Path integral control

Minimization wrt u yields: [11]

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

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.

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

Path integral control

Then the HJB becomes *linear* in ψ

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

with end condition $\psi(x,T) = \exp(-\phi(x)/\lambda)$ [12]

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

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

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

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

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

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

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

can be interpreted as a Kolmogorov backward equation for the process

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

The correspondong forward equation is

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

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:

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

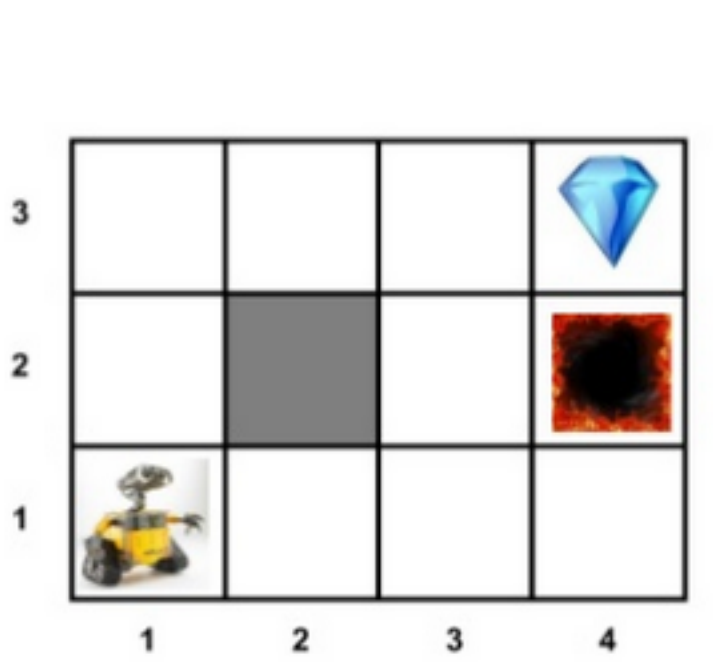
with τ a trajectory $x(t \to T)$. Then

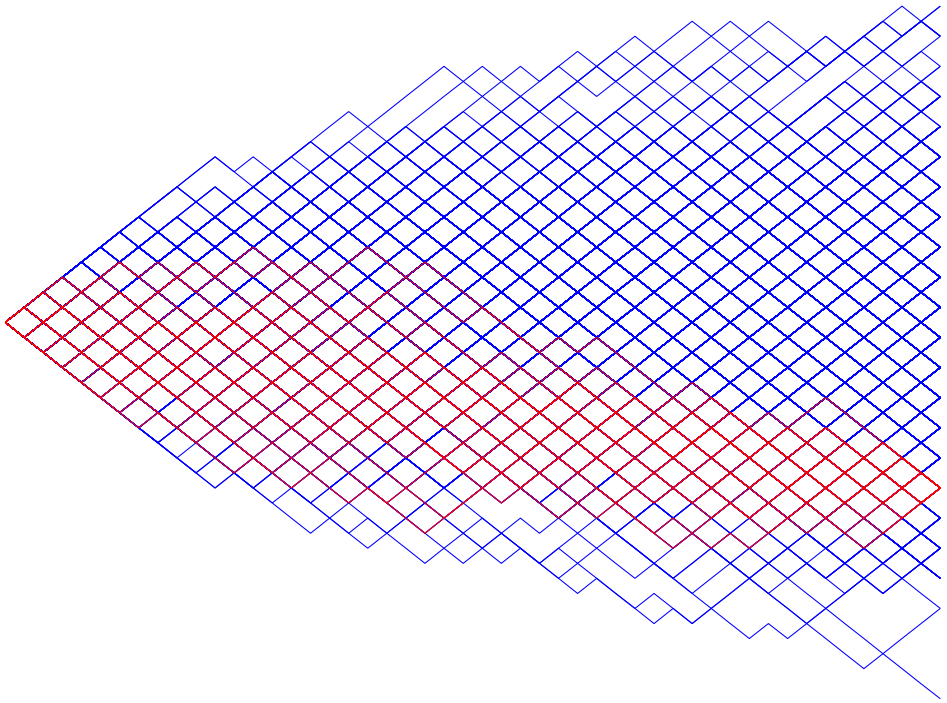
$$\psi(x,t) = \int dQ(\tau|x,t) \exp\left(-\frac{S(\tau)}{\lambda}\right)$$

$$S(\tau) = \phi(x(T)) + \int_t' dsV(x(s),s)$$

ψ can be computed by forward sampling the uncontrolled process.

Alternative derivation





Uncontrolled dynamics specifies distribution $q(\tau|x,t)$ over trajectories τ from x,t.

Cost for trajectory τ is $S(\tau|x,t) = \phi(x_T) + \int_t^{'} dsV(x_s,s)$.

Find optimal distribution $p(\tau|x,t)$ that minimizes $\mathbb{E}_p\ S$ and is 'close' to $q(\tau|x,t)$.

KL control

Find p^* that minimizes

$$C(p) = KL(p|q) + \mathbb{E}_p\, S \qquad KL(p|q) = \int d\tau p(\tau|x,t) \log \frac{p(\tau|x,t)}{q(\tau|x,t)}$$

The optimal solution is given by

$$\begin{aligned} p^*(\tau|x,t) &= \frac{1}{\psi(x,t)} q(\tau|x,t) \exp(-S(\tau|x,t)) \\ \psi(x,t) &= \int d\tau q(\tau|x,t) \exp(-S(\tau|x,t)) = \mathbb{E}_q e^{-S} \end{aligned}$$

The optimal cost is:

$$C(p^*) = -\log \psi(x,t)$$

Controlled diffusions

In the case of controlled diffusions, $p(\tau|x,t)$ is parametrised by functions $u(x,t)$, $q(\tau|x,t)$ corresponds to $u(x.t) = 0$:

$$\begin{aligned} dX_t &= f(X_t,t)dt + g(X_t,t)(u(X_t,t)dt + dW_t) \qquad \mathbb{E}(dW_i dW_j) = \nu_{ij}dt \\ C(p) &= \mathbb{E}_p\left(\int dt \frac{1}{2} u(X_t,t)'\nu^{-1}u(X_t,t) + S(\tau|x,t)\right) \end{aligned}$$

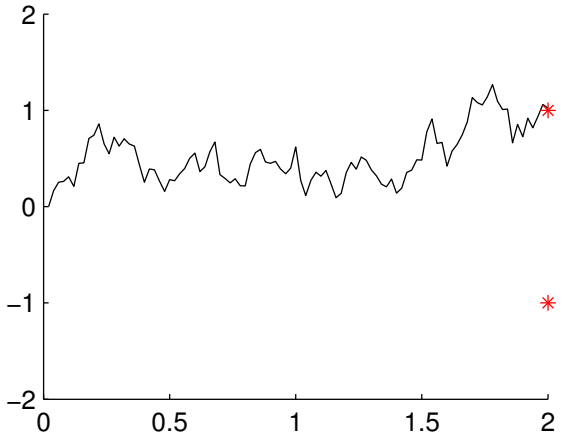
$J(x,t) = -\log\psi(x,t)$ is the solution of the Bellman equation.

p^* is generated by optimal control $u^*(x,t)$:

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

ψ, u^* can be computed by forward sampling from q.

Recap of the main idea



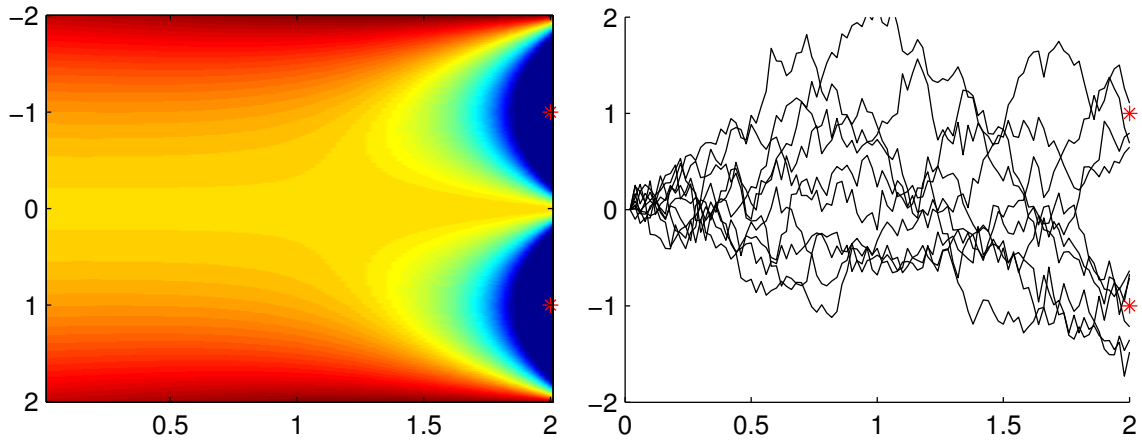
Consider a stochastic dynamical system

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

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.

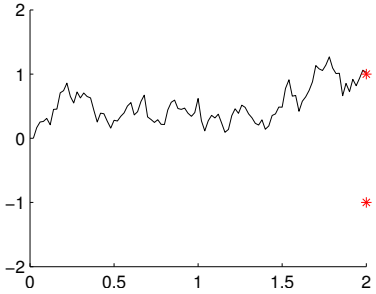
$$J(x,t) = \min u_{t:T} \mathbb{E}_u \left(\phi(X_T) + \int_t^T dt R(X_t, u(X_t,t)) \right) \qquad X_t = x$$

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



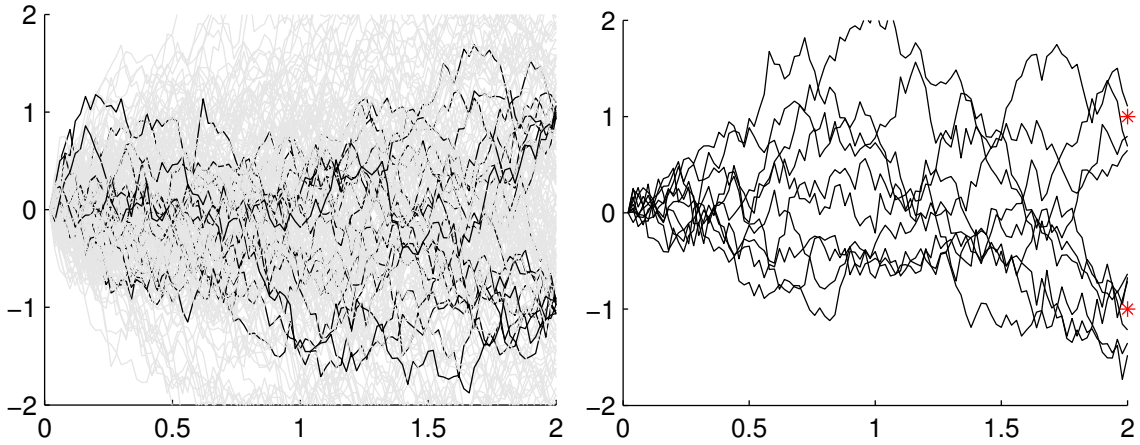
$$dX_t = \underbrace{f(X_t)dt + g(X_t)(u(X_t,t)dt}_{f(X_t,u)dt} + dW_t) \qquad X_0 = x_0$$

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

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

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

Path integral control theory



Equivalent formulation: Find distribution over trajectories p that minimizes [13]

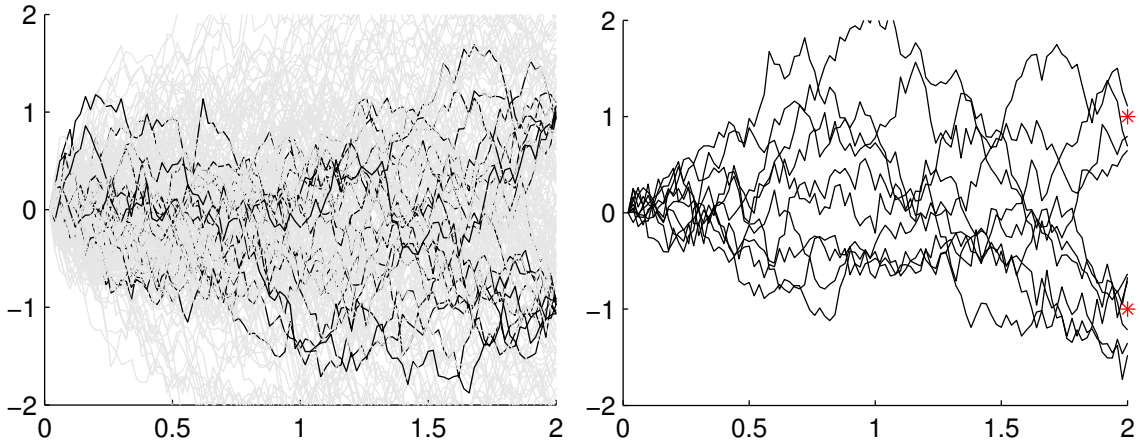
$$C(p) = \int d\tau p(\tau) \left(S(\tau) + \log \frac{p(\tau)}{q(\tau)} \right)$$

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

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

[13] $\mathbb{E}_u \int_0^T dt \frac{1}{2} u(X_t, t)^2 = \int d\tau p(\tau) \log \frac{p(\tau)}{q(\tau)}$.

Path integral control theory



Equivalent formulation: Find distribution over trajectories p that minimizes

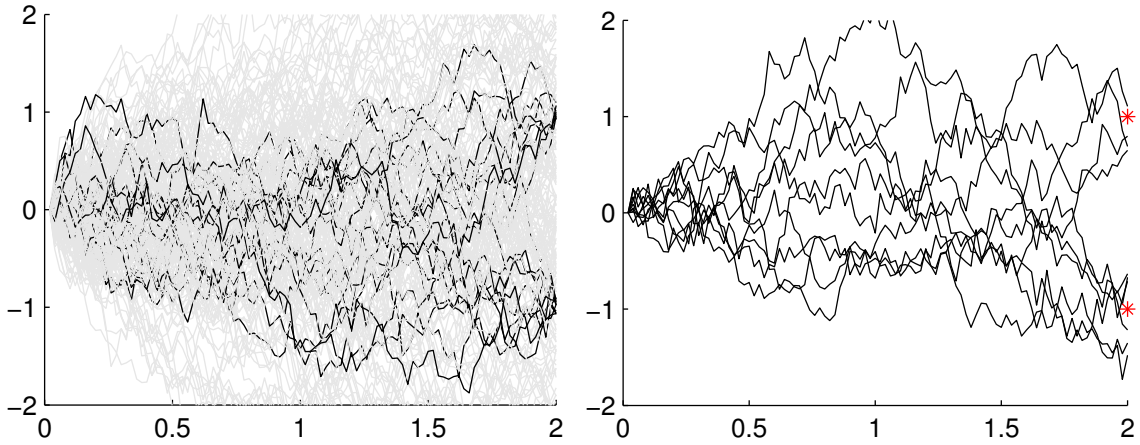
$$C(p) = \int d\tau p(\tau) \left(S(\tau) + \log \frac{p(\tau)}{q(\tau)} \right)$$

$q(\tau|x_0, 0)$ 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



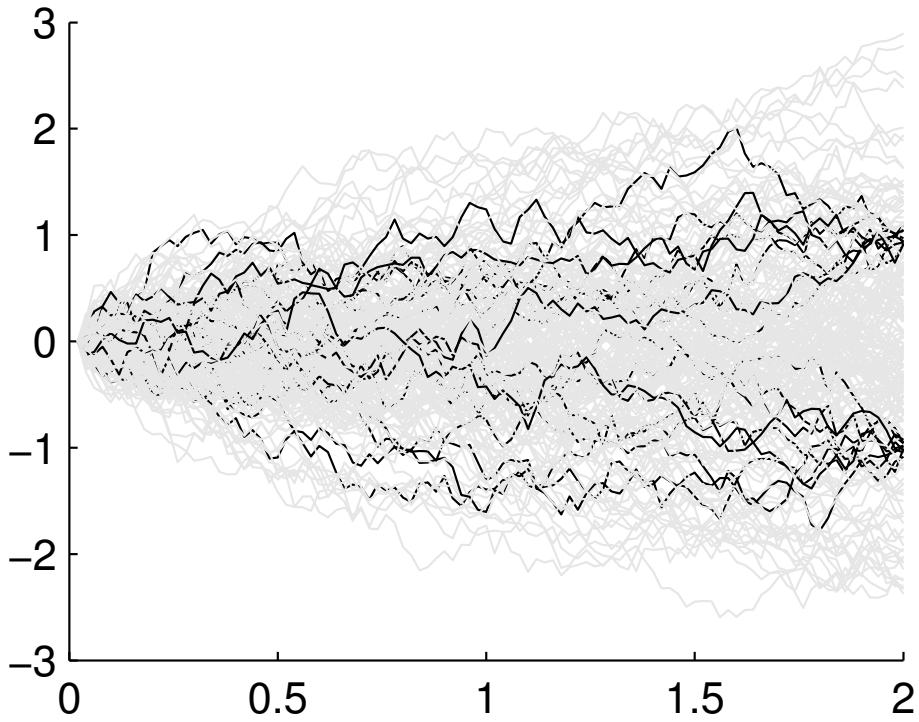
The optimal control cost is $C(p^*) = -\log \psi = J(x_0, 0)$ with

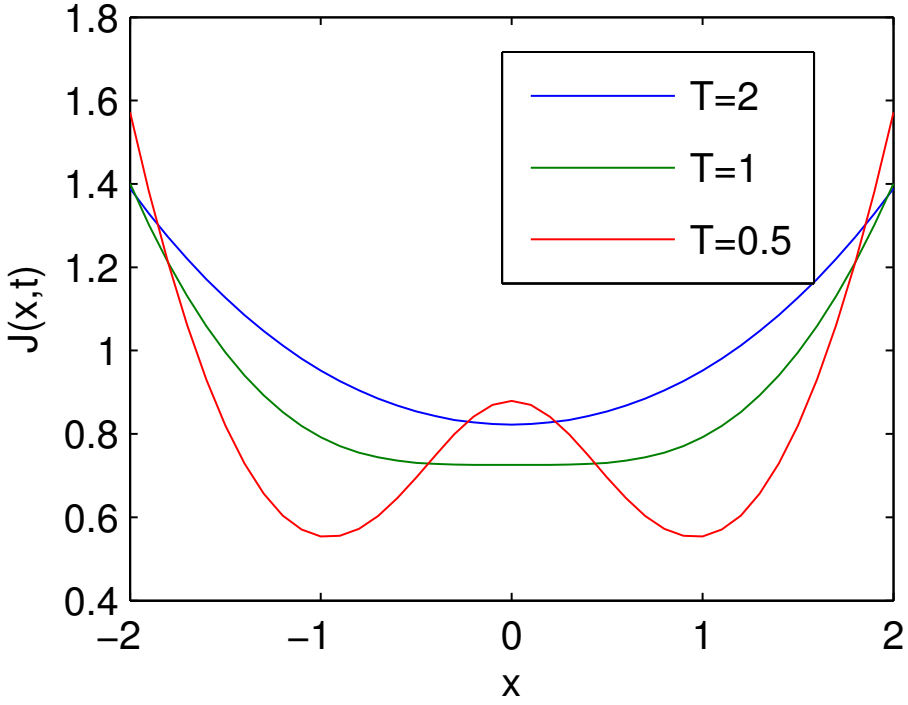
$$\psi = \int d\tau q(\tau) e^{-S(\tau)} = \mathbb{E}_q e^{-S}$$

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

Delayed choice

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



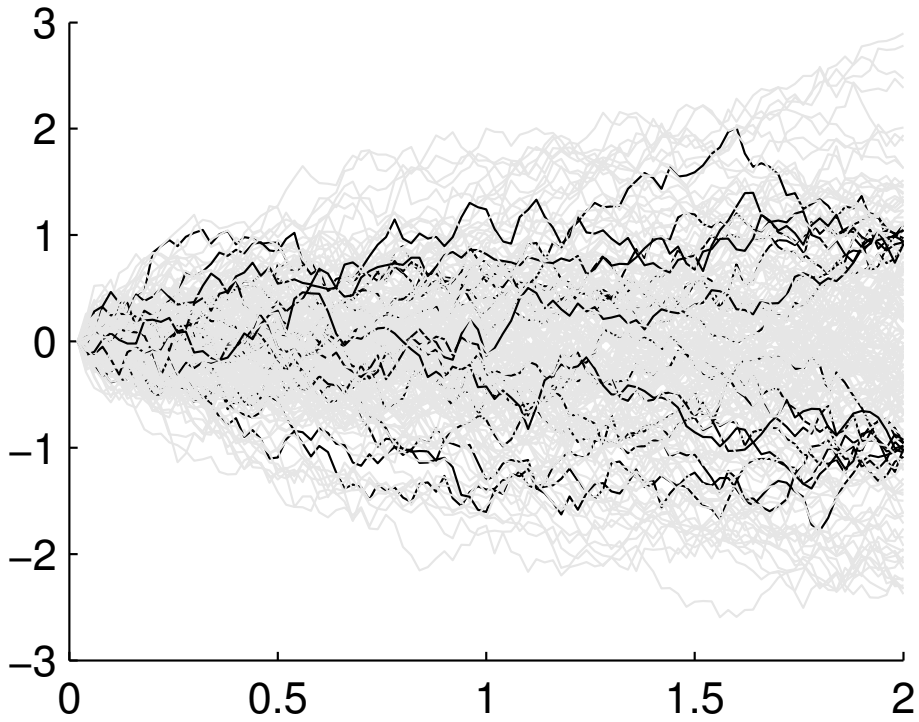


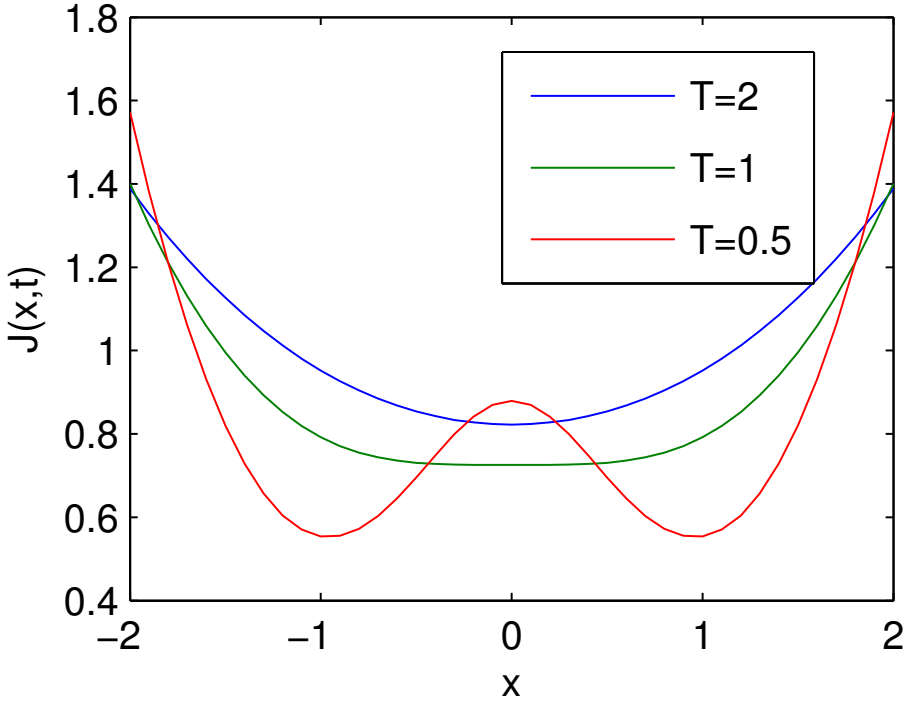
$$J(x,t) = -\nu \log \mathbb{E}_q \exp(-\phi(X_2)/\nu)$$

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."


JUV 0-1 ASR 48:31
VISIT JUVENTUS.COM
25 J. HOLEBAS
C. MARCHISIO 8
1:01 / 9:41

Delayed choice (details)

$$dX_t = udt + dW_t \qquad \mathbb{E}dW_t^2 = \nu dt$$

$V = 0$, path cost is $\frac{1}{2}u^2$ and end cost $\phi(z = \pm 1) = 0, \phi(z) = \infty$ else encodes two targets at $z = \pm 1$ at $t = T$.

PI recipe:

1.

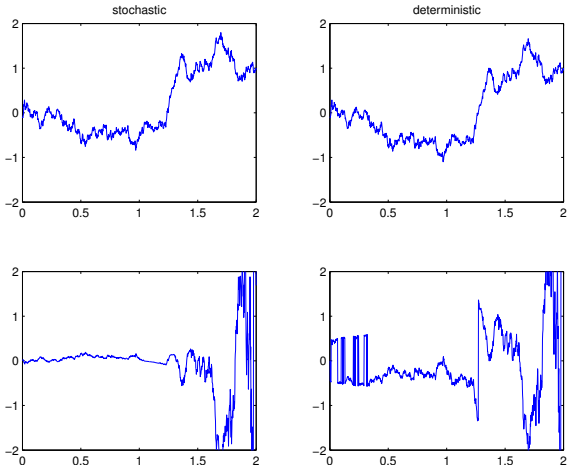
$$\begin{aligned}
\psi(x,t) &= \int dQ(\tau|x,t)\exp(-S(\tau)/\lambda) \\
S(\tau) &= \phi(x(T)) \\
\psi(x,t) &= \int dz q(z,T|x,t)\exp(-\phi(z)/\lambda) = q(1,T|x,t) + q(-1,T|x,t) \\
q(z,T|x,t) &= \mathcal{N}(z|x,\nu(T-t))
\end{aligned}$$

2. Compute

$$J(x,t) = -\lambda \log \psi(x,t) = \frac{1}{T-t}\left(\frac{1}{2}x^2 - \nu(T-t)\log 2\cosh\frac{x}{\nu(T-t)}\right)$$

3.

$$u(x,t) \quad = \quad -\nabla J(x,t) = \frac{1}{T-t}\left(\tanh\frac{x}{\nu(T-t)} - x\right)$$



Coordination of UAVs

(AAMAS 2015.mp4)

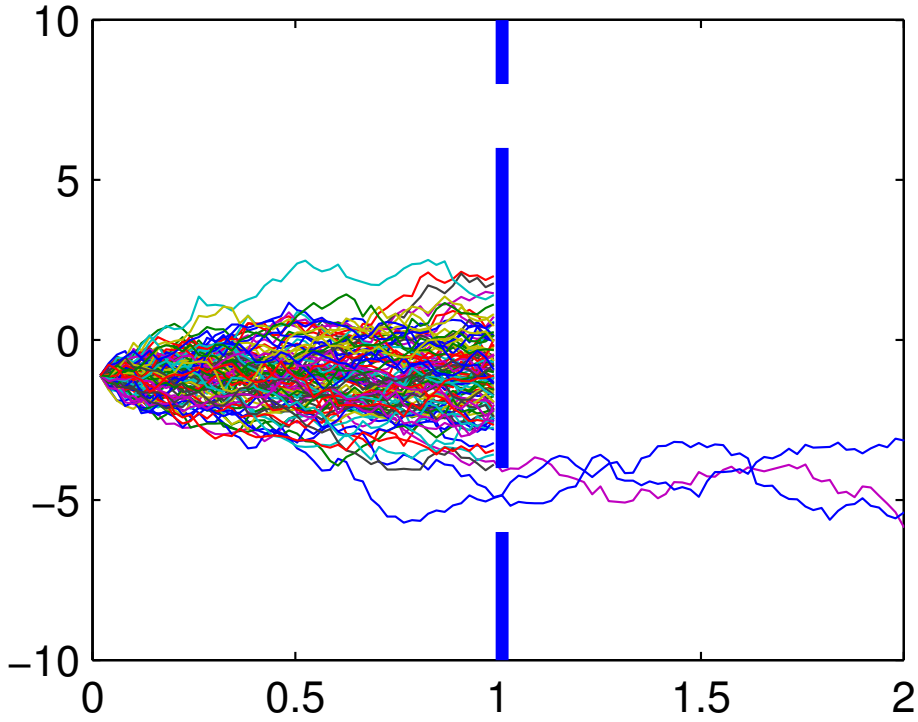
≈ 10.000 trajectories per iteration, 3 iterations per second.
Video at: `http://www.snn.ru.nl/~bertk/control_theory/PI_quadrotors.mp4`

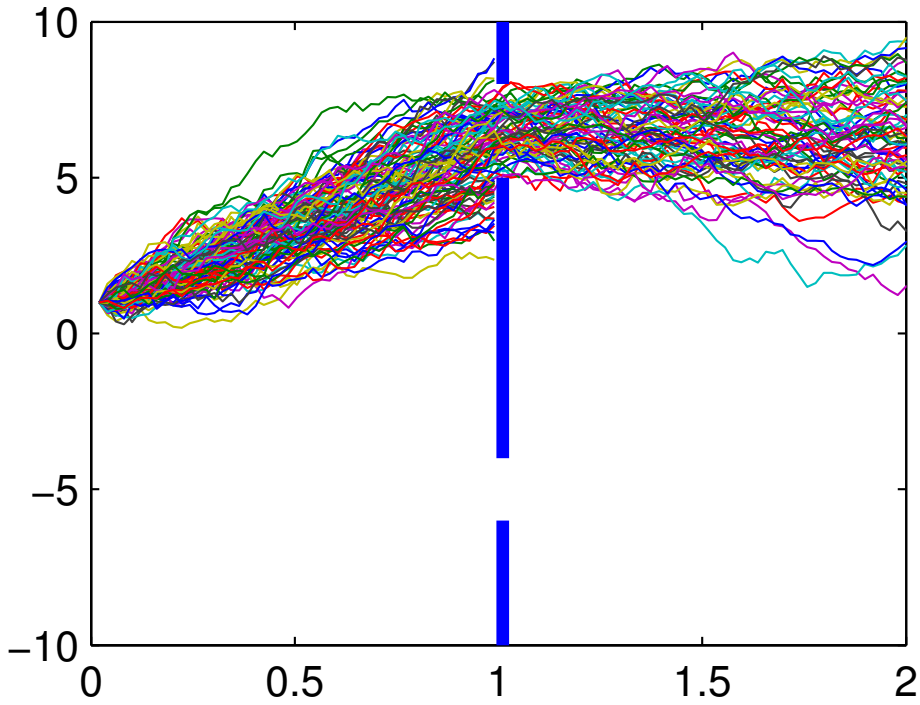
Gomez et al. 2015

Coordination of UAVs

Chao Xu ACC 2017

Importance sampling and control





$$\psi(x,t) = \mathbb{E}_q e^{-S} \qquad S(\tau|x,t) = \phi(x_T) + \int_t^T dsV(x_s,s)$$

Sampling is 'correct' but inefficient.

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

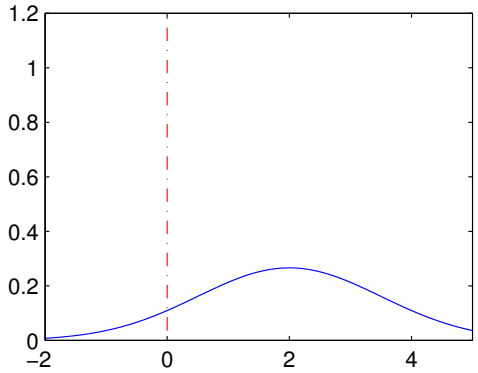
There are two extreme approaches to compute actions:

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

Intuitively, one can imagine that the most efficient approach is to combine both ideas (like ’just-in-time’ manufacturing):

- precompute ’basic motor skills’, the ’halffabrikaat’
- compute the appropriate action $u(x)$ from the basic motor skills

Importance sampling



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

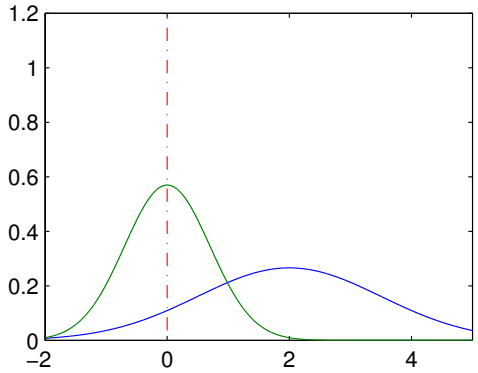
$$a = \text{Prob}(x < 0) = \int_{-\infty}^{\infty} I(x)q(x)dx$$

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

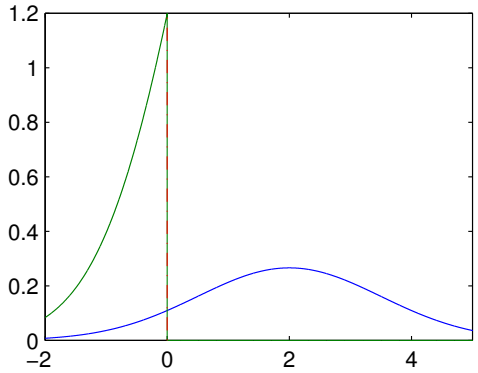
$$a = \text{Prob}(x < 0) = \int_{-\infty}^{\infty} I(x) \frac{q(x)}{p(x)} p(x) dx$$

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

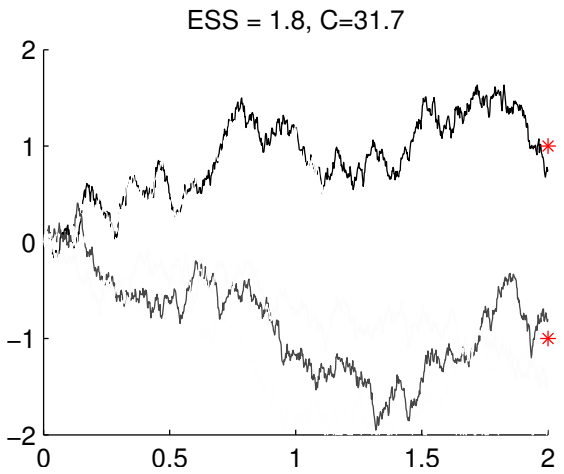
$$p^*(x) = \frac{q(x)I(x)}{a}$$

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

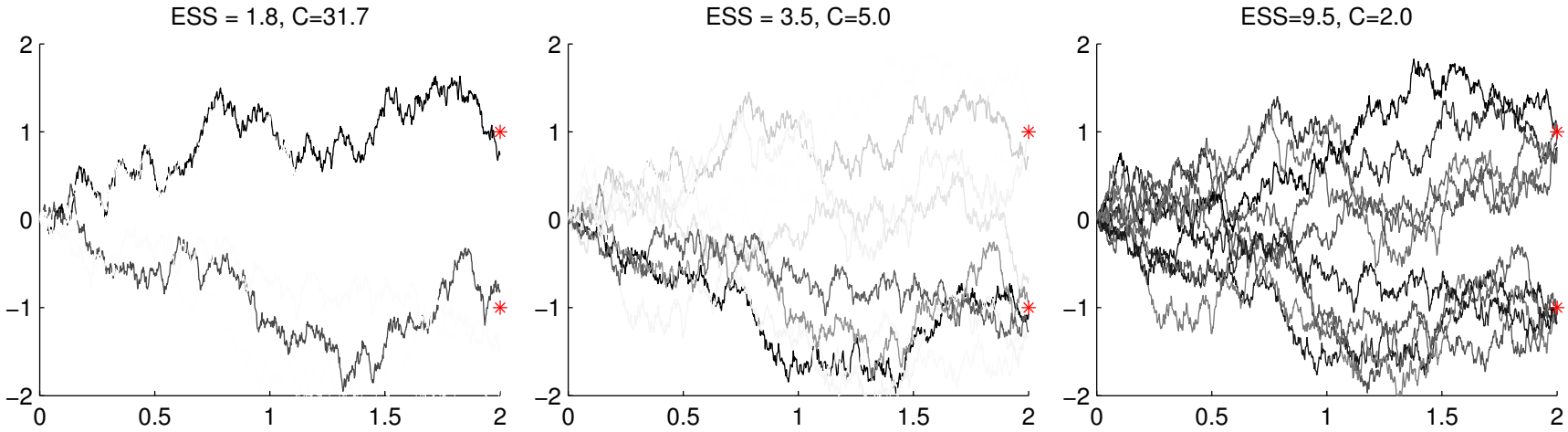
$$\tau_i \quad \sim \quad q(\tau) \qquad w_i = e^{-S(\tau_i)} \qquad \hat{\psi} = \frac{1}{N}\sum_i w_i$$

$\hat{\psi}$ unbiased estimate of ψ.

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

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

Importance sampling

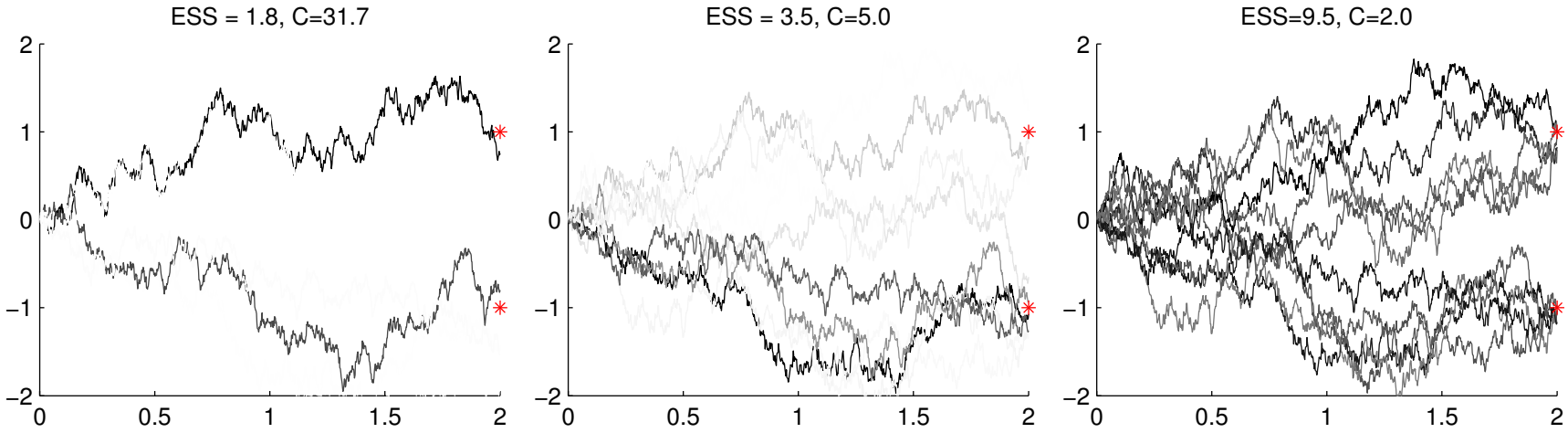


Sample N trajectories from controlled dynamics and reweight yields unbiased estimate of cost-to-go:

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

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

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

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 $p = p^*$ [14].

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

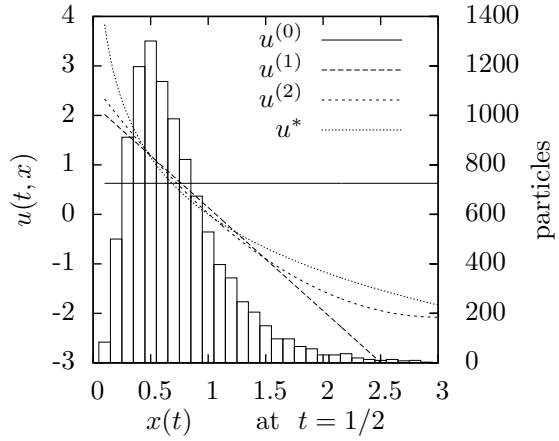
Example

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

$$dX_t = X_t\left(u(tX_t, t)dt + dW_t\right),$$

$$C = \mathbb{E}\,\frac{1}{2}\log(X_T)^2$$

$$u(x,t) = a(t) + b(t)x + c(t)x^2$$



	$u = 0$	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

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.

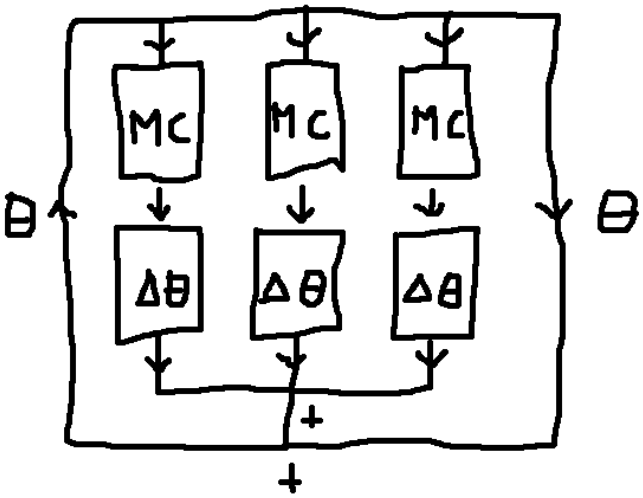
Estimate gradient by sampling.

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

Parallel sampling
Parallel gradient computation

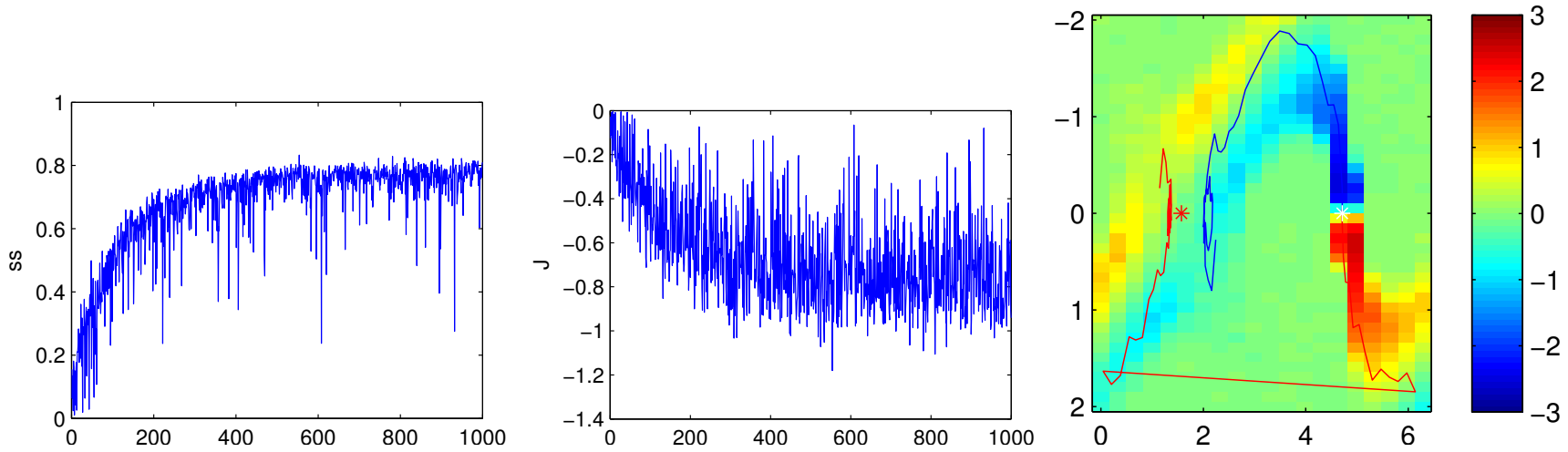


Inverted pendulum

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

Illustration of gradient descent learning Eq. **??** for a second order inverted pendulum problem. Left: Entropic sample size versus importance sampling iteration. Middle: Optimal cost to go versus importance sampling iteration. Right: Optimal control solution $\hat{u}(x_1, x_2)$ versus x_1, x_2 with $0 \leq x_1 \leq 2\pi$ and $-2 \leq x_2 \leq 2$.

Acrobot

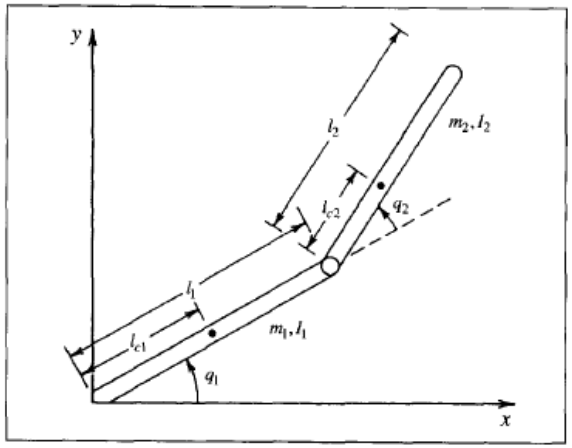


Fig. 1. The Acrobot.

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

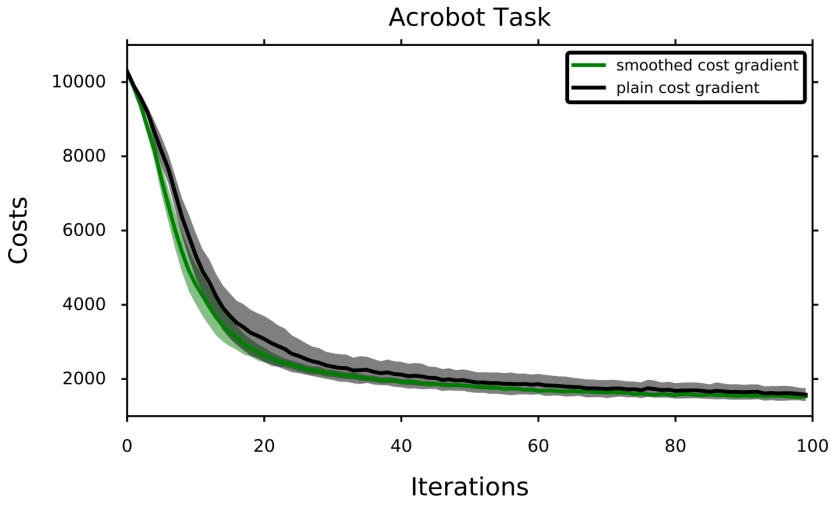
$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + h_2 + \phi_2 = \tau, \quad (2)$$

where

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

2 DOF, second order, under actuated, continuous stochastic control problem.
Task is swing-up from down position.

(acrobot.mp4)



Neural network 10 layers, 25 neurons per layer. Input is sin and cosine of both angles as well as angular velocity. No time as input. 100 iterations, with 10000 rollouts per iteration. Annealing such that ESS larger than 10 %. Took around 15 min with 100 cpu.

Acrobot (details)

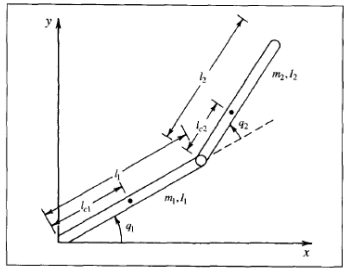


Fig. 1. The Acrobot.

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

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

where

$d_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos(q_2)) + I_1 + I_2$

$d_{22} = m_2 l_{c2}^2 + I_2$

$d_{12} = m_2(l_{c2}^2 + l_1 l_{c2}\cos(q_2)) + I_2$

$d_{21} = m_2(l_{c2}^2 + l_1 l_{c2}\cos(q_2)) + I_2$

$h_1 = -m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2^2 - 2m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2\dot{q}_1$

$h_2 = m_2 l_1 l_{c2}\sin(q_2)\dot{q}_1^2$

$\phi_1 = (m_1 l_{c1} + m_2 l_1)g\cos(q_1) + m_2 l_{c2} g\cos(q_1 + q_2)$

$\phi_2 = m_2 l_{c2} g\cos(q_1 + q_2).$

$q_1(0) = q_2(0) = -\pi/2$, $\dot{q}_1(0) = \dot{q}_2(0) = 0$, maximize final height

$$H = l_1 \sin q_1(T) + l_2 \sin q_2(T)$$

Acrobot (details)

$$d_{11}(q)\ddot{q}_1 + d_{12}(q)\ddot{q}_2 + h_1(q,\dot{q}) + \phi_1(q) = 0$$
$$d_{21}(q)\ddot{q}_1 + d_{22}\ddot{q}_2 + h_2(q,\dot{q}) + \phi_2(q) = u$$

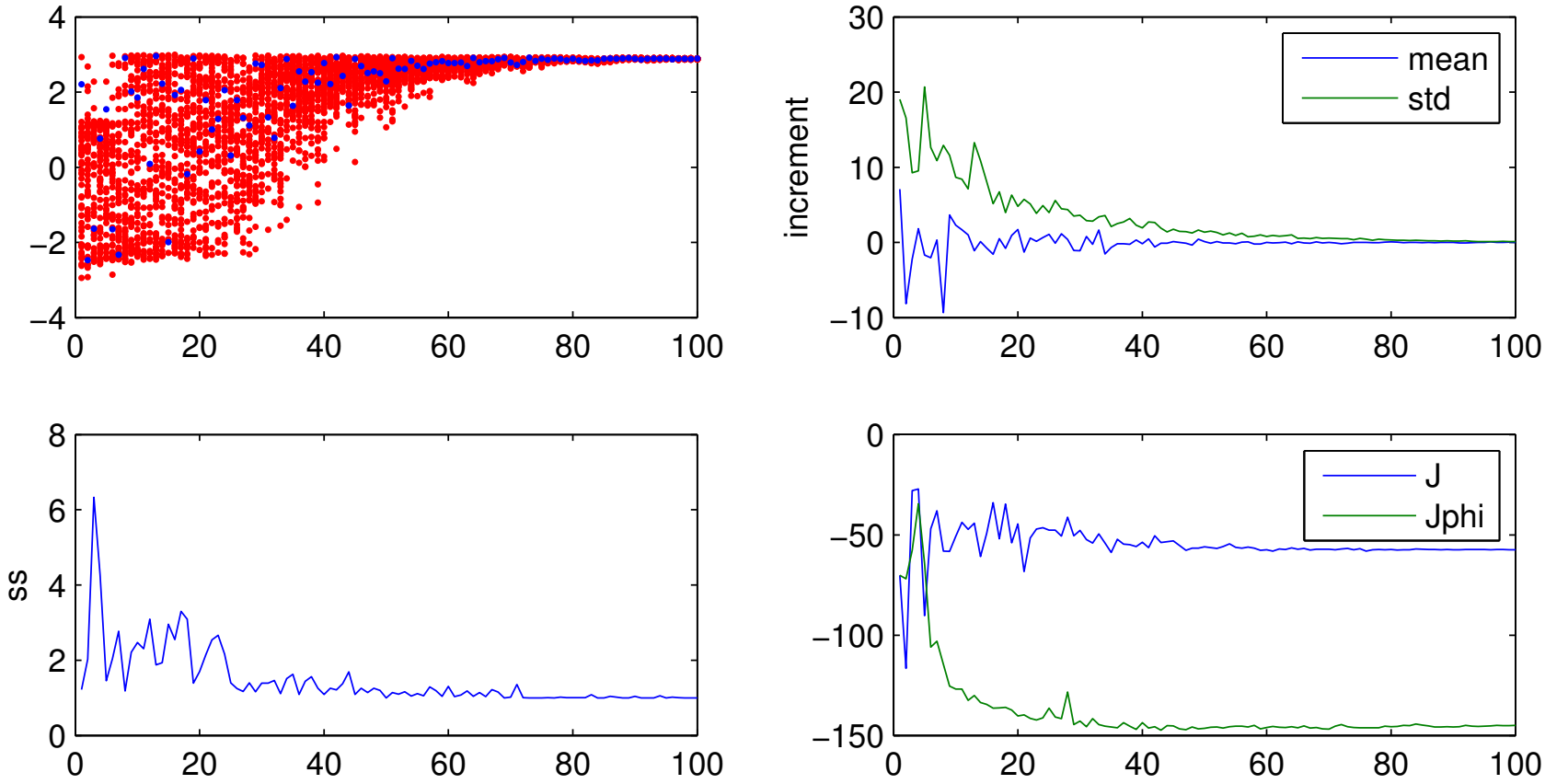
We can write these equations in standard form

$$dx_i = f_i(x)dt + g_i(x)udt$$

with $x_1 = q_1, x_2 = q_2, x_3 = \dot{q}_1, x_4 = \dot{q}_2$ and

$$\begin{aligned} f_1(x) &= x_3 & g_1(x) &= 0 \\ f_2(x) &= x_4 & g_2(x) &= 0 \\ f_3(x) &= \frac{-d_{22}(h_1+\phi_1)+d_{12}(h_2+\phi_2)}{D} & g_3(x) &= -\frac{d_{12}}{D} \\ f_4(x) &= \frac{d_{12}(h_1+\phi_1)-d_{11}(h_2+\phi_2)}{D} & g_4(x) &= \frac{d_{11}}{D} \end{aligned}$$

Acrobot (details)



100 iterations. At each iteration 50 stochastic trajectories were generated. The new control was computed from a deterministic trajectory. Noise was lowered at each iteration. Top left: final height for each stochastic trajectory for each iteration (red) and for each deterministic solution (blue).

Integrated sensorimotor control

Initialize control u_0
for $t = 0, \ldots$ **do**
    $data_t$ =act_in_the_world(u_t)
    $model_t$ = learn_model($u_t, data_t$)
    u_{t+1}= compute_control($model_t$)
end for

compute_control

for $k = 0, \ldots$ **do**
    $data_k$ = generate_data($model, u_k$) % Monte Carlo importance sampler
    u_{k+1} = learn_control($data_k, u_k$) % Deep or recurrent learning
end for

Integrated sensorimotor control

Initialize control u_0
for $t = 0, \ldots$ **do**
  $data_t$ =act_in_the_world(u_t)
  $model_t$ = learn_model($u_t, data_t$)
  u_{t+1}= compute_control($model_t$)
end for

compute_control

for $k = 0, \ldots$ **do**
  $data_k$ = generate_data($model, u_k$) % Monte Carlo importance sampler
  u_{k+1} = learn_control($data_k, u_k$) % Deep or recurrent learning
end for

- generate infinite data to learn infinitely complex

Integrated sensorimotor control

Initialize control u_0
for $t = 0, \ldots$ **do**
  $data_t$ =act_in_the_world(u_t)
  $model_t$ = learn_model($u_t, data_t$)
  u_{t+1}= compute_control($model_t$)
end for

compute_control

for $k = 0, \ldots$ **do**
  $data_k$ = generate_data($model, u_k$) % Monte Carlo importance sampler
  u_{k+1} = learn_control($data_k, u_k$) % Deep or recurrent learning
end for

- generate infinite data to learn infinitely complex
- $data_t$ and $data_k$ are the two realities of the brain

Towards sensorimotor integration

The brain is a Monte Carlo sampler

- Perception: Bayesian posterior computation
- Action: solving an optimal control problem through sampling

Both require the learning of a world model

Towards sensorimotor integration

The brain is a Monte Carlo sampler

- Perception: Bayesian posterior computation
- Action: solving an optimal control problem through sampling

Both require the learning of a world model

Action computation is optimized by adaptive importance sampling,

- this is a type of motor learning
- but is complemented by sampling (’halffabrikaat’)

Towards sensorimotor integration

The brain is a Monte Carlo sampler

- Perception: Bayesian posterior computation
- Action: solving an optimal control problem through sampling

Both require the learning of a world model

Action computation is optimized by adaptive importance sampling,

- this is a type of motor learning
- but is complemented by sampling (’halffabrikaat’)

Many open problems

- Sensing, acting interdependence
- action hierarchies in terms of action building blocks

Thank you!

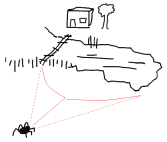
S. Thijssen and H. J. Kappen. ”Path Integral Control and State Dependent Feedback.” Phys. Rev. E 91, 032104 2015

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.

Thalmeier, D., Uhlmann, M., Kappen, H. J., Memmesheimer, R. M. (2015). Learning universal computations with spikes. Plos Computational Biology 2016

Thalmeier, D., Gomez, V. Kappen, H.J. Action selection in growing state spaces: Control of Network Structure Growth. Journal of Physics A (arXiv:1606.07777).



www.snn.ru.nl/~bertk