Path integral control theory

Bert Kappen
SNN Donders Institute for Neuroscience, Radboud University, Nijmegen
Gatsby Computational Neuroscience Unit, UCL London

May 28, 2020

The sensori-motor problem

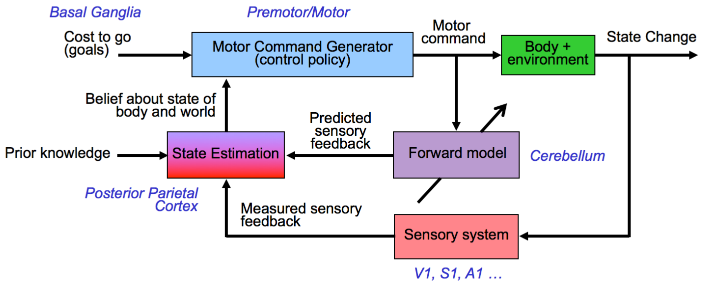
Brain is a sensori-motor machine:

- perception
- action
- perception causes action
- action causes perception
- learning by trial and error

The sensori-motor problem

Brain is a sensori-motor machine:

- perception
- action
- perception causes action
- action causes perception
- learning by trial and error



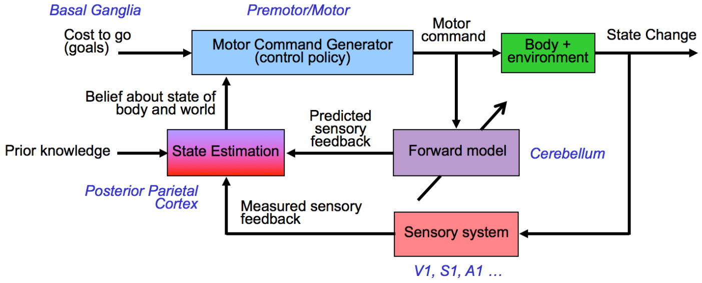
Separately, we understand perception and action (somewhat):

- Perception is (Bayesian) statistics, information theory, max entropy

The sensori-motor problem

Brain is a sensori-motor machine:

- perception
- action
- perception causes action
- action causes perception
- learning by trial and error



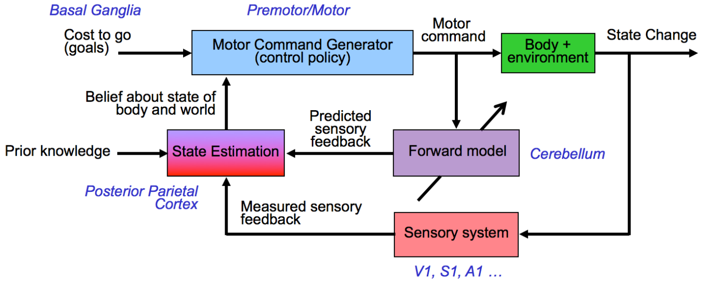
Separately, we understand perception and action (somewhat):

- Perception is (Bayesian) statistics, information theory, max entropy
- Learning is parameter estimation

The sensori-motor problem

Brain is a sensori-motor machine:

- perception
- action
- perception causes action
- action causes perception
- learning by trial and error



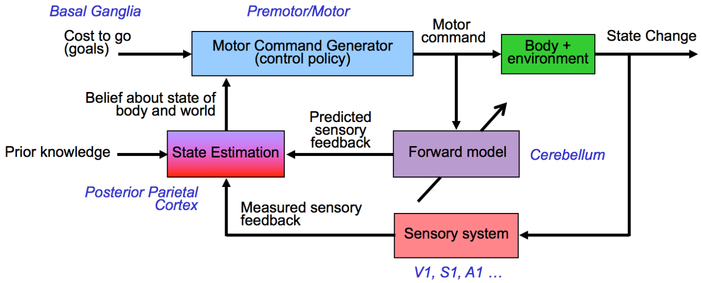
Separately, we understand perception and action (somewhat):

- Perception is (Bayesian) statistics, information theory, max entropy
- Learning is parameter estimation
- Action is control theory, but
  - computing 'backward in time'?
  - representing control policies, action hierarchies, learning multiple tasks?
  - model based vs. model free?

The sensori-motor problem

Brain is a sensori-motor machine:

- perception
- action
- perception causes action
- action causes perception
- learning by trial and error



Separately, we understand perception and action (somewhat):

- Perception is (Bayesian) statistics, information theory, max entropy
- Learning is parameter estimation
- Action is control theory, but
  - computing 'backward in time'?
  - representing control policies, action hierarchies, learning multiple tasks?
  - model based vs. model free?

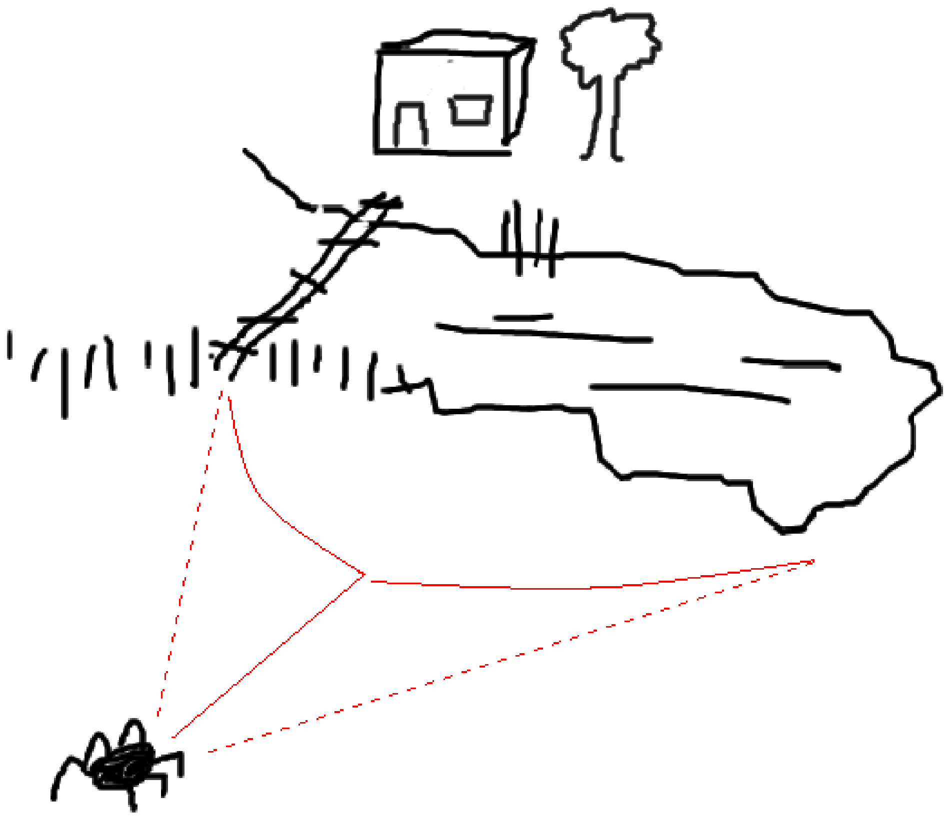
We seem to have no good theories for the combined sensori-motor problem.

- Sensing depends on actions, features depend on task(s)
- Dual control formalism seems too hard

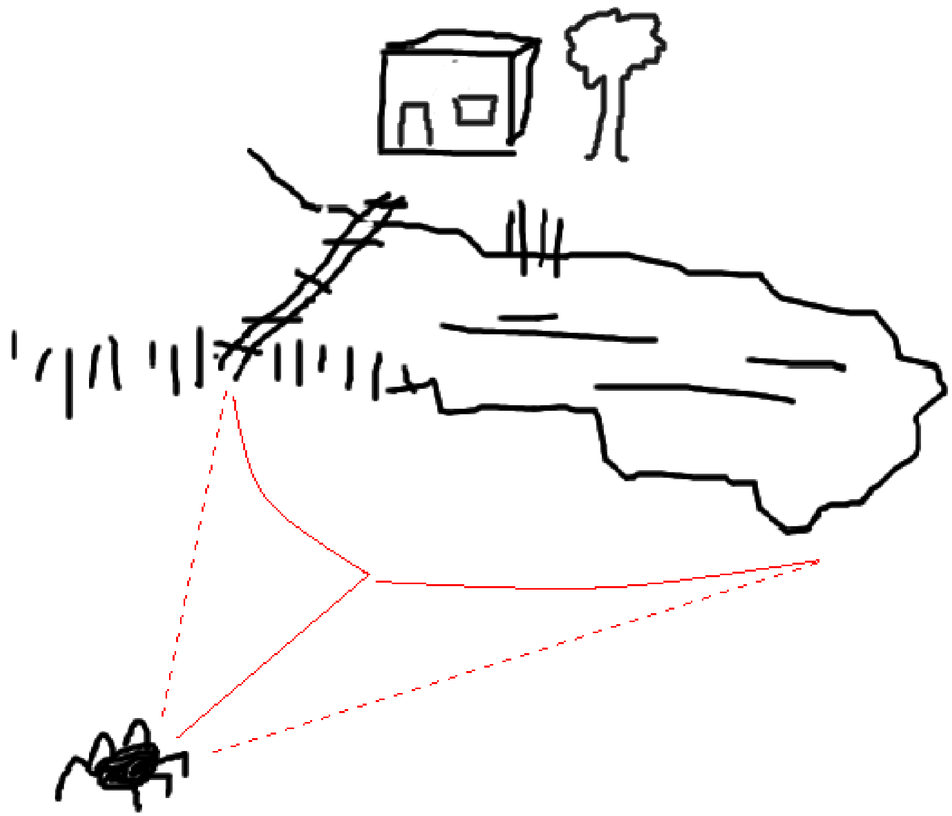
Optimal control theory

Given a current state and a future desired state, what is the best/cheapest/fastest way to get there.

Why stochastic optimal control?

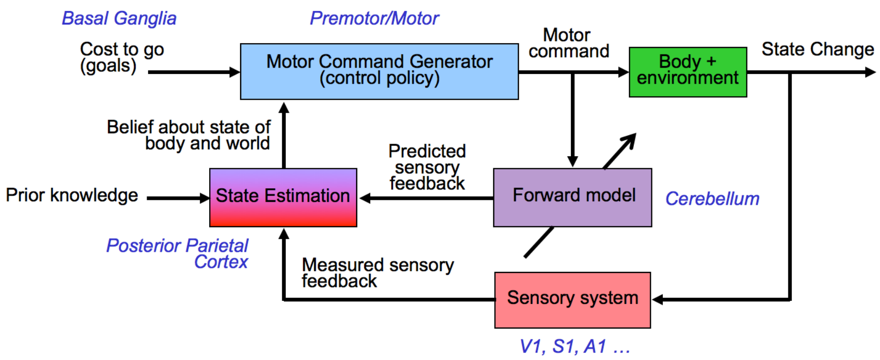


Why stochastic optimal control?



Optimality depends on the uncertainty.

Optimal control theory



$$\frac{dx}{dt} = f(x,u) \qquad x_0, u_{0:T} \rightarrow x_{0:T}$$

$$C(u_{0:T}, x_{0:T}) = \phi(x_T) + \int_0^T dt V(x_t, u_t)$$

Three hard problems:

- a learning and exploration problem: f, x, ϕ, V
- a stochastic optimal control computation: compute u^*
- a representation problem $u^*(x,t)$

The idea: Control, Inference and Learning

Path integral control theory
Express a control computation as an inference computation.
Compute optimal control using MC sampling

The idea: Control, Inference and Learning

Path integral control theory
Express a control computation as an inference computation.
Compute optimal control using MC sampling

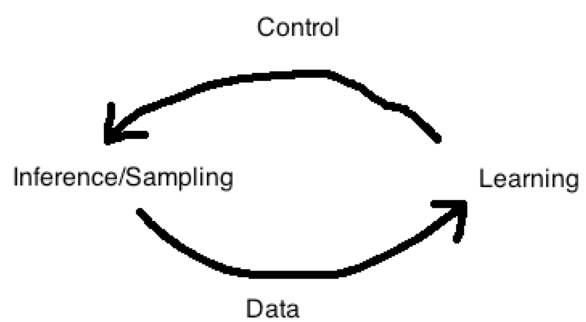
Importance sampling
Accellerate with importance sampling (=a state-feedback controller)
Optimal importance sampler is optimal control

The idea: Control, Inference and Learning

Path integral control theory
Express a control computation as an inference computation.
Compute optimal control using MC sampling

Importance sampling
Accellerate with importance sampling (=a state-feedback controller)
Optimal importance sampler is optimal control

Learning
Learn the controller from self-generated data
Use Cross Entropy method for parametrized controller



Outline

- Intro to optimal control theory
- Review of path integral control theory
- Importance sampling
  - Relation between optimal sampling and optimal control
- Cross entropy method for adaptive importance sampling (PICE)
  - A criterion for parametrized control optimization
  - Learning by gradient descent
- Some examples

Discrete time optimal control

Consider the control of a discrete time deterministic dynamical system:

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

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

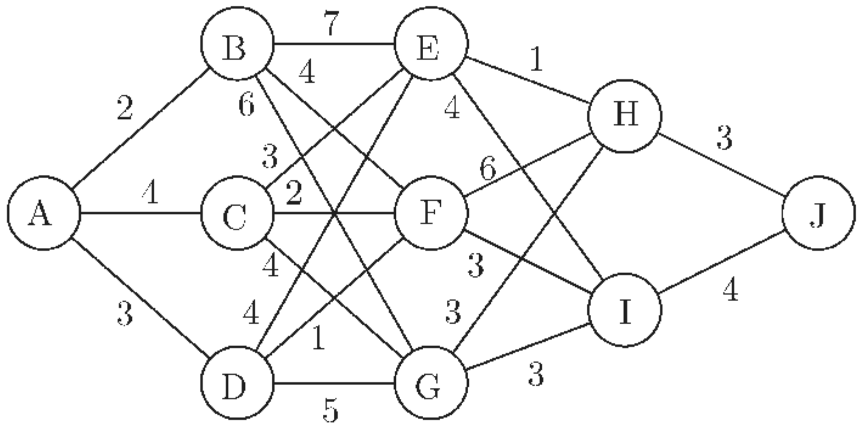
Given x_0 and $u_{0:T-1}$, we can compute $x_{1:T}$.

Define a cost for each sequence of controls:

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

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.

$$
\begin{aligned}
J(J) &= 0 \\
J(H) &= 3 \qquad J(I) = 4 \\
J(F) &= \min(6 + J(H), 3 + J(I)) = 7 \\
J(B) &= \min(7 + J(E), 4 + J(F), 2 + J(G)) = \ldots
\end{aligned}
$$

Minimal cost at time t easily expressable in terms of minimal cost at time $t + 1$.

Discrete time optimal control

Dynamic programming uses concept of optimal cost-to-go $J(t,x)$.

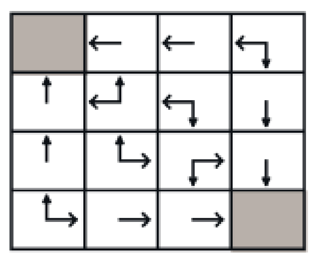
One can recursively compute $J(t,x)$ from $J(t+1,x)$ for all x in the following way:

$$
\begin{aligned}
J(t,x_t) &= \min_{u_t} \left(V(x_t,u_t) + J(t+1, x_t + f(t,x_t,u_t)) \right) \\
J(T,x) &= 0 \\
J(0,x) &= \min_{u_{0:T-1}} C(x, u_{0:T-1})
\end{aligned}
$$

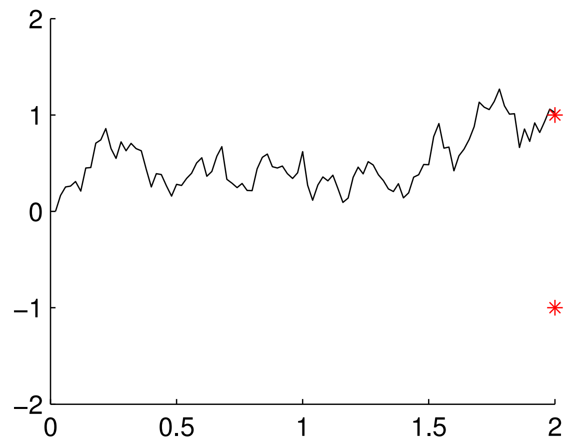
This is called the Bellman Equation.

Computes $u_t(x)$ for all intermediate t,x.

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



Stochastic control theory



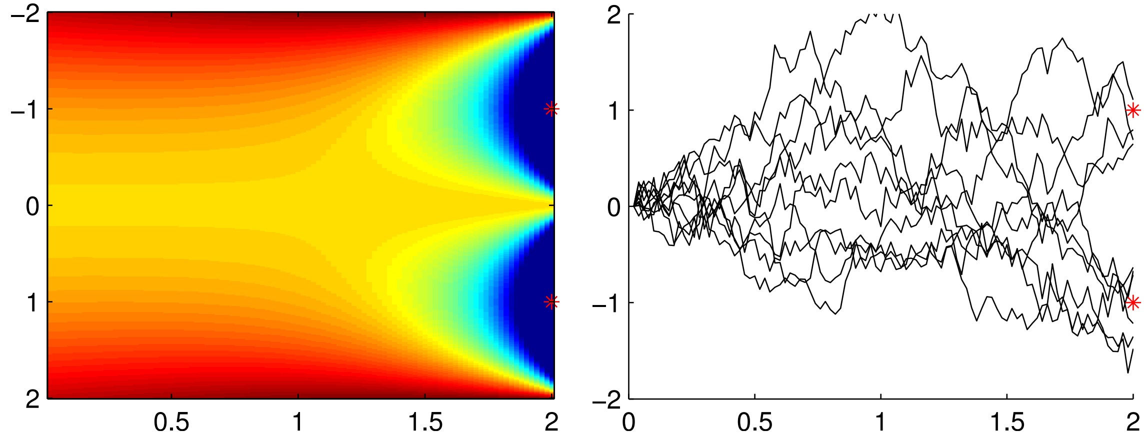
Consider a stochastic dynamical system

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

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

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

Control theory



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

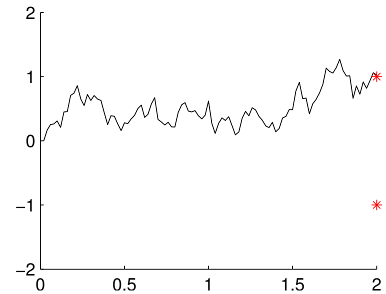
$$J(x, t) = \min u_{t:T} \mathbb{E}_u \left(\phi(X_T) + \int_t^T dt R(X_t, u(X_t, t)) \right) \qquad X_t = x$$

J satisfies a partial differential equation

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

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



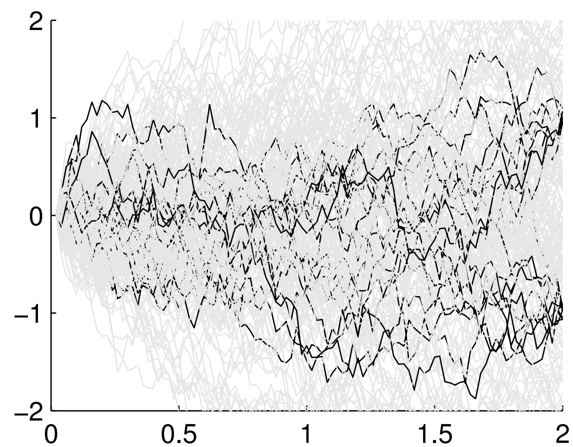
$$dX_t = \underbrace{f(X_t)dt + g(X_t)(u(X_t,t)dt}_{f(X_t,u)dt} + dW_t) \qquad X_0 = x_0$$

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

$$C(u|x_0) = \mathbb{E}\left(\phi(X_T) + \int_0^T dt\, \underbrace{V(X_t,t) + \frac{1}{2}u(X_t,t)^2}_{R(X_t,u(X_t,t))}\right) = \mathbb{E}\left(S(\tau) + \int_0^T dt \frac{1}{2}u(X_t,t)^2\right)$$

$$S(\tau) = \phi(X_T) + \int_0^T V(X_t,t)$$

Path integral control theory



Equivalent formulation: Define distributions

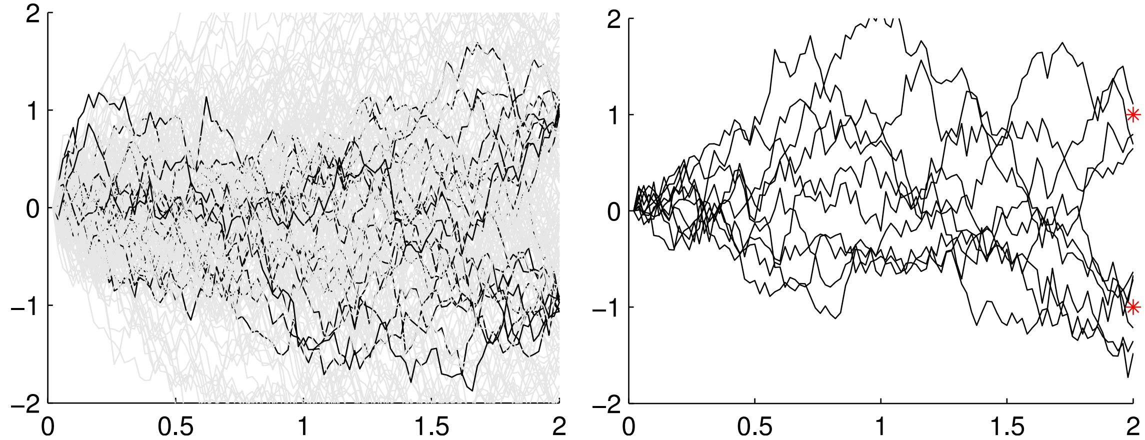
$$p(\tau|x_0): \qquad dX_t = f(X_t)dt + g(X_t)(u(X_t,t)dt + dW_t)$$
$$q(\tau|x_0): \qquad dX_t = f(X_t)dt + g(X_t)dW_t$$

Find distribution over trajectories p that minimizes

$$C(u|x_0) = \mathbb{E}\left(S(\tau) + \int_0^T dt \frac{1}{2} u(X_t,t)^2 \right) \quad \rightarrow \quad C(p|x_0) = \int d\tau p(\tau)\left(S(\tau) + \log \frac{p(\tau)}{q(\tau)} \right)$$

The optimal solution is given by $p^*(\tau|x_0) = \frac{1}{\psi(x_0)} q(\tau|x_0) e^{-S(\tau)}$

Path integral control theory

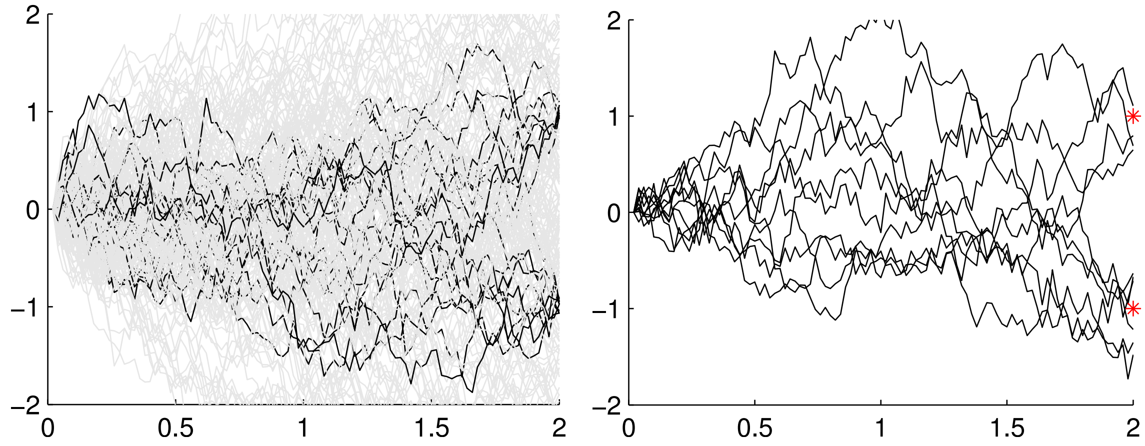


So we have two solutions to the same problem:

$$p^*(\tau|x_0) = \frac{1}{\psi(x_0)} q(\tau|x_0) e^{-S(\tau)} \qquad p(\tau|x_0, u^*(x,t))$$

These solutions are identical (Girsanov Thm).

Path integral control theory

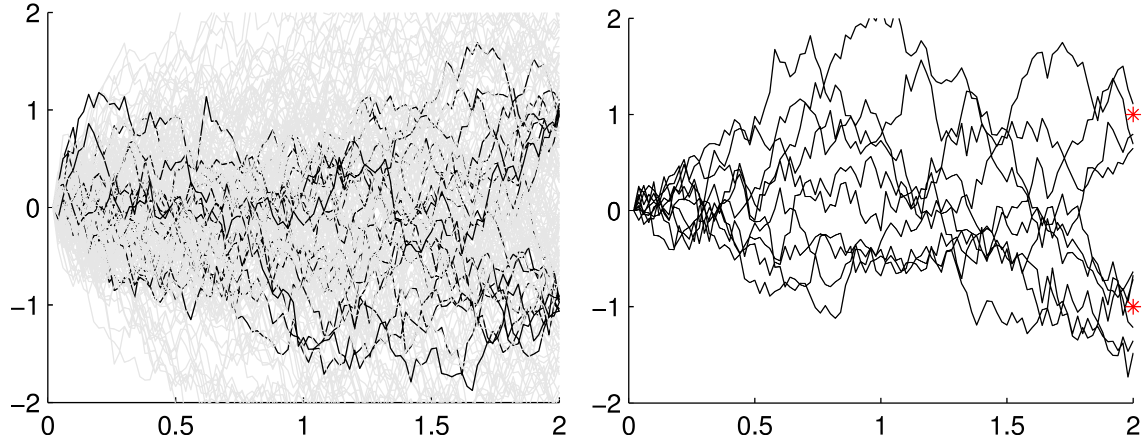


The optimal control cost is $C(p^*|x_0) = -\log\psi(x_0)$ with

$$\psi(x_0) \quad = \quad \int d\tau q(\tau|x_0)e^{-S(\tau)} = \mathbb{E}_q e^{-S}$$

Thus, we identify $J(x,t) = -\log\psi(x,t)$ as the optimal cost-to-go. $J(x,t)$ can be estimated by forward sampling from $q(\tau|x,t)$.

Path integral control theory

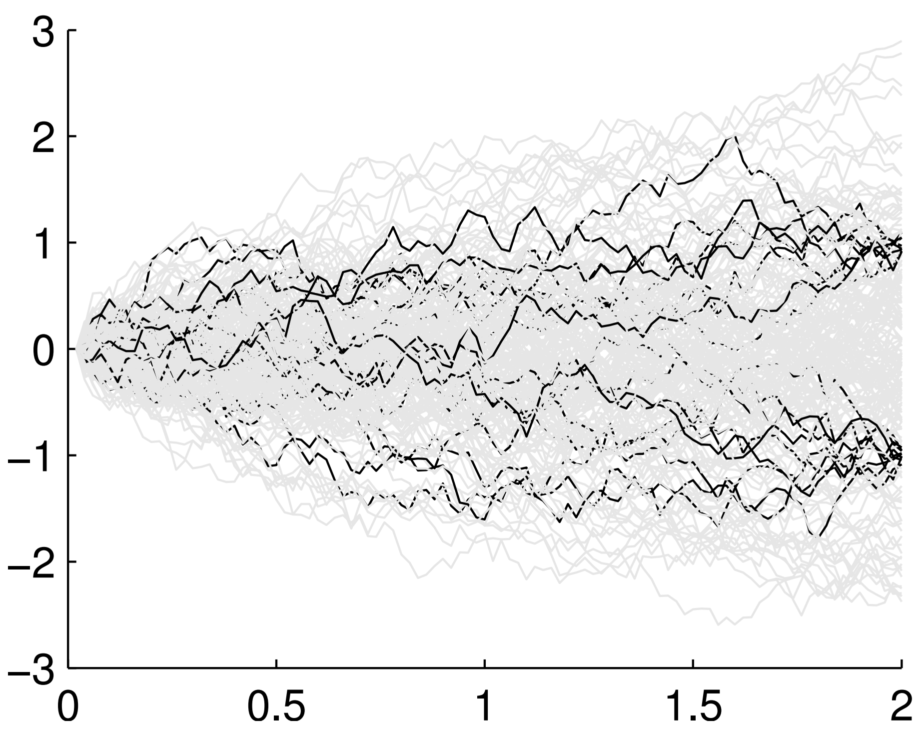


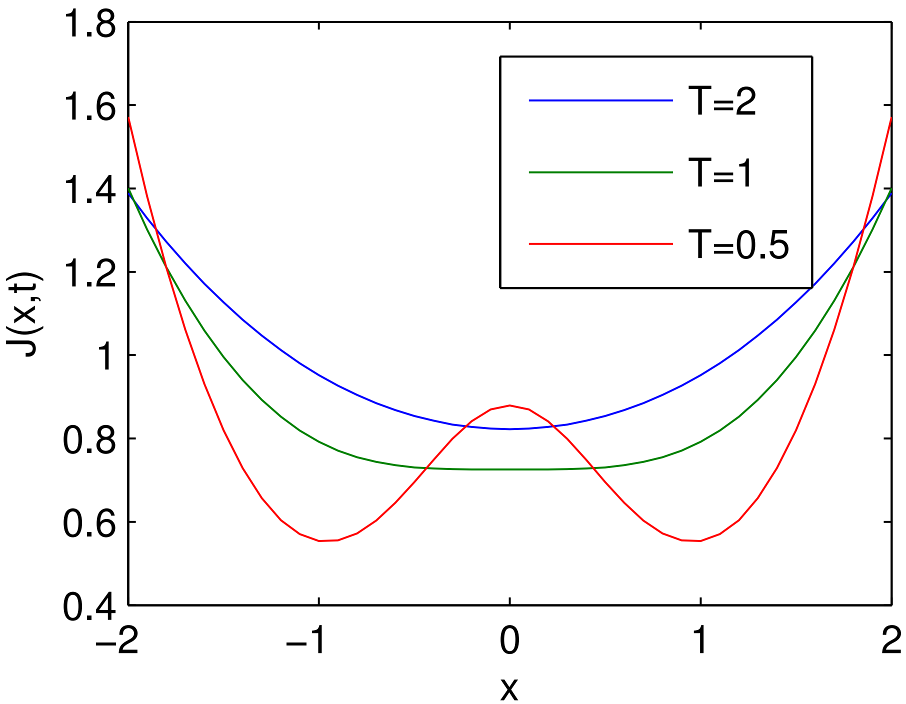
The optimal control

$$u^*(x,t)dt \quad = \quad \mathbb{E}_{p^*}(dW_t) = \frac{\mathbb{E}_q\left(dWe^{-S}\right)}{\mathbb{E}_q\left(e^{-S}\right)}$$

Delayed choice

Time-to-go $T = 2 - t$.



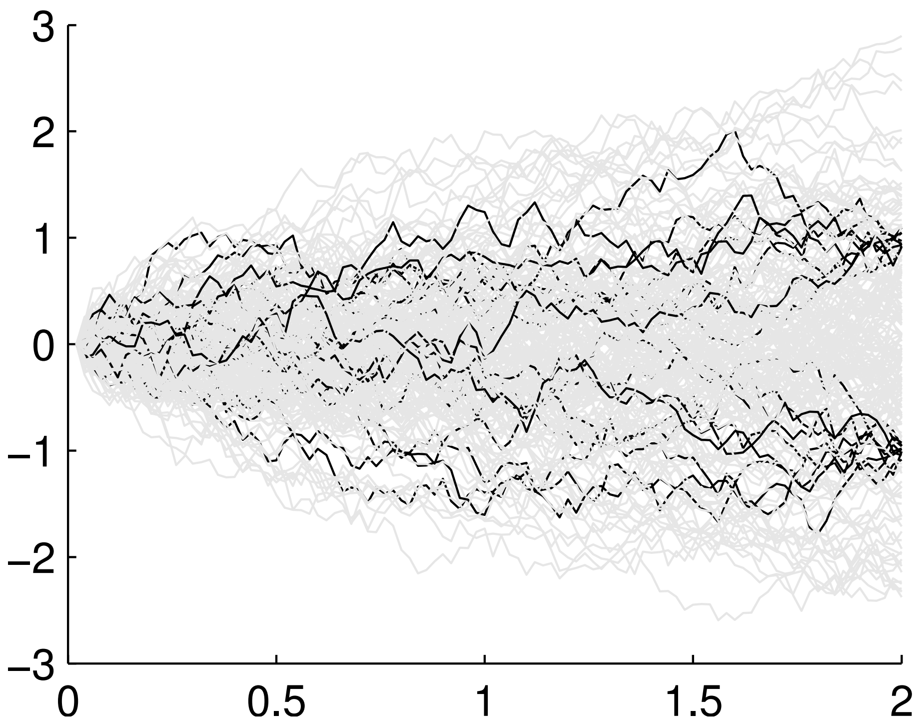


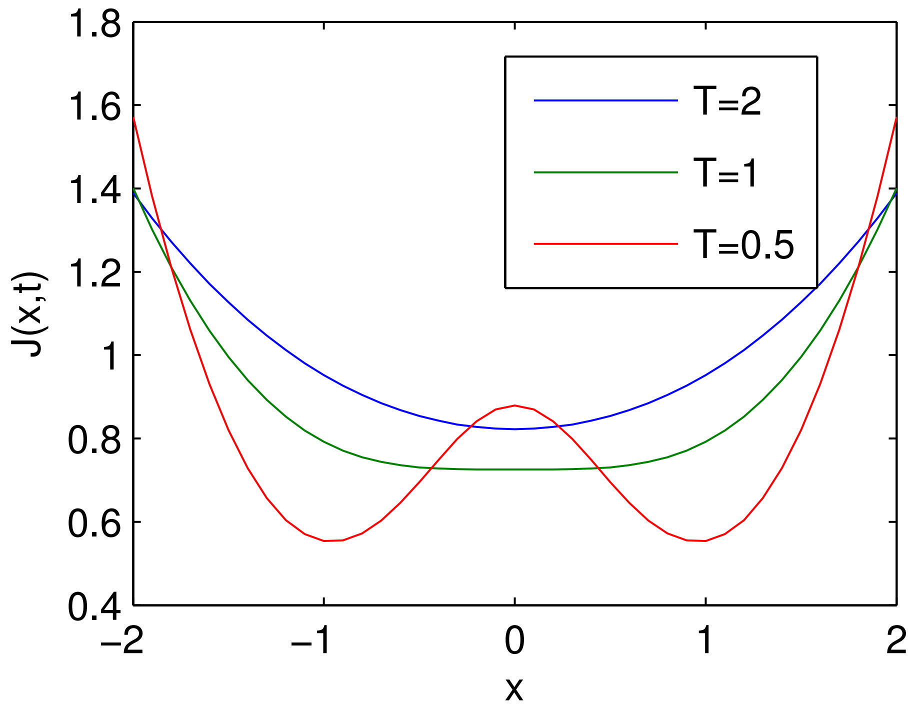
$$J(x,t) = -\nu \log \mathbb{E}_q \exp(-\phi(X_2)/\nu)$$

Decision is made at $T = \frac{1}{\nu}$

Delayed choice

Time-to-go $T = 2 - t$.





$$J(x,t) = -\nu \log \mathbb{E}_q \exp(-\phi(X_2)/\nu)$$

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

Some demonstrations

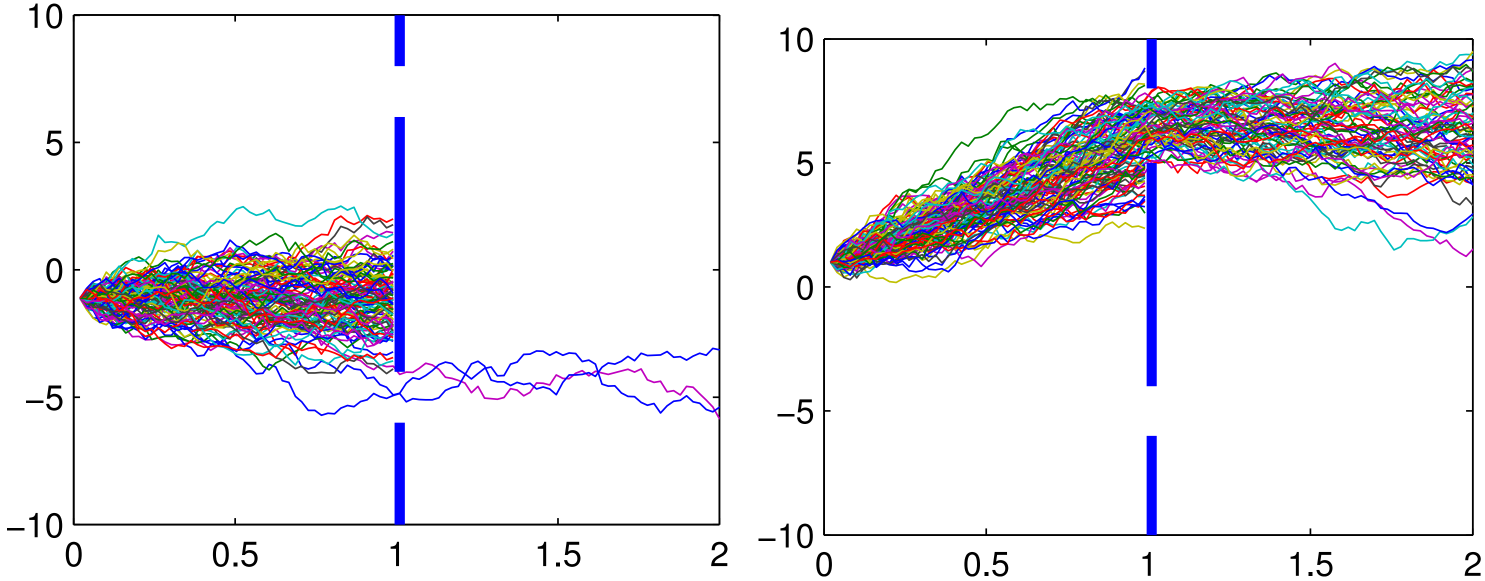
- Coordination of UAV (Gomez et al. 2015)
- Pocket drones (with TUDelft)
- Aggressive driving (Georgia Tech)

”To compute or not to compute, that is the question”

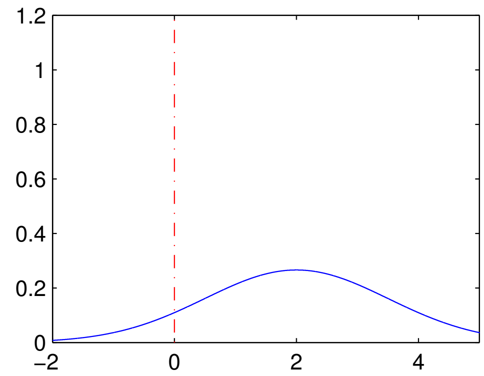
There are two extreme approaches to compute controls:

- precompute $u(x)$ for any possible situation x. Complex to learn and to store. Fast to execute
- compute $u(x)$ for the current situation x. Low learning and storage cost. Slow execution.

A compromise is the idea of importance sampling.



Importance sampling



Consider simple 1-d sampling problem. Given $q(x)$, compute

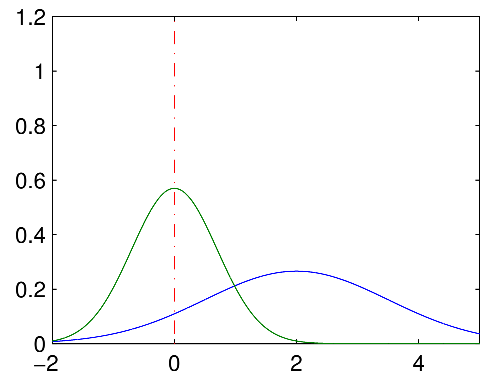
$$a = \mathrm{Prob}(x < 0) = \int_{-\infty}^{\infty} I(x)q(x)dx$$

with $I(x) = 0, 1$ if $x > 0, x < 0$, respectively.

Naive method: generate N samples $X_i \sim q$

$$\hat{a} = \frac{1}{N}\sum_{i=1}^{N} I(X_i) \qquad \mathbb{E}\hat{a} = a \qquad Var(\hat{a}) = \frac{1}{N}Var(I)$$

Importance sampling



Consider another distribution $p(x)$. Then

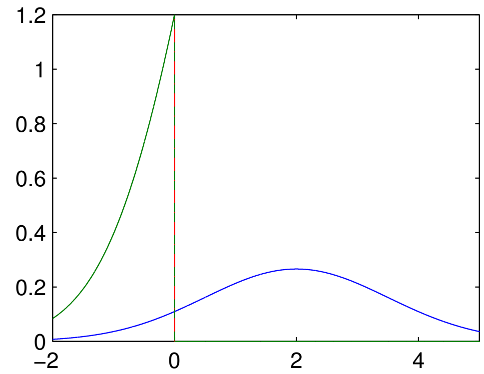
$$a = \mathrm{Prob}(x < 0) = \int_{-\infty}^{\infty} I(x)\frac{q(x)}{p(x)}p(x)dx$$

Importance sampling: generate N samples $X_i \sim p$

$$\hat{a} = \frac{1}{N}\sum_{i=1}^{N} I(X_i)\frac{q(X_i)}{p(X_i)} \qquad \mathbb{E}\hat{a} = a \qquad Var(\hat{a}) = \frac{1}{N}Var\left(I\frac{p}{q}\right)$$

Unbiased (= correct) for any p

Optimal importance sampling



The distribution

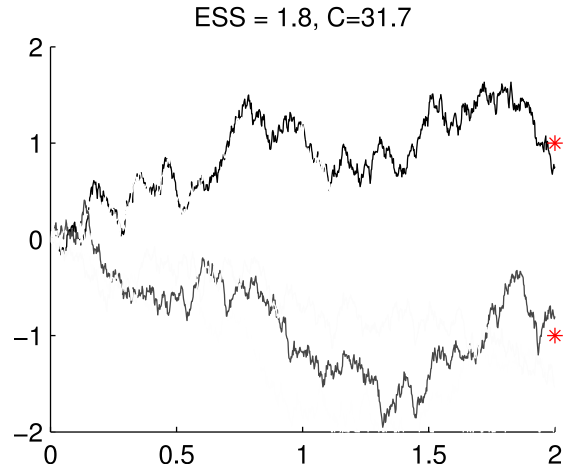
$$p^*(x) = \frac{q(x)I(x)}{a}$$

is the optimal importance sampler.

One sample $X \sim p^*$ is sufficient to estimate a:

$$\hat{a} = I(X)\frac{q(X)}{p^*(X)} = a \qquad \mathbb{E}\hat{a} = a \qquad Var(\hat{a}) = 0$$

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



Sample N trajectories from uncontrolled dynamics

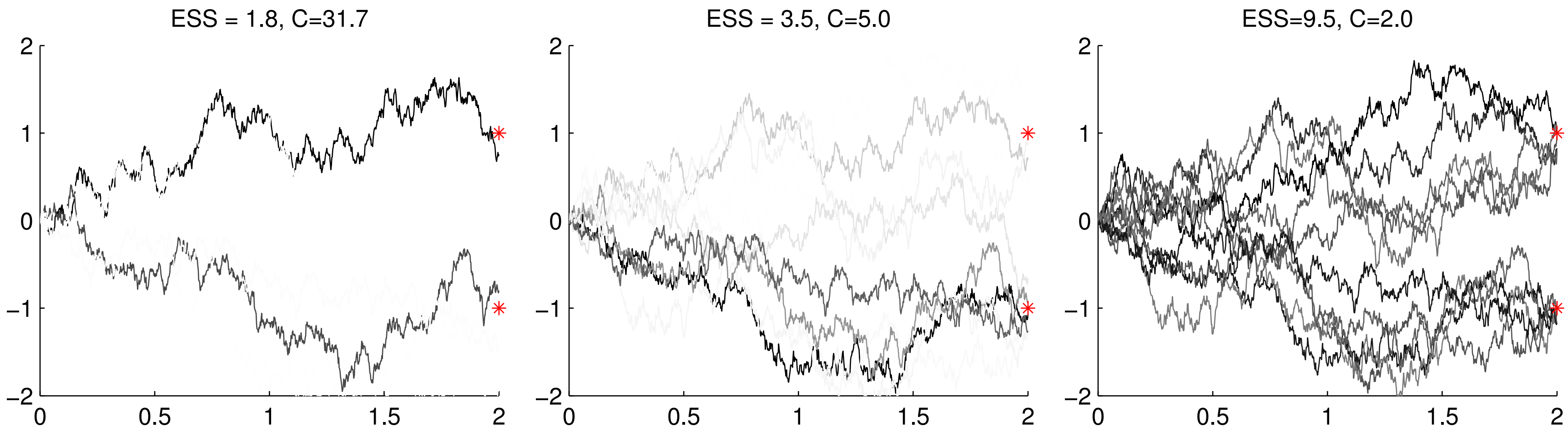
$$\tau_i \quad \sim \quad q(\tau) \qquad w_i = e^{-S(\tau_i)} \qquad \hat{\psi} = \frac{1}{N}\sum_i w_i$$

$\hat{\psi}$ unbiased estimate of ψ.

Effective sample size quantifies sampling efficiency

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

Importance sampling

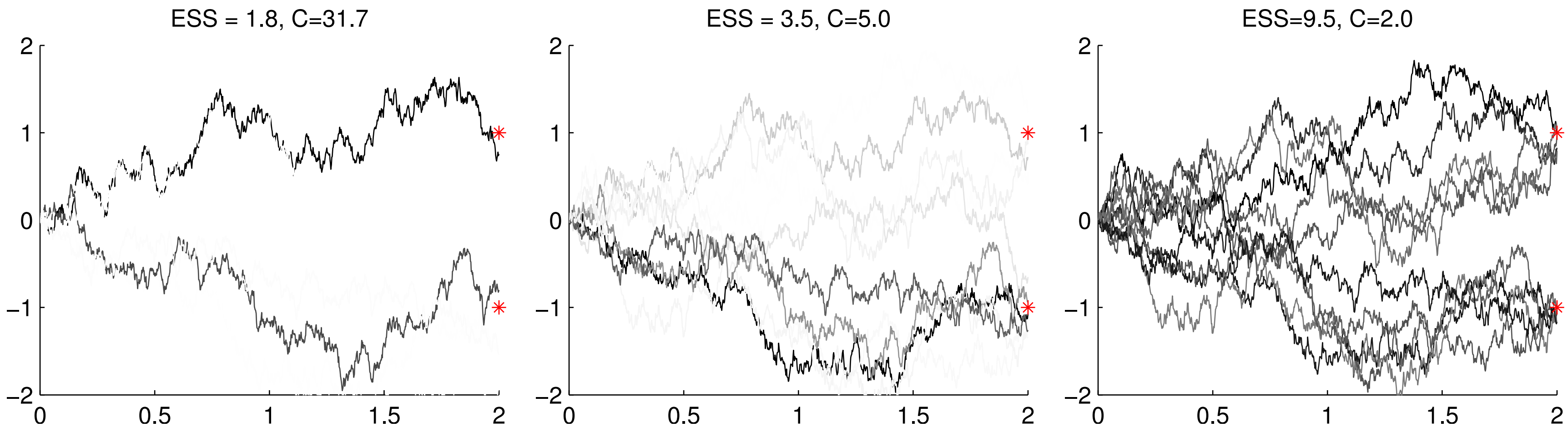


Sample N trajectories from controlled dynamics and reweight yields unbiased estimate of cost-to-go:

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

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

Importance sampling



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

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

Using Jensen's inequality:

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

Proof

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

Using Jensen's inequality:

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

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 $p = p^*$ [1].

[1] 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 $p_u(\tau)$, where $u(x, t|\theta)$ is a parametrized control function.

Following the Cross Entropy method, we minimise $KL(p^*|p_u)$.

$$\Delta\theta \propto -\frac{\partial KL(p^*|p_u)}{\partial\theta} \propto -\mathbb{E}_u e^{-S_u} \int_0^T dW_t \frac{\partial u(X_t, t|\theta)}{\partial\theta}$$

$u(x, t|\theta)$ is arbitrary.

Estimate gradient by sampling.

Kappen, Ruiz 2016

Adaptive importance sampling

for $k = 0, \ldots$ **do**

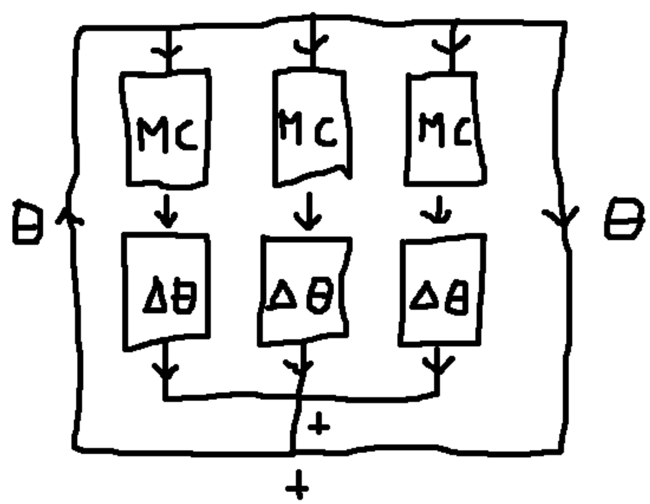
$data_k$ = generate_data($model, u_k$) % Importance sampler

u_{k+1} = learn_control($data_k, u_k$) % Gradient descent

end for

In each iteration we estimate the same control, but more accurately.

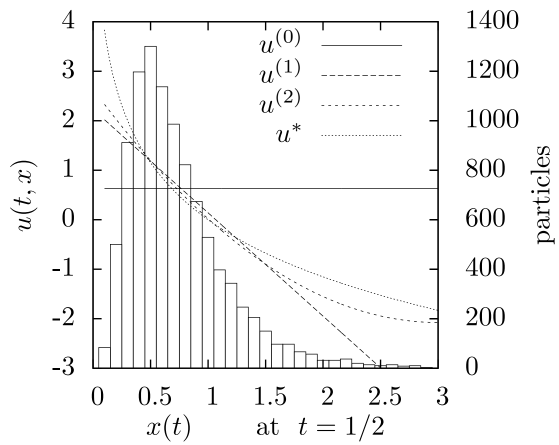
Parallel sampling
Parallel gradient computation



Example

Geometric Brownian motion on the interval $t = 0$ to T.

$$
\begin{aligned}
dX_t &= X_t \left(u(tX_t, t)dt + dW_t \right), \\
C &= \mathbb{E}\, \frac{1}{2}(\log X_T)^2 + \int_0^T \frac{1}{2} u(x,t)^2 \\
u(x,t) &= a(t) + b(t)x + c(t)x^2
\end{aligned}
$$



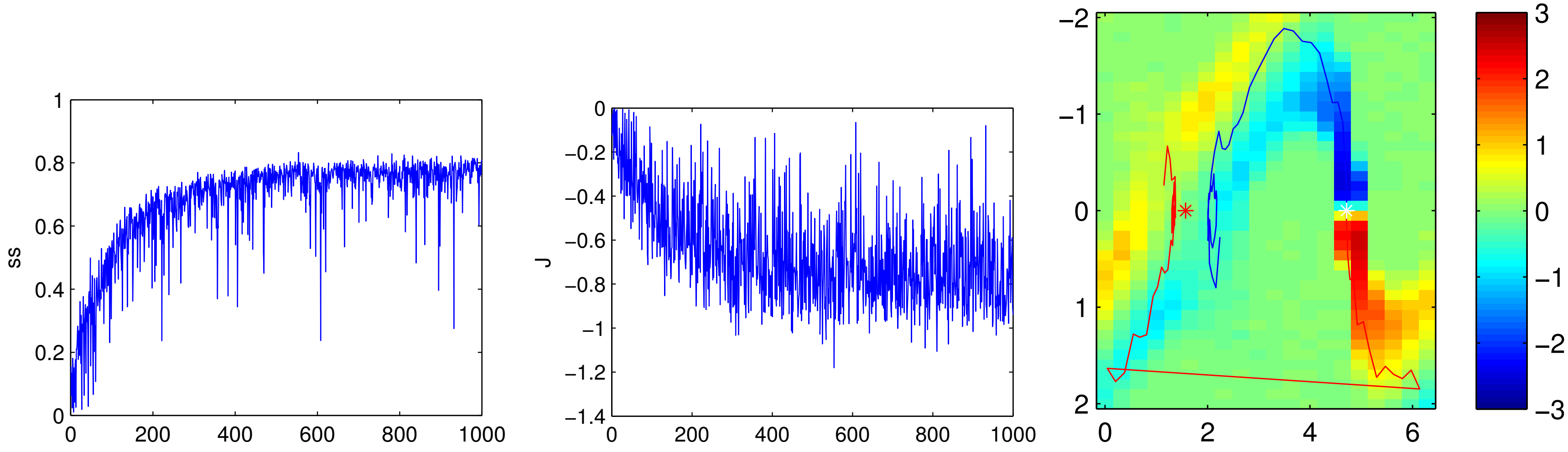
	$u = 0$	constant	linear	quadratic	optimal
C	7.526	5.139	1.507	1.461	1.420
FES(%)	34.3	42.08	87.5	95.2	99.3

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 $u(x,t) = u(x)$.

‘

Integrating perception, control and learning

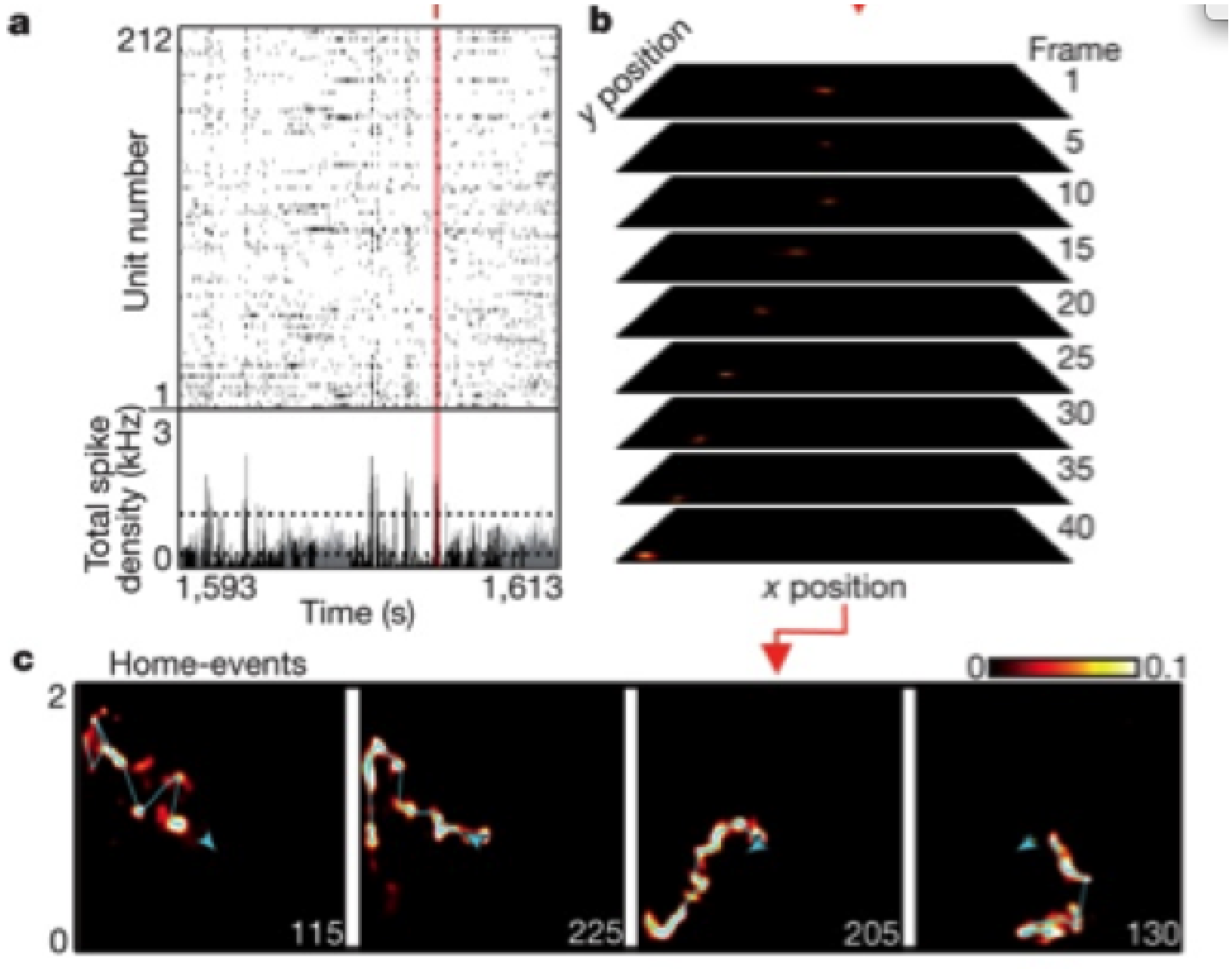
Path integral control theory suggest that controls can be computed by forward simulation in a world model.

Monte Carlo sampling for

- Perception: Bayesian posterior computation combining sensory data and prior world model
- Planning: simulate future trajectories in the world model
- Learning:
  - improve the sampler/controller from these samples
  - improve the world model

This provides an abstract model of what neural computation in the brain is.

Computing control by mental simulation



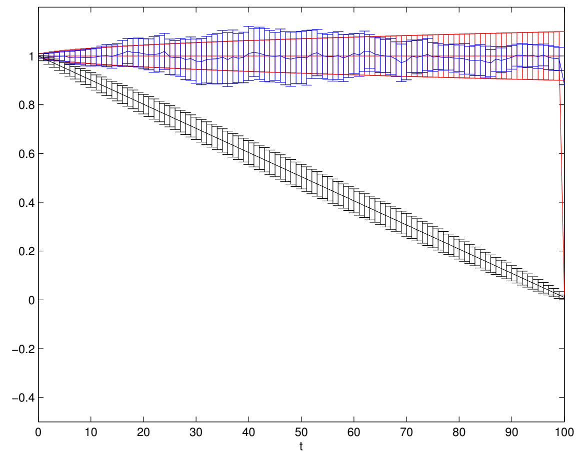
Pfeiffer & Foster (Nature 2013).

Time series inference

Prior process $p(x_{1:T}|x_0)$:

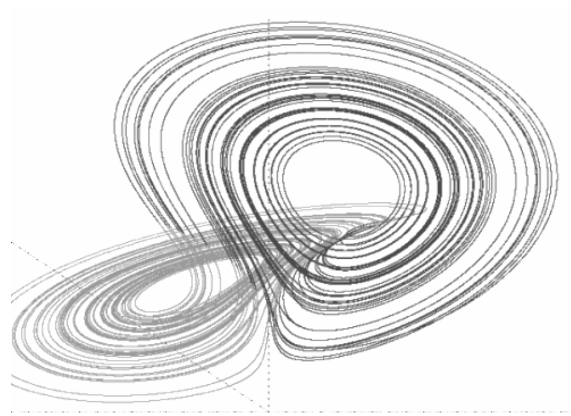
$$dX_t = dW_t \qquad x_0 = 1$$

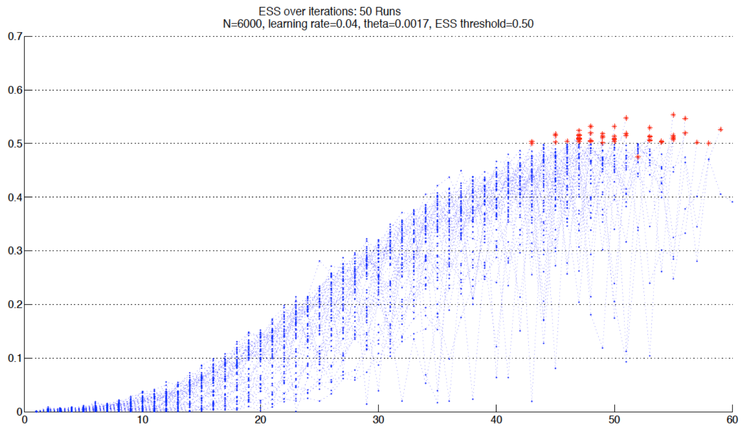
Observation at end time only: $p(y_T|x_T) = \exp(-\beta x_T^2)$

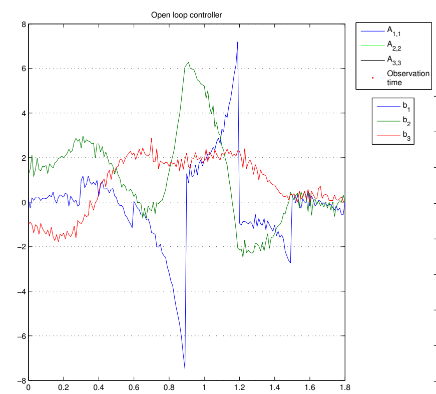


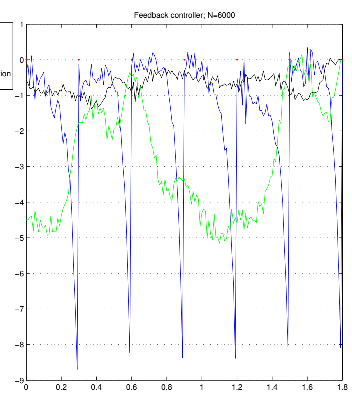
Sampling efficiently from the posterior distribution becomes a control problem.

Controlled noisy Lorenz attractor



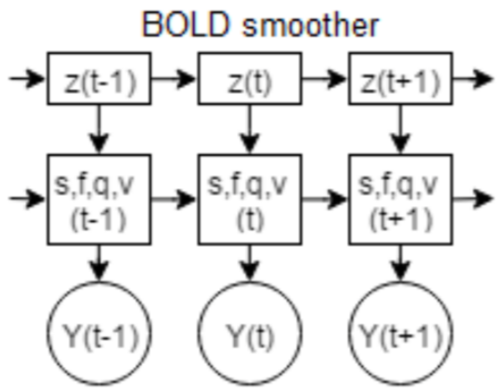


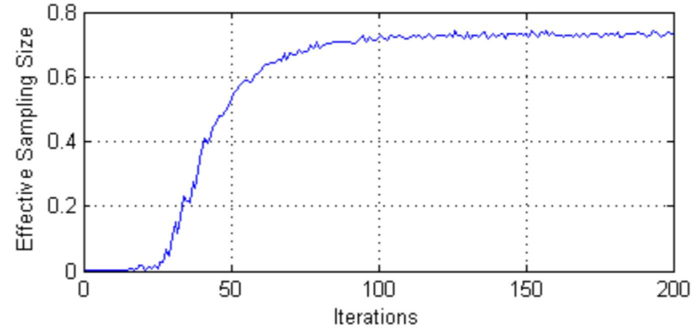


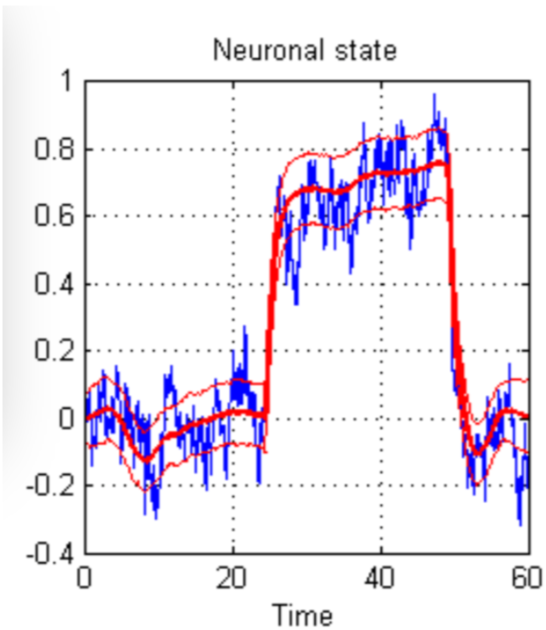


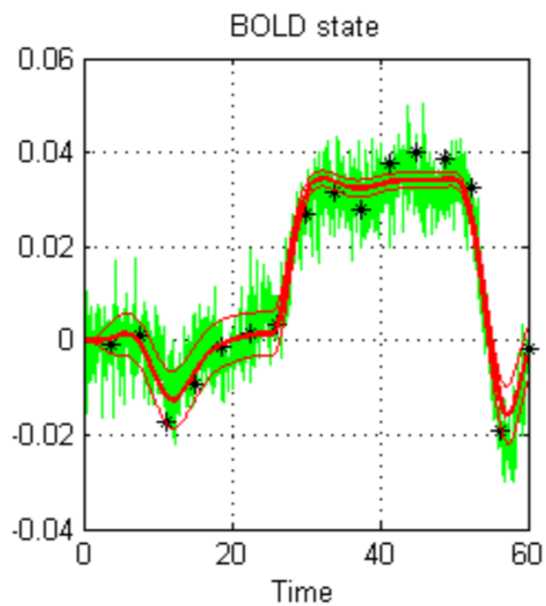
$$u(t, x) = A(t)x + b(t).\ N = 6000$$

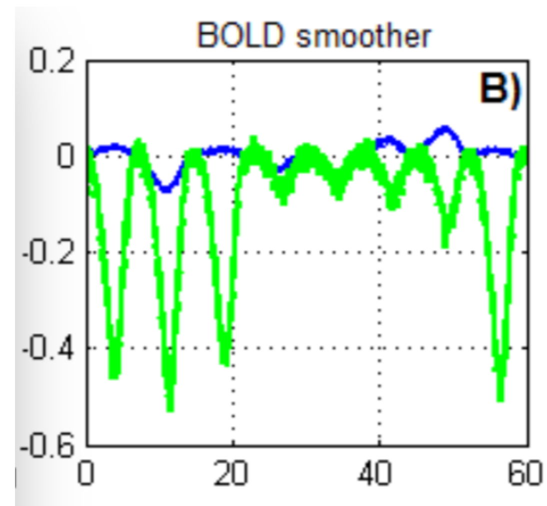
Neural activity from BOLD











$u(z,t) = a(t)z + b(t), N = 5000, K = 200$ iterations.

Summary

Path integral control is a class of control problems where the optimal control can be computed by MC sampling.

- It yields state of the art results for challenging non-linear, noisy, real-time control problems.

- It relates control theory (cost-to-go) and statistical physics (partition sum) and displays phase transitions

The sampling efficiency can be improved by importance sampling, which takes the form of an adaptive controller.

Optimal control and optimal sampling are related:

- better controllers are better samplers

- optimal controllers are optimal samplers

Iterative importance sampling has bootstrapping problem: poor initial controller yields poor samples yields poor controller ...

PI control improves particle filtering methods for time series smoothing problems.

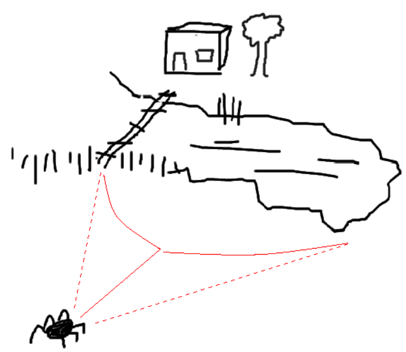
Thank you!

Kappen, Hilbert J. ”Linear theory for control of nonlinear stochastic systems.” Physical review letters 95.20 (2005): 200201.

S. Thijssen and H. J. Kappen. ”Path Integral Control and State Dependent Feedback.” Phys. Rev. E 91, 032104 2015

Kappen, Hilbert Johan, and Hans Christian Ruiz. ”Adaptive importance sampling for control and inference.” Journal of Statistical Physics 162.5 (2016): 1244-1266.

Ruiz, Hans-Christian, and Hilbert J. Kappen. ”Particle Smoothing for Hidden Diffusion Processes: Adaptive Path Integral Smoother.” IEEE Transactions on Signal Processing 65.12 (2017): 3191-3203.



www.snn.ru.nl/~bertk