

A survey of several approximation techniques in optimal control

Rémi Munos

Centre de Mathématiques Appliquées,
Ecole Polytechnique, France.

www.cmap.polytechnique.fr/~munos

Outline of the presentation:

1. Approximation of the value function
 - Continuous-time case:
 - Non-uniqueness of HJB equations
 - Discretization techniques
 - Discrete-time case
 - Approximate dynamic programming
2. Parameterization of the policy
 - Sensitivity of the performance measure w.r.t. control parameters:
 - Pathwise approach
 - Likelihood, Malliavin, Adjoint, Martingale approaches
 - Reinforcement learning approach

Optimal control problem (deterministic case)

State: $x(t) \in \Omega \subset \mathbb{R}^n$ follows the controlled dynamics:

$$\frac{dx(t)}{dt} = f(x(t), u(t)),$$

where $u(t) \in U$ is the **control** .

Goal: find the control $u(\cdot)$ that maximizes some performance measure:

$$J(x, u(\cdot)) = \int_0^T \gamma^t r(x(t), u(t)) dt + \gamma^T R(x(T))$$

where T is an exit time from the domain Ω .

The **value function** $V(x) = \sup_{u(\cdot)} J(x, u(\cdot))$ solves the

Hamilton-Jacobi-Bellman equation:

$$V(x) \ln \gamma + \max_{u \in U} [r(x, u) + \nabla V(x) \cdot f(x, u)] = 0, \text{ for } x \in \Omega,$$

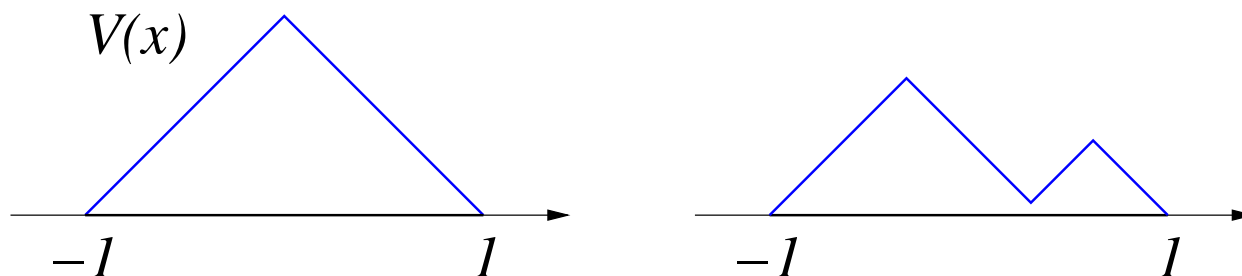
with boundary conditions: $V(x) \geq R(x)$ on $\partial\Omega$.

Solution to HJB, in what sense?

Example: consider a minimum exit-time problem in 1d from $\Omega = [-1, 1]$,

$$\frac{dx}{dt} = u \in \{-1, 1\}.$$

Then HJB is $|V'(x)| = 1$ for $x \in]-1, 1[$, and $V(-1) = V(1) = 0$.



Problems:

- V is not differentiable everywhere, thus there are no regular solution.
- There are an infinity of generalized solutions (i.e. almost everywhere differentiable)!!! Which one is the value function?

Need weak formulation \rightarrow **Viscosity solutions** [Crandall & Lions, 1983].

Approximate solution to HJB

Write HJB: $H(V, x) = 0$, where H is the *Hamiltonian* (also called *Bellman residual*):

$$H(V, x) = V(x) \ln \gamma + \max_{u \in U} [\nabla V(x) \cdot f(x, u) + r(x, u)].$$

Let us approximate the VF by a parameterized function V_α , say, e.g., with a neural network (the parameters α being the weights of the network).

Weight update: gradient descent on the error:

$$E(\alpha) = \frac{1}{2} \int [H(V_\alpha, x)]^2 dx.$$

Algorithm: draw random states x_t and perform a stochastic gradient descent:

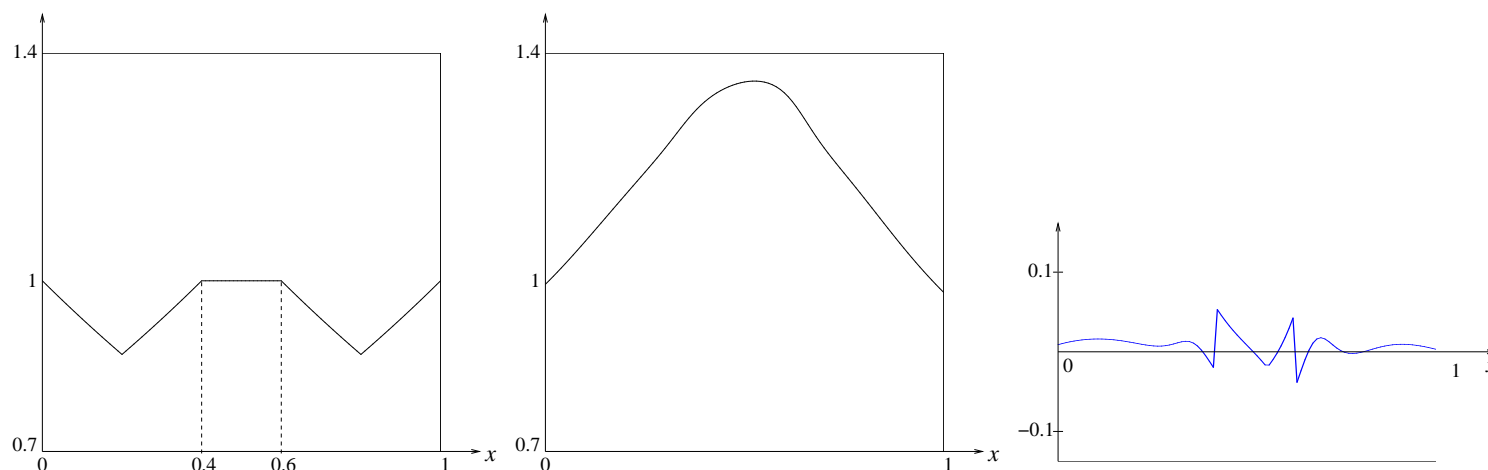
$$\alpha \leftarrow \alpha - \eta \nabla_\alpha [H(V_\alpha, x_t)]^2.$$

Illustration of the problem

1d problem $x_t \in [0, 1]$ with state dynamics $\frac{dx}{dt} = u \in \{-1, 1\}$. Current reward: $r(x) = -\ln \gamma \mathbf{1}_{x \in [0.4, 0.6]}$ and terminal reward $R(0) = 1, R(1) = 1$. Then $V(x) = \gamma^x \mathbf{1}_{x \leq 0.2} + \gamma^{0.4-x} \mathbf{1}_{x \in [0.2, 0.4]} + \mathbf{1}_{x \in [0.4, 0.6]} + \gamma^{x-0.6} \mathbf{1}_{x \in [0.6, 0.8]} + \gamma^{1-x} \mathbf{1}_{x \geq 0.8}$.

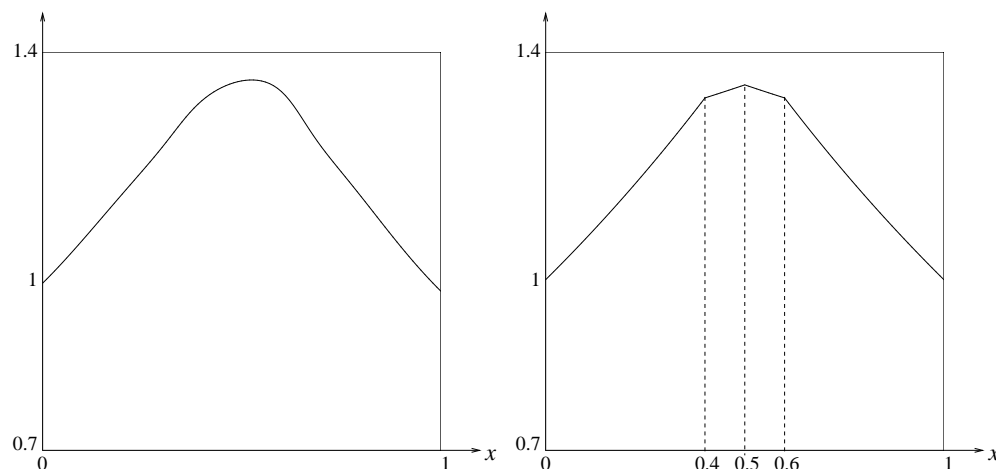
At places where V is differentiable, it solves:

$$V(x) \ln \gamma + |V'(x)| + r(x) = 0.$$



The VF, the network approximation (with 100 hidden units) and the residual. The gradient algorithm succeeded in minimizing the residual!

Explanation of the problem



The algorithm converges to a generalized solution different from the VF:

$$\gamma^{-x} \mathbf{1}_{x \leq 0.4} + [1 + (1 - \gamma^{0.4}) \gamma^{-x}] \mathbf{1}_{x \in [0.4, 0.5]} + [1 + (1 - \gamma^{0.4}) \gamma^{x-1}] \mathbf{1}_{x \in [0.5, 0.6]} + \gamma^{x-1} \mathbf{1}_{x \geq 0.6}.$$

The problem comes from non-uniqueness of generalized solutions to HJB equations.

→ The problem of minimizing residual error has an infinity of *global* minima (ill-posed problem).

See [Munos, Baird and Moore, 1999].

How to go around problem?

Several approaches enable to recover uniqueness of solution:

1. *Introduction of stochasticity*: then the VF is the unique smooth solution to HJB.
2. *Temporal discretization*, with some time step $h > 0$ gives a MDP whose VF V^h is the unique solution to a dynamic programming equation $V^h(x) = \mathcal{T}^h V^h(x)$, where \mathcal{T}^h is the Bellman operator.
3. *Policy Iteration*. Non-uniqueness of HJB is due to its non-linearity. For a given policy π , the linear PDE

$$W(x) \ln \gamma + \nabla W(x) \cdot f(x, \pi(x)) + r(x, \pi(x)) = 0$$

has a unique solution V^π . A policy iteration procedure builds a sequence of policies π_k satisfying in all x ,

$$\pi_{k+1}(x) \in \arg \max_{u \in U} [\nabla V^{\pi_k}(x) \cdot f(x, u) + r(x, u)].$$

Then V^{π_k} converges to V .

Approximation scheme. Ex: Finite-difference method

Consider a regular grid X_h on the domain. The HJB equation (deterministic case):

$$V(x) \ln \gamma + \max_{u \in U} [r(x, u) + \nabla V(x) \cdot f(x, u)] = 0$$

is discretized into

$$V^h(x) \ln \gamma + \max_{u \in U} \left[r(x, u) + \sum_{i=1}^d [\Delta_i^+ V^h(x) f_i^+(x, u) + \Delta_i^- V^h(x) f_i^-(x, u)] \right] = 0$$

where the gradient $\nabla V(x)$ is replaced by the finite difference quotient:

$$\begin{cases} \Delta_i^+ V^h(x) &= \frac{1}{h} [V(x + he_i) - V(x)] \\ \Delta_i^- V^h(x) &= \frac{1}{h} [V(x - he_i) - V(x)] \end{cases}$$

Numerical scheme

We deduce:

$$V^h(x) = \max_{u \in U} \left[\gamma^{\tau(x,u)} \sum_{i=1}^d p(x_i|x, u) V^h(x_i) + \tau(x, u) r(x, u) \right],$$

with

$$\tau(x, u) = \frac{h}{\sum_{i=1}^d |f_i(x, u)|} \quad \text{and} \quad p(x_i|x, u) = \frac{|f_i(x, u)|}{\sum_{j=1}^d |f_j(x, u)|}.$$

This is a **dynamic programming equation** for some **Markov Decision Process**, whose:

- **State space** is the set of grid points X_h
- **Transition probabilities** are $p(x_i|x, u)$, where $x \in X_h$ and $x_i = x + he_i \in X_h$.

Markov Decision process

State space X , *control* (or action) *space* U . Transition from a state x to a state y occurs with *probability* $p(y|x, u)$. A *reward* $r(x, u)$ is obtained.

Goal: find controller $u(\cdot)$ that maximizes

$$J(x, u(\cdot)) = \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t)$$

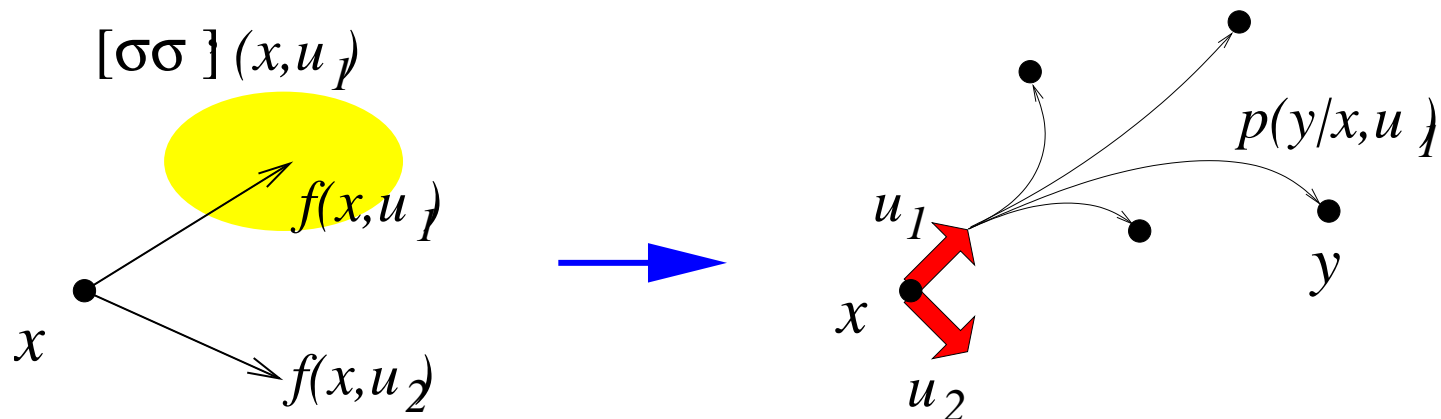
Def: a **policy** = a feed-back controller $u(t) = \pi(x_t)$.

The **value function** of a policy π is $V^\pi(x) = J(x, \pi(x))$.

The **optimal value function** $V^* = \max_{\pi} V^\pi$ solves the **dynamic programming equation**:

$$V^*(x) = \max_{u \in U} \left[\gamma \sum_y p(y|x, u) V^*(y) + r(x, u) \right].$$

Consistency of the scheme



Continuous process \rightarrow Markov decision process

$dx = f(x, u)dt + \sigma(x, u)dW_t$ \rightarrow $p(y|x, u)$

V solves HJB equation \rightarrow V^h solves DP equation

Consistency property (Kushner, 1990):

$$\begin{cases} \mathbb{E}[y - x] = f(x, u)h + o(h) \\ \text{Cov}[y - x] = [\sigma\sigma'](x, u)h + o(h) \end{cases}$$

Convergence of numerical schemes

Convergence of the value function $V^h \xrightarrow{h \rightarrow 0} V$ when the scheme is consistent ([Kushner, 1990], [Barles & Souganidis, 1991]).

V^h may be computed by *value iteration*:

$$V_{n+1}^h = \mathcal{T}^h V_n^h.$$

Then, because of the **strong contraction** property:

$$\|V_{n+1}^h - V^h\|_\infty \leq \lambda \|V_n^h - V^h\|_\infty, \text{ with } \lambda < 1,$$

we have

$$V_n^h \xrightarrow{n \rightarrow \infty} V^h \xrightarrow{h \rightarrow 0} V.$$

Convergence of reinforcement learning algorithms

Si the state dynamics is unknown \rightarrow estimation of the probabilities $p(x_j|x_i, u)$ from “observation”.

The strong contraction property does not hold anymore (thus $V_n^h \not\rightarrow V^h$).

However, if some **weak contraction** property holds,

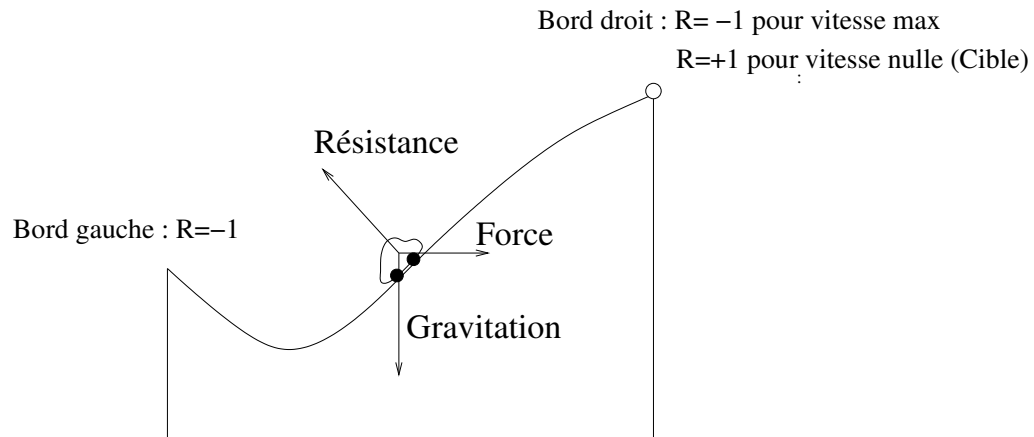
$$\|V_{n+1}^h - V^h\|_\infty \leq (1 - kh)\|V_n^h - V^h\|_\infty + o(h),$$

we have the convergence of perturbed schemes and $V_n^h \rightarrow V$ occurs when $n \rightarrow \infty$ and $h \rightarrow 0$.

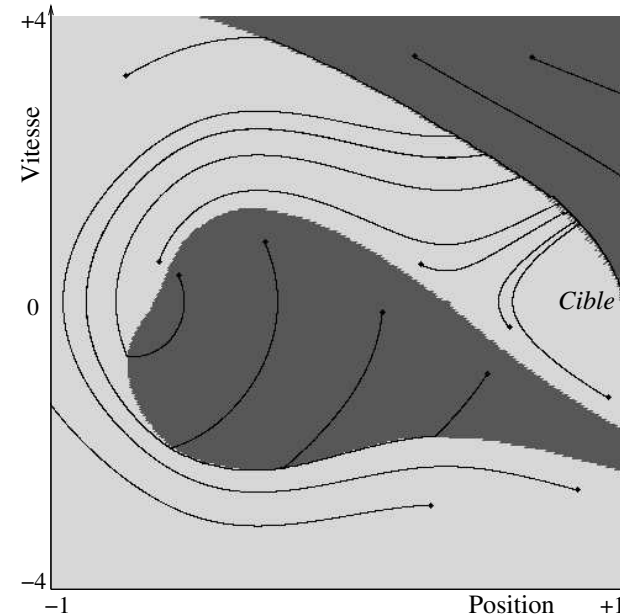
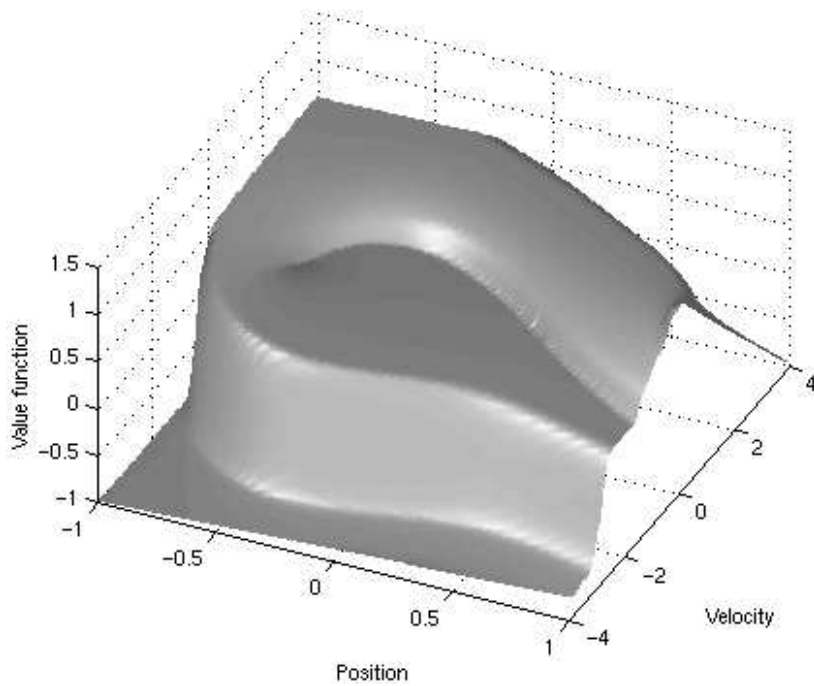
PhD work:

- Model-based and model free RL algorithms [Munos, 2000].
- Extension to the stochastic case [Munos & Bourgine, 1997].
- Comparison with *Q-learning* [Munos, 1997].

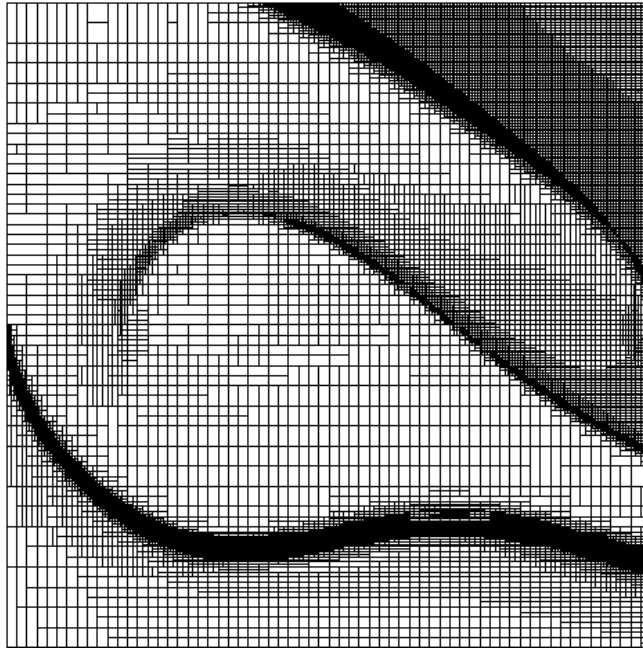
Variable resolution discretization. "Car on the Hill"



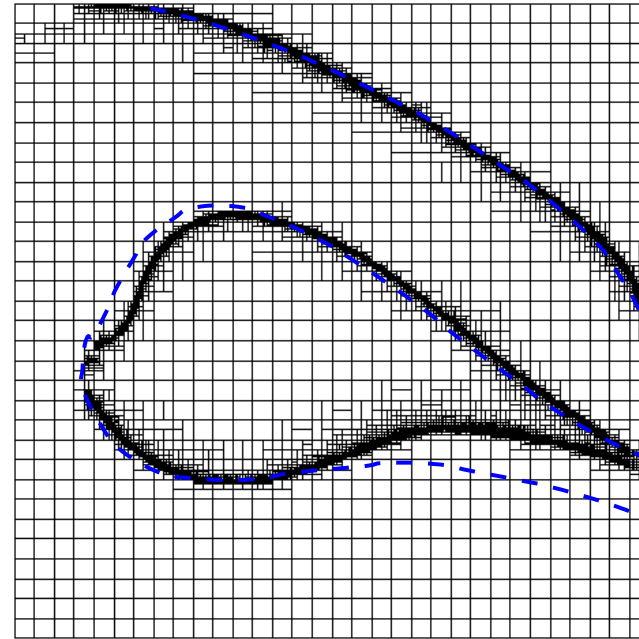
Goal: reach the top of the hill in minimum time and stop there. Avoid exiting from the left.



Refinement mesh criteria based on local information



(a) according to the VF



(b) according to the policy

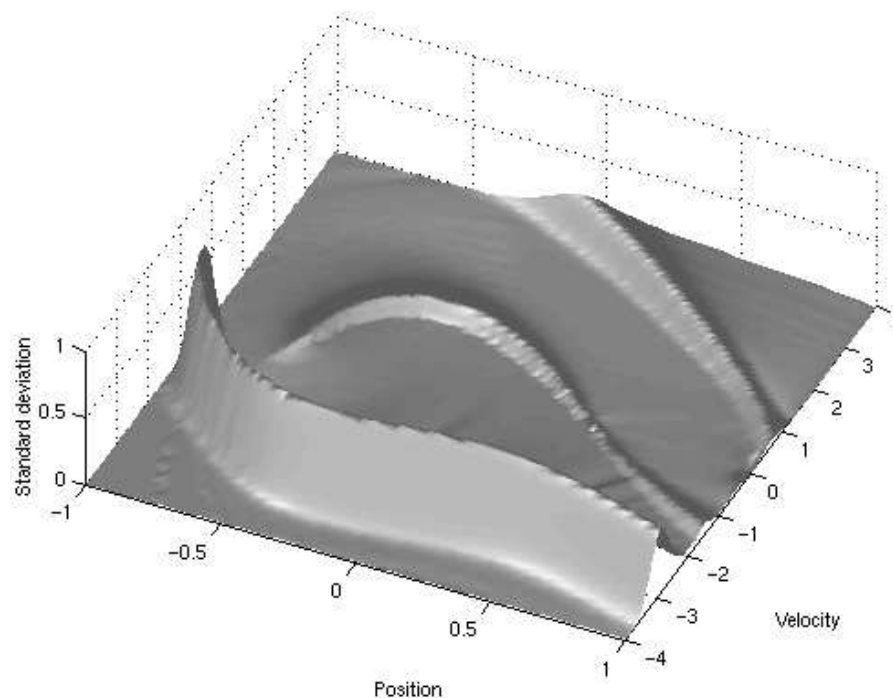
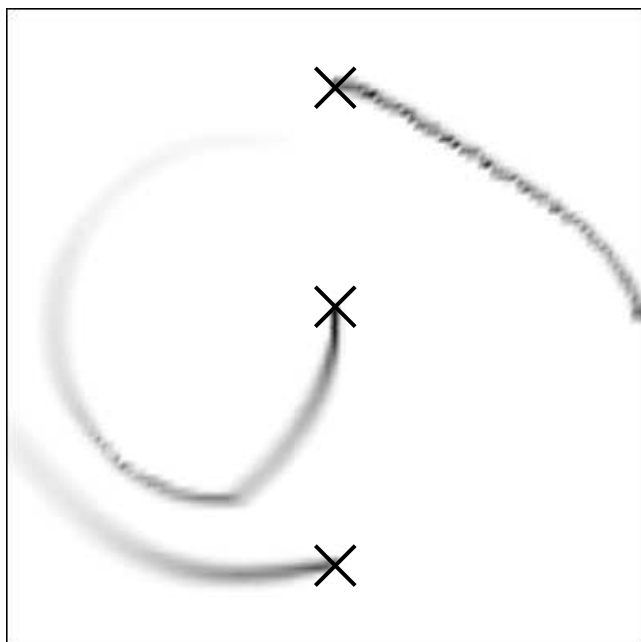
Non-local influence of the VF on the optimal control switching boundaries.

→ How should we set up available resources (i.e. grid points) in order to maximize the performance of the policies deduced from the corresponding VFs?

Towards a global mesh refinement heuristic

Good approximation of the VF at the switching boundaries of the optimal control, in order to localize it precisely. Two tools:

- The *influence* $I(y|x)$ of a reward $r(y)$ on $V(x)$,
- The *variance* $\sigma^2(x)$ introduced when discretizing the VF.

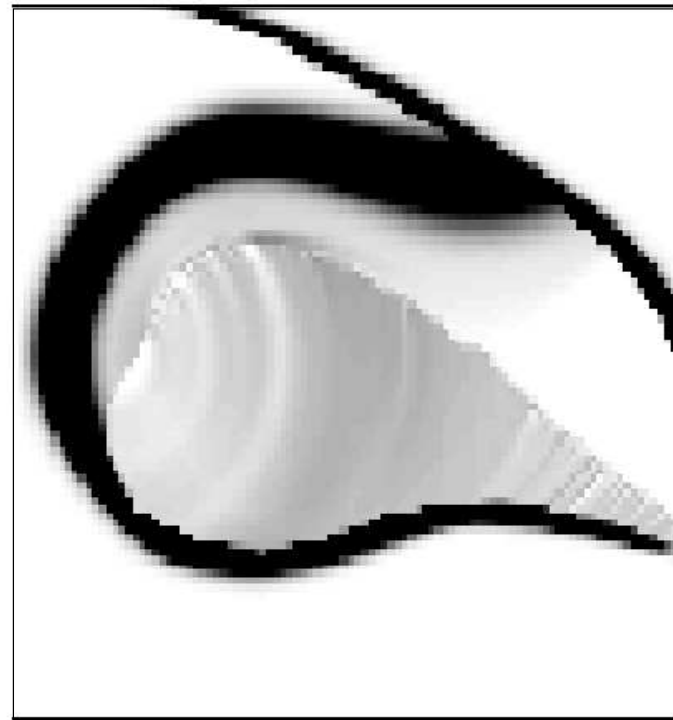


A global mesh refinement heuristic

[Munos & Moore, 2002] Select areas whose uncertainty on the VF has the highest influence on the switching boundaries of the optimal control.

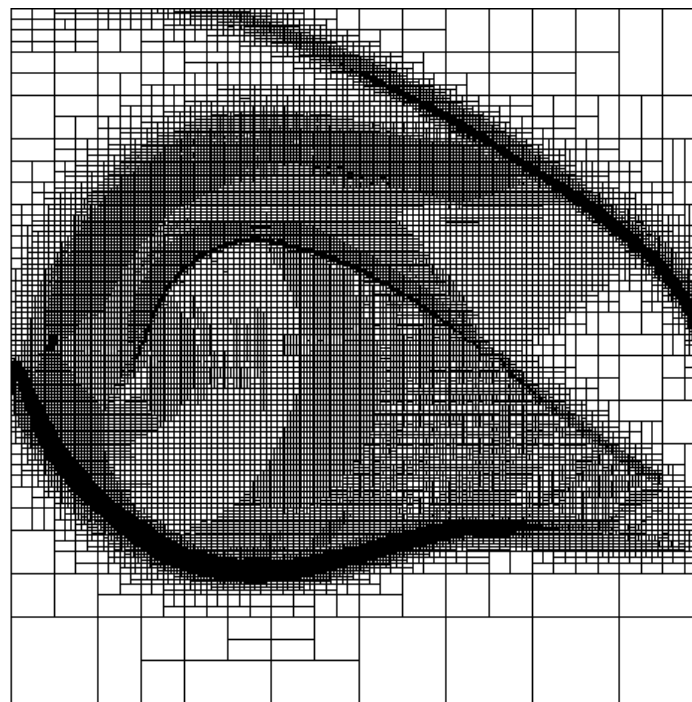
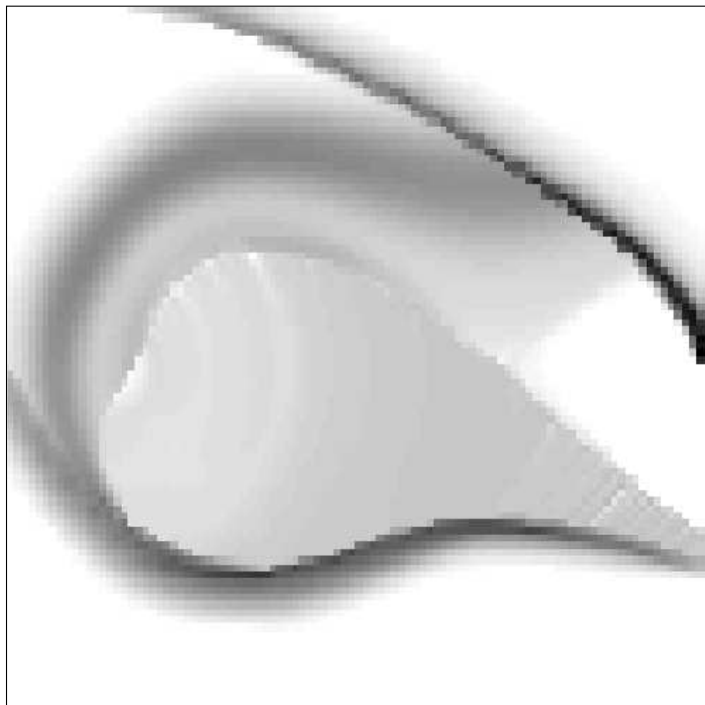


(a) Frontière de transition de la commande



(b) Influence sur ces états

Resulting grid



Performance: global criterion \gg local criteria $>$ uniform.

→ crucial in high dimensions

Ex. in **4d** and **5d**: *inverted pendulum, space shuttle, Acrobot, airplane rendez-vous* (with Olivier Sigaud for Dassault-Aviation),
in **6d**: *bicycle* (combined with random grids).

Discrete-time case:

Approximate Dynamic Programming

Generalization of usual error bounds in L_∞ norm to similar bounds in L_1 et L_2 norms, for:

- 1. Approximate value iteration**
- 2. Approximate policy iteration**
3. Same ideas may be generalized to other algorithms

Markov Decision Process

State space X , action space U , transition probabilities $p(y|x, u)$ and rewards function $r(x, u, y)$.

Definitions:

- A **policy** $\pi = \text{mapping } X \rightarrow U$,
- Performance measure of a policy π : the **value function** V^π . Ex. in the discounted infinite-time horizon case:

$$V^\pi(x) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r(x_t, a_t, y_t) \mid x_0 = x, u_t = \pi(x_t) \right]$$

(where $0 \leq \gamma < 1$ is the discount factor),

Optimal control problem: find the optimal policy π^* , i.e.

$$V^{\pi^*} = \max_{\pi} V^\pi$$

The corresponding value V^* is called the **optimal value function**.

Dynamic programming equation

Proposition: the optimal value function V^* solves the *dynamic programming equation*

$$V^* = \mathcal{T}V^*$$

where $\mathcal{T} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is the **Bellman operator**:

$$\mathcal{T}W(x) := \max_{u \in U} \sum_y p(y|x, u) [r(x, u, y) + \gamma W(y)].$$

Definition: a policy π is **greedy** w.r.t. W if for all state x ,

$$\pi(x) \in \arg \max_{u \in U} \sum_y p(y|x, u) [r(x, u, y) + \gamma W(y)]$$

Property: an policy greedy w.r.t. V^* is optimal.

Value Iteration algorithm:

$$V_{n+1} = \mathcal{T}V_n,$$

converges to V^* (since $\|V_{n+1} - V^*\|_\infty \leq \gamma \|V_n - V^*\|_\infty$).

Approximate value iteration

AVI algorithm:

$$V_{n+1} = \mathcal{A}TV_n,$$

where T is the *Bellman operator* and \mathcal{A} an *approximation operator*.

Error bound in L_∞ norm:

Proposition 1 [Bertsekas & Tsitsiklis, 1996] *If the approximation errors are uniformly bounded*

$$\|TV_n - \mathcal{A}TV_n\|_\infty \leq \varepsilon,$$

then, the asymptotic loss of using policy π_n , greedy w.r.t. the approximation V_n , instead of the optimal policy, is bounded by:

$$\limsup_{n \rightarrow \infty} \|V^* - V^{\pi_n}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \varepsilon.$$

Problem: usually, approximation operators minimize (weighted) L_1 or L_2 norms.

Example of AVI implementation

Stage n . Approximation V_n .

1. Select states $(x_k)_{k=1\dots K}$ sampled according to some distribution μ ,
2. Compute the backed-up values $v_k = \mathcal{T}V_n(x_k)$,
3. Define a new approximation $V_{n+1} \in \mathcal{F}$, by solving the L_2 -minimization problem:

$$\inf_{W \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^K [W(x_k) - v_k]^2 \simeq \|W - \mathcal{T}V_n\|_{\mu}^2$$

(where $\|u\|_{\mu} := [\sum_x \mu(x)u(x)^2]^{1/2}$ is the L_2 -norm weighted by μ).

Other examples: non-linear approximation (neural networks, adaptive wavelets), non-parametric methods (local linear regression, kernel methods).

Error bound in L_2 norm

Let μ a distribution on X . Define the *smoothness constant* C of the (discounted) future state distribution w.r.t. to μ : for all sequence of policies π_1, π_2, \dots , for all state y ,

$$(1 - \gamma)^2 \sum_{m=1}^{\infty} m \gamma^{m-1} \Pr\{x_m = y | x_0 \sim \mu, x_i \sim p(\cdot | x_{i-1}, \pi_i(x_{i-1}))\} \leq C \mu(y).$$

Theorem 1 [Munos, 2004] Assume that $\|\mathcal{A}TV_n - TV_n\|_{\mu} \leq \varepsilon$, then

$$\limsup_{n \rightarrow \infty} \|V^* - V^{\pi_n}\|_{\mu} \leq \frac{2\gamma}{(1 - \gamma)^2} \sqrt{C} \varepsilon.$$

Smoothness of the future state distribution

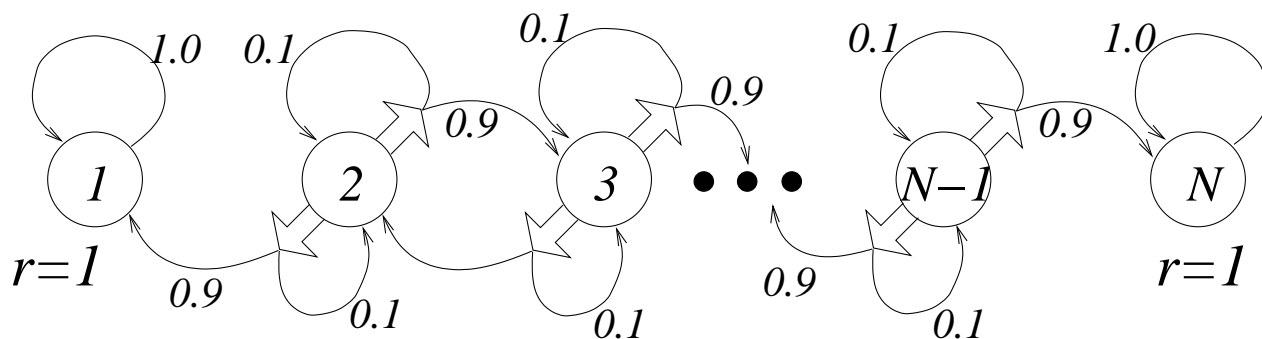
Assume uniform distribution $\mu = (\frac{1}{N} \dots \frac{1}{N})$.

- **C is maximum** when a specific state is successor, for some policy, of all states with probability 1. Then $C = N$ and the L_2 bound is not better than the L_∞ one.
- **C is minimum** when all transition probabilities are uniform. Then $C = 1$.

$C \in [1, N]$ expresses the smoothness of the (discounted) future state distribution w.r.t. the initial distribution μ .

Specific interest: in continuous space problems, the constant C is independent from the number of discretization points N .

Illustration on the “chain walk” MDP



Approximation $V_n(x) = \alpha_n + \beta_n(x)$ with $x \in \{1 \dots N\}$.

Let $V_0 = (0, \dots, 0)'$, then $\mathcal{T}V_0 = (1, 0, \dots, 0, 1)'$.

- **L_∞ -norm:** $V_1 = (\frac{1}{2}, \dots, \frac{1}{2})'$. Error $\|V_1 - \mathcal{T}V_0\|_\infty = \frac{1}{2}$. By induction, $\|V_{n+1} - \mathcal{T}V_n\|_\infty = \frac{1}{2}$.
- **L_2 -norm:** $V_1 = (\frac{2}{N}, \dots, \frac{2}{N})'$. Error $\|V_1 - \mathcal{T}V_0\|_2 = \frac{\sqrt{2N-4}}{N}$. By induction, $\|V_{n+1} - \mathcal{T}V_n\|_2 = \frac{\sqrt{2N-4}}{N}$.

Here $C = (1 - \gamma)^2 \sum_{m \geq 1} m \gamma^{m-1} (1 + 0.9m)$ is independent from N .

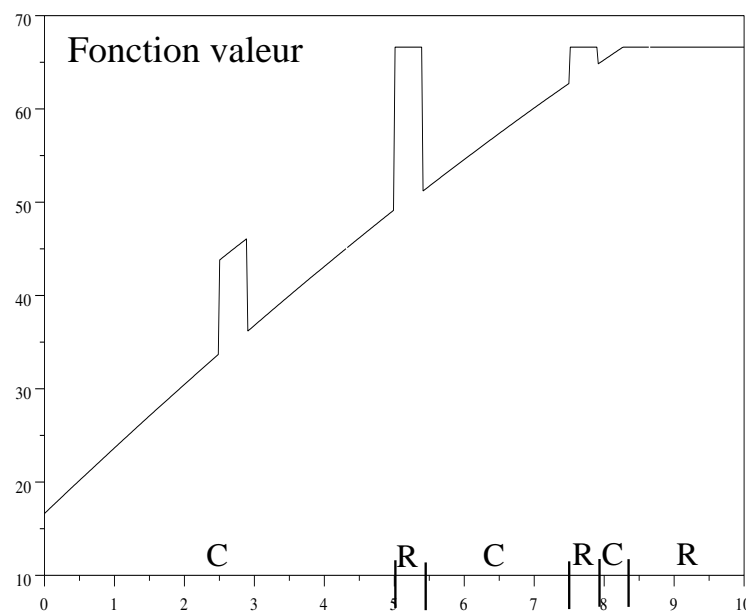
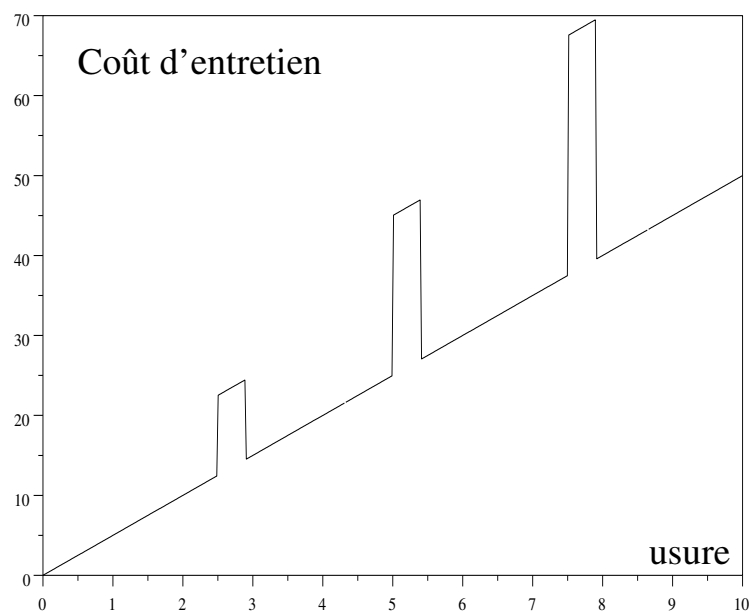
Tightness of the L_2 bound is $O(N^{-1/2})$, that of L_∞ bound is $O(1)$.

Optimal replacement problem

State: accumulated utilization of a durable (ex. odometer of a car).

Decisions:

- **Keep:** maintenance cost. New state $y \sim x + \exp(\beta)$
- **Replace:** fixed replacement cost. New state $y \sim \exp(\beta)$.



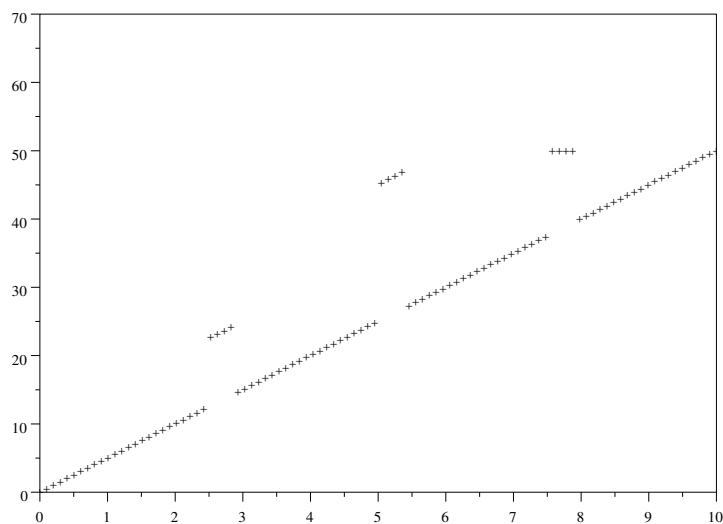
Here, $\beta = 0.6$, $\gamma = 0.6$. Here we have $C \simeq \beta x_{\max} = 6$.

Linear approximation

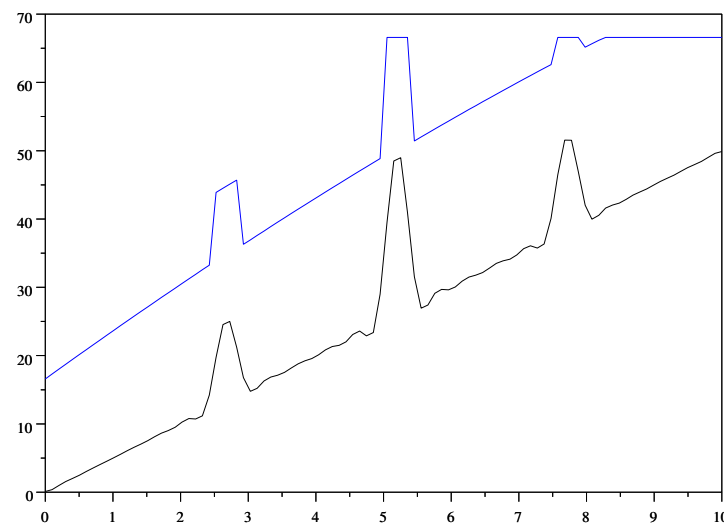
$$\text{Space } \mathcal{F} := \left\{ V_n(x) = \sum_{k=1}^{20} \alpha_k \cos\left(k\pi \frac{x}{x_{\max}}\right) \right\}.$$

Discretization on a uniform grid with N points.

First iteration: $V_0 = 0$,

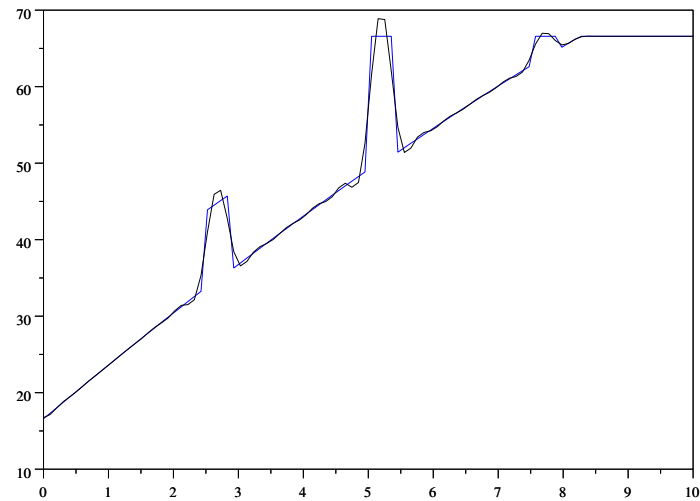
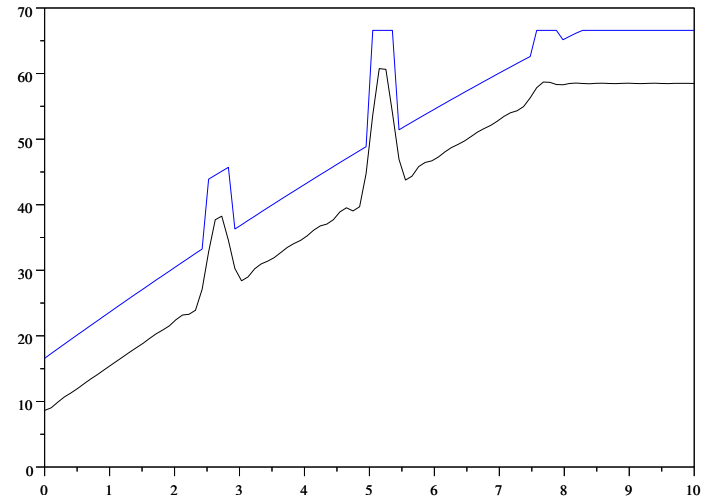
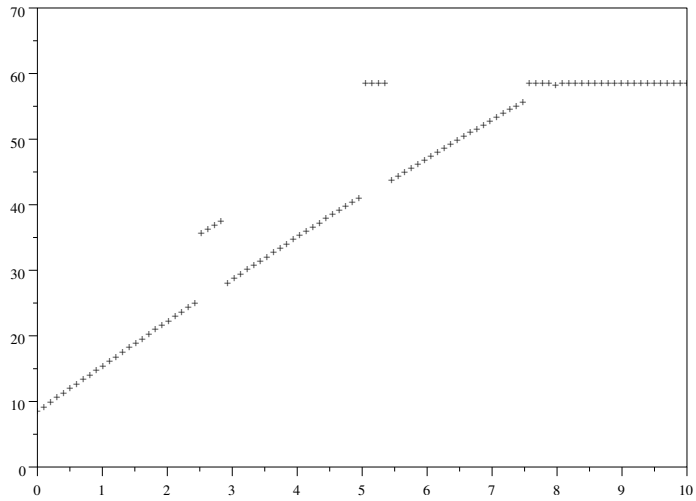


Iterate values $\{\mathcal{T}V_0(x_n)\}_{1 \leq n \leq N}$



Approximation $V_1 \in \mathcal{F}$ of $\mathcal{T}V_0$

Next iterations



Tightness of error bounds

	L_∞	L_1	L_2
$N = 200$	12.4	0.367	1.16
$N = 2000$	12.4	0.0552	0.897

Table 1: Approximation errors in L_∞ , L_1 and L_2 norms.

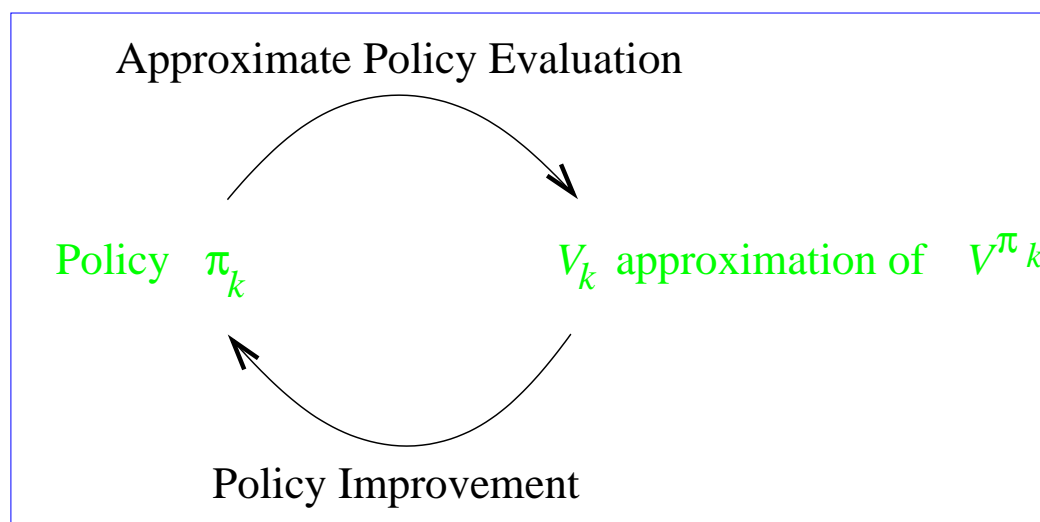
The cost function being discontinuous, the approximation error L_∞ cannot be lower than some value.

- L_2 (and L_1) error bounds allow to express the performance of AVI algorithm using the same norm as the one used in the minimization problem performed by the approximation operator.
→ usefulness and tightness of these bounds.

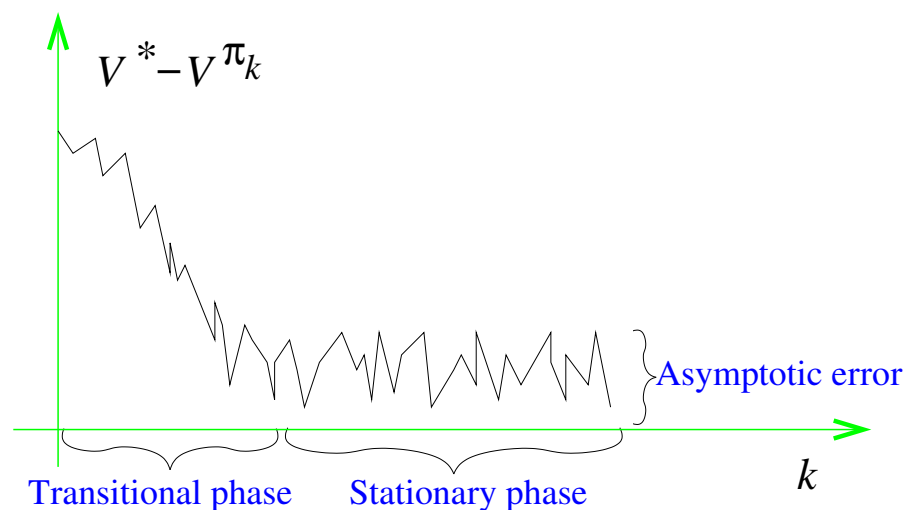
Approximate policy iteration

Proceeds in two steps:

- *Approximate policy evaluation step*: for a given policy π_k , we compute an approximation V_k of the value function V^{π_k} .
- *Policy improvement step*: we generate a new policy π_{k+1} greedy w.r.t. V_k .



Asymptotic performance of API



Error bound in L_∞ [Bertsekas & Tsitsiklis, 1996]:

Bound on the loss in performance $V^* - V^{\pi_k}$ resulting from using policy π_k instead of the optimal one, as a function of the *approximation errors*

$V_k - V^{\pi_k}$:

$$\limsup_{k \rightarrow \infty} \|V^* - V^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{k \rightarrow \infty} \|V_k - V^{\pi_k}\|_\infty.$$

L_2 error bounds

[Munos, 2003] Let μ be a distribution. Define the stochastic matrices:

$$S_n = \frac{(1-\gamma)^2}{2} (I - \gamma P^{\pi^*})^{-1} [P^{\pi_{n+1}} (I - \gamma P^{\pi_{n+1}})^{-1} + P^{\pi^*} (I - \gamma P^{\pi_n})^{-1}],$$
$$\tilde{S}_n = \frac{(1-\gamma)^2}{2} (I - \gamma P^{\pi^*})^{-1} [P^{\pi_{n+1}} (I - \gamma P^{\pi_{n+1}})^{-1} (I + \gamma P^{\pi_n}) + P^{\pi^*}].$$

Then $\mu_n := \mu S_n$ and $\tilde{\mu}_n := \mu \tilde{S}_n$ are distributions, and we have

$$\limsup_{n \rightarrow \infty} \|V^* - V^{\pi_n}\|_{2,\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{n \rightarrow \infty} \|V_n - T^{\pi_n} V_n\|_{2,\mu_n}$$
$$\limsup_{n \rightarrow \infty} \|V^* - V^{\pi_n}\|_{2,\mu} \leq \frac{2\gamma}{(1-\gamma)^2} \limsup_{n \rightarrow \infty} \|V_n - V^{\pi_n}\|_{2,\tilde{\mu}_n}$$

Linear approximation

Space of functions $\mathcal{F} = \{V_\alpha(x) = \sum_{k=1}^K \alpha_k \phi_k(x)\}_{\alpha \in \mathbb{R}^K}$ linearly parameterized by $\alpha \in \mathbb{R}^K$.

[Munos, 2003] The performance of API algorithm may be bounded as a function of the representational power of the approximation architecture:

$$\limsup_{n \rightarrow \infty} \|V^* - V^{\pi_n}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2} C \varepsilon_{\mathcal{F}}.$$

where C is a constant and $\varepsilon_{\mathcal{F}}$ is the **representational power of the approximation architecture**:

$$\varepsilon_{\mathcal{F}} := \max_{\pi_n} d(V^{\pi_n}, \mathcal{F}).$$

Generalization to other ADP methods

Example: Minimization of the Bellman residual

$$\inf_{V_\alpha \in \mathcal{F}} \|\mathcal{T}V_\alpha - V_\alpha\|.$$

L_∞ **bound** [Williams & Baird, 1993]: the performance of a policy π_α greedy w.r.t. V_α is bounded by the Bellman residual of V_α :

$$\|V^* - V^{\pi_\alpha}\|_\infty \leq \frac{2}{1-\gamma} \|\mathcal{T}V_\alpha - V_\alpha\|_\infty.$$

L_1 and L_2 **bounds**: let μ be a distribution, then

$$\|V^* - V^{\pi_\alpha}\|_\mu \leq \frac{2}{1-\gamma} \|\mathcal{T}V_\alpha - V_\alpha\|_{\mu_\alpha}$$

Part II. Parameterization of the policy (continuous time case)

Consider a parameterized policy π_α . The optimal control problem is replaced by a **parametric optimization problem**.

The state dynamics may be written:

$$dX_t^\alpha = f(X_t^\alpha, \alpha)dt + \sigma(X_t^\alpha, \alpha)dW_t.$$

Consider a finite temporal horizon T :

$$V^{\pi_\alpha}(x) = \mathbb{E} [r(X_T^\alpha) | X_0^\alpha = x].$$

One may search a local maximum of $\alpha \rightarrow V^{\pi_\alpha}$ with a gradient ascent method:

$$\alpha \leftarrow \alpha + \eta \partial_\alpha V^{\pi_\alpha}.$$

\rightarrow we need the gradient $\partial_\alpha V^{\pi_\alpha}$ (sensitivity of the performance measure w.r.t. control parameters).

Pathwise sensitivity

If r is smooth, one may put the derivation under the expectation, and deduce the estimator [Yang & Kushner, 1991]:

$$\partial_\alpha V^{\pi_\alpha} = \mathbb{E}[\nabla r(X_T^\alpha) Z_T],$$

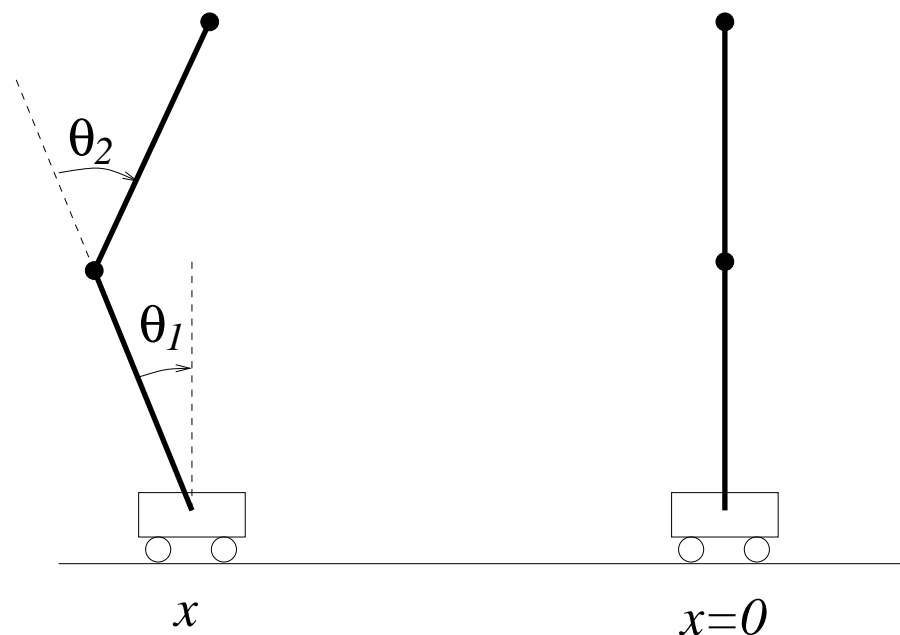
with $Z_t = \partial_\alpha X_t^\alpha$, the **state gradient**, that solves

$$dZ_t = (\partial_\alpha f_t + \nabla_x f_t Z_t)dt + (\partial_\alpha \sigma_t + \nabla_x \sigma_t Z_t)dW_t.$$

Example: double inverted pendulum

Space 6 dimension: **state** $x, v, \theta_1, \omega_1, \theta_2, \omega_2$.

Control: force applied to the cart.

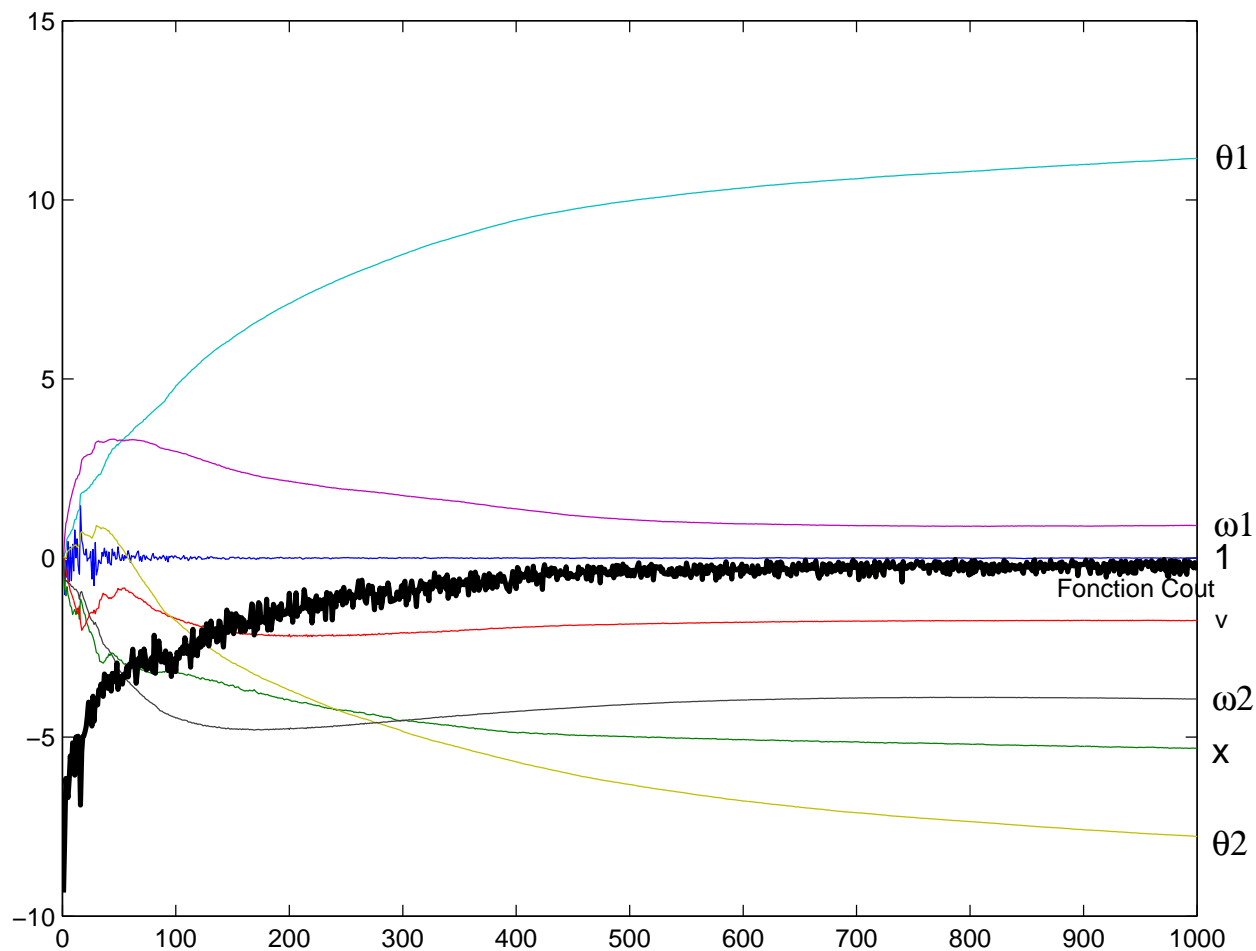


Maximize reward $r = -(x^2 + v^2 + \theta_1^2 + \omega_1^2 + \theta_2^2 + \omega_2^2)$ at final time $T = 1$.

We search for a linear controller.

double inverted pendulum

Etat initial choisi aleatoirement



Control:

$$u = -0.65 - 13.93 x - 1.81 v + 10.65 \theta_1 + 12.09 \omega_1 + 2.30 \theta_2 - 13.46 \omega_2.$$

Likelihood ratio method

However, what if r is not smooth ?

We would like to write $\mathbb{E}[r'(X_T^\alpha)]$ in the form $\mathbb{E}[r(X_T^\alpha)g(t, X_T^\alpha)]$. If the law of X_T^α is known, it is a usual integration by part formula:

$$\begin{aligned}\mathbb{E}[r'(X_T^\alpha)] &= \int r'(x)\rho_t(x)dx = - \int r(x)\frac{\rho_t'(x)}{\rho_t(x)}\rho_t(x)dx \\ &= \mathbb{E}[r(X_T^\alpha)(\log \rho_t)'(X_T^\alpha)]\end{aligned}$$

If the law on X_T^α is unknown, but if σ does not depend on α , then a change in probability yields the estimator [Yang & Kushner, 1991]:

$$\partial_\alpha V^{\pi_\alpha} = \mathbb{E} \left[r(X_T) \int_0^T [\sigma_t^{-1} \partial_\alpha f_t]' dW_t \right].$$

called the *score* or *likelihood ratio* method [Glynn, 1987], [Baxter, 2001].

Now, what if σ does depend on α ?

Malliavin calculus approach

We use an integration by part formula in the sense of Malliavin calculus.

Proposition 2 [Gobet & Munos, 2004b] *Assume that the Malliavin covariance matrix of X_T^α , defined by $\Gamma_T := \int_0^T \mathcal{D}_t X_T^\alpha [\mathcal{D}_t X_T^\alpha]' dt$, is invertible, then*

$$\partial_\alpha V(\alpha) = \frac{1}{T} \mathbb{E} \left[r(X_T) \delta \left(Z_T' \Gamma_T^{-1} \mathcal{D}.X_T^\alpha \right) \right].$$

Where δ is the Skorohod integral and $\mathcal{D}.X_T^\alpha$ the Malliavin derivative.

This formula is actually computationally very heavy (see the guide [Gobet & Munos, 2004c] for details of numerical implementation).

Adjoint approach

By differentiating the PDE solved by V^α , one derive the Feynman-Kac formula:

$$\partial_\alpha V(\alpha) = \int_0^T \mathbb{E} \left[\sum_{i=1}^d \partial_\alpha f_{i,t} \partial_{x_i} V^\alpha(t, X_t^\alpha) + \frac{1}{2} \sum_{i,j=1}^d \partial_\alpha [\sigma \sigma']_{ij,t} \partial_{x_i x_j}^2 V^\alpha(t, X_t^\alpha) \right] dt.$$

The processes $\partial_{x_i} V^\alpha(t, X_t^\alpha)$ and $\partial_{x_i x_j}^2 V^\alpha(t, X_t^\alpha)$ being the *adjoint states*.

We use integration by part formulas to make those processes explicit:

$$\nabla_x V^\alpha(t, X_t^\alpha) Y_t = \frac{1}{T-t} \mathbb{E} \left[r(X_t^\alpha) \left(\int_t^T [\sigma_s^{-1} Y_s]' dW_s \right)' \right].$$

Adjoint approach

Proposition 3 [Gobet & Munos, 2004b] *Assume σ is invertible, then*

$$\partial_\alpha V(\alpha) = \frac{1}{T} \mathbb{E} \left[r(X_T^\alpha) (H_T^f + H_T^\sigma) \right], \text{ where}$$

$$H_T^f = \int_0^T dt \partial_\alpha f_t \cdot \frac{(Y_t^{-1})'}{T-t} \int_t^T [\sigma_s^{-1} Y_s]' dW_s,$$

$$\begin{aligned} H_T^\sigma = & \int_0^T dt \sum_{i,j=1}^d \partial_\alpha [\sigma \sigma']_{ij,t} \left(\frac{2e_j}{T-t} \cdot [(Y_t^{-1})]' \int_{\frac{t+T}{2}}^T [\sigma_s^{-1} Y_s]' dW_s \right) \\ & \times \frac{e_i}{T-t} \cdot [(Y_t^{-1})]' \int_t^{\frac{t+T}{2}} [\sigma_s^{-1} Y_s]' dW_s \\ & + \frac{e_i}{T-t} \cdot \left\{ \nabla_x [(Y_t^{-1})]' \int_t^{\frac{t+T}{2}} [\sigma_s^{-1} Y_s]' dW_s \right\} Y_t^{-1} e_j \end{aligned}$$

This formula is actually much simpler than the previous one!

Numerically interesting when the number of parameters is large.

Martingale approach

We simply use martingale property of the processes: $[V^{\pi_\alpha}(t, X_t^\alpha)]_{0 \leq t \leq T}$ and $[\nabla_x V^{\pi_\alpha}(t, X_t^\alpha) \partial_\alpha X_t^\alpha]_{0 \leq t \leq T}$.

Proposition 4 [Gobet & Munos, 2004] *Assume that σ is invertible, then*

$$\begin{aligned} \partial_\alpha V^{\pi_\alpha} &= \mathbb{E} \left[r(X_T) \left(\frac{1}{T} \int_0^T [\sigma_s^{-1} Z_t]' dW_s \right. \right. \\ &\quad \left. \left. + \int_0^T \frac{dt}{(T-t)^2} \int_t^T [\sigma_s^{-1} (Z_s - Y_s Y_t^{-1} Z_t)]' dW_s \right) \right] \end{aligned}$$

Numerically simple.

Reinforcement learning algorithms?

Is it possible to define sensitivity estimators when the state dynamics is unknown from the decision maker?

Idea: use *stochastic policy* $\pi_\alpha \rightarrow$ replace $\partial_\alpha f$ by a likelihood ratio $\partial_\alpha \log \pi_\alpha$ of the policy.

Example: pathwise estimator in the deterministic case:

Discretize (with some time-step Δt) the process (X_t^α, Z_t) by $(X_t^{\Delta t}, Z_t^{\Delta t})$ by choosing at each discrete time $t \in \{j\Delta t\}$ a control u_t according to π_α and keep it for a time Δt . Then compute

$$\begin{aligned} Z_{t+\Delta t}^{\Delta t} &= Z_t^{\Delta t} + \partial_\alpha \log \pi_\alpha(u_t | X_t^{\Delta t}) \Delta X_t^{\Delta t} \\ &\quad + \nabla_x \log \pi_\alpha(u_t | X_t^{\Delta t}) \Delta X_t^{\Delta t} Z_t^{\Delta t} + \widehat{\nabla_x f}(X_t^{\Delta t}, u) Z_t^{\Delta t} \Delta t, \end{aligned}$$

with an estimator computed by least squares regression:

$$\widehat{\nabla_x f}(X_t^{\Delta t}, u) = (\Delta t)^{-1} (\overline{\Delta X X'} - \overline{\Delta X} \overline{X'}) (\overline{X X'} - \overline{X} \overline{X'})^{-1}.$$

Convergence of the RL estimator

The computation of $Z_t^{\Delta t}$ requires only the knowledge of the policy π_α and the trajectory $(X_s^{\Delta t})_{s \leq t}$.

Proposition 5 [Munos, 2005] *The discrete process converges*

$$(X_t^{\Delta t}, Z_t^{\Delta t}) \xrightarrow{\Delta t \rightarrow 0} (X_t^\alpha, Z_t) \text{ with probability 1,}$$

thus, the pathwise estimator converges:

$$\lim_{\Delta t \rightarrow 0} \nabla r(X_T^{\Delta t}) Z_T^{\Delta t} = \partial_\alpha V(\alpha) \text{ with probability 1.}$$

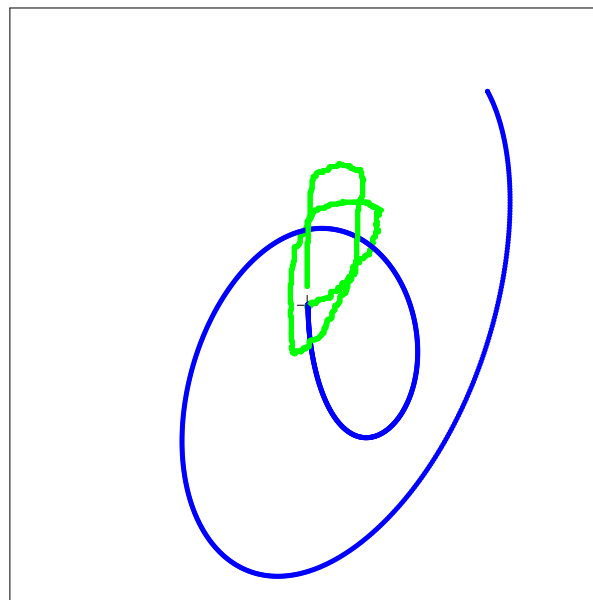
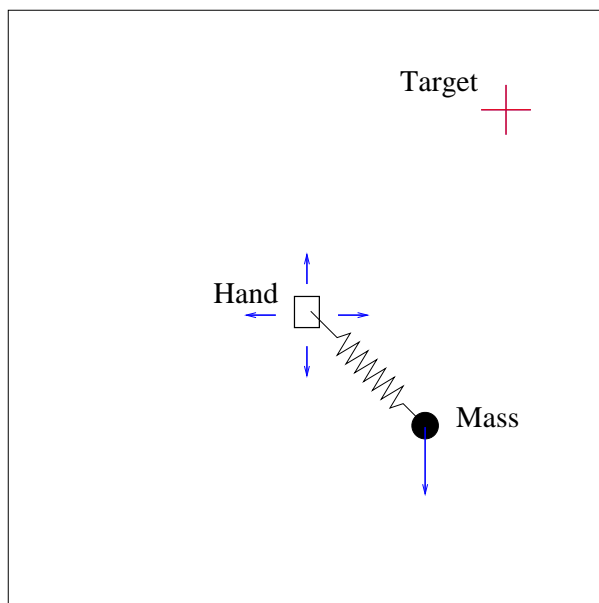
→ the use of stochastic policies compensate the lack of knowledge about the state dynamics.

→ Related to the observation of an “oscillatory behavior” in humans when learning new motor tasks?

Example: target problem with a spring

Space 6 dimension: **state** x_0, y_0, x, y, v_x, v_y .

Control: move direction of the hand. **Goal:** reach target at time $T = 1$.



Maximize reward $r = -x_0^2 - y_0^2 - (x - 2)^2 - (y - 2)^2$. Stochastic policy:

$$\pi_\alpha(u|x, t) = \frac{e^{Q_\alpha(x, t, u)}}{\sum_v e^{Q_\alpha(x, t, v)}} \text{ with } Q_\alpha \text{ linear in the state variables.}$$