A path integral approach to agent planning *

Hilbert J. Kappen
Department of Biophysics
Radboud University
Nijmegen, The Netherlands
b.kappen@science.ru.nl

Wim Wiegerinck
Department of Biophysics
Radboud University
Nijmegen, The Netherlands
w.wiegerinck@science.ru.nl

B. van den Broek
Department of Biophysics
Radboud University
Nijmegen, The Netherlands
b.vandenbroek@science.ru.nl

ABSTRACT

Control theory is a mathematical description of how to act optimally to gain future rewards. In this paper We discuss a class of non-linear stochastic control problems that can be efficiently solved using a path integral. In this control formalism, the central concept of cost-to-go or value function becomes a free energy and methods and concepts from statistical physics can be readily applied, such as Monte Carlo sampling or the Laplace approximation. When applied to a receding horizon problem in a stationary environment, the solution resembles the one obtained by traditional reinforcement learning with discounted reward. It is shown that this solution can be computed more efficiently than in the discounted reward framework. As shown in previous work, the approach is easily generalized to time-dependent tasks and is therefore of great relevance for modeling real-time interactions between agents.

1. INTRODUCTION

One of the central research topics of autonomous (multi-agent) systems is to design efficient methods that allow agents to plan their behaviour. Such planning can be seen as optimizing a sequence of actions to attain some future goal and is the general topic of control theory [11, 5]. In general, the stochastic non-linear control problem is intractable to solve and requires an exponential amount of memory and computation time. The reason is that the state space needs to be discretized and thus becomes exponentially large in the number of dimensions. Computing the expectation values means that all states need to be visited and requires the summation of exponentially large sums. The same intractabilities are encountered in reinforcement learning. The most efficient RL algorithms that solve the discounted reward problem (TD(λ) [12] and Q learning [13]) require millions of iterations to learn a task.

There are some stochastic control problems that can be solved efficiently. When the system dynamics is linear and the cost is quadratic (LQ control), the solution is given in terms of a number of coupled ordinary differential (Ricatti) equations that can be solved efficiently [11]. LQ control is useful to maintain a system such as for instance a chemical plant, operated around a desired point in state space and is therefore widely applied in engineering. However, it is a linear theory and too restricted to model the complexities of agent behavior. Another interesting case that can be solved efficiently is continuous control in the absence of noise [11]. One can apply the so-called Pontryagin Maximum Principle [9], which is a variational principle, that leads to a coupled system of ordinary differential equations with boundary conditions at both initial and final time. Although this deterministic control problem is not intractable in the above sense, solving the differential equation can still be rather complex in practice.

Recently, we have discovered a class of continuous non-linear stochastic control problems that can be solved more efficiently than the general case [7, 8]. These are control problems with a finite time horizon, where the control acts linearly and additive on the dynamics and the cost of the control is quadratic. Otherwise, the path cost and end cost and the intrinsic dynamics of the system are arbitrary. These control problems can have both time-dependent and time-independent solutions. The control problem essentially reduces to the computation of a path integral, which can be interpreted as a free energy. Because of its typical statistical mechanics form, one can consider various ways to approximate this path integral, such as the Laplace approximation [8], Monte Carlo sampling [8], mean field approximations or belief propagation [3].

Also, one can extend this control formalism to multiple agents that jointly solve a task. In this case the agents need to coordinate their actions not only through time, but also among each other. It was recently shown that the problem can be mapped on a graphical model inference problem and can be solved using the junction tree algorithm. Exact control solutions can be computed for instance with hundreds of agents, depending on the complexity of the cost function [15].

Non-linear stochastic control problems display features not shared by deterministic control problems nor by linear stochastic control. In deterministic control, only one globally optimal solution exists. In stochastic control, the optimal solution is a weighted mixture of suboptimal solutions. The weighting depends in a non-trivial way on the features

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

AAMAS '06 Honolulu, Hawaii USA

of the problem, such as the noise and the horizon time and on the cost of each solution. This multi-modality leads to surprising behavior is stochastic optimal control. For instance, the phenomenon of obstacle avoidance for autonomous systems not only needs to make the choice of whether to turn left or right, but also *when* such decision should be made. When the obstacle is still far away, no action is required, but there is a minimal distance to the obstacle when a decision should be made. This example was treated in [7] and it was shown that the decision is implemented by spontaneous symmetry breaking where one solution (go straight ahead) breaks in two solutions (turn left or right).

1.1 Exploration

Computing optimal behavior for an agent consists of two difficult subproblems. One is to compute the optimal behavior for a given environment, assuming that the environment is known to the agent. The second problem is to learn the environment. Here, we will mainly focus on the first problem, which is typically intractable and where the path integral approach can give efficient approximate solutions. The second problem is complicated by the fact that not all of the environment is of interest to the agent: only those parts that have high reward need to be learned. It is intuitively clear that a suboptimal control behavior that is computed by the agent, based on the limited part of the environment that he has explored, may be helpful to select the more interesting parts of the environment. But clearly, part of the agents behavior should also be purely explorative with the hope to find even more rewarding parts of the environment. This is known as the exploration-exploitation dilemma.

When the environment is known, there is no exploration issue and the optimal strategy can be computed, although this will typically require exponential time and/or memory. When the environment is not known, one should explore 'in some way' in order to learn the environment. The optimal way to explore is in general not part of the control problem.

1.2 Outline

In [15] we introduced path integral control theory [7, 8] as an attractive alternative for cooperative behaviour between agents in continuous domains. There, the task was to coordinate the behaviour such that at a fixed future time the agents are optimally allocated over a number of goals or tasks. In this paper we wish to consider path integral control with a finite receding horizon and compare it to the discounted reward problem traditionally considered in reinforcement learning. We restrict ourselves to the single agent case and at the end of the paper we will discuss its relevance for multi-agent systems.

In section 2, we introduce the special class of stochastic non-linear control problems introduced in [7, 8]. For this class of problems, the non-linear Hamilton-Jacobi-Bellman equation can be transformed into a linear equation by a log transformation of the cost-to-go. The transformation stems back to the early days of quantum mechanics and was first used by Schrödinger to relate the Hamilton-Jacobi formalism to the Schrödinger equation. The log transform was first used in the context of control theory by [4] (see also [5]).

Due to the linear description, the usual backward integration in time of the HJB equation can be replaced by computing expectation values under a forward diffusion process. The computation of the expectation value requires a stochastic integration over trajectories that can be described by a path integral. This is an integral over all trajectories starting at x, t, weighted by $\exp(-S/\nu)$, where S is the cost of the path (also know as the Action) and ν is the size of the noise.

The path integral formulation is well-known in statistical physics and quantum mechanics, and several methods exist to compute path integrals approximately. In section 2.1 we introduce the Laplace approximation, which approximates the integral by the path of minimal S. This approximation is exact in the limit of $\nu \to 0$, and the deterministic control law is recovered.

When noise is large, the Laplace approximation may not be sufficiently accurate A very generic and powerful alternative is Monte Carlo (MC) sampling, which we introduce in section 2.2. The theory naturally suggests a naive sampling procedure, but is also possible to devise more efficient samplers, such as importance sampling.

We illustrate the method on a time-invariant receding horizon control problem. The receding horizon problem is to optimize the expected cost over a fixed future time horizon. This problem is similar to the RL discounted reward cost. Finally, in section 3 we give a number of illustrative numerical examples.

2. PATH INTEGRAL CONTROL

In this section, we introduce a class of non-linear stochastic optimal control problems for cooperative behaviour between agents. For simplicity, we restrict ourselves to the one-dimensional case. [1] Let us assume that an agent at location x moves through an environment according to a stochastic dynamical equation:

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

$d\xi$ is a Wiener processes with $\left\langle d\xi^2 \right\rangle = \nu_{\alpha\beta} dt$ [2]. $f(x,t)$ is an arbitrary non-linear function, specifying the intrisic dynamics of the agent. u is the control term and (like the noise) is considered to act additively [3].

The control or planning problem is to find the control path $u(t \to t_f)$ between the current time t and some future time t_f such that the expected cost

$$C(x,t,u(t\to t_f)) = \left\langle \phi(x(t_f)) + \int_t^{t_f} d\tau \frac{R}{2} u(\tau)^2 + V(x(\tau),\tau) \right\rangle_x \tag{2}$$

is minimized, with x the state of the agent at time t. ϕ and V are arbitrary non-linear functions, but the cost of the control u is quadratic.

To solve the control problem one defines the optimal cost-to-go

$$J(t,x) = \min_{u(t\to t_f)} C(t,x,u(t\to t_f)) \tag{3}$$

[1] Generalization to higher dimensions is straightforward and x can also denote a vector of generalized coordinates (positions, velocities, orientation) of a number of agents. The general framework is discussed in [8].

[2] A Wiener process can be intuitively understood as the continuum limit of a random walk with time steps dt and space steps of $\mathcal{O}(\sqrt{\nu dt})$.

[3] Eq. 1 can be generalized to the case where $dx = f(x,t)dt + g(x,t)(udt + d\xi)$, but this generalization is not treated in this paper.

J satisfies a partial differential equation

$$\begin{aligned} -\partial_t J(x,t) &= -\frac{1}{2R}(\partial_x J(x,t))^2 + V(x,t) + f(x,t)\partial_x J(x,t) \\ &+ \frac{1}{2}\nu\partial_x^2 J(x,t) \quad (4) \\ u &= -R^{-1}\partial_x J(x,t) \quad (5) \end{aligned}$$

which is the *Stochastic Hamilton-Jacobi-Bellman Equation* with boundary condition $J(x,t_f) = \phi(x)$.

The HJB equation is a non-linear (due to the quadratic term) partial differential equation. We can, remove the non-linearity and this will turn out to greatly help us to solve the HJB equation. Define $\psi(x,t)$ through $J(x,t) = -\lambda \log \psi(x,t)$, with $\lambda = \nu R$ a constant. Then the HJB becomes

$$\begin{aligned} &-\partial_t \psi(x,t) = \\ &\left(-\frac{V(x,t)}{\lambda} + f(x,t)\partial_x + \frac{1}{2}\nu\partial_x^2\right)\psi(x,t) \quad (6) \end{aligned}$$

Eq. 6 must be solved backwards in time with $\psi(x,t_f) = \exp(-\phi(x)/\lambda)$.

The linearity allows us to reverse the direction of computation, replacing it by a diffusion process, in the following way. Let $\rho(y,\tau|x,t)$ describe a diffusion process for $\tau > t$ defined by the Fokker-Planck equation

$$\partial_\tau \rho = -\frac{V}{\lambda}\rho - \partial_y(f\rho) + \frac{1}{2}\nu\partial_y^2\rho \quad (7)$$

with $\rho(y,t|x,t) = \delta(y-x)$.

Define $A(x,t) = \int dy \rho(y,\tau|x,t)\psi(y,\tau)$. It is easy to see by using the equations of motion Eq. 6 and 7 that $A(x,t)$ is independent of τ. Evaluating $A(x,t)$ for $\tau = t$ yields $A(x,t) = \psi(x,t)$. Evaluating $A(x,t)$ for $\tau = t_f$ yields $A(x,t) = \int dy \rho(y,t_f|x,t)\psi(x,t_f)$. Thus,

$$\psi(x,t) = \int dy \rho(y,t_f|x,t)\exp(-\phi(y)/\lambda) \quad (8)$$

We arrive at the important conclusion that the optimal cost-to-go $J(x,t) = -\lambda \log \psi(x,t)$ can be computed either by backward integration using Eq. 6 or by forward integration of a diffusion process given by Eq. 7. The optimal control is given by Eq. 5.

Although Eq. 8 gives an explicit solution for the control problem, we often cannot compute the solution analytically and we must use either analytical approximations or sampling methods. For this reason, we write the diffusion kernel $\rho(y,t_f|x,t)$ in Eq. 8 as a path integral. This gives us a particular simple interpretation of how to estimate optimal control in terms of sampling trajectories. The result is [8]

$$\begin{aligned} \rho(y,t_f|x,t) &= \int [dx]_x^y \exp\left(-\frac{1}{\lambda}S_{\text{path}}(x(t\to t_f)\right) \\ S_{\text{path}}(x(t\to t_f) &= \int \frac{1}{2}(\dot{x}-f)^T R(\dot{x}-f) + V(x(\tau),\tau) \end{aligned}$$

$\int [dx]_x^y$ denote the sum over all trajectories that start at x and end at y. Substituting this in Eq. 8 we obtain

$$J(x,t) = -\lambda \log \int [dx]_x \exp\left(-\frac{1}{\lambda}S(x(t\to t_f))\right) \quad (9)$$

where the path integral $\int [dx]_x$ is over all trajectories starting at x and $S = S_{\text{path}} + \phi$. The path integral is a log partition sum. The partition sum is not over configurations, but over trajectories. $S(x(t\to t_f))$ plays the role of the energy of a trajectory and λ is the temperature.

2.1 The Laplace approximation

The simplest algorithm to approximate Eq. 9 is the Laplace approximation, which replaces the path integral by a Gaussian integral centered on the path that that minimizes the action. Let us discretize the path by n time segments: $x(t\to t_f) \approx x_{0:n}$. For each x_0 denote $x^*_{1:n} = \text{argmin}_{x_{1:n}} S(x_{0:n})$ the trajectory that minimizes S and $x^* = (x_0, x^*_{1:n})$. We expand $S(x)$ to second order around x^* : $S(x) = S(x^*) + \frac{1}{2}(x-x^*)^T H(x^*)(x-x^*)$, with $H(x^*)$ the $n\times n$ matrix of second derivatives of S, evaluated at x^*. When we substitute this approximation for $S(x)$ in Eq. 9, we are left with a n-dimensional Gaussian integral, which we can solve exactly. The resulting optimal value function is then given by

$$J_{\text{laplace}}(x_0) = S(x^*) + \frac{\lambda}{2}\log\left(\frac{\nu\epsilon}{\lambda}\right)^n \det H(x^*) \quad (10)$$

The control is computed through the gradient of J with respect to x_0. The second term, although not difficult to compute, has typically only a very weak dependence on x_0 and can therefore be ignored. In general, there may be more than one trajectory that is a local minimum of S. In this case, we use the trajectory with the lowest Action.

2.2 MC sampling

From the path integral Eq. 9 we infer that there is a simple way to compute it by sampling. The action contains a contribution from the drift and diffusion $\frac{R}{2}(\dot{x}-f)^2$, one from the potential V and one from the end cost ϕ. One can construct trajectories according to the drift and diffusion terms only and assign to each trajectory a cost according to both V and ϕ in the following way.

Define the stochastic process

$$dx = f(x,t)dt + d\xi \quad (11)$$

Then, Eq. 8 is estimated by

$$\begin{aligned} \hat{\psi}(x,t) &= \frac{1}{N}\sum_{i=1}^N \exp\left(-S_{\text{cost}}(x_i(t\to t_f))/\lambda\right) \\ S_{\text{cost}}(x(t\to t_f)) &= \phi(x(t_f)) + \int_t^{t_f} d\tau V(x(\tau),\tau) \quad (12) \end{aligned}$$

The computation of u requires the gradient of $\psi(x,t)$ which can be computed numerically by computing ψ at nearby points x and $x \pm \delta x$ for some suitable value of δx.

2.3 The receding horizon problem

Up to now, we have considered a control problem with a fixed end time. In this case, the control explicitly depends on time as $J(x,t)$ changes as a function of time. Below, we will consider reinforcement learning, which is optimal control in a stationary environment with a discounted future reward cost. We can obtain similar behavior within the path integral control approach by considering a finite receding horizon. We consider a dynamics that does not explicitly depend on time $f(x,t) = f(x)$ and a stationary environment: $V(x,t) = V(x)$ and no end cost: $\phi(x) = 0$. The optimal

cost-to-go is given by

$$J(x) = -\lambda \log \int dy \rho(y, t+T|x,t) = -\lambda \log \int [dx]_x \exp\left(-\frac{1}{\lambda} S_{\text{path}}(x(t \to t+T))\right) \quad (13)$$

with ρ the solution of the Fokker-Planck equation Eq. 7 or S_{path} the Action as given above.

Note, that because both the dynamics f and the cost V are time-independent, $\rho(y, t+T|x,t)$ and $J(x)$ do not depend on t. Therefore, if we consider a receding horizon where the end time $t_f = t+T$ moves with the actual time t, J gives the time-independent optimal cost-to-go to this receding horizon. The resulting optimal control is a time-independent function $u(x)$. The receding horizon problem is quite similar to the discounted reward problem of reinforcement learning.

3. NUMERICAL EXAMPLES

We now illustrate the difference and similarity of reinforcement learning and path integral control for a simple one dimensional example where the expected future reward within a discounted or receding horizon is optimized. The cost is given by V in figure 1 and the dynamics is simply moving to the left or the right.

For large horizon times, the optimal policy is to move from the local minimum to the global minimum of V (from right to left). The transient higher cost that is incurred by passing the barrier with high V is small compared to the long term gain of being in the global minimum instead of in the local minimum. For short horizon times the transient cost is too large and it is better to stay in the local minimum. We refer to these two qualitatively different policies as 'moving left' and 'staying put', respectively.

3.1 Reinforcement learning

In the case of reinforcement learning, the state space is discretized in 100 bins with $-2 < x < 3$. The action space is to move one bin to the left or one bin to the right: $u = \pm dx$. The dynamics is deterministic: $p_0(x'|x,u) = \delta_{x', x+u}$. The reward is given by $R(x,u,x') = -V(x')$, with $V(x)$ as given in figure 1. Reinforcement learning optimizes the expected discounted reward with discount factor γ with respect to π over all future trajectories. The discounting factor γ controls the effective horizon of the rewards through $t_{\text{hor}} = -1/\log\gamma$. Thus for $\gamma \uparrow 1$, the effective horizon time goes to infinity.

We use the policy improvement algorithm, that computes iteratively the value of a policy and then defines a new policy that is greedy with respect to this value function. The initial policy is the random policy that assigns equal probability to move left or right.

For $\gamma = 0.9$, the results are shown in fig. 1Top. J_1 is the value of the initial policy. J_∞ is the value of the policy that is obtained after convergence of policy improvement. The asymptotic policy found by the policy improvement algorithm is unique, as is checked by starting from different initial policies, and thus corresponds to the optimal policy. From the shape of J_∞ one sees that the optimal policy for the short horizon time corresponding to $\gamma = 0.9$ is to 'stay put'.

For $\gamma = 0.99$, the results are shown in fig. 1Bottom. In this case the asymptotic policy found by policy improvement is no longer unique and depends on the initial policy. J_∞ is

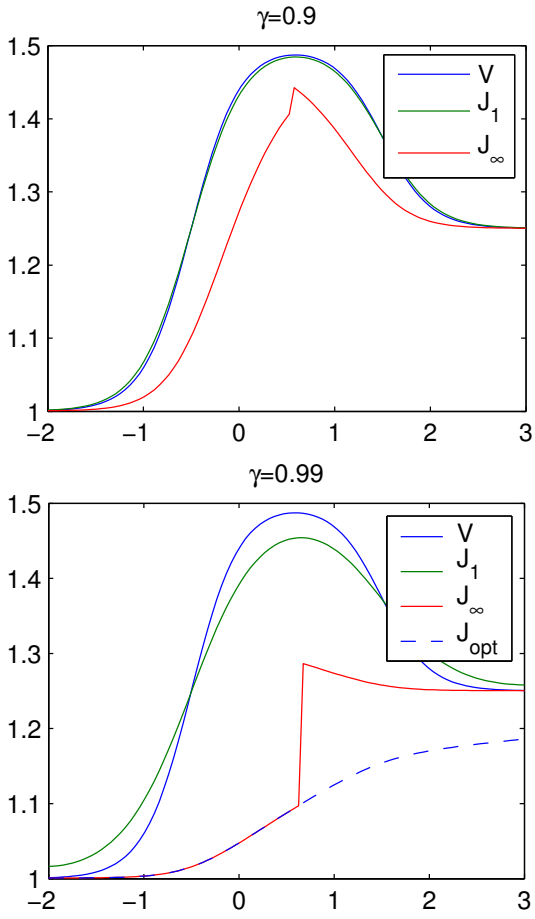

Figure 1: The policy improvement algorithm, that computes iteratively the value of a policy and then defines a new policy that is greedy with respect to this value function. In each figure, we show $V(x)$, the value $(1-\gamma)J_1(x)$ of the random initial policy, and $(1-\gamma)J_\infty(x)$ the value of the converged policy, all as a function of x.

the asymptotic policy found when starting from the random initial policy and is suboptimal. $J_{\rm opt}$ is the value of the optimal policy (always move to the left) , which is clearly better since it has a lower value for all x. Thus, for $\gamma = 0.99$ the optimal policy is to 'move left'.

This phenomenon that policy improvement may find multiple suboptimal solutions persist for all larger values of γ (larger horizon times). We also ran Q-learning on the reinforcement learning task of fig. 1 and found the optimal policy for $\gamma = 0.9, 0.99$ and 0.999 (results not shown).

3.2 Path integral control

We now compare reinforcement learning with the path integral control approach using a receding horizon time. The path integral control uses the dynamics Eq. 1 and cost Eq. 2 with $f(x,t) = 0$, $\phi(x) = 0$ and $V(x,t) = V(x)$ as given in fig. 1. The solution is given by Eq. 13.

For the Laplace approximation of J, we use Eq. 10 and the result for short horizon time $T = 3$ is given by the dashed line in fig. 2Bottom. In fig. 2Top we show the minimizing Laplace trajectories for different initial values of x. This solution corresponds to the policy to 'stay put'. For comparison, we also show $TV(x)$, which is the optimal cost-to-go if V would be independent of x.

For a relatively large horizon time $T = 10$, the Laplace solution of the cost-to-to and the minimizing trajectories are shown in figure 3.

In figs. 2 and 3 we also show the results of the MC sampling (dashed dotted line). For each x, we sample $N = 1000$ trajectories according to Eq. 11 and estimate the cost-to-go using Eq. 12.

The Laplace approximation is accurate for low noise and becomes exact in the deterministic limit. It is a 'global' solution in the sense that the minimizing trajectory is minimal with respect to the complete (known) state space. Therefore, one can assume that the Laplace results for low noise in figs. 2 and 3 are accurate. In particular in the case of a large horizon time and low noise (fig. 3), the Laplace approximation correctly proposes a policy to 'move left' whereas the MC sampler proposes (incorrectly) to 'stay put'.

The conditions for accuracy of the MC method are a bit more complex. The typical size of the area that is explored by the sampling process Eq. 11 is $x_{\rm mc} = \sqrt{\nu T}$. In order for the MC method to succeed, this area should contain some of the trajectories that make the dominant contributions to the path integral. When $T = 3, \nu = 1$, $x_{\rm mc} = 1.7$, which is sufficiently large to sample the dominant trajectories, which are the 'stay put' trajectories (those that stay in the local minima around $x = -2$ or $x = 3$). When $T = 10, \nu = 1$, $x_{\rm mc} = 3.2$, which is sufficiently large to sample the dominant trajectories, which are the 'move left' trajectories (those that move from anywhere to the global minimum around $x = -2$). Therefore, for high noise we believe the MC estimates are accurate.

For low noise and a short horizon ($T = 3, \nu = 0.01$), $x_{\rm mc} = 0.17$ which is still ok to sample the dominant 'stay put'. However, for low noise and a long horizon ($T = 10, \nu = 0.01$), $x_{\rm mc} = 0.3$ which is too small to likely sample the dominant 'move left' trajectories. Thus, the MC sampler is accurate in three of these four cases (sufficiently high noise or sufficiently small horizon). For large horizon times and low noise the MC sampler fails.

Thus, the optimal control for short horizon time $T = 3$ is

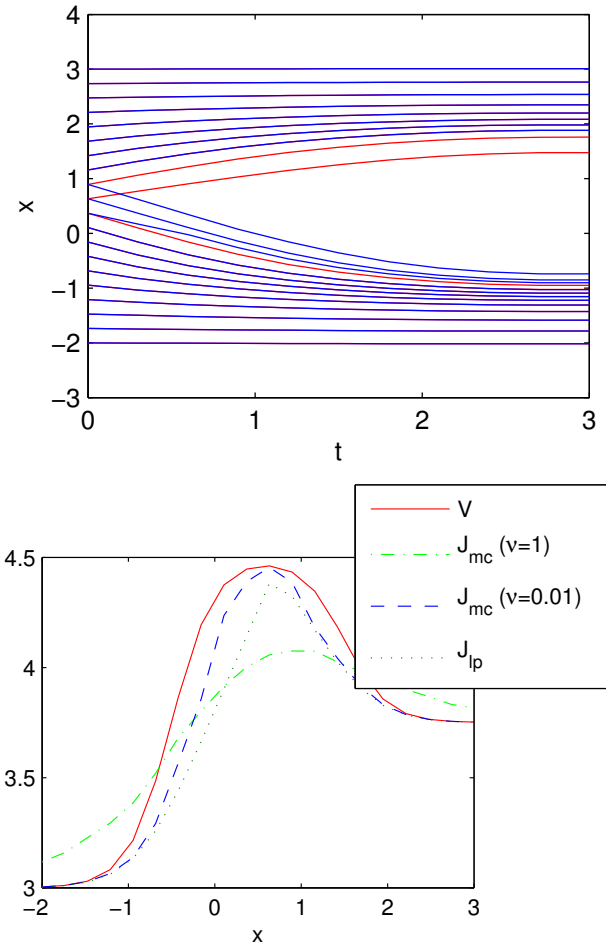

Figure 2: Top: Trajectories $x^*_{1:n}$ that minimize the Action S used in the Laplace approximation. $T = 3, R = 1$. Time discretization $dt = T/n, n = 10$. Bottom: Optimal cost-to-go $J(x)$ for different x using the Laplace approximation ($J_{\rm lp}$, dotted) and the MC sampling ($J_{\rm mc}$) for $\nu = 0.01$ (dashed) and $\nu = 1$ (dashed-dotted).

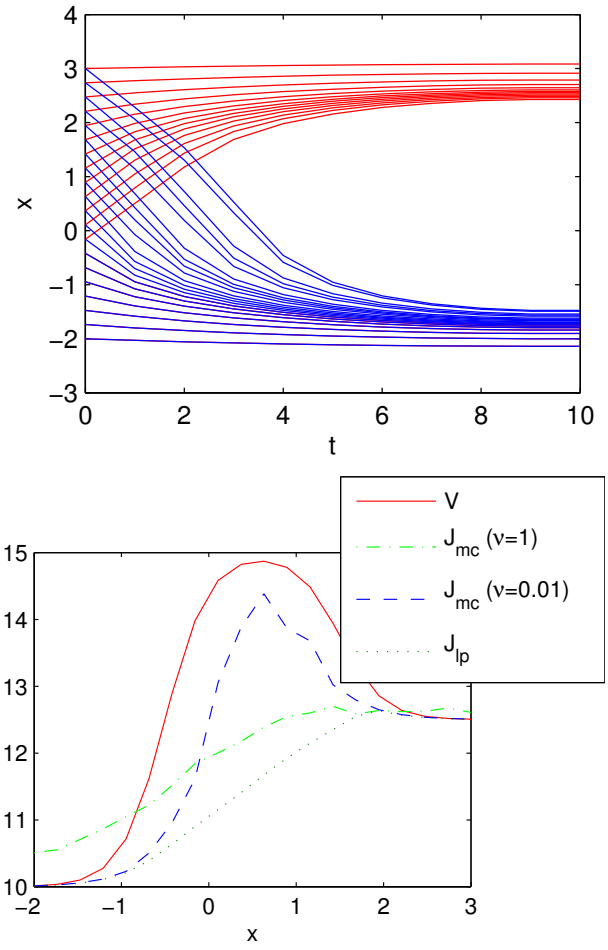

Figure 3: Top: Trajectories $x^*_{1:n}$ that minimize the Action S used in the Laplace approximation. $T = 10, R = 1$. Time discretization $dt = T/n, n = 10$. Bottom: Optimal cost-to-go $J(x)$ for different x using the Laplace approximation ($J_{\rm lp}$, dotted) and the MC sampling ($J_{\rm mc}$) for $\nu = 0.01$ (dashed) and $\nu = 1$ (dashed-dotted).

to 'stay put' more or less independent of the level of noise (fig. 2 $J_{\rm lp}$, $J_{\rm mc}(\nu = 0.01)$ and $J_{\rm mc}(\nu = 1)$). The optimal control for large horizon time $T = 10$ is to 'move left ' more or less independent of the level of noise (fig. 3 $J_{\rm lp}$, $J_{\rm mc}(\nu = 1)$).

Note, that the case of a large horizon time corresponds to the case of γ close to 1 for reinforcement learning. We see that the results of RL and path integral control qualitatively agree.

3.2.1 Exploration

When the environment is not known, one needs to learn the environment. One can proceed in one of two ways that we denote as model-based or model-free. The model-based approach is simply to first learn the environment and then compute the optimal control. The learning of the environment requires exploration of the environment. After the model is learned, there is an optimal control computation left to be done, which is typically intractable but can be computed efficiently within the path integral framework. The model-free approach is to interleave exploration (learning the environment) and exploitation (behave optimally in this environment).

The model-free approach leads to the exploration-exploitation dilemma. The intermediate controls are optimal for the limited environment that has been explored, but are of course not the true optimal controls. These controls can be used to optimally exploit the known environment, but in general give no insight how to explore. In order to compute the truly optimal control for any point x one needs to know the whole environment. At least, one needs to know the location and cost of all the low lying minima of V. If one explores on the basis of an intermediate suboptimal control strategy there is no guarantee that asymptotically one will indeed explore the full environment and thus learn the optimal control strategy.

Therefore we conclude that control theory has in principle nothing to say about how to explore. It can only compute the optimal controls for future rewards once the environment is known. The issue of optimal exploration is not addressable within the context of optimal control theory. This statement holds for any type of control theory and thus also for reinforcement learning or path integral control.

In the case of the receding horizon problem and path integral control, we propose naive sampling using the diffusion process Eq. 11 to explore states x and observe their costs $V(x)$. Note, that this exploration is not biased towards any control. We sample one very long trace at times $\tau = idt, i = 0, \ldots, N$, such that Ndt is long compared to the time horizon T. If at iteration i we are at a location x_i, we estimate $\psi(x_i, 0)$ by a single path contribution:

$$\psi(x_i, 0) = \exp\left(-\frac{dt}{\lambda}\sum_{j=i}^{j=i+n} V(x_j)\right) \qquad (14)$$

with $T = ndt$ and $x_j, j = i+1, \ldots, i+n$ the n states visited after state x_i. We can compute this expression on-line by maintaining running estimates of $\psi(x_j)$ values of recently visited locations x_j. At iteration i we initialize $\psi(x_i) = 1$ and update all recently visited $\psi(x_j)$ values with the current cost:

$$\begin{aligned}\psi(x_i) &= 1\\ \psi(x_j) &\leftarrow \psi(x_j)\exp\left(-\frac{dt}{\lambda}V(x_i)\right), \qquad j = i-n+2, \ldots, i-1\end{aligned}$$

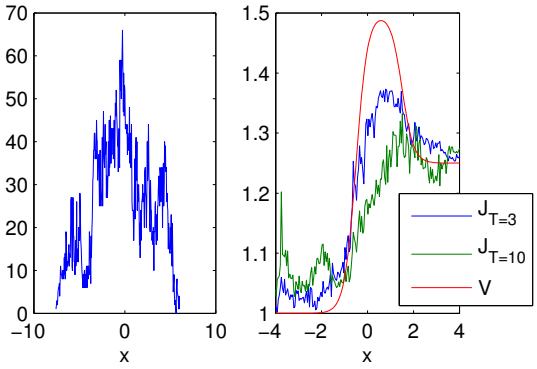

Figure 4: Sampling of $J(x)$ with one trajectory of $N = 8000$ iterations starting at $x = 0$. Left: The diffusion process Eq. 11 with $f = 0$ explores the area between $x = -7.5$ and $x = 6$. Shown is a histogram of the points visited (300 bins). In each bin x, an estimate of $\psi(x)$ is made by averaging all $\psi(x_i)$ with x_i from bin x (not shown). Right: $J_T(x)/T = -\nu \log \psi(x)/T$ versus x for $T = 3$ and $T = 10$ and $V(x)$ for comparison. Time discretization $dt = 0.02, \nu = 1, R = 1$.

The results are shown in fig. 4 for the one-dimensional problem introduced in fig 1. We use a run of $N = 8000$ iterations, starting at $x = 0$. The diffusion process explores in expectation an area of size $\sqrt{\nu N dt} = 12.3$ around the starting value. From this one run, one can estimate simultaneously $J(x)$ for different horizon times ($T = 3$ and $T = 10$ in this case). Note, that these results are similar to the MC results in fig. 3.

By exploring the space according to Eq. 11, we can learn the environment. Once learned, we can use it to compute the optimal exploitation strategy as we discussed before. As we discussed before, we have no principled way to explore. Instead of using Eq. 11 we could choose any other random or deterministic method to decide at which points in space we want to compute the immediate cost and the expected cost-to-go. Our estimated model of the environment at time t can tell us how to best exploit it between t and $t + T$, but does not provide any information about how to explore those parts of the state space that have not yet been explored.

There is however, one advantage to use Eq. 11 for exploration, and that is that it not only explores the state space and teaches us about $V(x)$ at each of these states, but at the same time provides a large number of trajectories $x_{i:i+n}, i = 1, \ldots$ that we can use to compute the expected cost to go. For different n, one obtains optimal control estimates for different horizon times.

3.3 Comparing RL and PI control

Let us here briefly summarize the main differences and similarities between the discounted reward problem using a reinforcement learning method and path integral control.

Suppose that the environment consists of X states. When the environment is not known, both RL and PI control must learn or sample the environment in some way which requires for either method $\mathcal{O}(X)$ steps. Subsequently, the discounted reward problem requires the solution of the Bellman equation, which is a system of X recursive equations involving X rewards and X unknowns, which are the values at these X states [4]. Through these equations, the value of each state depends on the value of each other state. One empirically observes that $t_{\text{cpu}} \propto 1/(1-\gamma)$ and if we define the horizon time as $T = -1/\log \gamma$ then $t_{\text{cpu}} \approx T$. The solution requires thus at least $\mathcal{O}(TX)$ steps, i.e. linear in the horizon time and exponential in the number of dimensions.

Path integral control is different in the sense that the closed form solution Eq. 8 gives the value of each state in terms of all X rewards, but this can be computed independent from the value (or cost-to-go) at other states. Thus, PI control can restrict the computation of optimal control to the points of interest (the location of the agent). The complexity to compute the PI control depends on the time discretization $n = T/dt$ and on the dimensionality of the problem d. The Laplace approximation requires the minimization of the Action, which due to its sparse structure can be done in $\mathcal{O}(nd)$ time.

The MC sampling requires a constant (possibly large) number of sampling trajectories each of length n and is therefore also proportional to n. How the required number of sample trajectories scales with d is not known in general for Monte Carlo sampling. However, the Monte Carlo sampler is well known to be a very efficient method to approximate high dimensional integrals. The standard argument is that if one is able to sample points independently from a proposal distribution, the MC method gives an unbiased estimate whose variance scales proportional to $1/\sqrt{N}$ with N the number of sample trajectories *independent of the dimension of the problem*. Thus, MC sampling requires $\mathcal{O}(Nn)$ steps with N possibly mildly depending on d. The key issue is then whether one can design a method that generate independent samples from a good proposal distribution.

Computation time for PI control increases linear with n which may grow sublinear with T because the appropriate time discretization for large horizon time need not necessary be the same as for small horizon time and therefore n may scale sub-linear with T. In fact, one could possibly choose n independent of T.

In the case of the discounted reward problem, the computation of the value of the states depends on γ and for different γ the Bellman equations need to be solved separately. In the case of PI control, the solution for larger horizon time can be obtained by simply running the diffusion process for more time. The optimal control computation for the larger horizon time makes then effective use of the previously computed solution for shorter horizon time. We saw an example of this in section 3.2.1.

With respect to exploration, RL and PI control are not very different. Both require to learn a model of the environment. In general, the control strategy that is optimal with respect to the partial environment that has been observed does not need to be a good strategy for exploration. If the objective is to learn a truly optimal control, the whole environment needs to be explored. When additional assumptions about the environment are made (for instance smoothness) this exploration can be made more ef-

[4] One may argue that online methods such as TD learning [12] and Q learning [13] are more efficient than solving the Bellman equation explicitly. This may be true for some problems, but in general convergence of these algorithms require that all states are visited 'sufficiently often' and thus their complexity is also proportional to X.

ficient by relying on interpolation and extrapolation between observed states. The issue of optimal exploration is not addressed within control theory (equally for PI control and for reinforcement learning) and requires extra criteria that determines the agents curiosity and willingness to explore.

4. DISCUSSION

In the most general, and most interesting, case, stochastic optimal control is intractable and this has been a major obstacle for applications both in artificial intelligence and in biological modeling. We have introduced a class of non-linear stochastic control problems that can be efficiently solved. In this control formalism, the central concept of cost-to-go becomes a free energy and methods and concepts from statistical physics can be readily applied. We have treated the Laplace approximation and MC sampling as examples of such efficient methods. Other methods are the mean field theory and belief propagation [3]. We believe that the PI control method is a unique approach to efficiently solve a class of continuous non-linear stochastic control problems.

We have discussed a time-independent delayed reward problem where the expected future cost in a receding horizon has to be minimized. This problem is traditionally solved using reinforcement learning and we have compared that approach to the path integral approach. Both methods give more or less the same qualitative behavior as a function of the horizon time and there seems to be a rather mild dependence on the noise in the problem. The advantage of the PI approach is that the solution can be computed more efficiently than using traditional RL methods.

An important issue not treated in this paper is planning in a receding horizon problem with a time-varying reward. This issue is very relevant for multi-agent systems, such as robot soccer or other problems where the optimal course of action of each agent depends on the expected future states of the other agents. Such problems cannot be treated in the classical RL framework because the optimal policy does not depend on time, but can be naturally incorporated in the PI control method. An example was treated in [15] where it was shown how a team of agents delays their decision to go to one of a number of target locations depending on what the other agents do. There is a non-trivial timing issue when to make these decisions and this solution manifests itself a a spontaneous symmetry breaking of the cost-to-go $J(x,t)$. In this case, the environment was stationary and the horizon was fixed (not receding). Similar (and probably more complex and interesting) behaviours are expected in such dynamical settings. In this case, the agent must learn a dynamical model of the expected behaviour of the other agents (and possibly itself).

Acknowledgments

This work is supported in part by the Dutch Technology Foundation and the BSIK/ICIS project.

5. REFERENCES

[1] R. Bellman and R. Kalaba. *Selected papers on mathematical trends in control theory*. Dover, 1964.

[2] D. E. Berlyne. *Conflict, Arousal. and Curiosity*. McGraw-Hill, New York, 1960.

[3] B. Broek, W. W., and H. Kappen. *Journal of AI Research*, 2006. In preparation.

[4] W. Fleming. Exit probabilties and optimal stochastic control. *Applied Math. Optim.*, 4:329–346, 1978.

[5] W. Fleming and H. Soner. *Controlled Markov Processes and Viscosity solutions*. Springer Verlag, 1992.

[6] U. Jönsson, C. Trygger, and P. Ögren. Lectures on optimal control. 2002.

[7] H. Kappen. A linear theory for control of non-linear stochastic systems. *Physical Review Letters*, 95:200201, 2005.

[8] H. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and Experiment*, page P11011, 2005.

[9] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The mathematical theory of optimal processes*. Interscience, 1962.

[10] S. Singh, A. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference*, Cambridge MA, 2005. MIT Press.

[11] R. Stengel. *Optimal control and estimation*. Dover publications, New York, 1993.

[12] R. Sutton and A. Barto. *Reinforcement learning: an introduction*. MIT Press, 1998.

[13] C. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge, England, 1989.

[14] R. White. Motivation reconsidered: The concept of competence. *Psychological Review*, 66:297–333, 1959.

[15] W. Wiegerinck, B. v. d. Broek, and H. Kappen. Stochastic optimal control in continuous space-time multi-agent systems. In *Proceedings UAI*. Association for Uncertainty in Artificial Intelligence, 2006. In press.

[16] J. Yong and X. Zhou. *Stochastic controls. Hamiltonian Systems and HJB Equations*. Springer, 1999.