t := \|(e_{p,t}, \dot{e}_{p,t})\|^2 + \|(e_{R,t}, e_{\omega,t})\|^2_{\mathrm{diag}(0,0,10,0,0,10)}$ where $e_{p,t} := p_t - p_{\mathrm{ref},t}$ is the position error, $e_{R,t} := \log(R_{\mathrm{ref},t}^T R_t)^\vee$ is the rotation error, and $e_{\omega,t} := \omega_t - R_t^T R_{\mathrm{ref},t} \omega_{\mathrm{ref},t}$ is the angular rate error.

Autonomous vehicle: In an autonomous vehicle, the state is defined as the position, orientation, velocity, yaw rate, and front wheel speed $x_t := (p_t, \theta_t, \dot{p}_t, \dot{\theta}_t, w_{F,t}) \in \mathbb{R}^7$. The action consists of the front wheel angular acceleration and steering angle $u_t := (\dot{w}^y_{F,t}, \delta_{F,t}) \in \mathbb{R}^2$. The vehicle has a mass of $1100\,\mathrm{kg}$, a wheelbase of $2.35\,\mathrm{m}$, a maximum wheel acceleration of $30\,\mathrm{rad/s^2}$, and a maximum steering angle of $30°$. We use a dynamic bicycle model with tire forces modeled by simplified Pacejka magic formulas [17]:

$$f_j^x = f_j^z D_x \sin(C_x \arctan((1 - E_x) B_x \kappa_j + E_x \tan B_x \kappa_j))$$
$$f_j^y = f_j^z D_y \sin(C_y \arctan((1 - E_y) B_y \alpha_j + E_y \tan B_y \alpha_j))$$

where $(f_j^x, f_j^y, f_j^z) \in \mathbb{R}^3$ is the tire force, κ_j, α_j are the slip ratio and slip angles, $j \in \{F, R\}$ is the front and rear tire index, and $B_x = 10$, $C_x = 1.9$, $D_x = 1$, $E_x = 0.97$, $B_y = 10$, $C_y = 2.5$, $D_y = 1$, $E_y = 1$ are tire coefficients. For trajectory tracking, TC-MPPI achieves position RMSE of $9.69\,\mathrm{cm}$ using $c_t := \|x_t - x_{\mathrm{ref},t}\|^2_{\mathrm{diag}(1,1,1,1,1,5,0)}$. Circuit tracking is conducted using $c_t := (50000 + 50\mathcal{D}_{\mathrm{coll}}(x_t))\,\mathcal{I}_{\mathrm{coll}}(x_t) + \|v_{\mathcal{B},t} - v_{\mathcal{B},\mathrm{des},t}\|^2_{\mathrm{diag}(0.1,0.5)}$ where $\mathcal{D}_{\mathrm{coll}} : \mathbb{R}^7 \to \mathbb{R}_+$ is the collision distance function, $\mathcal{I}_{\mathrm{coll}} : \mathbb{R}^7 \to \{0, 1\}$ is the collision indicator, $v_{\mathcal{B},t} \in \mathbb{R}^2$ is the vehicle velocity in the body frame, and $v_{\mathcal{B},\mathrm{des},t} = (20, 0)\,\mathrm{m/s}$ is the desired body velocity. Trajectory and velocity of the circuit tracking task are visualized in Fig. 7.

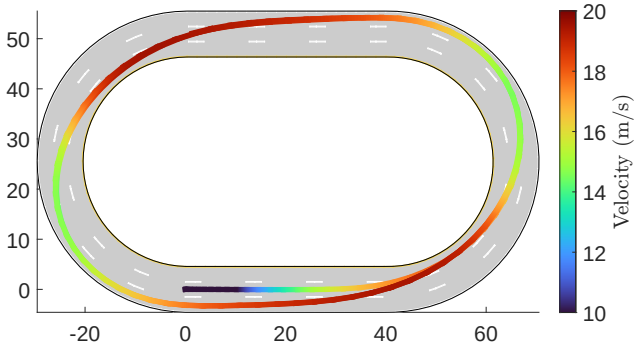

Fig. 7. Trajectory of autonomous vehicle circuit tracking using TC-MPPI. Vehicle speed is shown with a color map. The vehicle starts at the origin with zero velocity, accelerates to the desired velocity, and slows down at corners to prevent slip and collision.

V. Conclusion

In conclusion, we propose a sampling-based control algorithm that generates smooth action sequences. Our approach incorporates temporal correlation of actions in stochastic optimal control, which is equivalent to applying a quadratic cost on action derivatives. The sampling distribution is then derived as a conditional Gaussian distribution, enabling smooth action generation without requiring post-processing or additional state variables. We validate the proposed method through simulations on various platforms, including a pendulum, cart-pole, 2D bicopter, 3D quadcopter, and autonomous vehicle. The results demonstrate a significant reduction in action noise while improving overall performance. Future research directions include conducting hardware experiments, analyzing the effect of correlation parameters, quantifying robustness against unexpected disturbances and communication delays, integrating fast simulators [18], and extending our time correlation approach to other sampling-based methods [19], [20].

REFERENCES

[1] O. Von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, pp. 357–373, 1992.

[2] A. V. Rao, "Trajectory optimization: A survey," *Optimization and Optimal Control in Automotive Systems*, pp. 3–21, 2014.

[3] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic MPC for model-based reinforcement learning," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1714–1721, IEEE, 2017.

[4] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[5] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. M. Rehg, and E. A. Theodorou, "Robust sampling based model predictive control with sparse objective information," in *Robotics: Science and Systems*, vol. 14, 2018.

[6] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams, and E. A. Theodorou, "Robust model predictive path integral control: Analysis and performance guarantees," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1423–1430, 2021.

[7] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras, "Trajectory distribution control for model predictive path integral control using covariance steering," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1478–1484, IEEE, 2022.

[8] Z. Wang, O. So, K. Lee, and E. A. Theodorou, "Adaptive risk sensitive model predictive control with stochastic search," in *Proceedings of Learning for Dynamics and Control*, pp. 510–522, PMLR, 2021.

[9] J. Yin, Z. Zhang, and P. Tsiotras, "Risk-aware model predictive path integral control using conditional value-at-risk," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 7937–7943, IEEE, 2023.

[10] J. Yin, C. Dawson, C. Fan, and P. Tsiotras, "Shield model predictive path integral: A computationally efficient robust MPC method using control barrier functions," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7106–7113, 2023.

[11] R. Kusumoto, L. Palmieri, M. Spies, A. Csiszar, and K. O. Arras, "Informed information theoretic model predictive control," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2047–2053, IEEE, 2019.

[12] I. S. Mohamed, K. Yin, and L. Liu, "Autonomous navigation of AGVs in unknown cluttered environments: log-MPPI control strategy," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10240–10247, 2022.

[13] D. M. Asmar, R. Senanayake, S. Manuel, and M. J. Kochenderfer, "Model predictive optimized path integral strategies," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3182–3188, IEEE, 2023.

[14] K. Honda, N. Akai, K. Suzuki, M. Aoki, H. Hosogaya, H. Okuda, and T. Suzuki, "Stein variational guided model predictive path integral control: Proposal and experiments with fast maneuvering vehicles," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 7020–7026, IEEE, 2024.

[15] T. Kim, G. Park, K. Kwak, J. Bae, and W. Lee, "Smooth model predictive path integral control without smoothing," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10406–10413, 2022.

[16] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proceedings of IEEE Conference on Decision and Control*, pp. 5420–5425, IEEE, 2010.

[17] R. Rajamani, *Vehicle Dynamics and Control*. Springer Science & Business Media, 2nd ed., 2012.

[18] J. Lee, M. Lee, and D. Lee, "Large-dimensional multibody dynamics simulation using contact nodalization and diagonalization," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1419–1438, 2022.

[19] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, vol. 133. Springer, 2004.

[20] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *Proceedings of International Conference on Machine Learning*, pp. 281–288, 2012.