Optimal Limit-Cycle Control recast as Bayesian Inference

Yuval Tassa * **Tom Erez** ** **Emanuel Todorov** ***

** Interdisciplinary Center for Neural Computation, Hebrew University, Jerusalem, Israel, (e-mail: tassa@alice.nc.huji.ac.il).*
*** Department of Computer Science, Washington University, St. Louis, MO, USA, (e-mail: etom@cse.wustl.edu)*
**** Department of Computer Science and Engineering and the Department of Applied Mathematics, University of Washington Seattle, WA, USA, (e-mail: todorov@cs.washington.edu)*

Abstract: We introduce an algorithm that generates an optimal controller for stochastic nonlinear problems with a periodic solution, e.g. *locomotion.* Uniquely, the quantity we approximate is neither the Value nor Policy functions, but rather the stationary state-distribution of the optimally-controlled process. We recast the control problem as Bayesian inference over a graphical model with a ring topology. The posterior approximates the controlled stationary distribution with local gaussians along the optimal limit-cycle. Linear-feedback gains and open-loop controls are extracted from the covariances and the means, respectively. Complexity scales linearly or quadratically with the state dimension, depending on the dynamics approximation. We demonstrate our algorithm on a toy 2-dimensional problem and then on a challenging 23-dimensional simulated walking robot.

Keywords: optimal control, limit cycles, feedback control, control-estimation dualities.

1. INTRODUCTION

Optimal control of complex movements is of interest in engineering as well as biology. There are scientific reasons to believe that the brain optimizes motor behavior (Todorov, 2004), and engineering reasons to wish we had algorithms capable of doing the same for robots and other synthetic systems. A variety of global algorithms have been developed to approximate the optimal control law or cost-to-go function using predefined features (Sutton, 1998; Bertsekas, 2000). However these algorithms are only as good as the features provided to them, and a reliable procedure for choosing good features does not yet exist. Indeed the only algorithms that can presently be expected to work off-the-shelf for high-dimensional nonlinear systems are local methods. They generate either an open-loop trajectory, as in Pontryagin's maximum principle and pseudo-spectral methods (Stengel, 1994; Ross and Fahroo, 2004), or a trajectory and a local feedback control law, as in Differential Dynamic Programming (Jacobson and Mayne, 1970) and iterative linear-quadratic-Gaussian control (Todorov and Li, 2005). The local nature of these methods is of course a limitation, however many interesting behaviors involve stereotypical movements, and the ability to discover those movements and generate them in a stable manner is very useful.

This paper aims to address one of the major shortcomings of local methods – which is that they are limited to finite-horizon problem formulations. Instead we would like to have methods with similar efficiency but capable of solving infinite-horizon problems, in particular problems that give rise to complex periodic movements such as walking, running, swimming, flying (with wings), turning a screwdriver, etc. This requires optimization over cycles. Such optimization is difficult to cast as an optimal control problem because the underlying system becomes non-Markov. Here we overcome this difficulty by replacing the control problem with a dual Bayesian inference problem, and performing inference over a graphical model with loops. We use linear-Gaussian machinery (as in DDP and iLQG), thus when the dynamics are nonlinear we have to solve a sequential Bayesian inference problem: the solution at each iteration is used to re-linearize the system and define the inference problem for the next iteration. When the algorithm converges, the mean of the posterior gives the locally-optimal trajectory while the covariance of the posterior gives the local feedback control law. Since computing the correct covariance is important here, we perform inference using the variational approach of Mitter and Newton (2004), leading to an algorithm based on sparse matrix factorization rather than loopy belief propagation (where only the mean is guaranteed to be correct, see Weiss and Freeman (2001)).

The estimation-control duality which is at the heart of our new method arises within the recently-developed framework of linearly-solvable optimal control (Kappen, 2005; Todorov, 2009). Several control algorithms exploiting this duality (sometimes implicitly) have been developed (Attias, 2003; Toussaint, 2009; Kappen et al., 2009), however they are limited to finite-horizon formulations – which DDP and iLQG can already handle, with comparable efficiency as far as we can tell. The present paper exploits the estimation-control duality in more general graph structures for the first time.

2. RELATED WORK

The method presented below is related to three lines of research.

The first is classic local trajectory-optimization methods, such as the Maximum Principle of Pontryagin et al. (1962) and Differential Dynamic Programming of Jacobson and Mayne (1970). It is possible to use these methods to solve for limit cycles by "attaching" the first and last states. This can be done either approximately, by imposing a final-cost over distance from the initial state, or exactly, by employing multipliers which enforce the state constraint, as in the method of Lantoine and Russell (2008). We tried both of these approaches, and the inevitable result was a noticeable asymmetry around the attachment point, either in the state trajectory (when using final-cost), or in the controls (when using multipliers). The main insight is that these algorithms assume Markovity, which does not hold for a loop. One could also use a finite-horizon method with a very long horizon, that loops around the limit-cycle several times. By truncating the transients at both ends, we can get a decent approximation to the infinite-horizon solution. Clearly this is an inefficient use of computational resources, but can serve as a useful validation procedure for our algorithm.

The second related body of work involves directly optimizing the total cost of a limit-cycle, while enforcing periodicity. Wampler and Popovic (2009) and Ackermann and den Bogert (2010) are two recent examples, respectively from the computer graphics and biomechanics communities. The log-likelihood that we end up maximizing below is indeed analogous to such a cost, however our method generates a feedback controller around the limit-cycle, rather than simply open-loop controls.

Finally, the last several years have seen research into the subclass of stochastic nonlinear Optimal Control problems which are dual to Bayesian estimation. Specifically, Toussaint (2009) explores message-passing algorithms (Expectation Propagation) for the solution of Optimal Control problems. Murphy et al. (1999) and others have shown that when a graph has a loopy structure, message passing converges to the right mean but the wrong covariance. The procedure we describe below does not suffer from this drawback.

3. OPTIMAL CONTROL VIA BAYESIAN INFERENCE

The basic intuition behind the duality we exploit here is that the negative log-likelihood in estimation corresponds to a state-dependent cost in control, and the difference (KL divergence) between the prior and the posterior corresponds to a control-dependent cost. The class of stochastic optimal control problems which have Bayesian inference duals in the above sense have received a lot of attention recently, because these problems have a number of other interesting properties, including the fact that the (Hamilton-Jacobi) Bellman equation becomes linear after exponentiation (Kappen, 2005; Todorov, 2008).

3.1 Background on LMDPs and inference-control dualities

A linearly-solvable MDP (or LMDP) is defined by a state cost $q(x) \geq 0$ and a transition probability density $p(x'|x)$ corresponding to the notion of passive dynamics. The controller is free to specify any transition probability density $\pi(x'|x)$ with the restriction that $\pi(x'|x) = 0$ whenever $p(x'|x) = 0$. In infinite-horizon average-cost problems p, π are further required to be ergodic. The cost rate function is

$$\ell(x, \pi(\cdot|x)) = q(x) + D_{\mathrm{KL}}[\pi(\cdot|x) || p(\cdot|x)]$$

The KL divergence term is a control cost which penalizes deviations from the passive dynamics. Defining the *desirability* function $z(x) \triangleq \exp(-v(x))$ where $v(x)$ is the optimal cost-to-go, the optimal control is

$$\pi(x'|x) \propto p(x'|x)\, z(x')$$

The exponentiated Bellman equation becomes linear in z. In particular, for finite horizon problems this equation is

$$z_t(x) = \exp(-q(x)) \sum\nolimits_{x'} p(x'|x)\, z_{t+1}(x')$$

with $z_N(x)$ initialized from the final cost. One can also define the function $r(x)$ as the solution to the transposed equation:

$$r_{t+1}(x') = \sum\nolimits_{x} \exp(-q(x))\, p(x'|x)\, r_t(x)$$

with $r_0(x)$ being a delta function over the fixed initial state. Then the marginal density under the optimally-controlled stochastic dynamics can be shown to be

$$\mu_t(x) \propto r_t(x)\, z_t(x)$$

The duality to Bayesian inference is now clear: z is the backward filtering density, r is the forward filtering density, p is the dynamics prior, $q(x)$ is the negative log-likelihood (of some unspecified measurements), and μ is the marginal of the Bayesian posterior. We can also write down the density p^* over trajectories generated by the optimally-controlled stochastic dynamics, and observe that it matches the Bayesian posterior over trajectories in the estimation problem:

$$p^*(x_1, x_2, \cdots x_N|x_0) \propto \prod\nolimits_{t=1}^{N} \exp(-q(x_t))\, p(x_t|x_{t-1})$$

These LMDPs can be used to model the continuous systems we are primarily interested in as follows. It has been shown that for controlled Ito diffusions in the form

$$dx = a(x)\,dt + B(x)(u\,dt + \sigma d\omega) \qquad (1)$$

and cost functions in the form

$$\ell(x, u) = q(x) + \frac{1}{2\sigma^2}\|u\|^2 \qquad (2)$$

the stochastic optimal control problem is a limit of continuous-state discrete-time LMDPs. The LMDP passive dynamics are obtained via explicit Euler discretization with time step h:

$$p(x'|x) = \mathcal{N}\left(x + ha(x),\ h\,\sigma^2 B(x)\, B(x)^{\mathsf{T}}\right) \qquad (3)$$

where $\mathcal{N}$ denotes a Gaussian. The LMDP state cost is simply $hq(x)$. Note that the h-step transition probability of the controlled dynamics (with $u \neq 0$) is a Gaussian with the same covariance as (3) but the mean is shifted by $h\,B(x)\,u$. Using the formula for KL divergence between Gaussians, the general KL divergence control cost reduces to a more traditional control cost quadratic in u.

3.2 Periodic optimal control as Bayesian inference

Our goal now is to write down the trajectory probability p^* for infinite-horizon average-cost problems, and then interpret it as a Bayesian posterior. This cannot be done exactly, because here (3.1) involves infinitely-long trajectories which we cannot even represent unless they are periodic. Therefore we will restrict the density to the subset of periodic trajectories with period N. This of course is an approximation, but the hope is that most of the probability mass lies in the vicinity of such trajectories. Then $p^*(x_1, x_2, \cdots x_N)$ is the same as (3.1), except we have now defined $x_0 = x_N$.

While the trajectory probability for the control problem is no longer exact, (3.1) is still a perfectly valid Bayesian posterior for a graphical model with a loop. More precisely, $\exp(-q(x_t))$ are single-node potentials which encode evidence, while $p(x_t|x_{t-1})$ are pair-wise potentials which encode the prior. One caveat here is that, since the state space is continuous, the density may not be integrable. In practice however we approximate $p(x_t|x_{t-1})$ with a Gaussian, so integrability comes down to making sure that the joint covariance matrix is positive definite – which can be enforced in multiple ways (see below).

Once the Bayesian posterior over limit-cycle trajectories is computed, we need to recover the underlying control law for the stochastic control problem. The obvious approach is to set

$$\pi(x_t|x_{t-1}) = \frac{p^*(x_t, x_{t-1})}{p^*(x_{t-1})}$$

where $p^*(x_t, x_{t-1})$ and $p^*(x_{t-1})$ are the corresponding marginals of the trajectory probability, and then recover the physical control signal $u(x_t)$ by taking the mean. However this yields N different conditional distributions, and we need to somehow collapse them into a single conditional $\pi(x'|x)$ because the control problem we are solving is time-invariant. We have explored the two obvious ways to do the combination: average the π's weighted by the marginals $p^*(x_t)$, or use the π corresponding to the nearest neighbor. Empirically we found that averaging blurs the density too much, while the nearest neighbor approach works well. An even better approach is to combine all the $p^*(x_t, x_{t-1})$ into a mixture density, and then compute the conditional $\pi(x'|x)$ of the entire mixture. This can be done efficiently when the mixture components are Gaussians.

4. ALGORITHM

Probabilistic graphical models (Jordan, 1998) are an efficient way of describing conditional independence structures. A cycle-free directed graph (a tree) represents a joint probability as a product of conditionals

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\text{parents}(x_k))$$

This equation represents the factorization properties of p. *Message Passing* algorithms, which involve sequentially propagating local distributions along directed graphs, provably converge to the true posterior.

An alternative to directed graphical models are Markov Fields, whose graph is undirected, and may contain cycles. The joint distribution of a Markov Field is given by

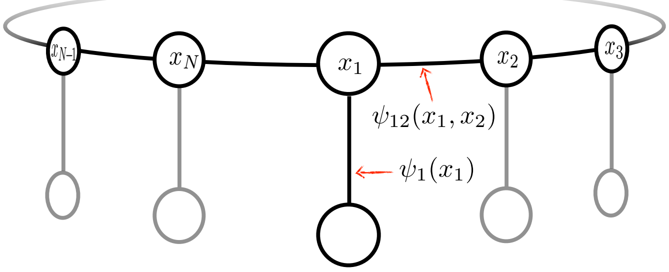


Fig. 1. Illustration of probabilistic graphical model. State costs are encoded in the leaf potentials $\psi_i(x_i)$. Dynamics and control costs are encoded in the edge potentials $\psi_{ij}(x_i, x_j)$. See section 4.1.

$$p(\mathbf{x}) \propto \prod_{c \in C} \psi_c(x_c),$$

where C is the set of maximal cliques in the graph, and the ψ are called *potential functions*. Message Passing algorithms are not guaranteed to converge on this type of model. In particular, for models where the nodes are distributed as gaussians (as we will assume below), Weiss and Freeman (2001) had shown that posteriors and marginals converge to correct means, but not to the correct variances.

4.1 Potential Functions

Let $\{x_i\}_{i=1}^N$ be a set of state variables $x_i \in I\!R^n$, with the conditional dependency structure of a cycle. Let ij index over the pairs of sequential states $ij \in \{(1,2),(2,3),\ldots,(N,1)\}$. The discrepancy between the controlled dynamics and the discrete-time passive dynamics for each pair is

$$a_{ij} = x_i + ha(x_i) - x_j.$$

The gaussian noise leads to pairwise potentials, corresponding to $p(x_j|x_i)$,

$$\psi_{ij}(x_i, x_j) = p(x_j|x_i) = \exp(-\tfrac{1}{2} a_{ij}^\mathsf{T} \Sigma_i^{-1} a_{ij}),$$

where $\Sigma_i = \ h\,\sigma^2 B(x_i)\, B(x_i)^\mathsf{T}$, as in (3).

The leaf potentials $\psi_i(x_i)$ are composed of two parts, the state-cost $q(x_i)$, and an optional prior on x_i. This prior, not used below, could be useful when we wish to clamp certain states to specified values. For example, in the finite-horizon case where the graph is a chain, we could place a gaussian prior on a known initial state

$$\psi_1(x_1) = \exp(-q(x_1))\mathcal{N}(x_1|m_1, \Sigma_1).$$

The joint distribution of the entire model is

$$p(\mathbf{x}) = p(x_1, x_2, \ldots, x_N) \sim \prod_{i=1}^{N} \psi_i(x_i) \prod_{ij=1}^{N} \psi_{ij}(x_i, x_j)$$

Where $\mathbf{x} = \text{stack}\{x_i\} = [x_1^\mathsf{T} x_2^\mathsf{T} \cdots x_N^\mathsf{T}]^\mathsf{T}$ is the stacked vector of all states. The negative log-likelihood is

$$l(\mathbf{x}) = \sum_i q_i(x_i) + \sum_{ij} \tfrac{1}{2} a_{ij}^\mathsf{T} \Sigma_i^{-1} a_{ij}. \qquad (4)$$

The first term is the total state-cost and the second term is the total control-cost.

4.2 Gaussian approximation

Modeling $p(\cdot)$ as a gaussian: $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|\bar{\mathbf{x}},\mathbf{S})$, is equivalent to fitting a quadratic model to the negative log likelihood.

$$l(\mathbf{x}) \approx \tfrac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^\mathsf{T}\mathbf{S}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) = l_0 + \mathbf{x}^\mathsf{T}\mathbf{g} + \tfrac{1}{2}\mathbf{x}^\mathsf{T}H\mathbf{x}. \quad (5)$$

The normalization term $\frac{1}{2}\log(\det(2\pi\mathbf{S}))$ is folded into the constant l_0. The mean $\bar{\mathbf{x}}$ (which maximizes the likelihood) and the covariance $\mathbf{S}$ are given by

$$\bar{\mathbf{x}} = -H^{-1}\mathbf{g} \quad (6a)$$

$$\mathbf{S} = H^{-1} \quad (6b)$$

4.3 Iterative inference

Given a current approximation to the mean $\bar{\mathbf{x}}$ of the Bayesian posterior over trajectories, we expand $l(\bar{\mathbf{x}}+\delta\mathbf{x})$ to second-order in $\delta\mathbf{x}$ by computing H and g, and then find a new mean as

$$\bar{\mathbf{x}}' = \bar{\mathbf{x}} + \operatorname*{argmin}_{\delta\mathbf{x}} l(\bar{\mathbf{x}}+\delta\mathbf{x}) = \bar{\mathbf{x}} - H(\bar{\mathbf{x}})^{-1}\mathbf{g}(\bar{\mathbf{x}}), \quad (7)$$

until convergence. The actual inversion of the precision matrix or Hessian H, as required by (6b), can be performed only once, at the end. During the iterations of (7), we can use iterative methods (e.g. preconditioned conjugate gradients) to solve $H\mathbf{y} = \mathbf{g}$, which are very cheap for sparse systems. Again, this process can be interpreted either as repeated estimation of a joint gaussian model, or sequential quadratic minimization of the total cost.

4.4 Dynamics model

We now turn to the computation of H and $\mathbf{g}$. Placing the cost Hessians on a the block-diagonal of the matrix $Q = \operatorname{diag}\{\frac{\partial^2}{\partial x^2}q_i(\bar{x}_i)\} \in \mathbb{R}^{nN\times nN}$ and stacking the local cost gradients $q_\mathbf{x} = \operatorname{stack}\{\frac{\partial}{\partial x}q_i(\bar{x}_i)\} \in \mathbb{R}^{nN}$, the last term of (4) can be approximated

$$\sum_i q_i(\bar{x}_i+\delta x_i) \approx \sum_i q_i(\bar{x}_i) + \delta\mathbf{x}^\mathsf{T}q_\mathbf{x} + \tfrac{1}{2}\delta\mathbf{x}^\mathsf{T}Q\delta\mathbf{x}$$

In order to quadratize the last term of (4), we must approximate the nonlinear dynamics (or rather the dynamic discrepancies a_{ij}). We can do this using either a linear or a quadratic model. The former is faster to compute at each iteration, while the latter is more accurate and thus could yield convergence in fewer iterations. Which approach is better probably depends on the problem; in the examples given below we found that the quadratic model works better.

Linear dynamics approximation: We expand the dynamic discrepancies to first order around our current approximation,

$$a_{ij}(\bar{x}_i+\delta x_i, \bar{x}_j+\delta x_j) = \bar{a}_{ij} + a_x(\bar{x}_i)\delta x_i - \delta x_j.$$

We construct the sparse matrix $A \in \mathbb{R}^{nN\times nN}$ as a stack of N block-rows of dimension $n \times nN$. For each pair ij, we place a negative identity matrix $-I_n$ on the j-th column-block and the dynamics Jacobians $a_x(x_i)$ on the i-th column-block. Additionally letting $\mathbf{a} = \operatorname{stack}\{a_i\} \in \mathbb{R}^{nN}$, we have in matrix form

$$\mathbf{a}(\bar{\mathbf{x}}+\delta\mathbf{x}) = \bar{\mathbf{a}} + A\delta\mathbf{x}.$$

Defining $M = \operatorname{diag}\{\Sigma_i^{-1}\} \in \mathbb{R}^{nN\times nN}$, the last term of (4) becomes $\frac{1}{2}(\bar{\mathbf{a}}+A\delta\mathbf{x})^\mathsf{T}M(\bar{\mathbf{a}}+A\delta\mathbf{x})$, and the second-order expansion around $\bar{\mathbf{x}}$ is seen to be

$$l(\bar{\mathbf{x}}+\delta\mathbf{x}) = l(\bar{\mathbf{x}}) + \delta\mathbf{x}^\mathsf{T}(q_\mathbf{x}+A^\mathsf{T}M\bar{\mathbf{a}}) + \tfrac{1}{2}\delta\mathbf{x}^\mathsf{T}(Q+A^\mathsf{T}MA)\delta\mathbf{x}.$$

Comparison with (5) shows that

$$\mathbf{g} = q_\mathbf{x} + A^\mathsf{T}M\bar{\mathbf{a}} \quad (8a)$$

$$H = Q + A^\mathsf{T}MA \quad (8b)$$

Quadratic dynamics approximation: We can achieve a more accurate approximation by considering a quadratic model of the passive dynamics

$$a_{ij}(\bar{x}_i+\delta x_i, \bar{x}_j+\delta x_j) = \bar{a}_{ij} + a_x(\bar{x}_i)\delta x_i + \tfrac{1}{2}\delta x_i^\mathsf{T}a_{xx}(\bar{x}_i)\delta x_i - \delta x_j,$$

where the left and right multiplications with the 3-tensor a_{xx} are understood as contractions on the appropriate dimensions. Though the gradient $\mathbf{g}$ is unaffected by the second order term, the Hessian picks up the product of second-order and zeroth-order terms. Let the set of $n \times n$ matrices $U_{ij} = a_{ij}^\mathsf{T}\Sigma_i^{-1}a_{xx}(\bar{x}_i)$, contracting with the leading dimension of the tensor a_{xx}. Now define the block-diagonal matrix $U = \operatorname{diag}\{U_{ij}\} \in \mathbb{R}^{nN\times nN}$. The new approximation is now

$$\mathbf{g} = q_\mathbf{x} + A^\mathsf{T}M\bar{\mathbf{a}} \quad (9a)$$

$$H = Q + A^\mathsf{T}MA + U \quad (9b)$$

In the experiments described below, the addition of the U term was found to significantly improve convergence.

4.5 Computing the policy

In this section, we describe how to obtain a local feedback control policy from the posterior marginals, that is the means $\bar{x}_i$ and the covariance $\mathbf{S}$. Let $S_i = \operatorname{cov}(x_i)$ be the i-th diagonal $n\times n$ block of $\mathbf{S}$ and S_{ij} be the cross-covariance of x_i and x_j, the $n \times n$ block of $\mathbf{S}$ at the i-th n-column and j-th n-row, so that the mean of the conditional is

$$E[x_j|x_i] = \bar{x}_j + S_{ij}S_i^{-1}(x_i - \bar{x}_i).$$

The feedback policy is then that control which produces the expected controlled dynamics:

$$u(x_i) = B^{-1}(\bar{x}_j + S_{ij}S_i^{-1}(x_i - \bar{x}_i) - a(\bar{x}_i)) \quad (10)$$

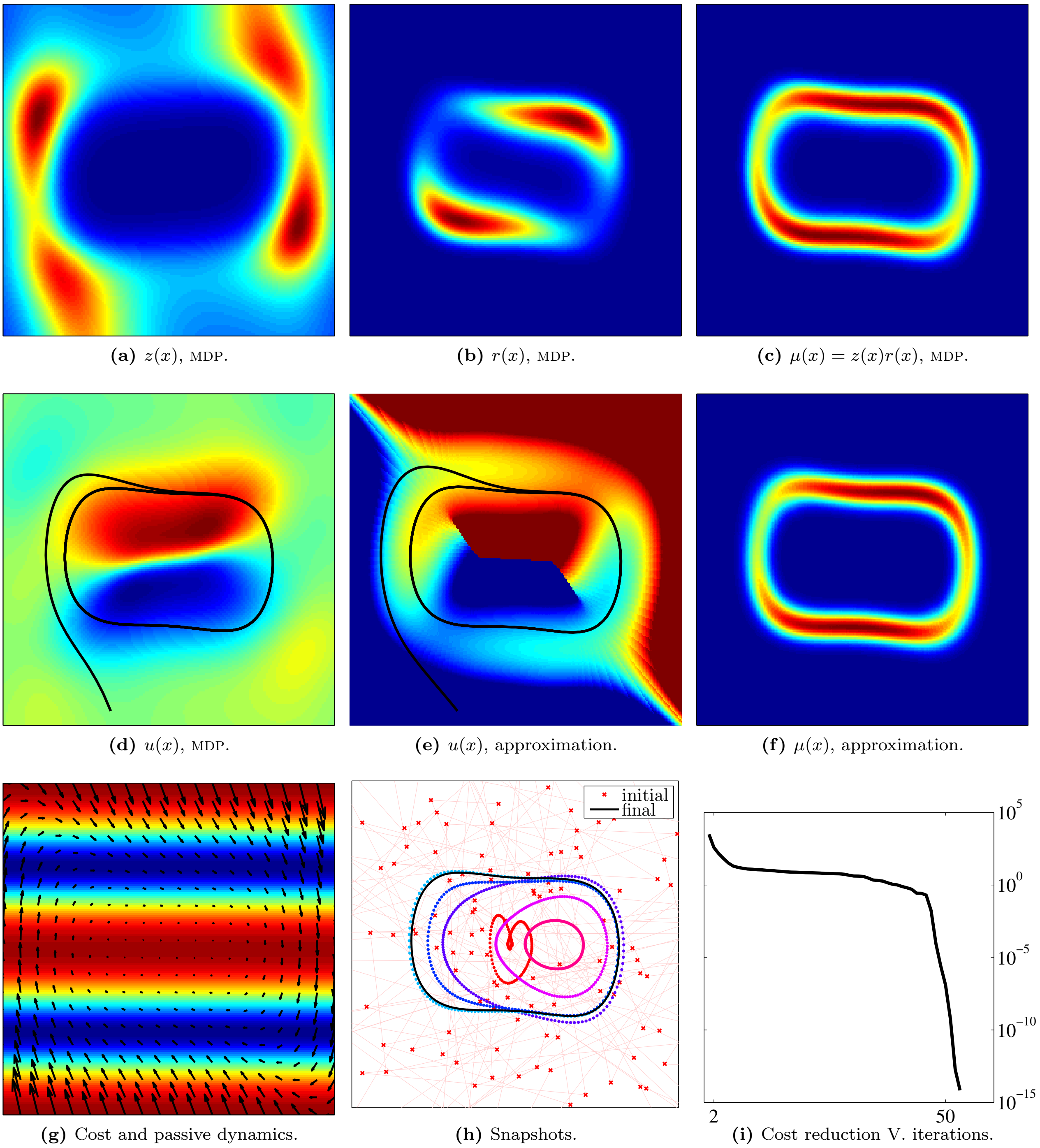
4.6 Algorithm summary

Given an initial approximation of $\bar{\mathbf{x}}$:

```
(a) Repeat until convergence:
        Compute $\mathbf{g}(\bar{\mathbf{x}})$ and $H(\bar{\mathbf{x}})$ with (9).
        Recompute $\bar{\mathbf{x}}$ with (7).
(b) Compute $\mathbf{S}$ with (6b), and the feedback control law
    with (10).
```

5. EXPERIMENTS

We demonstrate our algorithm on two simulated problems. A toy problem with 2 state dimensions and and a simulated walking robot with 23 state dimensions.



(a) $z(x)$, MDP. **(b)** $r(x)$, MDP. **(c)** $\mu(x) = z(x)r(x)$, MDP.

(d) $u(x)$, MDP. **(e)** $u(x)$, approximation. **(f)** $\mu(x)$, approximation.

(g) Cost and passive dynamics. **(h)** Snapshots. **(i)** Cost reduction V. iterations.

Fig. 2. **(a)-(h)** show the area $[-4, 4]^2 \in (x_1 \times x_1)$. **(a)-(d)**, MDP solutions for a discretized state-space. **(e)**, **(f)**, **(h)**, **(i)**, solution obtained by the proposed algorithm. **(a)** The exponentiated negative value function $z(x)$. **(b)** The forward filtering density $r(x)$. **(c)** The optimally controlled steady-state distribution, formed by elementwise product of $z(x)$ and $r(x)$. **(d)** The policy generated by the MDP solution, superimposed with one instantiation of the controlled dynamics. **(e)** The policy generated by the approximate solution, superimposed with one instantiation of the controlled dynamics. The color map is clipped to the values in (d), so saturated areas indicate misextrapolation. **(f)** The approximate distribution generated by our algorithm. For each pixel we measure the marginal of the state whose mean is nearest in the Euclidean sense. Note the similarity to (c). **(g)** The cost function, superimposed with a vector field of the passive dynamics. **(h)** Snapshots of the means for a particular convergence sequence, showing 8 configurations out of a total of 40. The red x's are the random initialization, followed by a jump to the center to decrease dynamical inconsistency, followed by a gradual convergence to the limit-cycle solution. **(i)** Convergence of the cost. Averaged over 15 runs with random initialization.

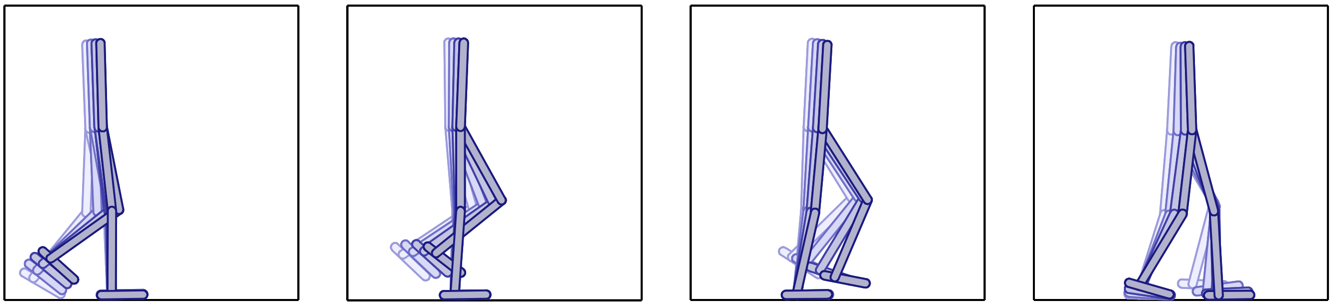


Fig. 3. Frames from the limit-cycle solution of an optimal walking gait. See section 5.2.

5.1 2D problem

The continuous diffusion consists of a non-linear spring damper system, subject to process noise in the velocity variable:

$$\begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -(x_1^3 + x_2^3)/6 \end{bmatrix} dt + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (udt + \sigma d\omega)$$

and cost function is

$$\ell(x_2, u) = c_x \left(1 - e^{-(x_2-2)^2} - e^{-(x_2+2)^2} \right) + \frac{u^2}{2\sigma^2}$$

The state cost coefficient is $c_x = 4$ and the noise variance is $\sigma^2 = 1/4$. In Figure 2(g), we show the cost function, overlayed by a vector plot of the drift, and one integrated trajectory of the passive dynamics.

We first solved this problem by discretizing the state-space and solving the resulting MDP. We used a 201×201 grid, leading to a 40401×40401 state transition matrix with 1.2×10^6 nonzeros. Discrete LMDPs can be solved by finding the leading eigenvector of a related matrix (Todorov, 2007). The results are shown in Figures 2(a)-2(d). Solving the MDP (using MATLAB's "eigs" function) took $86s$ on a standard PC.

We then solved the problem with our proposed algorithm. With 150 variables on the ring, the matrix H was 300×300, with 1800 nonzeros. Full convergence from a random initialization took an average $0.3s$. Of course this is not a fair comparison, since the MDP solver finds a global rather than local solution, yet the difference is striking. Once convergence of equation (7) has been achieved, we compute the posterior with (6). In order to plot the resulting distribution, for every pixel in Figure 2(f), we plot the value of the marginal of the closest (euclidean) gaussian. The similarity of figures 2(c) and 2(f) is remarkable. Of course, our proposed method is still local, and comparing Figures 2(d) and 2(e), we see that the generated policy is valid only close to the limit-cycle. In Figure 2(h), we see snapshots of the convergence of the means. Starting from a random initialization, the means jump to the center in order to decrease dynamical inconsistency, followed by a gradual convergence to the limit-cycle solution. In Figure 2(i), we show the cost averaged over 15 runs, relative to the minimum cost achieved over all the runs. We see that all runs converged to the global minimum, with a quadratic convergence rate towards the end.

5.2 Simulated walking robot

Our planar walking model is made of two legs and a trunk, each leg having three segments (thigh, shin and foot). The following parameter values can all be assumed to have the appropriate units of a self-consistent system (e.g. MKS). The length of the foot segments is 1, and all other segments are of length 2. The segment masses are 0.1 for the foot, 0.4 for the shin, 1 for the thigh, and 4 for the trunk. A control signal of dimension 6 acts on the joints (hips, knees, ankles), but not directly. In order to model the excitation-activation dynamics associated with muscular activity, we augment the state space with 6 first-order filters of the control signal, with a time constant of 25/1000. The seven segment angles, together with the planar position of center-of-mass, make for a system with 9 degrees-of-freedom, or 18 state dimensions. In order to allow the gait to take a limit-cycle form, we remove the horizontal position dimension of the center-of-mass, for a total of 23 state dimensions.

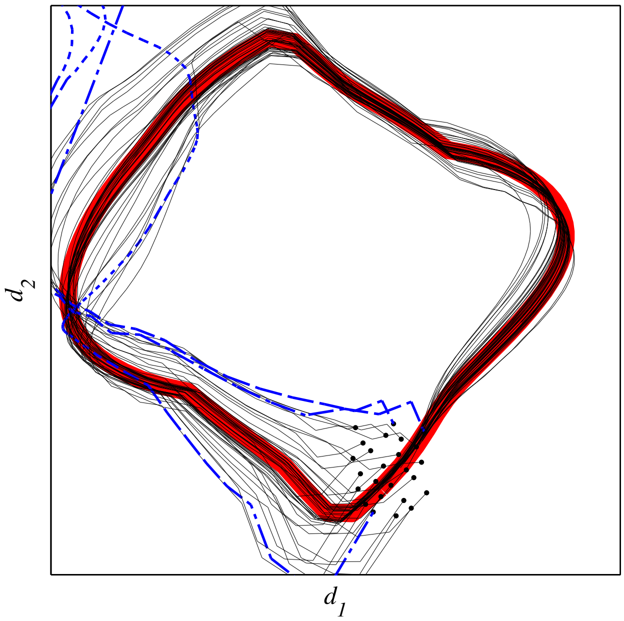


Fig. 4. Robustness to perturbations of the simulated walking robot, under the feedback control law. The axes d_1 and d_2 are the largest eigenvectors of the covariance of the $\bar{x}_i$. The optimal limit cycle is in thick red, superimposed with simulated trajectories. Converging trajectories are thin solid lines, diverging ones are thicker dashed lines. See section 5.3.

The equations of motion are simulated using our own general purpose simulator [1] . We imposed joint-angle constraints that ensure a biomechanically-realistic posture. Ground reaction forces are computed using the method described by Tassa and Todorov (2010). We used 80 time-steps of length 1/80, for a step period of 1. The matrix H was 1840×1840, with 105840 non-zeros. Each iteration took 0.2 seconds, with an average of 300 iterations until convergence (depending on initial conditions).

In order to produce upright walking, we use a cost function with three terms: First, a quadratic penalty for deviation of the center-of-mass's horizontal velocity v_x from a desired value of 2. Second, a linear reward for the vertical hight of the trunk's upper tip h_T, to promote upright posture. Third, a cost term that is quadratic in the muscle activation dimensions c. The total weighted state-cost was

$$q(x) = (v_x - 2)^2 - 0.1 h_T + 0.01\|c\|^2. \qquad (11)$$

Convergence for this problem was robust, with different initializations converging to the same solution. The resulting gait is demonstrated in Figure 3.

5.3 Feedback control law

One of the main advantages of the algorithm presented here is that the generated control law (10) includes feedback terms, forming an effective basin-of-attraction around the optimal limit-cycle. In Figure 4, we illustrate convergence to the limit cycle from perturbed states, of the simulated walking robot. The means $\bar{x}_i$ are projected onto $\{d_1, d_2\} \in \mathcal{R}^{23}$, the two leading eigenvectors of the covariance $\mathrm{cov}(\bar{x}_i)$ (i.e. the two leading PCA directions). One state is then randomly perturbed, and used as an initial state for a controlled trajectory. The distribution of the perturbations was chosen so that most trajectories (solid) are within the basin-of-attraction and converge to the limit-cycle (successful walking), while several (dashed) diverge (falling down).

6. DISCUSSION

We presented a method of solving Optimal Control problems with a periodic solution. Using the control-estimation duality which holds for problems of the form (1,2), we recast the problem as Bayesian inference, and maximized the likelihood of a gaussian approximation. The computational complexity of the methods scales either linearly or quadratically with the state dimension, depending on the order of the dynamics approximation.

Several interesting questions remain open.

In this paper we focused on limit cycles, but the presented algorithm is also valid for simple chains, where message passing algorithms of the type described by Toussaint (2009) are applicable. A performance comparison would be interesting.

The algorithm produces a local feedback control law around the trajectory, but the volume of the basin-of-attraction is limited by the validity of the gaussian approximation. One way of increasing it is by using higher-order approximations, like a mixture-of-gaussians. Another possibility is to combine this offline method with online methods like Model Predictive Control, by using the infinite-horizon cost-to-go as a final-cost for a finite-horizon trajectory optimizer.

ACKNOWLEDGEMENTS

We would like to thank Dvijotham Krishnamurthy for many insightful discussions. This work was supported by the US National Science Foundation.

REFERENCES

Ackermann, M. and den Bogert, A.J.V. (2010). Optimality principles for model-based prediction of human gait. *Journal of biomechanics*.

Attias, H. (2003). Planning by probabilistic inference. In *Proc. of the 9th Int. Workshop on Artificial Intelligence and Statistics*.

Bertsekas, D. (2000). *Dynamic programming and optimal control*. Athena Scientific, Belmont Mass., 2nd ed. edition.

Jacobson, D.H. and Mayne, D.Q. (1970). *Differential Dynamic Programming*. Elsevier.

Jordan, M.I. (1998). *Learning in graphical models*. Kluwer Academic Publishers.

Kappen, B., Gomez, V., and Opper, M. (2009). Optimal control as a graphical model inference problem. *arXiv*, 901.

Kappen, H.J. (2005). Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005, P11011.

Lantoine, G. and Russell, R.P. (2008). A hybrid differential dynamic programming algorithm for robust low-thrust optimization. In *AAS/AIAA Astrodynamics Specialist Conference and Exhibit*.

Mitter, S.K. and Newton, N.J. (2004). A variational approach to nonlinear estimation. *SIAM Journal on Control and Optimization*, 42(5), 18131833.

Murphy, K., Weiss, Y., and Jordan, M.I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in AI*, 467475.

Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., and Mishchenko, E.F. (1962). *The mathematical theory of optimal processes*. Interscience New York.

Ross, I. and Fahroo, F. (2004). Legendre pseudospectral approximations of optimal control problems. In *New Trends in Nonlinear Dynamics and Control and their Applications*, 327–342.

Stengel, R.F. (1994). *Optimal Control and Estimation*. Dover Publications.

Sutton, R. (1998). *Reinforcement learning : an introduction*. MIT Press, Cambridge Mass.

Tassa, Y. and Todorov, E. (2010). Stochastic complementarity for local control of discontinuous dynamics. In *Proceedings of Robotics: Science and Systems (RSS)*.

Todorov, E. (2007). Linearly-solvable markov decision problems. *Advances in neural information processing systems*, 19, 1369.

Todorov, E. (2008). General duality between optimal control and estimation. In *proceedings of the 47th ieee conf. on decision and control*.

Todorov, E. (2009). Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences*, 106(28), 11478.

[1] Available at alice.nc.huji.ac.il/~tassa/

Todorov, E. and Li, W. (2005). A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, 300–306. Portland, OR, USA. doi:10.1109/ACC.2005.1469949.

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9), 907–915. doi:10.1038/nn1309.

Toussaint, M. (2009). Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

Wampler, K. and Popovic, Z. (2009). Optimal gait and form for animal locomotion. *ACM Transactions on Graphics (TOG)*, 28(3), 18.

Weiss, Y. and Freeman, W.T. (2001). Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural computation*, 13(10), 21732200.