Part 4: Policy Gradient Reinforcement Learning

Sean Meyn

COGNITION & CONTROL IN COMPLEX SYSTEMS

Department of Electrical and Computer Engineering University of Florida

Inria International Chair Inria, Paris

Thanks to to our sponsors: NSF and ARO

Part 4: (Quasi) Policy Gradient Reinforcement Learning

Outline

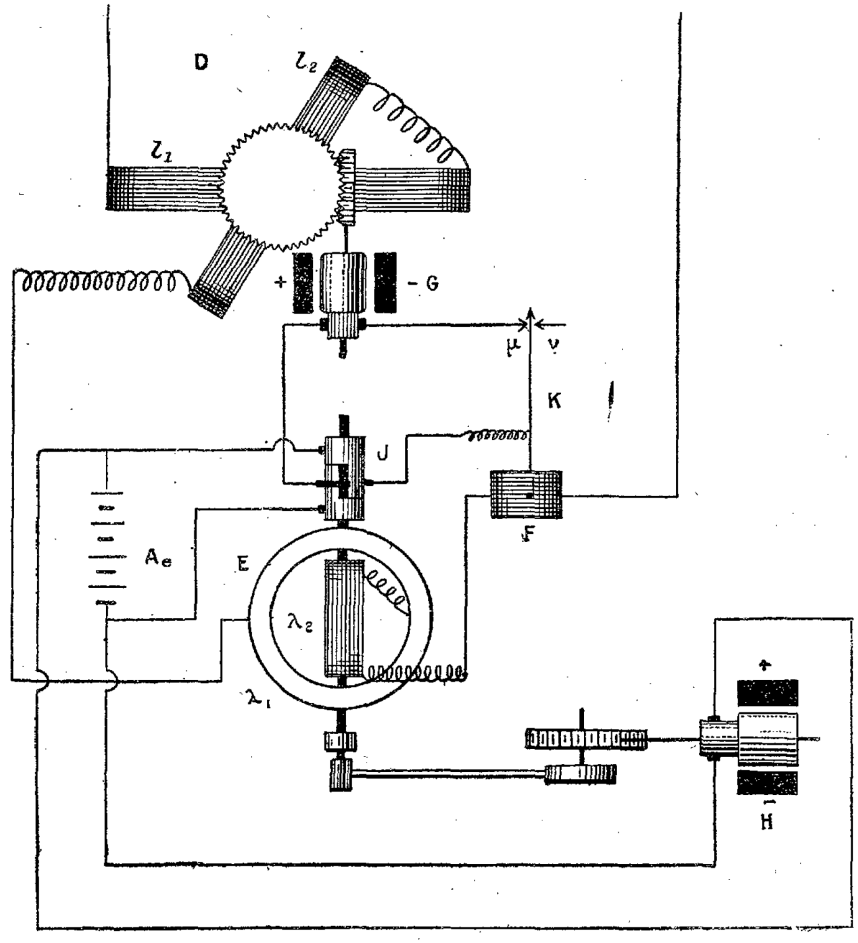


Fig. 5.

Policy Gradient Methods in Control

Extremum seeking from 1922 to 2010 [86]

In his 1922 paper, or invention disclosure, Leblanc [88] describes a mechanism to transfer power from an overhead electrical transmission line to a tram car using an ingenious non-contact solution. In order to maintain an efficient power transfer in what is essentially a linear, air-core, transformer/ capacitor arrangement with variable inductance, due to the changing air-gap, he identifies the need to adjust a (tram based) inductance (the input) so as to maintain a resonant circuit, or maximum power (the output). Leblanc explains a control mechanism of how to maintain the desirable maximum power transfer using what is essentially an extremum seeking solution.

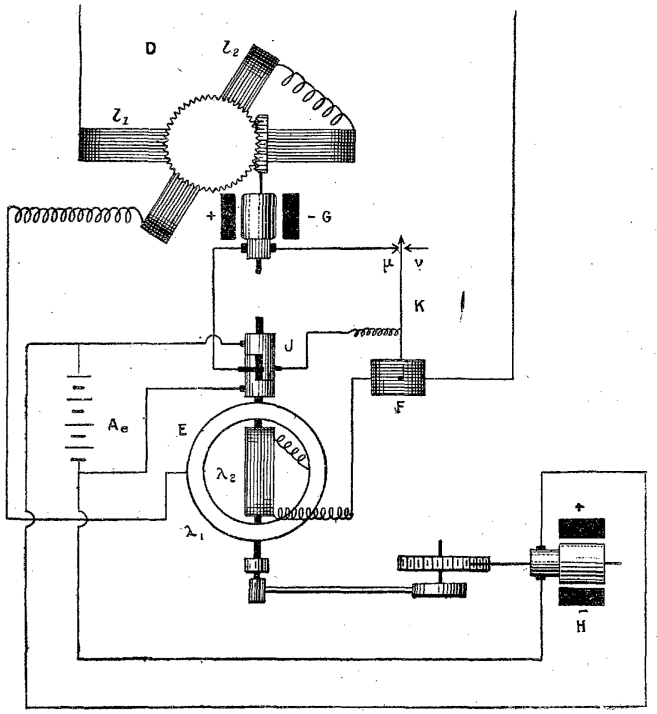


Fig. 5.

Extremum seeking from 1922 to 2010 [86]

In his 1922 paper, or invention disclosure, Leblanc [88] describes a mechanism to transfer power from an overhead electrical transmission line to a tram car using an ingenious non-contact solution. In order to maintain an efficient power transfer in what is essentially a linear, air-core, transformer/ capacitor arrangement with variable inductance, due to the changing air-gap, he identifies the need to adjust a (tram based) inductance (the input) so as to maintain a resonant circuit, or maximum power (the output). Leblanc explains a control mechanism of how to maintain the desirable maximum power transfer using what is essentially an extremum seeking solution.

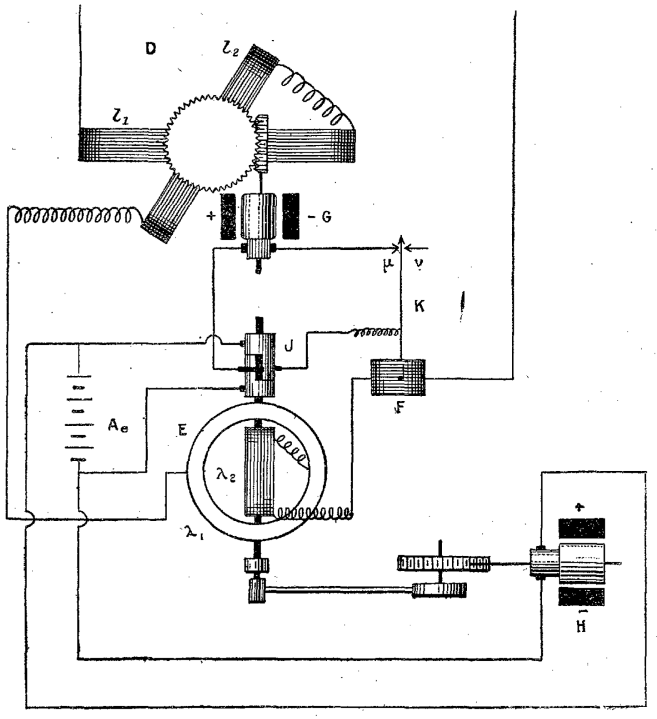


Fig. 5.

Iven, are you sure? Response on Sunday: [that is at the heart of] *what ESC is, non model based gradient descent...and that is what this circuit does, non model based, gradient descent by using forces that are proportional to finding the maximal energy transfer...*

Extremum seeking from 1922 to 2010 [86]

In his 1922 paper, or invention disclosure, Leblanc [88] describes a mechanism to transfer power from an overhead electrical transmission line to a tram car using an ingenious non-contact solution. In order to maintain an efficient power transfer in what is essentially a linear, air-core, transformer/ capacitor arrangement with variable inductance, due to the changing air-gap, he identifies the need to adjust a (tram based) inductance (the input) so as to maintain a resonant circuit, or maximum power (the output). Leblanc explains a control mechanism of how to maintain the desirable maximum power transfer using what is essentially an extremum seeking solution.

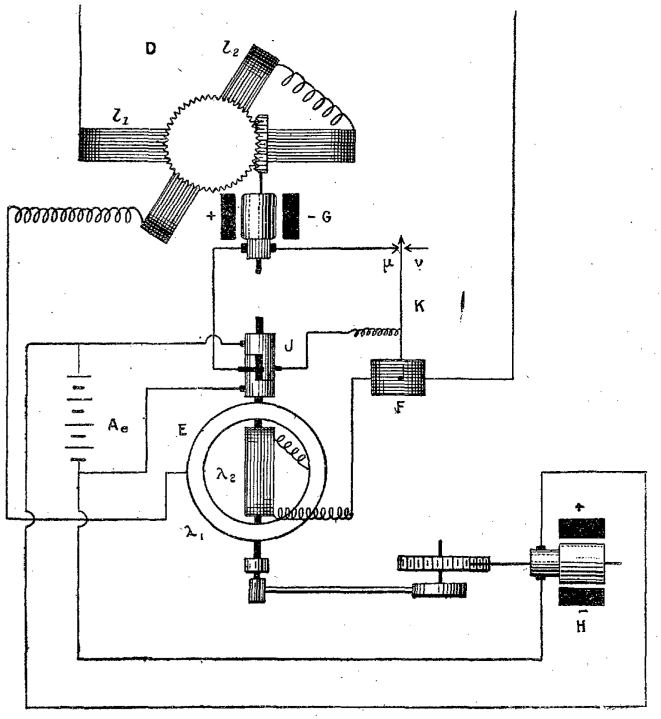


Fig. 5.

More history in 1962 and the 1980s: [87, 89, 88], and enormous activity from then until today.

Thanks to Florence Forbes and Iven Mareels for French paper discovery, translations, and history lessons

Self Tuning

Theory and applications of adaptive control—a survey Åström, 1983 [5]

The following parameter adjustment mechanism, called the 'MIT-rule', was used in the original MRAS

$$\frac{d\theta}{dt} = -k e \operatorname{grad}_\theta e. \quad (1)$$

In this equation e denotes the model error. The components of the vector θ are the adjustable regulator parameters. The

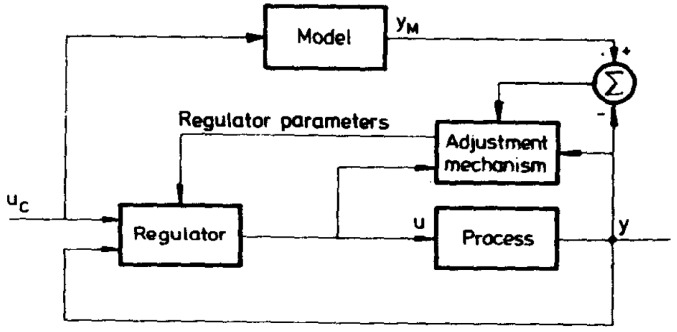


FIG. 2. Block diagram of model reference adaptive system (MRAS).

Karl Åström has been an inspiration from the start—for early history, see [5, 6, 3].

Self Tuning

Theory and applications of adaptive control—a survey Åström, 1983 [5]

The following parameter adjustment mechanism, called the 'MIT-rule', was used in the original MRAS

$$\frac{d\theta}{dt} = -k e \operatorname{grad}_\theta e. \quad (1)$$

In this equation e denotes the model error. The components of the vector θ are the adjustable regulator parameters. The

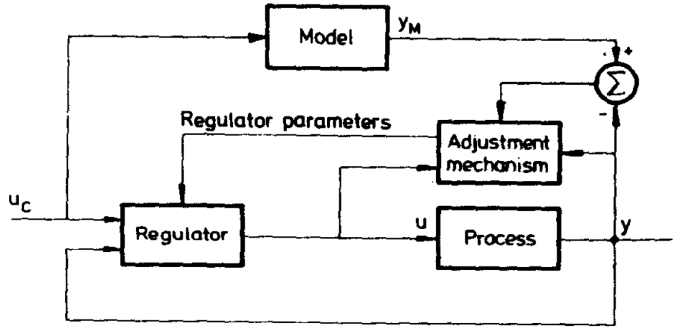


FIG. 2. Block diagram of model reference adaptive system (MRAS).

Karl Åström has been an inspiration from the start—for early history, see [5, 6, 3].

Adaptive Control Up To 1960 Åström, 1996 [6]

See also

Literature geared towards Lyapunov techniques:

- Liberzon's lecture notes: `http://liberzon.csl.illinois.edu/teaching/16ece517notes.pdf`
- Kokotovic et al [4]

Conclusions

Adaptive control was in a very interesting development in the mid-1960s. Many ideas such as extremal control, MRAS, STR, dual control, and neural networks, were born. It would take about two decades before the problems associated with adaptive control were reasonably well understood and adaptive techniques were finding use in industry. There are many reasons for the delay. The problems to be solved were difficult, funding for flight control dropped sharply because of accidents in flight tests, and new hardware was required for efficient implementation. A

The vast majority of Reinforcement Learning (RL) [9] and Neuro-Dynamic Programming (NDP) [1] methods fall into one of the following two categories:

(a) Actor-only methods work with a parameterized family of policies. The gradient of the performance, with respect to the actor parameters, is directly estimated by simulation, and the parameters are updated in a direction of improvement [4, 5, 8, 13]. A possible drawback of such methods is that the gradient estimators may have a large variance. Furthermore, as the policy changes, a new gradient is estimated independently of past estimates. Hence, there is no "learning," in the sense of accumulation and consolidation of older information.

(b) Critic-only methods rely exclusively on value function approximation and aim at learning an approximate solution to the Bellman equation, which will then hopefully prescribe a near-optimal policy. Such methods are indirect in the sense that they do not try to optimize directly over a policy space. A method of this type may succeed in constructing a "good" approximation of the value function, yet lack reliable guarantees in terms of near-optimality of the resulting policy.

Actor-critic methods aim at combining the strong points of actor-only and critic-only methods. The critic uses an approximation architecture and simulation to learn a value function, which is then used to update the actor's policy parameters

circa 2000

$$\nabla L(\theta) = \mathsf{E}\left[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\right]$$

Policy Gradient Methods in RL

1968 Origins [32, 33, 34] $L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$

Markov model: $\{P_\theta, c_\theta : \theta \in \mathbb{R}^d\}$ Goal: minimize $L(\theta)$, the average cost

Early Actor Only category: Williams 1992, REINFORCE [35]
Recent revival of AO, following Mania et al, 2018 [43]

1968 Origins [32, 33, 34] $L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$

Markov model: $\{P_\theta, c_\theta : \theta \in \mathbb{R}^d\}$ Goal: minimize $L(\theta)$, the average cost

Approach: gradient descent $\oplus$ our beloved Poisson equation [56, 57, 27]

$$c_\theta(x) + \sum_{x'} P_\theta(x, x')h_\theta(x') = h_\theta(x) + L(\theta)$$

1968 Origins [32, 33, 34] $L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$

Schweitzer: $\nabla L(\theta) = \mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\big]$

Markov model: $\{P_\theta,\, c_\theta : \theta \in \mathbb{R}^d\}$ Goal: minimize $L(\theta)$, the average cost

Approach: gradient descent $\oplus$ our beloved Poisson equation [56, 57, 27]

$$c_\theta(x) + \sum_{x'} P_\theta(x, x') h_\theta(x') = h_\theta(x) + L(\theta)$$

Denote $S_\theta(x, x') = \nabla \log(P_\theta(x, x'))$

$$\mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1}) + \nabla h_\theta\,(X_{k+1})\big] = \mathsf{E}\big[\nabla h_\theta\,(X_k)\big] + \nabla L(\theta)$$

1968 Origins [32, 33, 34] $L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$

Schweitzer: $\nabla L(\theta) = \mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\big]$

Markov model: $\{P_\theta,\, c_\theta : \theta \in \mathbb{R}^d\}$ Goal: minimize $L(\theta)$, the average cost

Approach: gradient descent $\oplus$ our beloved Poisson equation [56, 57, 27]

$$c_\theta(x) + \sum_{x'} P_\theta(x, x') h_\theta(x') = h_\theta(x) + L(\theta)$$

Details: Take the gradient of each side:

$$\nabla c_\theta(x) + \sum_{x'} \Big\{ [\nabla P_\theta(x, x')] h_\theta(x') + P_\theta(x, x') \nabla h_\theta\,(x') \Big\} = \nabla h_\theta(x) + \nabla L(\theta)$$

$$\nabla c_\theta(x) + \sum_{x'} P_\theta(x, x') \Big\{ S_\theta(x, x') h_\theta(x') + \nabla h_\theta\,(x') \Big\} = \nabla h_\theta(x') + \nabla L(\theta)$$

Denote $S_\theta(x, x') = \nabla \log(P_\theta(x, x'))$

$$\mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1}) + \nabla h_\theta\,(X_{k+1})\big] = \mathsf{E}\big[\nabla h_\theta\,(X_k)\big] + \nabla L(\theta)$$

Actor Critic Methods

$$L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$$

$$\nabla L(\theta) = \mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\big]$$

Follow design principle:

$$\text{ODE}: \quad \frac{d}{dt}\vartheta_t = -\nabla L(\vartheta_t)$$

$$\text{SA}: \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\big[\nabla c_\theta(X_n) + S_\theta(X_n, X_{n+1}) h_\theta(X_{n+1})\big]$$

Actor Critic Methods

$$L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$$

$$\nabla L(\theta) = \mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\big]$$

Follow design principle:

$$\text{ODE}: \quad \frac{d}{dt}\vartheta_t = -\nabla L(\vartheta_t)$$

$$\text{SA}: \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\big[\nabla c_\theta(X_n) + S_\theta(X_n, X_{n+1}) h_\theta(X_{n+1})\big]$$

For MDP, with randomized policy $\widetilde{\phi}^\theta$, S_θ is identified ...

$$\mathsf{E}\big[\nabla c_\theta(X_n) + S_\theta(X_n, X_{n+1}) h_\theta(X_{n+1})\big] = \mathsf{E}\big[Q_\theta(X_n, U_n) \nabla \log \widetilde{\phi}^\theta(U_n \mid X_n)\big]$$

Score function

https://en.wikipedia.org/wiki/Score_(statistics)

Actor Critic Methods

$$L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$$

$$\nabla L(\theta) = \mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\big]$$

Follow design principle:

$$\text{ODE}: \quad \frac{d}{dt}\vartheta_t = -\nabla L(\vartheta_t)$$

$$\text{SA}: \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\big[\nabla c_\theta(X_n) + S_\theta(X_n, X_{n+1}) h_\theta(X_{n+1})\big]$$

$$\text{SA}: \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\big[Q_{\theta_n}(X_n, U_n) \nabla \log \widetilde{\phi}^{\theta}(U_n \mid X_n)\big]$$

For MDP, with randomized policy $\widetilde{\phi}^{\theta}$, S_θ is identified ...

$$\mathsf{E}\big[\nabla c_\theta(X_n) + S_\theta(X_n, X_{n+1}) h_\theta(X_{n+1})\big] = \mathsf{E}\big[Q_\theta(X_n, U_n) \nabla \log \widetilde{\phi}^{\theta}(U_n \mid X_n)\big]$$

Score function

https://en.wikipedia.org/wiki/Score_(statistics)

... *however* ... Q?

Actor Critic Methods

$$L(\theta) = \lim_{k\to\infty} \mathsf{E}[c_\theta(X_k)]$$

$$\nabla L(\theta) = \mathsf{E}\big[\nabla c_\theta(X_k) + S_\theta(X_k, X_{k+1}) h_\theta(X_{k+1})\big]$$

Follow design principle:

$$\text{ODE}: \quad \frac{d}{dt}\vartheta_t = -\nabla L(\vartheta_t)$$

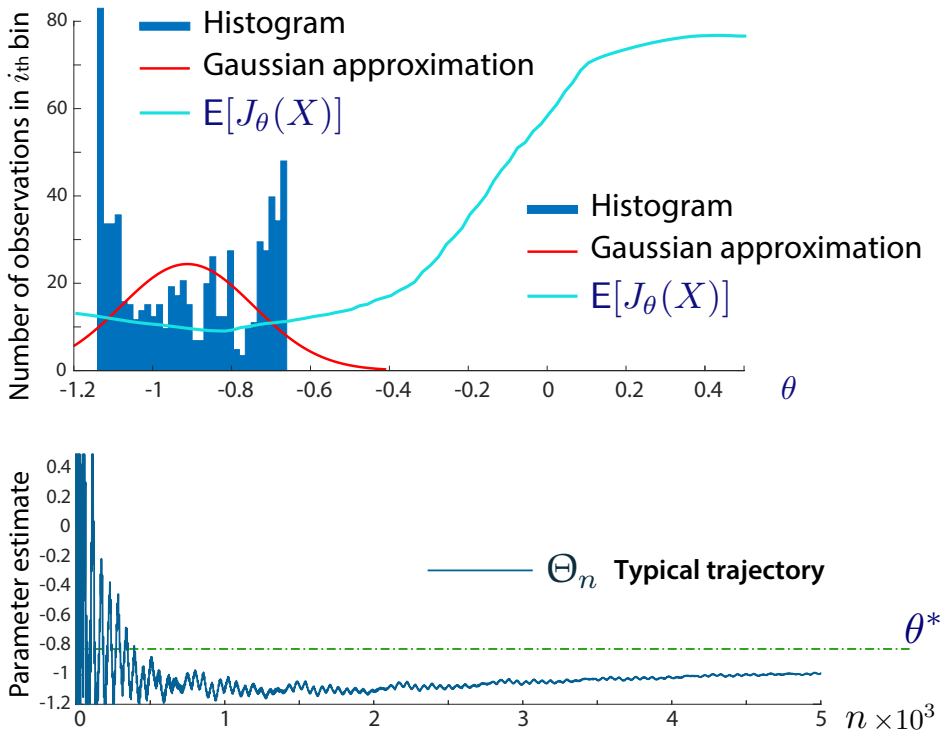
$$\text{SA}: \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\big[\nabla c_\theta(X_n) + S_\theta(X_n, X_{n+1}) h_\theta(X_{n+1})\big]$$

$$\text{SA}: \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\big[\{\widehat{Q}^{\theta_n}(X_n, U_n) - *\}\nabla \log \widetilde{\phi}^{\theta}(U_n \mid X_n)\big]$$

We need a critic! Key: only need approximation $\widehat{Q}^\theta$ to satisfy

$$\mathsf{E}\big[Q_\theta(X_n, U_n)\nabla \log \widetilde{\phi}^{\theta}(U_n \mid X_n)\big] = \mathsf{E}\big[\widehat{Q}^{\theta}(X_n, U_n)\nabla \log \widetilde{\phi}^{\theta}(U_n \mid X_n)\big]$$

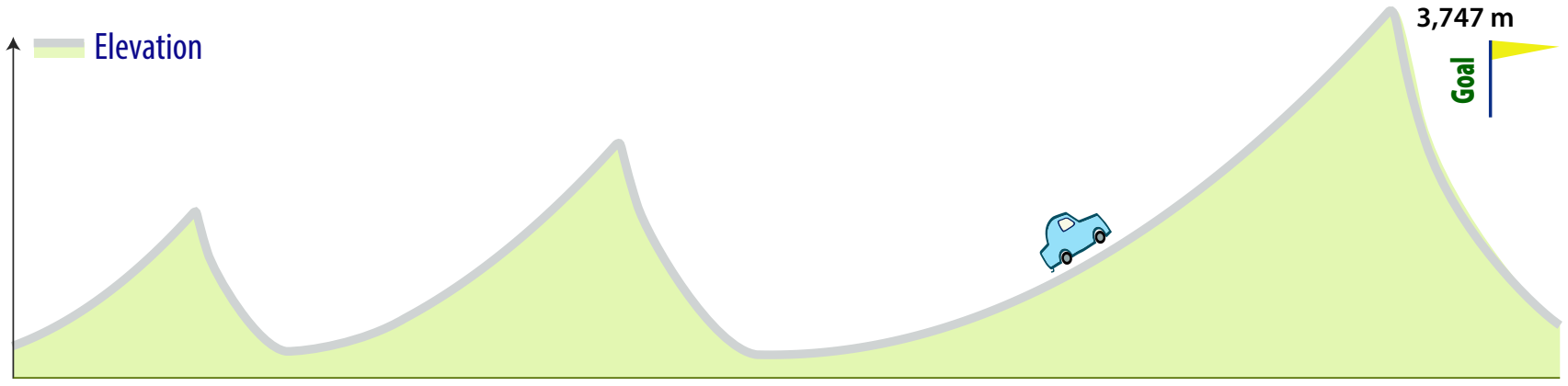
$\Longrightarrow$ Compatible features



qSGD and Policy Gradient RL

MountainCar

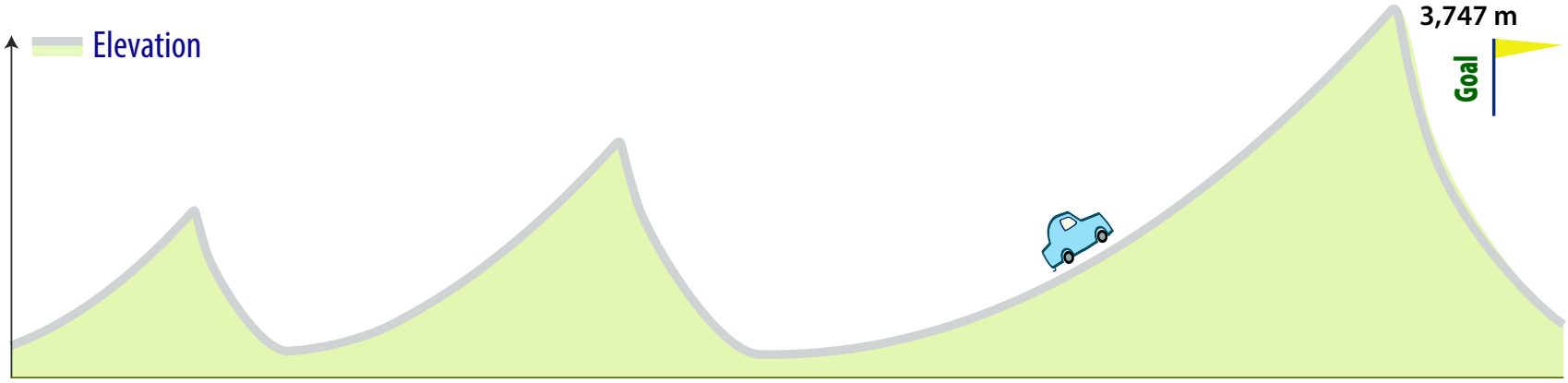
Rich's car has a very weak engine



The only way to reach the goal is to go backwards

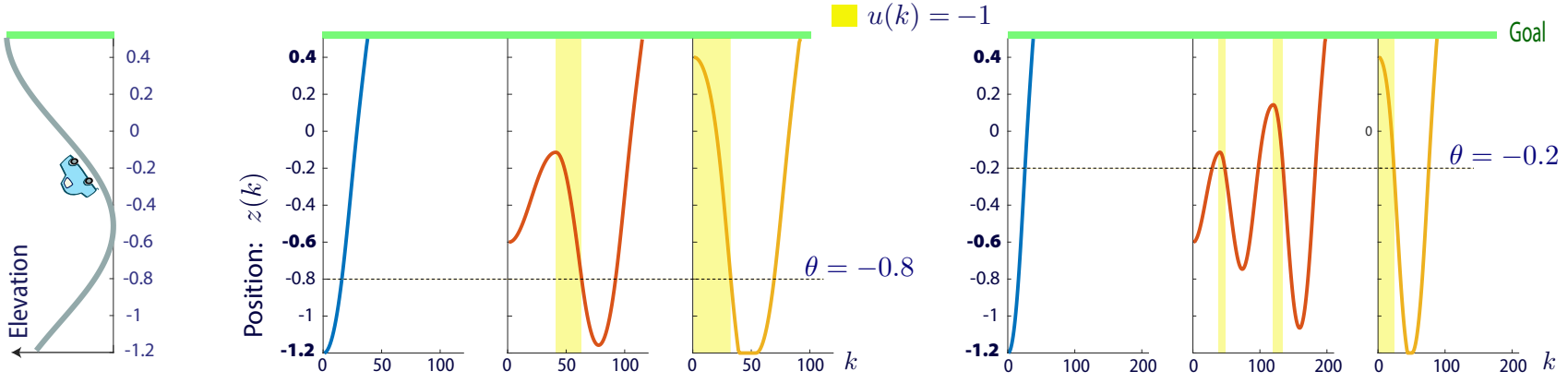
MountainCar

Rich's car has a very weak engine



The only way to reach the goal is to go backwards

Example policies (focusing on a single valley):



MountainCar

Goal: minimize travel time to destination $J(x)$

MountainCar

Goal: minimize travel time to destination $J(x)$

Approach: create a parameterized family of policies: $U_k = \phi^\theta(X_k)$

Minimize average loss: $$L(\theta) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \min\{J^{\max}, J_\theta(X_0^k)\}$$

$\{X_0^k : k \geq 1\}$ are initial conditions, most likely created by choice

$J_\theta(x)$ is the cost (perhaps average cost) from initial condition x

MountainCar

Goal: minimize travel time to destination $J(x)$
Approach: create a parameterized family of policies: $U_k = \phi^\theta(X_k)$

Minimize average loss: $$L(\theta) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \min\{J^{\max}, J_\theta(X_0^k)\}$$

$\{X_0^k : k \geq 1\}$ are initial conditions, most likely created by choice

$J_\theta(x)$ is the cost (perhaps average cost) from initial condition x

Let's try QSA:

$$\frac{d}{dt}\bar{\Theta}_t = a_t\{-\nabla L(\Theta_t)\} \quad \Longleftarrow \textit{Design for your goals}$$

$$\frac{d}{dt}\Theta_t = a_t f(\Theta_t, \xi_t) \quad \Longleftarrow \textit{qSGD approximation}$$

$$\theta_{n+1} = \theta_n + a_{n+1} f(\theta_n, \xi_{n+1}) \quad \Longleftarrow \textit{Euler/Runge-Kutta}$$

MountainCar

State space continuous, $x = (z, v)$ position and velocity, input zero or one

Model: $$Z_{k+1} = Z_k + V_k + D_k^z$$
$$V_{k+1} = V_k + \gamma U_k + D_k^v$$

D means disturbance

MountainCar

State space continuous, $x = (z, v)$ position and velocity, input zero or one

Model: $$Z_{k+1} = Z_k + V_k + D_k^z$$
$$V_{k+1} = V_k + \gamma U_k + D_k^v \qquad D \text{ means disturbance}$$

Policy: $$U_k = \begin{cases} 1 & \text{if } Z_k + V_k \leq \theta \\ \text{sign}(V_k) & \text{else} \end{cases}$$

The policy ϕ^θ "panics" (accelerates the car towards the goal) whenever the estimate of Z_{k+1} is at or below the threshold θ

MountainCar

State space continuous, $x = (z, v)$ position and velocity, input zero or one

Model: $$Z_{k+1} = Z_k + V_k + D_k^z$$
$$V_{k+1} = V_k + \gamma U_k + D_k^v \qquad D \text{ means disturbance}$$

Policy: $$U_k = \begin{cases} 1 & \text{if } Z_k + V_k \le \theta \\ \text{sign}(V_k) & \text{else} \end{cases}$$

The policy ϕ^θ "panics" (accelerates the car towards the goal) whenever the estimate of Z_{k+1} is at or below the threshold θ

Goal: Find best policy in this class

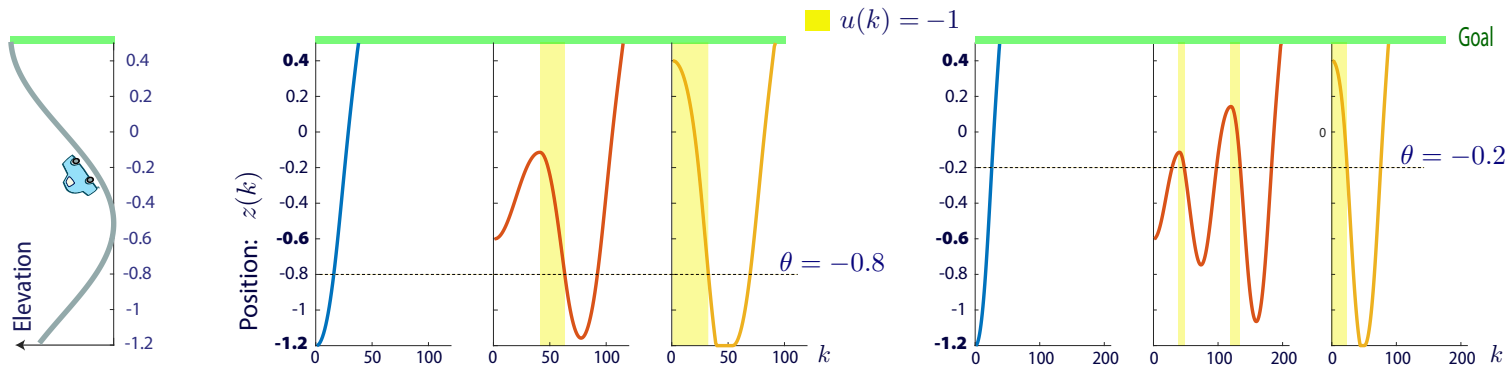
MountainCar

State space continuous, $x = (z, v)$ position and velocity, input zero or one

Model: $$Z_{k+1} = Z_k + V_k + D_k^z$$
$$V_{k+1} = V_k + \gamma U_k + D_k^v \qquad D \text{ means disturbance}$$

Policy: $$U_k = \begin{cases} 1 & \text{if } Z_k + V_k \le \theta \\ \text{sign}(V_k) & \text{else} \end{cases}$$

The policy ϕ^θ "panics" (accelerates the car towards the goal) whenever the estimate of Z_{k+1} is at or below the threshold θ



MountainCar

qSGD: $\frac{d}{dt}\Theta_t = -a_t \frac{1}{2\varepsilon} G \xi_t \big\{ L(\Theta_t + \varepsilon\xi_t) - L(\Theta_t - \varepsilon\xi_t) \big\}$

In discrete time:

$$\text{(1)} \qquad \theta_{n+1} = \theta_n + \alpha_{n+1} \big[-\tfrac{1}{2\varepsilon} \xi_n \big\{ L(\theta_n + \varepsilon\xi_n) - L(\theta_n - \varepsilon\xi_n) \big\} \big]$$

$$\text{(1a)} \qquad \theta_{n+1} = \theta_n - \alpha_{n+1} \tfrac{1}{\varepsilon} \xi_n L(\theta_n + \varepsilon\xi_n)$$

recall danger with this one ☠

MountainCar

qSGD: $\frac{d}{dt}\Theta_t = -a_t \frac{1}{2\varepsilon} G\xi_t \big\{ L(\Theta_t + \varepsilon\xi_t) - L(\Theta_t - \varepsilon\xi_t) \big\}$

In discrete time:

(1) $\theta_{n+1} = \theta_n + \alpha_{n+1} \big[-\frac{1}{2\varepsilon} \xi_n \big\{ L(\theta_n + \varepsilon\xi_n) - L(\theta_n - \varepsilon\xi_n) \big\} \big]$

(1a) $\theta_{n+1} = \theta_n - \alpha_{n+1} \frac{1}{\varepsilon} \xi_n L(\theta_n + \varepsilon\xi_n)$ *recall danger with this one* ☠

Slight extension:

$$L(\theta_n + \varepsilon\xi_n) = J_\theta(x)\,, \quad \theta = \theta_n + \varepsilon\xi_n^1\,, \quad x = x^0 + \varepsilon\xi_n^2$$

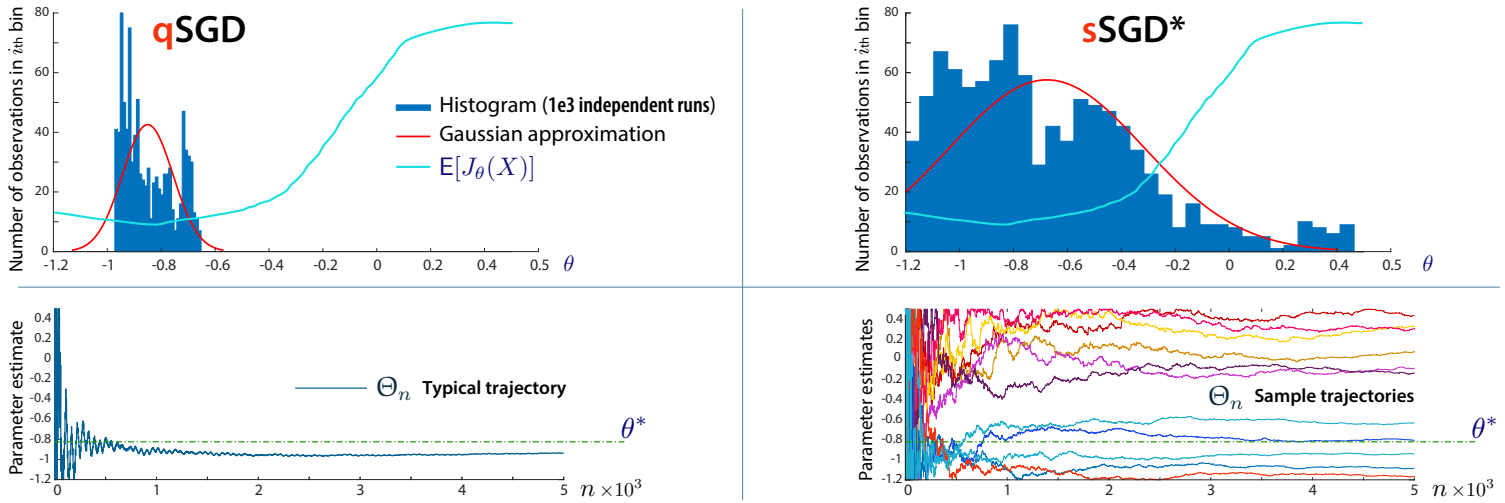
MountainCar

qSGD: $\frac{d}{dt}\Theta_t = -a_t \frac{1}{2\varepsilon} G \xi_t \big\{ L(\Theta_t + \varepsilon\xi_t) - L(\Theta_t - \varepsilon\xi_t) \big\}$

In discrete time:

(1) $$\theta_{n+1} = \theta_n + \alpha_{n+1}\Big[-\frac{1}{2\varepsilon}\xi_n \big\{ L(\theta_n + \varepsilon\xi_n) - L(\theta_n - \varepsilon\xi_n) \big\}\Big]$$

(1a) $$\theta_{n+1} = \theta_n - \alpha_{n+1}\frac{1}{\varepsilon}\xi_n L(\theta_n + \varepsilon\xi_n)$$ *recall danger with this one* ☠



qSGD (1a) and stochastic counterpart for Mountain Car ☠

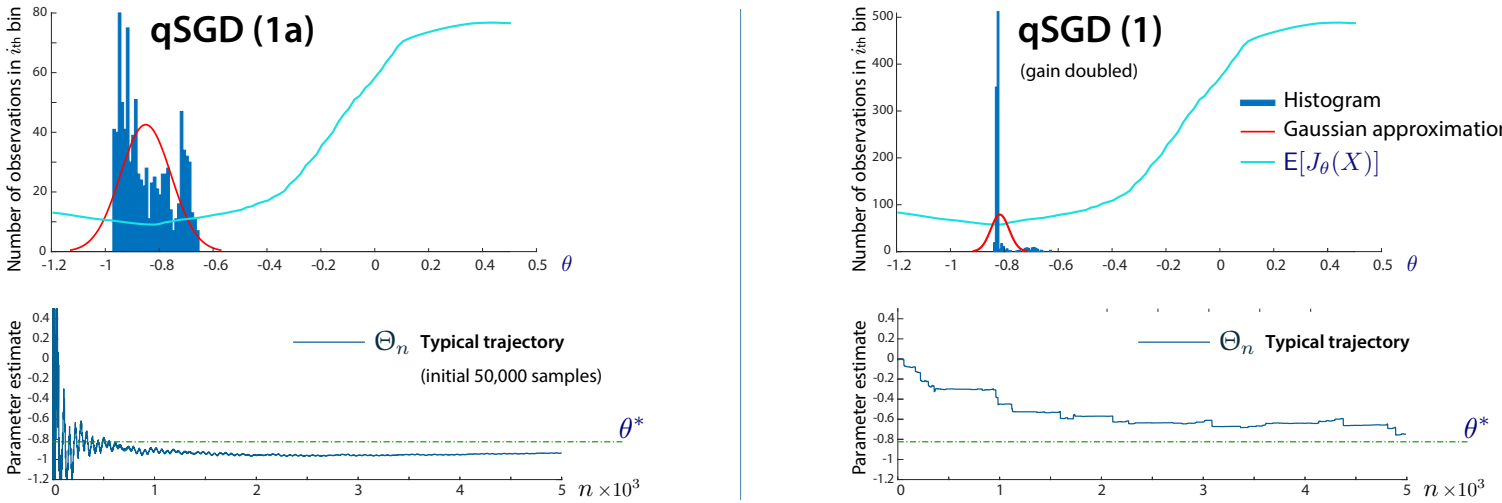
MountainCar

qSGD: $\frac{d}{dt}\Theta_t = -a_t \frac{1}{2\varepsilon} G\xi_t \big\{ L(\Theta_t + \varepsilon\xi_t) - L(\Theta_t - \varepsilon\xi_t) \big\}$

In discrete time:

$$\text{(1)} \quad \theta_{n+1} = \theta_n + \alpha_{n+1}\Big[-\frac{1}{2\varepsilon}\xi_n \big\{ L(\theta_n + \varepsilon\xi_n) - L(\theta_n - \varepsilon\xi_n) \big\}\Big]$$

$$\text{(1a)} \quad \theta_{n+1} = \theta_n - \alpha_{n+1}\frac{1}{\varepsilon}\xi_n L(\theta_n + \varepsilon\xi_n)$$ *recall danger with this one* ☠



Comparison of two qSGD algorithms

Conclusions

Probability Theory, Control, Reinforcement Learning: A Happy Marriage

I believe this, *even after all the effort to replace* W_n *with* ξ_t*!*

Conclusions

Probability Theory, Control, Reinforcement Learning: A Happy Marriage

I believe this, *even after all the effort to replace* W_n *with* ξ_t*!*

Theory for QSA and qSGD adapts/steals theory for SA:
Easier in part because of zero covariance, $\Sigma_\xi = 0$

Conclusions

Probability Theory, Control, Reinforcement Learning: A Happy Marriage

I believe this, *even after all the effort to replace* W_n *with* ξ_t*!*

Theory for QSA and qSGD adapts/steals theory for SA:
Easier in part because of zero covariance, $\Sigma_\xi = 0$

There is of course lots of exploration to do:

- Testing qSGD in more volatile environments
- Tackling dimension, $\theta \in \mathbb{R}^{10^6}$
- Other forms of acceleration (revisit classical and recent [54])

Conclusions

Probability Theory, Control, Reinforcement Learning: A Happy Marriage

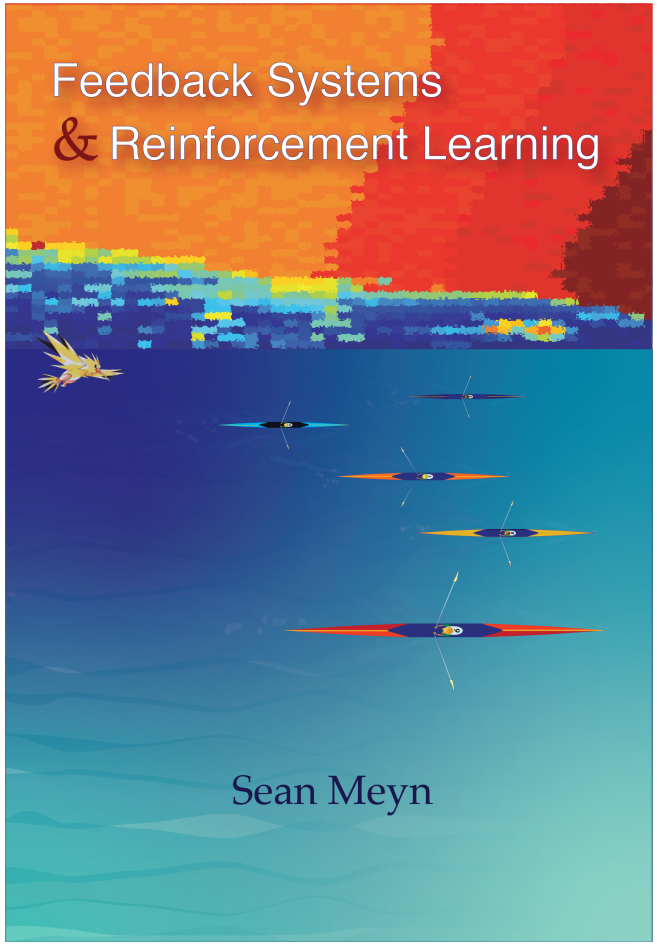
I believe this, *even after all the effort to replace* W_n *with* ξ_t*!*

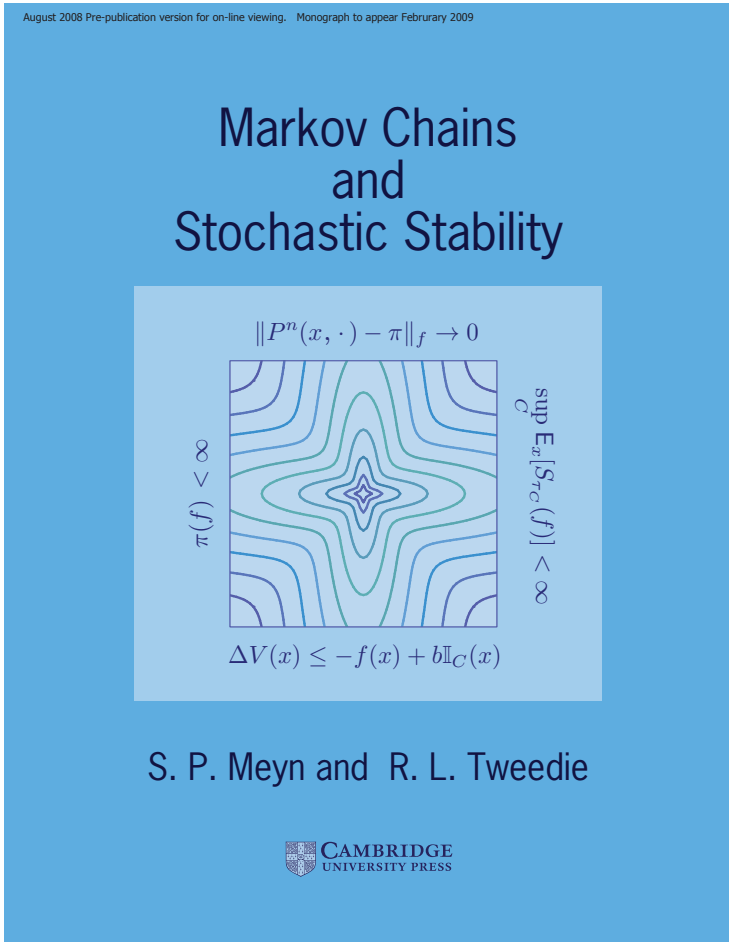
Theory for QSA and qSGD adapts/steals theory for SA:
Easier in part because of zero covariance, $\Sigma_\xi = 0$

There is of course lots of exploration to do:

- Testing qSGD in more volatile environments
- Tackling dimension, $\theta \in \mathbb{R}^{10^6}$
- Other forms of acceleration (revisit classical and recent [54])
- Optimizing the wind farms in Colorado

On to you, Andrey!


Feedback Systems
& Reinforcement Learning
Sean Meyn


August 2008 Pre-publication version for on-line viewing. Monograph to appear Februrary 2009
Markov Chains
and
Stochastic Stability
$\|P^n(x,\cdot)-\pi\|_f \to 0$
$\pi(f) < \infty$
$\sup_C \mathsf{E}_x[S_{\tau_C}(f)] < \infty$
$\Delta V(x) \le -f(x) + b\mathbb{I}_C(x)$
S. P. Meyn and R. L. Tweedie
CAMBRIDGE
UNIVERSITY PRESS

References

Control Background I

[1] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, USA, 2008 (recent edition on-line).

[2] K. J. Åström and K. Furuta. Swinging up a pendulum by energy control. *Automatica*, 36(2):287 – 295, 2000.

[3] K. J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

[4] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos. *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

[5] K. J. Åström. Theory and applications of adaptive control—a survey. *Automatica*, 19(5):471–486, 1983.

[6] K. J. Åström. Adaptive control around 1960. *IEEE Control Systems Magazine*, 16(3):44–49, 1996.

[7] B. Wittenmark. Stochastic adaptive control methods: a survey. *International Journal of Control*, 21(5):705–730, 1975.

[8] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, 1977.

Control Background II

[9] N. Matni, A. Proutiere, A. Rantzer, and S. Tu. From self-tuning regulators to reinforcement learning and back again. In *Proc. of the IEEE Conf. on Dec. and Control*, pages 3724–3740, 2019.

RL Background I

[10] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press. On-line edition at `http://www.cs.ualberta.ca/~sutton/book/the-book.html`, Cambridge, MA, 2nd edition, 2018.

[11] C. Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

[12] R. S. Sutton.*Learning to predict by the methods of temporal differences. Mach. Learn.*, 3(1):9–44, 1988.

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

[14] J. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16:185–202, 1994.

[15] T. Jaakola, M. Jordan, and S. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6:1185–1201, 1994.

[16] J. N. Tsitsiklis and B. Van Roy. *An analysis of temporal-difference learning with function approximation. IEEE Trans. Automat. Control*, 42(5):674–690, 1997.

[17] 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.

RL Background II

[18] D. Choi and B. Van Roy. *A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning. Discrete Event Dynamic Systems: Theory and Applications*, 16(2):207–239, 2006.

[19] S. J. Bradtke and A. G. Barto. *Linear least-squares algorithms for temporal difference learning. Mach. Learn.*, 22(1-3):33–57, 1996.

[20] J. A. Boyan. *Technical update: Least-squares temporal difference learning. Mach. Learn.*, 49(2-3):233–246, 2002.

[21] A. Nedic and D. Bertsekas. *Least squares policy evaluation algorithms with linear function approximation. Discrete Event Dyn. Systems: Theory and Appl.*, 13(1-2):79–110, 2003.

[22] C. Szepesvári. *The asymptotic convergence-rate of Q-learning*. In *Proceedings of the 10th Internat. Conf. on Neural Info. Proc. Systems*, 1064–1070. MIT Press, 1997.

[23] E. Even-Dar and Y. Mansour. *Learning rates for Q-learning. Journal of Machine Learning Research*, 5(Dec):1–25, 2003.

[24] M. G. Azar, R. Munos, M. Ghavamzadeh, and H. Kappen. *Speedy Q-learning.* In *Advances in Neural Information Processing Systems*, 2011.

RL Background III

[25] D. Huang, W. Chen, P. Mehta, S. Meyn, and A. Surana. *Feature selection for neuro-dynamic programming.* In F. Lewis, editor, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Wiley, 2011.

[26] A. M. Devraj, A. Bušić, and S. Meyn. Fundamental design principles for reinforcement learning algorithms. In *Handbook on Reinforcement Learning and Control*. Springer, 2020.

[27] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2007. *See last chapter on simulation and average-cost TD learning*

DQN:

[28] M. Riedmiller. *Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method.* In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, editors, *Machine Learning: ECML 2005*, pages 317–328, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[29] S. Lange, T. Gabel, and M. Riedmiller. *Batch reinforcement learning.* In *Reinforcement learning*, pages 45–73. Springer, 2012.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. *Playing Atari with deep reinforcement learning. ArXiv*, abs/1312.5602, 2013.

RL Background IV

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. *Human-level control through deep reinforcement learning. Nature*, 518:529–533, 2015.

Actor Critic / Policy Gradient

[32] P. J. Schweitzer. Perturbation theory and finite Markov chains. *J. Appl. Prob.*, 5:401–403, 1968.

[33] C. D. Meyer, Jr. The role of the group generalized inverse in the theory of finite Markov chains. *SIAM Review*, 17(3):443–464, 1975.

[34] P. W. Glynn. Stochastic approximation for Monte Carlo optimization. In *Proceedings of the 18th conference on Winter simulation*, pages 356–365, 1986.

[35] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[36] T. Jaakkola, S. P. Singh, and M. I. Jordan. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in neural information processing systems*, pages 345–352, 1995.

RL Background V

[37] X.-R. Cao and H.-F. Chen. Perturbation realization, potentials, and sensitivity analysis of Markov processes. *IEEE Transactions on Automatic Control*, 42(10):1382–1393, Oct 1997.

[38] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Trans. Automat. Control*, 46(2):191–209, 2001.

[39] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[40] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[41] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Trans. Automat. Control*, 46(2):191–209, 2001.

[42] S. M. Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002.

RL Background VI

[43] H. Mania, A. Guy, and B. Recht. Simple random search provides a competitive approach to reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1800–1809, 2018.

MDPs, LPs and Convex Q:

[44] A. S. Manne. Linear programming and sequential decisions. *Management Sci.*, 6(3):259–267, 1960.

[45] C. Derman. *Finite State Markovian Decision Processes*, volume 67 of *Mathematics in Science and Engineering*. Academic Press, Inc., 1970.

[46] V. S. Borkar. Convex analytic methods in Markov decision processes. In *Handbook of Markov decision processes*, volume 40 of *Internat. Ser. Oper. Res. Management Sci.*, pages 347–375. Kluwer Acad. Publ., Boston, MA, 2002.

[47] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Res.*, 51(6):850–865, 2003.

[48] D. P. de Farias and B. Van Roy. *A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. Math. Oper. Res.*, 31(3):597–620, 2006.

RL Background VII

[49] P. G. Mehta and S. P. Meyn. *Q-learning and Pontryagin's minimum principle.* In *Proc. of the IEEE Conf. on Dec. and Control*, pages 3598–3605, Dec. 2009.

[50] P. G. Mehta and S. P. Meyn. *Convex Q-learning, part 1: Deterministic optimal control. ArXiv e-prints:2008.03559*, 2020.

Gator Nation:

[51] A. M. Devraj and S. P. Meyn. *Fastest convergence for Q-learning. ArXiv* , July 2017 (extended version of NIPS 2017).

[52] A. M. Devraj. *Reinforcement Learning Design with Optimal Learning Rate*. PhD thesis, University of Florida, 2019.

[53] A. M. Devraj and S. P. Meyn. Q-learning with Uniformly Bounded Variance: Large Discounting is Not a Barrier to Fast Learning. *arXiv e-prints 2002.10301, and to appear AISTATS*, Feb. 2020.

[54] A. M. Devraj, A. Bušić, and S. Meyn. On matrix momentum stochastic approximation and applications to Q-learning. In *Allerton Conference on Communication, Control, and Computing*, pages 749–756, Sep 2019.

Stochastic Miscellanea I

[55] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*, volume 57 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, New York, 2007.

[56] P. W. Glynn and S. P. Meyn. *A Liapounov bound for solutions of the Poisson equation. Ann. Probab.*, 24(2):916–931, 1996.

[57] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, Cambridge, second edition, 2009. Published in the Cambridge Mathematical Library.

[58] R. Douc, E. Moulines, P. Priouret, and P. Soulier. *Markov Chains*. Springer, 2018.

Stochastic Approximation I

[59] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency and Cambridge University Press, Delhi, India & Cambridge, UK, 2008.

[60] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*, volume 22 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1990. Translated from the French by Stephen S. Wilson.

[61] 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.

[62] M. Benaïm. Dynamics of stochastic approximation algorithms. In *Séminaire de Probabilités, XXXIII*, pages 1–68. Springer, Berlin, 1999.

[63] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 23(3):462–466, 09 1952.

[64] D. Ruppert. *A Newton-Raphson version of the multivariate Robbins-Monro procedure. The Annals of Statistics*, 13(1):236–245, 1985.

[65] D. Ruppert. *Efficient estimators from a slowly convergent Robbins-Monro processes.* Technical Report Tech. Rept. No. 781, Cornell University, School of Operations Research and Industrial Engineering, Ithaca, NY, 1988.

Stochastic Approximation II

[66] B. T. Polyak. *A new method of stochastic approximation type. Avtomatika i telemekhanika, 98–107, 1990 (in Russian). Translated in Automat. Remote Control, 51 1991*.

[67] B. T. Polyak and A. B. Juditsky. *Acceleration of stochastic approximation by averaging. SIAM J. Control Optim.*, 30(4):838–855, 1992.

[68] V. R. Konda and J. N. Tsitsiklis. *Convergence rate of linear two-time-scale stochastic approximation. Ann. Appl. Probab.*, 14(2):796–819, 2004.

[69] E. Moulines and F. R. Bach. *Non-asymptotic analysis of stochastic approximation algorithms for machine learning.* In *Advances in Neural Information Processing Systems 24*, 451–459. Curran Associates, Inc., 2011.

[70] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn. Explicit Mean-Square Error Bounds for Monte-Carlo and Linear Stochastic Approximation. *arXiv e-prints*, 2002.02584, Feb. 2020.

[71] W. Mou, C. Junchi Li, M. J. Wainwright, P. L. Bartlett, and M. I. Jordan. On Linear Stochastic Approximation: Fine-grained Polyak-Ruppert and Non-Asymptotic Concentration. *arXiv e-prints*, page arXiv:2004.04719, Apr. 2020.

Optimization and ODEs I

[72] W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in neural information processing systems*, pages 2510–2518, 2014.

[73] B. Shi, S. S. Du, W. Su, and M. I. Jordan. Acceleration via symplectic discretization of high-resolution differential equations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5744–5752. Curran Associates, Inc., 2019.

[74] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[75] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, 1983.

QSA and Extremum Seeking Control I

[76] S. Chen, A. Bernstein, A. Devraj, and S. Meyn. Accelerating optimization and reinforcement learning with quasi-stochastic approximation. *arXiv:In preparation*, 2020.

[77] B. Lapeybe, G. Pages, and K. Sab. Sequences with low discrepancy generalisation and application to Robbins-Monro algorithm. *Statistics*, 21(2):251–272, 1990.

[78] S. Laruelle and G. Pagès. Stochastic approximation with averaging innovation applied to finance. *Monte Carlo Methods and Applications*, 18(1):1–51, 2012.

[79] S. Shirodkar and S. Meyn. Quasi stochastic approximation. In *Proc. of the 2011 American Control Conference (ACC)*, pages 2429–2435, July 2011.

[80] A. Bernstein, Y. Chen, M. Colombino, E. Dall'Anese, P. Mehta, and S. Meyn. Optimal rate of convergence for quasi-stochastic approximation. *arXiv:1903.07228*, 2019.

[81] A. Bernstein, Y. Chen, M. Colombino, E. Dall'Anese, P. Mehta, and S. Meyn. Quasi-stochastic approximation and off-policy reinforcement learning. In *Proc. of the IEEE Conf. on Dec. and Control*, pages 5244–5251, Mar 2019.

[82] Y. Chen, A. Bernstein, A. Devraj, and S. Meyn. Model-Free Primal-Dual Methods for Network Optimization with Application to Real-Time Optimal Power Flow. In *Proc. of the American Control Conf.*, pages 3140–3147, Sept. 2019.

QSA and Extremum Seeking Control II

[83] S. Bhatnagar and V. S. Borkar. Multiscale chaotic spsa and smoothed functional algorithms for simulation optimization. *Simulation*, 79(10):568–580, 2003.

[84] S. Bhatnagar, M. C. Fu, S. I. Marcus, and I.-J. Wang. Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):180–209, 2003.

[85] M. Le Blanc. Sur l'electrification des chemins de fer au moyen de courants alternatifs de frequence elevee [On the electrification of railways by means of alternating currents of high frequency]. *Revue Generale de l'Electricite*, 12(8):275–277, 1922.

[86] Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. M. Y. Mareels. Extremum seeking from 1922 to 2010. In *Proceedings of the 29th Chinese Control Conference*, pages 14–26, July 2010.

[87] P. F. Blackman. Extremum-seeking regulators. In *An Exposition of Adaptive Control*. Macmillan, 1962.

[88] J. Sternby. Adaptive control of extremum systems. In H. Unbehauen, editor, *Methods and Applications in Adaptive Control*, pages 151–160, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.

QSA and Extremum Seeking Control III

[89] J. Sternby. Extremum control systems–an area for adaptive control? In *Joint Automatic Control Conference*, number 17, page 8, 1980.

[90] K. B. Ariyur and M. Krstić. *Real Time Optimization by Extremum Seeking Control*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[91] M. Krstić and H.-H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4):595 – 601, 2000.

[92] S. Liu and M. Krstic. Introduction to extremum seeking. In *Stochastic Averaging and Stochastic Extremum Seeking*, Communications and Control Engineering. Springer, London, 2012.

[93] O. Trollberg and E. W. Jacobsen. On the convergence rate of extremum seeking control. In *European Control Conference (ECC)*, pages 2115–2120. 2014.

Selected Applications I

[94] N. S. Raman, A. M. Devraj, P. Barooah, and S. P. Meyn. *Reinforcement learning for control of building HVAC systems.* In *American Control Conference*, July 2020.

[95] K. Mason and S. Grijalva. A review of reinforcement learning for autonomous building energy management. *arXiv.org*, 2019. arXiv:1903.05196.

News from Andrey@NREL:

[96] A. Bernstein and E. Dall'Anese. Real-time feedback-based optimization of distribution grids: A unified approach. *IEEE Transactions on Control of Network Systems*, 6(3):1197–1209, 2019.

[97] A. Bernstein, E. Dall'Anese, and A. Simonetto. Online primal-dual methods with measurement feedback for time-varying convex optimization. *IEEE Transactions on Signal Processing*, 67(8):1978–1991, 2019.

[98] Y. Chen, A. Bernstein, A. Devraj, and S. Meyn. Model-free primal-dual methods for network optimization with application to real-time optimal power flow. In *2020 American Control Conference (ACC)*, pages 3140–3147, 2020.