Stochastic optimal control theory (and reinforcement learning)

Bert Kappen
Donders Center for Neuroscience, Radboud University
Nijmegen the Netherlands

March, 2025

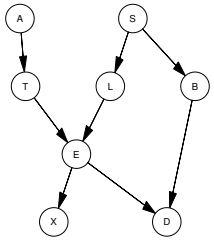
Current research

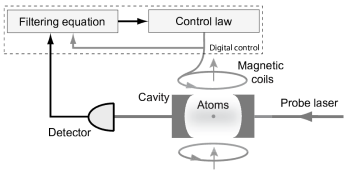
Bayesian methods
- Graphical models
- Boltzmann machines
- Approximate inference

Quantum computing
- Quantum Boltzmann machine
- Quantum annealing
- Quantum optimal control

Stochastic optimal control theory
- Path integral control theory

Missing person identification
- smart-research.nl

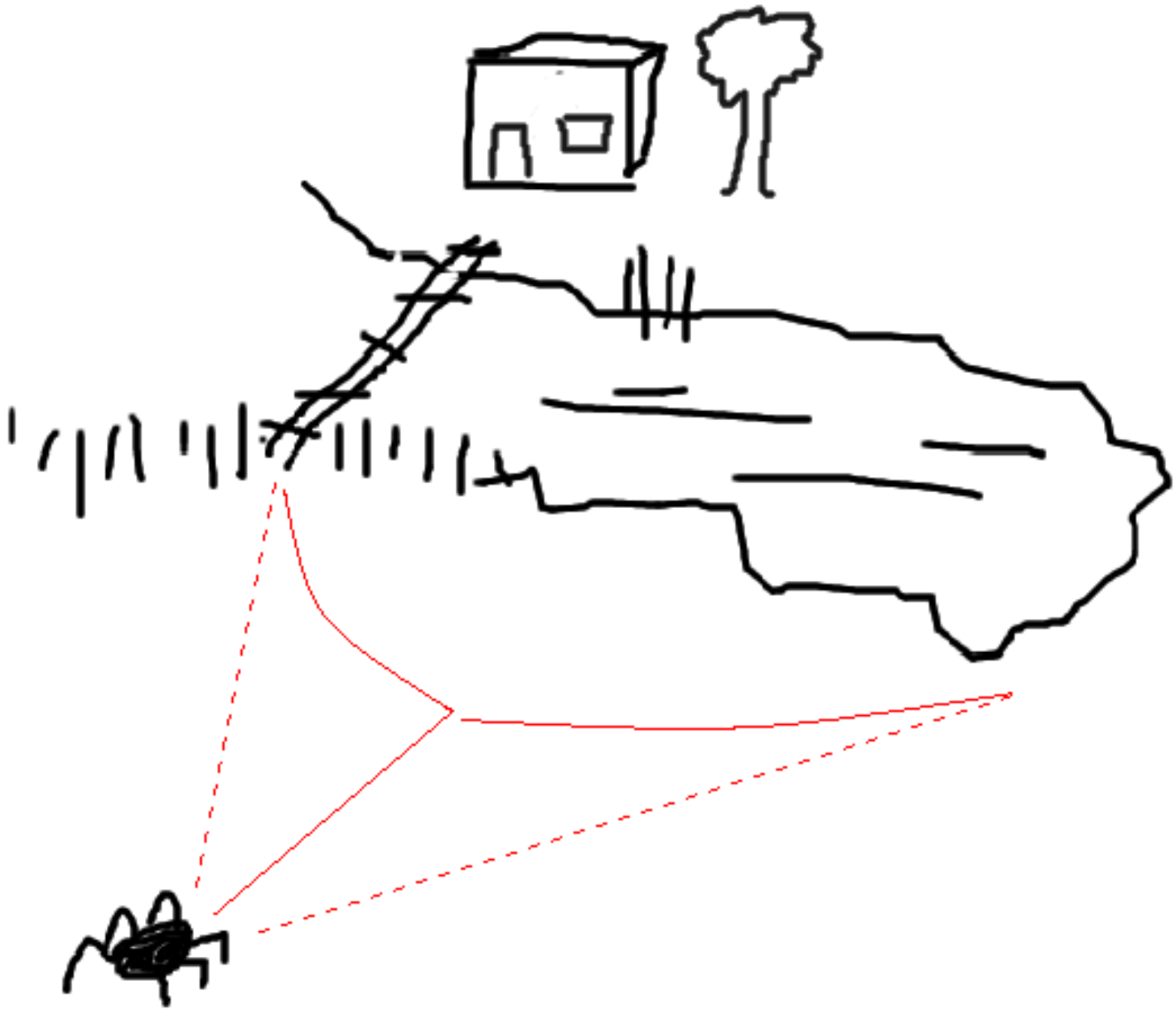




Introduction

Optimal control theory: Optimize sum of a path cost and end cost. Result is optimal control sequence and optimal trajectory.

Input: Cost function. **Output:** Optimal trajectory and controls.



Introduction

Control problems are delayed reward problems:

- Motor control: devise a sequece of motor commands to reach a goal
- finance: devise a sequence of buy/sell commands to maximize profit
- Learning, exploration exploitation

Types of optimal control problems

Finite horizon (fixed horizon time)

- Dynamics and environment may depend explicitly on time.
- Optimal control depends explicitly on time.

Types of optimal control problems

Finite horizon (moving horizon)

- Dynamics and environment are static.
- Optimal control is time independent.

Infinite horizon

- discounted reward, Reinforcement learning
- total reward, absorbing states
- average reward

Other issues:

- discrete vs. continuous state
- discrete vs. continuous time
- observable vs. partial observable

Overview

Lecture 1: Optimal control theory, discrete time

- Introduction of delayed reward problem in discrete time;
- Dynamic programming solution and deterministic Bellman equations;
- Extension to noisy case
- Examples
- Bandits: Optimal exploration by dynamic programming

Overview

Lecture 2: Optimal control theory, continuous time

- Solution in continuous time and states;
- Example: Mass on a spring
- Pontryagin maximum principle; Notion of an optimal (particle) trajectory
- Again Mass on a spring
- Stochastic differential equations
- Kolmogorov and Fokker-Plack equations
- Hamilton-Jacobi-Bellman equation (continuous state and time)
- LQ control, Ricatti equation;
- Example of LQ control
- Portfolio selection

Overview

Lecture 3: Stochastic optimal control theory

- Path integral control
- KL control theory and relation to path integral control
- Importance sampling
- Examples: Delayed choice, acrobot robot, racing car

Overview

Further topics

- Comparison PI control and RL
- Multi agent systems
- Mean field approximation for control: n joint arm, multi agents

(- Risk sensitive control)

(- Inference and control)

(- control of quantum systems)

Material

- H.J. Kappen. Optimal control theory and the linear Bellman Equation. In *Inference and Learning in Dynamical Models (Cambridge University Press 2010)*, edited by David Barber, Taylan Cemgil and Sylvia Chiappa `http://www.snn.ru.nl/~bertk/control/timeseriesbook.pdf`
- S. Thijssen, H.J. Kappen, Path integral control and state-dependent feedback, PRE 91, 2015 `http://link.aps.org/doi/10.1103/PhysRevE.91.032104`
- Dimitri Bertsekas, Dynamic programming and optimal control
- `http://www.snn.ru.nl/~bertk/control/cwi2025.html`
- `http://www.snn.ru.nl/~bertk/control_theory`

Lecture 1: Optimal control theory: discrete time

Discrete time control

Consider the control of a discrete time deterministic dynamical system:

$$x_{t+1} = x_t + f(t, x_t, u_t), \quad t = 0, 1, \ldots, T-1$$

x_t describes the *state* and u_t specifies the *control* or *action* at time t.

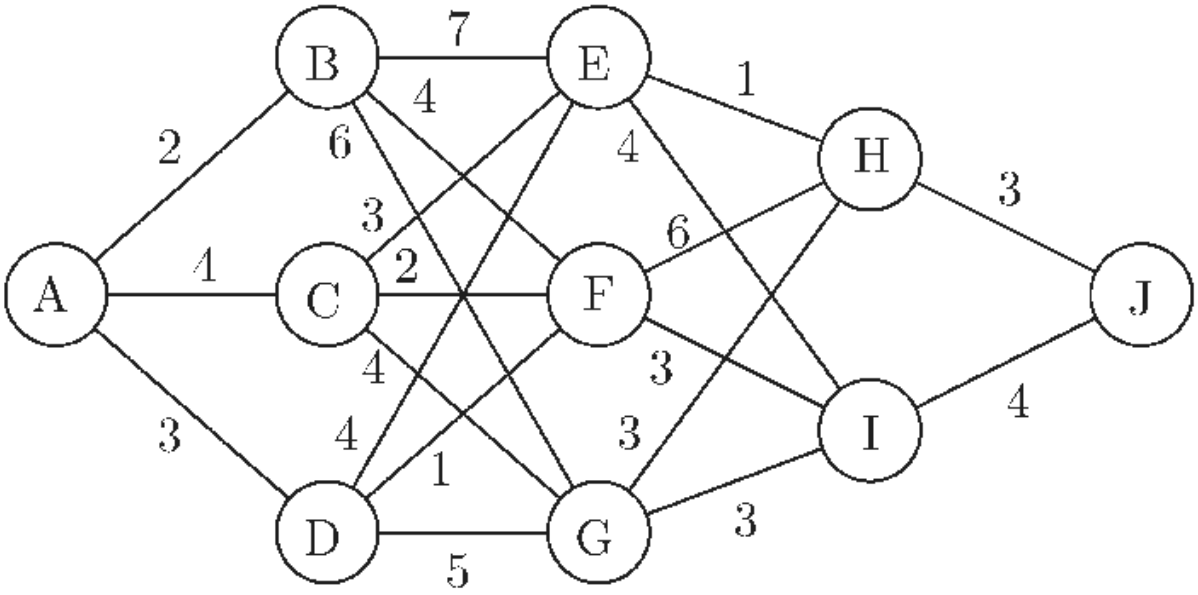
Given $x_{t=0} = x_0$ and $u_{0:T-1} = u_0, u_1, \ldots, u_T - 1$, we can compute $x_{1:T}$.

Define a cost for each sequence of controls:

$$C(x_0, u_{0:T-1}) = \phi(x_T) + \sum_{t=0}^{T-1} R(t, x_t, u_t)$$

The problem of optimal control is to find the sequence $u_{0:T-1}$ that minimizes $C(x_0, u_{0:T-1})$.

Dynamic programming



Find the minimal cost path from A to J.

$$C(J) = 0, C(H) = 3, C(I) = 4$$

$$C(F) = \min(6 + C(H), 3 + C(I))$$

Discrete time control

The optimal control problem can be solved by dynamic programming. Introduce the *optimal cost-to-go*:

$$J(t, x_t) = \min_{u_{t:T-1}} \left(\phi(x_T) + \sum_{s=t}^{T-1} R(s, x_s, u_s) \right)$$

which solves the optimal control problem from an intermediate time t until the fixed end time T, for all intermediate states x_t.

Then,

$$\begin{aligned} J(T, x) &= \phi(x) \\ J(0, x) &= \min_{u_{0:T-1}} C(x, u_{0:T-1}) \end{aligned}$$

Discrete time control

One can recursively compute $J(t, x)$ from $J(t+1, x)$ for all x in the following way:

$$\begin{aligned}
J(t, x_t) &= \min_{u_{t:T-1}} \left(\phi(x_T) + \sum_{s=t}^{T-1} R(s, x_s, u_s) \right) \\
&= \min_{u_t} \left(R(t, x_t, u_t) + \min_{u_{t+1:T-1}} \left[\phi(x_T) + \sum_{s=t+1}^{T-1} R(s, x_s, u_s) \right] \right) \\
&= \min_{u_t} \left(R(t, x_t, u_t) + J(t+1, x_{t+1}) \right) \\
&= \min_{u_t} \left(R(t, x_t, u_t) + J(t+1, x_t + f(t, x_t, u_t)) \right)
\end{aligned}$$

This is called the *Bellman Equation*.

Computes u as a function of x, t for all intermediate t and all x.

Discrete time control

The algorithm to compute the optimal control $u^*_{0:T-1}$, the optimal trajectory $x^*_{1:T}$ and the optimal cost is given by

1. Initialization: $J(T, x) = \phi(x)$

2. Backwards: For $t = T-1, \ldots, 0$ and for all x compute

$$\begin{aligned} u^*_t(x) &= \arg\min_u \{R(t, x, u) + J(t+1, x + f(t, x, u))\} \\ J(t, x) &= R(t, x, u^*_t) + J(t+1, x + f(t, x, u^*_t)) \end{aligned}$$

3. Forwards: For $t = 0, \ldots, T-1$ compute

$$x^*_{t+1} = x^*_t + f(t, x^*_t, u^*_t(x^*_t))$$

NB: the backward computation requires $u^*_t(x)$ for all x.

Stochastic case

$$x_{t+1} = x_t + f(t, x_t, u_t, w_t) \qquad t = 0, \ldots, T-1$$

At time t, w_t is a random value drawn from a probability distribution $p(w)$.

For instance,

$$\begin{aligned} x_{t+1} &= x_t + w_t, \qquad x_0 = 0 \\ w_t &= \pm 1, \qquad p(w_t = 1) = p(w_t = -1) = 1/2 \\ x_t &= \sum_{s=0}^{t-1} w_s \end{aligned}$$

Thus, x_t random variable and so is the cost

$$C(x_0) = \phi(x_T) + \sum_{t=0}^{T-1} R(t, x_t, u_t, \xi_t)$$

Stochastic case

$$
\begin{aligned}
C(x_0) &= \left\langle \phi(x_T) + \sum_{t=0}^{T-1} R(t, x_t, u_t, \xi_t) \right\rangle \\
&= \sum_{w_{0:T-1}} \sum_{\xi_{0:T-1}} p(w_{0:T-1}) p(\xi_{0:T-1}) \left(\phi(x_T) + \sum_{t=0}^{T-1} R(t, x_t, u_t, \xi_t) \right)
\end{aligned}
$$

with ξ_t, x_t, w_t random. Closed loop control: find *functions* $u_t(x_t)$ that minimizes the remaining expected cost when in state x at time t. $\pi = \{u_0(\cdot), \ldots, u_{T-1}(\cdot)\}$ is called a policy.

$$
\begin{aligned}
x_{t+1} &= x_t + f(t, x_t, u_t(x_t), w_t) \\
C_\pi(x_0) &= \left\langle \phi(x_T) + \sum_{t=0}^{T-1} R(t, x_t, u_t(x_t), \xi_t) \right\rangle
\end{aligned}
$$

$\pi^* = \text{argmin}_\pi C_\pi(x_0)$ is optimal policy.

Stochastic Bellman Equation

$$
\begin{aligned}
J(t, x_t) &= \min_{u_t} \langle R(t, x_t, u_t, \xi_t) + J(t+1, x_t + f(t, x_t, u_t, w_t)) \rangle \\
J(T, x) &= \phi(x)
\end{aligned}
$$

u_t is optimized for each x_t separately. $\pi = \{u_0, \ldots, u_{T-1}\}$ is optimal a policy.

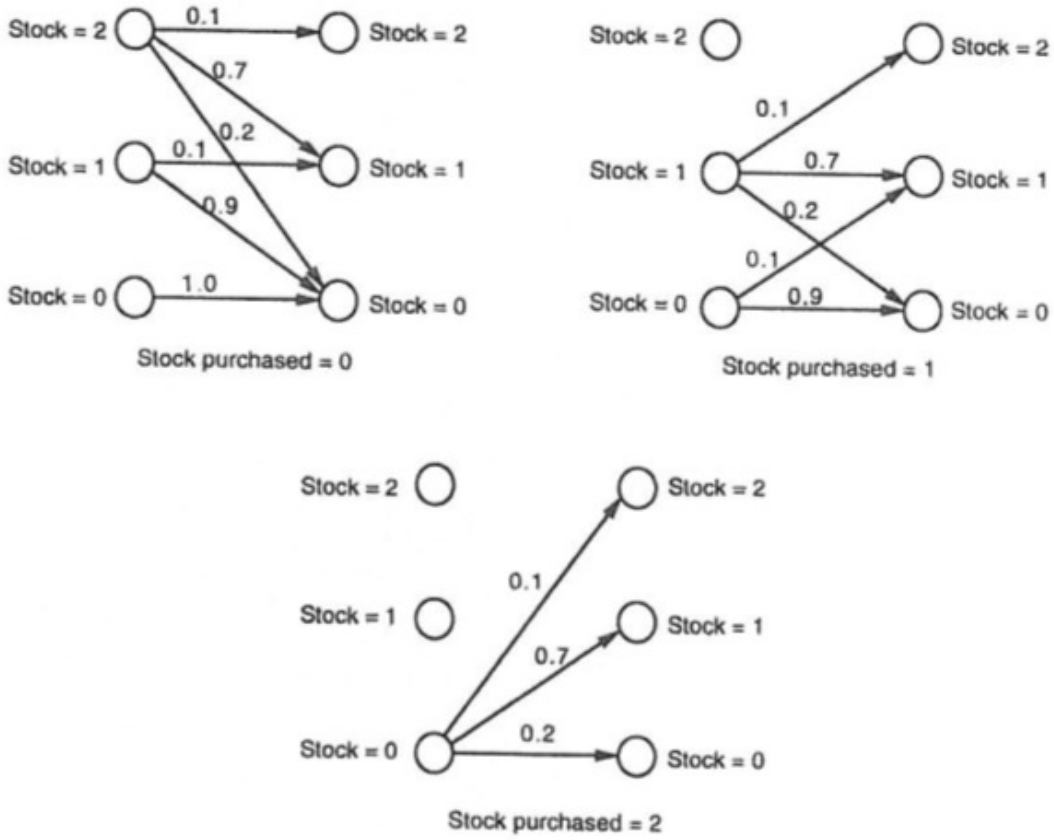
Inventory problem

- $x_t = 0, 1, 2$ stock available at the beginning of period t.
- u_t stock ordered at the beginning of period t. Maximum storage is 2: $u_t \leq 2 - x_t$.
- $w_t = 0, 1, 2$ demand during period t with $p(w = 0, 1, 2) = (0.1, 0.7, 0.2)$; excess demand is lost.
- u_t is the cost of purchasing u_t units. $(x_t + u_t - w_t)^2$ is cost of stock at end of period t.

$$
\begin{aligned}
x_{t+1} &= \max(0, x_t + u_t - w_t) \\
C(x_0, u_{0:T-1}) &= \left\langle \sum_{t=0}^{t=2} u_t + (x_t + u_t - w_t)^2 \right\rangle
\end{aligned}
$$

Planning horizon $T = 3$.

Inventory problem



Apply Bellman Equation

$$
\begin{aligned}
J_t(x_t) &= \min_{u_t} \langle R(x_t, u_t, w_t) + J_{t+1}(f(x_t, u_t, w_t)) \rangle \\
R(x, u, w) &= u + (x + u - w)^2 \\
f(x, u, w) &= \max(0, x + u - w)
\end{aligned}
$$

Start with $J_3(x_3) = 0, \forall x_3$.

Dynamic programming in action

Assume we are at stage $t = 2$ and the stock is x_2. The cost-to-go is what we order u_2 and how much we have left at the end of period $t = 2$.

$$
\begin{aligned}
J_2(x_2) &= \min_{0 \le u_2 \le 2 - x_2} u_2 + \left\langle (x_2 + u_2 - w_2)^2 \right\rangle \\
&= \min_{0 \le u_2 \le 2 - x_2} \Big(u_2 + 0.1 * (x_2 + u_2)^2 + 0.7 * (x_2 + u_2 - 1)^2 \\
&+ \quad 0.2 * (x_2 + u_2 - 2)^2 \Big) \\
J_2(0) &= \min_{0 \le u_2 \le 2} \Big(u_2 + 0.1 * u_2^2 + 0.7 * (u_2 - 1)^2 + 0.2 * (u_2 - 2)^2 \Big)
\end{aligned}
$$

$$
\begin{aligned}
u_2 = 0 \quad &: \quad rhs = 0 + 0.7 * 1 + 0.2 * 4 = 1.5 \\
u_2 = 1 \quad &: \quad rhs = 1 + 0.1 * 1 + 0.2 * 1 = 1.3 \\
u_2 = 2 \quad &: \quad rhs = 2 + 0.1 * 4 + 0.7 * 1 = 3.1
\end{aligned}
$$

Thus, $u_2(x_2 = 0) = 1$ and $J_2(x_2 = 0) = 1.3$

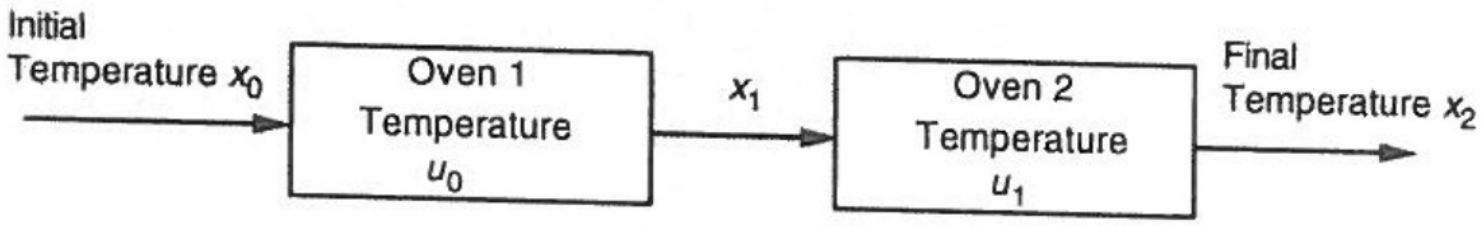
Inventory problem

The computation can be repeated for $x_2 = 1$ and $x_2 = 2$, completing stage 2 and subsequently for stage 1 and stage 0.

Stock	Stage 0 Cost-to-go	Stage 0 Optimal stock to purchase	Stage 1 Cost-to-go	Stage 1 Optimal stock to purchase	Stage 2 Cost-to-go	Stage 2 Optimal stock to purchase
0	3.67	1	2.5	1	1.3	1
1	2.67	0	1.2	0	0.3	0
2	2.608	0	1.68	0	1.1	0

Exercise: Two ovens

A certain material is passed through a sequence of two ovens. Aim is to reach pre-specified final product temperature x^* with minimal oven energy.



$x_{0,1,2}$ are the product temperatures initially, after pasing through oven 1 and after passing through oven 2. $u_{0,1}$ are the oven temperatures. The dynamics is

$$\begin{aligned} x_{t+1} &= (1-a)x_t + au_t \qquad t = 0, 1 \\ C &= r(x_2 - x^*)^2 + u_0^2 + u_1^2 \end{aligned}$$

- Find the optimal control solution u_0, u_1.
- Show that adding mean zero noise to the dynamics ($x_{t+1} = (1-a)x_t + au_t + w_t$ with $\langle w_t \rangle = 0$), does not change the optimal control solution.

Example: Two ovens

End cost-to-go is $J(2, x_2) = r(x_2 - x^*)^2$.

$$
\begin{aligned}
J(1, x_1) &= \min_{u_1} \left(u_1^2 + J(2, x_2) \right) = \min_{u_1} \left(u_1^2 + r((1-a)x_1 + au_1 - x^*)^2 \right) \\
u_1 &= \mu_1(x_1) = \frac{ra(x^* - (1-a)x_1)}{1 + ra^2} \\
J(1, x_1) &= \frac{r((1-a)x_1 - x^*)^2}{1 + ra^2} \\
J(0, x_0) &= \min_{u_0} \left(u_0^2 + J(1, x_1) \right) = \min_{u_0} \left(u_0^2 + \frac{r((1-a)x_1 - x^*)^2}{1 + ra^2} \right) \\
&= \min_{u_0} \left(u_0^2 + \frac{r((1-a)((1-a)x_0 + au_0) - x^*)^2}{1 + ra^2} \right) \\
u_0 &= \mu_0(x_0) = \frac{r(1-a)a(x^* - (1-a)^2 x_0)}{1 + ra^2(1 + (1-a)^2)} \\
J(0, x_0) &= \frac{r((1-a)^2 x_0 - x^*)^2}{1 + ra^2(1 + (1-a)^2)}
\end{aligned}
$$

Comments

- **Linear Quadratic Control:** Solution can be obtained in closed form because problem is linear quadratic.
- **Certainty equivalence:** Optimal control solution is unaffected by noise:

$$\begin{aligned} x_{t+1} &= (1-a)x_t + au_t + w_t \qquad t = 0, 1 \\ C &= r(x_2 - x^*)^2 + u_0^2 + u_1^2 \end{aligned}$$

with $\langle w_t \rangle = 0$.Then

$$\begin{aligned} J(1, x_1) &= \min_{u_1} \left(u_1^2 + \left\langle r((1-a)x_1 + au_1 + w_1 - x^*)^2 \right\rangle \right) \\ &= \min_{u_1} \left(u_1^2 + r((1-a)x_1 + au_1 - x^*)^2 + r \langle w_1 \rangle^2 \right) \end{aligned}$$

Exploitation versus Exploration: The Single-State Case

The k-armed bandit problem:

The agent is in a room with a collection of k gambling machines (each called a "one-armed bandit"). The agent is permitted a fixed number of pulls, h. Any arm may be pulled on each turn. The machines do not require a deposit to play; the only cost is in wasting a pull playing a suboptimal machine. When arm i is pulled, machine i pays off 1 or 0, with *unknown* probability p_i. What should the agent's strategy be?

Trade-off between

exploration: try many new arms

exploitation: stick with a good arm

The bandit problem is a control problem where the state space is the current belief about the bandits pay-off.

Bayesian model

Suppose that arm i is pulled n_i times giving w_i payoffs 1 and $n_i - w_i$ payoffs 0. When p_i is known, we can compute the probability

$$P(w_i|p_i, n_i) = \binom{n_i}{w_i} p_i^{w_i}(1-p_i)^{n_i-w_i}$$

But we dont know p_i.....

Consider the Beta distribution over the continuous variable $0 \leq x \leq 1$ parametrized by $\alpha, \beta > 0$ integers:

$$P(x|\alpha,\beta) = \frac{(\alpha+\beta-1)!}{(\alpha-1)!(\beta-1)!} x^{\alpha-1}(1-x)^{\beta-1} \qquad \langle x \rangle = \frac{\alpha}{\alpha+\beta}$$

The complex prefactor ensures normalization $\int_0^1 dx P(x|\alpha,\beta) = 1$.

We assume a prior distribution $P_0(p_i) = P(p_i|\alpha = \beta = 1) = 1$ to model our prior ignorance of the value of p_i (flat prior). All p_i are equally likely.

When pulling arm i n_i times giving w_i payoffs 1 and $n_i - w_i$ payoffs 0, the posterior distribution over p_i is given by Bayes rule:

$$\begin{aligned} P(p_i|n_i, w_i) &= \frac{P(w_i|p_i, n_i)P_0(p_i)}{\int dp_i P(w_i|p_i, n_i)P_0(p_i)} \propto p_i^{w_i}(1-p_i)^{n_i - w_i} \\ P(p_i|n_i, w_i) &= P(p_i|\alpha = w_i + 1, \beta = n_i - w_i + 1) \end{aligned}$$

The evidence (n_i, w_i) defines our *belief* in p_i (the distribution $P(p_i|\alpha = w_i + 1, \beta = n_i - w_i + 1)$). [1]

The belief state of the agent at time t is the tuple $\{n_1, w_1, \ldots, n_k, w_k\}$ with k the number of bandits and $\sum_{i=1}^{k} n_i = t$.

[1] Note, that the expected p_i is given as

$$\langle p_i \rangle = \frac{w_i + 1}{n_i + 2}$$

NB if you pull once: $n_i = w_i = 1$, the expected return is $2/3$.

Dynamic programming solution

Suppose we can pull in total h times one of the arms. Define t the current iteration, $0 \leq t \leq h$. At each t we wish to pull the 'best' arm based on our experience sofar.

We write $V_t^*(n_1, w_1, \ldots, n_k, w_k)$ as the expected remaining payoff at time $t = \sum_{i=1}^k n_i$, given that a total of h pulls are available, *and we use the remaining pulls optimally.*

The number of states is large. We get a rough estimate by noting that for given $n_1, \ldots, n_k$ the number of possible $w_1, \ldots, w_k$ is $\prod_{i=1}^k (n_i + 1) \propto h^k$, since n_i is of order h

Dynamic programming solution

If $t = \sum_i n_i = h$ there are no remaining pulls and $V^*_{t=h}(n_1, w_1, \ldots, n_k, w_k) = 0$.

If we know V^*_t for all states at iteration t, we can compute V^*_{t-1} for any belief state: [2]

$$
\begin{aligned}
V^*_{t-1}(n_1, w_1, \ldots, n_k, w_k) &= \max_{i \in \{1,\ldots,k\}} \langle \text{agent takes action } i \text{ at time } t-1 \text{ and optimally from } t \text{ onwards} \rangle \\
&= \max_{i \in \{1,\ldots,k\}} [\rho_i \{\text{arm i returns } 1 + V^*_t(n_1, w_1, \ldots, n_i + 1, w_i + 1, \ldots, n_k, w_k)\} \\
&+ (1 - \rho_i)\{\text{arm i returns } 0 + V^*_t(n_1, w_1, \ldots, n_i + 1, w_i, \ldots, n_k, w_k)\}] \\
&= \max_{i \in \{1,\ldots,k\}} \rho_i + \rho_i V^*_t(n_1, w_1, \ldots, n_i + 1, w_i + 1, \ldots, n_k, w_k) \\
&+ (1 - \rho_i) V^*_t(n_1, w_1, \ldots, n_i + 1, w_i, \ldots, n_k, w_k)
\end{aligned}
$$

We use $\rho_i = \langle p_i \rangle = \frac{w_i+1}{n_i+2}$ as our expected success probability for arm i based on past experience (and our prior).

[2] NB: Error in formula on pg. 243 of [1]. Immediate reward term is missing.

Example

h=4, two bandits. Notation: $V_t^*(n_1, w_1, n_2, w_2) = (n_1 w_1 n_2 w_2)$

Use Bellman equation to compute backwards all values:

- If $t = n_1 + n_2 = 4$ $V_t^*(n_1, w_1, n_2, w_2) = 0$
- Consider states with $t = n_1 + n_2 = 3$. For instance, [3]

$$
\begin{aligned}
(0030) &= \max(\rho_1(00), \rho_2(30)) = \max\left(\frac{1}{2}, \frac{1}{5}\right) = \frac{1}{2} = (3000) \\
(2211) &= \max(\rho_1(22), \rho_2(11)) = \max\left(\frac{3}{4}, \frac{2}{3}\right) = \frac{3}{4} = (1122) \\
(2111) &= \max(\rho_1(21), \rho_2(11)) = \max\left(\frac{2}{4}, \frac{2}{3}\right) = \frac{2}{3} = (1121) \\
&\ldots
\end{aligned}
$$

[3] $\rho_i(nw) = \frac{w+1}{n+2}$. Thus $\rho_1(00) = \rho(n_1 = 0, w_1 = 0) = \frac{1}{2}$. $\rho_2(30) = \rho(n_2 = 3, w_2 = 0) = \frac{1}{5}$.

- Consider states with $t = n_1 + n_2 = 2$. For instance,

$$\begin{aligned}(1111) &= \max\left[\rho_1 + \rho_1(2211) + (1-\rho_1)(2111), \rho_2 + \rho_2(1122) + (1-\rho_2)(1121)\right] \\ &= \frac{2}{3}\left(1 + \frac{3}{4}\right) + \frac{1}{3}\frac{2}{3} = 1.39\end{aligned}$$

with $\rho_1 = \rho_1(11) = 2/3$ and $\rho_2 = \rho_2(11) = 2/3$.

Matlab results:

t= 3:
(0030)=0.50 (0031)=0.50 (0032)=0.60 (0033)=0.80 (1020)=0.33 (1021)=0.50 (1022)=0.75
(1120)=0.67 (1121)=0.67 (1122)=0.75 (2010)=0.33 (2011)=0.67 (2110)=0.50 (2111)=0.67
(2210)=0.75 (2211)=0.75 (3000)=0.50 (3100)=0.50 (3200)=0.60 (3300)=0.80
t= 2:
(0020)=1.00 (0021)=1.08 (0022)=1.50 (1010)=0.72 (1011)=1.33 (1110)=1.33 (1111)=1.39
(2000)=1.00 (2100)=1.08 (2200)=1.50
t= 1:
(0010)=1.53 (0011)=2.03 (1000)=1.53 (1100)=2.03
t= 0:
(0000)=2.28

The values V_t^* are used to compute the optimal sequence of actions.

- First step: Pull arm 1. Since our prior beliefs are equal it does not matter which arm we pull.
  Suppose we win. Then our state is (1100) and $\rho_1 = 2/3, \rho_2 = 1/2$.

- Second step: Determine the optimal pull from state (1100) based on future expected reward:

$$\begin{aligned} & \text{argmax}\,(\rho_1 + \rho_1(2200) + (1-\rho_1)(2100), \rho_2 + \rho_2(1111) + (1-\rho_2)(1110)) \\ = \;& \text{argmax}\,(2/3 + 2/3 * 1.5 + 1/3 * 1.08, 1/2 + 1/2 * 1.39 + 1/2 * 1.33) \\ = \;& \text{argmax}(2.03, 1.86) \end{aligned}$$

  Thus pull arm 1.

- ...

Lecture 2: Optimal control theory, continuous time

Continuous limit

Replace $t+1$ by $t+dt$ with $dt \to 0$.

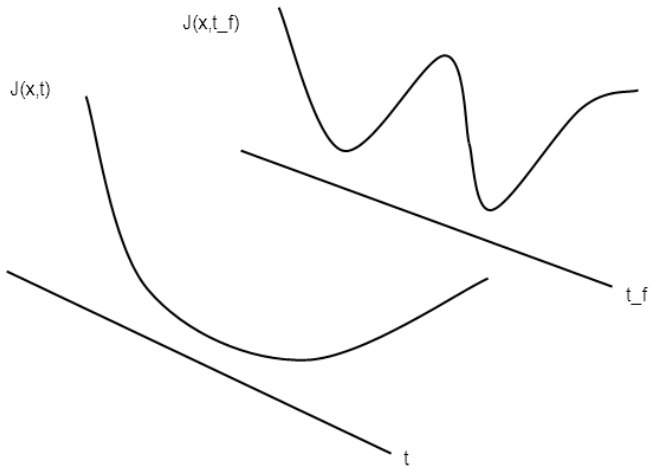
$$\begin{aligned} x_{t+dt} &= x_t + f(x_t, u_t, t)dt \\ C(x_0, u_{0\to T}) &= \phi(x_T) + \int_0^T d\tau R(\tau, x(\tau), u(\tau)) \end{aligned}$$

Assume $J(x,t)$ is smooth.

$$\begin{aligned} J(t,x) &= \min_u \left(R(t,x,u)dt + J(t+dt, x+f(x,u,t)dt)\right) \\ &\approx \min_u \left(R(t,x,u)dt + J(t,x) + \partial_t J(t,x)dt + \partial_x J(t,x) f(x,u,t)dt\right) \\ -\partial_t J(t,x) &= \min_u \left(R(t,x,u) + f(x,u,t)\partial_x J(x,t)\right) \end{aligned}$$

with boundary condition $J(x,T) = \phi(x)$.

Continuous limit



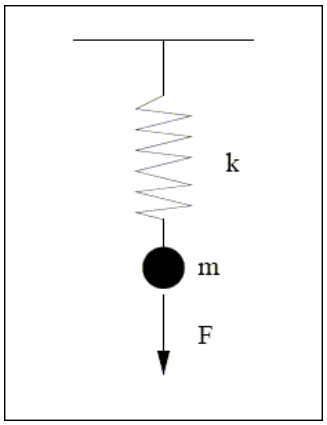
$$-\partial_t J(t,x) \quad = \quad \min_u \left(R(t,x,u) + f(x,u,t)\partial_x J(x,t)\right)$$

with boundary condition $J(x,T) = \phi(x)$.

This is called the *Hamilton-Jacobi-Bellman Equation*.

Computes the *anticipated potential* $J(t,x)$ from the future potential $\phi(x)$.

Example: Mass on a spring



The spring force $F_z = -z$ towards the rest position and control force $F_u = u$.

Newton's Law

$$F = -z + u = m\ddot{z}$$

with $m = 1$.

Control problem: Given initial position and velocity $z(0) = \dot{z}(0) = 0$ at time $t = 0$, find the control path $-1 < u(0 \to T) < 1$ such that $z(T)$ is maximal.

Example: Mass on a spring

Introduce $x_1 = z, x_2 = \dot{z}$, then

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= -x_1 + u \end{aligned}$$

The end cost is $\phi(x) = -x_1$; path cost $R(x, u, t) = 0$.

The HJB takes the form:

$$\begin{aligned} -\partial_t J &= \min_u \left(x_2 \frac{\partial J}{\partial x_1} - x_1 \frac{\partial J}{\partial x_2} + \frac{\partial J}{\partial x_2} u \right) \\ &= x_2 \frac{\partial J}{\partial x_1} - x_1 \frac{\partial J}{\partial x_2} - \left| \frac{\partial J}{\partial x_2} \right|, \quad u = -\text{sign} \left(\frac{\partial J}{\partial x_2} \right) \end{aligned}$$

Example: Mass on a spring

We try $J(t,x) = \psi_1(t)x_1 + \psi_2(t)x_2 + \alpha(t)$. The HJBE reduces to the ordinary differential equations

$$\begin{aligned} \dot{\psi}_1 &= \psi_2 \\ \dot{\psi}_2 &= -\psi_1 \\ \dot{\alpha} &= -|\psi_2| \end{aligned}$$

These equations must be solved for all t, with final boundary conditions $\psi_1(T) = -1, \psi_2(T) = 0$ and $\alpha(T) = 0$.

Note, that the optimal control only requires $\partial_x J(x,t)$, which in this case is $\psi(t)$ and thus we do not need to solve α. The solution for ψ is

$$\begin{aligned} \psi_1(t) &= -\cos(t-T) \\ \psi_2(t) &= \sin(t-T) \end{aligned}$$

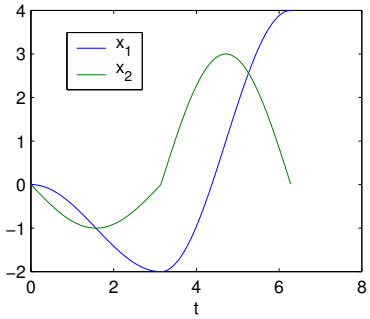
Example: Mass on a spring

The optimal control is

$$u(x,t) = -\text{sign}(\psi_2(t)) = -\text{sign}(\sin(t-T))$$

As an example consider $T = 2\pi$. Then, the optimal control is

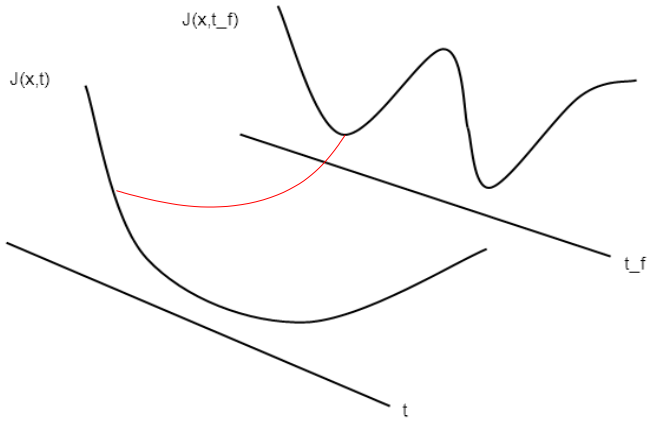
$$u = -1, \quad 0 < t < \pi$$
$$u = 1, \quad \pi < t < 2\pi$$



Pontryagin minimum principle

The HJB equation is a PDE with boundary condition at future time. The PDE is solved using discretization of space and time.

The solution is an optimal cost-to-go for all x and t. From this we compute the optimal trajectory and optimal control.



An alternative approach is a variational approach that directly finds the optimal trajectory and optimal control.

Pontryagin minimum principle

We can write the optimal control problem as a constrained optimization problem with independent variables $u(0 \to T)$ and $x(0 \to T)$

$$\min_{u(0\to T),x(0\to T)} \phi(x(T)) + \int_0^T dt R(x(t), u(t), t)$$

subject to the constraint

$$\dot{x} = f(x, u, t)$$

and boundary condition $x(0) = x_0$.

Introduce the Lagrange multiplier function $\lambda(t)$:

$$\begin{aligned} C &= \phi(x(T)) + \int_0^T dt \left[R(t, x(t), u(t)) - \lambda(t)(f(t, x(t), u(t)) - \dot{x}(t))\right] \\ &= \phi(x(T)) + \int_0^T dt[-H(t, x(t), u(t), \lambda(t)) + \lambda(t)\dot{x}(t))] \\ -H(t, x, u, \lambda) &= R(t, x, u) - \lambda f(t, x, u) \end{aligned}$$

Derivation PMP

The solution is found by extremizing C. This gives a necessary but not sufficient condition for a solution.

If we vary the action wrt to the trajectory x, the control u and the Lagrange multiplier λ, we get:

$$\begin{aligned}
\delta C &= \phi_x(x(T))\delta x(T) \\
&+ \int_0^T dt[-H_x\delta x(t) - H_u\delta u(t) + (-H_\lambda + \dot{x}(t))\delta\lambda(t) + \lambda(t)\delta\dot{x}(t)] \\
&= (\phi_x(x(T)) + \lambda(T))\,\delta x(T) \\
&+ \int_0^T dt\left[(-H_x - \dot{\lambda}(t))\delta x(t) - H_u\delta u(t) + (-H_\lambda + \dot{x}(t))\delta\lambda(t)\right]
\end{aligned}$$

For instance, $H_x = \frac{\partial H(t,x(t),u(t),\lambda(t))}{\partial x(t)}$.

We can solve $H_u(t, x, u, \lambda) = 0$ for u and denote the solution as

$$u^*(t, x, \lambda)$$

Assumes H convex in u.

The remaining equations are

$$\begin{aligned} \dot{x} &= H_\lambda(t, x, u^*(t, x, \lambda), \lambda) \\ \dot{\lambda} &= -H_x(t, x, u^*(t, x, \lambda), \lambda) \end{aligned}$$

with boundary conditions

$$x(0) = x_0 \qquad \lambda(T) = -\phi_x(x(T))$$

Mixed boundary value problem.

Again mass on a spring

Problem

$$\begin{aligned} \dot{x_1} &= x_2, \qquad \dot{x_2} = -x_1 + u \\ R(x,u,t) &= 0 \qquad \phi(x) = -x_1 \end{aligned}$$

Hamiltonian

$$\begin{aligned} H(t,x,u,\lambda) &= -R(t,x,u) + \lambda^T f(t,x,u) = \lambda_1 x_2 + \lambda_2(-x_1 + u) \\ H^*(t,x,\lambda) &= \lambda_1 x_2 - \lambda_2 x_1 - |\lambda_2| \qquad u^* = -\text{sign}(\lambda_2) \end{aligned}$$

The Hamilton equations

$$\begin{aligned} \dot{x} = \frac{\partial H^*}{\partial \lambda} \qquad &\Rightarrow \qquad \dot{x_1} = x_2, \qquad \dot{x_2} = -x_1 - \text{sign}(\lambda_2) \\ \dot{\lambda} = -\frac{\partial H^*}{\partial x} \qquad &\Rightarrow \qquad \dot{\lambda_1} = \lambda_2, \qquad \dot{\lambda_2} = -\lambda_1 \end{aligned}$$

with $x(t=0) = x_0$ and $\lambda(t=T) = (1,0)$.

Example

Consider the control problem:

$$\begin{aligned} dx &= udt \\ C &= \frac{\alpha}{2}x(T)^2 + \int_{t_0}^{T} dt \frac{1}{2}u(t)^2 \end{aligned}$$

with initial condition $x(t_0)$.

Solve the control problem using the PMP formalism.

Solution

The PMP recipe is

1. Construct the Hamiltonian

$$H(t,x,u,\lambda) = -R(t,x,u) + \lambda f(t,u,x) = -\frac{1}{2}u^2 + \lambda u$$

2. Construct the optimized Hamiltonian

$$H^*(t,x,\lambda) = H(t,x,u^*,\lambda) = \frac{1}{2}\lambda^2 \qquad u^* = \lambda$$

3. Solve the Hamilton equations of motion

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial H^*}{\partial \lambda} = \lambda \\ \frac{d\lambda}{dt} &= -\frac{\partial H^*}{\partial x} = 0 \end{aligned}$$

with boundary conditions $x(t_0)$ and $\lambda(t = T) = -\alpha x(T)$[4]. The solution for λ is constant $\lambda(t) = \lambda = -\alpha x(T)$. The solution for $x(t)$ is

$$x(t) \quad = \quad x(t_0) + \lambda(t - t_0)$$

Combining these two results, we get $\lambda = -\alpha x(T) = -\alpha(x(t_0) + \lambda(T - t_0))$, or

$$\lambda = \frac{-\alpha x(t_0)}{1 + \alpha(T - t_0)}$$

Since $u^* = \lambda$, this is the optimal control law.

[4]Note, that $\phi(x) = \frac{\alpha}{2}x^2$ so that $\phi_x = \alpha x$.

Brownian bridge

Due to certainty equivalence, this is also the optimal control law for

$$\begin{aligned} dx &= udt + d\xi \\ C &= \left\langle \frac{\alpha}{2} x(T)^2 + \int_{t_0}^{T} dt \frac{1}{2} u(t)^2 \right\rangle \end{aligned}$$

For $\alpha \to \infty$ the process is known as a Brownian bridge.

The control law and dynamics becomes

$$\begin{aligned} dx &= udt + d\xi \\ u &= \frac{-x(t_0)}{T - t_0} \end{aligned}$$

$x(T) \to 0$ w.p. 1.

Relation to classical mechanics

The equations look like classical mechanics

$$\begin{aligned} \dot{x} &= H_\lambda(t, x, u^*(t, x, \lambda), \lambda) \qquad x(0) = x_0 \\ \dot{\lambda} &= -H_x(t, x, u^*(t, x, \lambda), \lambda) \qquad \lambda(T) = -\phi_x(x(T)) \end{aligned}$$

In classical mechanics H is called the Hamiltonian. Consider the time evolution of H:

$$\begin{aligned} \dot{H} &= H_t + H_u \dot{u} + H_x \dot{x} + H_\lambda \dot{\lambda} = H_t \\ H(t, x, u, \lambda) &= -R(t, x, u) + \lambda f(t, u, x) \end{aligned}$$

So, for problems where R, f do not explicitly depend on time, H is a constant of the motion.

Example

Consider the control problem:

$$\begin{aligned} dx &= udt \\ C &= \int_{t_0}^{T} dt \frac{1}{2} u(t)^2 + V(x(t)) \end{aligned}$$

with initial condition $x(t_0)$.

1. $H(x, u, \lambda) = -\frac{1}{2}u^2 - V(x) + \lambda u$
2. $u^* = \lambda$, $H^*(x, \lambda) = \frac{1}{2}\lambda^2 - V(x)$
3.

$$\dot{x} = \frac{\partial H^*}{\partial \lambda} = \lambda \qquad \dot{\lambda} = -\frac{\partial H^*}{\partial \lambda} = \frac{\partial V(x)}{\partial x}$$

Control cost V play role of *minus* potential energy.
Control solution has constant *difference* of kinetic energy and state cost

Comments

The HJB method gives a sufficient (and often necessary) condition for optimality. The solution of the PDE is expensive.

The PMP method provides a necessary condition for optimal control. This means that it provides candidate solutions for optimality.

The PMP method is computationally less complicated than the HJB method because it does not require discretization of the state space.

Optimal control in continuous space and time contains many complications related to the existence, uniqueness and smoothness of the solution, particular in the absence of noise. In the presence of noise many of these intricacies disappear.

HJB generalizes to the stochastic case, PMP does not (at least not easy).

Stochastic differential equations

Consider the random walk on the line:

$$x_{t+1} = x_t + \xi_t \qquad \xi_t = \pm 1$$

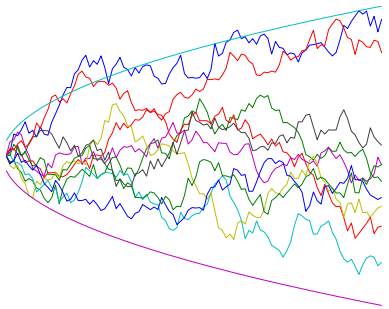
with $x_0 = 0$. We can compute

$$x_t = \sum_{i=1}^{t} \xi_i$$

Since x_t is a sum of random variables, x_t becomes Gaussian distributed with

$$\langle x_t \rangle = \sum_{i=1}^{t} \langle \xi_i \rangle = 0$$

$$\left\langle x_t^2 \right\rangle = \sum_{i,j=1}^{t} \left\langle \xi_i \xi_j \right\rangle = \sum_{i=1}^{t} \left\langle \xi_i^2 \right\rangle + \sum_{i,j=1, j\neq i}^{t} \left\langle \xi_i \xi_j \right\rangle = t$$



Note, that the fluctuations $\propto \sqrt{t}$.

Stochastic differential equations

In the continuous time limit we define

$$dx_t \quad = \quad x_{t+dt} - x_t = d\xi$$

with $d\xi$ an infinitesimal mean zero Gaussian variable with $\left\langle d\xi^2 \right\rangle = \nu dt$.

Then

$$\frac{d}{dt}\langle x \rangle \quad = \quad \lim_{dt\to 0}\left\langle \frac{x_{t+dt} - x_t}{dt} \right\rangle = \lim_{dt\to 0}\left\langle \frac{d\xi}{dt} \right\rangle = 0$$

$$\frac{d}{dt}\left\langle x^2 \right\rangle \quad = \quad \lim_{dt\to 0}\left\langle \frac{x_{t+dt}^2 - x_t^2}{dt} \right\rangle = \lim_{dt\to 0}\left\langle \frac{(x_t + d\xi)^2 - x_t^2}{dt} \right\rangle = \lim_{dt\to 0}\left\langle \frac{d\xi^2}{dt} \right\rangle = \nu$$

So for initial state x_0, $\langle x \rangle (t) = x_0$ and $\left\langle x^2 \right\rangle (t) = \nu t$ which fully specifies the Gaussian distribution:

$$\rho(x, t|x_0, 0) \quad = \quad \frac{1}{\sqrt{2\pi\nu t}} \exp\left(-\frac{(x - x_0)^2}{2\nu t} \right)$$

Consider the stochastic differential equation

$$x(t+dt) = x(t) + f(x(t),t)dt + \xi(t)$$

ξ is a Wiener process with $\langle \xi \rangle = 0, \left\langle \xi^2 \right\rangle = \nu dt$.

The probability to find the particle at y at time $t+dt$ given that it was at x at time t is given by

$$p(y,t+dt|x,t) \quad = \quad \langle \delta(y - x - f(x,t)dt - \xi) \rangle_\xi$$

where $\langle \rangle_\xi$ is expectation wrt the Wiener process.

Kolmogorov backward equation

Define $\psi(x,t) = p(z,T|x,t)$ the probability to reach a future state z at time T, given that it is currently at x,t. Clearly,

$$\begin{aligned}
\psi(x,t) &= p(z,T|x,t) = \int dy p(z,T|y,t+dt)p(y,t+dt|x,t) \\
&= \int dy \psi(y,t+dt)\left\langle \delta(y-x-f(x,t)dt-\xi)\right\rangle_\xi \\
&= \left\langle \psi(x+f(x,t)dt+\xi,t+dt)\right\rangle_\xi \\
&= \psi(x,t) + dt\partial_t\psi(x,t) + \left\langle f(x,t)dt+\xi\right\rangle_\xi \nabla\psi(x,t) \\
&+ \frac{1}{2}\left\langle (f(x,t)dt+\xi)^2\right\rangle_\xi \nabla^2\psi(x,t)
\end{aligned}$$

Thus,

$$-\partial_t\psi(x,t) = f(x,t)\nabla\psi(x,t) + \frac{1}{2}\nu\nabla^2\psi(x,t) \qquad \psi(x,T) = \delta(z-x)$$

This equation is known as the Kolmogorov backwards equation.

Fokker Plank (forward) equation

We can similarly derive a forward equation for the quantity $\rho(x,t) = p(x,t|x_0,0)$.

$$\begin{aligned}
\rho(y,t+dt) &= \int dx p(y,t+dt|x,t)\rho(x,t) \\
&= \int dx \langle \delta(y-x-f(x,t)dt-\xi)\rangle_\xi \rho(x,t) \\
&= \frac{1}{1+f'(y,t)dt}\langle \rho(y-f(y,t)dt-\xi,t)\rangle_\xi \\
&= \frac{1}{1+f'(y,t)dt}\langle \rho(y,t)-(f(y,t)dt+\xi)\nabla\rho(y,t)\rangle \\
&+ \left\langle \frac{1}{2}(f(y,t)dt+\xi)^2\nabla^2\rho(y,t)\right\rangle_\xi \\
&= \rho(y,t)-\nabla(f(y,t)\rho(y,t))dt+\frac{1}{2}\nu\nabla^2\rho(y,t)dt
\end{aligned}$$

Thus,

$$\partial_t \rho(x,t) = -\nabla(f(x,t)\rho(x,t)) + \frac{1}{2}\nu\nabla^2\rho(x,t), \qquad \rho(x,0) = \delta(x - x_0)$$

Example: Brownian motion

$$
\begin{aligned}
dx &= d\xi \qquad \left\langle d\xi^2 \right\rangle = \nu dt \\
\rho(x,t) &= p(x,t|x_0,0) = \frac{1}{\sqrt{2\pi\nu t}} \exp\left(-\frac{(x-x_0)^2}{2\nu t}\right) \\
\psi(x,t) &= p(z,T|x,t) = \frac{1}{\sqrt{2\pi\nu(T-t)}} \exp\left(-\frac{(x-z)^2}{2\nu(T-t)}\right)
\end{aligned}
$$

Forward and backward drift

For

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

The *expected forward drift* is

$$\langle dx \rangle = f(x,t)dt$$

The *expected backward drift* given $x(t+dt) = y$ can be computed using Bayes' rule:

$$\begin{aligned} p(y, t-dt|x,t) &= \frac{p(x,t|y,t-dt)\rho(y,t-dt)}{\rho(x,t)} \\ p(x,t|y,t-dt) &= \langle \delta(x-y-f(y,t-dt)dt-\xi) \rangle_\xi \end{aligned}$$

$$
\begin{aligned}
\langle x(t) - y(t-dt)\rangle_{x(t)=x} &= \int dy (x-y) p(y, t-dt|x,t) \\
&= \int dy (x-y) \langle \delta(x-y-f(y,t-dt)dt - \xi)\rangle \frac{\rho(y,t-dt)}{\rho(x,t)} \\
&= \frac{1}{\rho(x,t)} \left\langle \frac{1}{1+f'(x,t)dt} (f(x,t)dt + \xi)\rho(x - f(x,t)dt - \xi, t-dt) \right\rangle + O(dt^2) \\
&= \frac{1}{\rho(x,t)} \frac{1}{1+f'(x,t)dt} \langle (f(x,t)dt + \xi)(\rho(x,t) - \xi\rho'(x,t)\rangle + O(dt^2) \\
&= f(x,t)dt - \nu \nabla \log \rho(x,t) dt + O(dt^2) \equiv \tilde{f}(x,t)dt
\end{aligned}
$$

We see that the forward and backward drifts are different: given that we are at time t at location x the expected future drift is given by $f(x,t)$. The expected past drift into x is given by $\tilde{f}(x,t) = f(x,t) - \nu \nabla \log \rho(x,t)$.

Example: Brownian motion

$$
\begin{aligned}
dx &= d\xi \qquad x(0) = 0 \quad \left\langle d\xi^2 \right\rangle = \nu dt \\
\rho(x,t) &= \frac{1}{\sqrt{2\pi\nu t}} \exp\left(-\frac{x^2}{2\nu t}\right) \\
f(x,t) &= 0 \\
\tilde{f}(x,t) &= -\frac{x}{t}
\end{aligned}
$$

Stochastic optimal control

Consider a stochastic dynamical system

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

$d\xi$ Gaussian noise $\left\langle d\xi_i d\xi_j \right\rangle = \nu_{ij}(t, x, u)dt$.

The cost becomes an expectation:

$$C(t, x, u(t \to T)) \quad = \quad \left\langle \phi(x(T)) + \int_t^T d\tau R(t, x(t), u(t)) \right\rangle$$

over all stochastic trajectories starting at x with control path $u(t \to T)$.

Note, that $u(t)$ as part of $u(t \to T)$ is used at time t. Next move to $x + dx$ and repeat the optimization.

Stochastic optimal control

We obtain the Bellman recursion

$$
\begin{aligned}
J(t, x_t) &= \min_{u_t} R(t, x_t, u_t) + \langle J(t + dt, x_{t+dt}) \rangle \\
\langle J(t + dt, x_{t+dt}) \rangle &= \int dx_{t+dt} \mathcal{N}(x_{t+dt} | x_t, \nu dt) J(t + dt, x_{t+dt}) \\
&= J(t, x_t) + dt \partial_t J(t, x_t) + \langle dx \rangle \partial_x J(t, x_t) + \frac{1}{2} \left\langle dx^2 \right\rangle \partial_x^2 J(t, x_t) \\
\langle dx \rangle &= f(x, u, t) dt \\
\left\langle dx^2 \right\rangle &= \nu(t, x, u) dt
\end{aligned}
$$

Thus,

$$
-\partial_t J(t, x) = \min_u \left(R(t, x, u) + f(x, u, t) \partial_x J(x, t) + \frac{1}{2} \nu(t, x, u) \partial_x^2 J(x, t) \right)
$$

with boundary condition $J(x, T) = \phi(x)$.

Linear Quadratic control

The dynamics is linear

$$dx = [A(t)x + B(t)u + b(t)]dt + \sum_{j=1}^{m}(C_j(t)x + D_j(t)u + \sigma_j(t))d\xi_j, \quad \left\langle d\xi_j d\xi_{j'} \right\rangle = \delta_{jj'}dt$$

The cost function is quadratic

$$\begin{aligned} \phi(x) &= \frac{1}{2}x^T G x \\ R(x,u,t) &= \frac{1}{2}x^T Q(t)x + u^T S(t)x + \frac{1}{2}u^T R(t)u \end{aligned}$$

In this case the optimal cost-to-go is quadratic in x:

$$\begin{aligned} J(t,x) &= \frac{1}{2}x^T P(t)x + \alpha^T(t)x + \beta(t) \\ u(t) &= -\Psi(t)x(t) - \psi(t) \end{aligned}$$

Substitution in the HJB equation yields ODEs for P, α, β:

$$
\begin{aligned}
-\dot{P} &= PA + A^T P + \sum_{j=1}^{m} C_j^T P C_j + Q - \hat{S}^T \hat{R}^{-1} \hat{S} \\
-\dot{\alpha} &= [A - B\hat{R}^{-1}\hat{S}]^T \alpha + \sum_{j=1}^{m} [C_j - D_j \hat{R}^{-1}\hat{S}]^T P \sigma_j + Pb \\
\dot{\beta} &= \frac{1}{2} \left| \sqrt{\hat{R}} \psi \right|^2 - \alpha^T b - \frac{1}{2} \sum_{j=1}^{m} \sigma_j^T P \sigma_j \\
\hat{R} &= R + \sum_{j=1}^{m} D_j^T P D_j \\
\hat{S} &= B^T P + S + \sum_{j=1}^{m} D_j^T P C_j \\
\Psi &= \hat{R}^{-1} \hat{S} \\
\psi &= \hat{R}^{-1} (B^T \alpha + \sum_{j=1}^{m} D_j^T P \sigma_j)
\end{aligned}
$$

with $P(t_f) = G$ and $\alpha(t_f) = \beta(t_f) = 0$.

Example

Find the optimal control for the dynamics

$$dx = (x+u)dt + d\xi, \qquad \left\langle d\xi^2 \right\rangle = \nu dt$$

with end cost $\phi(x) = 0$ and path cost $R(x,u) = \frac{1}{2}(Qx^2 + Ru^2)$.

The Ricatti equations reduce to

$$\begin{aligned} -\dot{P} &= 2P + Q - R^{-1}P^2 \\ -\dot{\alpha} &= (1 - R^{-1}P)\alpha = 0 \\ \dot{\beta} &= \frac{1}{2}R^{-1}\alpha^2 - \frac{1}{2}\nu P = -\frac{1}{2}\nu P \end{aligned}$$

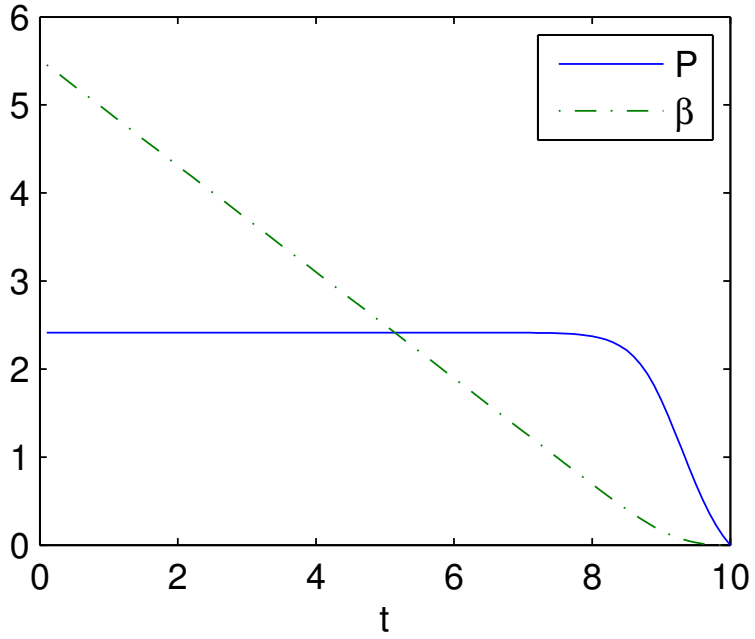
with $P(T) = \alpha(T) = \beta(T) = 0$ and

$$u(x,t) = -P(t)x$$

The solution is

$$P(t) = R\frac{\exp(2\sqrt{1+R^{-1}Q}(T-t))-1}{\frac{1}{1+\sqrt{1+R^{-1}Q}}\exp(2\sqrt{1+R^{-1}Q}(T-t))-\frac{1}{1-\sqrt{1+R^{-1}Q}}}$$

The optimal control is $u(x,t) = -R^{-1}P(t)x$.



Comments

Note, that in the last example the optimal control is independent of ν, i.e. optimal stochastic control equals optimal deterministic control.

In general:

- If $C_j = D_j = 0$ (only 'additive noise') $\dot{P}, \dot{\alpha}$ independent of noise σ, $\dot{\beta}$ depends on σ, but control independent of β. Thus control independent of σ (certainty equivalence)
- If $C_j \neq 0$ or $D_j \neq 0$, control depends on C_j, D_j, σ_j (no certainty equivalence)

Example: Portfolio selection

[5] Consider a market with p stocks and one bond. The bond price process is subject ot the following deterministic ordinary differential equation:

$$dP_0(t) = r(t)P_0(t)dt, \quad P_0(0) = p_0 > 0 \tag{1}$$

The other assets have price processes $P_i(t), i = 1, \ldots, p$ satisfying stochastic differential equations

$$dP_i(t) = P_i(t)\left(b_i(t)dt + \sum_{j=1}^{m} \sigma_{ij}(t)d\xi_j(t)\right), \quad P_i(0) = p_i > 0 \tag{2}$$

Consider an investor whose total wealth at time t is denoted by $x(t)$

$$x(t) = \sum_{i=0}^{p} N_i(t)P_i(t) \tag{3}$$

[5] This section is from [2] section 6.8 (pg. 335).

with N_i the number of stocks/bond of type i. Then

$$dx(t) = \left(r(t)x(t) + \sum_{i=1}^{p} (b_i(t) - r(t))u_i(t) \right) dt + \sum_{i=1}^{p} \sum_{j=1}^{m} \sigma_{ij}(t)u_i(t)d\xi_j(t) \tag{4}$$

with $u_i(t) = N_i(t)P_i(t), i = 1, \ldots, p$ the *portfolio* of the investor.

The objective of the investor is to maximize the mean terminal wealth $\left\langle x(t_f) \right\rangle$ and minimize at the same time the variance

$$\Sigma^2 = \left\langle x(t_f)^2 \right\rangle - \left\langle x(t_f) \right\rangle^2$$

This is a multi-objective optimization problem with an efficient frontier of optimal solutions: for each given mean there is a minimial variance.

These pairs can be found by minimizing the single objective criterion

$$\mu\Sigma^2 - \left\langle x(t_f) \right\rangle \tag{5}$$

for different values of the weighting factor μ.

This objective, however, is not an expectation value of some stochastic quantity due to the $\langle\cdot\rangle^2$ term. Consider a slightly different problem, minimizing the objective

$$\left\langle \mu x(t_f)^2 - \lambda x(t_f) \right\rangle \tag{6}$$

which is of the standard stochastic optimization form. One can show that one can construct a solution of Problem 5 by solving problem 6 for suitable $\lambda(\mu)$. [6]

Our goal is thus to minimize eq. 6 subject to the stochastic dynamics eq. 4.

This is an LQ problem. The solution is computed from the Ricatti equations

$$u_i(x,t) = \psi_i(t)x + \phi_i(t)$$

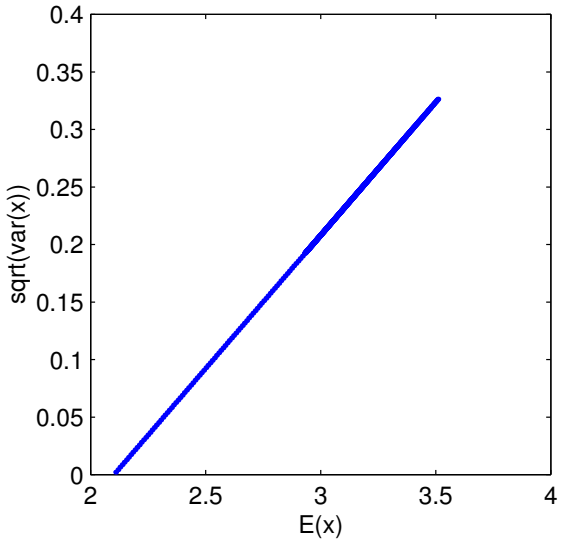
As an example we consider the simplest possible case: $p = m = 1$ and r, b, σ independent of time.

[6] and finding λ from

$$\lambda = 1 + \mu \left\langle x(t_f) \right\rangle (\lambda, \mu)$$

([2] Theorem 8.2 pg. 338)

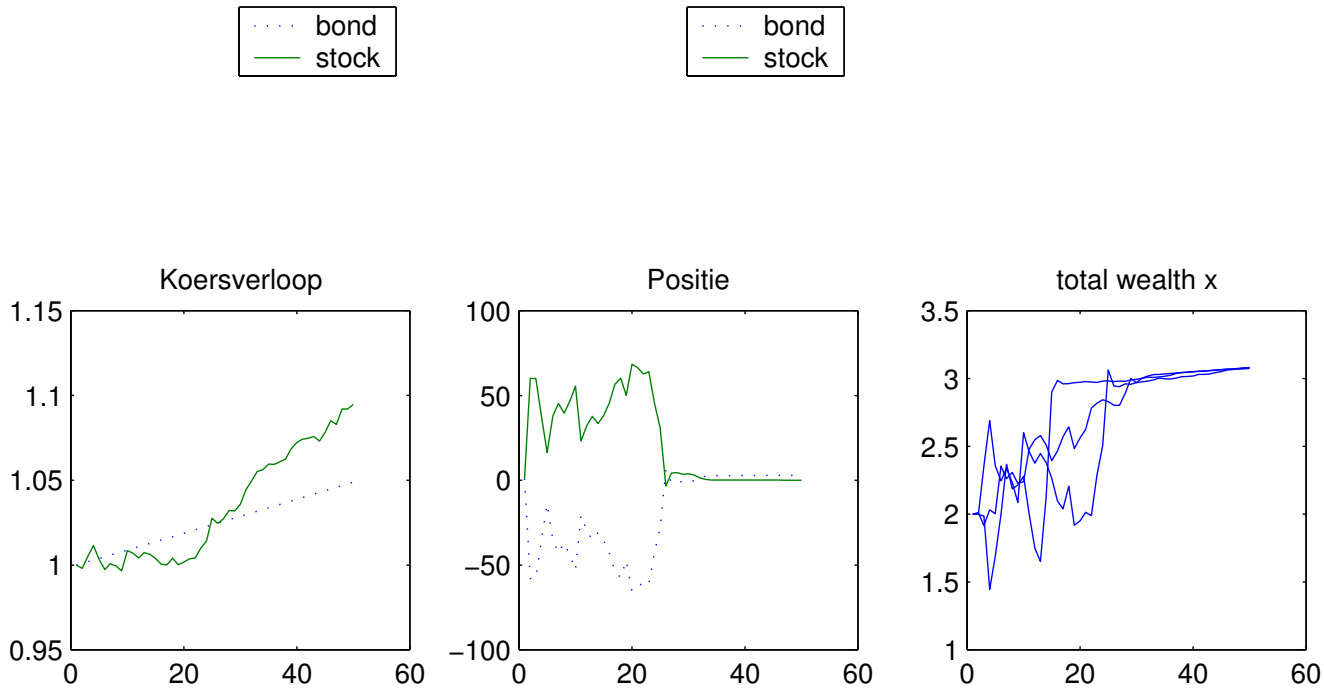
Efficient boundary



Parameter values are: $p = m = 1$. Trading period is one year weekly. annual bond rate 5 % ($r = 0.0009758$), annual expected stock rate is 10 % ($b = 0.0019$), volatility $\sigma = 2b$. $x_0 = 2$. Shows var x versus $\langle x \rangle$ scatter plot for various values of μ. Small μ corresponds to risky investments with high expected return and large fluctuation. $\mu \to \infty$ corresponds to riskless investment in bond only and a return of 5 %.

$\mu = 10$ corresponds to $\langle x \rangle = 3$ and $\sqrt{var} = 0.2$.

Making money



Simulation of optimal control with $\mu = 10$, The optimal strategy is to borrow many stocks and sell them as soon as the objective is achieved.

Indeed, $\langle x \rangle = 3$ as expected. The strategy to get at this 50 % increase in wealth is to buy many stocks and hope they will give the expected wealth increase. As soon as this occurs, all stocks are sold and the money is put in the bank.

Lecture 3: Path integral control theory

Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

For instance:

- For LQ control problems the optimal control computation is equivalent to 'Kalman smoothing'.

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

Variational inference:

$p(x_{1:n}) = \pi(x_{1:n})/Z$ is a probability distribution, compute

$$p(x_1) \quad = \quad \sum_{x_{2:n}} p(x_{1:n})$$

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

Variational inference:

$p(x_{1:n}) = \pi(x_{1:n})/Z$ is a probability distribution, compute

$$p(x_1) \quad = \quad \sum_{x_{2:n}} p(x_{1:n})$$

Define free energy

$$F(q) \quad = \quad \sum_{x_{1:n}} q(x_{1:n}) \log \frac{q(x_{1:n})}{\pi(x_{1:n})}$$

F is minimized by $q = p$.

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

Variational inference:

$p(x_{1:n}) = \pi(x_{1:n})/Z$ is a probability distribution, compute

$$p(x_1) \quad = \quad \sum_{x_{2:n}} p(x_1, x_{2:n})$$

Define free energy

$$F(q) \quad = \quad \sum_{x_{1:n}} q(x_{1:n}) \log \frac{q(x_{1:n})}{\pi(x_{1:n})}$$

F is minimized by $q = p$.

Restrict minimization to simple distributions $q(x_{1:n}) = q_1(x_1) \dots q_n(x_n)$ and minimize

$$p(x_1) \approx q_1(x_1)$$

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

Efficient inference:

- Variational inference, TAP
- Belief propagation, EP, Cluster Variation Method, Survey propagation
- convex relaxations
- Monte Carlo Sampling

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

In particular:

- Consider a class of control problems for which the Bellman equation can be transformed in a linear pde (using a log transform)
- 'Solve' as a Feynman-Kac path integral

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient state-of-the-art inference methods

The log transform first used in QM:

$$\hbar i \partial_t \Psi = H\Psi \qquad H(x,t) = V(x,t) - \frac{\hbar^2}{2}\partial_x^2$$

Write

$$\Psi \quad = \quad \sqrt{\rho}\exp\left(i\frac{S}{\hbar}\right)$$

then

$$\begin{aligned} -\partial_t S &= \frac{1}{2}(\nabla_x S)^2 - \frac{1}{2}\hbar^2 \frac{\partial_x^2 \sqrt{\rho}}{\sqrt{\rho}} + V \\ -\partial_t \rho &= \nabla_x(\rho \nabla_x S) \end{aligned}$$

Later used in Burgers Equation, and by Fleming and Mitter for control.

Lecture 3: Path integral control theory

General idea:

- Express the control problem as an inference problem
- Use efficient approximate inference methods

In particular:

- Consider a class of control problems for which the Bellman equation looks like the Mandelung equation
- Use the log transform to convert it into a Schrödinger-like backward equation
- Identify this equation as a Kolmogorov backward equation.
- Identify the corresponding forward diffusion process

Path integral control

$$
\begin{aligned}
dx_i &= f_i(x,t)dt + \sum_j g_{ia}(x,t)(u_a dt + d\xi_a) \\
C(t,x,u(t\to T)) &= \left\langle \phi(x(T)) + \int_t^T ds V(x,t) + \frac{1}{2}\sum_{ab} R_{ab}u_a u_b \right\rangle
\end{aligned}
$$

with $\langle d\xi_a d\xi_b \rangle = \nu_{ab} dt$.

The cost is an expectation over all stochastic trajectories starting at x with control path $u(t \to T)$.

The stochastic HJB equation becomes

$$
-\partial_t J = \min_u \left(\frac{1}{2} u^T R u + V + (\nabla J)^T (f + gu) + \frac{1}{2}\mathrm{Tr}\left(g\nu g^T \nabla^2 J \right) \right)
$$

which we need to solve with end boundary condition $J(x, t_f) = \phi(x)$.

Path integral control

Minimization wrt u yields: [7]

$$\begin{aligned} u &= -R^{-1} g^T \nabla J \\ -\partial_t J &= -\frac{1}{2}(\nabla J)^T g R^{-1} g^T (\nabla J) + V + (\nabla J)^T f + \frac{1}{2}\mathrm{Tr}\left(g \nu g^T \nabla^2 J\right) \end{aligned}$$

(our 'Mandelung equation')

Define $\psi(x,t)$ through $J(x,t) = -\lambda \log \psi(x,t)$ and impose a relation between R and ν:

$$R = \lambda \nu^{-1}$$

with λ a positive number.

[7] $u_a = -\sum_b \left(R^{-1}\right)_{ab} g_{ib}(x,t) \frac{\partial J(x,t)}{\partial x_i}$

The relation $R = \lambda \nu^{-1}$

$$
\begin{aligned}
dx_i &= f_i(x)dt + \sum_a g_{ia}(x)(u_a dt + d\xi_a) \\
C &= \left\langle \phi(x(T)) + \int_t^T ds V(x) + \frac{1}{2}\sum_{ab} R_{ab} u_a u_b \right\rangle
\end{aligned}
$$

Noise and control act in the same sub-space. Directions where noise is large, control is cheap and visa versa.

The relation $R = \lambda \nu^{-1}$

$$
\begin{aligned}
dx_i &= f_i(x)dt + \sum_a g_{ia}(x)(u_a dt + d\xi_a) \\
C &= \left\langle \phi(x(T)) + \int_t^T ds V(x) + \frac{1}{2}\sum_{ab} R_{ab} u_a u_b \right\rangle
\end{aligned}
$$

Noise and control act in the same sub-space. Directions where noise is large, control is cheap and visa versa.

Can be alternatively understood as a KL divergence between controlled and uncontrolled trajectories:

$$
\sum_\tau p(\tau|u) \log \frac{p(\tau|u)}{p(\tau|0)} = \int_0^T dt \frac{1}{2} u^T \nu^{-1} u
$$

λ plays the role of temperature.

Path integral control

Then the HJB becomes *linear* in ψ

$$\partial_t \psi = \left(\frac{V}{\lambda} - f^T \nabla - \frac{1}{2} \mathrm{Tr}\left(g \nu g^T \nabla^2 \right) \right) \psi$$

with end condition $\psi(x, T) = \exp(-\phi(x)/\lambda)$ (our Kolmogorov backward equation) [8]

[8] We sketch the derivation for $g = 1$.

$$-\frac{1}{2}(\nabla J)^T R^{-1} (\nabla J) + \frac{1}{2}\mathrm{Tr}\left(\nu \nabla^2 J\right) = -\frac{1}{2} \sum_{ij} \nabla_i J R^{-1}_{ij} \nabla_j J + \frac{1}{2} \lambda \sum_{ij} R^{-1}_{ij} \nabla_{ij} J$$

$$= \frac{1}{2} \sum_{ij} R^{-1}_{ij} \left(-\nabla_i J \nabla_j J + \lambda \nabla_{ij} J \right)$$

$$= \frac{1}{2} \sum_{ij} R^{-1}_{ij} \left(-\lambda^2 \frac{1}{\psi} \nabla_{ij} \psi \right)$$

since

$$-\nabla_i J \nabla_j J = -\lambda^2 \frac{1}{\psi^2} \nabla_i \psi \nabla_j \psi$$

$$\nabla_{ij} J = -\lambda \nabla_i \nabla_j \log \psi = -\lambda \nabla_i \left(\frac{1}{\psi} \nabla_j \psi \right) = \lambda \frac{1}{\psi^2} \nabla_i \psi \nabla_j \psi - \lambda \frac{1}{\psi} \nabla_{ij} \psi$$

Path integral control

The linearity allows us to revers the direction of time.

We identify $\psi(x,t) \propto p(z,T|x,t)$, then the Bellman equation

$$\partial_t \psi = \left(\frac{V}{\lambda} - f^T \nabla - \frac{1}{2} \mathrm{Tr}\left(g \nu g^T \nabla^2 \right) \right) \psi$$

can be interpreted as a Kolmogorov backward equation for the process

$$\begin{aligned} dx_i &= f_i(x,t)dt + \sum_a g_{ia}(x,t) d\xi_a \\ x(t) &= \dagger \quad \text{with probability} \quad V(x,t)dt/\lambda \\ x(T) &= \dagger \quad \text{with probability} \quad \phi(x)/\lambda \end{aligned}$$

Path integral control

The correspondong forward equation is

$$\partial_t \rho = -\frac{V}{\lambda}\rho - \nabla(f\rho) + \frac{1}{2}\mathrm{Tr}\nabla^2 g\nu g^T \rho$$

with $\rho(x,t) = p(x,t|z,0)$ and $\rho(x,0) = \delta(x-z)$.

Feynman-Kac formula

Denote $Q(\tau|x, s)$ the distribution over uncontrolled trajectories that start at x, t:

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

with τ a trajectory $x(t \to T)$. Then

$$\psi(x,t) = \int dQ(\tau|x,t) \exp\left(-\frac{S(\tau)}{\lambda}\right)$$

$$S(\tau) = \phi(x(T)) + \int_t^T dsV(x(s),s)$$

ψ can be computed by forward sampling the uncontrolled process.

Posterior distribution over optimal trajectories

$\psi(x,t)$ can be interpreted as a partition sum for the distribution over paths under optimal control:

$$P(\tau|x,t) = \frac{1}{\psi(x,t)} Q(\tau|x,t) \exp\left(-\frac{S(\tau)}{\lambda}\right)$$

The optimal cost-to-go is a free energy:

$$J(x,t) = -\lambda \log \int dQ(\tau|x,t) \exp\left(-\frac{1}{\lambda} S(\tau)\right)$$

The optimal control is an expectation wrt P:

$$u(x,t)dt = -R^{-1} g^T(x,t) \nabla_x J(x,t) dt = \int dP(\tau) d\xi(\tau) = \langle d\xi \rangle_P$$

KL control theory

x denotes state of the agent and $x_{1:T}$ is a path through state space from time $t = 1$ to T.

$q(x_{1:T}|x_0)$ denotes a probability distribution over possible future trajectories given that the agent at time $t = 0$ is is state x_0, with

$$q(x_{1:T}|x_0) = \prod_{t=0}^{T} q(x_{t+1}|x_t)$$

$q(x_{t+1}|x_t)$ implements the allowed moves.

$V(x_{1:T}) = \sum_{t=1}^{T} V(x_t)$ is the total cost when following path $x_{1:T}$.

The KL control problem is to find the probability distribution $p(x_{1:T}|x_0)$ that minimizes

$$C(p|x_0) \quad = \quad \sum_{x_{1:T}} p(x_{1:T}|x_0) \left(\log \frac{p(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} + V(x_{1:T}) \right) = KL(p||q) + \langle V \rangle_p$$

KL control theory

$p(x_{1:T}|x_0)$ and $q(x_{1:T}|x_0)$ distributions over trajectories.

Given q, find p that minimizes

$$C(p|x_0) \quad = \quad KL(p||q) + \langle V \rangle_p$$

The solution and the optimal control cost are

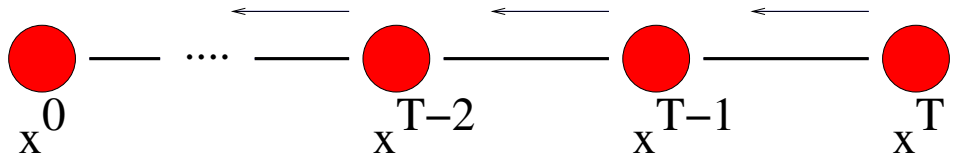
$$\begin{aligned} p(x_{1:T}|x_0) &= \frac{1}{\psi(x_0)} q(x_{1:T}|x_0) \exp\left(-V(x_{1:T})\right) \\ C &= -\log \psi(x_0) \\ \psi(x_0) &= \sum_{x_{1:T}} q(x_{1:T}|x_0) \exp\left(-V(x_{1:T})\right) \end{aligned}$$

NB: $\psi(x_0)$ is an integral over paths.

The optimal control at time $t = 0$ is given by

$$p(x_1|x_0) = \sum_{x_{2:T}} p(x_{2:T}|x_0) \propto q(x_1|x_0) \exp(-V(x_1))\beta_1(x_1)$$

with $\beta_t(x)$ the backward messages.



$$\begin{aligned} \beta_T(x_T) &= 1 \\ \beta_{t-1}(x_{t-1}) &= \sum_{x_t} q(x_t|x_{t-1}) \exp(-V(x_t))\beta_t(x_t) \end{aligned}$$

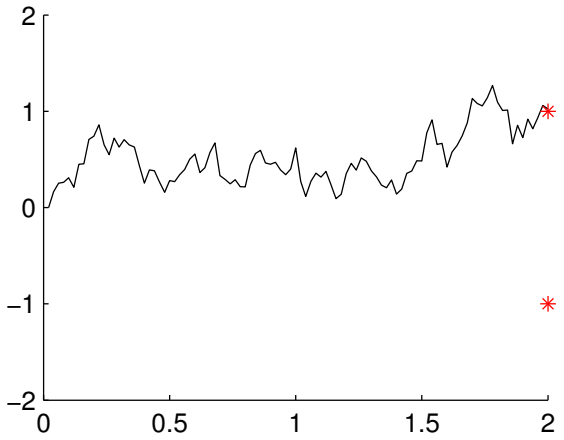
Link to continuous path integral formulation

The previous continuous path integral control can be obtained as a special case of the KL control formulation.

$$
\begin{aligned}
p(x_{t+dt}|x_t, u_t) &= \mathcal{N}(x_{t+dt}|x_t + f(x_t, t)dt + u(t, x_t)dt, \nu) \\
q(x_{t+dt}|x_t) &= \mathcal{N}(x_{t+dt}|x_t + f(x, t)dt, \nu) \\
C(p|x_0) &= KL(p|q) + \langle V \rangle \\
&= \sum_{x^{dt:T}} p(x^{dt:T}|x^0) \left(\sum_{t=dt}^{T} \frac{1}{2} u(t, x_t)^T \nu^{-1} u(t, x_t) + V(x_t) \right) \\
&\propto \left\langle \int_0^T dt \frac{1}{2} u(t, x_t)^T R u(t, x_t) + \lambda V(x_t) \right\rangle
\end{aligned}
$$

with $\lambda = R\nu$.

Control theory



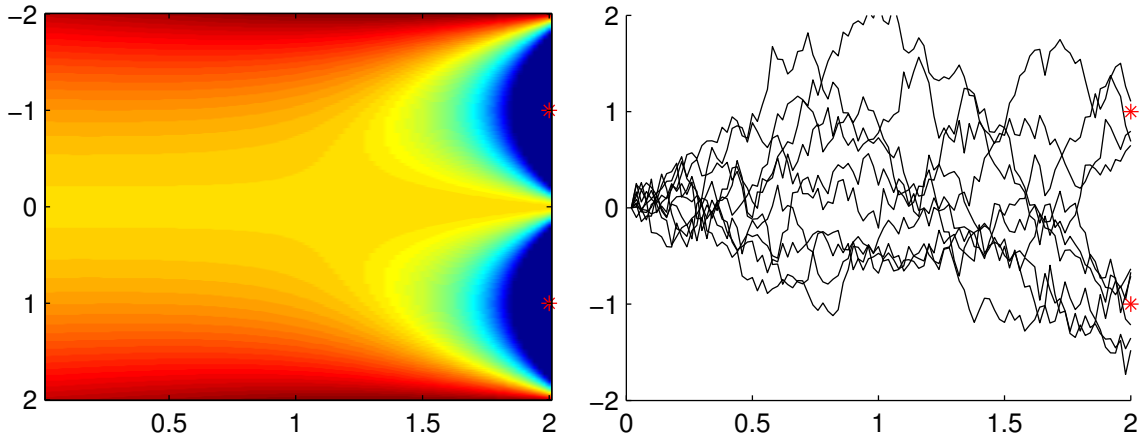
Consider a stochastic dynamical system

$$dX_t = f(X_t, u)dt + dW_t \qquad \mathbb{E}(dW_{t,i}dW_{t.j}) = \nu_{ij}dt$$

Given X_0 find control function $u(x,t)$ that minimizes the expected future cost

$$C = \mathbb{E}\left(\phi(X_T) + \int_0^T dtV(X_t, u(X_t, t))\right)$$

Control theory



Standard approach: define $J(x,t)$ is optimal cost-to-go from x,t.

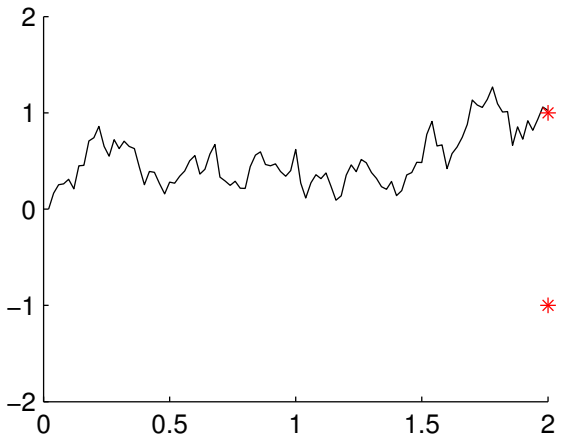
$$J(x,t) = \min u_{t:T} \mathbb{E}_u \left(\phi(X_T) + \int_t^T dt V(X_t, u(X_t,t)) \right) \qquad X_t = x$$

J satisfies a partial differential equation

$$-\partial_t J(t,x) \quad = \quad \min_u \left(V(x,u) + f(x,u)\nabla_x J(x,t) + \frac{1}{2}\nu \nabla_x^2 J(x,t) \right) \qquad J(x,T) = \phi(x)$$

with $u = u(x,t)$.This is HJB equation. Optimal control $u^*(x,t)$ defines distribution over trajectories $p^*(\tau)$ $(= p(\tau|x_0,0))$.

Path integral control theory

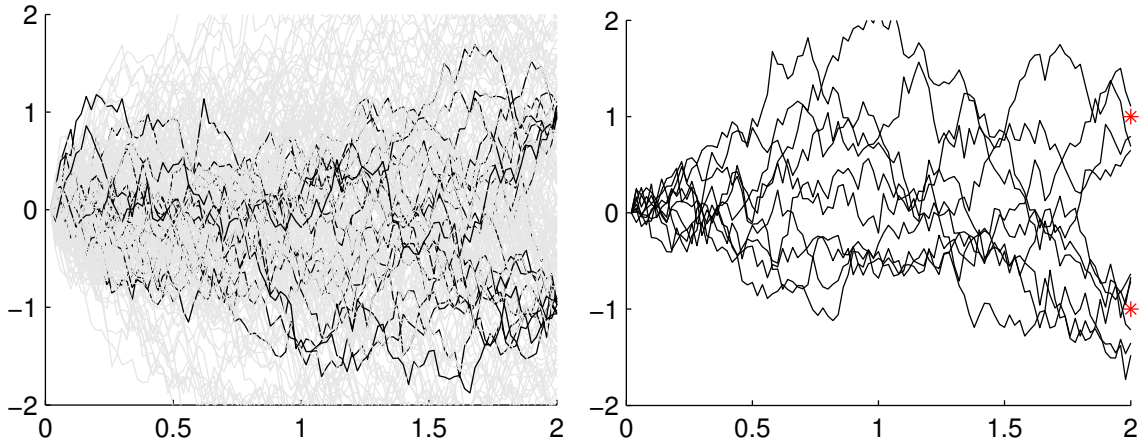


$$dX_t \quad = \quad f(X_t,t)dt + g(X_t,t)(u(X_t,t)dt + dW_t) \qquad X_0 = x_0$$

Goal is to find function $u(x,t)$ that minimizes

$$C = \mathbb{E}_u\left(S(\tau) + \int_0^T dt \frac{1}{2} u(X_t,t)^2 \right) \qquad S(\tau) \quad = \quad \phi(X_T) + \int_0^T V(X_t,t)$$

Path integral control theory



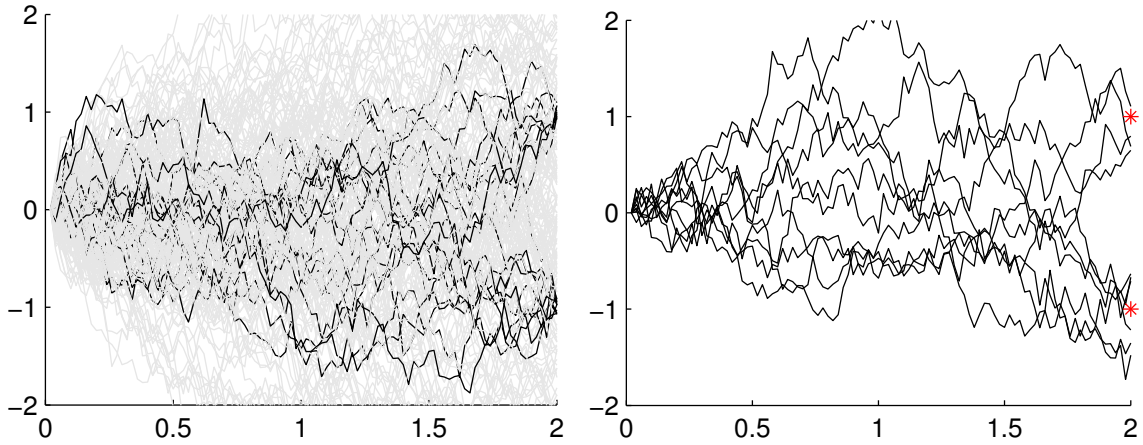
Equivalent formulation: Find distribution over trajectories p that minimizes

$$C(p) = \int d\tau p(\tau) \left(S(\tau) + \log \frac{p(\tau)}{q(\tau)} \right)$$

$q(\tau|x_0, 0)$ is distribution over *uncontrolled* trajectories.

The optimal solution is given by $p^*(\tau) = \frac{1}{\psi} q(\tau) e^{-S(\tau)}$

Path integral control theory



Equivalent formulation: Find distribution over trajectories p that minimizes

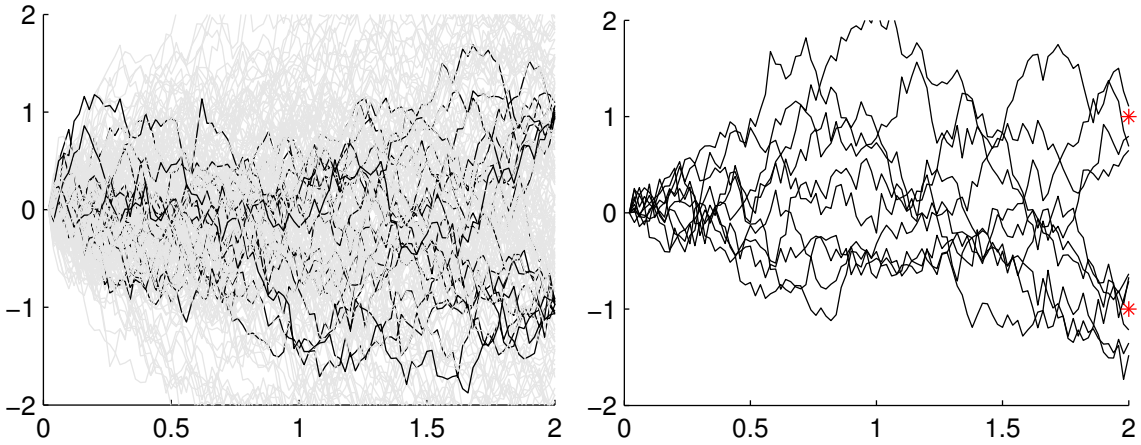
$$C(p) = \int d\tau p(\tau) \left(S(\tau) + \log \frac{p(\tau)}{q(\tau)} \right)$$

$q(\tau|x_0, 0)$ is distribution over *uncontrolled* trajectories.

The optimal solution is given by $p^*(\tau) = \frac{1}{\psi} q(\tau) e^{-S(\tau)} = p(\tau|u^*)$.

Equivalence of optimal control and discounted cost (Girsanov)

Path integral control theory



The optimal control cost is $C(p^*) = -\log \psi = J(x_0, 0)$ with

$$\psi = \int d\tau q(\tau) e^{-S(\tau)} = \mathbb{E}_q e^{-S}$$

$J(x,t)$ can be computed by forward sampling from q.

Recap

Control problem:

$$dx = f dt + g(u dt + d\xi) \qquad C = \left\langle \phi + \int_t^T V + \frac{1}{2} u^T R u \right\rangle \qquad R = \lambda \nu^{-1}$$

HJB is linear:

$$\partial_t \psi = H\psi \qquad J = -\lambda \log \psi$$

Solution is given by Feynman-Kac formula: $\psi = \int dQ(\tau) \exp\left(-\frac{S(\tau)}{\lambda}\right)$.
Q distribution over uncontrolled dynamics ($u = 0$).

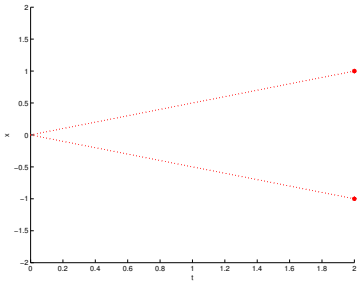
Posterior distribution over optimal controlled trajectories: $P(\tau) = \frac{1}{\psi} Q(\tau) \exp\left(-\frac{S(\tau)}{\lambda}\right)$.

Optimal control is expectation value: $u dt = \langle d\xi \rangle_P$.

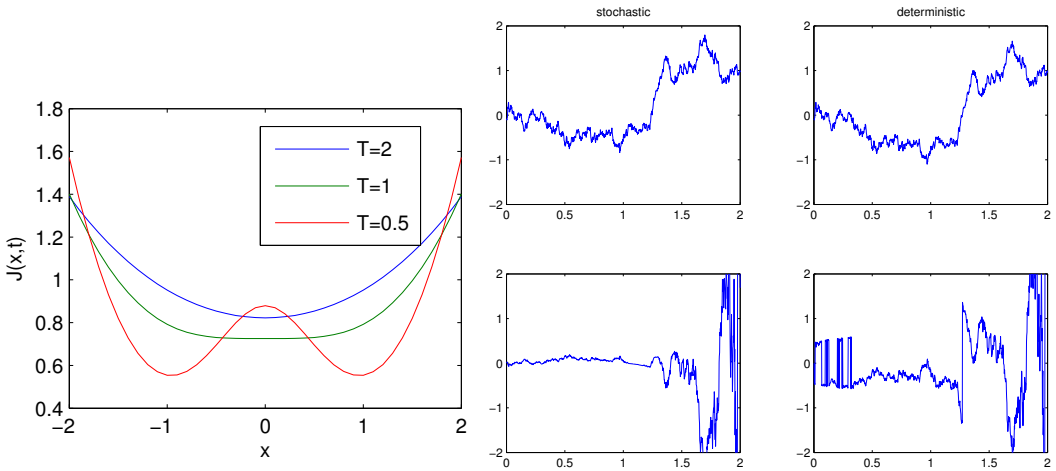
Delayed choice

$$dx = udt + d\xi \qquad \left\langle \xi^2 \right\rangle = \nu dt$$

$$C = \left\langle \phi(x_T) + \int_0^T dt \frac{1}{2} u(x_t, t)^2 \right\rangle$$

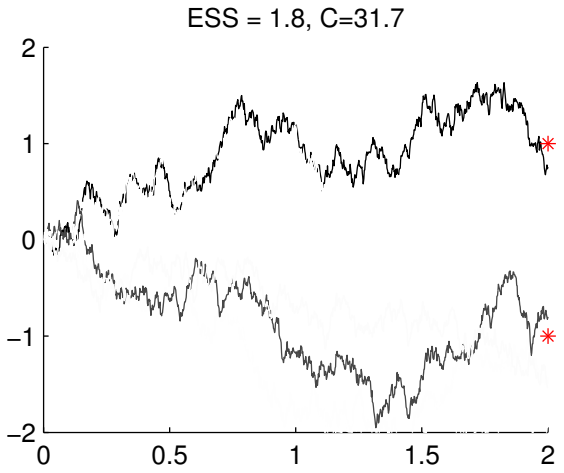


$\phi(x = \pm 1) = 0$ and $\phi(x) = \infty$, else.



”When the future is uncertain, delay your decisions.”

Estimating $\psi = \mathbb{E}e^{-S}$



Sample N trajectories from uncontrolled dynamics

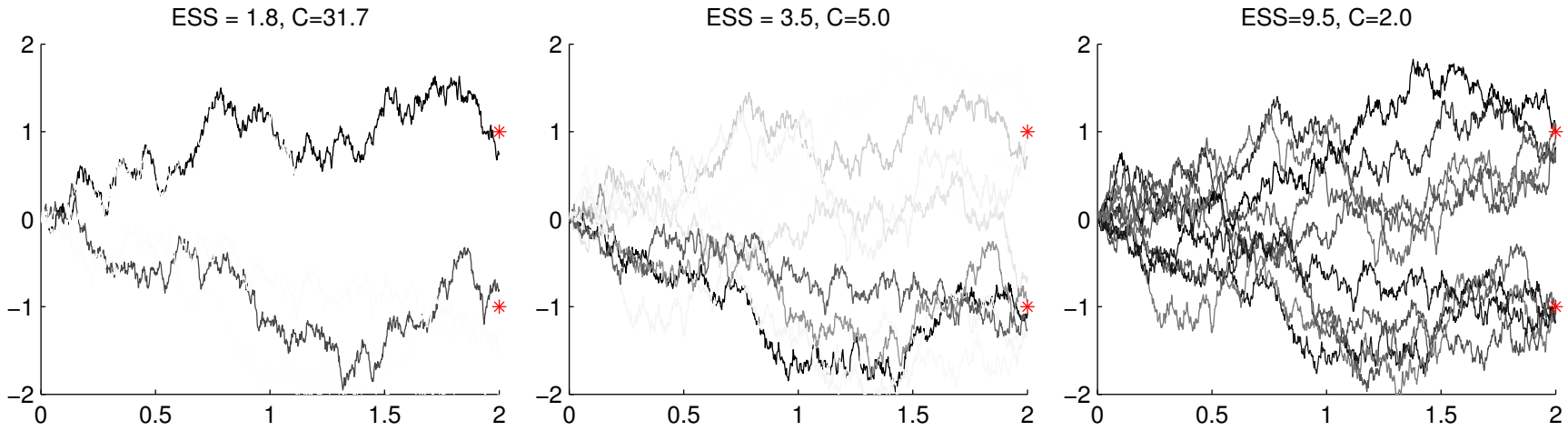
$$\tau_i \sim q(\tau) \qquad w_i = e^{-S(\tau_i)} \qquad \hat{\psi} = \frac{1}{N}\sum_i w_i$$

$\hat{\psi}$ unbiased estimate of ψ.

Sampling efficiency is inversely proportional to variance in (normalized) w_i.

$$ESS = \frac{N}{1 + N^2 Var(w)}$$

Importance sampling

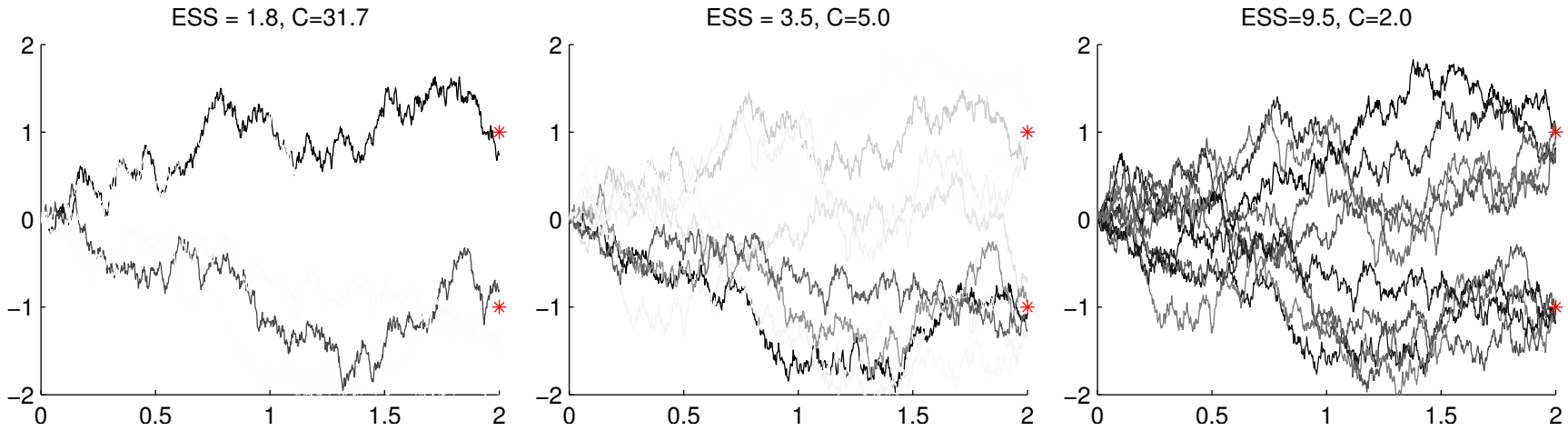


Sample N trajectories from controlled dynamics and reweight yields unbiased estimate of cost-to-go:

$$\tau_i \sim p(\tau) \qquad w_i = e^{-S(\tau_i)} \frac{q(\tau_i)}{p(\tau_i)} = e^{-S_u(\tau_i)} \qquad \hat{\psi} = \frac{1}{N} \sum_i w_i$$

$$S_u(\tau) = S(\tau) + \int_0^T dt \frac{1}{2} u(X_t, t)^2 + \int_0^T u(X_t, t) dW_t$$

Importance sampling



$$S_u(\tau) \quad = \quad S(\tau) + \int_0^T dt \frac{1}{2} u(X_t, t)^2 + \int_0^T u(X_t, t) dW_t$$

Thm:

• Better u (in the sense of optimal control) provides a better sampler (in the sense of effective sample size).

• Optimal $u = u^*$ (in the sense of optimal control) requires only one sample and $S_u(\tau)$ deterministic!

Thijssen, Kappen 2015

Proof

Control cost is $C(p) = \mathbb{E}_p\left(S(\tau) + \log \frac{p(\tau)}{q(\tau)}\right)$

Obviously: $C(p^*) \leq C(p)$ for all p, which is an instance of Jensen's inequality:

$$\begin{aligned} C^* &= -\log \sum_\tau q(\tau) e^{-S(\tau)} = -\log \sum_\tau p(\tau) e^{-S(\tau) - \log \frac{p(\tau)}{q(\tau)}} \\ &\leq \sum_\tau p(\tau)\left(S(\tau) + \log \frac{p(\tau)}{q(\tau)}\right) = C(p) \end{aligned}$$

The inequality is saturated when $S(\tau) + \log \frac{p(\tau)}{q(\tau)}$ has zero variance: left and right side evaluate to $S(\tau) + \log \frac{p(\tau)}{q(\tau)}$.

This is realized when $p = p^*$ [9].

[9] p^* exists when $\sum_\tau q(\tau) e^{-S(\tau)} < \infty$

The Path Integral Cross Entropy (PICE) method

We wish to estimate

$$\psi = \int d\tau q(\tau) e^{-S(\tau)}$$

The optimal (zero variance) importance sampler is $p^*(\tau) = \frac{1}{\psi} q(\tau) e^{-S(\tau)}$.

We approximate $p^*(\tau)$ with $p_{\hat{u}}(\tau)$, where $\hat{u}(x, t|\theta)$ is a parametrized control function.

Following the Cross Entropy method, we minimise $KL(p^*|p_{\hat{u}})$.

$$KL(p^*|p_{\hat{u}}) \quad \propto \quad -\mathbb{E}_{p^*} \log p_{\hat{u}} = -\mathbb{E}_{p_u} e^{-S_u} \log p_{\hat{u}}$$

with $u(x, t|\theta)$ arbitrary sampling control.

$$\Delta\theta \quad \propto \quad -\frac{KL(p^*|p_{\hat{u}})}{\partial\theta} \propto -\mathbb{E}_{\hat{u}} e^{-S_{\hat{u}}} \int_0^T dW_t \frac{\partial \hat{u}(X_t, t|\theta)}{\partial\theta}$$

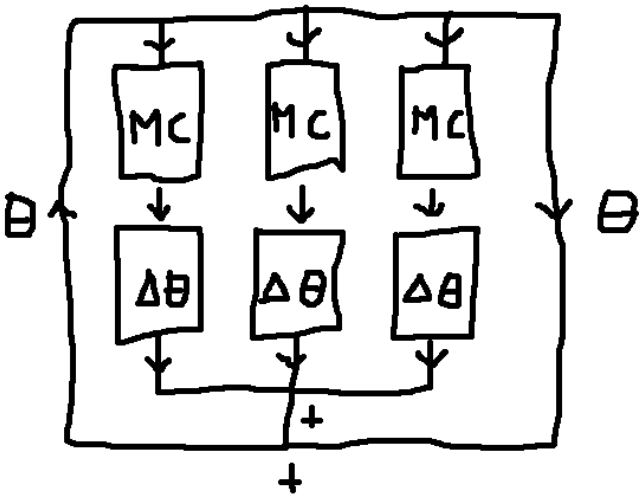
where in the last step we set $u = \hat{u}$.

[3]

Adaptive importance sampling

for $k = 0, \ldots$ **do**
  $data_k$ = generate_data($model, u_k$) % Importance sampler
  u_{k+1} = learn_control($data_k, u_k$) % Gradient descent
end for

Parallel sampling
Parallel gradient computation

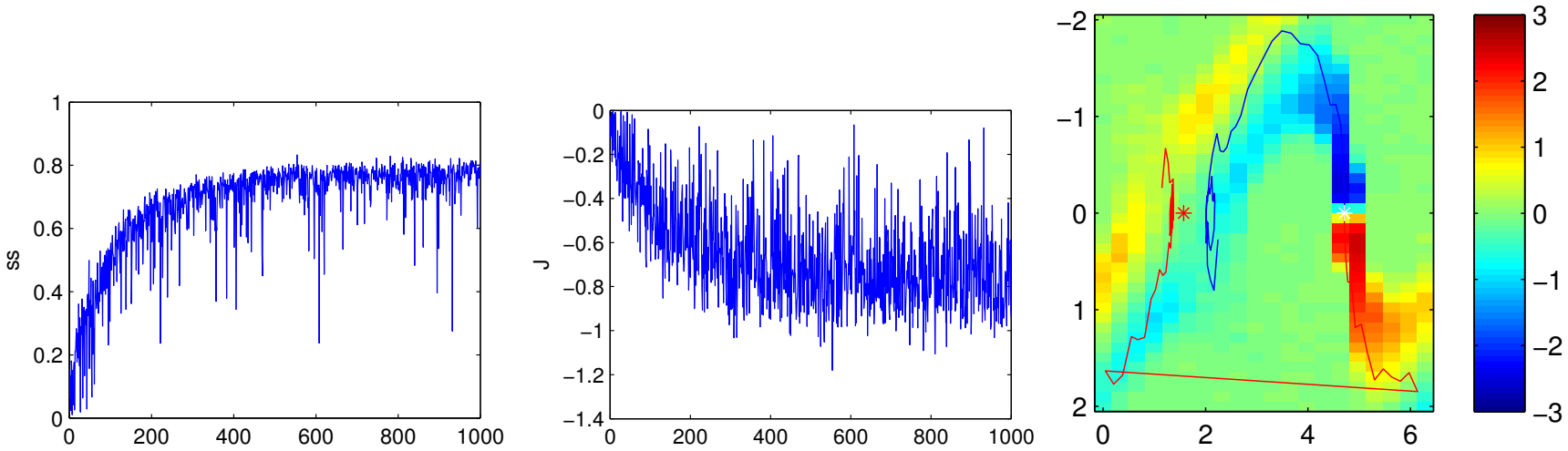


Inverted pendulum

Simple 2nd order pendulum with noise, $X = (\alpha, \dot{\alpha})$

$$\ddot{\alpha} = -\cos\alpha + u \qquad C = \mathbb{E}\int_0^T dt V(X_t) + \frac{1}{2}u(X_t, t)^2$$

Naive grid: $u(x) = \sum_k u_k \delta_{x,x_k}$.



$ESS < 1$ due to time discretization, finite sample size effects and $u(x,t) = u(x)$.

Acrobot

Swing up and stabilize underactuated stochastic double pendulum.

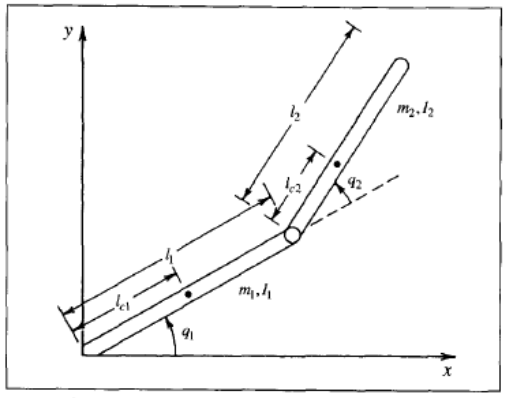


Fig. 1. The Acrobot.

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + h_1 + \phi_1 = 0 \quad (1)$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + h_2 + \phi_2 = \tau, \quad (2)$$

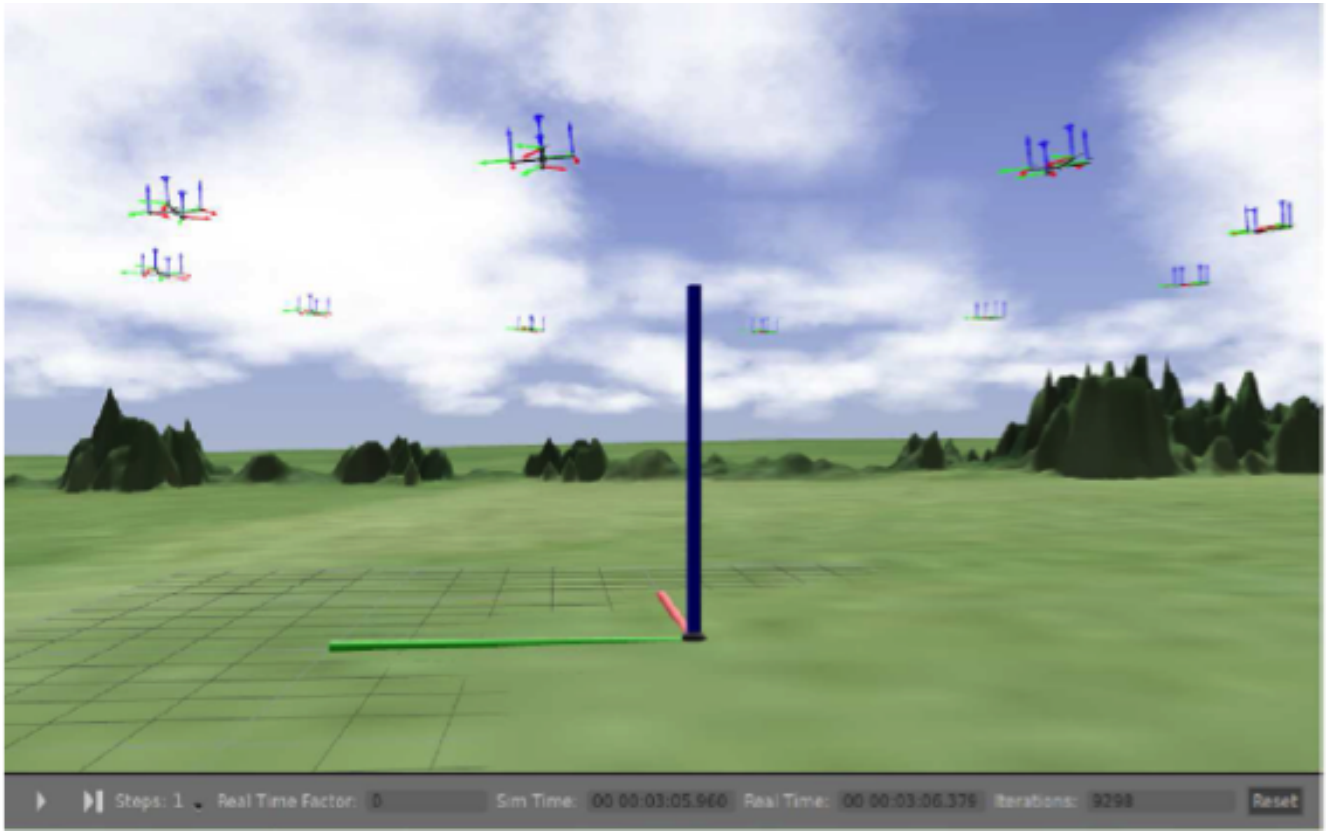
where

$$d_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos(q_2)) + I_1 + I_2$$
$$d_{22} = m_2 l_{c2}^2 + I_2$$
$$d_{12} = m_2(l_{c2}^2 + l_1 l_{c2}\cos(q_2)) + I_2$$
$$d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos(q_2)) + I_2$$
$$h_1 = -m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2^2 - 2m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2\dot{q}_1$$
$$h_2 = m_2 l_1 l_{c2}\sin(q_2)\dot{q}_1^2$$
$$\phi_1 = (m_1 l_{c1} + m_2 l_1)g\cos(q_1) + m_2 l_{c2} g\cos(q_1 + q_2)$$
$$\phi_2 = m_2 l_{c2} g\cos(q_1 + q_2).$$

(acrobot_dominik2016a.mp4)

Neural network 2 hidden layers, 50 neurons per layer. Input is position and velocity. 2000 iterations, with 30000 rollouts per iteration. 100 cores. 15 minutes

Coordination of UAVs



Centralized path integral solution computed in real time (simulation) for 10 quadrotors. Objective is to fly a holding pattern near a fixed location maintaining a minimal velocity and distance to other drones. Video at: `http://www.snn.ru.nl/~bertk/control_theory/PI_quadrotors.mp4`

[4]

Coordination of UAVs

This behavior was replicated on real quadrotors demonstrating high dimensional non-linear stochastic optimal control in real-time.

Chao Xu ACC 2017

‘

Agressive driving

https://www.youtube.com/watch?v=lD_6CLoa4rY

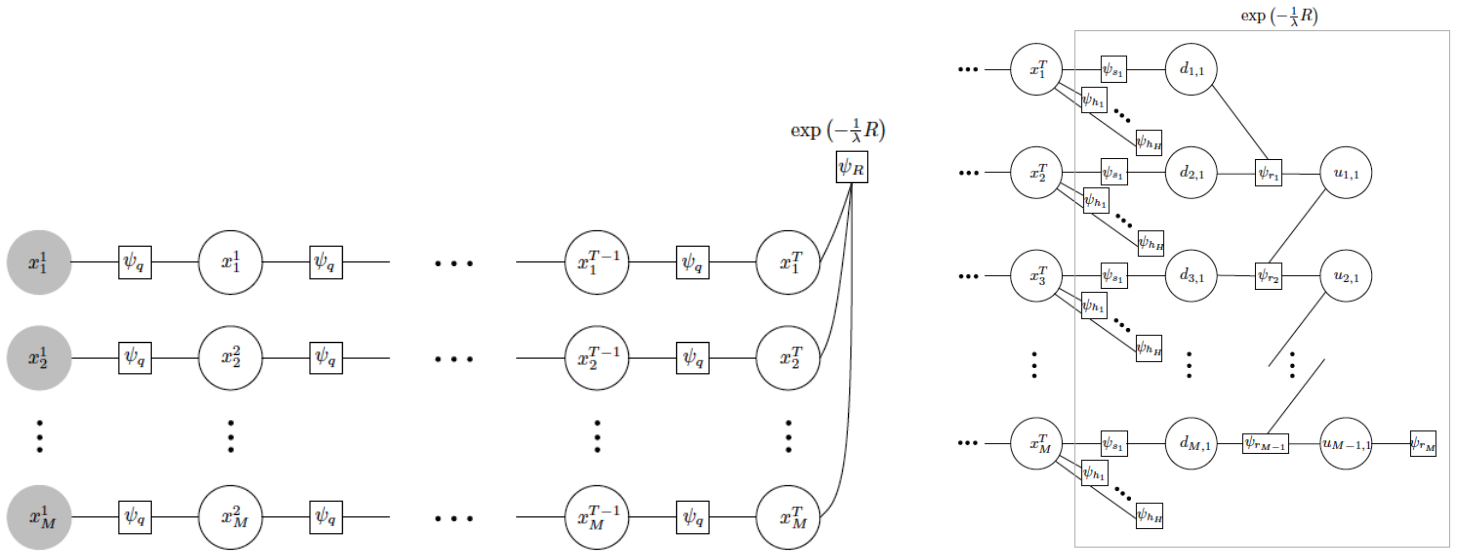
[5]

Multi Agent cooperative game

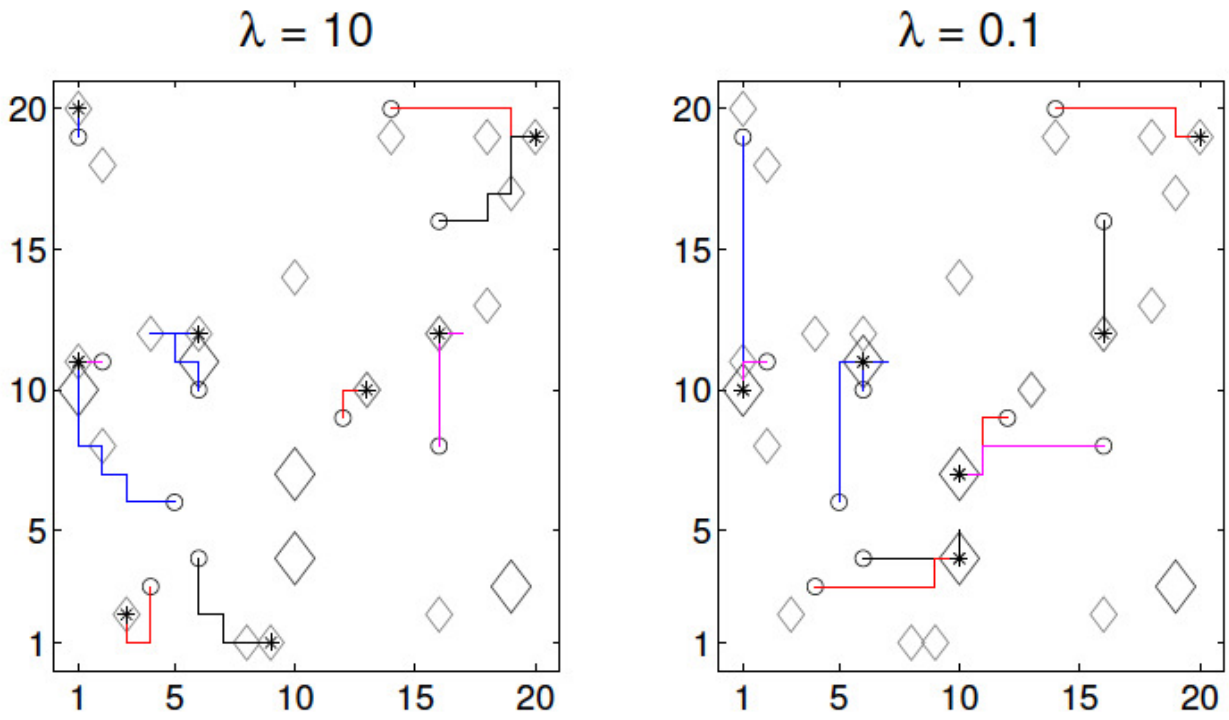
Model of cooperation: either hunt a hare alone or a stag together.

	Stag	Hare
Stag	$\mathbf{3}, \mathbf{3}$	$0, 1$
Hare	$1, 0$	$\mathbf{1}, \mathbf{1}$

We define the KL-stag-hunt game as a multi-agent version where agents move on a grid to hunt stag or hare.



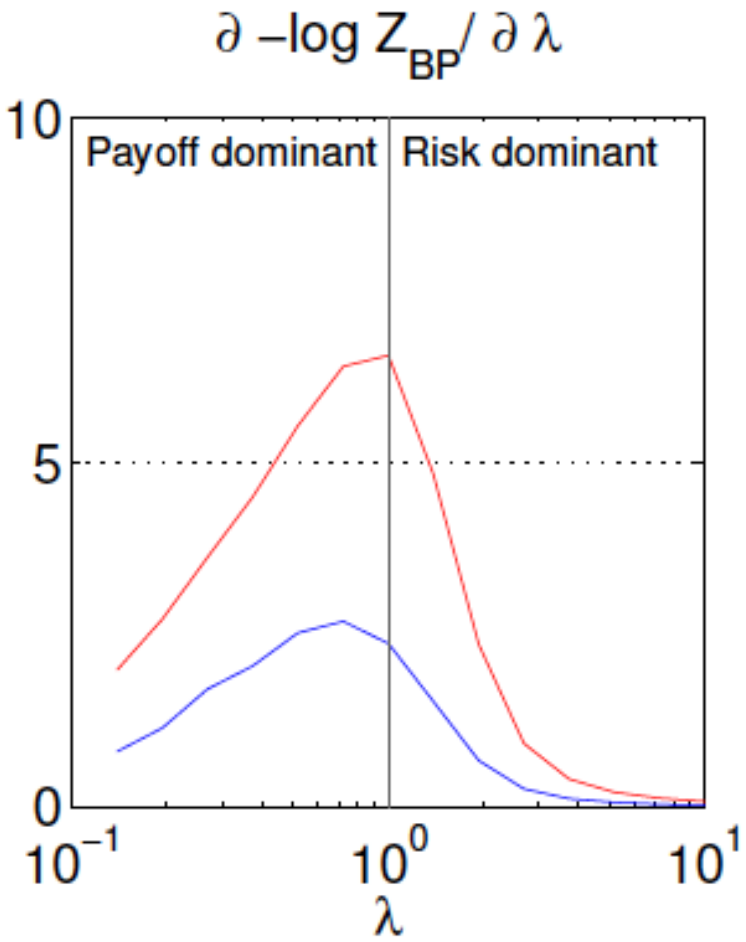
Approximate inference of the KL-stag-hunt problem

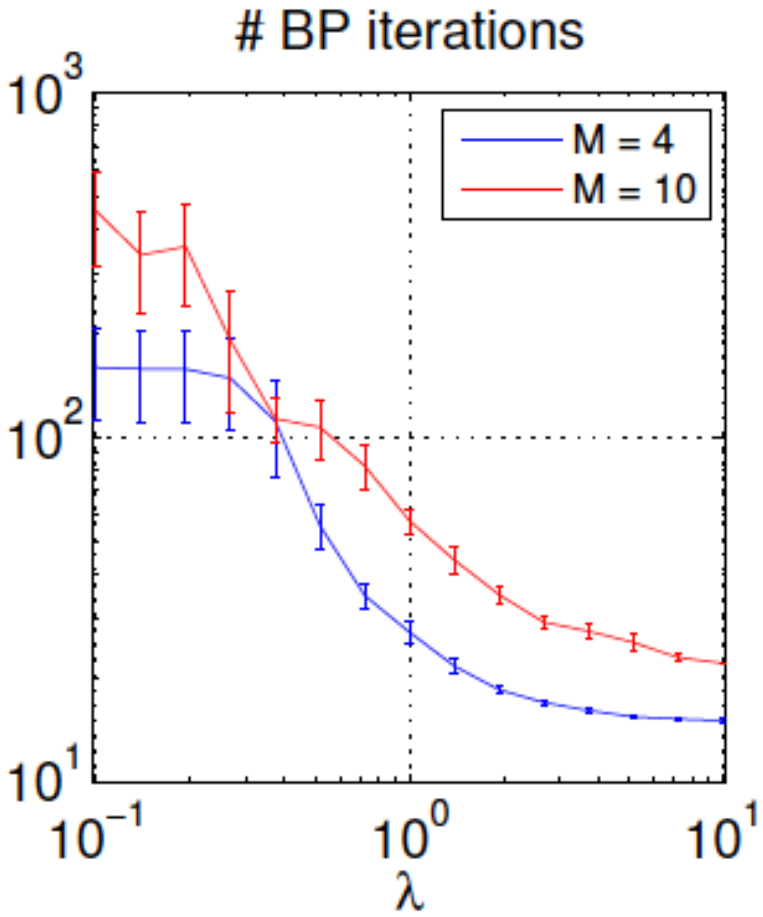


$M = 10$ agents, $N = 400$ locations, 10^{26} states per time slice

Sequential BP. If converges, converges in less than 500 iterations. Trajectories are marginal beliefs.

Phase transition (?)





References

[1] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence research*, 4:237–285, 1996.

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

[3] H.J. Kappen and H.C. Ruiz. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, pages 10.1007/s10955–016–1446–7, 2016.

[4] Vicenç Gómez, Sep Thijssen, Andrew Symington, Stephen Hailes, and Hilbert Kappen. Real-time stochastic optimal control for multi-agent quadrotor systems. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 26, pages 468–476, 2016.

[5] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Aggressive driving with model predictive path integral control. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1433–1440. IEEE, 2016.

[6] H.J. Kappen. Linear theory for control of non-linear stochastic systems. *Physical Review letters*, 95:200201, 2005.

[7] S. Thijssen and H.J. Kappen. Consistent adaptive multiple importance sampling and controlled diffusions. arXiv:1803.07966, 2018.

[8] Aarón Villanueva and Hilbert J Kappen. Stochastic optimal control of open quantum systems. arxiv.org/abs/2410.18635, 2024.

[9] R. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[10] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[11] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, Belmont, Massachusetts, 1996.

[12] A.G. Barto, R. S. Sutton, and C.W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):835–846, 1983.

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

[14] David Silver, Richard S Sutton, and Martin Müller. Temporal-difference search in computer go. *Machine learning*, 87(2):183–219, 2012.

[15] R.S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[16] J.J. Florentin. Optimal, probing, adaptive control of a simple Bayesian system. *International Journal of Electronics*, 13:165–177, 1962.

[17] P. R. Kumar. Optimal adaptive control of linear-quadratic-gaussian systems. *SIAM Journal on Control and Optimization*, 21(2):163–178, 1983.

[18] E.J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

Other topics

(- RL)
- Relation RL and PI control and exploration
- Variational approximation, n joint arm (Kappen tutorial 2011)
- Coordination of continuous agents using MF and BP (Wiegerinck et al. 2006, van den Broek et al. 2006)
- Risk sensitive path integral control (van den Broek 2010)
- Inference and control (Kappen tutorial 2011)
- Control of quantum systems

Reinforcement learning

Reinforcement learning [1]

”Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment.”

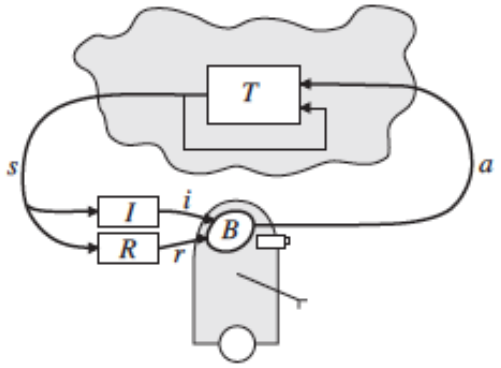


Figure 1: The standard reinforcement-learning model.

Agent’s action a changes the state of the world.
The state of the world s is observed through sensing i and a reinforcement signal (reward) signal r
Behaviour (actions) $a(s)$ or $a(i)$ should be such as to increase the long-run sum of rewards r.

Formally:
- discrete set of environment states $\mathcal{S}$
- discrete set of agent actions $\mathcal{A}$
- set of scalar reinforcement signals, (0,1) or real

Find optimal policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$

Environment:	You are in state 65. You have 4 possible actions.
Agent:	I'll take action 2.
Environment:	You received a reinforcement of 7 units. You are now in state 15. You have 2 possible actions.
Agent:	I'll take action 1.
Environment:	You received a reinforcement of -4 units. You are now in state 65. You have 4 possible actions.
Agent:	I'll take action 2.
Environment:	You received a reinforcement of 5 units. You are now in state 44. You have 5 possible actions.
⋮	⋮

Environment is
- *non-deterministic*: taking same action in same state may yield different next state
- *stationary*: the probability of the new state does not depend on time explicitly

This can be modelled as a Markov process $p(s'|s,a)$ (called Markov decision process).

Models of optimallity

The *finite horizon model*:

$$R = \sum_{t=0}^{h} r_t$$

Current time is $t = 0$. Does not care what happens after $t = h$.

Two uses:
- Fixed horizon: Take *h-step optimal action*, (h-1)-step optimal action,. . ., 1-step optimal action
- Receding horizon: Take always h-step optimal action

Models of optimallity

The *infinite horizon discounted model*:

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \qquad 0 \leq \gamma < 1$$

γ is "probability to live another step", and γ^t to live t more steps.
It is also a good mathematical trick to bound infinite sum.

Models of optimallity

The *average reward model*:

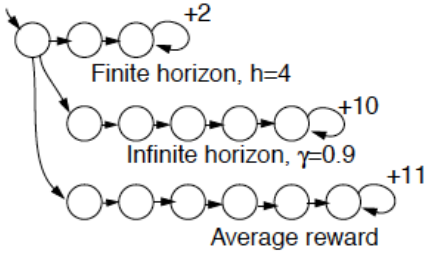
$$R = \lim_{h \to \infty} \frac{1}{h} \sum_{t=0}^{h} r_t$$

Is identical to the discounted reward model for $\gamma \to 1$.

Problem with this model is that there is no way to distinguish between two policies,

1. large initial rewards, followed by some policy π
2. small initial rewards, followed by the same policy π

Models of optimallity



Only single action from start state at $t = 0$ (upper left circle) with three choices.

Different criteria yield different optimal solutions:
Finite horizon $h = 5$ model yields for first choice: $R = \sum_{t=0}^{5} r_t = 0 + 0 + 2 + 2 + 2 = 6$ and zero for the other choices.
Discounted reward $\gamma = 0.9$ model yields expected rewards

$$R = \sum_{t=0}^{\infty} \gamma^t r_t = \left(2 \sum_{t=2}^{\infty} \gamma^t, 10 \sum_{t=5}^{\infty} \gamma^t, 11 \sum_{t=6}^{\infty} \gamma^t \right) = \left(2\gamma^2, 10\gamma^5, 11\gamma^6 \right) \frac{1}{1-\gamma} = (16.2, 59.0, 58.5)$$

Selects second choice.

Average reward model yields expected rewards: $R = \lim_{h \to \infty} \frac{1}{h} \sum_{t=0}^{h} r_t = (2, 10, 11)$. Selects third choice.

For the discounted reward case we used $\sum_{t=0}^{\infty} \gamma^t = \frac{1}{1-\gamma}$.

Proof: Define $S = \sum_{t=0}^{T} \gamma^t$. Then

$$\begin{aligned}(1-\gamma)S &= \sum_{t=0}^{T} \gamma^t - \sum_{t=0}^{T} \gamma^{t+1} = \sum_{t=0}^{T} \gamma^t - \sum_{t=1}^{T+1} \gamma^t = 1 - \gamma^{T+1} \\ \sum_{t=0}^{\infty} \gamma^t &= \lim_{T\to\infty} S = \lim_{T\to\infty} \frac{1-\gamma^{T+1}}{1-\gamma} = \frac{1}{1-\gamma}\end{aligned}$$

And

$$\sum_{t=2}^{\infty} \gamma^t = \gamma^2 \sum_{t=2}^{\infty} \gamma^{t-2} = \gamma^2 \sum_{t'=0}^{\infty} \gamma^{t'} = \frac{\gamma^2}{1-\gamma}$$

with $t' = t - 2$. And similar for $\sum_{t=5}^{\infty} \gamma^t$ and $\sum_{t=10}^{\infty} \gamma^t$.

Models of optimallity

γ defines an effective horizon time τ

$$\gamma^t = e^{-\frac{t}{\tau}} \qquad \tau = \frac{-1}{\log \gamma}$$

When $\gamma = 1 - \epsilon$ with ϵ small, we get $\tau \approx \frac{1}{\epsilon}$. [10]

Small h, τ learns policies that optimize for short term rewards. Large h, τ learns policies that optimizes for long term rewards.

For instance in the discounted reward case:

$$\begin{aligned} R &= \sum_{t=0}^{\infty} \gamma^t r_t = \left(2\gamma^2, 10\gamma^5, 11\gamma^6\right) \frac{1}{1-\gamma} \\ \gamma &= 0.2: \quad (0.1, 0.004, 0.0009) \\ \gamma &= 0.9: \quad (16.2, 59.0, 58.5) \end{aligned}$$

[10] For $\gamma = (0.5, 0.9, 0.999)$ we get $\tau = (1.4, 9.4, 999.5) \approx (2, 10, 1000)$.

Markov Decision Processes

A set of states $\mathcal{S}$, set of actions $\mathcal{A}$, reward function $R : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$.

A state transition function $T : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$, with $\Pi(\mathcal{S})$ is set of probability distributions over $\mathcal{S}$. We denote $T(s'|s, a)$.

The model is *first order Markov* because the distribution over next states s' only depend on current state and action s, a and no previous history.

We define $\pi : \mathcal{S} \to \mathcal{A}$ as a policy. Suppose

$$s_0 \to_\pi a_0 \to_T s_1 \to_\pi a_1 \to_T \dots$$

using policy π. Define the optimal value of a state as

$$V^*(s) = \max_\pi \left\langle \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right\rangle_{s_0 = s}$$

For the infinite-horizon discounted model, there exists an optimal deterministic stationary policy ([9]).

Markov Decision Processes

One can show that the optimal cost-to-go, aka the value function $V^*(s)$ satisfies the Bellman equation

$$V^*(s_0) = \max_{a_0} \left(R(s_0, a_0) + \gamma \sum_{s_1} T(s_1|s_0, a_0) V^*(s_1) \right) \tag{7}$$

The optimal policy is

$$\pi^*(s_0) = \text{argmax}_{a_0} \left(R(s_0, a_0) + \gamma \sum_{s_1} T(s_1|s_0, a_0) V^*(s_1) \right)$$

Derivation

First write

$$\begin{aligned} V^*(s_0) &= \max_{\pi} \sum_{t=0}^{\infty} \gamma^t \langle R(s_t, a_t) \rangle_{s_0} \\ &= \max_{a_0} \left(R(s_0, a_0) + \max_{a_1, a_2, \ldots} \sum_{t=1}^{\infty} \gamma^t \langle R(s_t, a_t) \rangle_{s_0} \right) \end{aligned}$$

with $\max_\pi = \max_{a_0, a_1, \ldots}$.

The expectation $\langle \ldots \rangle_{s_0}$ depends on the current state s_0 and the sequence of actions $a_0, a_1, \ldots, a_{t-1}$. Thus,

$$\begin{aligned} \langle R(s_t, a_t) \rangle_{s_0} &= \sum_{s_t} T(s_t | s_0, a_{0:t-1}) R(s_t, a_t) = \sum_{s_1} T(s_1 | s_0, a_0) \sum_{s_t} T(s_t | s_1, a_{1:t-1}) R(s_t, a_t) \\ &= \sum_{s_1} T(s_1 | s_0, a_0) \langle R(s_t, a_t) \rangle_{s_1} \end{aligned}$$

with $T(s_t | s_0, a_{0:t-1})$ the probability to transit from state s_0 to state s_t when actions $a_0, \ldots, a_{t-1}$ are taken.

$$
\begin{aligned}
\max_{a_1,a_2\ldots} \sum_{t=1}^{\infty} \gamma^t \langle R(s_t,a_t)\rangle_{s_0} &= \sum_{s_1} T(s_1|s_0,a_0) \max_{a_1,a_2\ldots} \sum_{t=1}^{\infty} \gamma^t \langle R(s_t,a_t)\rangle_{s_1} \\
&= \gamma \sum_{s_1} T(s_1|s_0,a_0) \max_{a_1,a_2\ldots} \sum_{t=1}^{\infty} \gamma^{t-1} \langle R(s_t,a_t)\rangle_{s_1} \\
&= \gamma \sum_{s_1} T(s_1|s_0,a_0) \max_{a_1,a_2\ldots} \sum_{t=0}^{\infty} \gamma^{t} \langle R(s_{t+1},a_{t+1})\rangle_{s_1} \\
&= \gamma \sum_{s_1} T(s_1|s_0,a_0) V^*(s_1)
\end{aligned}
$$

Putting everything together, we get Eq. 7.

Value iteration

```
initialize V(s) arbitrarily
loop until policy good enough
    loop for s ∈ S
        loop for a ∈ A
            Q(s,a) := R(s,a) + γ Σ_{s'∈S} T(s,a,s') V(s')
        V(s) := max_a Q(s,a)
    end loop
end loop
```

Value iteration converges to V^* ([9])

Stopping criterion (Williams & Baird 1993):
if $\max_s |V_t(s) - V_{t-1}(s)| = \epsilon$ then $\max_s |\pi_t(s) - \pi^*(s)| \leq 2\epsilon\gamma/(1-\gamma)$

Computational complexity is $O(|\mathcal{S}|^2|\mathcal{A}|)$, or $O(|\mathcal{S}||\mathcal{A}|)$ when constant number of next states per state (sparse T).

iterations polynomial in $1/(1-\gamma)$.

Policy iteration

Manipulates the policy directly, rather than indirectly through the value function:

```
choose an arbitrary policy π'
loop
    π := π'
    compute the value function of policy π:
        solve the linear equations
            V_π(s) = R(s, π(s)) + γ Σ_{s'∈S} T(s, π(s), s') V_π(s')
    improve the policy at each state:
        π'(s) := arg max_a (R(s, a) + γ Σ_{s'∈S} T(s, a, s') V_π(s'))
until π = π'
```

V_π is the value of policy π. The policy update is greedy with respect to V_π.

Model-free or model based approach

The previous formulas assume that the environment ($T(s'|s,a)$ and $R(s,a)$) are known in advance.

The RL research is mostly concerned with the situation that the environment is not known.

Model free: Learn a controller without learning a model.
- fast, works well for simple tasks,
- No transfer to other tasks

Model based: first learn a model, and then use it to derive a controller
- slow, but works for more complex tasks
- transfer to other tasks in the same environment

Hybrid: Learn a model and a controller simultaneously

Exploration

When the model is not known, learning the model requires in principle to visit (physically!) all states. This holds for both model based and model free approaches.

Visiting all states is not feasible because
- there are too many states
- it takes too much time, in particular for real robots
- it may be dangerous!

One thus needs some exploration approach, that visits only the subset of 'interesting' high reward states. Given that the environment is unknown, this is fundamentally impossible.

Most exploration strategies are simply to try random moves. Obviously, this can be inefficient.

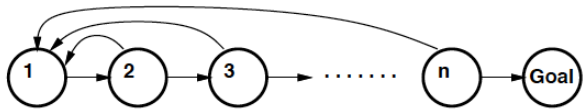


Figure 5: In this environment, due to Whitehead (1991), random exploration would take take $O(2^n)$ steps to reach the goal even once, whereas a more intelligent exploration strategy (e.g. "assume any untried action leads directly to goal") would require only $O(n^2)$ steps.

The most naive approach

To understand the complexity of learning the optimal policy, consider the simplest model free method

1. choose an initial policy π
2. estimate $V_\pi(s)$ for given policy π
   (a) For each states s, run N sample trajectories of length h:

$$V_\pi(s) \quad = \quad \left\langle \sum_{t=1}^{\infty} \gamma^t r_t \right\rangle \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{h} \gamma^t r_t^i$$

3. Repeat for all policies π or use some smart search strategy over policies.

The problem is hard because there are many states and there are many policies.

Model free policy iteration TD(0)

A *stochastic approximation* for policy iteration is obtained as follows:

Consider *experience tuple* (s, a, r, s') under policy π.

$$V(s) := V(s) + \alpha_t(r + \gamma V(s') - V(s))$$

This stochastic rule is known as temporal difference learning TD(0). On average, TD(0) is equal to policy iteration [11]

When $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$ and all states are visited sufficiently often, this algorithm converges to the solution $V_\pi(s)$ of policy iteration [10].

A possible choice is $\alpha_t = 1/t$.

[11] Identify $r = R(s, \pi(s))$. Multiply both sides by $T(s'|s, \pi(s))$ and sum over s' gives

$$0 = \left(R(s, \pi(s)) + \gamma \sum_{s'} T(s'|s, \pi(s) V(s') - V(s) \right)$$

which is the policy iteration update equation.

TD(λ)

TD(0) converges but makes poor use of the data: only the immediate previous state is updated.

TD(λ) updates every state according to discount $0 \leq \lambda \leq 1$:

$$d_t \quad = \quad r_t + \gamma V(s_{t+1}) - V(s_t)$$

When $s_1 \rightarrow s_2$:

$$V(s_1) \quad := \quad V(s_1) + \alpha_{m_{s_1}} d_1$$

When $s_2 \rightarrow s_3$:

$$\begin{aligned} V(s_1) \quad &:= \quad V(s_1) + \alpha_{m_{s_1}} \lambda d_2 \\ V(s_2) \quad &:= \quad V(s_2) + \alpha_{m_{s_2}} d_2 \end{aligned}$$

m_s is the number of times state s has been visited.

TD(λ)

In general at iteration t:[12]

$$\begin{aligned} d_t &= r_t + \gamma V(s_{t+1}) - V(s_t) \\ \epsilon(s) &= \sum_{k=1}^{t} \lambda^{t-k} \delta_{s,s_k} \qquad \forall s \\ V(s) &:= V(s) + \alpha_{m_s} d_t \epsilon(s) \qquad \forall s \end{aligned}$$

Note, that *all* past states are updated, not only the current state, proportional to their eligibility $\epsilon(s)$ that decays over time

t	state	$\epsilon(s)$
1	1	$(\lambda^0, 0, 0)$
2	2	$(\lambda^1, \lambda^0, 0)$
3	3	$(\lambda^2, \lambda^1, \lambda^0)$
4	1	$(\lambda^3 + \lambda^0, \lambda^2, \lambda^1)$

$TD(\lambda)$ converges under similar conditions as $TD(0)$ [11].

[12] NB Error in Kaelbling formula

Adaptive Heuristic Critic [12]

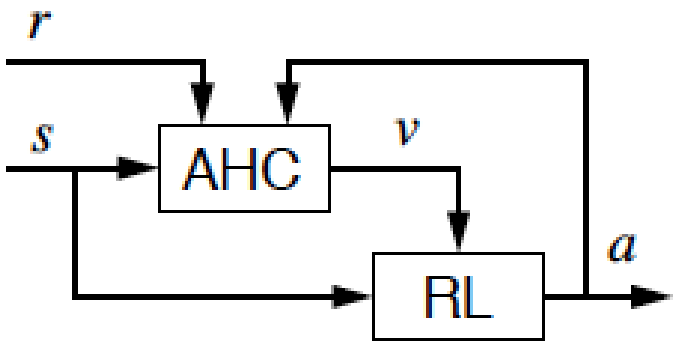


Figure 4: Architecture for the adaptive heuristic critic.

AHC is adaptive version of policy iteration
- Critic: compute estimate of V_π for policy π using stochastic policy iteration
- Actor: optimise π' based on (the current best estimate of) V_π.

NB: Only version with full convergence of 'inner loop' critic for fixed policy can be guaranteed to converge to optimal policy.

Q learning [13]

Denote $Q(s,a)$ the optimal expected value of state s when taking action a and then proceeding optimally. That is

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} T(s'|s,a) \max_{a'} Q(s',a')$$

and $V^*(s) = \max_a Q(s,a)$.

Using stochastic approximation, we obtain
- Generate s' from environment $T(s'|s,a)$
- Update

$$Q(s,a) = Q(s,a) + \alpha_t(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

- Generate a' either random or $\operatorname{argmax} Q(s',a')$.

Q learning converges when states are visited infinitely often and Robbins-Munro criteria.

Dyna

Idea: combine model based and model free.

Dyna operates in a loop of interaction with the environment. Given an experience tuple $\langle s, a, s', r \rangle$, it behaves as follows:

- Update the model, incrementing statistics for the transition from s to s' on action a and for receiving reward r for taking action a in state s. The updated models are $\hat{T}$ and $\hat{R}$.
- Update the policy at state s based on the newly updated model using the rule

$$Q(s,a) := \hat{R}(s,a) + \gamma \sum_{s'} \hat{T}(s,a,s') \max_{a'} Q(s',a') \quad ,$$

  which is a version of the value-iteration update for Q values.
- Perform k additional updates: choose k state-action pairs at random and update them according to the same rule as before:

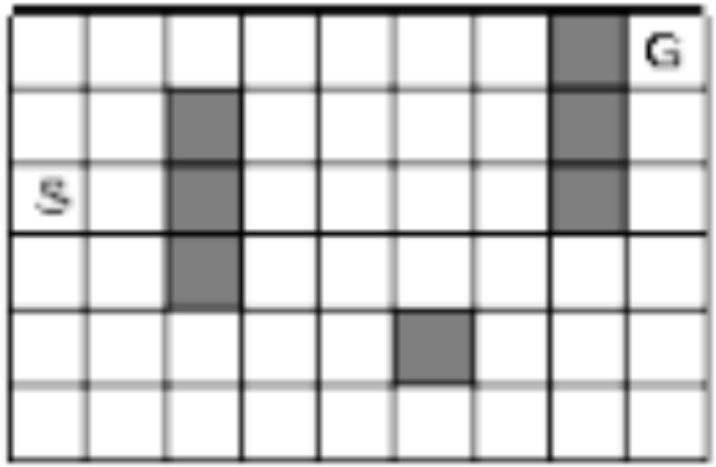
$$Q(s_k,a_k) := \hat{R}(s_k,a_k) + \gamma \sum_{s'} \hat{T}(s_k,a_k,s') \max_{a'} Q(s',a') \quad .$$

- Choose an action a' to perform in state s', based on the Q values but perhaps modified by an exploration strategy.

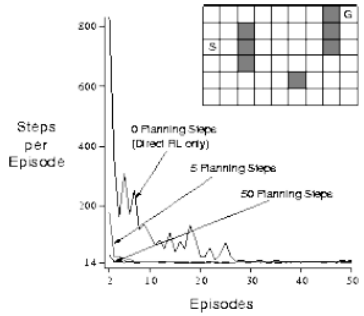
Sutton 1990

Dyna example

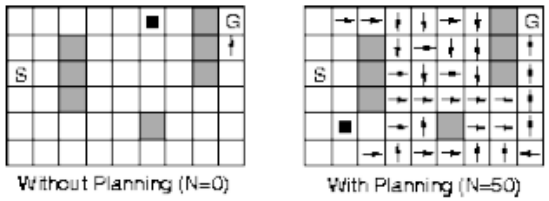
Maze:
- In each of the 46 states there are 4 actions (N,E,S,W) which take the agent to the corresponding state. When movement is blocked by obstacle, no movement results.
- reward is zero for all states and transitions except into the goal state G.
- after reaching the goal state the episode ends. agent returns to start state S.
- $\gamma = 0.95$ (discount rate), $\alpha = 0.1$ (learning rate), $\epsilon = 0.1$ (epsion-greedy exploration rate).



Dyna example

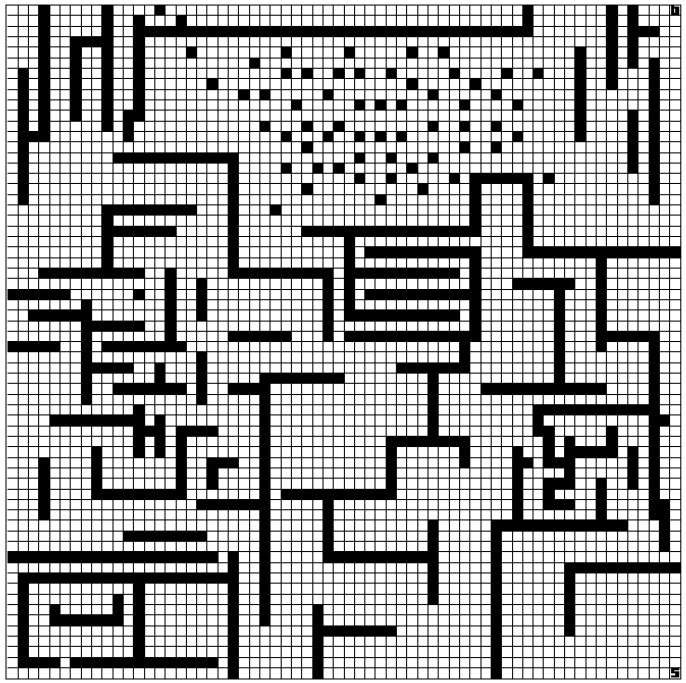


Number of steps to reach the goal versus learning episodes (average over 30 runs). First episode requires 1700 steps.



Policies found by non-planning $k = 0$ (denoted by $N = 0$ in the figure caption) and planning ($k = 50$) Dyna halfway through the second episode. $k = 0$ solution(=normal Q-learning) has only updated policy for next-to-goal state. $k = 50$ Dyna has learned environment model from first episode which is used to learn policies for all states.

Dyna larger example



	Steps before convergence	Backups before convergence
Q-learning	531,000	531,000
Dyna	62,000	3,055,000
prioritized sweeping	28,000	1,010,000

Table 1: The performance of three algorithms described in the text. All methods used the exploration heuristic of "optimism in the face of uncertainty": any state not previously visited was assumed by default to be a goal state. Q-learning used its optimal learning rate parameter for a deterministic maze: $\alpha = 1$. Dyna and prioritized sweeping were permitted to take $k = 200$ backups per transition. For prioritized sweeping, the priority queue often emptied before all backups were used.

3277 states shortest path problem formulated as discounted RL problem. Goal state (upper right corner) has reward 1, all other states have reward 0. Start state is lower right corner. Dyna (and prioritised sweeping) used $N = 200$ backups per transition.

Generalizations

A shortcoming of the Dyna method is that the planning steps are done at random.
- improvement can be made by *prioritized sweeping* (Moore & Atkenson 1993) by updating the states with highest priority.

Combining Dyna with Monte-Carlo tree search yields state-of-the-art performance on 9×9 computer Go [14]

A comparison between RL and PI control

Reinforcement learning

We consider a stochastic dynamics given by a first order Markov process, that assigns a probability to the transition of x to x' under action u: $p_0(x'|x,u)$. We assume that x and u are discrete, as is usually done.

We introduce a reward that depends on our current state x , our current action u and the next state x': $R(x,u,x')$. The expected reward when we take action u in state x is given as

$$R(x,u) = \sum_{x'} p_0(x'|x,u) R(x,u,x')$$

We define a *policy* $\pi(u|x)$ as the conditional probability to take action u given that we are in state x. Given the policy π and given that we start in state x_t at time t, the probability to be in state x_s at time $s > t$ is given by

$$\begin{aligned} p_\pi(x_s; s|x_t; t) \quad = \quad & \sum_{u_{s-1}, x_{s-1}, \ldots, u_{t+1}, x_{t+1}, u_t} p_0(x_s|x_{s-1}, u_{s-1}) \ldots \\ & \ldots \pi(u_{t+1}|x_{t+1}) p_0(x_{t+1}|x_t, u_t) \pi(u_t|x_t). \end{aligned}$$

p_π is a stationary Markov process i.e. $p_\pi(x';t+s|x;t)$ is independent of t, we can write

$$p_\pi(x';t+s|x;t) = p_\pi(x'|x;s-t)$$

The *expected future discounted reward* in state x is defined as:

$$J_\pi(x) = \sum_{s=0}^{\infty} \sum_{x',u'} \pi(u'|x') p_\pi(x'|x;s) R(x',u') \gamma^s \tag{8}$$

with $0 < \gamma < 1$ the discount factor.

We can write a recursive relation for J_π [13]

$$J_\pi(x) = \sum_u \pi(u|x) A_\pi(x,u) \qquad A_\pi(x,u) = \sum_{x'} p_0(x'|x,u)[R(x,u,x') + \gamma J_\pi(x')$$

Given J_π for the current policy π we construct a new deterministic policy

$$\pi'(u|x) \quad = \quad \delta_{u,u(x)}, \qquad u(x) = \arg\max_u A_\pi(x,u) \tag{9}$$

It can be shown (see [15]) that the solution for $J_{\pi'}$ is as least as good as the solution J_π in the sense that

$$J_{\pi'}(x) \geq J_\pi(x), \forall x$$

[13]

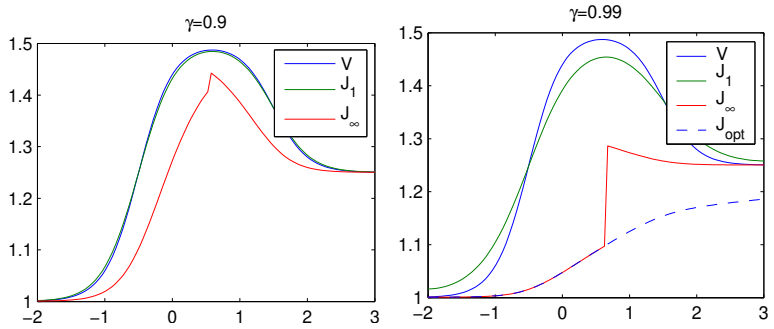
$$\begin{aligned}
J_\pi(x) \quad &= \quad \sum_u \pi(u|x) R(x,u) + \sum_{s=1}^{\infty} \sum_{x',u'} \pi(u'|x') p_\pi(x'|x;s) R(x',u') \gamma^s \\
&= \quad \sum_u \pi(u|x) R(x,u) + \gamma \sum_{s=1}^{\infty} \sum_{x',u'} \sum_{x'',u''} \\
&\quad \pi(u'|x') p_\pi(x'|x'';s-1) p_0(x''|x,u'') \pi(u''|x) R(x',u') \gamma^{s-1} \\
&= \quad \sum_{u,x'} \pi(u|x) p_0(x'|x,u)[R(x,u,x') + \gamma J_\pi(x')] = \sum_u \pi(u|x) A_\pi(x,u)
\end{aligned}$$

Policy iteration:

$$\pi^0 \rightarrow J_{\pi 0} \rightarrow \pi^1 \rightarrow J_{\pi 1} \rightarrow \pi^2 \ldots$$

Policy iteration converges to a stationary value function $J^*(x)$ that is a fixed point of the above procedure, but maybe a local optimum.

Receding horizon problem



14

For $\gamma = 0.9$ the optimal policy is to 'stay put'.

For $\gamma = 0.99$, policy iteration develops a local minimum. The value of the policy 'always move left' is better. Thus, for $\gamma = 0.99$ the optimal policy is to 'move left'.

The local minima problem of policy iteration persists at larger γ but is resolved when using Q-learning ($\gamma = 0.9, 0.99$ and 0.999). [15]

[14] The state space is discretized in 100 bins with $-2 < x < 3$. The actions are $u = \pm dx$. The dynamics is deterministic: $p_0(x'|x,u) = \delta_{x',x+u}$. The reward is given by $R(x,u,x') = -V(x')$. γ controls the effective horizon time $T \approx -1/\log\gamma$. We use policy iteration.

[15] The number of value iterations scales as $1/(1-\gamma)$ and thus can become quite large. The number of policy improvement steps in this simple example is only 1. Smoothing the policy updates ($\pi \leftarrow \alpha\pi + (1-\alpha)\pi_{\text{new}}$ increases the number of policy improvement steps, but does not change fixed points of the algorithm.

Path integral control

$$
\begin{aligned}
dx &= udt + d\xi \\
C(x,u) &= \langle S(\tau) \rangle \qquad S(\tau) = \int_t^{t+T} dt \frac{R}{2} u(t)^2 + Ru(t)d\xi(t) + V(x(t))
\end{aligned}
$$

with $\tau = x_{t:t+T}$ a trajectory

The optimal cost-to-go is given by

$$
J(x) = -\lambda \log \int dy \rho(y, t+T|x,t) = -\lambda \log \int d\tau e^{-S(\tau)/\lambda}
$$

with ρ the solution of the Fokker-Planck equation.

Note, that C, ρ and $J(x), u(x)$ do not depend on t.

We use the MC sampling method and the Laplace approximation to find approximate solutions.

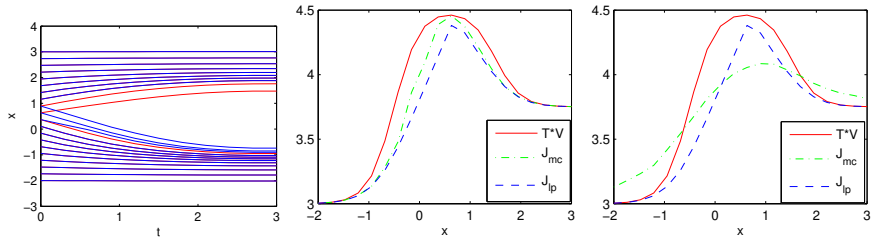


Figure 1: $T = 3$. Left: Laplace trajectories. Middle: $\nu = 0.01$. Right: idem for $\nu = 1$.

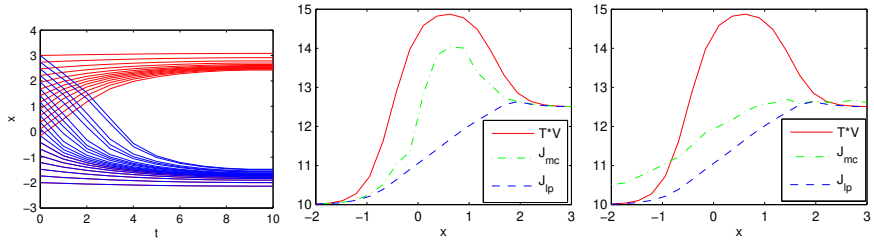


Figure 2: $T = 10$. Left: Laplace trajectories. Middle: $\nu = 0.01$. Right: idem for $\nu = 1$.

The Laplace approximation is accurate for low noise. The MC appoximation is accurate when $\sqrt{\nu T}$ is large compared to the exploration area. This is true for high noise and for low noise when the 'stay' policy happens to be optimal. The results of RL and PI control qualitatively agree.

Exploration

What to do in unknown environment?

Model-based: first learn the environment and then compute the optimal control.

Model-free: interleave exploration (learning the environment) and exploitation (behave optimally in this environment).

The model-free approach leads to the exploration-exploitation dilemma:
- The computed 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.
- The issue of optimal exploration is not addressable within the context of optimal control theory (or RL!).

An exception to this is when one has prior knowledge about the environment:
- when the costs is a smooth function of the state variables, one can extrapolate a learned cost model to unknown parts
- when the environment and cost are drawn from some known probability distribution (k-armed bandit).

Exploration

Thus optimal exploration is random.

Consider the previous 1 d problem. We sample one long ($Ndt \gg T = ndt$) trajectory $x_i, i = 1, \ldots, N$ with states as $x_{i+1} = x_i + d\xi_i$.

We estimate $\psi(x)$ for all x from this single trajectory:

$$\psi(x_i, 0) = \exp\left(-\frac{dt}{\lambda}\sum_{j=i+1}^{j=i+n} V(x_j)\right)$$

16

[16] 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:

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

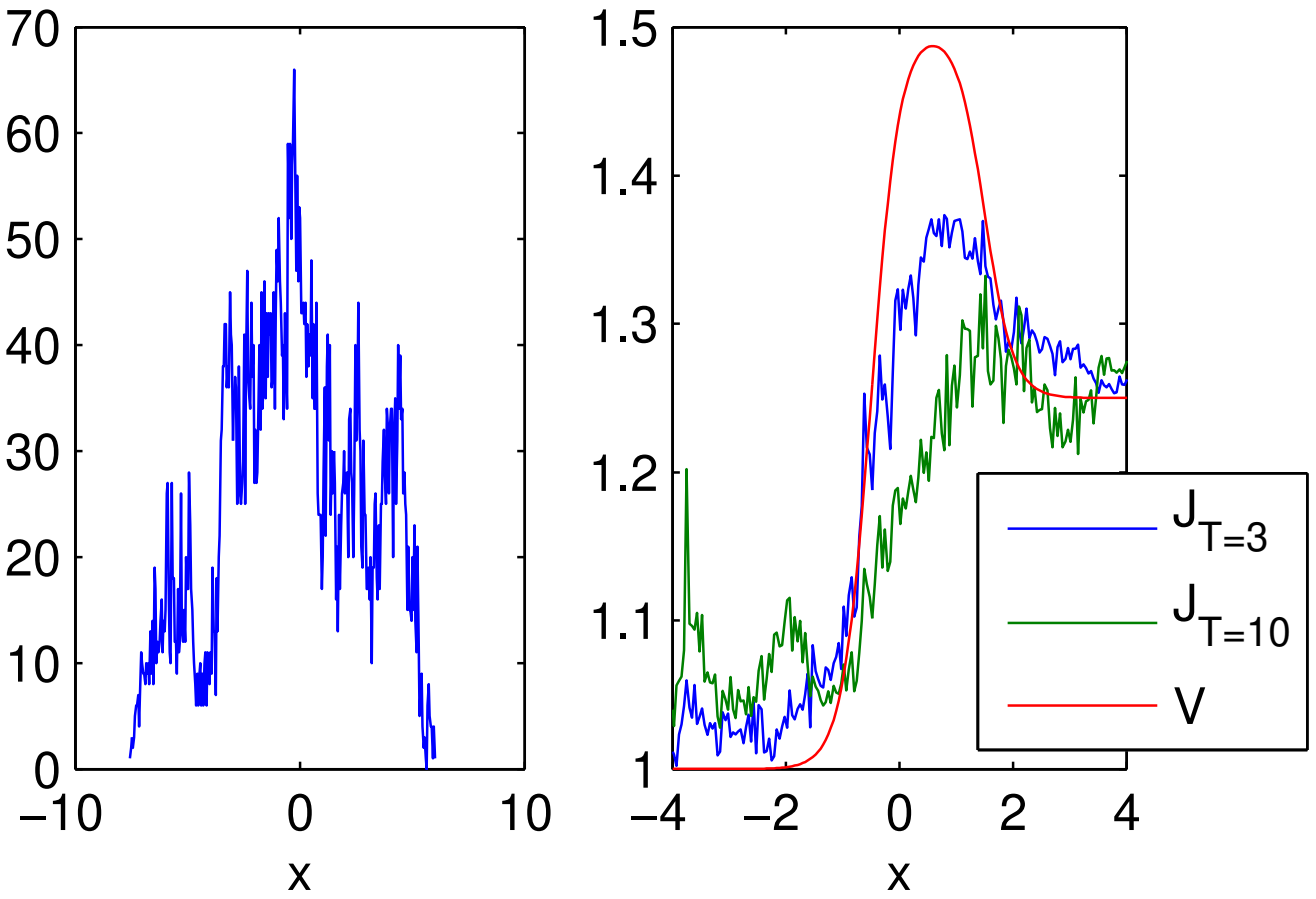


Figure 3: Sampling of $J(x)$ for all x and different T with one trajectory of $N = 8000$ iterations starting at $x = 0$. Left: Histogram of states visited (300 bins). Right: In each bin x, $\psi(x)$ and $J(x)$ are estimated for different T. Time discretization $dt = 0.02, \nu = 1, R = 1$.

A neural implementation

The brain represents the environment in terms of neural maps. These maps are topologically organized, in the sense that nearby neurons represent nearby locations in the environment. Examples of such maps are found in sensory areas as well as in motor areas. In the latter case, nearby neuron populations encode nearby motor acts.

Consider a one-dimensional environment that we encode with a one-dimensional array of neurons, $i = 1, \ldots, m$ and denote the firing rate of the neurons at time t by $\rho_i(t)$.

We assume that the animal has learned a model of the world (the neural map, the dynamics and the rewards). In addition to sensing, the neural map can then also be used for planning.

A neural implementation

We assume that the neural array implements a space-discretized version of the forward diffusion process as given by the Fokker-Planck equation:

$$\frac{d\rho_i}{dt} = -\frac{V_i}{\lambda}\rho_i(t) + \frac{\nu}{2}\sum_j D_{ij}\rho_j(t)$$

with D the diffusion matrix $D_{ii} = -2, D_{ii+1} = D_{ii-1} = 1$ and all other entries of D are zero. V_i is the cost at location x_i. Each neuron updates its firing rate on the basis of the activity of itself and its nearest neighbors. Further, we assume that there is some additional inhibitory lateral connectivity in the network such that the total firing rate in the map is normalized: $\sum_i \rho_i(t) = 1$.

By running the network dynamics from $t = 0$ to T in the absence of external stimuli, the animal can 'think ahead' and compute the best current action.

A neural implementation

For $T = 5$ the optimal action is to move to the right (the nearest local minimum of V). For $T = 10$, the peak around $x = 2$ disappears and a peak around $x = -1$ appears. The optimal action is to move to the left.

This is quite different from the reinforcement learning paradigm, where for each value of γ the Bellman equations should be solved.

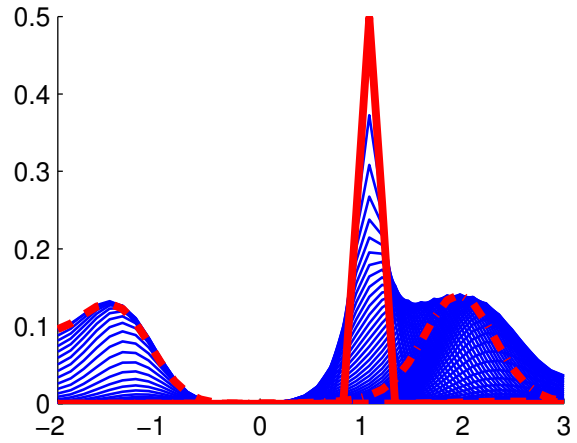


Figure 4: Thinking ahead. Blue lines show the time evolution of $\rho_i(t)$. In red, thick solid, dot-dashed and dashed lines at $t = 0.1, 5$ and $t = 10$, respectively.

17

[17] In this very simple example, the decision whether to move left or right can be inferred simply from the mode of $\rho(y, T|x, 0)$. In general this does not need to be true and in any case, for the correct estimation of the size of the optimal control, the gradient of $\psi(x) = \int dy \rho(y, T|x, 0)$ must be computed.

Comparing RL and PI control

- Time dependent versus time independent control problems
- Discrete state space versus continuous state space
- For receding horizon problems:
  - all $J(x)$ interdependent from Bellman equation (RL) versus $J(x)$ independent (PI)
  - No reusability to compute J for different γ (RL) versus some reusability for (PI) [18]
  - Exploration can be defined independent of exploration: random exploration allows for online estimation of value function (PI)

[18] For example, suppose that we know the solution for horizon times T: $\psi_T(x) = \int dy \rho_T(y|x)$. We can use this to compute a solution $\psi_{2T}(x) = \int dz \rho_{2T}(z|x) = \int dz dy \rho_T(z|y)\rho_T(y|x) = \int dy \psi_T(y)\rho_T(y|x)$.

The variational approximation

The variational method

Consider an arm consisting of n joints of length 1. The location of the ith joint in the 2d plane is

$$x_i = \sum_{j=1}^{i} \cos\theta_i \qquad y_i = \sum_{j=1}^{i} \sin\theta_i$$

with $i = 1, \ldots, n$. Each of the joint angles is controlled by a variable u_i. The dynamics of each joint is

$$d\theta_i = u_i dt + d\xi_i, \quad i = 1, \ldots, n$$

with $d\xi_i$ independent Gaussian noise with $\left\langle d\xi_i^2 \right\rangle = \nu dt$. Denote by $\vec{\theta}$ the vector of joint angles, and $\vec{u}$ the vector of controls.

The variational method

The expected cost for the control path $\vec{u}_{t:T}$ is

$$
\begin{aligned}
C(\vec{\theta}, t, \vec{u}_{t:T}) &= \left\langle \phi(\theta(T)) + \int_t^T \frac{1}{2}\vec{u}^T(t)\vec{u}(t) \right\rangle \\
\phi(\vec{\theta}) &= \frac{\alpha}{2}\left((x_n(\vec{\theta}) - x_{\text{target}})^2 + (y_n(\vec{\theta}) - y_{\text{target}})^2\right)
\end{aligned}
$$

with $x_{\text{target}}, y_{\text{target}}$ the target coordinates of the end joint.

The variational method

Because $V = 0, f = 0, g = 1$, the solution to uncontrolled dynamics is Gaussian [19]

$$\psi(\vec{\theta^0}, t) \quad = \quad \int d\vec{\theta} \left(\frac{1}{\sqrt{2\pi\nu(T-t)}} \right)^n \exp\left(-\sum_{i=1}^{n} (\theta_i - \theta_i^0)^2 / 2\nu(T-t) - \phi(\vec{\theta})/\nu \right)$$

The control at time t for all components i is given by

$$u_i \quad = \quad \frac{1}{T-t} \left(\langle \theta_i \rangle - \theta_i^0 \right) \tag{10}$$

where $\langle \theta_i \rangle$ is the expectation value of θ_i computed wrt the probability distribution

$$p(\vec{\theta}) = \frac{1}{\psi(\vec{\theta^0}, t)} \exp\left(-\sum_{i=1}^{n} (\theta_i - \theta_i^0)^2 / 2\nu(T-t) - \phi(\vec{\theta})/\nu \right) \tag{11}$$

[19] This is not exactly correct because θ is a periodic variable. One should use the solution to diffusion on a circle instead. We can ignore this as long as $\sqrt{\nu(T-t)}$ is small compared to 2π.

The variational method

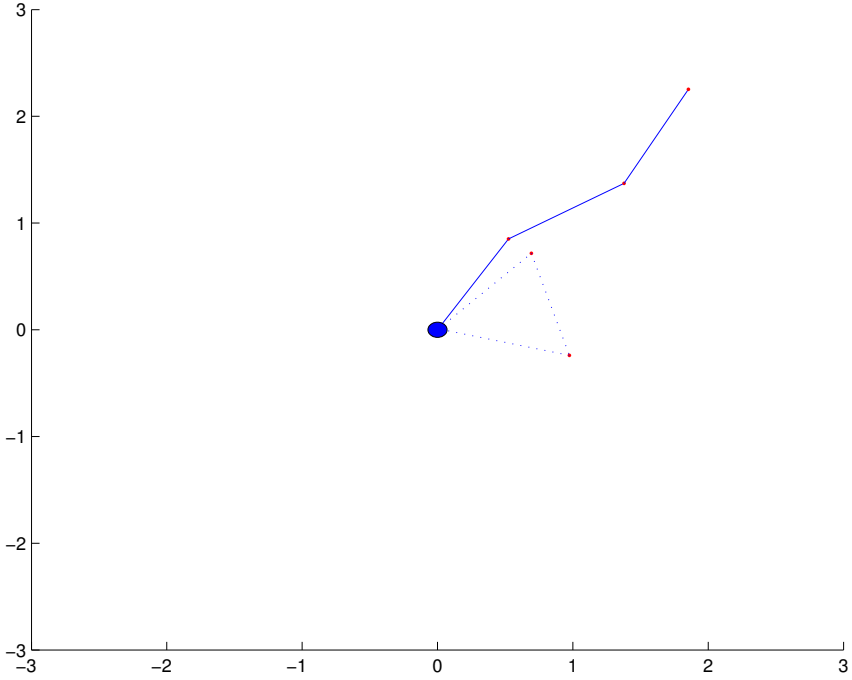
We compute the expectations $\left\langle \vec{\theta} \right\rangle$ by introducing a factorized Gaussian variational distribution $q(\vec{\theta}) = \prod_{i=1}^{n} \mathcal{N}((\theta_i|\mu_i, \sigma_i)$. We compute μ_i and σ_i by by minimizing the KL divergence between $q(\vec{\theta})$ and $p(\vec{\theta})$:

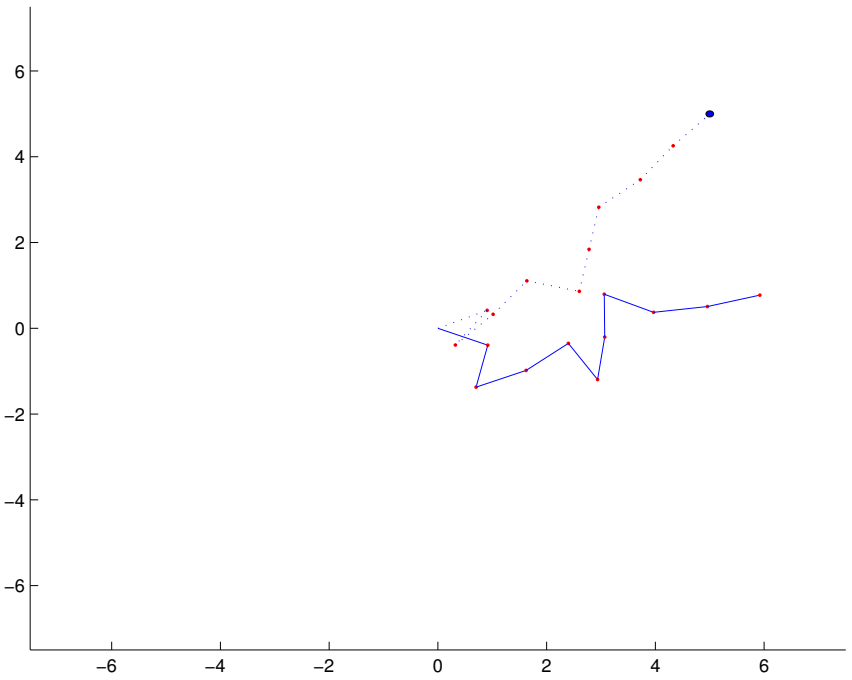
$$
\begin{aligned}
KL &= \int d\theta q(\theta) \log \frac{q(\theta)}{p(\theta)} \\
&= -\sum_{i=1}^{n} \log \sqrt{2\pi\sigma_i^2} + \log \psi(\vec{\theta^0}, t) + \frac{1}{2\nu(T-t)} \sum_{i=1}^{n} \left(\sigma_i^2 + (\mu_i - \theta_i^0)^2\right) + \frac{1}{\nu} \left\langle \phi(\vec{\theta}) \right\rangle_q
\end{aligned}
$$

where we omit irrelevant constants. $\left\langle \phi(\vec{\theta}) \right\rangle$ can be computed in closed form. Setting the derivative of the KL with respect to μ_i and σ_i^2 equal to zero:

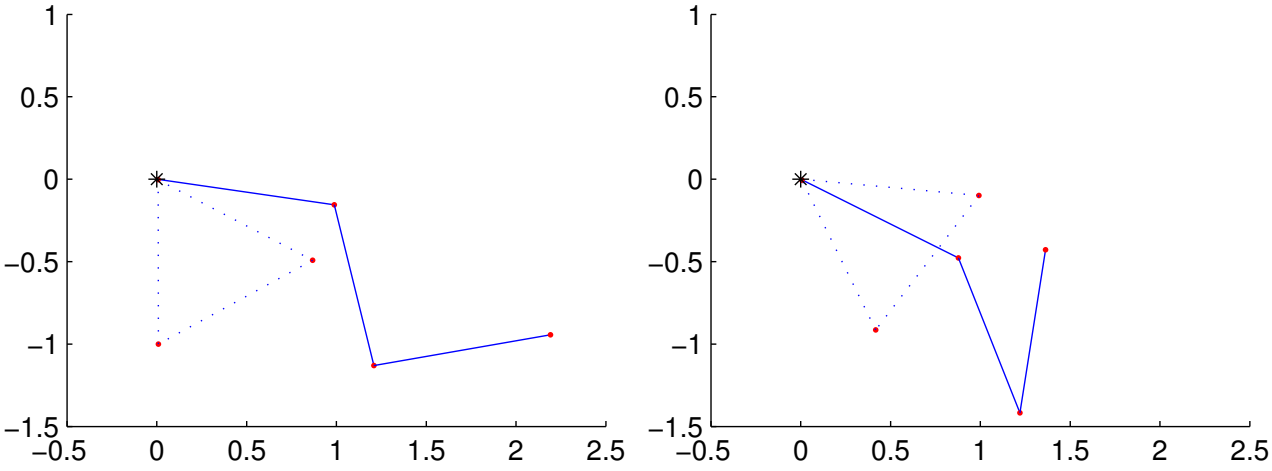
$$
\begin{aligned}
\mu_i &\leftarrow \theta_i^0 + \alpha(T-t)\left(\sin\mu_i e^{-\sigma_i^2/2}(\langle x_n \rangle - x_{\text{target}}) - \cos\mu_i e^{-\sigma_i^2/2}(\langle y_n \rangle - y_{\text{target}})\right) \\
\frac{1}{\sigma_i^2} &\leftarrow \frac{1}{\nu}\left(\frac{1}{(T-t)} + \alpha e^{-\sigma_i^2} - \alpha\left(\langle x_n \rangle - x_{\text{target}}\right)\cos\mu_i e^{-\sigma_i^2/2} - \alpha\left(\langle y_n \rangle - y_{\text{target}}\right)\sin\mu_i e^{-\sigma_i^2/2}\right)
\end{aligned}
$$

The variational method





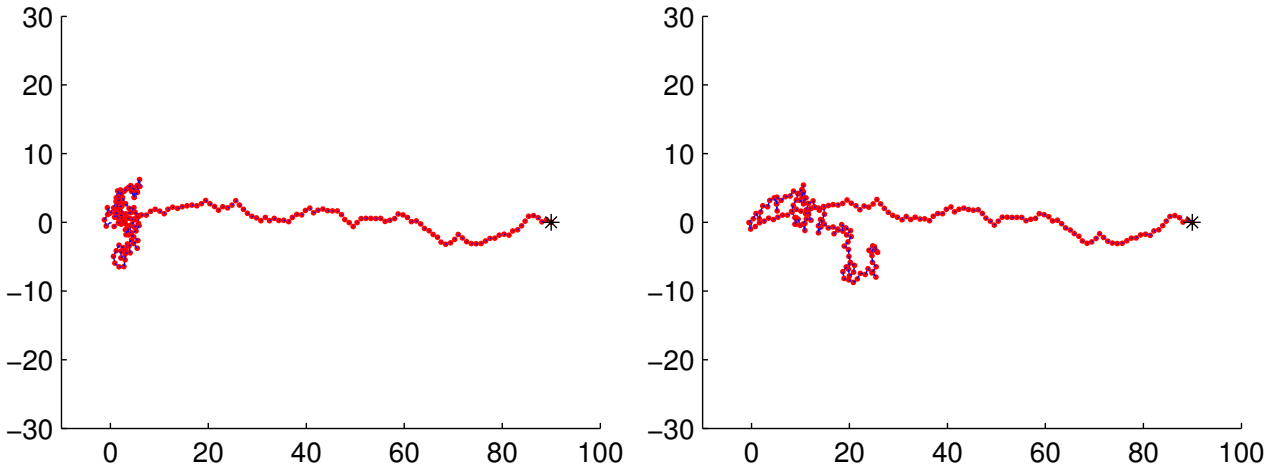
The variational method



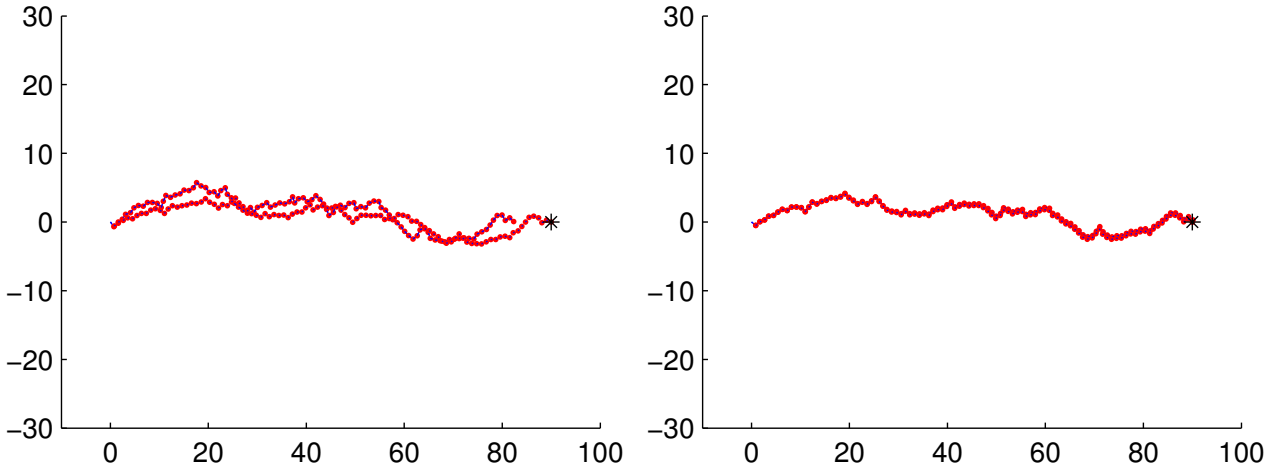
(a) $t = 0.05$ (b) $t = 0.55$

(c) $t = 1.8$ (d) $t = 2.0$

The variational method



(e) $t = 0.05$ (f) $t = 0.55$



(g) $t = 1.8$ (h) $t = 2.0$

Note, that the computation of $\langle \theta_i \rangle$ solves the coordination problem between the different joints.

Once $\langle\theta_i\rangle$ is known, each θ_i is steered independently to its target value $\langle\theta_i\rangle$ using the control law Eq. 10. The computation of $\langle\theta_i\rangle$ in the variational approximation is very efficient and can be used to control arms with hundreds of joints.

Coordination of agents

n agents with independent dynamics

$$dx_\alpha = (f_\alpha(x_\alpha, t) + u_\alpha) + d\xi_\alpha, \quad \alpha = 1, \ldots, n$$

should coordinate their actions to minimize a cost at a future time $t = T$:

$$\phi(y_1, \ldots, y_n) \quad y_\alpha \in \{z_1, \ldots z_k\}$$

and $\phi = \infty$ elsewhere.

Coordination of agents

Then,

$$\begin{aligned}
\Psi(x_1,\ldots,x_n,t) &= \int dy_1 \ldots dy_n \prod_\alpha \rho(y_\alpha, T|x_\alpha, t) \exp(-\phi(y_1,\ldots,y_n)/\nu) \\
&= \sum_{\vec{y}} \exp(-E(\vec{y}|\vec{x},t)/\nu) \\
p(\vec{y}) &= \frac{1}{Z}\exp(-E(\vec{y}|\vec{x},t)/\nu) \\
u_\alpha(\vec{x},t) &= -\partial_{x_\alpha} J = \left\langle \frac{\partial \log \rho(y_\alpha, T|x_\alpha, t)}{\partial x_\alpha} \right\rangle
\end{aligned}$$

with $\vec{x} = (x_1,\ldots,x_n), \vec{y} = (y_1,\ldots,y_n)$.

E has a graphical model structure if ϕ has.

Pseudo code

Loop:

1. Compute the cost and its log derivative for each agent to move to each target:

$$\rho(z_i, T|x_\alpha, t), \quad i = 1, \ldots, k, \quad \alpha = 1, \ldots, n$$

   This path integral can be estimated using MC sampling or variational approximation.
2. Compute u_α using graphical model inference in $p(\vec{y})$ (exact, BP, MF).

A simple 1d example

Intrinsic dynamics $f_\alpha = 0$, $V(x_1, \ldots, x_n) = 0$:

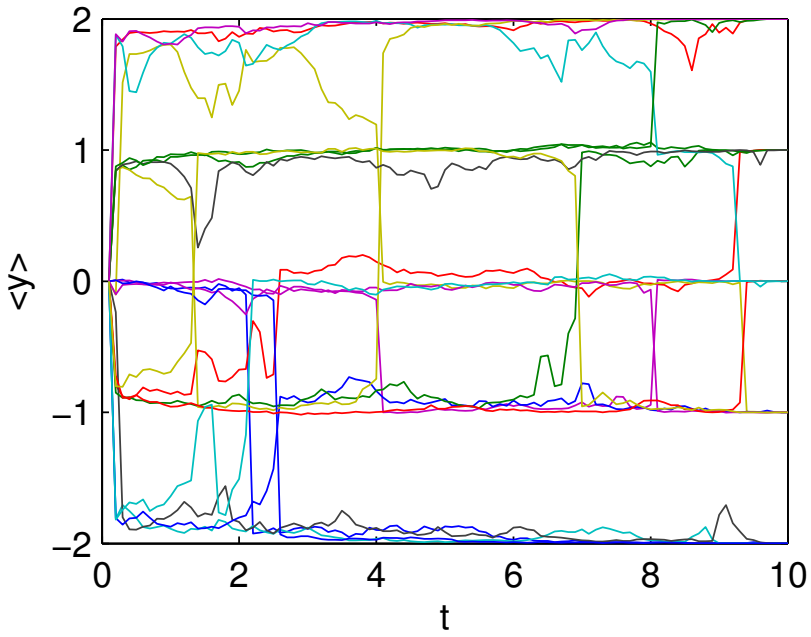
$$p(y_\alpha, T | x_\alpha, t) \propto \exp(-(y_\alpha - x_\alpha)^2 / 2\nu(T - t))$$

End cost $\phi(y_1, \ldots, y_n) = \sum_{j=1}^{k} (n_j(\vec{y}) - n_j)^2$, with $n_j(\vec{y})$ the # of agents that go to target j.

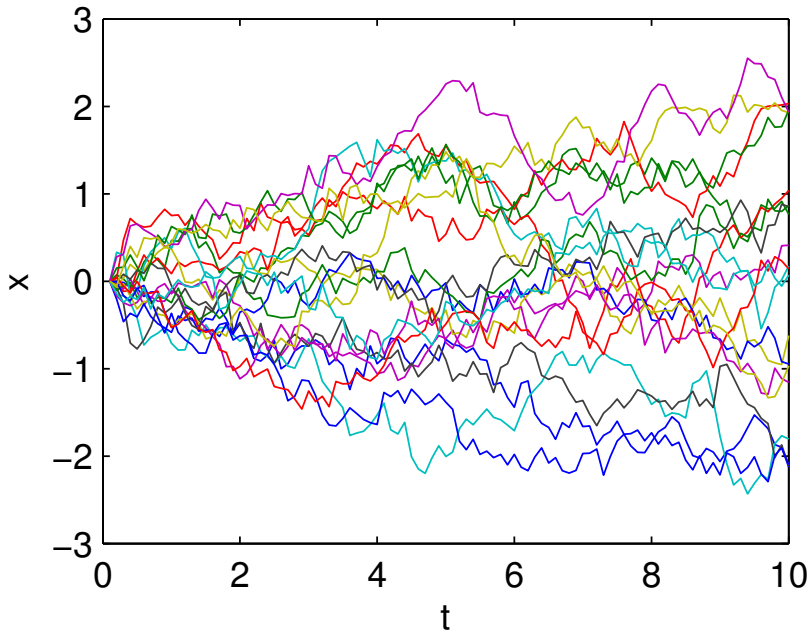
Optimal control is for agent α is

$$u_\alpha = \frac{1}{T - t}(\langle y_\alpha \rangle - x_\alpha)$$

A simple 1d example

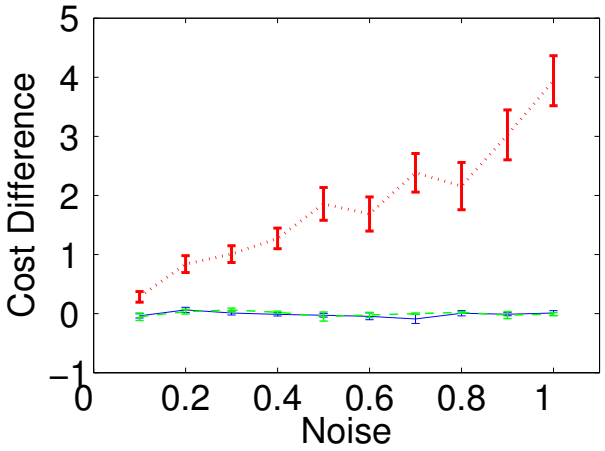


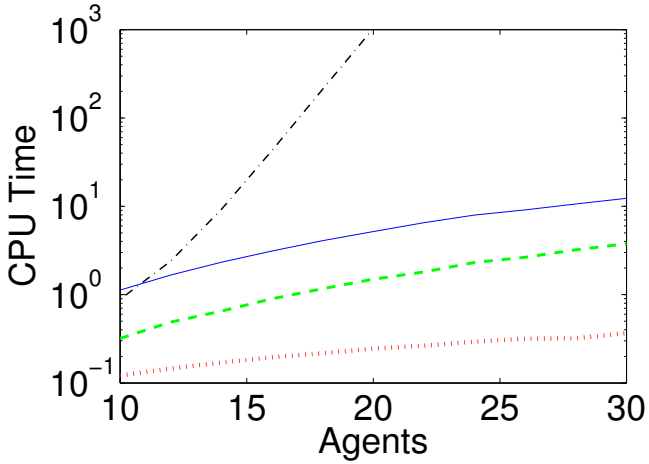
(i) Agent predicted target $\langle y_\alpha \rangle$



(j) Agent position x

A simple 1d example





Control cost
greedy control (red)
MF control (blue)
BP control (green)

CPU time
exact control (black)
MF control (blue)
BP control (green)
greedy control (red)

Nonlinear Coordination

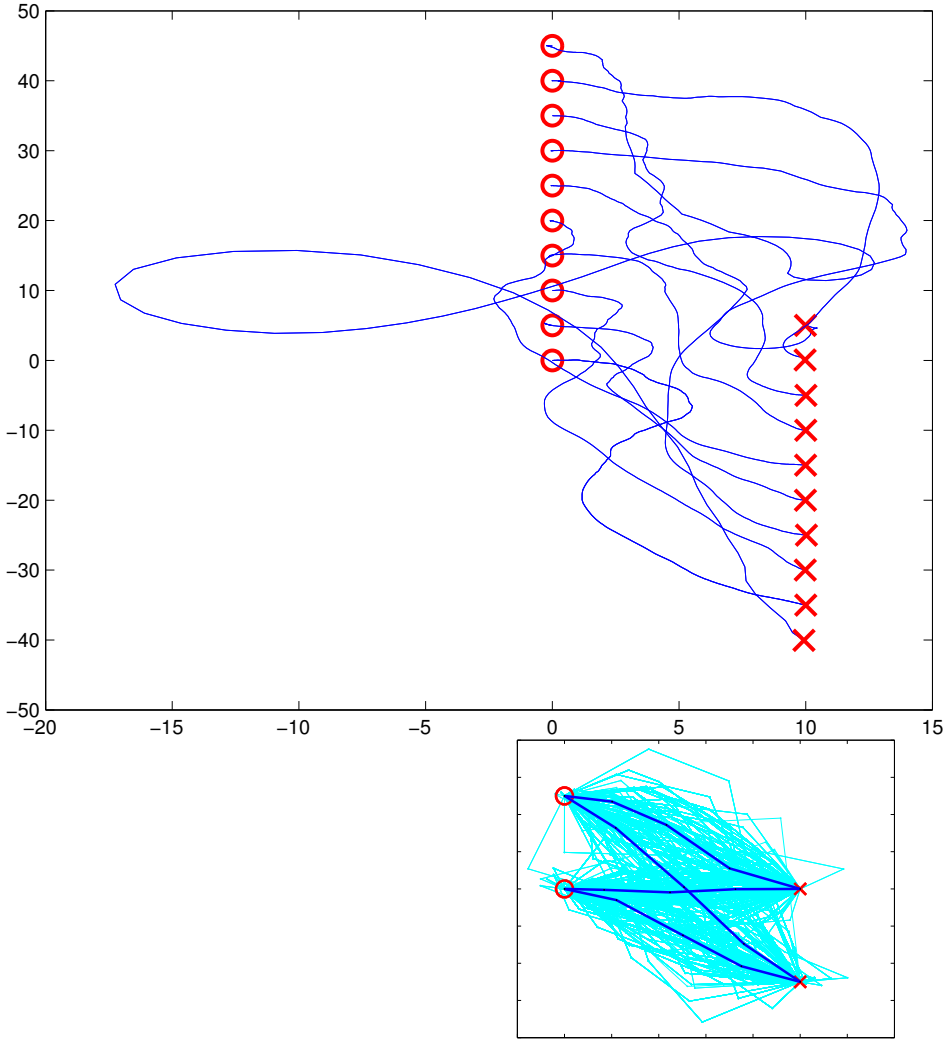
Agents $a = 1, \ldots, n$ in $2D$:

$$
\begin{aligned}
dx_a(t) &= v_a(t) \cos \varphi_a(t)\, dt \\
dy_a(t) &= v_a(t) \sin \varphi_a(t)\, dt \\
dv_a(t) &= u_a(t)dt + d\xi_a(t) \\
d\varphi_a(t) &= \omega_a(t)dt + d\zeta_a(t)
\end{aligned}
$$

Initial states O, $v_a(0) = 0$, $\varphi_a(0) = 0$
Targets X, $v_a(T) = 0$, $\varphi_a(T) = 0$

Sample paths specified at
$t_i = t + i\, dt$,
$i = 0, \ldots, 6$, $dt = (T - t)/6$

Example of 10 agents & 10 targets:



Sample paths:

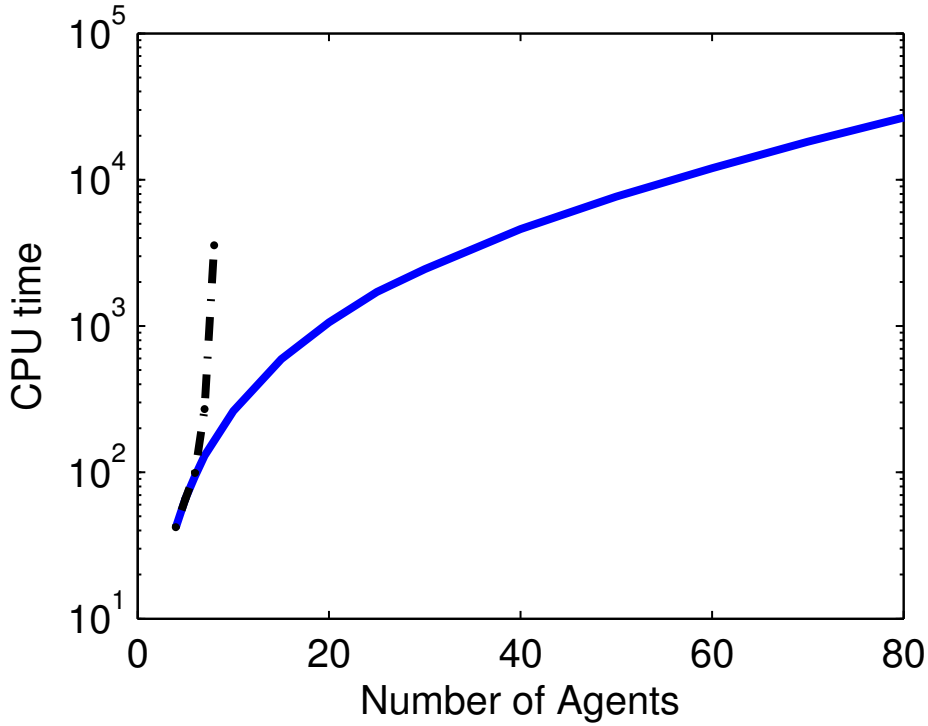
Computation Time

Inference methods:

Junction Tree ($\cdot - \cdot$)
MF (—)

(100 sample paths per agent-target)

CPU time (s) vs. number of agents:



(# agents = # targets)

JT : exponential in number of agents
(intractable for # agents > 10)
MF : polynomial in number of agents

Risk sensitive control

It is relatively straightforward to generalize the path integral method to optimize a cost of the form

$$\begin{aligned} \tilde{C} &= \phi(x_T) + \int \frac{1}{2} u^T R u + V(x) \\ C &= \frac{1}{\theta} \log \left\langle \exp(\theta \tilde{C}) \right\rangle \end{aligned}$$

For $\theta = 0$ the risk neutral control is recovered. For θ small:

$$C = \left\langle \tilde{C} \right\rangle + \frac{\theta}{2} \left(\left\langle \tilde{C}^2 \right\rangle - \left\langle \tilde{C} \right\rangle^2 \right) + h.o.$$

$\theta > 0$ is risk averse, $\theta < 0$ is risk seeking.

vd Broek et al. UAI 2010

Risk sensitive control

We illustrate the behavior for the (well known) LQ case. $V = f = 0, \phi = \alpha/2x^2$.

The optimal control is given by

$$u = \frac{-\alpha x}{R + \alpha(T-t)(1-\nu R\theta)}$$

For $\theta < 0$ control is weaker
For $0 < \theta < 1/R\nu$ control is stronger
In both cases control increases with time.

For $\theta > 1/R\nu$, control is only well-defined when the denominator is positive:

$$\alpha(T-t) < \frac{R}{\nu R\theta - 1}$$

Control decreases with time. For larger time-to-go, the expected cost is infinite.

vd Broek et al. UAI 2010

Inference and control

As an example of the intricacies of joint inference and control , consider the simple LQ control problem [16, 17]

$$dx = \alpha u dt + d\xi \tag{12}$$

$$C(x_0, \theta_0, u(0 \to T)) = \left\langle \phi(x(T)) + \int_0^T dt R(x, u, t) \right\rangle \tag{13}$$

with α *unobserved* and x observed. Path cost $R(x, u, t)$ and end cost $\phi(x)$ and noise variance ν are given.

Although α is unobserved, we have a means to observe α indirectly through the sequence $x_t, u_t, t = 0, \ldots$. Each time step we observe dx and u and we can thus update our belief about α using the Bayes formula:

$$p_{t+dt}(\alpha|dx, u) \propto p(dx|\alpha, u) p_t(\alpha) \tag{14}$$

$p(dx|\alpha, u)$ is Normal in dx with variance νdt
$p_t(\alpha)$ our belief at time t about the values of α

The information that we receive about α increases with u, because the $\alpha u dt$ term dominates the $d\xi$ term. However, large u values are more costly and also may drive us away from our target state $x(T)$.

Thus, the optimal control is a balance between optimal inference and minimal control cost.

The solution is to augment the state space with parameters θ_t (sufficient statistics) that describe $p_t(\alpha) = p(\alpha|\theta_t)$ and θ_0 known, which describes our initial belief in the possible values of α. The cost that must be minimized is

$$C(x_0, \theta_0, u(0 \to T)) = \left\langle \phi(x(T)) + \int_0^T dt R(x, u, t) \right\rangle \tag{15}$$

where the average is with respect to the noise $d\xi$ as well as the uncertainty in α.

NB: the average over α depends on θ_t which is not known beforehand.

For simplicity, consider the example that α attains only two values $\alpha = \pm 1$. Then $p_t(\alpha|\theta) = \sigma(\alpha\theta)$, with the sigmoid function $\sigma(x) = \frac{1}{2}(1 + \tanh(x))$. The update equation Eq. 14 implies a dynamics for θ:

$$d\theta = \frac{u}{\nu} dx = \frac{u}{\nu}(\alpha u dt + d\xi)$$

20

With $z_t = (x_t, \theta_t)$ we obtain a standard HJB Eq.

$$-\partial_t J(t,z)dt \quad = \quad \min_u \left(R(t,x,u)dt + \langle dz \rangle_z \partial_z J(z,t) + \frac{1}{2} \left\langle dz^2 \right\rangle_z \partial_z^2 J(z,t) \right)$$

with boundary condition $J(z,T) = \phi(x)$ (NB independent of θ).

[20] The rhs of the Bayes rule is

$$p(dx|\alpha,u)p(\alpha|\theta_t) \propto \exp\left(-\frac{(dx - \alpha u dt)^2}{2\nu dt}\right) \exp(\alpha\theta_t) \propto \exp\left(\frac{dx\alpha u}{\nu} + \alpha\theta_t\right) = \exp\left(\alpha\left(\theta_t + \frac{dxu}{\nu}\right)\right)$$

The result is

$$-\partial_t J \quad = \quad \min_u \left(R(x,u,t) + \bar{\alpha} u \partial_x J + \frac{u^2 \bar{\alpha}}{\nu} \partial_\theta J + \frac{1}{2} \nu \partial_x^2 J + \frac{1}{2} \frac{u^2}{\nu} \partial_\theta^2 J + u \partial_x \partial_\theta J \right)$$

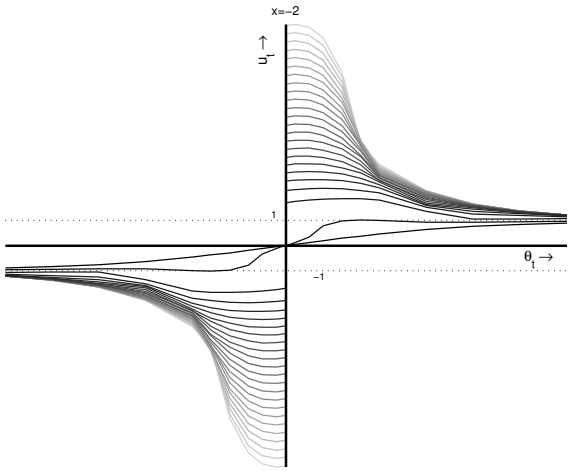
[21] with boundary conditions $J(x, \theta, T) = \phi(x)$.

Thus, the dual control problem (joint inference on α and control problem on x) has become an ordinary control problem in x, θ (Florentin, 1962).

Note that if R, ϕ are quadratic and α is known, this is an LQ problem. However, when α is not known, the corresponding dual control problem is not LQ (because of the additional u dependent terms).

[21] The expectation values appearing in this equation are conditioned on (x_t, θ_t) and are averages over $p(\alpha|\theta_t)$ and the Gaussian noise. $\langle dx \rangle_{x,\theta} = \bar{\alpha} u dt$, $\langle d\theta \rangle_{x,\theta} = \frac{\bar{\alpha} u^2}{\nu} dt$, $\left\langle dx^2 \right\rangle_{x,\theta} = \nu dt$, $\left\langle d\theta^2 \right\rangle_{x,\theta} = \frac{u^2}{\nu} dt$, $\langle dx d\theta \rangle = u dt$, with $\bar{\alpha} = \tanh(\theta)$ the expected value of α for a given value θ.

Probing

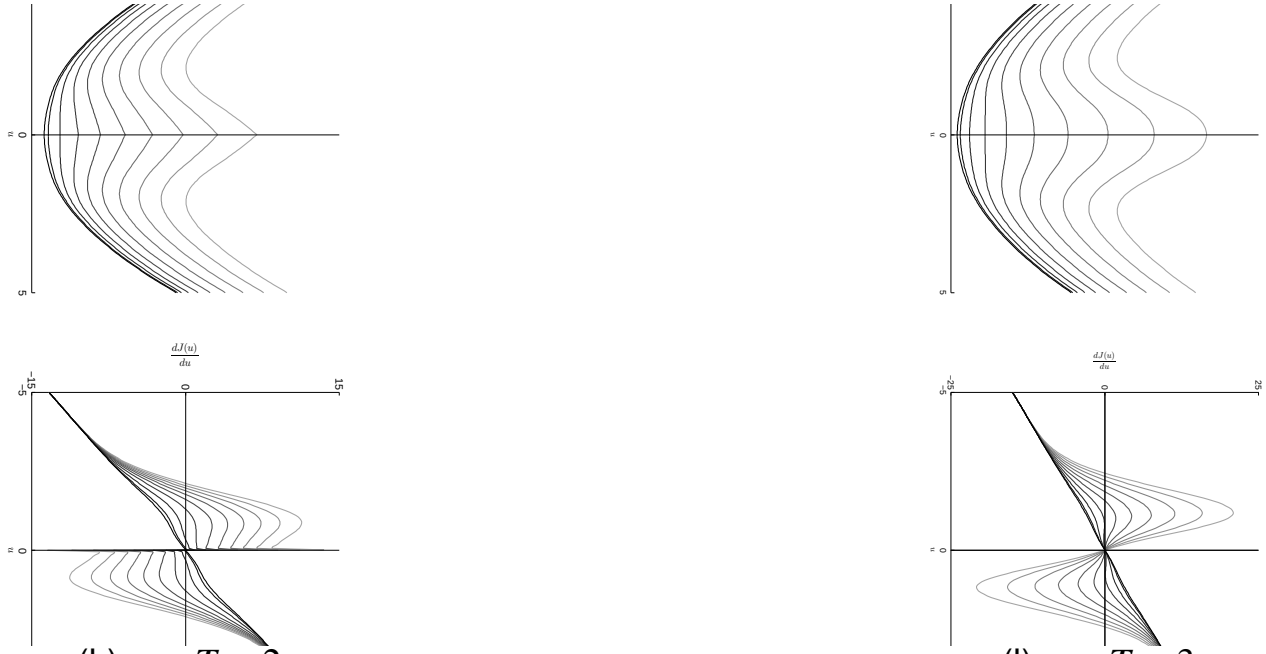


Dual control solution with end cost $\phi(x) = x^2$ and path cost $\int_t^{t_f} dt' \frac{1}{2} u(t')^2$ and $\nu = 0.5$. Plot shows the deviation of the control from the certain case: $u_t(x, \theta)/u_t(x, \theta = \pm\infty)$ as a function of θ for different values of t and $x = 2$. The curves with the larger values are for larger times-to-go.

'Probing': u is much larger when α is uncertain (θ small) then when α is certain $\theta = \pm\infty$.

Symmetry breaking and non-differentiability of J

The observed probing behavior arises as the result of a symmetry breaking in the right hand side of the Bellman equation.



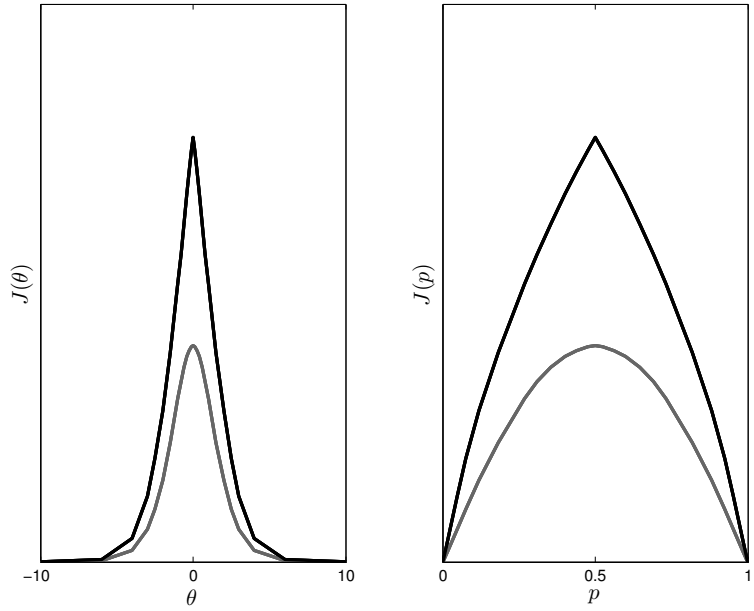
(k) $t = T - 2$ (l) $t = T - 3$

Figure 5: Rhs of the Bellman equation as a function of u and its derivative for $\theta = 0$. The different curves correspond to different values of x. Explorative behavior ($u \neq 0$) arises in the no-knowledge state $\theta = 0$ by proposing non-zero controls. The singularity is absent at $t = T - 2$ and present starting from $t = T - 3$.

Symmetry breaking and non-differentiability of J

As a result of the local minima in the Bellman optimization, the optimal value function is not differentiable.

The optimal cost-to-go is convex in the belief [18].



Left) $J_t(x, \theta)$ for $t = T - 2, x = -2$ (grey) and $t = T - 2, x = -6$ (black) versus θ Right) Same as left, but as a function of the belief $p = p(b = 1|\theta)$.