Advancing Trajectory Optimization with Approximate Inference: Exploration, Covariance Control and Adaptive Risk

Joe Watson, Jan Peters

***Abstract*— Discrete-time stochastic optimal control remains a challenging problem for general, nonlinear systems under significant uncertainty, with practical solvers typically relying on the certainty equivalence assumption, replanning and/or extensive regularization. Control-as-inference is an approach that frames stochastic control as an equivalent inference problem, and has demonstrated desirable qualities over existing methods, namely in exploration and regularization. We look specifically at the input inference for control (I2C) algorithm, and derive three key characteristics that enable advanced trajectory optimization: An 'expert' linear Gaussian controller that combines the benefits of open-loop optima and closed-loop variance reduction when optimizing for nonlinear systems, adaptive risk sensitivity for regularized exploration, and performing covariance control through specifying the terminal state distribution.**

I. INTRODUCTION

The control of stochastic environments is a ubiquitous problem across many domains, but remains challenging computationally for the general case. Stochastic optimal control (SOC) solvers trade off computational complexity, exploitation of domain knowledge, use of simplifying assumptions and/or numerical sensitivity. In this work, we discuss these trade-offs from the perspective of *control-as-inference* [1], [2], which seeks to frame stochastic control as a probabilistic inference problem. This translation from optimization to inference can be seen as a subset of probabilistic numerics [3], which utilize statistical methods to solve numerical problems, providing uncertainty quantification, regularization and faster convergence. These viewpoints are made considering input inference for control (I2C) [4], a fully probabilistic inference-based solver that frames SOC as input estimation. Unlike classical control theory, an inference-based approach naturally lends itself to the manipulation of uncertainties, while also benefiting from mature approximate inference methods for complex dynamics and uncertainties where exact inference is intractable. Considering nonlinear trajectory optimization, a key quality is the ability to iteratively explore in a numerically stable manner due to the lack of a closed-form solution. Many SOC algorithms require regularization heuristics such as line search or trust regions to achieve this, as well as initializing with a sufficiently random solution to encourage progress. We show inference methods naturally achieve this, using belief akin to a trust region and leveraging an adaptive *risk-seeking* strategy for exploration. Another numerical issue in nonlinear SOC is optimizing open- or closed-loop. Open-loop strategies are brittle, but simpler to compute, whereas closed-loop controllers offer additional stability and therefore reduce the variance of the state distribution. However, optimizing with feedback tends to yield actuation-heavy solutions due to the interplay between exploration and local feedback during optimization. We show how using the belief in the controller during optimization allows for this interplay to be managed, producing superior results on two nonlinear tasks. Finally, covariance control [14] is achieved using only a minor adjustment to I2C due to its similarity to inference, enabling nonlinear distributional control that is simpler compared to alternative approaches.

The paper is structured as follows. Section II discusses the SOC literature and related theory, Section III describes control-as-inference and the I2C algorithm, and Section III details the extensions to I2C that enable advanced trajectory optimization.

II. STOCHASTIC OPTIMAL CONTROL

In this section we outline SOC, baseline methods, the risk-based variant and covariance control.

A. The Finite-Horizon, Discrete-Time Setting

We specifically consider a stochastic, discrete-time, fully-observed, nonlinear, time-varying dynamical system, $\boldsymbol{f}_t$, with state $\boldsymbol{x} \in \mathbb{R}^{d_x}$ and input $\boldsymbol{u} \in \mathbb{R}^{d_u}$, desiring the optimal controls over a time horizon T that minimizes the cost functions $C_{0:T}$ in expectation,

$$\min_{\boldsymbol{u}_{0:T\text{-}1}} \quad \mathbb{E}[C_T(\boldsymbol{x}_T) + \textstyle\sum_{t=0}^{T\text{-}1} C_t(\boldsymbol{x}_t, \boldsymbol{u}_t)] \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{f}_t(\boldsymbol{x}_t, \boldsymbol{u}_t). \tag{2}$$

From Bellman's dynamic programming perspective, the optimal controls can be derived from the value function V_t for the expected cost. For $t = 0, \dots, T\text{-}1$,

$$V_t(\boldsymbol{x}) = \min_{\boldsymbol{u}} \mathbb{E}[C_t(\boldsymbol{x}, \boldsymbol{u}) + V_{t+1}(\boldsymbol{f}_t(\boldsymbol{x}, \boldsymbol{u}))], \tag{3}$$

initializing V_T using the terminal cost. For time-varying linear dynamics, quadratic costs and Gaussian disturbances, the classic LQG solution can be derived in closed-form from Equation 3. However, in practice this value function is intractable globally given arbitrary dynamics, objectives and uncertainties. As a result, solvers involve either devising a means of approximating the value function, or tackling the constrained optimization problem directly instead, ignoring the temporal structure.

Differential dynamic programming (DDP) [5] exploits the temporal structure of the SOC task, iteratively constructing local Taylor series approximations of the dynamics and cost

The authors are with the Department of Computer Science, Technische Universität Darmstadt, Germany.
Contact watson@ias.informatik.tu-darmstadt.de.

to compute a local approximate value function for closed-loop optimization. The stochastic DDP extension (SDDP) to consider the impact of Gaussian disturbances on the expected value functions (SDDP) [6]. To mitigate the computational burden of computing the Hessian, a Gauss-Newton approximation of DDP is used (iLQR [7], iLQG [8]). These methods all require regularization and line-search routines for stable convergence. Guided policy search (GPS) is a method that uses iLQG to solve a variational form of trajectory optimization [9]. This results in a maximum entropy linear Gaussian control law, and a KL divergence constraint on the trajectory distribution is used to stabilize the optimization.

For nonlinear, deterministic systems, sequential quadratic programming (SQP) has been used to compute the optimal action- or state-action sequence when framed as a constrained optimization problem, with quadratic convergence under mild assumptions [10]. Linearizing along this trajectory, the LQG solution has then been applied to compute an approximate solution to the SOC problem (T-LQG [11], T-PFC [12]), justified by a small noise performance bound [13]. While mature and highly-optimized SQP solvers can be used off the shelf, they can suffer computationally in the SOC domain. By failing to exploit the temporal structure of the problem (multiple-shooting) or by considering numerically sensitive objectives (single-shooting), SQPs do not scale gracefully under long planning horizons.

B. Covariance Control

While SOC solvers are typically concerned with optimizing the expected cost over the mean trajectory, methods have also been devised to control higher order statistics. Covariance control [14] specifically looks at constraining the mean and covariance of the terminal state to a target distribution $p(\boldsymbol{x}_T^*)$. The linear Gaussian setting has been extensively studied, for both discrete [15] and continuous time [16], where it can be shown that a solution exists should the system be controllable and $\boldsymbol{\Sigma}_{\boldsymbol{x}_T^*} - \boldsymbol{\Sigma}_{\boldsymbol{\eta}_T} > 0$, given process noise covariance $\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t}$. The hard constraint can be tackled by decomposing the problem into feedforward control for the mean, and linear feedback control for the covariance [15]. The discrete-time case corresponds to minimizing the relative entropy between two MDPs and minimum-energy LQG [17], [15], where the terminal cost corresponds to the Lagrange multiplier of the constraint. The nonlinear Gaussian case has been tackled using stochastic DDP [18] and through the combination of sequential convex programs and statistical linearization [19]. The problem can also be viewed as a form of optimal transport and the Schrödinger bridge, which seeks to find the mapping (i.e. dynamical system) that transforms one distribution to another [20]. The solution here is iterative, using forward and backward Riccati equations until both constraints are satisfied.

C. Risk-Sensitive Control

Introduced by Jacobson, risk-sensitive linear exponential quadratic Gaussian (LEQG) control [21], [22] derives a control law that, unlike LQG, is dependent on the severity of uncertainty in the system dynamics. This sensitivity is determined by the scaling parameter σ in the now exponentiated objective,

$$\min_{\boldsymbol{u}_{0:T\text{-}1}} \text{-}\frac{2}{\sigma}\mathbb{E}\left[\exp\left(\text{-}\frac{\sigma}{2}\left[C_T(\boldsymbol{x}_T) + \sum_{t=0}^{T\text{-}1} C_t(\boldsymbol{x}_t, \boldsymbol{u}_t)\right]\right)\right], \quad (4)$$

for quadratic costs $C_{0:T}$ and $\sigma \in \mathbb{R}$. The consequence of this objective transformation is that, while they still form a LQG-like discrete algebraic Riccati equation (DARE), the weights of the value function now also depend on the covariance of the Gaussian disturbance $\boldsymbol{\eta}_t$ and risk sensitivity σ, resulting in the transformation,

$$\boldsymbol{P}_{\text{LEQG},t} = (\boldsymbol{P}^{\text{-}1}_{\text{LQG},t} + \sigma\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t})^{\text{-}1}, \quad (5)$$

$$\text{given } V_t(\boldsymbol{x}_t) = \boldsymbol{x}_t^\top \boldsymbol{P}_t \boldsymbol{x}_t + 2\boldsymbol{p}_t^\top \boldsymbol{x}_t + p_t. \quad (6)$$

While the relationship between the linear feedback law and the value function is unchanged from the LQG case, the adjustment of the value function results in 'risk-preferring' ($\sigma{>}0$) or 'risk-averse' ($\sigma{<}0$) strategies under uncertainty. On a high level, this tuning can be viewed as a mean / variance trade-off in the evaluated cost. Moreover, for $\sigma{=}0$ (risk-neutral) the control strategy reduces to the LQG result. This behavior has lead to interest in risk-sensitive methods from domains such as quantitative finance and behavioral sciences. The relationship between system uncertainty and risk makes this formulation interesting from the inference perspective, which is discussed in Section IV-B.

III. CONTROL AS INFERENCE

The duality between optimal control and statistical inference techniques dates back to the work of Kalman [23] from his work on the LQG problem. Later, relative entropy (KL divergence) was shown as a means of framing the SOC problem [17], also known as 'probabilistic control design' [24]. While path integral control [1] strengthened the connection in continuous-time, adopting methods from probabilistic graphical models demonstrated the relation between message passing for inference and the discrete-time Riccati equations in SOC [25], [26]. AICO [2] demonstrated that, for open-loop nonlinear trajectory optimization, the linearization-based approximate message passing computation resembled Gauss-Newton SOC, but converged faster due to its exploratory forward pass and use of priors over control inputs. However, AICO's exploration prior was fixed due to its dual role as the control regularization, and the translation of the cost to the distributions is hand tuned. Input inference for control [4] builds on AICO, extending the probabilistic perspective by framing optimal control as input estimation. The closed-loop controller is defined as the conditional distribution of the state-action trajectory, as in posterior policy iteration (PPI) [27], [28], and crucially the graphical model is defined such that the controls have independent priors, allowing for iterative, variable exploration that corresponds to maximum entropy control. Moreover, the translation of the cost function for the graphical model, through a scaling term, is jointly optimized during inference. The combination

$$\max_{\boldsymbol{X},\boldsymbol{U},\boldsymbol{\theta}} p(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Z},\boldsymbol{\theta}) = p(\boldsymbol{x}_0)p(\boldsymbol{z}_T|\boldsymbol{x}_T,\boldsymbol{\theta}) \prod_{t=0}^{T\text{-}1} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t,\boldsymbol{u}_t) \prod_{t=0}^{T\text{-}1} p(\boldsymbol{z}_t|\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\theta}) \prod_{t=0}^{T\text{-}1} p(\boldsymbol{u}_t|\boldsymbol{x}_t), \tag{7}$$

$$\text{Dynamics} \qquad p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t,\boldsymbol{u}_t): \qquad \boldsymbol{x}_{t+1} = \boldsymbol{f}_t(\boldsymbol{x}_t,\boldsymbol{u}_t) + \boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t}), \tag{8}$$

$$\text{Cost} \qquad p(\boldsymbol{z}_t|\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{\theta}): \qquad \boldsymbol{z}_t = \boldsymbol{g}_t(\boldsymbol{x}_t,\boldsymbol{u}_t) + \boldsymbol{\xi}_t, \qquad \boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}_{\boldsymbol{\xi}}(\boldsymbol{\theta})), \tag{9}$$

of principled exploration and probabilistic regularization enables competitive performance against comparable algorithms for nonlinear optimal control. To understand how I2C works, there are four critical components: The latent variable model, expectation maximization, message passing and the transformation of the control cost functions into likelihoods.

A. Latent Variable Models

As in state estimation, I2C uses a sequential latent variable model for the state-action trajectory over time. However, in the control-as-inference setting, there is no data. Instead, the 'known' quantity is the desired trajectory $\boldsymbol{Z}{=}\boldsymbol{z}_{0:T}$, which is some transformation $\boldsymbol{g}_t(.)$ of the state-action space. The transformation is task specific, e.g. $\boldsymbol{z}{=}[\boldsymbol{x},\boldsymbol{u}]^\top$ for LQR, $\boldsymbol{z}{=}\boldsymbol{u}$ for minimum-energy control, or $\boldsymbol{z}$ represents cartesian coordinates for operational space control. Regardless of form, it is assumed to be fixed and part of the objective. The inference problem is therefore computing the most likely state-action distribution that reconstructs this desired trajectory, which can be done through optimizing the joint likelihood (Equation 7). As $\boldsymbol{Z}$ is specified, rather than sampled data, the belief in I2C is with respect to *optimality*, rather than typical statistical uncertainty.

B. Cost Functions and Constraints as Likelihoods

In practice, this desired trajectory $\boldsymbol{Z}$ will not be perfectly achieved. Therefore, we incorporate additive uncertainty into the observation model to explain this error (Equation 9). In Section II-A, we discussed LQR for its tractable quadratic the control costs. These distances can be expressed more concisely as a Mahalanobis distance, $\|\boldsymbol{y}{-}\boldsymbol{x}\|^2_{\boldsymbol{S}}{=}(\boldsymbol{y}{-}\boldsymbol{x})^\top\boldsymbol{S}^{\text{-}1}(\boldsymbol{y}{-}\boldsymbol{x})$. Assuming Gaussian distributions, we can draw a connection between LQR and I2C, as the 'energy' / log-likelihood of this distribution is also a Mahalanobis distance. For Gaussian state space models, the log-likelihood takes an attractive quadratic form,

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}_\eta), \tag{10}$$

$$\begin{aligned} -\mathcal{L}_p(\boldsymbol{y},\boldsymbol{x}) &= -\log p(\boldsymbol{y},\boldsymbol{x}) \\ &= \|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|^2_{\boldsymbol{\Sigma}_\eta} + d_x \log|2\pi\boldsymbol{\Sigma}_\eta|. \end{aligned} \tag{11}$$

However, while the control objective in LQG is affine-invariant, the log-likelihood is not. Therefore, for equivalence with the observation log-likelihood, there is an unknown scale factor (α) to relate the LQG cost weights ($\boldsymbol{Q},\boldsymbol{R}$) to the *precision* (inverse covariance) of the 'disturbance' $\boldsymbol{\xi}$ in I2C, $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^{\text{-}1}{=}\alpha\,\text{diag}(\boldsymbol{Q},\boldsymbol{R})$. In Section IV-B, we discuss how α relates to risk sensitivity. As the only unknown model parameter in the inference problem ($\boldsymbol{\theta}{=}\{\alpha\}$), a benefit of I2C is that α can be automatically tuned during inference.

C. Inference of the Graphical Model

From Equation 7, we see that the general I2C inference problem depends on two unknowns: the latent state-action distribution $p(\boldsymbol{X},\boldsymbol{U})$ and model parameters $\boldsymbol{\theta}$. Performing inference with both these unknown quantities jointly is intractable. Fortunately, for a Gaussian dynamical system computation can be achieved iteratively using expectation maximization (EM) [29], [30], which guarantees monotonic improvement. In the EM convention, the E-step corresponds to estimating the latent state-action distribution given the model, while the M-step the optimizes the model parameters $\boldsymbol{\theta}$ to maximize the *expected* log-likelihood given the estimated latent distribution. Using Equation 11, this is of the form,

$$\begin{aligned} -\mathbb{E}[\mathcal{L}_p(\boldsymbol{y},\boldsymbol{x})] =& \text{tr}\{\boldsymbol{\Sigma}_{\boldsymbol{\eta}}^{\text{-}1}\,\mathbb{E}[(\boldsymbol{y}{-}\boldsymbol{f}(\boldsymbol{x}))(\boldsymbol{y}{-}\boldsymbol{f}(\boldsymbol{x}))^\top]\} \\ &+ d_x \log|2\pi\boldsymbol{\Sigma}_{\boldsymbol{\eta}}|. \end{aligned} \tag{12}$$

This EM procedure is iterated till convergence. While the M-step improvement can be expressed in closed-form [4], the E-step requires closer consideration.

To perform the E-step efficiently, message passing methods are a flexible approach for designing such algorithms [31]. Passing linear Gaussian messages is especially straightforward, following simple rules that can be expressed in closed form. For nonlinear Gaussian messages, approximate inference can be performed in several ways. Akin to the various nonlinear Bayesian filters, Taylor series approximations, quadrature and Monte Carlo methods can be used to approximate the marginalization integral [32]. As the graph is a Gaussian dynamical system, the E-step relates closely to general recursive Bayesian smoothing, using Equations 8 and 9 as the Gaussian dynamics and 'observation' model respectively. However, the key separation from typical smoothing is the inclusion of the input as part of the probabilistic state. With a prior on the state-action distribution at time t, during the forward message passing we compute $p(\overrightarrow{\boldsymbol{x}}_t, \overrightarrow{\boldsymbol{u}}_t)$ from the incoming $\overrightarrow{\boldsymbol{x}}_t$ computed in the previous timestep by

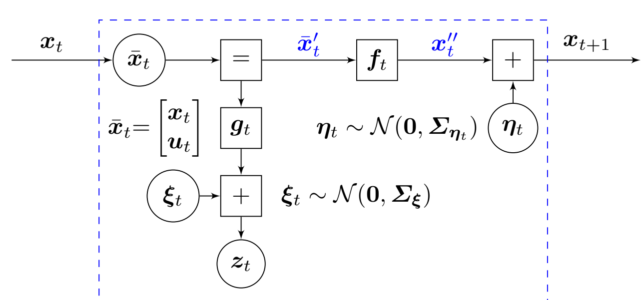


Fig. 1. The graphical model of I2C for a single timestep.

TABLE I

The evaluation of SOC algorithms on finite-horizon input-constrained control tasks, comparing variations based on tackling the stochastic (S) or certainty equivalent (CE) setting and using open-loop (FF), closed-loop (FB) or expert (E) controllers during optimization. Despite each algorithm using different numerical methods, these features identify similarities in performance. Percentiles were computed from 100 rollouts.

Environment	**10th, 90th Cost Percentiles** ($\times 10^3$)								
	I2C (S, E)	I2C (CE, E)	I2C (S, FF)	I2C (S, FB)	I2C (CE, FF)	I2C (CE, FB)	T-PFC (CE, FF)	iLQR (CE, FB)	GPS (S, FB)
Pendulum	13.46, 21.53	**12.81, 17.11**	17.72, 21.94	19.23, 21.43	13.97, 26.77	19.49, 22.31	18.10, 26.31	23.33, 26.46	19.45, 20.91
Cartpole	85.06, 87.43	**81.83, 83.87**	89.53, 94.67	93.53, 95.71	86.93, 89.75	121.89, 123.88	111.31, 118.57	142.23, 145.78	120.80, 122.45

marginalizing over the conditional distribution,

$$p(\overrightarrow{\boldsymbol{u}}_t) = \int p(\boldsymbol{u}_t|\boldsymbol{x}_t{=}\overrightarrow{\boldsymbol{x}}_t)p(\overrightarrow{\boldsymbol{x}}_t)d\overrightarrow{\boldsymbol{x}}_t, \tag{13}$$

$$\boldsymbol{\mu}_{\overrightarrow{\boldsymbol{u}}_t} = \boldsymbol{\mu}_{\boldsymbol{u}_t} + \boldsymbol{K}_t(\boldsymbol{\mu}_{\overrightarrow{\boldsymbol{x}}_t} - \boldsymbol{\mu}_{\boldsymbol{x}_t}), \tag{14}$$

$$\boldsymbol{\Sigma}_{\overrightarrow{\boldsymbol{u}\boldsymbol{u}}_t} = \boldsymbol{\Sigma}_{\boldsymbol{u}_t} - \boldsymbol{\Sigma}_{\boldsymbol{u}\boldsymbol{x}_t}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{x}\boldsymbol{x}_t}\boldsymbol{\Sigma}^{\top}_{\boldsymbol{u}\boldsymbol{x}_t} + \boldsymbol{K}_t\boldsymbol{\Sigma}_{\overrightarrow{\boldsymbol{x}}_t}\boldsymbol{K}_t^{\top}, \tag{15}$$

$$\boldsymbol{\Sigma}_{\overrightarrow{\boldsymbol{u}\boldsymbol{x}}_t} = \boldsymbol{\Sigma}_{\boldsymbol{u}\boldsymbol{x}_t}, \quad \boldsymbol{K}_t = \boldsymbol{\Sigma}_{\boldsymbol{u}\boldsymbol{x}_t}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{x}\boldsymbol{x}_t}. \tag{16}$$

Where $\boldsymbol{K}_t$ is the I2C feedback control gain, which has been shown to be equivalent to maximum entropy LQR in the deterministic setting [4]. Note that if the state and action are independent (i.e. $\boldsymbol{\Sigma}_{ux_t}{=}\boldsymbol{0}$) or the state distribution is unchanged ($\overrightarrow{\boldsymbol{x}}_t{=}\boldsymbol{x}_t$), then no 'feedback' is applied and the input distribution remains as the prior $\overrightarrow{\boldsymbol{u}}_t{=}\boldsymbol{u}_t$. This feedback term also exhibits the 'turn-off phenomena' from dual control [33], as the gain is attenuated as the state uncertainty increases [4].

IV. INFERENCE-BASED ADVANTAGES

In this section we introduce algorithmic improvements for I2C and explain existing qualities from the SOC perspective.

A. *Optimizing Expert Linear Gaussian Controllers*

When performing trajectory optimization, considering the open- or closed-loop setting is a crucial distinction. The open-loop approach is a simpler optimization problem, but may be brittle and is limited to shorter planning horizons and levels of stochasticity. Closed-loop optimization provides extra stability due to the local control law, however the numerical procedure has an unfortunate side-effect in the nonlinear setting: The feedback fights exploration during optimization, resulting in highly-actuated solutions that are likely sub-optimal. As I2C derives its controller from the joint state-action distribution, both open- and closed-loop optimization can be incorporated into the E-step by considering the independent ($\boldsymbol{\Sigma}_{\overrightarrow{\boldsymbol{u}\boldsymbol{x}}_t}{=}\boldsymbol{0}$) or full joint distribution respectively.

In Table 1 we compare these variations on two stochasic, nonlinear swing-up tasks, looking at open- and closed-loop optimization for both I2C and baseline SOC solvers, which also vary between considering the actual stochastic problem or a certainty equivalent approximation. For both tasks, open-loop methods resulted in better optima but were also high variance in the cost, while the closed-loop alternatives had much lower variance but sub-optimal performance on the simulated systems due to their over-actuation. Reassuringly, the results of the I2C variant and equivalent baseline solver were generally similar due to the comparable computation.

These results suggest that ideally we should be able to combine the benefits of open- and closed-loop optimization to improve performance during inference. With I2C this is indeed possible, by adjusting the conditional distribution (π_t) to act as an 'expert' controller, which utilizes the probability that the local controller applies to the current state $p(\boldsymbol{x}_t{=}\boldsymbol{x})$,

$$\pi_t(\boldsymbol{u}|\boldsymbol{x}) = p(\boldsymbol{x}_t{=}\boldsymbol{x})\,p(\boldsymbol{u}_t \mid \boldsymbol{x}_t{=}\boldsymbol{x}) + p(\boldsymbol{x}_t{\neq}\boldsymbol{x})\,p(\boldsymbol{u}_t). \tag{17}$$

This weighting softens the controller to fall back to the open-loop controls $p(\boldsymbol{u}_t)$ as $\boldsymbol{x}$ moves sufficiently far from estimated optimal state trajectory. Therefore, the feedback effect is reduced during significant exploration in the E-step, and as the open-loop controls are well regularized this avoids highly-actuated trajectories forming. Table 1 demonstrates the effectiveness of this addition, where this expert controller matches the open-loop optima but with the closed-loop variance reduction. For a Gaussian distribution, $p(\boldsymbol{x}_t{=}\boldsymbol{x})$ is defined as the confidence interval $\|\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}_t}\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{x}_t}} \leq \mathcal{X}^2_k(p)$ for probability p, where $\mathcal{X}^2_k$ is the chi-square distribution. In practice, we also found the stochastic setting required greater care regarding hyperparameter tuning. This is because under greater uncertainty the inaccuracy in the approximate inference is more severe, thus requiring greater regularization during inference. Moreover, from the probabilistic numerics view, greater uncertainty corresponds to greater numerical regularization e.g. the turn-off phenomena in the feedback gains, which may also be at play. Studying this phenomena is greater detail, especially with more accurate approximate inference techniques, will be a topic of future work.

B. *Adaptive Risk-Sensitivity*

For Gaussian I2C, the expected likelihood in Equation 7 may be reformulated to expose the quadratic cost function from the observation likelihood,

$$\mathbb{E}\left[A(\boldsymbol{X},\boldsymbol{U},\alpha)\exp\left(-\frac{\alpha}{2}\left[C_T(\boldsymbol{x}_T) + \sum_{t=0}^{T\text{-}1} C_t(\boldsymbol{x}_t,\boldsymbol{u}_t)\right]\right)\right] \tag{18}$$

where $A(.)$ contains the remaining likelihood and normalizing terms. Comparing this form to Equation 4, α may be interpreted as equivalent to risk sensitivity [27]. For both LEQG and I2C, $\sigma{=}\alpha{=}0$ corresponds to the deterministic LQR result. However, for I2C, $\alpha > 0$ due to the probabilistic treatment, resulting in only risk-seeking behavior possible, with risk neural control as the limit of $\alpha{\to}0$. Risk-seeking behavior is an inherent issue for control-as-inference methods, due to the probabilistic formulation naturally attenuating the effect of unlikely trajectories in the objective. Related methods such as GPS and path integral control also share this property. However, due to the other terms at play in I2C, α and σ do not have a direct correspondence, nor do their resultant trajectories exactly match. This discrepancy can be identified by examining the linear Gaussian message passing equations. First note in the deterministic setting (without $\boldsymbol{\eta}_t$),

$$\boldsymbol{\Lambda}_{\boldsymbol{x}_t''} = \boldsymbol{\Lambda}_{\boldsymbol{x}_{t+1}}, \text{ where } \boldsymbol{x}_t'' \text{ denotes the state after } \boldsymbol{f}. \quad (19)$$

Moreover in the LQR equivalent case, where there are sufficiently uninformative priors placed on $\boldsymbol{U}_t$,

$$\boldsymbol{\Lambda}_{\boldsymbol{x}_{t+1}} = \boldsymbol{\Lambda}_{\overrightarrow{\boldsymbol{x}}_{t+1}} + \boldsymbol{\Lambda}_{\overleftarrow{\boldsymbol{x}}_{t+1}} \approx \boldsymbol{\Lambda}_{\overleftarrow{\boldsymbol{x}}_{t+1}} \text{ for large } \boldsymbol{\Sigma}_{\overrightarrow{\boldsymbol{x}}_{t+1}}. \quad (20)$$

This provides an inference-based perspective on why LQR can be solved without considering the forward propagation. However, considering disturbance $\boldsymbol{\eta}_t$,

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_{t+1}} = \boldsymbol{\Sigma}_{\boldsymbol{x}_t''} + \boldsymbol{\Sigma}_{\boldsymbol{\eta}_t}, \quad \boldsymbol{\Lambda}_{\boldsymbol{x}_{t+1}} = (\boldsymbol{\Lambda}^{-1}_{\boldsymbol{x}_t''} + \boldsymbol{\Sigma}_{\boldsymbol{\eta}_t})^{-1}, \quad (21)$$

$$\boldsymbol{\Lambda}^{-1}_{\boldsymbol{x}_t''} = (\boldsymbol{\Lambda}_{\overrightarrow{\boldsymbol{x}}_t''} + \boldsymbol{\Lambda}_{\overleftarrow{\boldsymbol{x}}_t''})^{-1}, \quad (22)$$

$$= \boldsymbol{\Lambda}^{-1}_{\overleftarrow{\boldsymbol{x}}_t''} - \boldsymbol{\Lambda}^{-1}_{\overleftarrow{\boldsymbol{x}}_t''}(\boldsymbol{\Lambda}^{-1}_{\overrightarrow{\boldsymbol{x}}_t''} + \boldsymbol{\Lambda}^{-1}_{\overleftarrow{\boldsymbol{x}}_t''})^{-1}\boldsymbol{\Lambda}^{-1}_{\overleftarrow{\boldsymbol{x}}_t''}, \quad (23)$$

$$\boldsymbol{\Lambda}_{\boldsymbol{x}_{t+1}} \approx (\boldsymbol{\Lambda}^{-1}_{\overleftarrow{\boldsymbol{x}}_t''} + \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\eta}_t})^{-1}. \quad (24)$$

Therefore, LEQG's risk sensitivity may be viewed as a means to approximate the prior state distribution while using LQR's Riccati equation, through the transformed disturbance term $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\eta}_t} = \sigma \boldsymbol{\Sigma}_{\boldsymbol{\eta}_t}$. Moreover, we can see that setting $\sigma > 0$ (i.e. risk avoidance) acts to reduce the uncertainty in the trajectory in a way that probabilistically invalid, as $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\eta}_t}$ must be positive definite.

While risk-seeking behavior is typically seldom desired, α's influence on both exploration and risk suggests that risk-seeking control is useful (for nonlinear systems) by greedily seeking the desired trajectory. Moreover, further consideration of this term may improve the robustness of the controller by limiting risky strategies.

C. Covariance Control as Inference

For I2C, the idea of a terminal cost can be tackled in two ways. Previously, we have used the idea of a terminal observation function $\boldsymbol{g}_T(\boldsymbol{x})$, which allows for arbitrary terminal costs, as described in Section III-B. However, another approach is to work with the terminal latent state distribution. In other time-series inference applications, e.g. state estimation, the terminal state posterior is set to the prior as there is no additional information to use. However, for control we can avoid the cost function design and the cost-to-likelihood translation by setting the terminal distribution directly, but are now limited to stipulating the desired state directly rather through than a useful transformation (e.g. $\cos\theta$). This approach is equivalent to covariance control (Section II-B) as, like solving a Schrödinger bridge, iterations of forward and backward Riccati equations are performed until the boundary condition are satisfied. In this setting, $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_T}$ now acts as the Lagrange multiplier. Examining the expected log-likelihood (Equation 12) term for the terminal state ($\mathcal{L}_{\boldsymbol{z}_T}$) for the direct state optimization case $\boldsymbol{z}_T = \boldsymbol{x}_T + \boldsymbol{\xi}_T$, we get

$$\mathcal{L}_{\boldsymbol{z}_T} \propto \|\boldsymbol{\mu}_{\boldsymbol{z}_T} - \boldsymbol{\mu}_{\boldsymbol{x}_T}\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{\xi}}} + \mathrm{tr}\{\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_T}\boldsymbol{\Sigma}_{\boldsymbol{x}_T}\} + \text{const.} \quad (25)$$

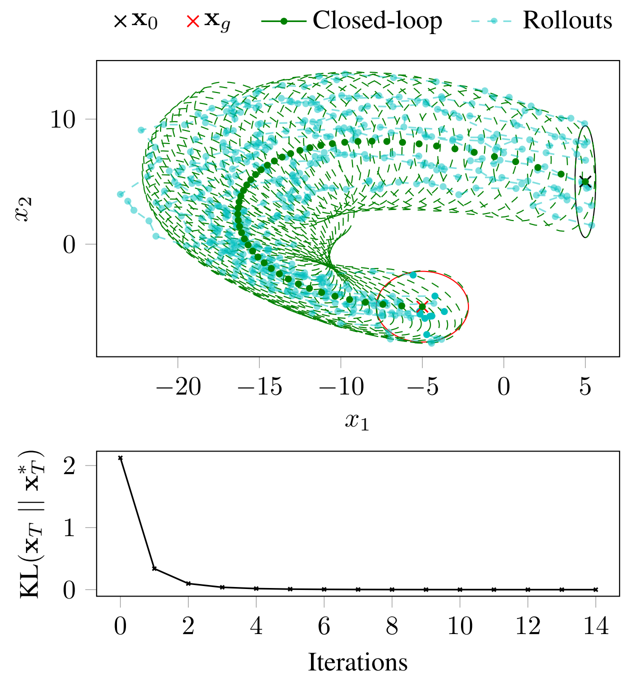


Fig. 2. I2C for exact minimum-energy linear Gaussian covariance control on an unstable system, with a fixed small α and $\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t} = \mathrm{diag}(0.1, 0.1)$. The KL divergence is between the terminal goal and closed-loop distributions.

which is equivalent the LQG correspondence proved in Equation 41 of Goldshtein et al. [15], where $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_T}$ is the terminal cost / Lagrange multiplier matrix. However, rather than compute this term, using our probabilistic framework we can set the terminal distribution directly via the posterior of $\boldsymbol{x}_T$. In this case, the inference iteratively seeks to satisfy the boundary conditions on $\boldsymbol{x}_0$ and $\boldsymbol{x}_T$. Figure 2 demonstrates covariance control on a linear Gaussian system, where inference is exact. It should be noted that I2C uses linear Gaussian controllers, whose uncertainty is required to achieve the desired state distribution, whereas the previous literature solves the task using deterministic control laws. This variation of I2C naturally translates to nonlinear systems (Figure 3), avoiding the complexity of the additional forward sampling required for DDP-based covariance control [18]. However, the terminal boundary constraint still requires a means of being applied in an gradual manner, due to the iterative aspect of the nonlinear optimization. The terminal state distribution can be shifted from the prior to the constraint by 'annealing' [34] the prior during inference

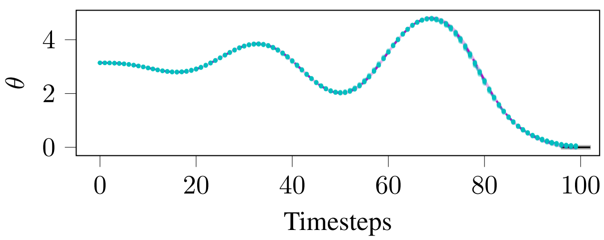
$$p(\boldsymbol{x}_T) = p(\boldsymbol{x}_T^*)\, p(\overrightarrow{\boldsymbol{x}}_T)^{\beta}, \quad \text{for } 1 \geq \beta \geq 0, \quad (26)$$

where β is the annealing temperature on the prior from the forward pass. By decreasing β over iterations, $p(\boldsymbol{x}_T) \to p(\boldsymbol{x}_T^*)$ as $\beta \to 0$. This annealing strategy has previously been applied to regularize inference algorithms.

V. CONCLUSION

In this work, we have discussed control as inference and the I2C algorithm from the perspective of stochastic optimal control. We have analysed I2C with respect to the

Fig. 3. Nonlinear minimum-energy covariance control on the pendulum swing-up task, using I2C with approximate inference. Plot depicts the inferred trajectory - - - for target distribution ——, with simulated rollouts - - .



risk-sensitive control, highlighting the similarities between LEQG, inference, and how I2C utilizes adaptive risk during EM. Moreover, by analyzing the numerical consequences of open- and closed-loop optimization with I2C and baseline solvers, we have motivated an 'expert' linear Gaussian controller that leverages the state belief to balance exploration with stabilizing feedback, which achieves superior results on simulated nonlinear SOC tasks. Finally, we demonstrated how covariance control can be implemented with a minor adjustment to I2C, enabling exact and approximate solutions in the linear and nonlinear setting respectively.

VI. ACKNOWLEDGMENTS

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No. 640554 (SKILLS4ROBOTS) and 713010 (GOAL Robots). The paper was inspired in part by many insightful discussions with Hany Abdulsamad and Boris Belousov.

REFERENCES

[1] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," in *International Conference on Automated Planning and Scheduling*, 2013.

[2] M. Toussaint and A. Storkey, "Probabilistic inference for solving discrete and continuous state Markov Decision Processes," in *International Conference of Machine Learning*, 2006.

[3] P. Hennig, M. A. Osborne, and M. Girolami, "Probabilistic numerics and uncertainty in computations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2015.

[4] J. Watson, H. Abdulsamad, and J. Peters, "Stochastic optimal control as approximate input inference," in *Conference on Robot Learning*, 2019.

[5] D. H. Jacobson and D. Q. Mayne, "Differential dynamic programming," 1970.

[6] E. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming," in *American Control Conference*, 2010.

[7] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *International Conference on Informatics in Control, Automation and Robotics*, 2004.

[8] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*, 2005.

[9] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, 2013.

[10] A. E. Bryson, *Applied optimal control: Optimization, estimation and control*. Routledge, 2018.

[11] M. Rafieisakhaei, S. Chakravorty, and P. R. Kumar, "T-lqg: Closed-loop belief space planning via trajectory-optimized lqg," in *IEEE International Conference on Robotics and Automation*, 2017.

[12] K. S. Parunandi and S. Chakravorty, "T-pfc: A trajectory-optimized perturbation feedback control approach," *IEEE Robotics and Automation Letters*, 2019.

[13] W. H. Fleming, "Stochastic control for small noise intensities," *SIAM Journal on Control*, 1971.

[14] A. F. Hotz and R. E. Skelton, "A covariance control theory," in *IEEE Conference on Decision and Control*, 1985.

[15] M. Goldshtein and P. Tsiotras, "Finite-horizon covariance control of linear time-varying systems," in *IEEE Conference on Decision and Control*, 2017.

[16] Y. Chen, T. T. Georgiou, and M. Pavon, "Optimal steering of a linear stochastic system to a final probability distribution, part i," *IEEE Transactions on Automatic Control*, 2016.

[17] A. Beghi, "On the relative entropy of discrete-time markov processes with given end-point densities," *IEEE Transactions on Information Theory*, 1996.

[18] Z. Yi, Z. Cao, E. Theodorou, and Y. Chen, "Nonlinear covariance control via differential dynamic programming," in *American Control Conference*, 2020.

[19] E. Bakolas and A. Tsolovikos, "Greedy finite-horizon covariance steering for discrete-time stochastic nonlinear systems based on the unscented transform," in *American Control Conference*, 2020.

[20] Y. Chen, "Modeling and control of collective dynamics: From schrödinger bridges to optimal mass transport," Ph.D. dissertation, University of Minnesota, 2016.

[21] D. Jacobson, "Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games," *IEEE Transactions on Automatic Control*, 1973.

[22] P. Whittle, "Risk-sensitive linear/quadratic/gaussian control," *Advances in Applied Probability*, 1981.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, 1960.

[24] J. Šindelář, I. Vajda, and M. Kárnỳ, "Stochastic control optimal in the kullback sense," *Kybernetika*, 2008.

[25] M. Toussaint and A. Storkey, "Probabilistic inference for solving discrete and continuous state markov decision processes," in *International Conference on Machine learning*, 2006.

[26] C. Hoffmann and P. Rostalski, "Linear optimal control on factor graphs - a message passing perspective -," *International Federation of Automatic Control*, 2017.

[27] K. Rawlik, M. Toussaint, and S. Vijayakumar, "On stochastic optimal control and reinforcement learning by approximate inference," in *International Joint Conference on Artificial Intelligence*, 2013.

[28] K. C. Rawlik, "On probabilistic inference approaches to stochastic optimal control," Ph.D. dissertation, The University of Edinburgh, 2013.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.

[30] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Tech. Rep., 1996.

[31] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, 2007.

[32] B. D. Anderson and J. B. Moore, *Optimal filtering*, 2012.

[33] Y. Bar-Shalom and E. Tse, "Dual effect, certainty equivalence, and separation in stochastic control," *IEEE Transactions on Automatic Control*, 1974.

[34] N. Ueda and R. Nakano, "Deterministic annealing variant of the em algorithm," in *Neural Information Processing Systems*, 1994.

[35] A. Solin, "Cubature integration methods in non-linear Kalman filteringand smoothing," Bachelor's thesis, Aalto University, 2010.

APPENDIX

Experimental Details

For the experiment reported in Table 1, task parameters are similar to prior work [4], but with the Cartpole environment now operating at a 4ms timestep for a time horizon of 500. For I2C, cubature quadrature was used for approximate inference [35]. T-PFC was implemented using the transcription method (i.e. $(T{+}1)d_x{+}Td_u$ state space and $T{+}1$ constraints) .