Latent Kullback Leibler Control for Continuous-State Systems using Probabilistic Graphical Models

Takamitsu Matsubara[†] **Vicenç Gómez**[§,*] **Hilbert J. Kappen**[§]

[†] Graduate School of Information Science. Nara Institute of Science and Technology (NAIST). Nara, Japan
[§] Donders Institute for Brain, Cognition and Behaviour. Radboud University Nijmegen, the Netherlands
[*] Department of Information and Communication Technologies. Universitat Pompeu Fabra. Barcelona, Spain

Abstract

Kullback Leibler (KL) control problems allow for efficient computation of optimal control by solving a principal eigenvector problem. However, direct applicability of such framework to continuous state-action systems is limited. In this paper, we propose to embed a KL control problem in a probabilistic graphical model where observed variables correspond to the continuous (possibly high-dimensional) state of the system and latent variables correspond to a discrete (low-dimensional) representation of the state amenable for KL control computation. We present two examples of this approach. The first one uses standard hidden Markov models (HMMs) and computes exact optimal control, but is only applicable to low-dimensional systems. The second one uses factorial HMMs, it is scalable to higher dimensional problems, but control computation is approximate. We illustrate both examples in several robot motor control tasks.

1 INTRODUCTION

Recent research in stochastic optimal control theory has identified a class of problems known as Kullback-Leibler (KL) control problems (Kappen et al., 2012) or linearly solvable Markov decision problems (LSMDPs) (Todorov, 2006). For these (discrete) problems, the set of actions and the cost function are restricted in a way that makes the Bellman equation linear and thus more efficiently solvable, for instance, by solving the principal eigenvector of a certain linear operator (Todorov, 2009a).

However, direct applicability of this framework to continuous state-action systems, such as robot motor control, is limited. The main problem is the curse of dimensionality, which appears because discretization quickly leads to a combinatorial explosion. This problem has been addressed using function approximation methods in (Todorov, 2009b). Instead of directly solving a discrete-state LSMDP, these methods approximate the so-called desirability function, which is defined in the continuous-state space. Kinjo et al. (2013) combined this function approximation scheme with system identification on a real robot navigation task. However, approaches based on the continuous-state formulation of KL control problems have several limitations: they require to solve a quadratic programming problem, a more computationally demanding problem than computing the principal eigenvector. Also, there is no guarantee of convergence to a positive solution. Alternative formulations that address these limitations have been recently proposed (Zhong and Todorov, 2011a,b). Zhong and Todorov (2011a) used a soft aggregation method to solve KL-control problems in an aggregated space. Both approaches, however, require the model of system dynamics, which is often not available in real-world applications (Kinjo et al., 2013).

In this paper, we propose to embed a KL-control problem in a probabilistic graphical model with mixed continuous and discrete variables. The continuous variables correspond to the (possibly high-dimensional) state of the system and the discrete variables correspond to a latent (low-dimensional) representation of the state which is amenable for KL control computation. The model parameters are first learned using data from the real system running with exploring controls. The control input to the real system is then computed as a filtering step combined with the solution of the KL-control problem in the latent space.

We present two examples of this approach: the first one uses a standard hidden Markov model (HMM) in which inference can be computed exactly, but is only applicable to low-dimensional continuous systems. The second one uses factorial HMMs (FHMMs) and is applicable to higher dimensional problems, although optimal control can only be approximated. We illustrate both examples in several robot motor control tasks. In particular, we experimentally demonstrate that the second example with FHMMs is scalable to high-dimensional problems (e.g., 25 dimensional problem) that may not be solvable by other approaches.

2 KULLBACK LEIBLER CONTROL PROBLEMS

We briefly summarize the class of KL control problems introduced by Todorov (2006) in the infinite-horizon average-cost formulation (see also Todorov, 2009a).

Let $\mathcal{X} = \{1, \ldots, N\}$ be a finite set of states and $\mathcal{U}(x)$ be a set of admissible control actions at state $x \in \mathcal{X}$. Consider the transition probability $p(x'|x)$ that describes the system dynamics in the absence of control. Such *uncontrolled dynamics* assigns zero probability for physically forbidden state transitions. Denote the transition probability given action $u \in \mathcal{U}(x)$ as $p(x'|x,u)$ and the immediate cost for being in state x and taking action u as $\ell(x,u) \geq 0$.

For infinite-horizon problems, the objective is to find a control law $u = \pi(x)$ that minimizes the average cost

$$\lim_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{n-1} \ell(x_t, \pi(x_t))\right] = \sum_x \Pi(x)\ell(x,\pi(x)) \quad (1)$$

where n is the number of time-steps and $\Pi(x) = \lim_{t\to\infty} p(x_t = x|x_0, \pi)$ is the stationary distribution of states under control law π, which we assume exists and is independent of x_0, i.e., $p(x_t = x|x_0,\pi)$ is assumed ergodic.

The following Bellman equation defined for the (differential) cost-to-go function $v(x)$ minimizes Eq. (1)

$$c + v(x) = \min_{u\in\mathcal{U}(x)} \left\{\ell(x,u) + \mathbb{E}_{x'\sim p(\cdot|x,u)}[v(x')]\right\}, \quad (2)$$

where c is the average cost that does not depend on the starting state.

Minimizing Eq. (2) is in general hard, but in some cases it can be done efficiently. KL control problems are a class of problems for which Eq. (2) becomes linear under the following assumptions:

(i) the controls directly specify state transition probabilities, i.e. $p(x'|x,u) = u(x'|x)$. The action vector $u(\cdot|x)$ is a probability distribution over next states given the current state x.

(ii) the immediate cost function has the following form

$$\ell(x,u) = \alpha q(x) + \mathrm{KL}\left(u(\cdot|x) \parallel p(\cdot|x)\right),$$

where $q(x) \geq 0$ is an arbitrary state-dependent cost and KL is the Kullback Leibler divergence between the controlled and the uncontrolled dynamics, reflecting how much the control changes the normal behavior of the system. Parameter α allows to balance the two cost terms.

Define the exponentiated cost-to-go (desirability) function $z(x) = \exp(-v(x))$ and the linear operator

$$\mathcal{G}[z](x) = \sum_{x'} p(x'|x)z(x') = \mathbb{E}_{x'\sim p(\cdot|x,u)}[z(x')].$$

The resulting minimization takes the form

$$\min_{u\in\mathcal{U}(x)}\left\{\alpha q(x) + \mathrm{KL}\left(u(\cdot|x) \left\| \frac{p(\cdot|x)z(\cdot)}{\mathcal{G}[z](x)}\right.\right) - \log \mathcal{G}[z](x)\right\}.$$

At the global minimum, the Bellman equation becomes

$$\exp(-c)z(x) = \exp(-\alpha q(x))\mathcal{G}[z](x)$$

or in matrix form

$$\lambda\mathbf{z} = \mathbf{GPz} \quad (3)$$

where $\mathbf{G}$ is a $N \times N$ diagonal matrix with elements $\exp(-\alpha q(x))$ and $\lambda = \exp(-c)$. From Eq. (3), it follows that $\mathbf{z}$ is any eigenvector of the matrix $\mathbf{GP}$ with eigenvalue λ. The optimal average cost becomes $c = -\ln\lambda$. Thus, the minimal solution is given by the principal eigenvector of $\mathbf{GP}$: the eigenvector $\mathbf{z}^*$ with largest eigenvalue, which can be efficiently computed using the power iteration method (Todorov, 2006). The optimal control is given by

$$u^*(x'|x) = \frac{p(x'|x)\mathbf{z}^*(x')}{\mathcal{G}[\mathbf{z}^*](x)}. \quad (4)$$

3 LATENT KULLBACK LEIBLER CONTROL

The previously described framework is not directly applicable for continuous systems. For such cases, we propose to learn a discrete hidden representation and dynamics amenable for efficient computation from the observed continuous variables. Our approach can be summarized in the following three steps:

1. Learn a probabilistic graphical model from data samples obtained for the real system

2. Solve the KL control problem in the latent space of the probabilistic graphical model

3. Compute control in the observed space

This general method is directly applicable to arbitrary continuous state-action systems, while in this paper we focus on the following deterministic control-affine systems that typically describe discrete-time robot dynamics:

$$y_{t+1} = y_t + \Delta t \left(\mathbf{f}(y_t) + \mathbf{B}(y_t)\boldsymbol{\tau}_t \right), \qquad (5)$$

where $y_t \in \mathbb{R}^D$ is the state variable of the system, $\boldsymbol{\tau}_t \in \mathbb{R}^d$ is the control input, $\mathbf{f}(y_t) \in \mathbb{R}^D$ is the uncontrolled dynamics, $\mathbf{B}(y_t) \in \mathbb{R}^{D \times d}$ is the control matrix and Δt is the discrete-time step-size.

Two particular realizations of this general approach are described in the next section. The first one uses standard HMMs, which are the most natural way to model sequences of observations. However, it is only applicable to systems in which the relevant region of the state-space is small, such as low-dimensional systems, or largely constrained high-dimensional systems. The second one uses factorial HMMs, which assume factorized uncontrolled dynamics and can scale up to higher dimensional problems.

4 EXACT CONTROL COMPUTATION USING HIDDEN MARKOV MODELS

In this section, we describe an example of latent KL control based on standard hidden Markov models.

4.1 LEARNING HMMS FOR KL CONTROL

Consider the hidden Markov model with hidden states $x_t \in \{1, \ldots, N\}$, stochastic state transition matrix $\mathbf{P}$ with entries $P_{ij} = p(x_{t+1} = j | x_t = i)$ and Gaussian observation model $p(\mathbf{y}_t | x_t = k) = \mathcal{N}(\mu_k, \Sigma_k)$.

We generate sample trajectories $\mathcal{D} = \{\mathbf{y}_t, \ldots, \mathbf{y}_T\}$ from the real system driven solely by exploration noise (uncontrolled dynamics) and use them to learn the parameters $\boldsymbol{\theta}_{\mathrm{HMM}} = \{\mathbf{P}, \mu_{1:N}, \Sigma_{1:N}\}$. After learning, the matrix $\mathbf{P}$ encodes a coarse description of the observed dynamics in a latent space and the Gaussian means and variances capture the relevant regions in this space. More precisely, considering the system of Eq. (5), we set exploration noise as $\boldsymbol{\tau}_t = \epsilon_t$ for $t = 1 \ldots T$, where $\epsilon_t \in \mathbb{R}^d \sim \mathcal{N}(0, \Sigma_\epsilon)$. The choice of such a zero-mean Gaussian distribution is motivated by the relationship between the KL action cost and the input-norm cost: in the continuous setting the KL cost reduces to a quadratic energy cost (Todorov, 2009a; Kappen et al., 2012), which coincides with a commonly used input-norm cost for energy-efficient or smooth motor control behavior (Mitrovic et al., 2010). The covariance matrix Σ_ϵ is a free parameter. For low exploration noise, one would expect the learned model to be a poor approximation since only a small fraction of the state space is visited. Conversely, large noise values would result in too flexible models with unrealistic state transitions. The correct noise value is therefore a trade-off between these two scenarios.

Given $\mathcal{D}$, the parameters $\boldsymbol{\theta}_{\mathrm{HMM}}$ can be learned, for instance, using the standard Expectation-Maximization (EM) algorithm (Baum-Welch algorithm).

4.2 CONTROL COMPUTATION IN LATENT SPACE

To define a KL control problem in the latent space, we first need a state-dependent cost function expressed in terms of the latent variable x. Let $\tilde{q}(\mathbf{y}_t)$ and $q(x_t)$ be the cost functions in observation and latent spaces, respectively. We define $q(x_t)$ given $\tilde{q}(\mathbf{y}_t)$ using $\exp(-q(x_t)) = \int_{\mathbf{y}_t} \exp(-\alpha\tilde{q}(\mathbf{y}_t))p(\mathbf{y}_t|x_t)d\mathbf{y}_t$. Furthermore, if $\tilde{q}(\mathbf{y}_t)$ is given in quadratic form $\tilde{q}(\mathbf{y}_t) = (\mathbf{y}_t - \mu_q)^T \Sigma_q^{-1}(\mathbf{y}_t - \mu_q) = ||\mathbf{y}_t - \mu_q||^2_{\Sigma_q^{-1}}$ and the observation model is Gaussian $p(\mathbf{y}_t|x_t) = \mathcal{N}(\mu_x, \Sigma_x)$, we can obtain $q(x_t)$ analytically:

$$\begin{aligned} q(x_t) &= -\ln\left\{ \int_{\mathbf{y}_t} \exp\left(-\alpha\tilde{q}(\mathbf{y}_t)\right) p(\mathbf{y}_t|x_t) d\mathbf{y}_t \right\} \\ &= -\ln\left\{ \frac{|\mathbf{S}|^{1/2}}{|\Sigma_x|^{1/2}} \exp\left[-\frac{1}{2} ||\mu_q - \mu_x||^2_{\mathbf{M}^{-1}} \right] \right\} \end{aligned}$$

where, $\mathbf{S} = (\alpha\Sigma_q^{-1} + \Sigma_x^{-1})^{-1}$ and $\mathbf{M} = \alpha^{-1}\Sigma_q + \Sigma_x$.

The (latent) KL control problem can now be formulated using state cost $q(x_t)$ and uncontrolled dynamics $\mathbf{P}$ as in Eq. (3). The optimal state transition $u^*(x_{t+1}|x_t)$ under controlled dynamics is given by Eq. (4).

4.3 CONTROL COMPUTATION IN OBSERVED SPACE

We are now ready to describe how to use latent KL control in the real system. Given an observation sequence $\mathbf{y}_{1:t}$ until time t, we can compute predictive distributions of the next observation $\mathbf{y}_{t+1}$ under both the uncontrolled dynamics $p(x_{t+1}|x_t)$ and the optimally controlled dynamics $u^*(x_{t+1}|x_t)$ in the latent space as:

$$\begin{aligned} p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}) &= \sum_{x_{t:t+1}} p(\mathbf{y}_{t+1}|x_{t+1}) p(x_{t+1}|x_t) u(x_t|\mathbf{y}_{1:t}) \\ u(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}) &= \sum_{x_{t:t+1}} p(\mathbf{y}_{t+1}|x_{t+1}) u^*(x_{t+1}|x_t) u(x_t|\mathbf{y}_{1:t}) \end{aligned}$$

where $u(x_t|\mathbf{y}_{1:t})$ denotes the filtered state at time t following the controlled process that evolves according to

$u^*(x'|x)$. Since we keep the previous filtered estimate $u(x_{t-1}|\mathbf{y}_{1:t-1})$, this computation is simply as

$$u(x_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|x_t)\sum_{x_{t-1}} u^*(x_t|x_{t-1})u(x_{t-1}|\mathbf{y}_{1:t-1})}{u(\mathbf{y}_t|\mathbf{y}_{1:t-1})}.$$

We finally compute the control input command to the system such that the "difference" between the uncontrolled and optimal behaviors is reduced

$$\boldsymbol{\tau}_t = \mathbf{K}(\bar{\mathbf{y}}^u_{t+1|1:t} - \bar{\mathbf{y}}^p_{t+1|1:t}), \tag{6}$$

where $\bar{\mathbf{y}}^u_{t+1|1:t}$ and $\bar{\mathbf{y}}^p_{t+1|1:t}$ are the expectations of $\mathbf{y}$ over $u(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})$ and $p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})$ respectively and $\mathbf{K}$ is a gain matrix to be tuned. The gain $\mathbf{K}$ can be optimally computed if the model of system dynamics is available (Todorov, 2009b), however, in this paper we focus on the model-free scenario and leave it as a free parameter.

5 APPROXIMATE CONTROL USING FACTORIAL HIDDEN MARKOV MODELS

For high-dimensional problems that require to cover large regions of the state space, the previous approach becomes infeasible, since the cardinality required for the latent variable grows exponentially. In this section, we consider an alternative model with a multi-dimensional latent variable and constrained state transitions. We consider each dimension independent from the rest in the absence of control. These assumptions are naturally expressed using factorial HMMs. The advantage is that we can capture complex latent dynamics more efficiently. The price to pay is that exact optimal control computation in the latent space is no longer feasible and different approximation schemes have to be used. We describe this approach in the following sections.

5.1 FACTORIAL HIDDEN MARKOV MODELS

FHMM is a special type of HMM to model sequences of observations originated from multiple latent dynamical processes that interact to generate a single output (Ghahramani and Jordan, 1997; Murphy, 2012). The state is represented by a collection of variables $\mathbf{x}_t = \{x^{(1)}_t, \ldots, x^{(m)}_t, \ldots, x^{(M)}_t\}$ each of them having K possible values. The latent state $\mathbf{x}_t$ is thus a M-dimensional variable with K^M possible values.

We will use a 1-of-K encoding, such that each state component $\mathbf{x}^{(m)}_t$ will be denoted using a $K \times 1$ vector, where each of the K discrete values corresponds to a 1 in one position and 0 elsewhere.

The assumption is that the transition model factorizes among the individual components

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \prod_{m=1}^{M} p^{(m)}(\mathbf{x}^{(m)}_t|\mathbf{x}^{(m)}_{t-1}), \tag{7}$$

where $p^{(m)}(\mathbf{x}^{(m)}_t|\mathbf{x}^{(m)}_{t-1})$ is the state transition matrix $\mathbf{P}^{(m)}$ for the m-th chain. We assume the Gaussian observation model, which is defined as

$$p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}\left(\sum_{m=1}^{M} \mathbf{W}^{(m)}\mathbf{x}^{(m)}_t, \Sigma\right) \tag{8}$$

where $\mathbf{W}^{(m)}$ is a $D \times K$ weight matrix that contains in its columns the contributions to the means for each of the possible configurations of $\mathbf{x}^{(m)}_t$. The marginal over $\mathbf{y}_t$ is thus a Gaussian mixture model, with K^M Gaussian mixture components, each having a constant covariance matrix Σ.

The parameters $\boldsymbol{\theta}_{\text{FHMM}} = \{\mathbf{P}^{1:M}, \mathbf{W}^{1:M}, \Sigma\}$ can be learned using EM, as before. In this case, however, the E-step becomes intractable, since the forward-backward step has time complexity $\mathcal{O}(TMK^{M+1})$. An alternative approximation that works well in practice is the structured mean field approximation, which has time complexity $\mathcal{O}(TMK^2I)$, where I is the number of mean field iterations (see Ghahramani and Jordan, 1997; Murphy, 2012, for details).

5.2 CONTROL COMPUTATION IN LATENT SPACE

In a similar way as in Section 4.2, we need first to define a cost function in the latent space $q(\mathbf{x}_t)$ to be able to formulate a KL control problem. A natural way to define $q(\mathbf{x}_t)$ given the observation model of Eq. (8) and the cost function in observation space $\tilde{q}(\mathbf{y}_t)$ is

$$q(\mathbf{x}_t) = \alpha\tilde{q}\left(\sum_{m=1}^{M} \mathbf{W}^{(m)}\mathbf{x}^{(m)}_t\right). \tag{9}$$

Computing the exact optimal control using Eq. (3) in FHMMs requires to transform the model into a single chain model with K^M states, which is intractable. We assume approximate controlled dynamics $u_{\text{ap}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ and associated stationary distribution $\Pi_{\text{ap}}(\mathbf{x}_t)$ that factorize in its components:

$$u_{\text{ap}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \prod_{m=1}^{M} u^{(m)}_{\text{ap}}(\mathbf{x}^{(m)}_t|\mathbf{x}^{(m)}_{t-1})$$

$$\Pi_{\text{ap}}(\mathbf{x}_t) = \prod_{m=1}^{M} \Pi^{(m)}_{\text{ap}}(\mathbf{x}^{(m)}_t).$$

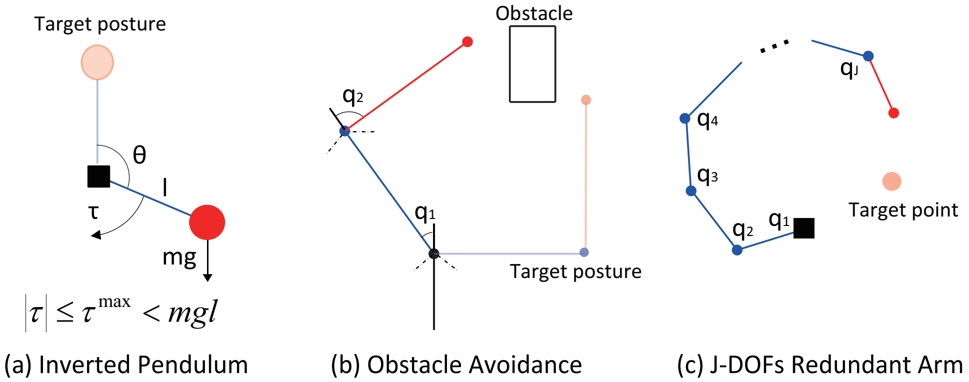


Figure 1: Motor control problems with simulated robots: **(a) Pendulum swing up with limited torque.** The state variable is $y = [\theta, \omega]^T$ where $\omega = \dot{\theta}$, $|\theta| \le \pi$, $|\omega| \le 4\pi$. The control input is the torque at the joint τ. Uncontrolled dynamics and control matrix are given as $\mathbf{f}(y) = [0\ 1; g\sin(\theta)/l\ \mu/\omega]$, $\mathbf{B}(y) = 1/ml^2$, respectively. Parameters values are $m = l = 1$, $g = 9.8$, $\mu = 0.25$ and $\tau^{max} = 5.0$ that satisfies $\tau^{max} < mgl$; **(b) Robot arm control with obstacle.** The state variable is $y = [q_1, q_2]^T \in \mathcal{S}$ where $\mathcal{S}$ is the state space that satisfies the joint angle limits and no collisions with the obstacle. The control input is $\boldsymbol{\tau} = \dot{y}$. The uncontrolled dynamics and control matrix are $\mathbf{f}(y) = [0, 0]^T$ and $\mathbf{B}(y) = I_D$; **(c) Multi-DOF redundant arm reaching task.** The state variable is $y_t = [q_1(t), \ldots, q_J(t)]^T$, $q_i(t) \in \mathcal{S}$ is the i-th joint angle and $\mathcal{S}$ is the state space that satisfies the joint angle limit $-0.5\pi \le q_i(t) \le 0.5\pi$. The control input, uncontrolled dynamics and control matrix are as in (b), but for J dimensions. In all examples we use first-order Euler method for numerical integration.

These assumptions imply that the KL cost term can also be decomposed such that Eq. (1) becomes

$$\sum_{\mathbf{x}_t} \prod_{m=1}^{M} \Pi_{\text{ap}}^{(m)}(\mathbf{x}_t^{(m)}) \times \left(q(\mathbf{x}_t) + \sum_{m=1}^{M} \text{KL}\left(u_{\text{ap}}^{(m)}(\cdot|\mathbf{x}_t^{(m)}) \middle\| p^{(m)}(\cdot|\mathbf{x}_t^{(m)}) \right) \right). \tag{10}$$

We can minimize Eq. (10) iteratively using sequential updates: for each chain m, update the parameters $u_{\text{ap}}^{(m)}$ and $\Pi_{\text{ap}}^{(m)}$ assuming the parameters for the other chains fixed so that it minimizes the marginal state-dependent cost

$$Q^{(m)}(\mathbf{x}_t^{(m)}) = \sum_{\mathbf{x}_t^{(i)}, i \ne m} \prod_{i \ne m} \Pi_{\text{ap}}^{(i)}(\mathbf{x}_t^{(i)}) q(\mathbf{x}_t) \tag{11}$$

and the corresponding KL cost. Each update corresponds to a sub-problem of the type of Eq. (3) and can be solved as a principal eigenvector problem. The average cost monotonically decreases at each iteration and its convergence is guaranteed. We call this scheme *Variational KL minimization* **(VKL)**.

Note however, **VKL** requires summing over all the values of the $M - 1$ chains to obtain the marginal state-dependent cost, and thus it has time complexity $O(K^{M-1})$, which is still intractable. We further approximate this computation by taking the expected state of the other chains according to their individual stationary distributions

$$Q^{(m)}(\mathbf{x}_t^{(m)}) \approx \alpha \tilde{q} \left(\mathbf{W}^{(m)} \mathbf{x}_t^{(m)} + \sum_{i \ne m} \mathbf{W}^{(i)} \Pi_{\text{ap}}^{(i)} \right), \tag{12}$$

where $\Pi_{\text{ap}}^{(i)}$ is a K-dimensional vector with the stationary distribution of chain i. Evaluation of Eq. (12) only requires $O(KM)$ steps, and it is therefore tractable. We refer this approximation as *Approximate Variational KL minimization* **(AVKL)**.

We refer to the control computed using either **VKL** and **AVKL** as u_{ap}^* in the rest of this section.

5.3 CONTROL COMPUTATION IN OBSERVED SPACE

Having approximated our optimal control law in the latent space, we need to define a control law for the real (observed) system given sequence of observations $\mathbf{y}_{1:t}$. We follow the same approach as in Section 4.3. First, we obtain estimates for the expected values of the next observed state under both controlled and uncontrolled dynamics as $\bar{\mathbf{y}}_{t+1|1:t}^{u}$ and $\bar{\mathbf{y}}_{t+1|1:t}^{p}$, respectively. Second, we apply the controller of Eq. (6).

The first step requires to solve a filtering problem to obtain $u(\mathbf{x}_t|\mathbf{y}_{1:t})$, which is intractable for this model. We use an approximate approach based on structured mean field, as in the model learning step (Section 5.1, E-step). However, instead of keeping the last filtered estimate $u(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ as before, we keep the filtered estimate at time-step $t-H$, i.e. $u(\mathbf{x}_{t-H}|\mathbf{y}_{1:t-H})$ and perform *offline* structured mean field using the last H observations $\mathbf{y}_{t-H:t}$. This approach improves considerably the accuracy of the filtered estimates $u(\mathbf{x}_t|\mathbf{y}_{1:t}) = \prod_m u^{(m)}(\mathbf{x}^{(m)}_t|\mathbf{y}_{1:t})$ and at the same time, it is more efficient than structured mean field on the entire sequence of past observations.

Once we have filtered estimates of the latent state, the expectation of $\mathbf{y}_{t+1}$ over predictive distribution $u(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})$ can be approximated using samples

$$\begin{aligned}\bar{\mathbf{y}}^u_{t+1|1:t} &= \int \mathbf{y}_{t+1} u(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}) d\mathbf{y}_{t+1} \\ &\approx \frac{1}{L}\sum_{\mu=1}^{L}\sum_{m=1}^{M} \mathbf{W}^{(m)}\hat{\mathbf{x}}^{(m)}_\mu\end{aligned}$$

where $\hat{\mathbf{x}}^{(m)}_\mu$ are samples drawn from the posterior distribution of the latent component according to the approximated controlled dynamics

$$\begin{aligned}\hat{\mathbf{x}}^{(m)}_\mu &\sim u^{(m)}(\mathbf{x}^{(m)}_{t+1}|\mathbf{y}_{1:t}) \\ &= \sum_{\mathbf{x}^{(m)}_t} u^{*,(m)}_{\mathrm{ap}}(\mathbf{x}^{(m)}_{t+1}|\mathbf{x}^{(m)}_t) u^{(m)}(\mathbf{x}^{(m)}_t|\mathbf{y}_{1:t}). \quad (13)\end{aligned}$$

Similarly, we can estimate $\bar{\mathbf{y}}^p_{t+1|1:t}$ using samples from

$$p^{(m)}(\mathbf{x}^{(m)}_{t+1}|\mathbf{y}_{1:t}) = \sum_{\mathbf{x}^{(m)}_t} p^{(m)}(\mathbf{x}^{(m)}_{t+1}|\mathbf{x}^{(m)}_t) u^{(m)}(\mathbf{x}^{(m)}_t|\mathbf{y}_{1:t}).$$

We show in the next section that for relatively small values of the window length H and the number of samples L, the resulting controls are satisfactory.

6 SIMULATION RESULTS

In this section, we apply our method to three benchmark (simulated) robot motor control problems: (a) pendulum swing-up with limited torque (Doya, 2000), (b) robot arm control with obstacles (Sugiyama et al., 2007), and (c) multi-degrees of freedom (DOF) redundant arm reaching task (Theodorou et al., 2010). Figure 1 illustrates these problems. The first two examples correspond to the approach using HMMs of Section 4 whereas the third shows an application using FHMMs as described in Section 5.

For learning the HMM parameters, we use identical and independent exploration noise in all controlled dimensions parameterized by σ^2_ϵ, i.e. $(\Sigma_\epsilon)_{ij} = \delta_{ij}\sigma^2_\epsilon$.

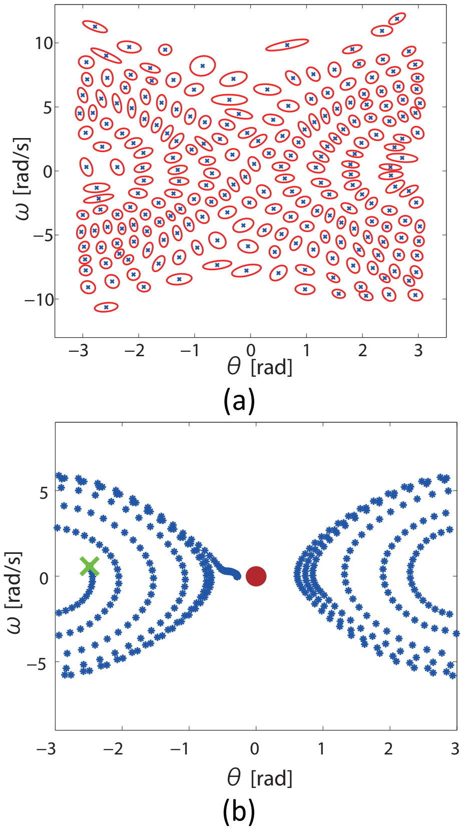


Figure 2: Pendulum swing-up task results: **(a)** Observation model after learning the HMM with $N = 225$ hidden states and $\sigma_\epsilon = 1.5$. Each hidden state corresponds to a two-dimensional Gaussian distribution with mean indicated by a cross and contour with equal probability density shown as an ellipse. **(b)** Typical controlled behaviour in the phase plane. The cross and the circle show initial and target states respectively.

Both tasks consider a two-dimensional observed continuous state and a one-dimensional latent variable. The complexity of the method strongly depends on the number of hidden values N. For this experiments, we simply choose N large enough ($N = 255$ in both scenarios) to obtain a model that accurately describes the system dynamics. We learn the full parameter vector $\boldsymbol{\theta}_{\mathrm{HMM}}$ using EM with K-means initialization for the Gaussian means.

6.1 PENDULUM SWING-UP TASK

This is a non-trivial problem when the maximum torque $\tau^{\max}$ is smaller than the maximal load torque mgl. The optimal control requires to take an energy-efficient strategy: swing the pendulum several times to build up momentum and also decelerate the pendulum early enough to prevent it from falling over.

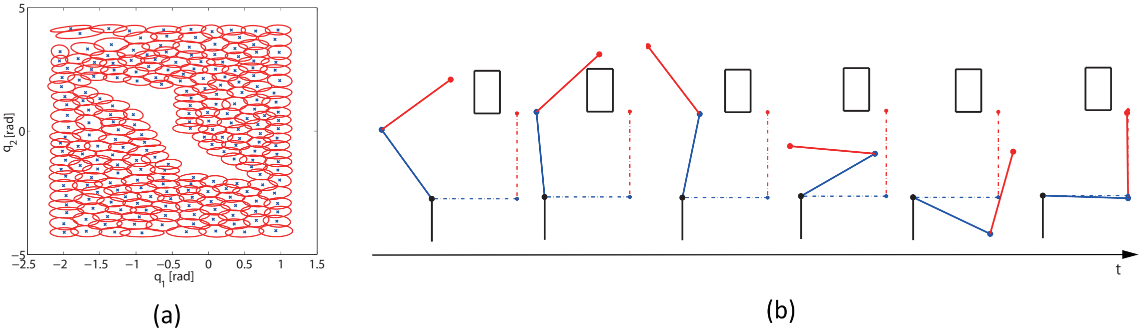


Figure 3: Results on the robot arm with an obstacle. **(a)** Learned HMM with $N = 225$, $\Sigma_\epsilon = \text{diag}\{1.5, 1.5\}$ and $T = 3 \cdot 10^4$ samples. **(b)** Controlled robot arm behavior at different time steps. The robot successfully reaches the target posture avoiding the obstacle.

Fig. 2(a) shows the 2-dimensional observation model after learning with exploration noise $\sigma_\epsilon^2 = 1.5$. We can see that the HMM is able to capture a discrete, coarse representation of the continuous state.

For control computation, we define a quadratic cost $\tilde{q}(\mathbf{y}_t) = \mathbf{y}^T \Sigma_q^{-1} \mathbf{y}$, where $\Sigma_q = \text{diag}\{0.005, 0.02\}$, and set the scale parameter $\alpha = \alpha_0 \Delta t / \sigma_\epsilon^2$ to prevent the scaling effect of the exploration noise variance σ_ϵ^2 in the KL cost ($\alpha_0 = 0.2$). The gain matrix is $\mathbf{K} = \text{diag}\{50, 10\}$. The eigenvector computation only takes $3 \cdot 10^{-2}$ seconds [1]. The computation of control input (see Section 4.3) takes $3 \cdot 10^{-3}$ seconds per time-step. The resulting controller successfully maintains the pendulum in a region of $|\theta| \leq 0.5$ continuously in all tested random initializations and it is optimal in terms of energy-efficiency. A typical controlled behavior of the pendulum is shown in Fig. 2(b).

For comparison, we also implemented standard value iteration (VI) (Sutton and Barto, 1998), which requires knowledge of the true pendulum dynamics and uses a fully discretized state-action space. For consistency, we choose as a cost function $r(\mathbf{y}, \mathbf{u}) = \alpha \tilde{q}(\mathbf{y}_t) + \frac{1}{2}||\mathbf{u}||^2$ and the same error tolerance 10^{-8} for both value iteration method and power method. VI requires a very fine discretization ($N \geq 1225$ states) and at least 20 seconds of CPU-time, which are roughly an order of magnitude larger than the values obtained using the proposed method.

6.2 ROBOT ARM CONTROL WITH OBSTACLE

In this second task, we aim to control a two-joint robot arm from an initial posture to the target posture while avoiding an obstacle. The presence of the obstacle makes this task difficult to solve using standard trajectory interpolation methods, see Fig. 1(b) for details.

Fig. 3(a) shows the 2D observation model learned using the same setup as before. As the empty region in the middle of the plot indicates, the model successfully captures the physically impossible state transitions that would bring the robot arm through the obstacle.

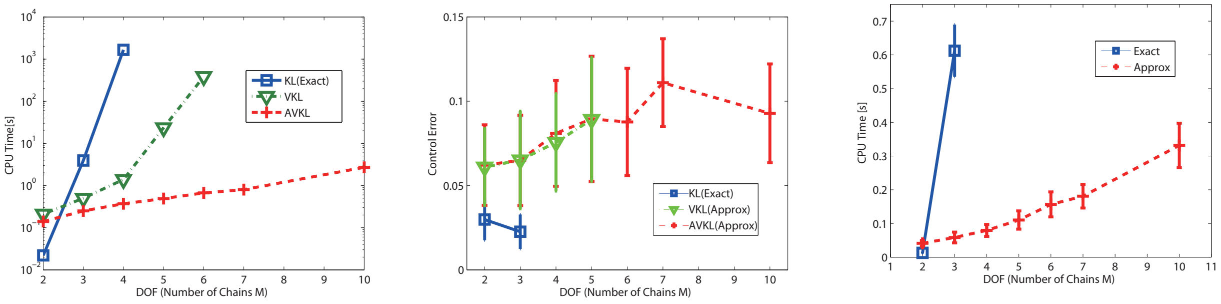
For this problem, we set the cost function as $\tilde{q}(\mathbf{y}_t) = (\mathbf{y} - \mathbf{g})^T \Sigma_q^{-1} (\mathbf{y} - \mathbf{g})$, where $\Sigma_q = \text{diag}\{0.01, 0.01\}$ and $\mathbf{g} = [-\pi/2, \pi/2]^T$. In this case, we use $\alpha_0 = 0.05$ and $\mathbf{K} = \text{diag}\{3.0, 0.5\}$ to set the scale parameter and the gain matrix, respectively. Computation time of the optimal control is approximately 0.03 seconds using the same specifications as in the previous example. Fig. 3(b) illustrates the typical controlled robot behavior. The robot arm first decreases the angle q_2 and then modifies q_1 reaching the target posture while successfully avoiding the obstacle.

6.3 REACHING TASK

The third task consists of a multi-DOF planar robot arm with J joints and joint-limit constraints as shown in Fig. 1(c). The J joints are of equal length $l = 1$ and connected to a fixed base. Each joint dynamics of this robot model is decoupled, and therefore suitable for our method using FHMMs.

The goal is to control the joint angles to reach a target position $\mathbf{t}^{\text{target}}$ with the end-effector of the robot arm. For $J \gg 2$ the control policy has to make a choice among many possible trajectories in the joint space. Moreover, considering joint-limit constraints limits direct application of standard methods for inverse kinematic, e.g. Jacobian inverse techniques (Yoshikawa,

[1] Core-i7 2.8GHz-CPU, 8GB memory and MATLAB.



(a) CPU time for KL minimization

(b) Control Error

(c) CPU time for computing control input

Figure 4: **Multi-DOF robot reaching task:** Comparison between **KL(exact)**, **VKL** and **AVKL**. **VKL** and **AVKL** can efficiently compute near optimal controller comparable to exact KL minimization. **AVKL** scales to high-dimensional problems. **KL(exact)** and **VKL** are only feasible for $J < 4$ and $J < 6$, respectively.

1990). The cost function for this task is

$$\tilde{q}(\mathbf{y}) = \| \mathbf{t}^{\text{target}} - \mathbf{T}(\mathbf{y}) \|, \tag{14}$$

where $\mathbf{T}(\cdot)$ is the forward kinematics model that maps a joint angle vector to the corresponding end-effector position in the task space

$$\mathbf{T}(\mathbf{y}) = \begin{bmatrix} \sum_{n=1}^{J} \cos\left(\sum_{j=1}^{n} y_j\right) \\ \sum_{n=1}^{J} \sin\left(\sum_{j=1}^{n} y_j\right) \end{bmatrix}.$$

Although the dynamics decouples for each joint, the cost function couples all the joint angles making the problem difficult.

We analyze the scaling properties with the number J of degrees of freedom, comparing the different strategies described in Section 5.2: **KL (exact)** minimization, **VKL** and **AVKL**. The exact solution uses K^M states and performs exact inference. For approximate methods, we use as many latent dimensions (chains) as joints $M = J$, with $K = 20$ and $H = 2J$ time-steps for approximate filtering. Note that M could be smaller than J, as long as the learned hidden representation captures well the underlying structure and dynamics. We set $M = J$ to simplify the evaluation.

Convergence of variational eigen-computations **VKL** and **AVKL** is reached after approximately 10 iterations in this task (each iteration requires an update of all the parameters of the J joints). Learning the parameters of the FHMM is sensitive to local minima. In practice, we choose $\mathbf{W}^{(m)}$ so that each factored state represents each joint dynamics and only learn the uncontrolled dynamics (transition probabilities). Also, $\mathbf{t}^{\text{target}}$ is set to one of the $\mathbf{w}_i^{(m)}$ to prevent space quantization errors in this comparison.

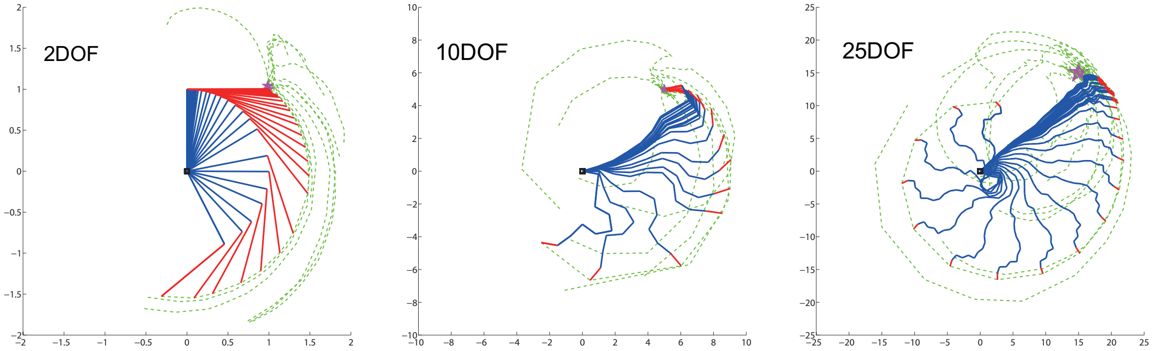
Fig. 4 illustrates the comparison. Whereas **KL (exact)** and **VKL** are only feasible for $J < 5$ and $J < 7$ respectively, **AVKL** is applicable to a larger number of joints. Fig. 4(a) shows CPU-time for control computation in the latent space (Section 5.2), which scales exponentially for both **KL (exact)** and **VKL** and approximately linear for **AVKL**.

Fig. 4(b) shows the error Eq. (14) averaged over 200 trials with randomly initialized joints. Although exact control computation can be performed for $M < 5$, exact inference is only possible for $M < 4$. We can observe that the resulting controls are satisfactory and errors do not differ significantly between **VKL** and **AVKL**. Notice that the **AVKL** error remains approximately constant as a function of M.

Fig. 4(c) shows CPU-time for the control computation in the observed space (Section 5.3). While CPU-time for exact computation quickly increases, our approximate approach results in a roughly linear increase.

Examples of controlled robot behaviors for a different number of degrees of freedom are shown in Fig. 5. In all cases, the robot successfully reaches the goal while satisfying the joint-limit constraints starting from several initial postures.

From these results we can conclude that it is feasible to learn FHMMs for high-dimensional systems with uncoupled uncontrolled dynamics and that latent KL control is an effective method to near-optimally control such systems.



(a) 2DOF

(b) 10DOF

(c) 25DOF

Figure 5: **Multi-DOF robot reaching task:** Examples of robot trajectories. The arm successfully reaches the target position while satisfying the joint-limit constraints from several initializations. Green lines show end-effector trajectories for different initializations. Blue and red lines indicate intermediate and end links.

7 DISCUSSION

We have proposed a novel solution that combines the KL control framework with probabilistic graphical models in the infinite horizon, average cost setting. Our approach learns a coarse, discrete representation amenable for efficient computation to near-optimally control continuous-state systems. We have presented two examples, using hidden Markov models (HMMs) and factorial HMMs (FHMMs), and we have shown evidence that our proposed method is feasible in three robotic tasks. In particular, we have demonstrated that the second example with FHMMs is scalable to higher dimensional problems.

The presented latent KL control approach (with HMMs) resembles the one of Zhong and Todorov (2011a) which considers an "aggregated" space similar to the latent space of the HMM. However, note that whereas for Zhong and Todorov (2011a) the real model is *required in the observed space*, in our case we *learn* an *approximate* model in which observations are coupled through the latent variables. Their main computational bottleneck is the "double" numerical integration over the observed space for computing the "aggregated" state transition probability. In our case, we replace such a problem by a probabilistic graphical model learning problem.

The control performance strongly depends on the quality of the learned model, which requires choosing a proper exploration noise and a proper initialization of the graphical model parameters. Current work is focused in alternative learning methods that efficiently sample interesting regions of the state space and exploit the ergodic nature of the problems. Extension to more complex scenarios is also being considered.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number 25540085 and by the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreement 270327 (CompLACS).

References

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Comput.*, 12(1):219–245.

Ghahramani, Z. and Jordan, M. (1997). Factorial hidden markov models. *Mach. Learn.*, 29(2-3):245–273.

Kappen, H. J., Gómez, V., and Opper, M. (2012). Optimal control as a graphical model inference problem. *Mach. Learn.*, 87(2):159–182.

Kinjo, K., Uchibe, E., and Doya, K. (2013). Evaluation of linearly solvable Markov decision process with dynamic model learning in a mobile robot navigation task. *Front. Neurorobot.*, 7:1–13.

Mitrovic, D., Nagashima, S., Klanke, S., Matsubara, T., and Vijayakumar, S. (2010). Optimal feedback control for anthropomorphic manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'10)*, pages 4143–4150.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Sugiyama, M., Hachiya, H., Towell, C., and Vijayakumar, S. (2007). Value function approximation on non-linear manifolds for robot motor control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'07)*, pages 1733–1740.

Sutton, R. S. and Barto, A. G. (1998). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition.

Theodorou, E., Buchli, J., and Schaal, S. (2010). Reinforcement learning of motor skills in high dimensions: A path integral approach. In *Proceedings of the IEEE*

International Conference on Robotics and Automation (ICRA'10), pages 2397–2403.

Todorov, E. (2006). Linearly-solvable markov decision problems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1369–1376.

Todorov, E. (2009a). Efficient computation of optimal actions. *PNAS*, 106(28):11478–11483.

Todorov, E. (2009b). Eigenfunction approximation methods for linearly-solvable optimal control problems. In *Proceedings of the 2nd IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 161–168.

Yoshikawa, T. (1990). *Foundations of Robotics: Analysis and Control*. The MIT Press.

Zhong, M. and Todorov, E. (2011a). Aggregation methods for linearly-solvable Markov decision process. In *World Congress of the International Federation of Automatic Control*, pages 11220–11225.

Zhong, M. and Todorov, E. (2011b). Moving least-squares approximations for linearly-solvable stochastic optimal control problems. *J. Control. Theory. Appl.*, 9(3):451–463.