EP for Efficient Stochastic Control with Obstacles

Thomas Mensink[1] and **Jakob Verbeek**[2] and **Bert Kappen**[3]

Abstract. We address the problem of continuous stochastic optimal control in the presence of hard obstacles. Due to the non-smooth character of the obstacles, the traditional approach using dynamic programming in combination with function approximation tends to fail. We consider a recently introduced special class of control problems for which the optimal control computation is reformulated in terms of a path integral. The path integral is typically intractable, but amenable to techniques developed for approximate inference.We argue that the variational approach fails in this case due to the non-smooth cost function. Sampling techniques are simple to implement and converge to the exact results given enough samples. However, the infinite cost associated with hard obstacles renders the sampling procedures inefficient in practice. We suggest Expectation Propagation (EP) as a suitable approximation method, and compare the quality and efficiency of the resulting control with an MC sampler on a car steering task and a ball throwing task. We conclude that EP can solve these challenging problems much better than a sampling approach.

1 Introduction

Recent research has focussed on developing algorithms that increase the level of autonomy of autonomous systems, such as UAVs, rescue robots or the Mars rover. A critical problem, however, which prevents the widespread adoption of these methods outside of simulation, is a lack of robustness. Existing control algorithms typically do not take into account the inherent uncertainty that arises in the real world, which can lead to catastrophic failure.

Many real life systems have constraints on the allowed states. For example a joint of a robot-arm has a restriction on its angle, or a car has a restriction on the road it uses. We incorporate these restrictions as hard obstacles in the cost function of the system. Optimal control in the presence of Wiener noise, can be quantitatively different from the deterministic control [4]. It is often solved with the Pontryagin Minimum Principle or with the Hamilton-Jacobi-Bellman equation. Both are differential equations, and difficult to solve due to boundary conditions and/or the curse of dimensionality [10].

The duality between stochastic optimal control and inference methods is long known for the linear quadratic Gaussian (LQG) case, using the Kalman Filter [10]. However generalization to non-LQG problems is still an open area of research. Traditionally, the optimal control is computed in the noise-free system, and a local LQG model is used as approximation of the system stochastics. E.g. in [16] a probabilistic inference method is introduced, which is exact for the LQG case, and generalizes to non-LQG cases by using approximate inference techniques.

[1] XRCE & INRIA Rhône-Alpes, Grenoble, France, thomas.mensink@inria.fr
[2] LEAR - INRIA Rhône-Alpes, Grenoble, France, jakob.verbeek@inria.fr
[3] Radboud University, Nijmegen, The Netherlands, b.kappen@science.ru.nl

Another direction of research is to find problem classes where the duality exists. A specific class of continuous non-linear control problems was identified that can be written as an integral over a forward diffusion process, which can be interpreted as a free energy [4, 14]. This allows the use of different approximation techniques, for example Monte-Carlo (MC) sampling or analytical approximations such as variational methods [3] and Expectation Propagation (EP) [8]. The path integral approach is promising for multi-agent control and coordination problems [1, 17, 18] and for several robot tasks [11, 12].

In this paper we study the influence of hard obstacles where the state cost is infinity. Infinite costs are problematic for Monte Carlo sampling techniques because it will kill (assign zero probability to) trajectories that hit obstacles. The situation is very similar to the re-sampling/re-weighting issues encountered in particle filters. This makes MC sampling inefficient and unreliable for these systems.

Variational methods minimize the KL-divergence such that the approximate posterior is peaked in regions of high probability and is forced to zero in regions where the true density is zero. This is problematic when the approximating distribution has infinite support (such as a Gaussian) because the variance of the approximating distribution must shrink to zero in the presence of hard obstacles. As a result variational methods lead to the (approximate) maximum a-posteriori (MAP) state, instead of the (approximate) expected marginal state required for control. The MAP solution is the cheapest possible path, ignoring the noise, and may cause the system deviate from the ideal path and hit an obstacle, generating infinite cost. While for some of the simpler problems we might choose very specific variational distributions (which have only support outside the obstacles), this is difficult to generalize to more complex problems.

In this paper we propose EP to approximate the optimal control, as it does not suffer from this problem. It optimizes the KL-divergence by means of moment matching between the target distribution and the approximation. This is well behaved, as it does not require the approximation to have zero support in regions with infinite cost. We demonstrate that in complex environments the EP approach is more efficient and more effective than the MC sampling method.

Recently Toussaint also used EP to solve stochastic control problems, in his case EP is used to perform an approximate E-step in an EM procedure to approximate optimal control [15, 16]. The path integral approach we use in this paper is valid for a more special class of control problems, in which the optimal control can be computed directly in terms of the path integral. We use EP to evaluate this path integral once, and there is no need for EM iterations as in the (more general) framework of Toussaint.

In the next section we give brief introduction to path integrals for optimal control, and in Section 3 we describe how we use EP to approximate the path integral. We present experimental results on two different problems in Section 4, where we compare EP to MC sampling. In Section 5 we conclude our paper.

2 Path Integral Optimal Control

In this section we highlight the important parts of the path integral theory, for more details see [4, 5]. For the class of non-linear control problems identified below, the Hamilton-Jacobi-Belman equation can be transformed into a linear equation. As a result of the linearity, the backward computation in time can be replaced by a forward diffusion process. The forward diffusion process can then be computed by a path integral, which can be interpreted as a free energy. The stochastic optimal control problem becomes essentially a probabilistic inference problem, which we can solve using approximate inference techniques.

We consider the class of control problems, where the system state $x \in \mathbb{R}^D$ can follow arbitrarily complex dynamics f and can be subject to arbitrarily complex cost C, however the control u is limited to the simple linear-quadratic form

$$dx = (f(x,t) + B\ u)dt + d\xi, \tag{1}$$

$$C(x_i, t_i, u(t_i \to t_f)) = \left\langle \phi(x_f) + \int_{t_i}^{t_f} dt\ \left(\frac{1}{2} u(t)^T R\ u(t) + V(x_t,t)\right)\right\rangle_{x_i}. \tag{2}$$

The initial state x_i at time t_i is fixed, and the final state x_f at time t_f is free. The cost of ending in state x_f is given by $\phi(x_f)$, and the path cost is divided in a part which is quadratic in u, and a control-independent function $V(x_t,t)$. The subscript x_i on the expectation value indicates that the expectation is over all stochastic trajectories starting at x_i. The additive noise $d\xi$ is a Wiener process with $\langle d\xi_i, d\xi_j\rangle = \nu_{ij} dt$.

The goal is to find the control trajectory $u(t_i \to t_f)$ such that $C(x_i, t_i, u(t_i \to t_f))$ is minimal. For this purpose we define the *optimal cost-to-go function* $J(x_t,t)$ from any intermediate time t and state x_t as the minimum achievable cost from thereon: $J(x_t,t) = \min_{u(t\to t_f)} C(x_t, t, u(t \to t_f))$.

This class of problems can be solved using dynamic programming following the stochastic Hamilton-Jacobi-Bellman equation, or by the forward integration of a diffusion process. Both methods use partial differential equations, and suffer from the curse of dimensionality. However the advantage of the forward diffusion process is that it can be approximated by well-known approximate inference techniques such as MC sampling, variational methods, or EP.

To reverse the direction of computation, from backwards in time to forwards in time, we define $\psi(x_t,t)$ through $J(x_t,t) = -\lambda \log \psi(x_t,t)$, with λ a constant scalar given by: $\nu = \lambda B R^{-1} B^T$. In the one dimensional case such a λ always exists. In general it restricts the choices for the matrices ν and R: in directions with low noise, control should be expensive (see [5] for details).

The forward path integral of the diffusion process, for $t < t_f$, is

$$\psi(x_t,t) = \int dy \rho(y, t_f | x_t, t) \psi(y, t_f), \tag{3}$$

with $\psi(y,t_f) = \exp(-\phi(y)/\lambda)$. The diffusion process $\rho(y,t_f|x_t,t)$ is defined by the Fokker-Planck equation

$$\partial_\tau \rho = -\frac{V(x_t,t)}{\lambda}\rho - \partial_y(f(x_t,t)\rho) + \frac{1}{2}\sum_{ij}\nu_{ij}\frac{\partial^2}{\partial_{y_i}\partial_{y_j}}\rho, \tag{4}$$

with drift $f(x_t,t)\,dt$ and diffusion $d\xi$, and an additional term due to the potential $V(x_t,t)$, the initial condition is $\rho(y,t_i|x_i,t_i) = \delta(y - x_i)$. The stochastic simulation of this diffusion process is given by

$$dx = f(x_t,t)\ dt + d\xi, \tag{5}$$

$$x_{t+dt} = \begin{cases} x_t + dx & \text{with probability } 1 - \frac{V(x_t,t)dt}{\lambda}, \\ \dagger & \text{otherwise,} \end{cases} \tag{6}$$

where the $\dagger$ operation terminates, or "kills", the procedure without producing a sample. The optimal control has the closed form solution

$$u = \frac{R^{-1}B^T\lambda}{\psi(x_t,t)}\frac{\partial}{\partial x_i}\int_{t_i}^{t_f} dy \rho(y,t_f|x_i,t_i)\psi(y,t_f), \tag{7}$$

$$= R^{-1}B^T\lambda\ (\partial_{x_i} f(x_i,t_i)dt + I)^T(\nu dt)^{-1} (\langle x_{i+dt}\rangle - f(x_i,t_i)dt - x_i), \tag{8}$$

where $\langle x_{i+dt}\rangle$ is the expected value of the state x_{i+dt}, according to the density of the normalised diffusion process. Note that the intractability of the computation of u is in the estimation of $\langle x_{i+dt}\rangle$ which is the expected state at the next time under the distribution ρ with the end term $\psi(x_f,t_f)$. This is a 'smoothed' estimate that includes all future paths and end costs.

The control signal u in (7) can also be estimated using trajectories sampled using the stochastic simulation of the diffusion process (4)

$$u \approx \frac{1}{dt\ \sum_{j\in alive}\psi(x_f^j)}\sum_{j\in alive}\psi(x_f^j)\ d\xi^{(j)}(t_i), \tag{9}$$

where $d\xi^{(j)}(t_i)$ is the noise realization at the initial time t_i for trajectory j. This is intuitive: we weight the initial noise directions, $d\xi^{(j)}(t_i)$, by their success at the final time, $\psi(x_f^j)$.

In the setting with hard obstacles we expect the stochastic simulation to frequently hit an obstacle, and terminate without producing a sample. Thus even if we run the simulation many times, the approximate control signal computed using (9) will be based on a few samples only, and might be of poor quality. Fixing the number of desired samples, and restarting the simulation each time a sample is killed is inefficient and results in high computational cost. Instead, we sample N trajectories in parallel and use a re-sampling approach to ensure that we obtain N sampled trajectories. Each trajectory that is killed at time t is replaced by a randomly selected sample that did survive up to time t, both trajectories are thus identical up to time t. Although this results in correlated trajectories, the estimate of the control signal u is more reliable as it averages over more trajectories for a comparable computational cost. Throughout this paper we always use this re-sampling strategy for MC samplers to obtain a given number of sampled trajectories.

Our principal interest in this paper is the influence of hard obstacles on control. We model these directly in the cost function $V(x_t,t)$, by assigning an infinite cost to all states outside the allowed domain, and zero inside. We define $\mathcal{V}_t(x_t) = \exp(-\frac{V(x_t,t)}{\lambda})$, which equals 1 for states within the allowed domain, and 0 outside.

3 EP for Optimal Control

The forward diffusion process can be seen as a first order Markov chain. We use $t = 0, \ldots, T$ as a discrete time index from t_i to t_f. The non-normalized likelihood of a particular realization equals

$$P(\mathbf{x}_{1:T}|x_0) = \prod_{t=1}^{T}\mathcal{V}_t(x_t)\mathcal{F}(x_{t\text{-}1}, x_t)\psi(x_T), \tag{10}$$

$$\mathcal{F}(x_{t\text{-}1}, x_t) = \mathcal{N}(x_t; x_{t\text{-}1} + f(x_{t\text{-}1}, t_{t\text{-}1})dt, \nu dt). \tag{11}$$

The path integral equals the marginal on x_T, $\rho(x_T, t_f|x_0, t_0) = \int d\mathbf{x}_{1:T\text{-}1}\, P(\mathbf{x}_{1:T}|x_0)$, in the limit of $T \to \infty$.

The optimal control is either defined using the derivative of the path integral with respect to x_i (7), or using the expected value of x_{i+dt} (8). As $P(\mathbf{x}_{1:T}|x_0)$ is intractable, we will approximate it within a family of distributions that allows for a tractable evaluation of the expectation of x_1. In particular, we define $Q(\mathbf{x}_{1:T}|x_0)$ as a product of Gaussian factors on individual variables,

$$Q(\mathbf{x}_{1:T}|x_0) = \prod_{t=1}^{T} \alpha(x_t)\beta(x_t) \prod_{d=1}^{D} \tilde{\mathcal{V}}_t^d(x_t). \tag{12}$$

The interactions through the Gaussian potentials of the original chain are approximated by the $\alpha(x_t)$ and $\beta(x_t)$ factors, which play the same role as the messages in the sum-product algorithm [7]. The hard obstacles can be written as interval constraints on each dimension d at time t, and are approximated by $\tilde{\mathcal{V}}_t^d(x_t)$. We clarify the advantages of this factorisation below.

A key difference with the MC sampling approach is that here we take explicitly the end-cost potential $\psi(x_f)$ into account. While this is straightforward in this approximation, the inclusion of the end-cost potential in the forward MC sampler is not. A possible strategy could be to include a backward smoothing sampling pass, see e.g. [6], but this is beyond the scope of this paper.

We use the EP framework to iteratively refine the parameters of each factor given the other approximation factors. Each update is based on the minimisation of $\mathrm{KL}(P\|Q)$, in contrast to variational methods where $\mathrm{KL}(Q\|P)$ is minimized. While convergence and stability are not guaranteed for EP, it often outperforms variational methods [8]. We briefly introduce the EP algorithm on the basis of its updates for the different factors, and refer the interested reader to [8, 9] for more details.

The current approximation for the site x_t, $q(x_t)$, is given by

$$q(x_t) = \alpha(x_t)\beta(x_t) \prod_{d=1}^{D} \tilde{\mathcal{V}}_t^d(x_t). \tag{13}$$

To update the factor $\alpha(x_t)$, we first remove it from the current approximation $q(x_t)$, to give

$$q^{\setminus \alpha_t}(x_t) = \beta(x_t) \prod_{d=1}^{D} \tilde{\mathcal{V}}_t^d(x_t). \tag{14}$$

We then combine $q^{\setminus \alpha_t}$ and the real factor $\mathcal{F}(x_{t\text{-}1}, x_t)$, and minimize the KL-divergence to get

$$q'_{\alpha_t}(x_t) = \mathrm{Proj}\left[q^{\setminus \alpha_t}(x_t) \int \mathcal{F}(x_{t\text{-}1}, x_t) \; \alpha(x_{t\text{-}1}) \prod_{d=1}^{D} \tilde{\mathcal{V}}_{t\text{-}1}^d(x_{t\text{-}1})\, dx_{t\text{-}1} \right]. \tag{15}$$

We use $\mathrm{Proj}\,[p]$ to denote the minimization with respect to the KL-divergence, which results in moment matching for approximating distributions in the exponential family. In our case, using the Gaussian family, the moments are the mean and covariance. The update for $\alpha(x_t)$ is given by

$$\alpha^{new}(x_t) = \frac{q'_{\alpha_t}(x_t)}{q^{\setminus \alpha_t}(x_t)}. \tag{16}$$

The updates of the β factors are done analogously, in this case the projection step (15) becomes

$$q'_{\beta_t}(x_t) = \mathrm{Proj}\left[q^{\setminus \beta_t}(x_t) \int \mathcal{F}(x_t, x_{t+1}) \; \beta(x_{t+1}) \prod_{d=1}^{D} \tilde{\mathcal{V}}_{t+1}^d(x_{t+1})\, dx_{t+1} \right]. \tag{17}$$

Note that the integrals in (15) and (17) only contain the Gaussian approximations, $\tilde{\mathcal{V}}_t^d(x_t)$, of the interval constraints. Therefore, updating the $\alpha(x_t)$ and $\beta(x_t)$ potentials reduce to inference in a Gaussian Markov chain. In the case of linear system dynamics, they can be solved using the Kalman filter and Rauch-Tung-Streibel smoother equations [10]. For non-linear system dynamics we can appeal to approximate filtering and smoothing techniques, see e.g. [13, 19].

The projection step for $\beta(x_T)$ deserves special attention as it involves the end cost potential $\psi(x_T)$,

$$q'_{\beta_T}(x_T) = \mathrm{Proj}\left[q^{\setminus \beta_T}(x_T)\; \psi(x_T) \right]. \tag{18}$$

Depending on the form of $\psi(x_T)$, we can either analytically compute the moments, or approximate them numerically e.g. by evaluation on a dense regular multidimensional grid over x_T.

The updates for the potentials of the interval constraints $\tilde{\mathcal{V}}_t^d$ are similarly. The projection step becomes

$$q'_{\tilde{\mathcal{V}}_t^d}(x_t) = \mathrm{Proj}\left[q^{\setminus \tilde{\mathcal{V}}_t^d}(x_t)\; \mathcal{V}_t^d(x_t) \right], \tag{19}$$

where $q^{\setminus \tilde{\mathcal{V}}_t^d}(x_t)$ is a Gaussian function and $\mathcal{V}_t^d(x_t)$ is an interval function. In the one dimensional setting we thus have to compute the moments of a truncated Gaussian. Which is easily done using the Gaussian error function (Erf) [2]. For the higher dimensional case, we proceed by updating potentials that approximate the interval constraints one dimension at a time. Without loss of generality, we rewrite the multidimensional Gaussian as a one-dimensional prior on the dimension of interest, and a conditional on the remaining ones. Let $x_t = \{z_1, \ldots, z_D\}$, and assume we update for dimension z_1, we write $q^{\setminus \tilde{\mathcal{V}}_t^d}(x_t) = \mathcal{N}(z_1)\mathcal{N}(z_2, \ldots, z_D|z_1)$. The projection (19) for z_1 becomes

$$\mathrm{Proj}\left[\mathcal{N}(z_1)\mathcal{N}(z_2, \ldots, z_D|z_1)\; \mathcal{V}_t^1(x_t)\right] = \mathrm{Proj}\left[\mathcal{N}(z_1)\; \mathcal{V}_t^1(x_t)\right] \mathcal{N}(z_2, \ldots, z_D|z_1), \tag{20}$$

which is performed as in the one dimensional case.

Using the EP framework we are free to choose a schedule for the factor refinements. In practice we find it advantageous to first update all $\alpha(x_t)$ and $\beta(x_t)$ factors before updating the interval constraints. In this way unnecessary conversions between natural and moment parameters are avoided, and the immediate inclusion of the filter step focuses the approximation on areas of the state space with low long-term costs.

In our experiments we fix a time step dt to define a Markov chain over T states $x_1, \ldots x_T$ from the initial time t_i to the final time t_f. We then execute the control algorithm outlined below in Algorithm 1, note that the inference is over Markov chains that get progressively shorter as we proceed.

4 Experiments

To compare controllers based on EP and MC sampling we consider simulations of two systems. The first is the 'particle in a box' problem, where a particle is subject to diffusion and must be controlled

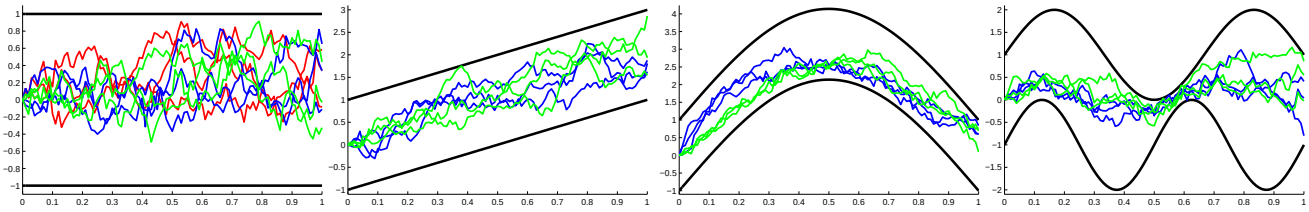

Figure 1. Simulations of the particle in a box in the environments referred to as "Wall 1" up to "Wall 4" from left to right (horizontal axis gives time, the vertical axis the state). We show 3 trajectories using the EP controller *(blue)*, the MCMC controller *(green)*, and the DTA-OCT controller *(red)* where available.

for $t \leftarrow 1$ **to** T **do**
  Observe current state x_t;
  Approximate marginals $q(x_{t'}),\ t' = t+1, \ldots, T$ using EP;
  Evaluate control signal using $q(x_{t+1})$ in equation (8);
  Execute control;
end

Algorithm 1: Optimal Control Approximated with EP

so that it does not exit a predefined interval. There is no end-cost, so the optimal control is influenced only by the hard obstacles and the system noise. In this setting we study the influence of dt, the noise ν, and the starting position x_0. Except for the case where the allowed interval is time-invariant, this system has no exact solution.

The second is a ball throwing problem, where we have to control an arm to throw a ball near the centre of a board, with restrictions on the angle and velocity of the arm. This problem differs from the first as the position of the arm is not controlled directly, but only via the velocity. More importantly, in this case there is an end-cost that prefers the ball to hit the board near its centre, which plays an important role in the optimal control. Therefore, it is not enough to merely keep the system in the domain allowed by the constraints.

4.1 Particle in a box

In the particle in a box system the state $x \in \mathbb{R}$ is subject to the following dynamics

$$dx = u\ dt + d\xi, \tag{21}$$

$$x_{t+1} = \begin{cases} x_t + dx & \text{if } \mathcal{V}_{t+1}(x_{t+1}) = 1, \\ \dagger & \text{otherwise.} \end{cases} \tag{22}$$

The control problem may be interpreted as a car steering task. In general the interval constraint $\mathcal{V}_t(x_t)$ can vary with time. In the case where (i) the interval constraint is time invariant, which can be set to $[-1, 1]$ without loss of generality, and (ii) there is no additional cost function for the final state, the solution to (4) for continuous time is

$$\rho(y, t_f | x, t_0) = \mathcal{N}(y; x, \sigma) + \sum_{n=1}^{\infty} (-1)^n \Big(\mathcal{N}(y; \mathbf{n}\ x, \nu) + \mathcal{N}(y; -\mathbf{n}\ x, \nu) \Big), \tag{23}$$

where $\mathbf{n} = 2n + (-1)^n$. This can be understood by the fact that each of the Gaussian terms satisfy (4) and the infinite sum is constructed such that ρ vanishes at the limits $x = -1$ and $x = 1$.

In our experiments we simulate the system over discrete time steps, and therefore we use a discrete time approximation of the optimal continous time controller (DTA-OCT). In this discrete time simulation the optimal control for continuous time might be suboptimal.

In Fig. 1 we show simulations for different environments, controlled using EP or MC sampling. The sampler uses $N = 50,000$ samples so that it is approximately as fast as the EP controller. In each environment we simulate 25 trajectories using each controller. We simulate the system with $dt = .01$, thus after each control step EP and the MC sampler are run for one time step less, and $\nu = 1$. Note that in any discrete simulation there remains a probability to hit the walls, due to the infinite support of the Gaussian noise.

The first environment is defined by time invariant walls, and can be solved exactly using (23) for continuous time. The second is defined by walls shifting linearly with time. Two more complex environments are also included where the walls shift non-linearly with time; in the last case the size of the interval also changes over time.

In Table 1 we give an overview of the simulations of the different environments. We show for each environment, how often each controller succeeds in reaching the end position (Suc.), and the average amount of control used by the successful simulations (Cost).

For the linear environments (wall 1 and wall 2), the MC sampler performs comparable to the EP approximation in the number of successful runs, and it uses less control. For the more complex environments, where the amount of control is relatively large compared to the displacement due to the noise, we observe that the EP approximation clearly outperforms the MC sampler based on the number of successful runs.

Looking at the trajectories, we see that the EP controller is more conservative and keeps the particle closer to the centre of the interval constraints. Of course, this is only possible at the expense of a stronger control signal, and thus higher costs. In Fig. 2 we show the incurred costs for the 25 simulations in the fixed-walls environment (wall 1), for varying starting position, noise levels, and time discretisation. Indeed we observe higher costs using the EP controller, but also that it is the only controller that avoids the walls (and thus infinite cost) in almost all simulations.

Table 1. Quantitative results of 'Particle in a box'.

	EP		MC	
	Suc.	Cost	Suc.	Cost
Wall 1	24	1.04	23	0.70
Wall 2	24	1.83	23	1.47
Wall 3	22	5.02	13	4.35
Wall 4	15	2.50	7	1.98

4.2 Ball Throwing

Here a robot arm throws a ball at end time t_f, to hit a board near to its centre. The state of the arm at any time is given by $z_t = [\theta(t), \dot{\theta}(t)]$, with θ the angle and velocity $\dot{\theta}$, we control only the velocity of the

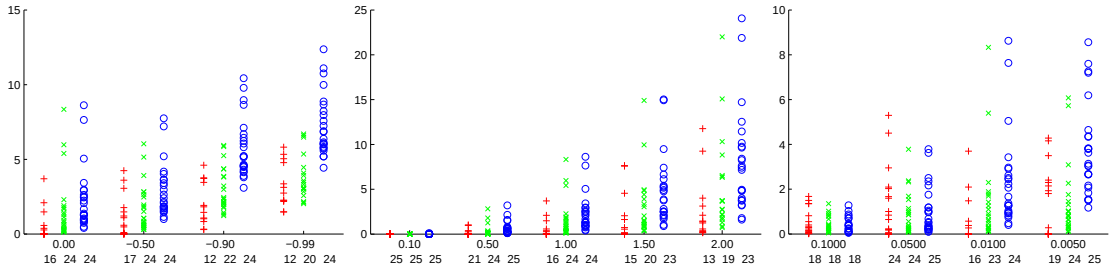

Figure 2. Control costs for the particle in a box for 25 simulations in the fixed wall environment using the DTA-OCT controller *(red +)*, the MC sampler *(green ×)*, and EP *(blue ∘)*. We vary the initial position from 0 to -0.99 (the allowed interval is $[-1, 1]$) *(left)*, the noise ν *(middle)*, and dt *(right)*. We show the incurred cost for each simulation, and below the horizontal axis we note the number of successful simulations for each controller, where the walls were not hit (simulations that do hit the wall would incur infinite cost).

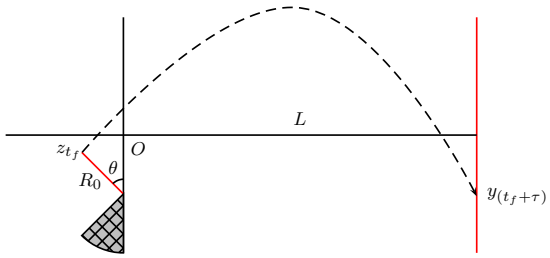

Figure 3. Schematic of the ball throwing system. On the left in red: the arm of length R_0 has position $z_{t_f} = [\theta, \dot{\theta}]$ at end time t_f. The shaded area marks the unauthorized arm configurations. The board at distance L is shown in red on the right. The trajectory of the ball is shown by the dashed line, it hits the board at $y(t_f + \tau)$.

arm. The velocity is subject to noise, whereas the relation between the velocity and the angle is deterministic.

The system is illustrated in Fig. 3. The angle of the arm is restricted to $[-\frac{3}{4}\pi, \pi]$, and the velocity is constrained to the interval $[-20, 20]$. The length of the arm is R_0, and its end is located at the origin when $\theta = 0$, the board is located at distance L from the arm. After throwing the ball at time t_f, it flies to the board while being subject to gravitational forces. The goal is to hit the board as close as possible to its centre. The position of the ball on the board is given by $y(t_f + \tau)$ where τ is the flight time. The end cost is given by $y(t_f + \tau)^2$. The system parameters we use are $\nu = 1$, $R = 1$, $dt = .1$, $t_f = 2$, $R_0 = 1$, $L = 7$, and gravitational constant $g = 10$.

Let $z_t = (\theta(t), \dot{\theta}(t))^T$ be the state of the system at time t. The state at time $t + 1$ is the given by

$$z_{t+1} = \begin{cases} z_t + dz & \text{if } \mathcal{V}_{t+1}(z_{t+1}) = 1, \\ \dagger & \text{otherwise,} \end{cases} \tag{24}$$

$$dz = dt(Az_t + Bu_t) + d\xi, \tag{25}$$

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad d\xi = \nu \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{26}$$

The chosen settings for ν and R, and the dynamics of B, results in $\lambda = 1$. To compute the end cost $\psi(z_f) = \exp(-y(t_f + \tau)^2/\lambda)$, we need the flight time τ and position $(x(t_f), y(t_f))$:

$$x(t_f) = R_0 \sin\theta(t_f), \qquad \dot{x}(t_f) = R_0\dot{\theta}(t_f)\cos\theta(t_f),$$
$$y(t_f) = R_0(\cos\theta(t_f) - 1), \qquad \dot{y}(t_f) = -R_0\dot{\theta}(t_f)\sin\theta(t_f),$$

$$x(t_f + \tau) = x(t_f) + \dot{x}(t_f)\tau = L, \tag{27}$$

$$\tau = \begin{cases} \frac{L - x(t_f)}{\dot{x}(t_f)} & \text{if } \dot{x}(t_f) > 0, \\ \infty & \text{otherwise,} \end{cases} \tag{28}$$

$$y(t_f + \tau) = y(t_f) + \dot{y}(t_f)\tau - \frac{1}{2}g\tau^2. \tag{29}$$

To compute the $\beta(x_T)$ message (18) we evaluate $\psi(z_f)$ over a dense grid over the authorized region of the state space.

We compare the controllers based on EP and MC sampling by running 50 simulations. In Fig. 4 we show for each simulation the state $z(t_f)$ at the end time using the controllers, as well as the end cost associated with all configurations. The MC controller regularly leads to solutions with very high end-cost obtained by throwing the ball at very low speed, and hitting the board far from its centre. We see that both controllers prefer overarm throws using starting state $z(t_i) = \mathbf{0}$.

In this setting the MC sampler performs poorly, because sampling according the system dynamics will yield very few samples at configurations associated with a low end-cost. This is due to the relatively small regions in the state space where $\psi(x_f)$ is non-zero, and the relatively large amount of required control compared to the system noise. In contrast, the EP controller consistently succeeds to reach configurations with low end-cost.

In Table 2 we give a quantitative account of the experimental results, comparing EP and MC with different number of samples.

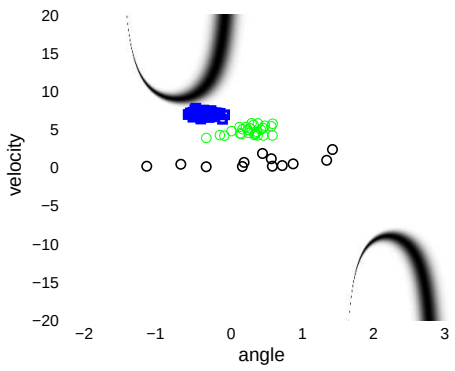

Figure 4. End-configurations $z(t_f)$ using the EP *(blue squares)* and MC controller *(green circles)*, with $N = 100,000$. The end-cost is shown in shades of grey (darker is lower cost). The MC controller regularly yields very high end-cost obtained by throwing the ball at very low speed and hitting the board far from its center, these "unacceptable" end-configurations are shown in *black circles*.

The EP controller always stays in the allowed configuration space, whereas the MC controller does so in about 60-80% of the simulations. More importantly, we see that the MC controller only leads to acceptable results in 20-60% of the cases, where we call a throw acceptable if the ball hits the board at $|y| \leq 75$, thus incurring an end cost below 5625. While the MC controller requires less control when the trajectory is acceptable, the EP controller always hits the board in an acceptable state.

These results are consistent with those observed for the particle in a box system: by using a stronger control signal the EP controller successfully avoids unauthorized system configurations.

Table 2. Quantitative results for the ball throwing task by running 50 simulations with each controler. We show the number of successful and acceptable simulations (where the board is hit at $|y| \leq 75$), the incurred cost, and the run time (relative to EP). The costs are broken down over control cost and end cost, averaged over all acceptable trajectories.

	Suc.	Acc.	Control	End	Total	Time
EP	50	50	1037	14	1050	1.0
MC 100k	33	13	540	219	759	0.9
MC 500k	39	27	614	188	803	3.4
MC 1000k	43	35	690	160	850	6.9

Finally, we note that while the ball throwing problem has two solutions, overarm and underarm throws, only one of them is preferred using the start position $z(t_i) = \mathbf{0}$, see Fig. 4. By changing the starting velocity, we can favour underarm throws. We can also change the end cost function to only allow one type of throw. We ran 25 simulations using the EP controller for different system settings, and report the results in Table 3. We observe that, depending on the starting velocity, when both throws are allowed the EP controller behaves exactly the same as when one of the throws is forced.

In rare cases when both throws are allowed the controller does not use the throw that would lead to minimum cost, see $\dot{\theta}(t_i) = 1.25$ in the table. This is possibly due to the fact that our EP approximation is uni-modal, while the end cost function is bi-modal. Better results might be obtained using a bi-modal approximation, e.g. using a mixture of uni-modal approximations using one approximation to model each throw separately.

Table 3. Path cost and end cost for different starting states $\dot{\theta}(t_i)$, allowing either both throws or only one. Throws with minimum costs are marked bold.

	Both		Underarm		Overarm	
$\dot{\theta}(t_i)$	path	end	path	end	path	end
-5.00	**977**	**15**	3034	19	**977**	**15**
-2.50	**817**	**15**	2190	19	**817**	**15**
-1.25	**876**	**15**	1880	18	**876**	**15**
0.00	**1012**	**15**	1650	18	**1012**	**15**
1.25	1247	16	1501	18	**1211**	**17**
2.50	**1439**	**18**	**1439**	**18**	1473	18
5.00	**1582**	**18**	**1582**	**18**	2187	22

5 Conclusion

We addressed the problem of stochastic optimal control in the presence of hard constraints on the system state, and introduced an EP approximation to the path integral in these settings. Through simulations of two control problems we have shown that controllers based on the EP approximation are much more successful at avoiding hard obstacles in the system configuration than controllers based on MC sampling.

The amount of control needed to reach highly rewarded end states as compared to the system noise on the controlled variables seems to be an important factor for the success of the MC sampling based controller. In the relatively easy setting of the particle in a box problem with time invariant constraints we observed that the EP controller can be overly safe and use more control than necessary. However, this is largely compensated by the superior performance in the more complex settings with time variant constraints, as well as in the ball throwing problem. In these more complex domains the sampling approach leads to a controller that often fails to obey the hard constraints. The EP controller, on the other hand, almost always successfully obeys the constraints and results in competitive overall costs.

The control in the presence of uncertainty and hard obstacles is a challenging problem and currently a bottleneck to obtain robust control strategies for real world applications. We have shown that using the linear Bellman approach in combination with the EP method yields a very reliable solution for some challenging control problems. We believe that this method can be of significant relevance for practical applications.

REFERENCES

[1] B. van den Broek, W. Wiegerinck, and H. Kappen, 'Graphical model inference in optimal control of stochastic multi-agent systems', *Journal of AI Research*, **32**, 95–122, (2008).

[2] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, volume 1, Wiley, 2nd edn., 1994.

[3] M. Jordan, Z. Ghahramani, T. Jaakola, and L. Saul, 'An introduction to variational methods for graphical models', *Machine Learning*, **37**, 183–233, (1999).

[4] H. Kappen, 'Path integrals and symmetry breaking for optimal control theory', *Journal of statistical mechanics: theory and experiment*, (2005).

[5] H. Kappen, 'An introduction to stochastic control theory, path integrals and reinforcement learning', in *Proceedings 9th Granada seminar on computational physics*, (2006).

[6] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, 'Fast particle smoothing: if I had a million particles', in *ICML*, (2006).

[7] F. Kschischang, B. Frey, and H.-A. Loeliger, 'Factor graphs and the sum-product algorithm', *IEEE Transactions on Information Theory*, **47**(2), 498–519, (Feb 2001).

[8] T. Minka, 'Expectation propagation for approximate Bayesian inference', in *Uncertainty in Artificial Intelligence*, pp. 362–369, (2001).

[9] M. Seeger, 'Expectation propagation for exponential families', Technical report, Dept. of EECS, University of California at Berkeley, (2008).

[10] R. Stengel, *Optimal Control and Estimation*, volume 1, Dover publications, 2nd edn., 1993.

[11] E. Theodorou, J. Buchli, and S. Schaal, 'Learning policy improvements with path integrals', in *AISTATS*, (2010).

[12] E. Theodorou, J. Buchli, and S. Schaal, 'reinforcement learning of motor skills in high dimensions: a path integral approach.', in *ICRA*, (2010).

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.

[14] E. Todorov, 'General duality between optimal control and estimation', in *47th IEEE Conference on Decision and Control*, (2008).

[15] M. Toussaint, 'Pros and cons of truncated Gaussian EP in the context of approximate inference control', in *NIPS-Workshop on Probabilistic Approaches for Robotics and Control*, (2009).

[16] M. Toussaint, 'Robot trajectory optimization using approximate inference', in *25nd International Conference on Machine Learning*, (2009).

[17] W. Wiegerinck, B. van den Broek, and H. Kappen, 'Stochastic optimal control in continuous space-time multi-agent systems', in *Uncertainty in Artificial Intelligence.*, pp. 528–535, (2006).

[18] W. Wiegerinck, B. van den Broek, and H. Kappen, 'Optimal on-line scheduling in stochastic multi-agent systems in continuous space and time', in *Proceedings AAMAS*, p. 8 pages, (2007).

[19] A. Ypma and T. Heskes, 'Novel approximations for inference in nonlinear dynamical systems using expectation propagation', *Neurocomputing*, **69**, 85–99, (2005).