Q-Learning
and Pontryagin's Minimum Principle

Sean Meyn

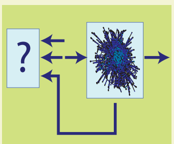
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois

Joint work with Prashant Mehta

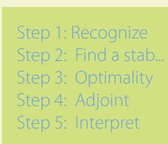
Research support: NSF: ECS-0523620
AFOSR: FA9550-09-1-0190

ECE ILLINOIS
Department of Electrical and Computer Engineering

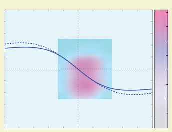
Outline



Coarse models - what to do with them?



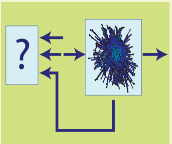
Q-learning for nonlinear state space models



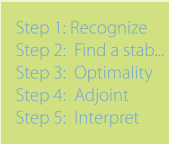
Example: Local approximation

Example: Decentralized control

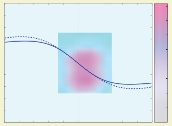
Outline



Coarse models - what to do with them?



Q-learning for nonlinear state space models

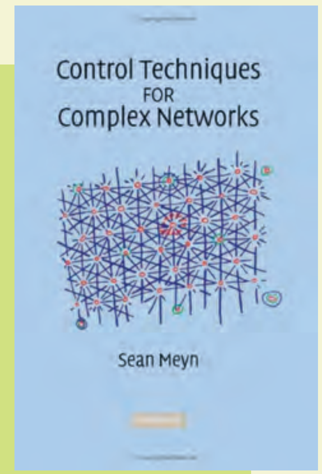


Example: Local approximation

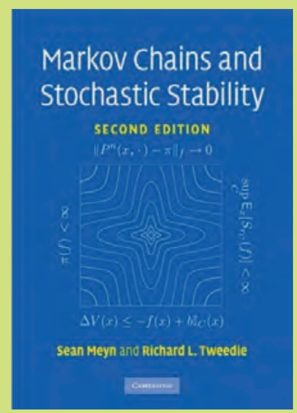
Example: Decentralized control

Coarse Models: *A rich collection of model reduction techniques*

Many of today's participants have contributed to this research.
A biased list:

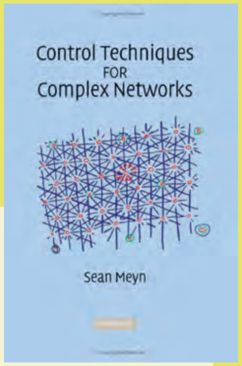


- *Fluid models*: Law of Large Numbers scaling, most likely paths in large deviations
- *Workload relaxation* for networks
  Heavy-traffic limits



- *Clustering*: spectral graph theory
  Markov spectral theory
- *Singular perturbations*
- *Large population limits*: Interacting particle systems

Workload Relaxations



An example from CTCN:

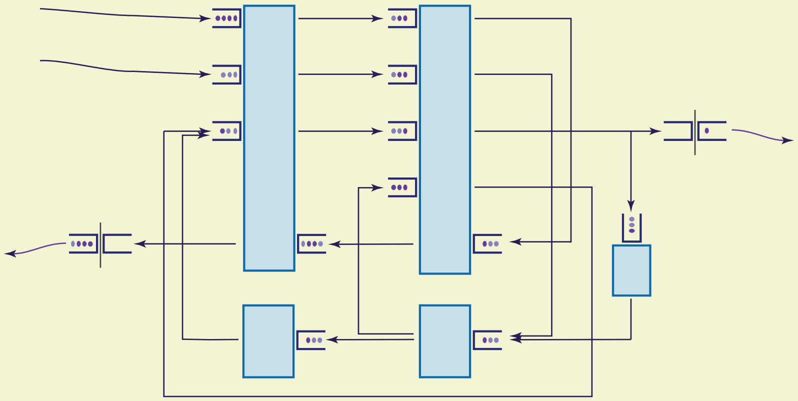


Figure 7.1: Demand-driven model with routing, scheduling, and re-work.

Workload at two stations evolves as a two-dimensional system
Cost is projected onto these coordinates:

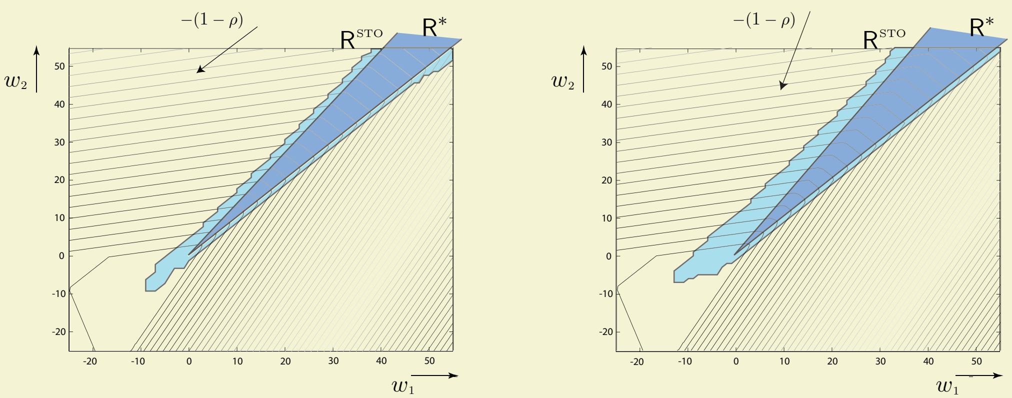
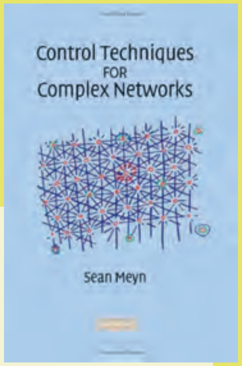


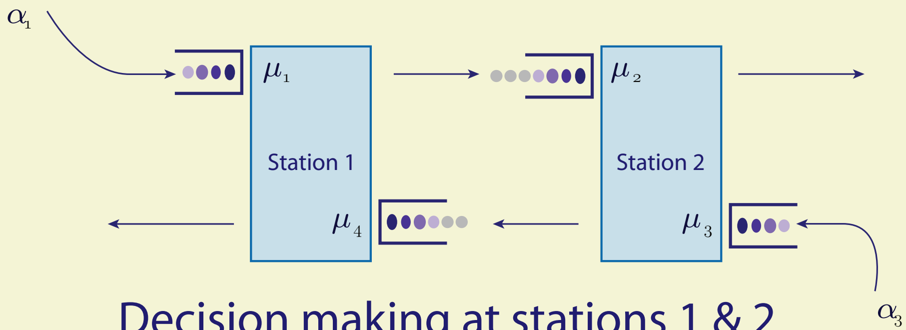
Figure 7.2: Optimal policies for two instances of the network shown in Figure 7.1. In each figure the optimal stochastic control region $\mathsf{R}^{\mathrm{STO}}$ is compared with the optimal region R^* obtained for the two dimensional fluid model.

Optimal policy for relaxation = hedging policy for full network

Workload Relaxations and Simulation



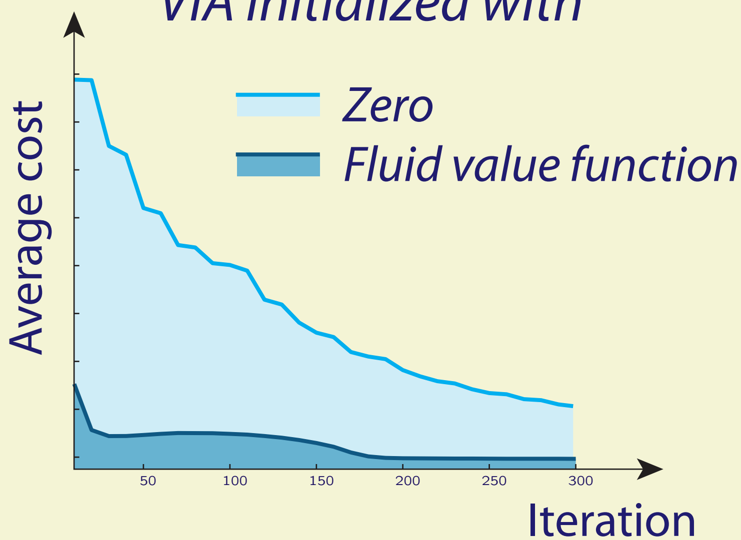
An example from CTCN:



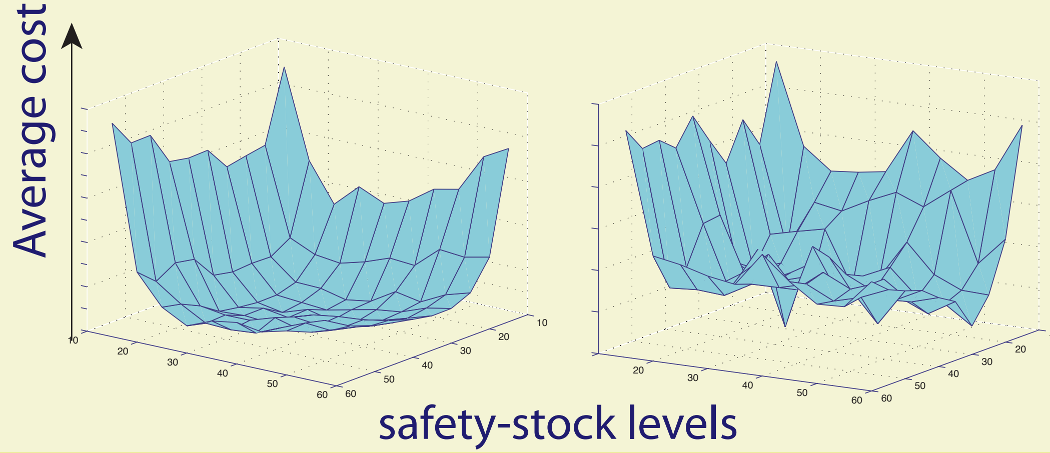
Decision making at stations 1 & 2
e.g., setting safety-stock levels

DP and simulations accelerated
using *fluid value function* for *workload relaxation*

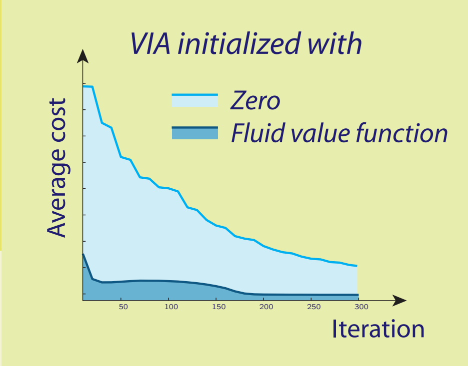
VIA initialized with



Simulated mean with
and without control variate:



What To Do With a Coarse Model?

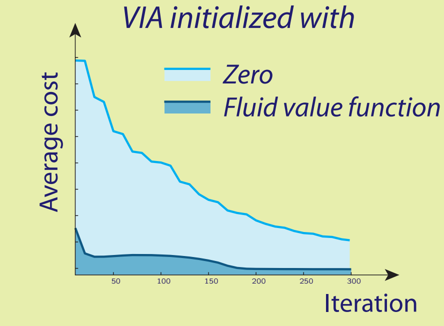


Setting: we have qualitative or partial quantitative insight regarding optimal control

The network examples relied on specific network structure

What about other models?

What To Do With a Coarse Model?



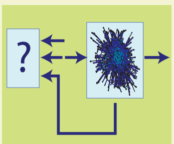
Setting: we have qualitative or partial quantitative insight regarding optimal control

The network examples relied on specific network structure

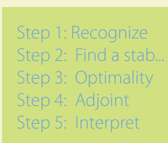
What about other models?

An answer lies in a new formulation of Q-learning

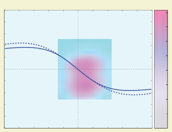
Outline



Coarse models - what to do with them?



Q-learning for nonlinear state space models

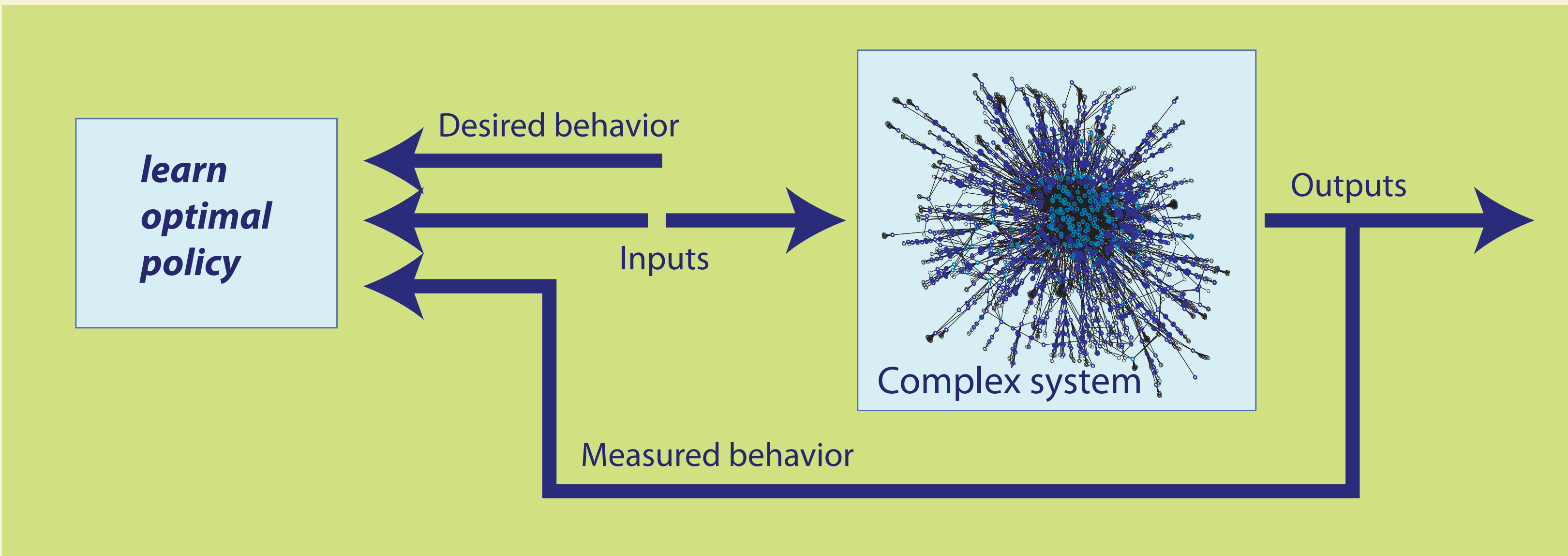


Example: Local approximation

Example: Decentralized control

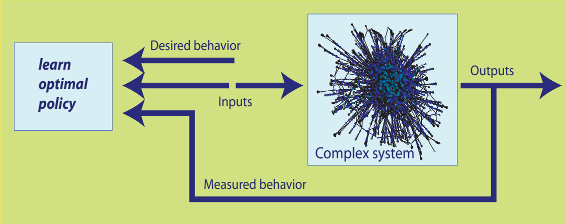
What is Q learning?

Identify optimal policy based on observations:



Watkin’s 1992 formulation applied to finite state space MDPs

What is Q learning?



Watkin's 1992 formulation applied to finite state space MDPs

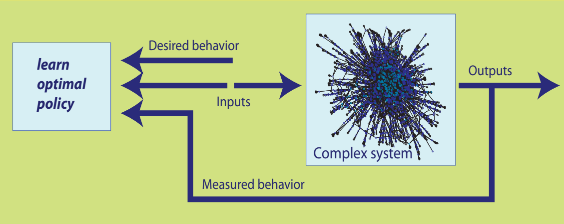
Watkins and P. Dayan, 1992

Goal: Find the best approximation to dynamic programming equations over a parameterized class, based on observations using a non-optimal policy.

Watkin's algorithm known to be effective only for

- Finite state-action space
- Complete parametric family

What is Q learning?



Watkin's 1992 formulation applied to finite state space MDPs

Watkins and P. Dayan, 1992

Goal: Find the best approximation to dynamic programming equations over a parameterized class, based on observations using a non-optimal policy.

Watkin's algorithm known to be effective only for
Finite state-action space
Complete parametric family

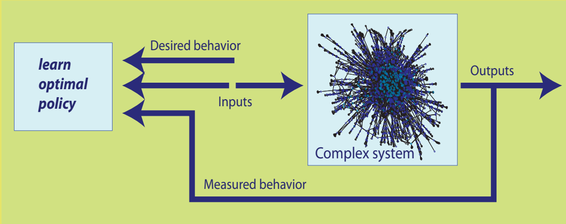
Extensions: when cost depends on control, but dynamics are oblivious

Duff, 1995
Tsitsiklis and Van Roy, 1999
Yu and Bertsekas, 2007

Approach: Similar to *differential dynamic programming*

Differential dynamic programming
D. H. Jacobson and D. Q. Mayne
American Elsevier Pub. Co. 1970

What is Q learning?



Watkin's 1992 formulation applied to finite state space MDPs

This lecture:

Deterministic formulation: Nonlinear system on Euclidean space,

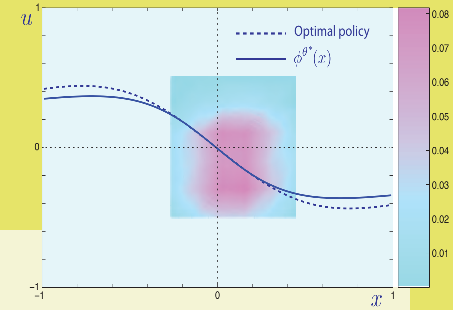
$$\frac{d}{dt}x(t) = f(x(t), u(t)), \qquad t \geq 0$$

Infinite-horizon discounted cost criterion,

$$J^*(x) = \inf \int_0^\infty e^{-\gamma s} c(x(s), u(s))\, ds, \qquad x(0) = x$$

with c a non-negative cost function.

What is Q learning?



Deterministic formulation: Nonlinear system on Euclidean space,

$$\frac{d}{dt}x(t) = f(x(t), u(t)), \qquad t \geq 0$$

Infinite-horizon discounted cost criterion,

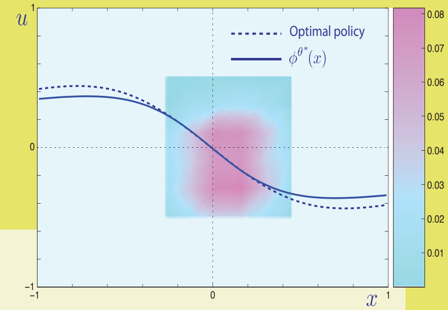
$$J^*(x) = \inf \int_0^\infty e^{-\gamma s} c(x(s), u(s))\, ds, \qquad x(0) = x$$

with c a non-negative cost function.

Differential generator: For any smooth function h,

$$\mathcal{D}_u h\,(x) := (\nabla h\,(x))^{T} f\,(x, u)$$

What is Q learning?



Deterministic formulation: Nonlinear system on Euclidean space,

$$\tfrac{d}{dt}x(t) = f(x(t), u(t)), \qquad t \geq 0$$

Infinite-horizon discounted cost criterion,

$$J^*(x) = \inf \int_0^\infty e^{-\gamma s} c(x(s), u(s))\, ds, \qquad x(0) = x$$

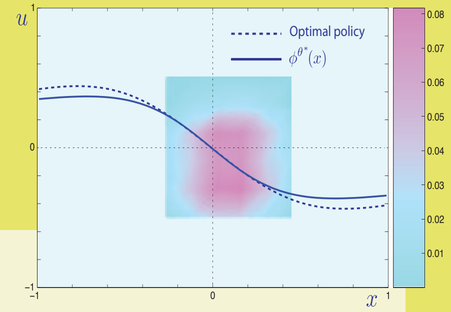
with c a non-negative cost function.

Differential generator: For any smooth function h,

$$\mathcal{D}_u h\,(x) := (\nabla h\,(x))^{\intercal} f(x, u)$$

HJB equation: $\quad \min_u \bigl(c(x, u) + \mathcal{D}_u J^*\,(x)\bigr) = \gamma J^*(x)$

What is Q learning?



Deterministic formulation: Nonlinear system on Euclidean space,

$$\frac{d}{dt}x(t) = f(x(t), u(t)), \qquad t \ge 0$$

Infinite-horizon discounted cost criterion,

$$J^*(x) = \inf \int_0^\infty e^{-\gamma s} c(x(s), u(s))\, ds, \qquad x(0) = x$$

with c a non-negative cost function.

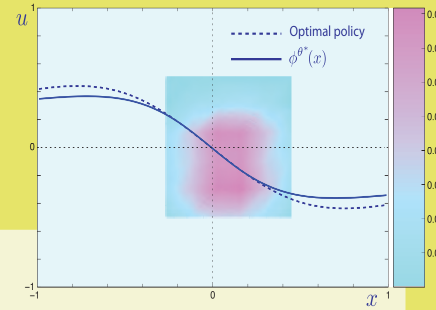
Differential generator: For any smooth function h,

$$\mathcal{D}_u h(x) := (\nabla h(x))^{\intercal} f(x, u)$$

HJB equation: $$\min_u \Big(c(x, u) + \mathcal{D}_u J^*(x) \Big) = \gamma J^*(x)$$

The *Q-function* of Q-learning is this function of two variables

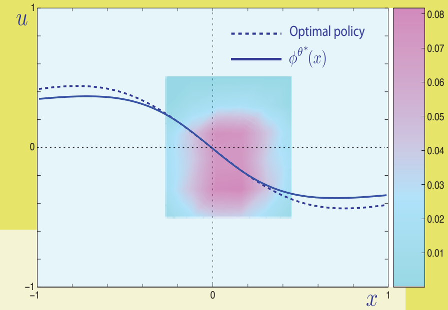
Q learning - Steps towards an algorithm



Sequence of five steps:

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



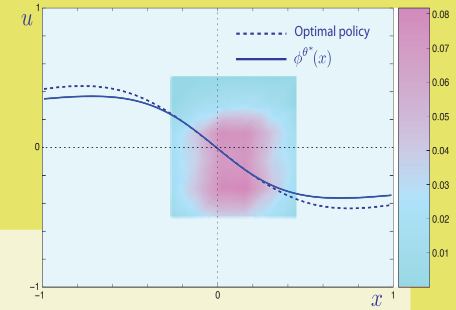
Sequence of five steps:

- Step 1: Recognize fixed point equation for the Q-function
- Step 2: Find a stabilizing policy that is ergodic
- Step 3: Optimality criterion - minimize Bellman error
- Step 4: Adjoint operation
- Step 5: Interpret and simulate!

Goal - seek the best approximation,
within a parameterized class

$$H^\theta(x,u) = \theta^T \psi(x,u), \qquad \theta \in \mathbb{R}^d$$

Q learning - Steps towards an algorithm



Step 1: Recognize fixed point equation for the Q-function

Q-function: $$H^*(x,u) = c(x,u) + \mathcal{D}_u J^*(x)$$

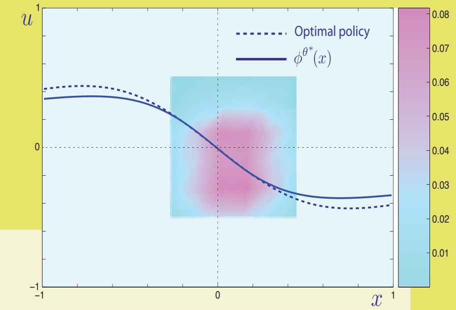
Its minimum: $$\underline{H}^*(x) := \min_{u \in \mathsf{U}} H^*(x,u) = \gamma J^*(x)$$

Fixed point equation:

$$\mathcal{D}_u \underline{H}^*(x) = -\gamma(c(x,u) - H^*(x,u))$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



Step 1: Recognize fixed point equation for the Q-function

Q-function: $$H^*(x,u) = c(x,u) + \mathcal{D}_u J^*(x)$$

Its minimum: $$\underline{H}^*(x) := \min_{u \in \mathsf{U}} H^*(x,u) = \gamma J^*(x)$$

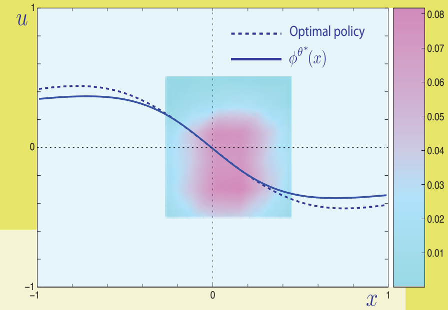
Fixed point equation:

$$\mathcal{D}_u \underline{H}^*(x) = -\gamma\big(c(x,u) - H^*(x,u)\big)$$

Key observation for learning: For any input-output pair,

$$\mathcal{D}_u \underline{H}^*(x) = \frac{d}{dt}\underline{H}^*(x(t))\Big|_{\substack{x=x(t)\\ u=u(t)}}$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!



Q learning - LQR example

Linear model and quadratic cost,

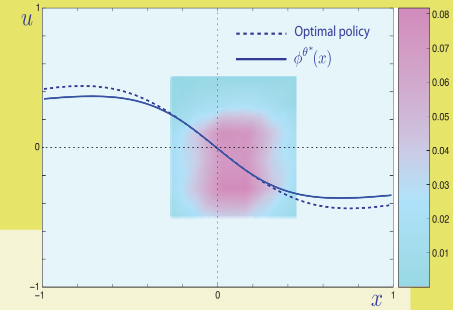
Cost: $$c(x,u) \;=\; \tfrac{1}{2}x^TQx \;+\; \tfrac{1}{2}u^TRu$$

Q-function: $$H^*(x) = c(x,u) + (Ax + Bu)^T P^* x$$

Solves Riccatti eqn

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - LQR example



Linear model and quadratic cost,

Cost: $$c(x,u) \;=\; \tfrac{1}{2}x^{\scriptscriptstyle T}Qx \;+\; \tfrac{1}{2}u^{\scriptscriptstyle T}Ru$$

Q-function: $$H^*(x) = c(x,u) + (Ax+Bu)^{\scriptscriptstyle T}P^*x$$

Solves Riccati eqn

Q-function approx:

$$H^\theta(x,u) = c(x,u) + \tfrac{1}{2}\sum_{i=1}^{d_x}\theta_i^x x^{\scriptscriptstyle T}E^i x + \sum_{j=1}^{d_{xu}}\theta_j^x x^{\scriptscriptstyle T}F^i u$$

Minimum:

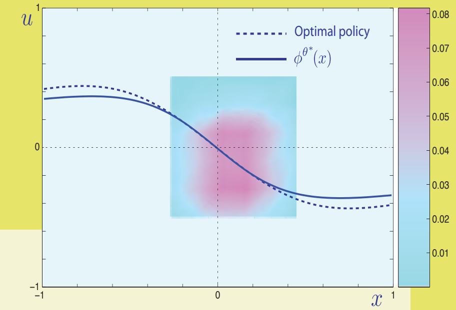
$$\underline{H}^\theta(x) = \tfrac{1}{2}x^{\scriptscriptstyle T}\Big(Q + E^\theta - F^{\theta^{\scriptscriptstyle T}}R^{-1}F^\theta\Big)\,x$$

Minimizer:

$$u^\theta(x) = \phi^\theta(x) = -R^{-1}F^\theta x$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



Step 2: Stationary policy that is ergodic?

Assume the LLN holds for continuous functions

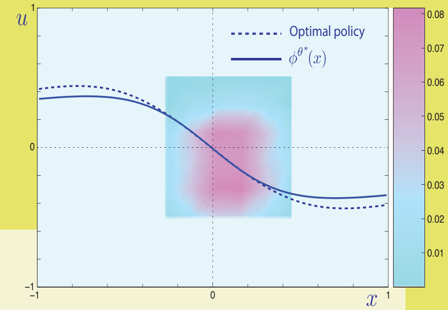
$$F\colon \mathbb{R}^{\ell} \times \mathbb{R}^{\ell_u} \to \mathbb{R}$$

As $T \to \infty$,

$$\frac{1}{T}\int_0^T F(x(t),u(t))\,dt \longrightarrow \int_{\mathsf{X}\times\mathsf{U}} F(x,u)\,\varpi(dx,du)$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



Step 2: Stationary policy that is ergodic?

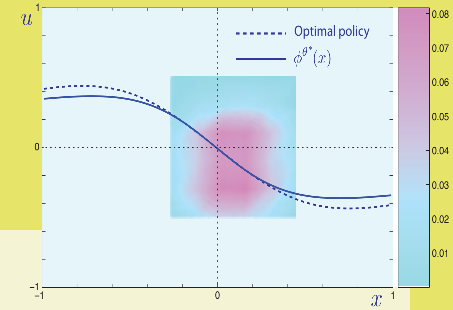
Suppose for example the input is scalar, and the system is *stable*

[Bounded-input/Bounded-state]

Can try a linear combination of sinusouids

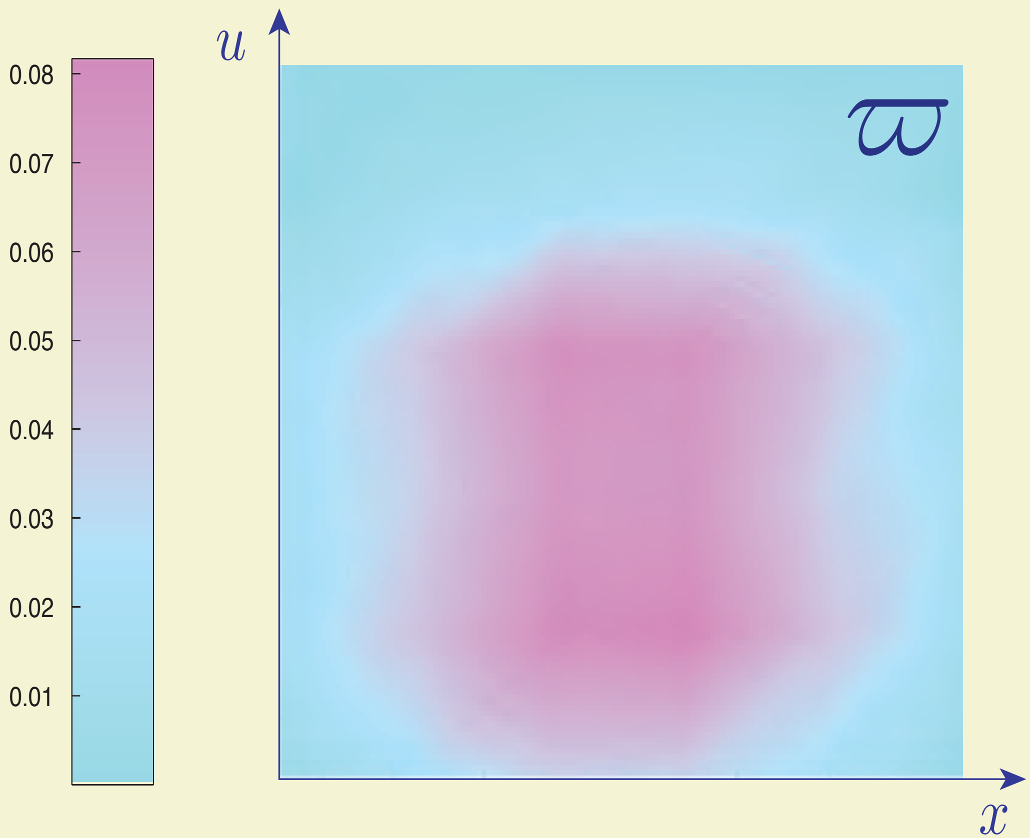
Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



Step 2: Stationary policy that is ergodic?

Suppose for example the input is scalar, and the system is *stable*
[Bounded-input/Bounded-state]

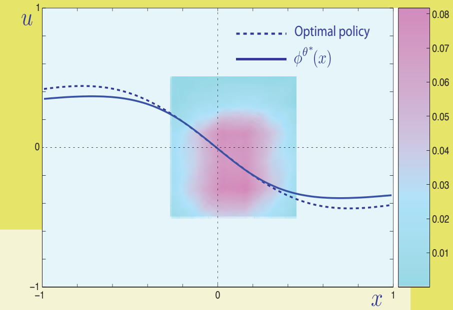


Can try a linear combination of sinusoids

$$u(t) = A(\sin(t) + \sin(\pi t) + \sin(et))$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



Step 3: Bellman error

$$\mathcal{L}^{\theta}(x,u) \quad := \quad \mathcal{D}_u \underline{H}^{\theta}(x) + \gamma(c - H^{\theta}), \qquad \theta \in \mathbb{R}^d$$

Based on observations, minimize the mean-square Bellman error:

$$\mathcal{E}_{\mathrm{Bell}}(\theta) \quad := \quad \int [\mathcal{L}^{\theta}]^2 \varpi(dx,du) \;:=\; \langle \mathcal{L}^{\theta},\, \mathcal{L}^{\theta} \rangle_{\varpi}$$

First order condition for optimality: $\langle \mathcal{L}^{\theta}, \mathcal{D}_u \underline{\psi}_i^{\theta} - \gamma \psi_i^{\theta} \rangle_{\varpi} = 0$

with $\underline{\psi}_i^{\theta}(x) = \psi_i^{\theta}(x, \phi^{\theta}(x))$,

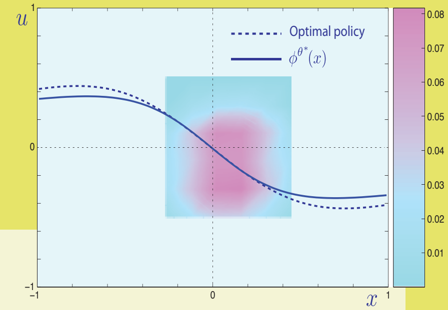
$1 \le i \le d$

$$\mathcal{D}_u \underline{H}^{\theta}(x) = \frac{d}{dt} \underline{H}^{\theta}(x(t)) \Big|_{\substack{x=x(t)\\ u=u(t)}}$$

$$\mathcal{D}_u \underline{\psi}_i^{\theta}(x) = \frac{d}{dt} \underline{\psi}_i^{\theta}(x(t)) \Big|_{\substack{x=x(t)\\ u=u(t)}}$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Convex Reformulation



Step 3: Bellman error

$$\mathcal{L}^\theta(x,u) \quad := \quad \mathcal{D}_u \underline{H}^\theta(x) + \gamma(c - H^\theta), \qquad \theta \in \mathbb{R}^d$$

Based on observations, minimize the mean-square Bellman error:

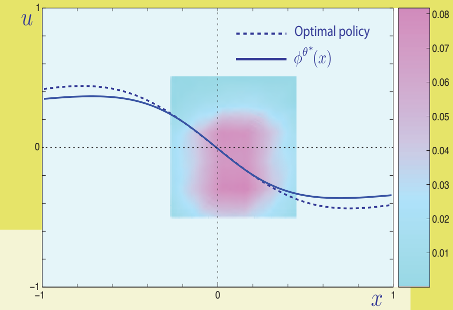
$$\mathcal{E}_{\mathrm{Bell}}(\theta) \quad := \quad \int [\mathcal{L}^\theta]^2 \varpi(dx, du) := \langle \mathcal{L}^\theta, \mathcal{L}^\theta \rangle_\varpi$$

$$\mathcal{L}^\theta(x,u) \quad := \quad \mathcal{D}_u G^\theta(x) + \gamma(c - H^\theta), \qquad \theta \in \mathbb{R}^d$$

$$G^\theta(x) \le H^\theta(x,u), \qquad \text{all } x,\ u$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - LQR example



Linear model and quadratic cost,

Cost: $$c(x,u) = \tfrac{1}{2}x^TQx + \tfrac{1}{2}u^TRu$$

Q-function: $$H^*(x) = c(x,u) + (Ax+Bu)^T P^* x$$

Solves Riccati eqn

Q-function approx:

$$H^\theta(x,u) = c(x,u) + \frac{1}{2}\sum_{i=1}^{d_x} \theta_i^x x^T E^i x + \sum_{j=1}^{d_{xu}} \theta_j^x x^T F^i u$$

Approximation to minimum

$$G^\theta(x) = \tfrac{1}{2}x^T G^\theta x$$

Minimizer:

$$u^\theta(x) = \phi^\theta(x) = -R^{-1}F^\theta x$$

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm

$$\cancel{\begin{aligned}\mathcal{D}_u \underline{H}^\theta(x) &= \tfrac{d}{dt}\underline{H}^\theta(x(t)) \\ \mathcal{D}_u \underline{\psi}_i^\theta(x) &= \tfrac{d}{dt}\underline{\psi}_i^\theta(x(t))\end{aligned}}$$

Step 4: Causal smoothing to avoid differentiation

For any function of two variables, $g : \mathbb{R}^\ell \times \mathbb{R}^{\ell_w} \to \mathbb{R}$
Resolvent gives a new function,

$$R_\beta g\,(x, w) = \int_0^\infty e^{-\beta t} g(x(t), \xi(t))\, dt$$

Skip to examples

$$\cancel{\mathcal{D}_u \underline{H}^\theta(x) = \tfrac{d}{dt}\underline{H}^\theta(x(t))}$$

$$\cancel{\mathcal{D}_u \underline{\psi}_i^\theta(x) = \tfrac{d}{dt}\underline{\psi}_i^\theta(x(t))}$$

Q learning - Steps towards an algorithm

Step 4: Causal smoothing to avoid differentiation

For any function of two variables, $g : \mathbb{R}^\ell \times \mathbb{R}^{\ell_w} \to \mathbb{R}$
Resolvent gives a new function,

$$R_\beta g\,(x, w) = \int_0^\infty e^{-\beta t} g(x(t), \xi(t))\, dt \;, \qquad \beta > 0$$

controlled using the nominal policy

$$u(t) = \phi(x(t), \xi(t)), \qquad t \geq 0$$

stabilizing & ergodic

Q learning - Steps towards an algorithm

$$\cancel{\mathcal{D}_u \underline{H}^\theta(x) = \frac{d}{dt}\underline{H}^\theta(x(t))}$$

$$\cancel{\mathcal{D}_u \underline{\psi}_i^\theta(x) = \frac{d}{dt}\underline{\psi}_i^\theta(x(t))}$$

Step 4: Causal smoothing to avoid differentiation

For any function of two variables, $g : \mathbb{R}^\ell \times \mathbb{R}^{\ell_w} \to \mathbb{R}$
Resolvent gives a new function,

$$R_\beta g\,(x,w) = \int_0^\infty e^{-\beta t} g(x(t),\xi(t))\,dt \ , \qquad \beta > 0$$

Resolvent equation:

$$R_\beta \mathcal{D} = \beta R_\beta - I$$

Q learning - Steps towards an algorithm

$$\cancel{\begin{aligned}\mathcal{D}_u \underline{H}^\theta(x) &= \tfrac{d}{dt}\underline{H}^\theta(x(t)) \\ \mathcal{D}_u \underline{\psi}_i^\theta(x) &= \tfrac{d}{dt}\underline{\psi}_i^\theta(x(t))\end{aligned}}$$

Step 4: Causal smoothing to avoid differentiation

For any function of two variables, $g : \mathbb{R}^\ell \times \mathbb{R}^{\ell_w} \to \mathbb{R}$

Resolvent gives a new function,

$$R_\beta g\,(x, w) = \int_0^\infty e^{-\beta t} g(x(t), \xi(t))\, dt \ , \qquad \beta > 0$$

Resolvent equation:

$$R_\beta \mathcal{D} = \beta R_\beta - I$$

Smoothed Bellman error:

$$\begin{aligned}\mathcal{L}^{\theta,\beta} &= R_\beta \mathcal{L}^\theta \\ &= [\beta R_\beta - I]\underline{H}^\theta + \gamma R_\beta (c - H^\theta)\end{aligned}$$

Q learning - Steps towards an algorithm

$$\mathcal{D}_u \underline{H}^\theta\,(x) = \tfrac{d}{dt}\underline{H}^\theta(x(t))$$

$$\mathcal{D}_u \underline{\psi}_i^\theta\,(x) = \tfrac{d}{dt}\underline{\psi}_i^\theta(x(t))$$

Smoothed Bellman error:

$$\mathcal{E}_\beta(\theta) \;:=\; \tfrac{1}{2}\|\mathcal{L}^{\theta,\beta}\|^2_\varpi$$

$$\nabla\mathcal{E}_\beta(\theta) = \langle \mathcal{L}^{\theta,\beta}, \nabla_\theta \mathcal{L}^{\theta,\beta}\rangle_\varpi$$

$=$ *zero* at an optimum

Step 4: Causal smoothing to avoid differentiation

Q learning - Steps towards an algorithm

$$\cancel{\mathcal{D}_u \underline{H}^\theta(x) = \tfrac{d}{dt}\underline{H}^\theta(x(t))}$$

$$\cancel{\mathcal{D}_u \underline{\psi}_i^\theta(x) = \tfrac{d}{dt}\underline{\psi}_i^\theta(x(t))}$$

Smoothed Bellman error:

$$\mathcal{E}_\beta(\theta) \;:=\; \tfrac{1}{2}\|\mathcal{L}^{\theta,\beta}\|_\varpi^2$$

$$\nabla\mathcal{E}_\beta(\theta) = \langle \mathcal{L}^{\theta,\beta}, \nabla_\theta \mathcal{L}^{\theta,\beta}\rangle_\varpi$$

$=$ *zero* at an optimum

Involves terms of the form $\langle R_\beta g, R_\beta h\rangle$

Step 4: Causal smoothing to avoid differentiation

Q learning - Steps towards an algorithm

$$\cancel{\mathcal{D}_u \underline{H}^\theta(x) = \tfrac{d}{dt}\underline{H}^\theta(x(t))}$$

$$\cancel{\mathcal{D}_u \underline{\psi}_i^\theta(x) = \tfrac{d}{dt}\underline{\psi}_i^\theta(x(t))}$$

Smoothed Bellman error: $\mathcal{E}_\beta(\theta) := \frac{1}{2}\|\mathcal{L}^{\theta,\beta}\|_\varpi^2$

$$\nabla\mathcal{E}_\beta(\theta) = \langle \mathcal{L}^{\theta,\beta}, \nabla_\theta \mathcal{L}^{\theta,\beta}\rangle_\varpi$$

Adjoint operation:

$$R_\beta^\dagger R_\beta = \frac{1}{2\beta}\left(R_\beta^\dagger + R_\beta\right)$$

$$\langle R_\beta g, R_\beta h\rangle = \frac{1}{2\beta}\big(\langle g, R_\beta^\dagger h\rangle + \langle h, R_\beta^\dagger g\rangle\big)$$

Step 4: Causal smoothing to avoid differentiation

$$\mathcal{D}_u \underline{H}^\theta(x) = \frac{d}{dt}\underline{H}^\theta(x(t))$$

$$\mathcal{D}_u \underline{\psi}_i^\theta(x) = \frac{d}{dt}\underline{\psi}_i^\theta(x(t))$$

Q learning - Steps towards an algorithm

Smoothed Bellman error: $\mathcal{E}_\beta(\theta) \;:=\; \frac{1}{2}\|\mathcal{L}^{\theta,\beta}\|^2_\varpi$

$$\nabla \mathcal{E}_\beta(\theta) = \langle \mathcal{L}^{\theta,\beta}, \nabla_\theta \mathcal{L}^{\theta,\beta}\rangle_\varpi$$

Adjoint operation:

$$R_\beta^\dagger R_\beta = \frac{1}{2\beta}\,(R_\beta^\dagger + R_\beta)$$

$$\langle R_\beta g, R_\beta h\rangle = \frac{1}{2\beta}\big(\langle g, R_\beta^\dagger h\rangle + \langle h, R_\beta^\dagger g\rangle\big)$$

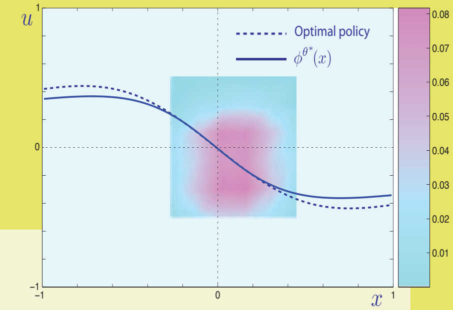
Adjoint realization: *time-reversal*

$$R_\beta^\dagger g\,(x,w) = \int_0^\infty e^{-\beta t}\mathsf{E}_{x,\,w}[g(x^\circ(-t),\xi^\circ(-t))]\,dt$$

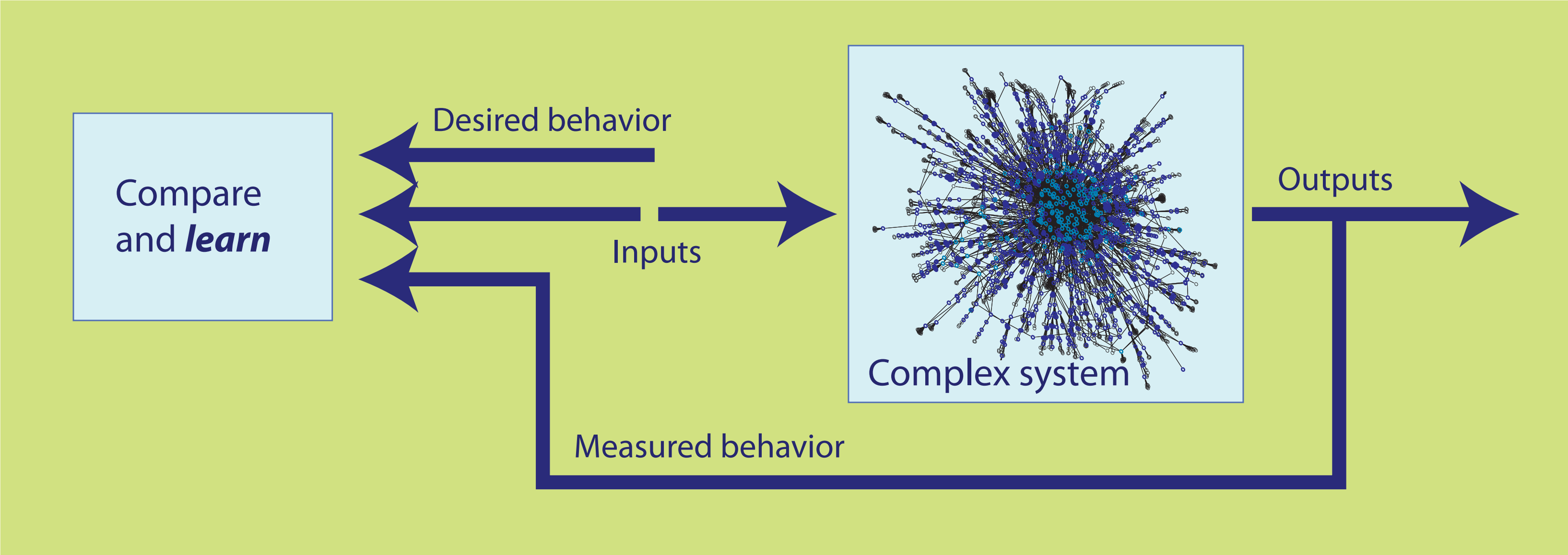
expectation conditional on $x^\circ(0) = x,\ \xi^\circ(0) = w.$

Step 4: Causal smoothing to avoid differentiation

Q learning - Steps towards an algorithm



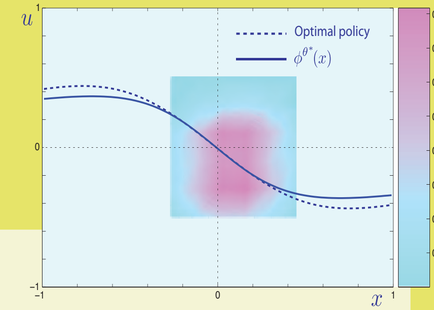
After Step 5: Not quite adaptive control:



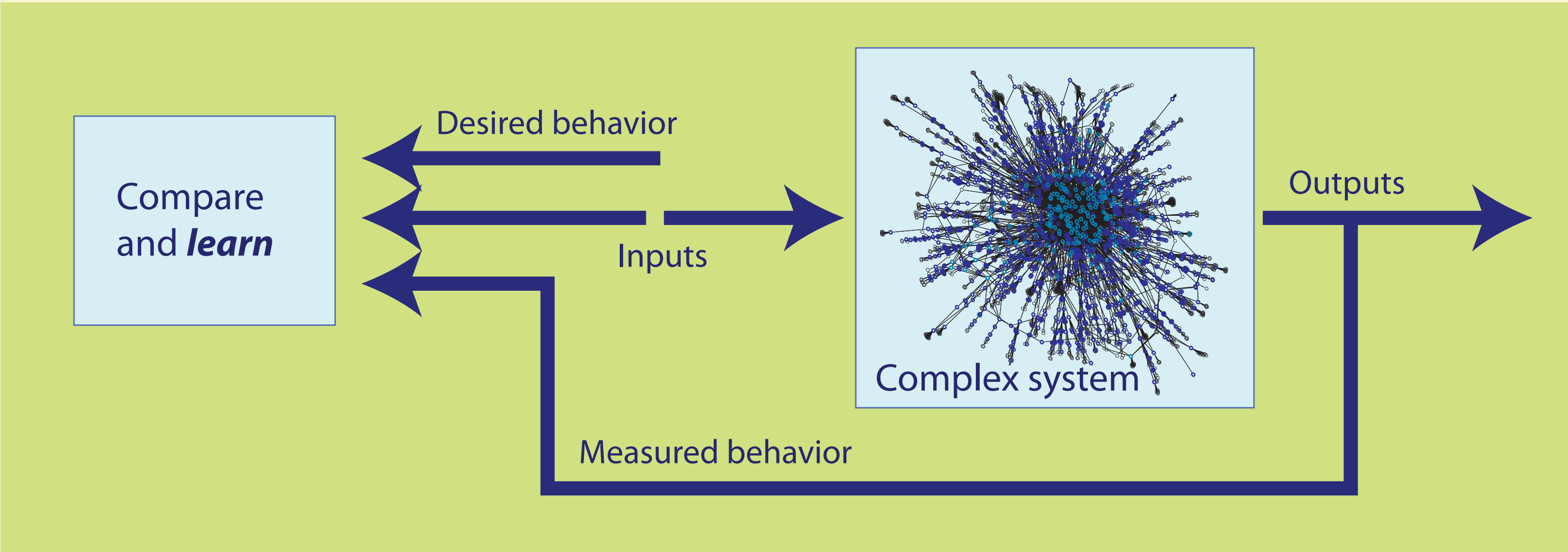
Ergodic input applied

Step 1: Recognize fixed point equation for the Q-function
Step 2: Find a stabilizing policy that is ergodic
Step 3: Optimality criterion - minimize Bellman error
Step 4: Adjoint operation
Step 5: Interpret and simulate!

Q learning - Steps towards an algorithm



After Step 5: Not quite adaptive control:



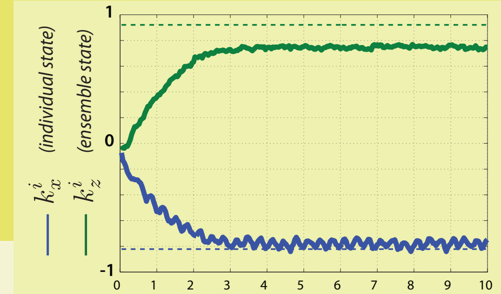
Ergodic input applied

Based on observations minimize the mean-square Bellman error:

$$\mathcal{E}_{\rm Bell}(\theta) \quad := \quad \int \big[\mathcal{L}^\theta\big]^2 \varpi(dx, du)$$

$$\mathcal{L}^\theta(x,u) \quad := \quad \mathcal{D}_u \underline{H}^\theta(x) + \gamma(c - H^\theta)\,, \qquad \theta \in \mathbb{R}^d$$

Deterministic Stochastic Approximation



Gradient descent:

$$\frac{d}{dt}\theta = -\varepsilon \langle \mathcal{L}^\theta, \mathcal{D}_u \nabla_\theta \underline{H}^\theta - \gamma \nabla_\theta H^\theta \rangle_\varpi$$

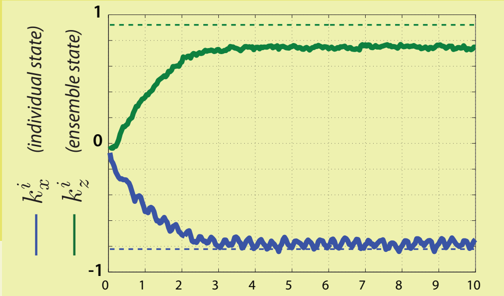
Converges* to the minimizer of the mean-square Bellman error:

$$\begin{aligned} \mathcal{E}_{\mathrm{Bell}}(\theta) \quad &:= \quad \int \big[\mathcal{L}^\theta\big]^2 \varpi(dx, du) \\ \mathcal{L}^\theta(x,u) \quad &:= \quad \mathcal{D}_u \underline{H}^\theta(x) + \gamma(c - H^\theta) \end{aligned}$$

$$\frac{d}{dt} h(x(t)) \Big|_{\substack{x = x(t) \\ w = \xi(t)}} = \mathcal{D}_u h(x)$$

* Convergence observed in experiments!
For a convex re-formulation of
the problem, see Mehta & Meyn 2009

Deterministic Stochastic Approximation



Stochastic Approximation

$$\frac{d}{dt}\theta = -\varepsilon_t \mathcal{L}_t^\theta \Big(\frac{d}{dt}\nabla_\theta \underline{H}^\theta\,(x^\circ(t)) - \gamma \nabla_\theta H^\theta(x^\circ(t), u^\circ(t))\Big)$$

$$\mathcal{L}_t^\theta := \frac{d}{dt}\underline{H}^\theta\,(x^\circ(t)) + \gamma(c(x^\circ(t), u^\circ(t)) - H^\theta(x^\circ(t), u^\circ(t)))$$

Gradient descent:

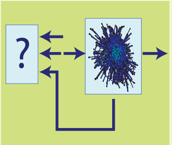
$$\frac{d}{dt}\theta = -\varepsilon \langle \mathcal{L}^\theta, \mathcal{D}_u \nabla_\theta \underline{H}^\theta - \gamma \nabla_\theta H^\theta \rangle_\varpi$$

Mean-square Bellman error:

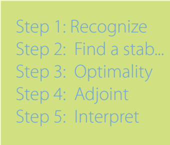
$$\begin{aligned} \mathcal{E}_{\text{Bell}}(\theta) &:= \int [\mathcal{L}^\theta]^2\, \varpi(dx, du) \\ \mathcal{L}^\theta(x,u) &:= \mathcal{D}_u \underline{H}^\theta\,(x) + \gamma(c - H^\theta) \end{aligned}$$

$$\frac{d}{dt} h(x(t)) \Big|_{\substack{x=x(t)\\ w=\xi(t)}} = \mathcal{D}_u h\,(x)$$

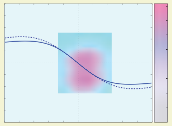
Outline



Coarse models - what to do with them?



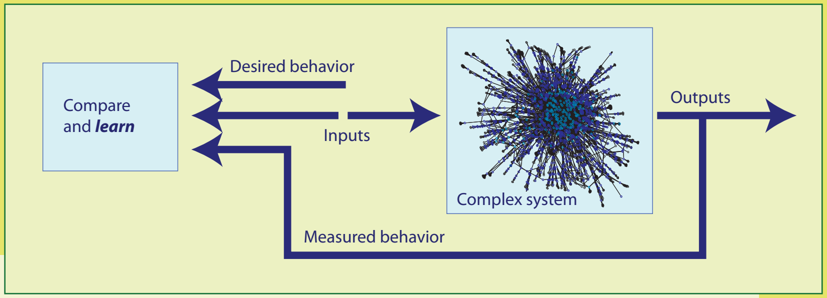
Q-learning for nonlinear state space models



Example: Local approximation

Example: Decentralized control

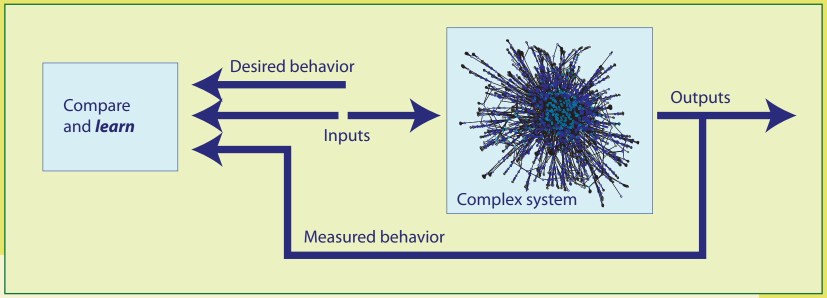
Q learning - Local Learning



Cubic nonlinearity:

$$\frac{d}{dt}x = -x^3 + u, \qquad c(x,u) = \frac{1}{2}x^2 + \frac{1}{2}u^2$$

Q learning - Local Learning

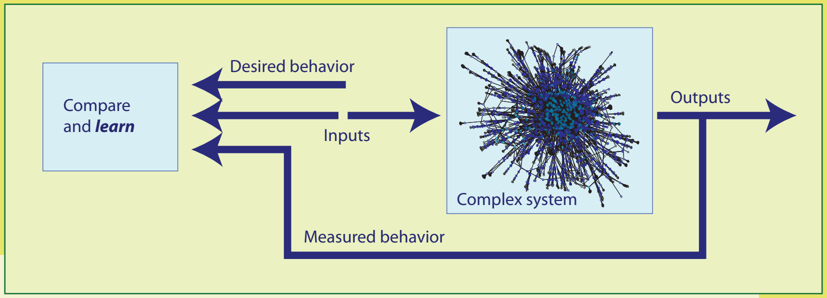


Cubic nonlinearity: $\frac{d}{dt}x = -x^3 + u, \qquad c(x,u) = \frac{1}{2}x^2 + \frac{1}{2}u^2$

HJB:

$$\min_u \Big(\tfrac{1}{2}x^2 + \tfrac{1}{2}u^2 + (-x^3 + u)\nabla J^*(x) \Big) = \gamma J^*(x)$$

Q learning - Local Learning



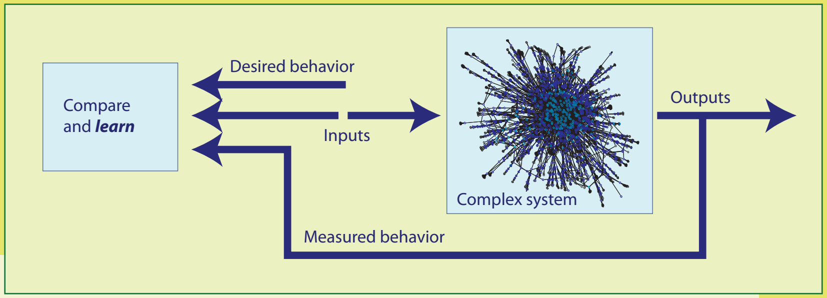
Cubic nonlinearity: $\frac{d}{dt}x = -x^3 + u, \qquad c(x,u) = \frac{1}{2}x^2 + \frac{1}{2}u^2$

HJB: $\min_u \Big(\frac{1}{2}x^2 + \frac{1}{2}u^2 + (-x^3 + u)\nabla J^*(x) \Big) = \gamma J^*(x)$

Basis:

$$H^\theta(x,u) = c(x,u) + \theta^x x^2 + \theta^{xu} \frac{x}{1+2x^2} u$$

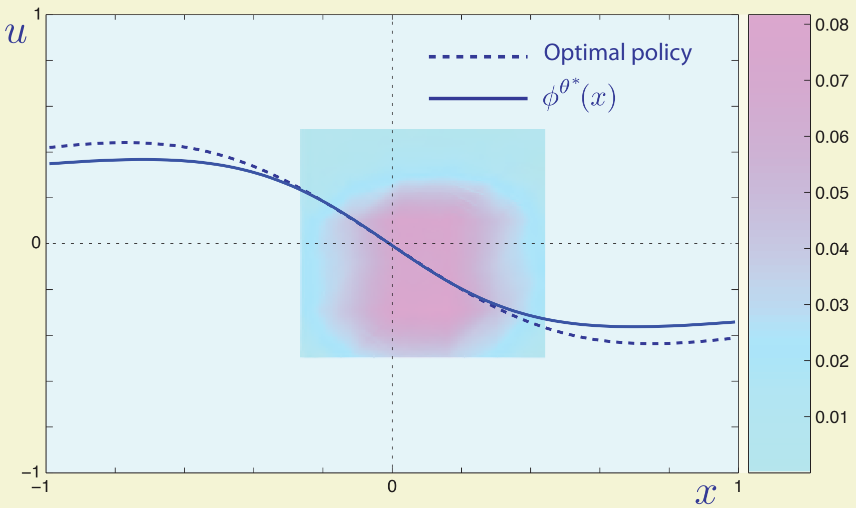
Q learning - Local Learning



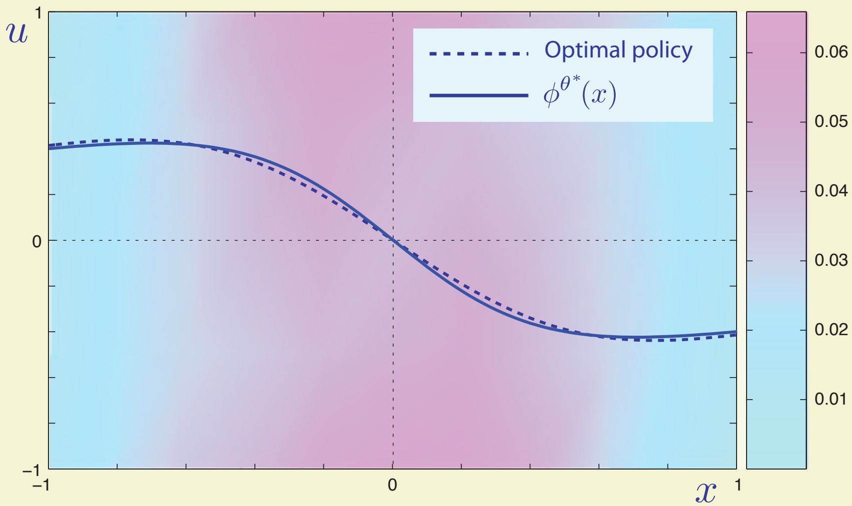
Cubic nonlinearity: $\frac{d}{dt}x = -x^3 + u, \qquad c(x,u) = \frac{1}{2}x^2 + \frac{1}{2}u^2$

HJB: $\min_u \left(\frac{1}{2}x^2 + \frac{1}{2}u^2 + (-x^3 + u)\nabla J^*(x) \right) = \gamma J^*(x)$

Basis: $H^\theta(x,u) = c(x,u) + \theta^x x^2 + \theta^{xu} \frac{x}{1+2x^2} u$



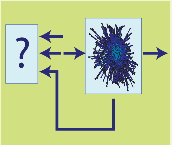
Low amplitude input



High amplitude input

$$u(t) = A(\sin(t) + \sin(\pi t) + \sin(et))$$

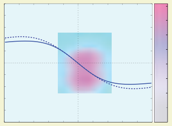
Outline



Coarse models - what to do with them?

Step 1: Recognize
Step 2: Find a stab...
Step 3: Optimality
Step 4: Adjoint
Step 5: Interpret

Q-learning for nonlinear state space models



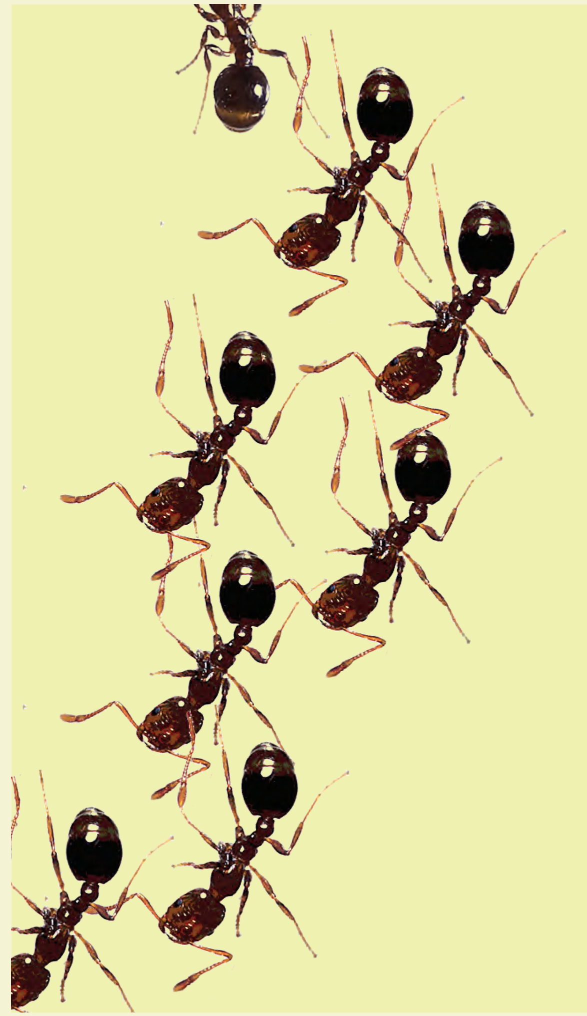
Example: Local approximation

Example: Decentralized control

Multi-agent model

M. Huang, P. E. Caines, and R. P. Malhame. Large-population cost-coupled LQG problems with nonuniform agents: Individual-mass behavior and decentralized ε-Nash equilibria. *IEEE Trans. Auto. Control*, 52(9):1560–1571, 2007.

Huang et. al. Local optimization for global coordination



Multi-agent model

Model: Linear autonomous models - global cost objective

HJB: Individual state + global average

Basis: Consistent with low dimensional LQG model

Results from five agent model:

Multi-agent model

Model: Linear autonomous models - global cost objective

HJB: Individual state + global average

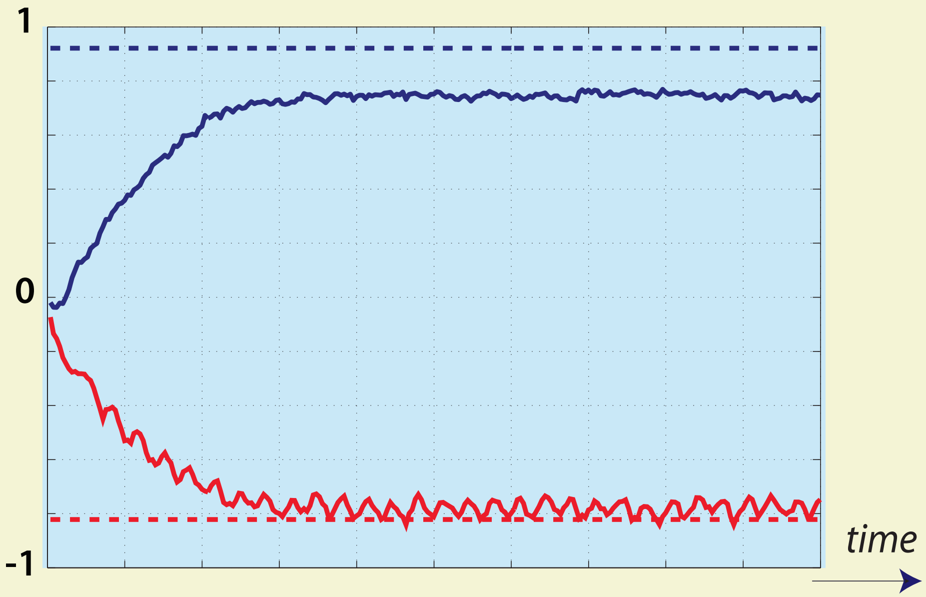
Basis: Consistent with low dimensional LQG model

Results from five agent model:

Estimated state feedback gains

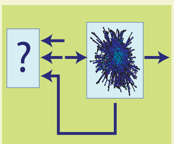
k_x^i *(individual state)*

k_z^i *(ensemble state)*

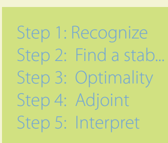


Gains for agent 4: Q-learning sample paths and gains predicted from ∞-agent limit

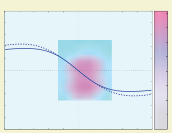
Outline



Coarse models - what to do with them?



Q-learning for nonlinear state space models



Example: Local approximation

Example: Decentralized control

...Conclusions

Conclusions

Coarse models give tremendous insight

They are also tremendously useful
for design in approximate dynamic programming algorithms

Conclusions

Coarse models give tremendous insight

They are also tremendously useful
for design in approximate dynamic programming algorithms

Q-learning is as fundamental as the Riccati equation - this
should be included in our first-year graduate control courses

Conclusions

Coarse models give tremendous insight

They are also tremendously useful
for design in approximate dynamic programming algorithms

Q-learning is as fundamental as the Riccati equation - this
should be included in our first-year graduate control courses

Current research: Algorithm analysis and improvements
Applications in biology and economics
Analysis of game-theoretic issues
in coupled systems

References

[1] D. H. Jacobson. Differential dynamic programming methods for determining optimal control of non-linear systems. PhD thesis, Univ. of London, 1967

[2] D. H. Jacobson and D. Q. Mayne. Differential dynamic programming. American Elsevier Pub. Co., New York, NY, 1970.

[3] C. J. C. H. Watkins. Learning from Delayed Rewards. PhD thesis, King's College, Cambridge, UK, 1989.

[4] C. J. C. H. Watkins and P. Dayan. Q-learning. Machine Learning, 8(3-4):279–292, 1992.

[5] J. N. Tsitsiklis and B. Van Roy. Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. IEEE Trans. Automat. Control, 44(10):1840–1851, 1999.

[6] V. S. Borkar and S. P. Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. SIAM J. Control Optim., 38(2):447–469, 2000.

[7] H. Yu and D. P. Bertsekas. Q-learning algorithms for optimal stopping based on least squares. In Proc. European Control Conference (ECC), July 2007.

[8] C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queueing networks. Preprint available at http://moallemi.com/ciamac/research-interests.php, 2008.

[9] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf. Brief paper: Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. Automatica, 43(3):473–481, 2007.

[10] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis. Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica, 45(2):477 – 484, 2009.

[11] S. P. Meyn. Control Techniques for Complex Networks. Cambridge University Press, Cambridge, 2007.

[12] S. G. Henderson, S. P. Meyn, and V. B. Tadi?. Performance evaluation and policy selection in multiclass networks. Discrete Event Dynamic Systems: Theory and Applications, 13(1-2):149–189, 2003. Special issue on learning, optimization and decision making (invited).

[13] W. Chen, D. Huang, A. Kulkarni, J. Unnikrishnan, Q. Zhu, P. Mehta, S. Meyn, and A. Wierman. Approximate dynamic programming using fluid and diffusion approximations with applications to power management. 48th IEEE Conference on Decision and Control, December 16-18 2009.