The Two-Stage PI^2 Control Strategy

Peter Varnai and Dimos V. Dimarogonas,[1] *Senior Member, IEEE*

***Abstract*— Policy improvement with path integrals (PI^2) is a stochastic optimal control method generally regarded as a reinforcement learning algorithm. Recent work, however, suggests that the reinforcement learning aspect of PI^2 actually appears when optimizing feedforward controls which will lead to optimal closed-loop performance once combined with feedback controls. These feedbacks are necessary to achieve the predicted performance, yet have been largely neglected in the literature and applications due to their complexity. In this work, we show that the feedbacks actually take a simple-to-implement form for a wide range of system dynamics, paving way for future research and applications of PI^2. The correctness of the results is demonstrated through numerical simulations.**

***Index Terms*— Stochastic optimal control, path integral policy improvement, Feynman-Kac theorem, nonlinear control systems.**

I. INTRODUCTION

POLICY improvement with path integrals (PI^2) is a reinforcement learning algorithm developed for solving stochastic optimal control problems [1]. The main idea is to linearize the stochastic Hamilton–Jacobi–Bellman (HJB) equations underlying the control problem to allow optimal feedbacks to be calculated from path integrals, *i.e.*, open-loop roll-outs and corresponding costs, of the dynamical system. PI^2 then emerges as an application of this path integral optimal control formalism to optimize control policies parameterized by so-called dynamic motion primitives. Over time, heuristic modifications have improved the algorithm [2], which is now commonly categorized as a general black-box policy search method alongside the likes of CEM and CMA-ES [3].

Our recent work [4], however, suggests that PI^2 is instead better interpreted as a two-stage control strategy. In the first stage, feedforward controls that yield optimal performance when augmented with closed-loop feedback are sought iteratively. The iterative updates resemble the previous black-box policy search interpretation of PI^2, including its heuristic improvements, and are rigorously shown to stem from natural gradient descent in [4]. In the second stage, the closed-loop feedback that is necessary to achieve the performance predicted by the first stage is then implemented during real-time operation. To the best of our knowledge, the form of this feedback has not changed much in subsequent research [5] since the derivations in [1], where they served to motivate the original, black-box type PI^2 algorithm updates. The complexity of the results therein inhibits their ease of implementation, leading to a limited number of practical applications [6], [7].

In this paper, we turn our attention back to the closed-loop feedback aspect of the PI^2 control strategy, simplifying and expanding upon the original results presented in [1]. Our main contributions can be summarized as follows:

- We extend the previous PI^2 theory to handle general process noises (not just in the direction of the input) by proposing a novel closed-loop feedback law. A previously necessary assumption relating the quadratic feedback regularization and noise covariance matrices, usually given as $\lambda \mathbf{R}^{-1} = \mathbf{\Sigma}$, is also avoided.
- We provide a simpler derivation of the optimal closed-loop controls based on the key idea that calculations involving forward or backward Euler approximations of continuous-time dynamics must yield the same final result as the discretization time step $\Delta t \to 0$. This avoids the need to introduce so-called 'generalized costs' as usually done in the literature [1], [5].

Although some steps in our derivations follow the line of thought in [1] and [4], the above changes lead to fundamentally different, but also much simpler expressions for the closed-loop control actions. Due to space limitations, in this paper we therefore only present the main steps and highlight the prominent differences within the derivation, and leave a more detailed technical presentation for [8]. Therein, we also discuss additional corollaries of this work, such as further simplifications stemming from the $\lambda \mathbf{R}^{-1} = \mathbf{\Sigma}$ assumption, and the immediate applicability of the theory to a wider range of system dynamics using the generalized matrix inverse.

Together with [4], this work offers a novel view of PI^2 as a simple-to-implement and computationally parallelizable two-stage control strategy. It is our hope that the presented theory will provide a firmer theoretical foundation for recent and future research regarding PI^2 [9]–[12] and renew interest on this topic, as it has already enjoyed success in many practical applications [13]–[15].

The rest of this paper is organized as follows. Section II provides necessary background and notation, and Section III outlines the PI^2 control strategy. The main steps for deriving the optimal feedback controls are then presented in Section IV. Finally, Section V demonstrates the correctness of the theoretical results numerically through a simulation study.

II. PRELIMINARIES

We review the basics of functional calculus following the theory from [16] extended to the multivariate case, as it is necessary in order to take path-dependent costs into account.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Swedish Research Council (VR), and the SSF COIN project.

[1]Both authors are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Brinellvägen 8, 114 28 Stockholm, Sweden (emails: varnai@kth.se, dimos@kth.se).

Let $T > 0$ denote the time horizon of an optimal control problem, and for any $t \in [0, T]$, let Λ_t denote the set of all RCLL (right continuous, left limit) functions mapping each point $s \in [0, t]$ to $\mathbb{R}^p$. A trajectory (path) of length t is then denoted by $\tau_t \in \Lambda_t$, and its value at time s is denoted by $\boldsymbol{\tau}_t(s) \in \mathbb{R}^p$. The set of all possible paths for all possible time intervals is given by $\Lambda := \bigcup_{t\in[0,T]} \Lambda_t$.

A *functional* $V : \Lambda \to \mathbb{R}$ associates a real number to paths in the set Λ. Its space and time derivatives are defined and operate similarly as those of traditional functions. In particular, for a path $\tau_t \in \Lambda_t$, spacial shift $\boldsymbol{h} \in \mathbb{R}^p$, and temporal shift $\delta t > 0$, let us introduce

$$\boldsymbol{\tau}_t^{\boldsymbol{h}}(s) = \begin{cases} \boldsymbol{\tau}_t(s), & s < t \\ \boldsymbol{\tau}_t(t) + \boldsymbol{h}, & s = t \end{cases} \quad \text{and} \quad \boldsymbol{\tau}_{t,\delta t}(s) := \begin{cases} \boldsymbol{\tau}_t(s), & s \leq t \\ \boldsymbol{\tau}_t(t), & s \in (t, t+\delta t] \end{cases}$$

The (directional) space and time derivatives of the functional V can now be defined as

$$\Delta_{\boldsymbol{x}}^{\boldsymbol{h}} V(\tau_t) = \lim_{\epsilon \to 0} \tfrac{1}{\epsilon} \left[V(\tau_t^{\epsilon \cdot \boldsymbol{h}}) - V(\tau_t) \right] \tag{1a}$$

$$\Delta_t V(\tau_t) = \lim_{\delta t \to 0^+} \tfrac{1}{\delta t} \left[V(\tau_{t,\delta t}) - V(\tau_t) \right]. \tag{1b}$$

By choosing the direction $\boldsymbol{h}$ as the unit basis vectors $\boldsymbol{e}_i$ corresponding to the (i)-th dimensions of the space $\mathbb{R}^p$, the gradient of the functional can be formed as the vector

$$\boldsymbol{\Delta}_{\boldsymbol{x}} V(\tau_t) := \begin{bmatrix} \Delta_{\boldsymbol{x}}^{\boldsymbol{e}_1} V(\tau_t) & \dots & \Delta_{\boldsymbol{x}}^{\boldsymbol{e}_p} V(\tau_t) \end{bmatrix}^{\mathrm{T}}. \tag{1c}$$

The Hessian is then formed as the matrix

$$\boldsymbol{\Delta}_{\boldsymbol{x}\boldsymbol{x}} V(\tau_t) := \begin{bmatrix} \boldsymbol{\Delta}_{\boldsymbol{x}} \left(\Delta_{\boldsymbol{x}}^{\boldsymbol{e}_1} V(\tau_t) \right) & \dots & \boldsymbol{\Delta}_{\boldsymbol{x}} \left(\Delta_{\boldsymbol{x}}^{\boldsymbol{e}_p} V(\tau_t) \right) \end{bmatrix}. \tag{1d}$$

Finally, for a matrix $\mathbf{M} \in \mathbb{R}^{p \times p}$ we define the Lie derivative induced by the flow $\dot{\boldsymbol{x}} = \mathbf{M}\boldsymbol{x}$ as

$$\mathcal{L}_{\mathbf{M}} V(\tau_t) := \mathbf{M}^{\mathrm{T}} \boldsymbol{\Delta}_{\boldsymbol{x}} V(\tau_t), \tag{2}$$

which satisfies $\boldsymbol{x}^{\mathrm{T}} \mathcal{L}_{\mathbf{M}} V(\tau_t) = \Delta_{\boldsymbol{x}}^{\boldsymbol{h}} V(\tau_t)$ for $\boldsymbol{h} = \dot{\boldsymbol{x}} = \mathbf{M}\boldsymbol{x}$.

III. The two-stage PI^2 control strategy

Consider a control system with state dynamics

$$\dot{\boldsymbol{x}}_t = \boldsymbol{f}(\boldsymbol{x}_t, t) + \mathbf{g}(\boldsymbol{x}_t)\boldsymbol{u}_t + \boldsymbol{\Sigma}_{x,t}^{1/2} \boldsymbol{\epsilon}_{x,t}, \tag{3}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ is the system state, $\boldsymbol{u}_t \in \mathbb{R}^m$ is the control input, $\boldsymbol{\epsilon}_{x,t} \in \mathbb{R}^n$ is zero-mean Gaussian white noise with covariance $\boldsymbol{\Sigma}_{x,t} \geq \mathbf{0}$, $\boldsymbol{f}(\cdot)$ describes the autonomous system dynamics, and $\mathbf{g}(\cdot)$ describes the influence of the input.

The control input $\boldsymbol{u}_t$ will be composed of a feedforward and a feedback term, the former of which is generated as

$$\dot{\boldsymbol{k}}_t = \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_{k,t}^{1/2} \boldsymbol{\epsilon}_{k,t}, \tag{4}$$

where $\boldsymbol{k}_t \in \mathbb{R}^m$ is the feedforward, $\boldsymbol{\mu}_t \in \mathbb{R}^m$ is its nominal derivative, and $\boldsymbol{\epsilon}_{k,t} \in \mathbb{R}^m$ is added zero-mean Gaussian white noise with covariance $\boldsymbol{\Sigma}_{k,t} \geq \mathbf{0}$. Together, $\boldsymbol{x}$ and $\boldsymbol{k}$ can be regarded as an abstracted system state $\boldsymbol{z}$ with dynamics

$$\dot{\boldsymbol{z}}_t \equiv \begin{bmatrix} \dot{\boldsymbol{x}}_t \\ \dot{\boldsymbol{k}}_t \end{bmatrix} := \boldsymbol{F}_t(\boldsymbol{x}_t) + \mathbf{G}(\boldsymbol{x}_t)\boldsymbol{v}_t + \boldsymbol{\Xi}_t^{1/2} \boldsymbol{w}_t, \tag{5}$$

where the introduced quantities are

$$\boldsymbol{F}_t(\boldsymbol{x}_t) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_t, t) \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{G}(\boldsymbol{x}_t) = \begin{bmatrix} \mathbf{g}(\boldsymbol{x}_t) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$
$$\boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{u}_t \\ \boldsymbol{\mu}_t \end{bmatrix}, \quad \boldsymbol{\Xi}_t = \begin{bmatrix} \boldsymbol{\Sigma}_{x,t} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{k,t} \end{bmatrix}, \quad \boldsymbol{w}_t = \begin{bmatrix} \boldsymbol{\epsilon}_{x,t} \\ \boldsymbol{\epsilon}_{k,t} \end{bmatrix}.$$

Note that $\boldsymbol{v}_t$ can be regarded as an abstracted control input to the system. Furthermore, as an extension to the works [1] and [4], the dynamics (5) now allow the noise $\boldsymbol{w}_t$ to affect both the states $\boldsymbol{x}$ and $\boldsymbol{k}$ with arbitrary covariance $\boldsymbol{\Xi}_t \geq \mathbf{0}$, and not just in the direction of the input as $\mathbf{G}(\boldsymbol{x}_t)\boldsymbol{\Xi}_t^{1/2}\boldsymbol{w}_t$.

Let τ_T denote the trajectory of the abstracted state $\boldsymbol{z}_t$ during a time horizon $T > 0$, *i.e.*, $\tau_T := \{\boldsymbol{z}_t \mid 0 \leq t \leq T\}$. A cost $S_t(\tau_T)$ is then assigned to each trajectory in the form

$$S_t(\tau_T) \equiv S(\tau_T, t) := \phi(\tau_T) + \int_t^T q(\boldsymbol{z}_s, s)\, \mathrm{d}s, \tag{6}$$

where $\phi(\tau_T)$ is a terminal cost and $q_t(\boldsymbol{z}_t) \equiv q(\boldsymbol{z}_t, t)$ is an instantaneous running cost. The expected value

$$V(\tau_t) = \mathbb{E}\left[S_t(\tau_T) \mid \tau_t \right] \tag{7}$$

of this cost as the system continues evolving from a trajectory τ_t according to the dynamics (5) then depends on the feedback law for the abstracted inputs $\boldsymbol{v}_t$.

The main goal of PI^2 is to be able to calculate both $V(\tau_t)$ and $\boldsymbol{v}_t$ from open-loop samples of system trajectories (*roll-outs*) in a highly parallelizable manner. In case of the dynamics (5), we will show that this goal can be accomplished by defining the feedback law as

$$\boldsymbol{v}_t := \arg\min_{\boldsymbol{v}_t} \mathbb{E}\left[\dot{V}(\tau_t) \mid \tau_t \right] + (\boldsymbol{v}_t - \boldsymbol{v}_{0t})^{\mathrm{T}} \mathbf{R}_t (\boldsymbol{v}_t - \boldsymbol{v}_{0t}). \tag{8}$$

Here, $\boldsymbol{v}_{0t}$ are feedforward control actions defined as

$$\boldsymbol{v}_{0t} = \begin{bmatrix} \boldsymbol{k}_t^{\mathrm{T}} & \boldsymbol{\mu}_{0t}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{9}$$

for some controller parameters $\boldsymbol{\mu}_{0t} \in \mathbb{R}^m$, and the penalty

$$\mathbf{R}_t = \frac{\lambda}{\alpha_t(\boldsymbol{z}_t)} \begin{bmatrix} \mathbf{P}_{0t} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{0t} \end{bmatrix} := \frac{\lambda}{\alpha_t(\boldsymbol{z}_t)} \mathbf{R}_{0t} \tag{10}$$

stems from a nominal block-diagonal regularization matrix $\mathbf{R}_{0t} \in \mathbb{R}^{2m \times 2m}$, $\mathbf{R}_{0t} > \mathbf{0}$ scaled by a constant $\lambda > 0$ and a variable factor

$$\alpha_t(\boldsymbol{z}_t) = \left\| \mathcal{L}_{\boldsymbol{\Xi}_t^{1/2}} V(\tau_t) \right\|_2^2 \Big/ \left\| \mathcal{L}_{\mathbf{G}(\boldsymbol{x}_t)\mathbf{R}_{0t}^{-1/2}} V(\tau_t) \right\|_2^2. \tag{11}$$

The Lie derivatives in this latter expression are defined by (2). Intuitively, $\alpha_t(\boldsymbol{z}_t)$ measures how much easier it is to change the value functional in the direction of the noise as opposed to the directions least penalized for the feedback. The PI^2 feedback (8) essentially aims to minimize the expected cost $V(\tau_t)$ without deviating too much from the feedforwards given by $\boldsymbol{v}_{0t}$. The variable λ is a design parameter controlling our willingness to correct for system noise in exchange for the expended input effort.

A practical realization of the PI^2 control strategy consists of the following two stages:

I. The parameters $\boldsymbol{\mu}_{0t}$ of the feedforwards $\boldsymbol{v}_{0t}$ in (9) are optimized to minimize the expected closed-loop cost

$V(\tau_t)$ given the feedback law (8). This can be done through natural gradient descent and is explained in detail in our previous work [4].

II. The closed-loop controls $\boldsymbol{v}_t$ given by (8) are calculated and implemented during real-time operation.

This work focuses on this latter, second stage. In the following section, we show that the required calculations are much simpler than suggested by previous work on this topic [1].

IV. Derivation of the PI^2 feedback controls

In this section, we derive an expression for the closed-loop control inputs $\boldsymbol{v}_t$ given by (8) that are to be implemented during real-time operation as part of the PI^2 control strategy. To this end, we first show that the expected value functional $V(\tau_t)$ under such feedbacks can be determined from open-loop sample trajectories of the system (5) around the feedforwards (9) using the so-called Feynman-Kac theorem. The results are then used to express the control inputs $\boldsymbol{v}_t$, again as a function of such open-loop roll-outs.

The derivations will make use of the following lemma.

Lemma 1. *Assume the system (5) evolves under the feedback law given by (8). Then, the control inputs are linked to the expected cost (7) by the equation:*

$$\boldsymbol{v}_t = \boldsymbol{v}_{0t} - \frac{1}{2}\mathbf{R}_t^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t). \tag{12}$$

Furthermore, $V(\tau_t)$ satisfies the PDE

$$\begin{aligned}-\Delta_t V(\tau_t) =& q_t + \left(\boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t)\right)^{\mathrm{T}}\left(\boldsymbol{F}_t(\boldsymbol{x}_t) + \mathbf{G}(\boldsymbol{x}_t)\boldsymbol{v}_{0t}\right)\\ &- \frac{1}{2}\left(\boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t)\right)^{\mathrm{T}}\mathbf{G}(\boldsymbol{x}_t)\mathbf{R}_t^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t)\\ &+ \frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Delta}_{\boldsymbol{zz}}V(\tau_t)\boldsymbol{\Xi}_t\right),\end{aligned} \tag{13}$$

with boundary condition $V(\tau_T) = S_T(\tau_T) = \phi(\tau_T)$.

Proof (sketch). First we approximate $V(\tau_t + \mathrm{d}t)$ using a Taylor-series expansion around $V(\tau_t)$. This is used to express $\mathbb{E}\left[\dot{V}(\tau_t) \mid \tau_t\right]$ and solve (8) for the controls $\boldsymbol{v}_t$. The second part of the lemma is then obtained by substituting these results into the dynamic programming equation $V(\tau_t) = \mathbb{E}\left[S_t(\tau_T) \mid \tau_t\right] = q_t\mathrm{d}t + \mathbb{E}\left[V(\tau_{t+\mathrm{d}t}) \mid \tau_t\right]$. The proof uses standard techniques from stochastic optimal control, see *e.g.*, [1], and the details are given separately in [8]. □

A. Application of the Feynman-Kac theorem

It is well-known that under the logarithmic transformation

$$V(\tau_t) = -\lambda \log \Psi(\tau_t) \tag{14}$$

of the value functional, the quadratic terms in the nonlinear PDE (13) can potentially be canceled out. The solution of the resulting linear PDE can then be approximated through open-loop sampling using the Feynman-Kac theorem [1]. This section shows that with the chosen penalty matrix (10), this cancellation does indeed occur, as well as how the sampling procedure is influenced by the feedforwards $\boldsymbol{v}_{0t}$ in (12).

Following the logarithmic transformation (14), the partial derivatives of $V(\tau_t)$ can be expressed as

$$\Delta_t V(\tau_t) = -\frac{\lambda}{\Psi(\tau_t)}\Delta_t\Psi(\tau_t), \quad \boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t) = -\frac{\lambda}{\Psi(\tau_t)}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t), \tag{15a}$$

and

$$\boldsymbol{\Delta}_{\boldsymbol{zz}}V(\tau_t) = \frac{\lambda}{\Psi^2(\tau_t)}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}} - \frac{\lambda}{\Psi(\tau_t)}\boldsymbol{\Delta}_{\boldsymbol{zz}}\Psi(\tau_t). \tag{15b}$$

Furthermore, from (11) and the definition (2) of Lie derivatives, the penalty scaling factor $\alpha_t(\boldsymbol{z}_t)$ can be written as:

$$\alpha_t(\boldsymbol{z}_t) = \frac{\|\boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t)\|^2_{\boldsymbol{\Xi}_t}}{\|\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}V(\tau_t)\|^2_{\mathbf{R}_{0t}^{-1}}} = \frac{\|\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\|^2_{\boldsymbol{\Xi}_t}}{\|\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\|^2_{\mathbf{R}_{0t}^{-1}}}. \tag{16}$$

Inserting the partial derivative transformations (15) into (13), the right-hand side of the PDE becomes

$$\begin{aligned}&q_t - \frac{\lambda}{\Psi(\tau_t)}\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\left(\boldsymbol{F}_t(\boldsymbol{x}_t) + \mathbf{G}(\boldsymbol{x}_t)\boldsymbol{v}_{0t}\right)\\ &\quad -\frac{1}{2}\frac{\lambda^2}{\Psi^2(\tau_t)}\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\mathbf{G}(\boldsymbol{x}_t)\mathbf{R}_t^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\\ &+\frac{1}{2}\operatorname{tr}\left[\left(\frac{\lambda}{\Psi^2(\tau_t)}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}} - \frac{\lambda}{\Psi(\tau_t)}\boldsymbol{\Delta}_{\boldsymbol{zz}}\Psi(\tau_t)\right)\boldsymbol{\Xi}_t\right].\end{aligned}$$

Comparing the quadratic terms of $\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)$, there is indeed an opportunity for cancellation in case:

$$\begin{aligned}&\frac{1}{2}\frac{\lambda^2}{\Psi^2(\tau_t)}\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\mathbf{G}(\boldsymbol{x}_t)\mathbf{R}_t^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\\ &\qquad = \frac{1}{2}\operatorname{tr}\left(\frac{\lambda}{\Psi^2(\tau_t)}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\boldsymbol{\Xi}_t\right).\end{aligned}$$

Substituting in $\mathbf{R}_t^{-1} = \frac{\alpha_t(\boldsymbol{z}_t)}{\lambda}\mathbf{R}_{0t}^{-1}$ from (10), dividing both sides by $\frac{\lambda}{2\Psi^2(\tau_t)}$, and using the trace identity $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ to yield a scalar within, this condition simplifies to:

$$\begin{aligned}&\alpha_t(\boldsymbol{z}_t)\left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\mathbf{G}(\boldsymbol{x}_t)\mathbf{R}_{0t}^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\\ &\qquad = \left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\boldsymbol{\Xi}_t\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t),\end{aligned}$$

or, expressed differently, to:

$$\alpha_t(\boldsymbol{z}_t)\left\|\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right\|^2_{\mathbf{R}_{0t}^{-1}} = \|\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\|^2_{\boldsymbol{\Xi}_t}.$$

This equation clearly holds due to the chosen form (16) for the multiplier $\alpha_t(\boldsymbol{z}_t)$, allowing the quadratic terms to cancel out. We emphasize that this cancellation is possible here without the previously necessary assumption in [1] and [4] relating the quadratic feedback regularization $\mathbf{R}_{0t}$ and noise covariance matrix $\boldsymbol{\Xi}_t$. With our notation, this assumption would take the form $\mathbf{G}(\boldsymbol{x}_t)\mathbf{R}_{0t}^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} = \boldsymbol{\Xi}_t$. We instead have a (less stringent) assumption that $\alpha_t(\boldsymbol{z}_t)$ is finite[1].

With the derived cancellation, the PDE (13) transforms into the simplified form

$$\begin{aligned}-\Delta_t\Psi(\tau_t) = &-\frac{\Psi(\tau_t)}{\lambda}q_t + \left(\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)\right)^{\mathrm{T}}\left(\boldsymbol{F}_t(\boldsymbol{x}_t) + \mathbf{G}(\boldsymbol{x}_t)\boldsymbol{v}_{0t}\right)\\ &+ \frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Delta}_{\boldsymbol{zz}}\Psi(\tau_t)\boldsymbol{\Xi}_t\right)\end{aligned} \tag{17}$$

[1] A sufficient condition for this is if the null space of $\mathbf{R}_{0t}^{-1/2}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}$ is contained in that of $\boldsymbol{\Xi}_t^{1/2}$; see [8] for details.

with the boundary condition $\Psi(\tau_T) = \exp(-\frac{1}{\lambda}\phi(\tau_T))$. The Feynman-Kac theorem for functionals [4], [16] then states that the solution to this transformed PDE at a given trajectory τ_t can be obtained as

$$\Psi(\tau_t) = \mathbb{E}_{OL}\left[\exp\left(-\frac{1}{\lambda}S_t(\tau_T)\right) \mid \tau_t\right], \tag{18}$$

where $\mathbb{E}_{OL}[\cdot \mid \tau_t]$ denotes taking the expectation by sampling the system (5) from continuations of τ_t in an open-loop fashion with $\boldsymbol{v}_t = \boldsymbol{v}_{0t}$ according to the dynamics:

$$\dot{\boldsymbol{z}}_t = \boldsymbol{F}_t(\boldsymbol{x}_t) + \mathbf{G}(\boldsymbol{x}_t)\boldsymbol{v}_{0t} + \boldsymbol{\Xi}_t^{1/2}\boldsymbol{w}_t, \tag{19a}$$

i.e., according to:

$$\begin{bmatrix}\dot{\boldsymbol{x}}_t \\ \dot{\boldsymbol{k}}_t\end{bmatrix} = \begin{bmatrix}\boldsymbol{f}(\boldsymbol{x}_t, t) + \mathbf{g}(\boldsymbol{x}_t)\boldsymbol{k}_t \\ \boldsymbol{\mu}_{0t}\end{bmatrix} + \begin{bmatrix}\boldsymbol{\Sigma}_{x,t}^{1/2}\boldsymbol{\epsilon}_{x,t} \\ \boldsymbol{\Sigma}_{k,t}^{1/2}\boldsymbol{\epsilon}_{k,t}\end{bmatrix}. \tag{19b}$$

Combining (14) and the result (18), the value functional corresponding to the PI^2 control strategy becomes

$$V(\tau_t) = -\lambda \log \mathbb{E}_{OL}\left[\exp\left(-\frac{1}{\lambda}S_t(\tau_T)\right) \;\middle|\; \tau_t\right]. \tag{20}$$

Our previous work [4] focused on determining the optimal parameters $\boldsymbol{\mu}_{0t}$ of the feedforwards $\boldsymbol{v}_{0t}$ such that the expected closed-loop cost (20) is minimized given an initial state τ_0. Other works such as [1], [6] can be interpreted as having zero feedforwards, instead relying on the feedbacks to find the optimal path. This requires injecting additional exploration noise during runtime and lowers the chance of sampling the best performing trajectories, hence necessitating more sophisticated sampling or iterative update methods to make the algorithm computationally feasible.

B. Feedback controls

In this section, we show that the PI^2 feedbacks (12) take a simpler form than what has been used in the original derivation [1] and subsequent works, *e.g.*, [5], [6], [11].

Substituting the penalty matrix (10) and the gradients (15) into the closed-loop controls (12), the PI^2 controls become:

$$\boldsymbol{v}_t = \boldsymbol{v}_{0t} + \frac{\alpha_t(\boldsymbol{z}_t)}{2\Psi(\tau_t)}\mathbf{R}_{0t}^{-1}\mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t), \tag{21}$$

where $\alpha_t(\boldsymbol{z}_t)$ is given by (16). We begin by approximating the quantity $\Psi(\tau_t)$ and its gradient $\Delta_{\boldsymbol{z}}\Psi(\tau_t)$ in this expression using sampled open-loop trajectories via (18).

The expectation (18) can be directly rewritten in the integral form

$$\Psi(\tau_t) = \int \pi_{OL}(\tau_{t^+:T}|\tau_t)\exp\left(-\frac{1}{\lambda}S_t(\tau_t{:}\tau_{t^+:T})\right)\mathrm{d}\tau_{t^+:T}, \tag{22}$$

where $\tau_{t^+:T} := \{\boldsymbol{z}_s \mid t < s \leq T\}$, $\pi_{OL}(\tau_{t^+:T}|\tau_t)$ denotes the probability of obtaining the trajectory $\tau_{t^+:T}$ as a continuation from τ_t, and $\tau_t{:}\tau_{t^+:T}$ denotes the concatenation of the two trajectory pieces τ_t and $\tau_{t^+:T}$. The transformed value functional itself can thus be directly approximated using N sample trajectory continuations $\tau_{t^+:T}^{(i)}$, $i = 1, \ldots, N$, as:

$$\begin{aligned}\Psi(\tau_t) &\approx \frac{1}{N}\sum_{i=1}^{N}\exp\left(-\frac{1}{\lambda}S_t(\tau_t{:}\tau_{t^+:T}^{(i)})\right) \\ &:= \frac{1}{N}\sum_{i=1}^{N}\exp\left(-\frac{1}{\lambda}S_t^{(i)}\right).\end{aligned} \tag{23}$$

Next, we must express the gradient $\boldsymbol{\Delta}_{\boldsymbol{z}}\Psi(\tau_t)$. To this end, we transfer to a discretized setting with time step $\Delta t \to 0$, and denote the value of quantities at discrete time instances k by an overhead bar, *i.e.*, $\bar{\boldsymbol{z}}_k := \boldsymbol{z}_{k\Delta t}$. Trajectories are also discretized in a similar fashion across $K = T/\Delta t$ time instances as

$$\bar{\tau}_k := (\bar{\boldsymbol{z}}_0, \bar{\boldsymbol{z}}_1, \ldots, \bar{\boldsymbol{z}}_k) \text{ and } \bar{\tau}_{k^+:K} := (\bar{\boldsymbol{z}}_{k+1}, \bar{\boldsymbol{z}}_{k+2}, \ldots, \bar{\boldsymbol{z}}_K).$$

We also define a discretized approximation of the cost (6):

$$\bar{S}_k(\bar{\tau}_k{:}\bar{\tau}_{k^+:K}) := \bar{\phi}(\bar{\tau}_k{:}\bar{\tau}_{k^+:K}) + \sum\nolimits_{k'=k}^{K-1} q(\bar{\boldsymbol{z}}_{k'}, k'\Delta t)\Delta t, \tag{24}$$

allowing the transformed value functional (22) to be rewritten in the limit $K \to \infty$ and equivalently $\Delta t \to 0$ as:

$$\begin{aligned}\Psi(\tau_t) = \lim_{\Delta t\to 0}\int &\pi_{OL}(\bar{\tau}_{k^+:K}|\bar{\tau}_k) \\ &\cdot \exp\left(-\frac{1}{\lambda}\bar{S}_k(\bar{\tau}_k{:}\bar{\tau}_{k^+:K})\right)\mathrm{d}\bar{\tau}_{k^+:K}.\end{aligned} \tag{25}$$

The probability of obtaining the trajectory continuation $\bar{\tau}_{k^+:K}$ is

$$\pi_{OL}(\bar{\tau}_{k^+:K}|\bar{\tau}_k) = \prod_{k'=k}^{K-1}\pi_{OL}(\bar{\boldsymbol{z}}_{k'+1}|\boldsymbol{z}_{k'}), \tag{26}$$

whose evaluation reduces to finding the probability $\pi_{OL}(\bar{\boldsymbol{z}}_{k'+1}|\boldsymbol{z}_{k'})$ of obtaining the consecutive states within the trajectory. At time instance $s = k'\Delta t$, let us define

$$\bar{\boldsymbol{H}}_{k'} := \boldsymbol{F}_s(\bar{\boldsymbol{x}}_{k'}) + \mathbf{G}(\bar{\boldsymbol{x}}_{k'})\bar{\boldsymbol{v}}_{0k'} \tag{27}$$

and

$$\bar{\boldsymbol{w}}_{k'} := \int_s^{s+\Delta t}\boldsymbol{\Xi}_\sigma^{1/2}\boldsymbol{w}_\sigma \mathrm{d}\sigma, \tag{28}$$

a random variable of dimension $p = m + n$ with covariance $\bar{\boldsymbol{\Xi}}_{k'} := \boldsymbol{\Xi}_s\Delta t$. The open-loop dynamics (19) then allows us to relate consecutive values of $\boldsymbol{z}_t$ using a forward Euler approximation scheme. In the limit $\Delta t \to 0$, we have:

$$\bar{\boldsymbol{z}}_{k'+1} = \bar{\boldsymbol{z}}_{k'} + \bar{\boldsymbol{H}}_{k'}\Delta t + \bar{\boldsymbol{w}}_{k'}. \tag{29}$$

This shows that the difference $\bar{\boldsymbol{z}}_{k'+1} - \bar{\boldsymbol{z}}_{k'}$ is a random variable with mean $\bar{\boldsymbol{H}}_{k'}\Delta t$ and covariance $\bar{\boldsymbol{\Xi}}_{k'}$, and therefore:

$$\begin{aligned}\pi_{OL}(\bar{\boldsymbol{z}}_{k'+1}|\bar{\boldsymbol{z}}_{k'}) = &\frac{1}{\sqrt{\det(2\pi\bar{\boldsymbol{\Xi}}_{k'})}} \\ &\cdot \exp\left(-\frac{1}{2}\left\|\bar{\boldsymbol{z}}_{k'+1} - \bar{\boldsymbol{z}}_{k'} - \bar{\boldsymbol{H}}_{k'}\Delta t\right\|_{\bar{\boldsymbol{\Xi}}_{k'}^{-1}}^2\right).\end{aligned} \tag{30}$$

Substituting this result into (26), the probability of the trajectory continuation $\bar{\tau}_{k^+:K}$ can be expressed as

$$\pi_{OL}(\bar{\tau}_{k^+:K}|\bar{\tau}_k) = \frac{1}{\bar{D}_k}\exp\left(-\frac{1}{\lambda}\bar{T}_k(\bar{\tau}_{k^+:K})\right), \tag{31}$$

where the introduced terms are

$$\bar{D}_k = \prod_{k'=k}^{K-1} \sqrt{\det(2\pi\bar{\boldsymbol{\Xi}}_{k'})} \tag{32}$$

and

$$\bar{T}_k(\bar{\tau}_{k^+:K}) = \frac{\lambda}{2} \sum_{k'=k}^{K-1} \left\| \bar{\boldsymbol{z}}_{k'+1} - \bar{\boldsymbol{z}}_{k'} - \bar{\boldsymbol{H}}_{k'} \Delta t \right\|^2_{\bar{\boldsymbol{\Xi}}_{k'}^{-1}}. \tag{33}$$

Remark 1. *Different discrete approximations become equivalent in the limit $\Delta t \to 0$. Instead of (29), we also could have used a backward Euler scheme according to the relation:*

$$\bar{\boldsymbol{z}}_{k'+1} = \bar{\boldsymbol{z}}_{k'} + \bar{\boldsymbol{H}}_{k'+1} \Delta t + \bar{\boldsymbol{w}}_{k'+1}. \tag{34}$$

This implies that in the limit $\Delta t \to 0$, both $\bar{\boldsymbol{H}}_{k'}$ and $\bar{\boldsymbol{\Xi}}_{k'}$ in (30)-(32) could be replaced by $\bar{\boldsymbol{H}}_{k'+1}$ and $\bar{\boldsymbol{\Xi}}_{k'+1}$, respectively. The gradients of these latter values with respect to the state $\bar{\boldsymbol{z}}_{k'}$ are clearly zero, and therefore the gradients of the former values must also vanish in the limit $\Delta t \to 0$, as the two approximations must yield the same result.

Remark 2. *In case $\bar{\boldsymbol{\Xi}}_{k'}$ is only positive semi-definite, its inverse and determinant in (30) must be replaced by the generalized inverse and pseudo-determinant, respectively [8].*

We can now substitute the result (31) back into (25) to obtain the transformed value functional

$$\Psi(\tau_t) = \lim_{\Delta t \to 0} \int \frac{1}{\bar{D}_k} \cdot \exp\left[-\frac{1}{\lambda} \left(\bar{S}_k(\bar{\tau}_k{:}\bar{\tau}_{k^+:K}) + \bar{T}_k(\bar{\tau}_{k^+:K}) \right) \right] \mathrm{d}\bar{\tau}_{k^+:K}. \tag{35}$$

At time $t = k\Delta t$, the gradient $\boldsymbol{\Delta}_{\boldsymbol{z}} \Psi(\tau_t) = \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \Psi(\tau_t)$ of this term can be expressed using the chain rule and (31) as:

$$\boldsymbol{\Delta}_{\boldsymbol{z}} \Psi(\tau_t) = -\tfrac{1}{\lambda} \lim_{\Delta t \to 0} \int \pi_{OL}(\bar{\tau}_{k^+:K} | \bar{\tau}_k) \exp\left(-\tfrac{1}{\lambda} \bar{S}_k(\bar{\tau}_k{:}\bar{\tau}_{k^+:K}) \right) \cdot \left(\boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{S}_k(\bar{\tau}_k{:}\bar{\tau}_{k^+:K}) + \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{T}_k(\bar{\tau}_{k^+:K}) \right) \mathrm{d}\bar{\tau}_{k^+:K}.$$

Similarly to (23), this can be approximated with $i = 1, \ldots, N$ trajectory roll-outs as:

$$\boldsymbol{\Delta}_{\boldsymbol{z}} \Psi(\tau_t) \approx -\frac{1}{\lambda} \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{1}{\lambda} S_t^{(i)} \right) \cdot \lim_{\Delta t \to 0} \left(\boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{S}_k(\bar{\tau}_k{:}\bar{\tau}^{(i)}_{k^+:K}) + \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{T}_k(\bar{\tau}^{(i)}_{k^+:K}) \right). \tag{36}$$

The gradient with respect to $\bar{S}_k(\cdot)$ at time $t = k\Delta t$ is

$$\lim_{\Delta t \to 0} \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{S}_k(\bar{\tau}_k, \bar{\tau}_{k^+:K}) = \lim_{\Delta t \to 0} \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{\phi}(\bar{\tau}_k{:}\bar{\tau}_{k^+:K}) = \boldsymbol{\Delta}_{\boldsymbol{z}_t} \phi(\tau_T), \tag{37}$$

because the gradient of the step-wise cost $q(\cdot)$ in (24) is negligible due to its Δt multiplier. On the other hand, evaluating the gradient of $\boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{T}_k(\bar{\tau}_{k^+:K})$ from (33) yields:

$$\begin{aligned} \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \bar{T}_k(\bar{\tau}_{k^+:K}) &= \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \frac{\lambda}{2} \sum_{k'=k}^{K-1} \left\| \bar{\boldsymbol{z}}_{k'+1} - \bar{\boldsymbol{z}}_{k'} - \bar{\boldsymbol{H}}_{k'} \Delta t \right\|^2_{\bar{\boldsymbol{\Xi}}_{k'}^{-1}} \\ &= \boldsymbol{\Delta}_{\bar{\boldsymbol{z}}_k} \frac{\lambda}{2} \left\| \bar{\boldsymbol{z}}_{k+1} - \bar{\boldsymbol{z}}_k - \bar{\boldsymbol{H}}_k \Delta t \right\|^2_{\bar{\boldsymbol{\Xi}}_k^{-1}} \\ &= -\lambda \bar{\boldsymbol{\Xi}}_k^{-1} \left(\bar{\boldsymbol{z}}_{k+1} - \bar{\boldsymbol{z}}_k - \bar{\boldsymbol{H}}_k \Delta t \right) \\ &= -\lambda \frac{1}{\Delta t} \boldsymbol{\Xi}_t^{-1} \bar{\boldsymbol{w}}_k \end{aligned} \tag{38}$$

where $\bar{\boldsymbol{w}}_k$ is defined in (28). The dependency of $\bar{\boldsymbol{H}}_k$ and possibly $\bar{\boldsymbol{\Xi}}_k$ on $\bar{\boldsymbol{z}}_k$ was ignored in this derivation, as the corresponding gradients must vanish due to the reasons outlined in Remark 1. This considerably simplifies both the derivation and the results compared to [1].

Substituting (23) and (36)-(38) back into (21) finally yields the PI^2 feedback controls as:

$$\boldsymbol{v}_t = \boldsymbol{v}_{0t} - \frac{\alpha_t(\boldsymbol{z}_t)}{2\lambda} \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} \boldsymbol{\delta}_t, \tag{39}$$

where the term $\boldsymbol{\delta}_t$ is defined as

$$\boldsymbol{\delta}_t := \sum_{i=1}^{N} w_{\mathrm{PI}^2}^{(i)} \left(\boldsymbol{\Delta}_{\boldsymbol{z}_t} \phi(\tau_T^{(i)}) - \lim_{\Delta t \to 0} \frac{\lambda}{\Delta t} \boldsymbol{\Xi}_t^{-1} \bar{\boldsymbol{w}}_k^{(i)} \right), \tag{40}$$

and the introduced PI^2 weights are

$$w_{\mathrm{PI}^2}^{(i)} = \frac{\exp\left(-\frac{1}{\lambda} S_t^{(i)} \right)}{\sum_{j=1}^{N} \exp\left(-\frac{1}{\lambda} S_t^{(j)} \right)}. \tag{41}$$

Finally, the scaling factor $\alpha_t(\boldsymbol{z}_t)$ can be expressed by substituting (36)-(38) into (16) to yield

$$\alpha_t(\boldsymbol{z}_t) = \frac{\boldsymbol{\delta}_t^{\mathrm{T}} \boldsymbol{\Xi}_t \boldsymbol{\delta}_t}{\boldsymbol{\delta}_t^{\mathrm{T}} \mathbf{G}(\boldsymbol{x}_t) \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} \boldsymbol{\delta}_t}, \tag{42}$$

which allows the feedback (39) to be written compactly as

$$\boldsymbol{v}_t = \boldsymbol{v}_{0t} - \frac{1}{2\lambda} \frac{\boldsymbol{\delta}_t^{\mathrm{T}} \boldsymbol{\Xi}_t \boldsymbol{\delta}_t}{\boldsymbol{\delta}_t^{\mathrm{T}} \mathbf{G}(\boldsymbol{x}_t) \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} \boldsymbol{\delta}_t} \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} \boldsymbol{\delta}_t. \tag{43}$$

The main computational effort for determining this feedback is to simulate $i = 1, \ldots, N$ roll-outs and compute their corresponding trajectory costs $S_t^{(i)}$ in order to calculate the weights $w_{\mathrm{PI}^2}^{(i)}$ in (41). This can be sped up considerably as it is a completely parallelizable operation. The term $\boldsymbol{\delta}_t$ in (40) and $\boldsymbol{v}_t$ in (43) are then readily computable, making PI^2 a potentially feasible control strategy for real-time applications.

We conclude this section by comparing our result (43) with the expression for the feedback given by [1], Section 2.4. To do so, we set $\boldsymbol{v}_{0t} = \mathbf{0}$, $\boldsymbol{\Delta}_{\boldsymbol{z}_t} \phi(\tau_T^{(i)}) = \mathbf{0}$, and $\mathbf{G}(\boldsymbol{x}_t) \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} = \boldsymbol{\Xi}_t$ according to the missing feedforwards, the cost definition, and the $\lambda \mathbf{R}^{-1} = \boldsymbol{\Sigma}$ assumption therein, respectively. In this case, $\alpha_t(\boldsymbol{z}_t) = 1$ as seen from (42), and the feedback (43) simplifies to:

$$\boldsymbol{v}_t = \frac{1}{2\lambda} \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} \left(\mathbf{G}(\boldsymbol{x}_t) \mathbf{R}_{0t}^{-1} \mathbf{G}(\boldsymbol{x}_t)^{\mathrm{T}} \right)^{-1} \tilde{\boldsymbol{\delta}}_t, \tag{44}$$

where $\tilde{\boldsymbol{\delta}}_t = \sum_{i=1}^{N} w_{\mathrm{PI}^2}^{(i)} \lim_{\Delta t \to 0} \frac{\lambda}{\Delta t} \bar{\boldsymbol{w}}_k^{(i)}$ as obtained from (40). Compared to (18)-(20) in [1], there is a factor $\frac{1}{2}$ difference due to slightly different phrasings of the optimal control problem (and $\frac{1}{\lambda}$ cancels out from $\tilde{\boldsymbol{\delta}}_t$). More importantly, however, our definition (41) of $w_{\mathrm{PI}^2}^{(i)}$ implies that even with a state-dependent $\mathbf{G}(\boldsymbol{x}_t)$ matrix, it is not necessary to generalize the cost $S_t^{(i)}$ for the PI^2 weight calculation. A complicated term denoted by $\boldsymbol{b}_t$ in [1] is also missing from (44). Our results therefore suggest a simpler-to-implement PI^2 feedback controller than previously thought.

In the next section, the correctness of the derived feedback (43) is verified numerically by showing that the corresponding closed-loop performance matches the prediction (20).

V. Simulation results

We demonstrate the PI^2 control strategy and the correctness of the theoretical results using an optimal control problem involving the unicycle system

$$\begin{bmatrix}\dot{x}_t\\ \dot{y}_t\\ \dot{\theta}_t\end{bmatrix} = \begin{bmatrix}\cos(\theta_t) & 0\\ \sin(\theta_t) & 0\\ 0 & 1\end{bmatrix}\left(\begin{bmatrix}v_t\\ \omega_t\end{bmatrix} + \boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\epsilon}_t\right) \tag{45}$$

where $\boldsymbol{\Sigma}_0 = 0.01\mathbf{I}$. We note that the results of [1] are not applicable to such non-holonomic systems. Feedforwards for the control inputs $\boldsymbol{u}_t = [v_t \quad \omega_t]^{\mathrm{T}}$ are given by $\boldsymbol{k}_t = [k_{v,t} \quad k_{\omega,t}]^{\mathrm{T}}$. Denoting $\hat{\boldsymbol{x}}_t = [x_t \quad y_t]^{\mathrm{T}}$, a cost aiming to reach a goal at $\hat{\boldsymbol{x}}_g = [3.5 \quad 2.0]^{\mathrm{T}}$ and avoid an obstacle at $\hat{\boldsymbol{x}}_o = [2.0 \quad 1.0]^{\mathrm{T}}$ using minimal control effort is defined as:

$$S(\tau_t) = 2\left\|\hat{\boldsymbol{x}}_T - \hat{\boldsymbol{x}}_g\right\|_2 + \int_t^T 500\min(1.2 - \left\|\hat{\boldsymbol{x}}_t - \hat{\boldsymbol{x}}_o\right\|_2^2, 0.2) + 0.2k_{v,t}^2 + k_{\omega,t}^2\,\mathrm{d}t,$$

The problem is defined for a horizon $T = 10$ and simulated with time step $\Delta t = 0.01s$. The PI^2 feedback law parameters are defined as $\mathbf{P}_{0t} = \begin{bmatrix}100 & 0\\ 0 & 20\end{bmatrix}$, $\mathbf{Q}_{0t} \to \infty\mathbf{I}$, and $\lambda = 0.01$.

First, the optimal parameterization of the nominal derivatives of the feedforwards $\boldsymbol{k}_t$ are determined using the theory described in [4]. Resulting open-loop trajectories are shown in Figure 1 in gray, with the thick blue curve showing the nominal, noiseless path. The open-loop performance is $\mathbb{E}_{OL}[S] \approx 97$, and the expected closed-loop performance can be calculated using (20) to be $\mathbb{E}_{CL,\text{theoretical}}[S] \approx 1.08$.

The PI^2 feedbacks (43) are then implemented and a sample of 100 closed-loop runs are simulated using a different number of N roll-outs for feedback calculation. Sample closed-loop paths are depicted in Figure 1 in green. The achieved expected costs are summarized in Table 1, and show the correctness of the theoretical prediction as $N \to \infty$.

TABLE I: Achieved average closed-loop costs as a function of the number of N roll-outs used for feedback calculation. The results are approximated from 100 sample runs.

$\boldsymbol{v}_t$ calculation	$N = 500$	$N = 5000$	$N = 50000$	theoretical
$\mathbb{E}_{CL}[S]$	2.5	1.21	1.14	1.08

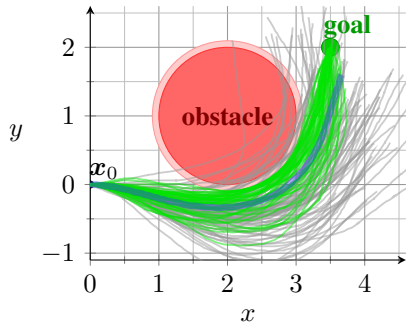

Fig. 1: Sample open-loop (gray) and closed-loop (green) trajectories obtained for the simulation example.

VI. Conclusions

This paper presents a novel view of PI^2 as a two-stage control strategy, first determining optimal feedforward controls and then implementing feedbacks during real-time operation. Compared to previous results, the theoretical derivations of the feedbacks are greatly simplified, and the final result is shown to take a simple form for a wide range of system dynamics. The correctness of the theoretical derivations was demonstrated numerically through simulations for the first time according to the authors' knowledge. The results could potentially ease the implementation of PI^2 and make it a more viable control strategy for practical applications. The simplified theoretical derivations also lay foundations for and offer a multitude of directions for future research, such as improving the sample efficiency of the feedback control calculations, discrete-time variants of the algorithm, and extensions to the multi-agent case.

References

[1] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research (JMLR)*, vol. 11, pp. 3137–3181, 2010.

[2] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *Proc. 29th International Conference on Machine Learning (ICML)*, 2012, pp. 1547–1554.

[3] O. Sigaud and F. Stulp, "Policy search in continuous action domains: an overview," *Neural Networks*, vol. 113, pp. 28–40, 2019.

[4] P. Varnai and D. V. Dimarogonas, "Path integral policy improvement: an information-geometric approach," *Journal of Machine Learning Research (JMLR)*, submitted for publication, preprint available online on ResearchGate DOI: 10.13140/RG.2.2.13969.76645.

[5] S. Satoh, H. J. Kappen, and M. Saeki, "An iterative method for nonlinear stochastic optimal control based on path integrals," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 262–276, 2016.

[6] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guid., Cont., and Dyn.*, vol. 40, no. 2, pp. 344–357, 2017.

[7] M. Hibbard, Y. Wasa, and T. Tanaka, "Path integral control for stochastic dynamic traffic routing problems," in *Proc. 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 261–267.

[8] P. Varnai and D. V. Dimarogonas, "The two-stage PI^2 control strategy," Available online on ResearchGate DOI: 10.13140/RG.2.2.18017.84322, 2021.

[9] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, and S. Levine, "Path integral guided policy search," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3381–3388.

[10] T. Lefebvre and G. Crevecoeur, "Path integral policy improvement with differential dynamic programming," in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, 2019, pp. 739–745.

[11] N. Wan, A. Gahlawat, N. Hovakimyan, E. A. Theodorou, and P. G. Voulgaris, "Cooperative path integral control for stochastic multi-agent systems," in *Proc. 2021 American Control Conference (ACC)*, 2021, pp. 1262–1267.

[12] K. Yamamoto, R. Ariizumi, T. Hayakawa, and F. Matsuno, "Path integral policy improvement with population adaptation," *IEEE Transactions on Cybernetics*, pp. 1–11, 2020.

[13] F. Ficuciello, D. Zaccara, and B. Siciliano, "Synergy-based policy improvement with path integrals for anthropomorphic hands," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1940–1945.

[14] W. Zhu, X. Guo, Y. Fang, and X. Zhang, "A path-integral-based reinforcement learning algorithm for path following of an autoassembly mobile robot," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4487–4499, 2020.

[15] C. Celemin, G. Maeda, J. Ruiz-del Solar, J. Peters, and J. Kober, "Reinforcement learning of motor skills using policy search and human corrective advice," *The International Journal of Robotics Research (IJRR)*, vol. 38, no. 14, pp. 1560–1580, 2019.

[16] B. Dupire, "Functional Itô calculus," *Quantitative Finance*, vol. 19, no. 5, pp. 721–729, 2019.