Task-Driven Estimation and Control via Information Bottlenecks

Vincent Pacelli and Anirudha Majumdar

***Abstract*— Our goal is to develop a principled and general algorithmic framework for *task-driven estimation and control* for robotic systems. State-of-the-art approaches for controlling robotic systems typically rely heavily on accurately estimating the full state of the robot (e.g., a running robot might estimate joint angles and velocities, torso state, and position relative to a goal). However, full state representations are often excessively rich for the specific task at hand and can lead to significant computational inefficiency and brittleness to errors in state estimation. In contrast, we present an approach that eschews such rich representations and seeks to create *task-driven representations*. The key technical insight is to leverage the theory of *information bottlenecks* to formalize the notion of a "task-driven representation" in terms of information theoretic quantities that measure the *minimality* of a representation. We propose novel iterative algorithms for automatically synthesizing (offline) a task-driven representation (given in terms of a set of task-relevant variables (TRVs)) and a performant control policy that is a function of the TRVs. We present *online* algorithms for estimating the TRVs in order to apply the control policy. We demonstrate that our approach results in significant robustness to *unmodeled* measurement uncertainty both theoretically and via thorough simulation experiments including a spring-loaded inverted pendulum running to a goal location.**

I. Introduction

State-of-the-art techniques for controlling robotic systems typically rely heavily on accurately estimating the full state of the system and maintaining rich geometric representations of their environment. For example, a common approach to navigation is to build a dense occupancy map produced by scanning the environment and to use this map for planning and control. Similarly, control for walking or running robots typically involves estimating the full state of the robot (e.g., joint angles, velocities, and position relative to a goal location). However, such representations are often overly detailed when compared to a *task-driven representation*.

One example of a task-driven representation is the "gaze heuristic" from cognitive psychology [1], [2], [3]. When attempting to catch a ball, an agent can estimate the ball's position and velocity, model how it will evolve in conjunction with environmental factors like wind, integrate the pertinent differential equations, and plan a trajectory in order to arrive at the ball's final location. In contrast, cognitive psychology studies have shown that humans use a dramatically simpler strategy that entails maintaining the angle that the human's gaze makes with the ball at a constant value. This method reduces a number of hard-to-monitor variables (e.g., wind speed) into a single easily-estimated variable. Modulating this variable alone results in accomplishing the task.

The gaze heuristic example highlights the two primary advantages of using a task-driven representation. First, a control policy that uses such a representation is more efficient to employ online since fewer variables need to be estimated. Second, since only a few prominent variables need to be estimated, fewer sources of measurement uncertainty result in a more robust policy. While one can sometimes manually design task-driven representations for a given task, we currently lack a principled theoretical and algorithmic framework for *automatically synthesizing* such representations. The goal of this paper is to develop precisely such an algorithmic approach.

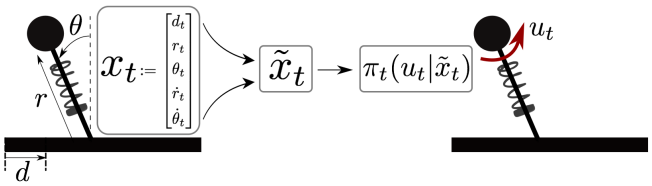

Fig. 1. A schematic of our technical approach. We seek to synthesize (offline) a minimalistic set of task-relevant variables (TRVs) $\tilde{x}_t$ that create a bottleneck between the full state x_t and the control input u_t. These TRVs are estimated online in order to apply the policy π_t. We demonstrate our approach on a spring-loaded inverted pendulum model whose goal is to run to a target location. Our approach automatically synthesizes a *one-dimensional* TRV $\tilde{x}_t$ sufficient for achieving this task.

Statement of Contributions. The main technical contribution of this paper is to formulate the synthesis of task-driven representations as an optimization problem using information bottleneck theory [4]. We present *offline* algorithms that encode the full state of the system into a set of *task-relevant variables (TRVs)* and simultaneously identify a performant policy (restricted to be a function of the TRVs) using novel iterative algorithms that exploit the structure of this optimization problem in a number of dynamical settings including discrete-state, linear-Gaussian, and nonlinear systems. We present *online* algorithms for estimating the TRVs in order to apply the control policy. We demonstrate that our approach yields policies that are robust to *unmodeled* measurement uncertainty both theoretically (using results from the theory of risk metrics) and in a number of simulation experiments including running using a spring-loaded inverted pendulum (SLIP) model (Fig.1).

A. Related Work

By far the most common approach in practice for controlling robotic systems with nonlinear dynamics and partially observable state is to *independently* design an estimator for the *full* state (e.g., a Kalman filter [5]) and a controller that assumes perfect state information (e.g., designed using robust control techniques such as H_∞ control [6], sliding

. This work was supported by the National Science Foundation (NSF) [IIS-1755038] and a Google Faculty Research Award.

. The authors are with the Mechanical and Aerospace Engineering department at Princeton University, NJ, 08540, USA {vpacelli, ani.majumdar}@princeton.edu

mode control [7], [8], passivity-based control [9], [10], or Lyapunov-based control [8], [10]). While this strategy is optimal for Linear-Quadratic-Gaussian (LQG) problems due to the *separation principle* [11], it can produce a brittle system due to errors in the state estimate for nonlinear systems (since the separation principle does not generally hold in this setting). We demonstrate that our task-driven approach affords significant robustness when compared to approaches that perform full state estimation and control assuming the separation principle (see Section V for numerical examples). Moreover, in contrast to traditional robust estimation and control techniques, our approach does not rely on explicit models of measurement uncertainty. We demonstrate that robustness can be achieved implicitly as a *by-product* of task-driven representations.

Historically, the work on designing *information constrained* controllers has been pursued within the networked control theory literature [12], [13], [14]. Recently, the optimal co-design of data-efficient sensors and performant controllers has also been explored beyond network applications. One set of approaches — inspired by the cognitive psychology concept of *bounded rationality* [15], [16] — is to limit the information content of a control policy measured with respect to a default stochastic policy [17], [18], [19], [20], [21]. Another set of examples comes from the sensor selection problem in robotics, which involves selecting a minimal number of sensors or features to use for estimating the robot's state [22], [23], [24]. While the work highlighted above shares our goal of designing "minimalistic" (i.e., informationally frugal) controllers, our approach here is fundamentally different. In particular, our goal is to design task-driven *representations* that form abstractions which are sufficient for the purpose of control. Online estimation and control is performed purely based on this representation without resorting to estimating the full state of the system (in contrast to the work highlighted above, which either assumes access to the full state or designs estimators for it).

A number of previous authors consider the construction of minimal-information representations. In information theory, this approach is typically referred to as the Information Bottleneck (IB) [4]. Recently, these ideas have been co-opted for designing control policies. In [25], a learning-based approach is suggested to find minimal-information state representations and control policies which use them. Our work differs in that we provide *analytic* (i.e., model-based) methods for finding such representations and policies and we explicitly characterize the resulting robustness. Another branch of work considers the construction of LQG policies that achieve a performance goal while minimizing an information-theoretic quantity such as the mutual information between inputs and outputs [26], [27] or Massey's directed information [28], [29]. In contrast to these works, our derivation handles nonlinear systems and also presents robustness results for the resulting controllers, which have not been discussed to our knowledge in existing literature.

The work on *actionable information* in vision [30], [31] attempts to find invariant and task-relevant representations for visual tasks in the form of *minimal complete representations* (minimal sufficient statistics of a full or "complete" representation). While highly ambitious in scope, this work has largely been devoted to studying visual decision tasks (e.g., recognition, detection, categorization). The algorithmic approach taken in [30], [31] is thus tied to the specifics of visual problems (e.g., designing visual feature detectors that are invariant to nuisance factors such as contrast, scale, and translation). Our goals and technical approach here are complementary. We do not specifically target visual decision problems; instead, we seek to develop a general framework that is applicable to a broad range of robotic control problems and allows us to *automatically* synthesize task-driven representations (without manually designing feature detectors).

II. Task-Driven Representations and Controllers

Our goal in this section is to formally define the notion of a "task-driven representation" and formulate an optimization problem that automatically synthesizes such representations. We focus on control tasks in this paper and assume that a task is defined in terms of a (partially observable) Markov decision process ((PO)MDP). Let $x_t \in \mathcal{X},\ u_t \in \mathcal{U},\ y_t \in \mathcal{Y}$ denote the full state of the system, the control input, and the sensor output at time t respectively. Here the state space $\mathcal{X}$, input space $\mathcal{U}$, and output space $\mathcal{Y}$ may either be discrete or continuous. We assume that the dynamics and sensor model of the system are known and given by potentially time-varying conditional distributions $p_t(x_{t+1}|u_t, x_t),\ \sigma_t(y_t|x_t)$ respectively. Let $c_0(x_0, u_0), c_1(x_1, u_1), \ldots, c_{T-1}(x_{T-1}, u_{T-1}), c_T(x_T)$ be a sequence of cost functions that encode the robot's desired behavior. The robot's goal is to identify a control policy $\pi_t(u_t|y_t)$ that minimizes the expected value of these cost functions when the policy is executed *online*, i.e. minimize $\sum_{t=0}^{T} \mathbb{E}_{p_t} c_t(x_t, u_t) + \mathbb{E}_{p_T} c_T(x_T)$. In general, this optimization problem is infinite dimensional and challenging to solve.

The key idea behind our technical approach is to define a principled information-theoretic notion of *minimality* of representations for tasks. Figure 1 illustrates the main idea for doing this. Here $\tilde{x}_t \in \tilde{\mathcal{X}}$ are *task-relevant variables* (TRVs) that constitute a *representation*; one can think of the representation as a "sketch" of the full state that the robot uses for computing control inputs via $\pi_t(u_t|\tilde{x}_t)$. Ideally, such a representation filters out all the information from the state that is not relevant for computing control inputs — avoiding the introduction of unnecessary estimation error. A minimalistic representation $\tilde{x}_t$ should thus create a *bottleneck* between the full state and the control input. We make this notion precise by leveraging the theory of information bottlenecks [4] and finding a stochastic mapping $q_t(\tilde{x}_t|x_t)$ that minimizes the *mutual information* between x_t and $\tilde{x}_t$

$$\mathbb{I}(x_t; \tilde{x}_t) \coloneqq \mathbb{D}(p_t(x_t, \tilde{x}_t) \| p_t(x_t) p_t(\tilde{x}_t)), \tag{1}$$

while ensuring the policy performs well (i.e., achieves low expected cost). Here, $\mathbb{D}(\cdot\|\cdot)$ represents the Kullback-Leibler (KL) divergence between two distributions. Intuitively, minimizing the mutual information corresponds to designing

TRVs $\tilde{x}_t$ that are as independent of the state x as possible. The map $q_t(\tilde{x}_t|x_t)$ is thus "squeezing out" all the irrelevant information from the state while maintaining enough information for choosing good control inputs. This representation formalizes our notion of a *task-driven representation*.

Formally, we pose the problem of finding task-driven representations as the following *offline* optimization problem, which we refer[1] to as $\mathcal{OPT}$:

$$\min_{p_t, q_t, \pi_t} \quad \sum_{t=0}^{T} \Big[\mathbb{E}_{p_t} c_t(x_t, u_t) + \frac{1}{\beta} \mathbb{I}(x_t; \tilde{x}_t) \Big]. \quad (\mathcal{OPT})$$

We note that the unconstrained problem $\mathcal{OPT}$ is equivalent to a constrained version where the mutual information is minimized subject to a constraint on the expected cost of the full-state MDP and β is the corresponding Lagrange multiplier. A heuristic approach for choosing an appropriate value for β is discussed in Section V.

So far, we have limited our discussion to the case where the robot has access to the full state of the system. However, the real benefit of the task-driven perspective is evident in the partially observable setting, where the robot only indirectly observes the state of the system via sensor measurements $y_0, \ldots, y_t$, where $y_t \in \mathcal{Y}_t$. We denote the probability of observing a particular measurement in a given state by $\sigma_t(y_t|x_t)$. The prevalent approach for handling such settings is to design an estimator for the full state x_t. Our key idea here is to perform estimation for the TRVs $\tilde{x}_t$ instead of x_t. Specifically, our overall approach has two phases:

Offline. Synthesize the maps $\{q_t, \pi_t\}_{t=0}^{T-1}$ by solving $\mathcal{OPT}$.

Online. Estimate current TRV $\tilde{x}_t$ using sensor measurements and use this estimate to compute inputs via $\pi_t(u_t|\tilde{x}_t)$.

Perhaps the clearest benefit of our approach is the fact that the representation $\tilde{x}_t$ may be significantly lower-dimensional than the full state of the system; this can lead to significant reductions in online computations. Another advantage of the task-driven approach is robustness to estimation errors. To see this, let $p_t(x_t, \tilde{x}_t, u_t)$ denote the joint distribution over the state, representation, and inputs at time t that results when we apply the policy obtained by solving Problem $\mathcal{OPT}$ in the fully observable setting. Now, let $\tilde{p}_t(x_t, \tilde{x}_t, u_t)$ denote the distribution for the partially observable setting, i.e., when we estimate $\tilde{x}_t$ online using the robot's sensor measurements and use this estimate to compute control inputs.

Theorem 2.1: Let $\tilde{p}_t(x_t, \tilde{x}_t, u_t)$ be the distribution resulting from any estimator that satisfies the following condition:

$$\begin{aligned} &\mathbb{D}(\tilde{p}_t(x_t, \tilde{x}_t, u_t) \| p_t(x_t, \tilde{x}_t, u_t)) \quad (2) \\ &\leq \frac{1}{\beta} \mathbb{D}(p_t(x_t, \tilde{x}_t) \| p_t(x_t) p_t(\tilde{x}_t)). \end{aligned}$$

Then, we have the following upper bound on the total expected cost:

$$\sum_{t=0}^{T} \mathbb{E}_{\tilde{p}_t} c_t(x_t, u_t) \leq \sum_{t=0}^{T} \Big[\rho\big(c_t(x_t, u_t)\big) + \frac{1}{\beta} \mathbb{I}(x_t; \tilde{x}_t) \Big], \quad (3)$$

where ρ is the *entropic risk metric* [32, Example 6.20]:

$$\rho\big(c_t(x_t, u_t)\big) \coloneqq \log \Big[\mathbb{E}_{p_t} \exp(c_t(x_t, u_t)) \Big]. \quad (4)$$

Proof: The proof follows from the Donsker-Varadhan change of measure formula [33, Theorem 2.3.2] and is presented in the extended version of this paper [34]. ■

Intuitively, this theorem shows that *any* estimator for $\tilde{x}_t$ (in the partially observable setting) that results in a distribution $\tilde{p}_t(x_t, \tilde{x}_t, u_t)$ that is "close enough" to the distribution $p_t(x_t, \tilde{x}_t, u_t)$ in the fully observable case (i.e., when their KL divergence is less than $\frac{1}{\beta}$ times the KL divergence between $p_t(x_t, \tilde{x}_t)$ and the joint distribution $p_t(x_t)p_t(\tilde{x}_t)$ over x_t and $\tilde{x}_t$ that results when $\tilde{x}_t$ is assumed to be *independent* of x_t), the expected cost of the controller in the partially observable case is *guaranteed* to be bounded by the right hand side (RHS) of (3). Notice that this RHS is similar to the cost function of $\mathcal{OPT}$. In particular, the expected value operator is a *linearization* of the entropic risk metric[2] ρ. By solving $\mathcal{OPT}$, we are minimizing (a linear approximation of) an upper bound on the expected cost even when our state is only partially observable (as long as our estimator for $\tilde{x}_t$ ensures condition (2)). Once $\mathcal{OPT}$ is solved, we can use Theorem 2.1 to obtain a robustness bound by evaluating the RHS of (3).

III. Algorithms for Synthesizing Representations

In this section, we outline our approach for solving $\mathcal{OPT}$ offline. We note that $\mathcal{OPT}$ is non-convex in general. While one could potentially apply general non-convex optimization techniques such as gradient-based methods, computing gradients quickly becomes computationally expensive due to the large number of decision variables involved (even in the setting with finite state and action spaces, we have decision variables corresponding to $q_t(\tilde{x}_t|x_t)$ and $\pi_t(u_t|\tilde{x}_t)$ for every possible value of $x_t, \tilde{x}_t$ and u_t at every time step). Our key insight here is to exploit the structure of $\mathcal{OPT}$ to propose an efficient iterative algorithm in three different dynamical settings: discrete, linear-Gaussian, and nonlinear-Gaussian. These settings are particularly convenient to work with because they allow the objective of $\mathcal{OPT}$ to be *computed in closed-form*. In each setting, the algorithm iterates over the following three steps:

1) Fix $\{q_t, \pi_t\}_{t=0}^{T-1}$ and solve for $\{p_t\}_{t=0}^{T}$ using the forward dynamical equations.
2) Fix $\{p_t\}_{t=0}^{T}, \{\pi_t\}_{t=0}^{T-1}$ and solve for $\{q_t\}_{t=0}^{T-1}$ by satisfying necessary conditions for optimality.
3) Fix $\{p_t\}_{t=0}^{T}, \{q_t\}_{t=0}^{T-1}$ and solve for $\{\pi_t\}_{t=0}^{T-1}$ by solving a convex optimization problem.

In our implementation, we iterate over these steps until convergence (or until an iteration limit is reached). This is a common strategy employed in solving similar kinds of MDPs with information-theoretic objectives [26], [38]. While we cannot currently guarantee convergence, our iterative

1. Technically, $\mathcal{OPT}$ is a kind of rate-distortion problem, not an information bottleneck problem, as the constraint is not specified using a divergence as a distortion function. However, $q_t(\tilde{x}_t|x_t)$ limits the flow of information from x_t to u_t so we use the term bottleneck as a conceptual aid.

2. This risk metric has a long history in robust control (including a close link to H_∞ control) [35], [36], [37]. We note, however, that it can sometimes be conservative in cases with rare, bad events.

procedure is extremely efficient (since all the computations above can be performed either in closed-form or via a convex optimization problem) and produces good solutions in practice (see Section V). We describe instantiations of each step for three different dynamical settings below.

A. Discrete Systems

In order to solve Step 1, note that the forward dynamics of the system for fixed $q_t(\tilde{x}_t|x_t), \pi_t(u_t|\tilde{x}_t)$ are given by:

$$p_t(x_{t+1}|x_t) = \sum_{u,\tilde{x}} p_t(x_{t+1}|x_t,u)\pi_t(u|\tilde{x})q_t(\tilde{x}|x_t),$$
$$p_{t+1}(x_{t+1}) = \sum_{x} p_t(x_{t+1}|x)p_t(x). \tag{5}$$

The Lagrangian functional for $\mathcal{OPT}$ is $\mathcal{L} = \sum_{t=0}^{T} \mathcal{L}_t$ where

$$\begin{aligned}\mathcal{L}_t = &\sum_{\tilde{x},u,x} c_t(x,u)\pi_t(u|\tilde{x})q_t(\tilde{x}|x)p_t(x)\\ &- \sum_{x'} \nu_{t+1}(x')\left(p_{t+1}(x') - \sum_{x,u,\tilde{x}} p_t(x'|x,u)\pi_t(u|\tilde{x})q_t(\tilde{x}|x)p_t(x)\right)\\ &+ \frac{1}{\beta}\sum_{x,\tilde{x}} q_t(\tilde{x}|x)p_t(x)\log\left(\frac{q_t(\tilde{x}|x)}{q_t(\tilde{x})}\right).\end{aligned}$$

where $\nu_t(x_t)$ are Lagrange multipliers. The Lagrange multipliers that normalize distribution variables are omitted since they do not contribute to the analysis. The following proposition demonstrates the structure of $q_t(\tilde{x}_t|x_t)$ using the first-order necessary condition (FONC) for optimality [39].

Theorem 3.1: A necessary condition for $q_t(\tilde{x}|x)$ to be optimal for $\mathcal{OPT}$ is that

$$q_t(\tilde{x}_t|x_t) = \frac{q_t(\tilde{x}_t)\exp\left(-\beta\mathbb{E}(\nu_{t+1} + c_t|x_t,\tilde{x}_t)\right)}{Z_t(x_t)}, \tag{6}$$

where $\nu_T(x_T) = c_T(x_T)$ and

$$\nu_t(x_t) = \mathbb{E}(c_t + \nu_{t+1}|x_t) + \frac{1}{\beta}\mathbb{D}\left(q_t(\tilde{x}_t|x_t)\|q_t(\tilde{x}_t)\right), \tag{7}$$

$$Z_t(x_t) = \sum_{\tilde{x}} q_t(\tilde{x})\exp\left(-\beta[\mathbb{E}(c_t + \nu_{t+1}|x_t,\tilde{x}))]\right). \tag{8}$$

Proof: Equation (7) is derived by setting the functional derivative $\delta\mathcal{L}/\delta p_t(x_t) = 0$ and solving for ν_t. Repeating this process for $\delta\mathcal{L}/\delta q_t(x_t|\tilde{x}_t)$ yields (6). Equation (8) is the normalization of $q_t(\tilde{x}_t|x_t)$. Proof is provided in [34]. ■

This proposition demonstrates that $q_t(\tilde{x}_t|x_t)$ is a *Boltzmann distribution* with $Z_t(x_t)$ as the *partition function* and β playing the role of inverse temperature. In order to solve Step 2, we simply evaluate the closed-form expression (6).

It is easily verified that the function $\nu_t(x_t)$ is the *cost-to-go function* for $\mathcal{OPT}$. Thus $\mathcal{OPT}$ can be written as a dynamic programming problem using $\nu_t(x_t)$:

$$\min_{q_t,\pi_t} \quad \mathbb{E}_{p_t}\left(c_t + \nu_{t+1} + \frac{1}{\beta}\mathbb{D}\big(q_t(\tilde{x}_t|x_t)||q_t(\tilde{x}_t)\big)\right). \tag{$\mathcal{DP}$}$$

This allows us to solve Step 3. In particular, when $p_t(x_t), q_t(\tilde{x}_t|x_t)$ are fixed, $\mathcal{DP}$ is a linear programming problem in π_t and can thus be solved efficiently.

B. Linear-Gaussian Systems with Quadratic Costs

A discrete-time linear-Gaussian (LG) system is defined by the transition system

$$x_{t+1} = A_t x_t + B_t u_t + \epsilon_t, \quad \epsilon_t \sim N(0,\Sigma_{\epsilon_t}), \tag{9}$$

where $\mathcal{X} = \mathbb{R}^n$, $\mathcal{U} = \mathbb{R}^m$ and $x_0 \sim N(\bar{x}_0, \Sigma_{x_0})$. We assume that the cost function is quadratic:

$$c_t(x,u) \coloneqq \frac{1}{2}(x-g_t)^{\mathrm{T}}Q_t(x-g_t) + \frac{1}{2}(u-w_t)^{\mathrm{T}}R_t(u-w_t),$$

with $Q_t,\ R_t \succeq 0, R_T = 0$. We explicitly parameterize the TRVs and control policy as:

$$\tilde{x}_t = C_t x_t + a_t + \eta_t, \quad u_t = K_t\tilde{x}_t + h_t, \tag{10}$$

where the random variable $\eta_t \sim N(0,\Sigma_{\eta_t})$ is additive process noise. This structure dictates that $p_t(x_t),\ q_t(\tilde{x}_t|x_t)$ are Gaussians $N(\bar{x}_t,\Sigma_{x_t}),\ N(\bar{\tilde{x}}_t,\Sigma_{\tilde{x}_t})$ respectively, with $\bar{\tilde{x}}_t = C_t\bar{x}_t + a_t,\ \Sigma_{\tilde{x}_t} = C_t\Sigma_{x_t}C_t^{\mathrm{T}} + \Sigma_{\eta_t}$. This allows for both Steps 1 and 2 to be computed in closed form. The latter is presented in the following theorem.

Theorem 3.2: Define the notational shorthand $G_t \coloneqq {C_t}^{\mathrm{T}}(C_t\Sigma_{x_t}C_t^{\mathrm{T}} + \Sigma_{\eta_t})^{-1}C_t,\ M_t \coloneqq (A_t + B_tK_tC_t)$. For the LG system, the necessary condition (6) is equivalent to the conditions

$$\begin{aligned}C_t =& -\beta\Sigma_{\eta_t}K_t^{\mathrm{T}}B_t^{\mathrm{T}}P_{t+1}A_t,\\ a_t =& -\Sigma_{\eta_t}\big(\beta K_t^{\mathrm{T}}B_t^{\mathrm{T}}(b_{t+1} + P_{t+1}B_th_t)\\ &+ \beta K_t^{\mathrm{T}}R_t(h_t - w_t) - \Sigma_{\tilde{x}_t}^{-1}\bar{\tilde{x}}_t\big),\\ \Sigma_{\eta_t}^{-1} =& \Sigma_{\tilde{x}_t}^{-1} + \beta K_t^{\mathrm{T}}(B_t^{\mathrm{T}}P_{t+1}B_t + R_t)K_t,\end{aligned} \tag{11}$$

where the cost-to-go function is the recursively defined quadratic function $\nu_t(x) = \frac{1}{2}x_t^{\mathrm{T}}P_tx_t + {b_t}^{\mathrm{T}}x_t + \text{constant}$ with values $P_T = Q_{T+1},\ b_T = -Q_Tg_T$ and

$$\begin{aligned}P_t =& Q_t + \beta^{-1}G_t + C_t^{\mathrm{T}}K_t^{\mathrm{T}}R_tK_tC_t + {M_t}^{\mathrm{T}}P_{t+1}M_t,\\ b_t =& {M_t}^{\mathrm{T}}P_{t+1}B_t(h_t + K_ta_t) - Q_tg_t - \beta^{-1}G_t\bar{x}_t\\ &+ C_t^{\mathrm{T}}K_t^{\mathrm{T}}R_t(K_ta_t + h_t - w_t) + {M_t}^{\mathrm{T}}b_{t+1}.\end{aligned} \tag{12}$$

Proof: The KL-divergence term in (7) is quadratic: $\mathbb{D}\left(q(\tilde{x}|x_t)||q_t(\tilde{x})\right) = \frac{1}{2}(\bar{x}_t - x_t)^{\mathrm{T}}G_t(\bar{x}_t - x_t)$ up to a constant. Since c_t is quadratic in x_t, the form for $\nu_t(x_t)$ is derived by backward induction starting with $\nu_T(x_T) = \mathbb{E}(c_T)$. The forms for $C_t, a_t, \Sigma_{\eta_t}$ are derived by taking the logarithm of both sides of (6), plugging (12) in for $\nu_t(x_t)$, completing the square, and exponentiating both sides to produce a Gaussian. The constant term in (12) is collected into $Z_t(x_t)$. A complete proof can be found in [34]. ■

Finally, when q_t, p_t are fixed, $\mathcal{DP}$ is the unconstrained convex quadratic program with decision variables K_t, h_t, and can be solved very efficiently [40].

C. Nonlinear-Gaussian Systems

When the dynamics are nonlinear-Gaussian (NLG), i.e. when (9) is changed to

$$x_{t+1} = f(x_t,u_t) + \epsilon_t, \quad \epsilon_t \sim N(0,\Sigma_{\epsilon_t}), \tag{13}$$

minimizing $\mathcal{OPT}$ is challenging due to $p_t(x_t)$ no longer being Gaussian. We tackle this challenge by leveraging our results

for the LG setting and adapting the iterative Linear Quadratic Regulator (iLQR) algorithm [41], [42], [43].

Given an initial nominal trajectory $\{\hat{x}_t, \hat{u}_t\}_{t=0}^{T}$, the matrices $\{A_t, B_t\}_{t=0}^{T-1}$ are produced by linearizing $f(x_t, u_t)$ along the trajectory. The pair (A_t, B_t) describes the dynamics of a perturbation $\delta x_t = x_t - \hat{x}_t$ in the neighborhood of x_t for a perturbed input $\delta u_t = u_t - \hat{u}_t$ in the neighborhood of u_t:

$$\delta x_{t+1} = A_t \delta x_t + B_t \delta u_t + \epsilon_t, \quad \delta x_0 \sim N(0, \Sigma_{x_t}). \quad (14)$$

We compute (a quadratic approximation of) the perturbation costs $\delta c_t(\delta x, \delta u) := c_t(\hat{x}_t + \delta x, \hat{u}_t + \delta u)$ subject to (14). We can then apply the solution method outlined in Section III-B to search for an optimal $\{\delta x_t, \delta u_t\}_{t=0}^{T}$. We then update the nominal state and input trajectories to $\{\hat{x}_t + \delta x_t, \hat{u}_t + \delta u_t\}_{t=0}^{T}$ and repeat the entire process until the nominal trajectory converges.

IV. Online Estimation and Control

Once the task-driven representation and policy have been synthesized offline, we can leverage them for computationally-efficient and robust online control. The key idea behind our online approach is to use the robot's sensor measurements $\{y_i\}_{i=1}^{t}$ to *only* estimate the TRVs $\tilde{x}_t$. Once $\tilde{x}_t$ has been estimated, the control policy $\pi_t(u_t|\tilde{x}_t)$ can be applied. This is in stark contrast to most prevalent approaches for controlling robotic systems, which aim to accurately estimate the full state x_t. We describe our online estimation approach below.

We maintain a belief distribution $\text{bel}(\tilde{x}_t)$ over the TRV-space $\widetilde{\mathcal{X}}$ and update it at each time t using a Bayes filter [5]. Specifically, we perform two steps every t:

1) **Process Update.** The system model is used to update the belief-state to the current time step: $\overline{\text{bel}}(\tilde{x}_t) = \sum_{\tilde{x}_{t-1}} q_{t-1}(\tilde{x}_t|\tilde{x}_{t-1}, u_{t-1})\text{bel}(\tilde{x}_{t-1})$.
2) **Measurement Update.** The measurement model is used to integrate the observation y_t into the belief-state: $\text{bel}(\tilde{x}_t) \propto \sigma_t(y_t|\tilde{x}_t)\overline{\text{bel}}(\tilde{x}_t)$.

To apply this filter, the distributions $q_t(\tilde{x}_{t+1}|\tilde{x}_t, u_t)$ and $\sigma_t(y_t|\tilde{x}_t)$ are precomputed *offline*. Bayes' theorem states $p_t(x_t|\tilde{x}_t) = q_t(\tilde{x}_t|x_t)p_t(x_t)/q_t(\tilde{x}_t)$. Consequently,

$$p_t(x_{t+1}|\tilde{x}_t, u_t) = \sum_{x_t} p_t(x_{t+1}|u_t, x_t)p_t(x_t|\tilde{x}_t),$$
$$q_t(\tilde{x}_{t+1}|\tilde{x}_t, u_t) = \sum_{x_{t+1}} q_t(\tilde{x}_{t+1}|x_{t+1})p_t(x_{t+1}|\tilde{x}_t, u_t),$$
$$\sigma_t(y_t|\tilde{x}_t) = \sum_{x_t} \sigma_t(y_t|x_t)p_t(x_t|\tilde{x}_t).$$

In the discrete case, the above equations can be evaluated directly. In the LG case, $p_t(x_t|\tilde{x}_t)$ is given by the minimum mean squared error estimate of x_t given $\tilde{x}_t$. If $y_t \sim N(D_t\bar{x}_t, \Sigma_{\omega_t})$, $D_t \in \mathbb{R}^{l\times n}$, then $\sigma_t(y_t|\tilde{x}_t)$ and the mean of $q_t(\tilde{x}_{t+1}|\tilde{x}_t, u_t)$ are described by the equations

$$\bar{\tilde{x}}_{t+1} = C_{t+1}A_t[\bar{x}_t + \Sigma_{x_t}C_t^{\mathrm{T}}\Sigma_{\tilde{x}_t|x_t}(\tilde{x}_t - \bar{\tilde{x}}_t)] + C_{t+1}B_t\bar{u}_t,$$
$$\bar{y}_t = D_t\bar{x}_t + D_t\Sigma_{x_t}C_t^{\mathrm{T}}\Sigma_{\tilde{x}_t|x_t}(\tilde{x}_t - \bar{\tilde{x}}_t),$$
$$\Sigma_{y_t} = D_t\Sigma_{\tilde{x}_t|x_t}C_t^{\mathrm{T}}\Sigma_{\eta_t}^{-1}D_t^{\mathrm{T}} + \Sigma_{\omega_t}.$$

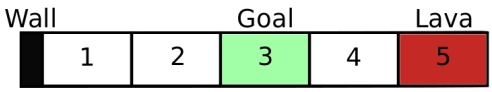

Fig. 2. The *lava scenario* [44] consists of five states connected in a line. The robot is allowed to move one step in either direction unless it enters the lava pit on the far right, which is absorbing. The robot receives a reward of 5 points upon entering the goal state and a penalty reward of -1 point otherwise. The terminal rewards are 10 points when in the goal and -10 points when in the lava.

This system is a partially observable linear-Gaussian system with process and measurement noise induced by the randomness of both $\tilde{x}_t$ and y_t. The process and measurement updates are standard Kalman filter updates applied to this system.

In the NLG case, we use an extended Kalman filter over $\delta\tilde{x}_t$ (the perturbed TRV). Specifically, we use the linearized dynamics of $\delta\tilde{x}_t$ and apply the LG systems approach.

Given a belief $\text{bel}(\tilde{x}_t)$, we compute the control input by sampling $u_t \sim \pi_t(u_t|\tilde{x}_t^\star)$, where $\tilde{x}_t^\star$ is the maximum likelihood TRV $\tilde{x}_t^\star := \max_{\tilde{x}_t} \text{bel}(\tilde{x}_t)$. Alternatively, one can sample the TRV from $q_t(\tilde{x}_t|x_t)$, but the MLE method is similar to how many Bayesian filters are implemented (e.g. Kalman filters) and allows for a more direct comparison.

V. Examples

In this section, we demonstrate the efficacy of our task-driven control approach on both a discrete-state scenario and a SLIP model. To select the value of β, the algorithm is run 10 times with β set to evenly spaced values in a listed interval. The controller used is the one with the lowest value of β subject to the expected cost of the controller being below a particular value. This choice selects the performant controller with the least state information in the TRVs.

A. Lava Problem

The first example (Fig. 2), adapted from [44] demonstrates a setting where the separation principle *does not hold*. If the robot's belief distribution is $[0.3, 0.4, 0.0, 0.3, 0.0]^{\mathrm{T}}$ while residing in state 4, the optimal action corresponding to the MLE of the state is to move right — not the optimal left.

Our algorithm was run with three TRVs and $\beta \in [0.001, 1]$ for 30 iterations. The value $\beta = 0.001$ was found to give a negative expected cost. The robot's initial condition was sampled from the aforementioned belief distribution. Online, the robot was modeled to have a faulty sensor localizes to the correct state with a probability of 0.5, with a uniformly random incorrect state returned otherwise.

Fig. 3 compares our algorithm's performance with a separation principle approach, i.e. solving the MDP with perfect state information and then performs MLE for the state online. Interestingly, our algorithm produces a *deterministic policy* that moves the robot left three times — ensuring it is in state 1 — then moves right twice and remains in the goal. With this policy, it is impossible for the robot to enter the lava state, producing *a more performant, lower variance trajectory than the separation principle-based solution under measurement noise*. The solution is deterministic at low values of β because the distribution in (6) is almost uniform — thereby requiring the policy to be effectively open-loop.

B. SLIP Model Problems

Next, we apply the NLG variant of our algorithm to the SLIP model [45], [46], [47], which is depicted in Fig. 1. The SLIP model is common in robotics for prototyping legged locomotion controllers. It consists of a single leg whose lateral motion is derived from a spring/piston combination. At touchdown, the state of the robot is given by $[d, \theta, \dot{r}, \dot{\theta}]^{\mathrm{T}}$ where d is displacement of the head from the origin, θ is the touchdown angle, and $\dot{r}, \dot{\theta}$ are the radial and angular velocities. The system input is $\Delta\theta$, the change in the next touchdown angle. The parameters are the head mass, $M = 1$, the spring constant, $k = 300$, gravity $g = 9.8$, and leg length $r_{max} = 1$. Despite the model's simplicity, the touchdown return map eludes a closed-form description, so MATLAB's `ode45` is used to compute and linearize the return map.

The goal is to place the head of the robot at $d = 3.2$ after three hops. This experiment is based on a set of psychology experiments that examined the cognitive information used by humans for foot placement while running [48], [49]. Our NLG algorithm was run with $\beta \in [1, 200]$, control cost matrices $R_t = 10$ for all t, and a terminal state cost as the squared distance of the robot from $d = 3.2$. The initial distribution was Gaussian with mean $[0, 0.3927, -3.273, -6.788]^{\mathrm{T}}$, which is in the vicinity of a fixed point of the return map, and covariance $10^{-3}I$. The process covariance was $\Sigma_{\epsilon_t} = 10^{-4}\mathrm{diag}(1, 0.1, 0.5, 0.5)$. Here, $\beta = 23.11$.

The results for our simulation are shown in Fig. 4. The algorithm is compared with iLQG solutions with correct and incorrect measurement models. The believed measurement model was a noisy version of the state with covariance $\Sigma_{\omega_t} = 10^{-4}I$ while the actual measurement model used $\Sigma_{\omega_t} = 10^{-3}S^{\mathrm{T}}S$ where the entries of S sampled from a standard uniform distribution each trial. The correct iLQG solution is a locally optimal solution to the problem due to the separation principle. However, when modeling error is introduced, the iLQG solution's performance degrades rapidly. Meanwhile, *the TRV-based control policy is a reliable (i.e. lower variance) and performant control strategy despite this modeling error.* In addition, the solution found by our algorithm satisfies $\mathrm{rank}(C_t) = 1$ for all t. Therefore, *the online estimator needs to only track a single TRV* corresponding to this subspace.

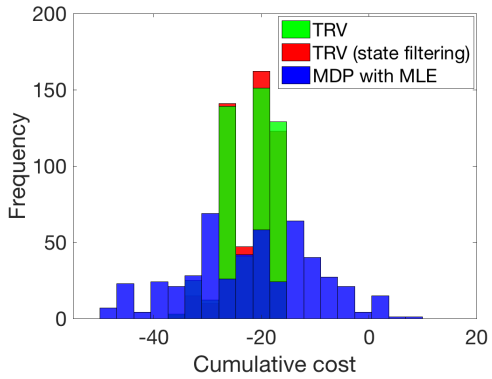

Fig. 3. This figure summarizes the outcome of 500 simulations of the Lava Problem with different control strategies. Each controller used a Bayesian filter to track the current belief distribution. The exact MDP solution (blue) applied the input corresponding to its MLE state. The TRV solutions sampled from the conditional distribution corresponding to their stochastic control policies given the MLE estimates of the state (red) or TRV (green).

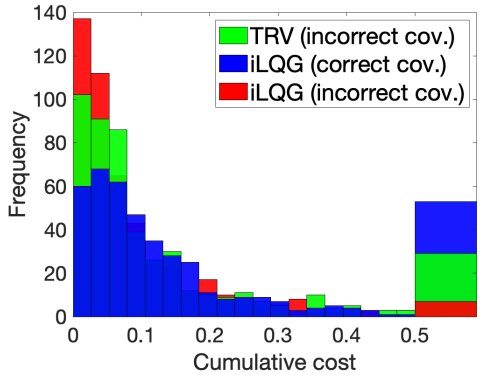

Fig. 4. This figure summarizes the outcome of 500 simulations of the SLIP Problem with different control strategies. Measurement covariance matrices were randomly sampled. For the iLQG control policy, a Kalman filter tracked the current state estimate, and the corresponding control was applied. For the TRV policy, a Kalman filter was maintained on the TRVs, and the control input corresponding to the MLE TRV was applied.

VI. Conclusion

We presented an algorithmic approach for task-driven estimation and control for robotic systems. In contrast to prevalent approaches for control that rely on accurate full-state estimation, our approach synthesizes a set of task-relevant variables (TRVs) that are sufficient for achieving the task. Our key insight is to pose the search for TRVs as an information bottleneck optimization problem. We solve this problem offline to identify TRVs and a control policy based on the TRVs. Online, we only estimate the TRVs to apply the policy. Our theoretical results suggest that this approach affords robustness to *unmodeled* measurement uncertainty. This is validated by thorough simulations, including a SLIP model running to a target location. Our simulations also demonstrate that our approach finds highly compressed TRVs (e.g. a *one-dimensional* TRV for the SLIP model).

Challenges and Future Work. On the algorithmic front, we plan to develop approaches that directly minimize the RHS of (3) instead of a linear approximation of it. We expect that this may lead to improved robustness (as suggested by Theorem 2.1). On the practical front, we plan to implement our approach on a hardware platform that mimics the gaze heuristic and other examples including navigation problems (where the full state includes a description of the environment, e.g. in terms of an occupancy map). Perhaps the most exciting direction is to explore *active* versions of our approach where the control policy minimizes task-relevant uncertainty, in contrast to current approaches (e.g., belief space planning) that minimize full-state uncertainty.

We believe that the approach presented in this paper along with the indicated future directions represent an important step towards developing a principled, general framework for task-driven estimation and control.

REFERENCES

[1] G. Gigerenzer, *Gut feelings: The intelligence of the unconscious*. Penguin, 2007.

[2] P. McLeod, N. Reed, and Z. Dienes, "Psychophysics: How fielders arrive in time to catch the ball," *Nature*, vol. 426, no. 6964, p. 244, 2003.

[3] D. M. Shaffer, S. M. Krauchunas, M. Eddy, and M. K. McBeath, "How dogs navigate to catch frisbees," *Psychological Science*, vol. 15, no. 7, pp. 437–441, 2004.

[4] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Annu. Allerton Conf. Communication, Control, and Computing*, 1999.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[6] B. A. Francis, *A course in H-infinity control theory*. Berlin; New York: Springer-Verlag, 1987.

[7] C. Edwards and S. Spurgeon, *Sliding mode control: theory and applications*. CRC Press, 1998.

[8] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[9] R. Ortega, J. A. L. Perez, P. J. Nicklasson, and H. J. Sira-Ramirez, *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013.

[10] H. K. Khalil, "Nonlinear systems," *Prentice-Hall, New Jersey*, vol. 2, no. 5, pp. 5–1, 1996.

[11] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[12] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proc. of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[13] A. S. Matveev and A. V. Savkin, *Estimation and control over communication networks*. Springer Science & Business Media, 2009.

[14] T. Tanaka, K.-K. K. Kim, P. A. Parrilo, and S. K. Mitter, "Semidefinite programming approach to Gaussian sequential rate-distortion trade-offs," *IEEE Trans. Automat. Control*, vol. 62, no. 4, pp. 1896–1910, 2017.

[15] H. A. Simon, "Theories of bounded rationality," *Decision and organization*, vol. 1, no. 1, pp. 161–176, 1972.

[16] G. Gigerenzer and R. Selten, *Bounded rationality: The adaptive toolbox*. MIT press, 2002.

[17] N. Tishby and D. Polani, "Information theory of decisions and actions," in *Perception-Action Cycle*. Springer, 2011, pp. 601–636.

[18] J. Grau-Moya, F. Leibfried, T. Genewein, and D. A. Braun, "Planning with information-processing constraints and model uncertainty in Markov decision processes," in *Joint European Conf. Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 475–491.

[19] E. Todorov, "Efficient computation of optimal actions," *Proc. Nat. Academy of Sciences*, vol. 106, no. 28, pp. 11 478–11 483, 2009.

[20] D. A. Braun, P. A. Ortega, E. Theodorou, and S. Schaal, "Path integral control and bounded rationality," in *IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning (ADPRL)*. IEEE, 2011, pp. 202–209.

[21] 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 *IEEE Int. Conf. Robotics and Automation*, 2017.

[22] J. L. Williams, J. W. Fisher III, and A. S. Willsky, "Performance guarantees for information theoretic active inference," in *Artificial Intelligence and Statistics*, 2007, pp. 620–627.

[23] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "Control and sensing co-design," *arXiv preprint arXiv:1802.08376*, 2018.

[24] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," in *IEEE Int. Conf. Robotics and Automation*. IEEE, 2017, pp. 3886–3893.

[25] A. Achille and S. Soatto, "A separation principle for control in the age of deep learning," *Annu. Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, p. null, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105140

[26] R. Fox and N. Tishby, "Minimum-information LQG control part i: Memoryless controllers," in *55th Conf. Decision and Control*. IEEE, 2016, pp. 5610–5616.

[27] ——, "Minimum-information LQG control part ii: Retentive controllers," in *Proc. 55th Conf. Decision and Control*. IEEE, 2016, pp. 5603–5609.

[28] J. L. Massey, "Causality, feedback and directed information," in *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, 1990, pp. 303–305.

[29] T. Tanaka, P. M. Esfahani, and S. K. Mitter, "LQG control with minimum directed information: Semidefinite programming approach," *IEEE Trans. Automat. Control*, vol. 63, no. 1, pp. 37–52, 2018.

[30] S. Soatto, "Actionable information in vision," in *Machine Learning for Computer Vision*. Springer, 2013, pp. 17–48.

[31] ——, "Steps towards a theory of visual information: Active perception, signal-to-symbol conversion and the interplay between sensing and control," *arXiv preprint arXiv:1110.2053*, 2011.

[32] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[33] R. M. Gray, *Entropy and information theory*. Springer Science & Business Media, 2011.

[34] V. Pacelli and A. Majumdar, "Task-driven estimation and control via information bottlenecks," *Extended version*, 2018. [Online]. Available: https://irom-lab.princeton.edu/publications/

[35] P. Whittle, "Risk-sensitive linear/quadratic/Gaussian control," *Advances in Applied Probability*, vol. 13, no. 4, pp. 764–777, 1981.

[36] ——, "Risk sensitivity, a strangely pervasive concept," *Macroeconomic Dynamics*, vol. 6, no. 1, p. 5, 2002.

[37] K. Glover and J. C. Doyle, "State-space formulae for all stabilizing controllers that satisfy an h_∞-norm bound and relations to relations to risk sensitivity," *Systems & Control Letters*, vol. 11, no. 3, pp. 167–172, 1988.

[38] R. Fox and N. Tishby, "Optimal selective attention in reactive agents," *arXiv preprint arXiv:1512.08575*, 2015.

[39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[40] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[41] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proc. American Control Conf.* IEEE, 2005, pp. 300–306.

[42] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *Proc. Int. Conf. on Informatics in Control, Automation, and Robotics*, 2004, pp. 222–229.

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

[44] P. R. Florence, "Integrated perception and control at high speed," Master's thesis, Massachusetts Institute of Technology, 2017.

[45] R. T. M'Closkey and J. W. Burdick, "Periodic motions of a hopping robot with vertical and forward motion," *Int. J. of Robotics Research*, vol. 12, no. 3, pp. 197–218, 1993.

[46] W. J. Schwind and D. E. Koditschek, "Control of forward velocity for a simplified planar hopping robot," in *Proc. Int. Conf. Robotics and Automation, 1995. Proceedings.,*, vol. 1. IEEE, 1995, pp. 691–696.

[47] H. Geyer, "Simple models of legged locomotion based on compliant limb behavior," Ph.D. dissertation, Verlag nicht ermittelbar, 2005.

[48] W. H. Warren Jr., D. S. Young, and D. N. Lee, "Visual control of step length during running over irregular terrain." *J. of Experimental Psychology: Human Perception and Performance*, vol. 12, no. 3, p. 259, 1986.

[49] W. H. Warren, "Action-scaled information for the visual control of locomotion," in *Closing the Gap*. Psychology Press, 2012, pp. 261–296.