Reinforcement Learning in Control Theory

Farnaz Adib Yaghmaie

LINKÖPING UNIVERSITY

1 Introduction to RL
2 RL for Continuous-time Systems
3 Simulation results
4 Open Problems

LINKÖPING UNIVERSITY

Introduction to RL

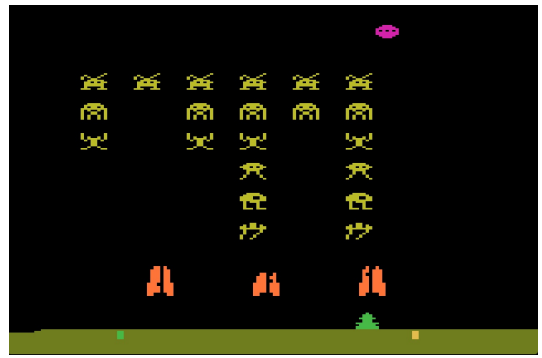
Machine Learning

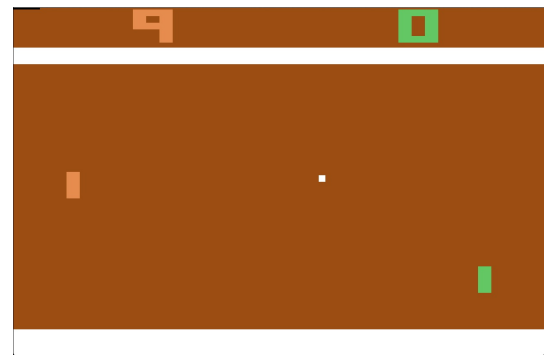
- Supervised Learning
- Unsupervised Learning
- **Reinforcement Learning**

Finding suitable actions to take in a given situation in order to maximize a reward [1]**.**

[1] Richard S Sutton & Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

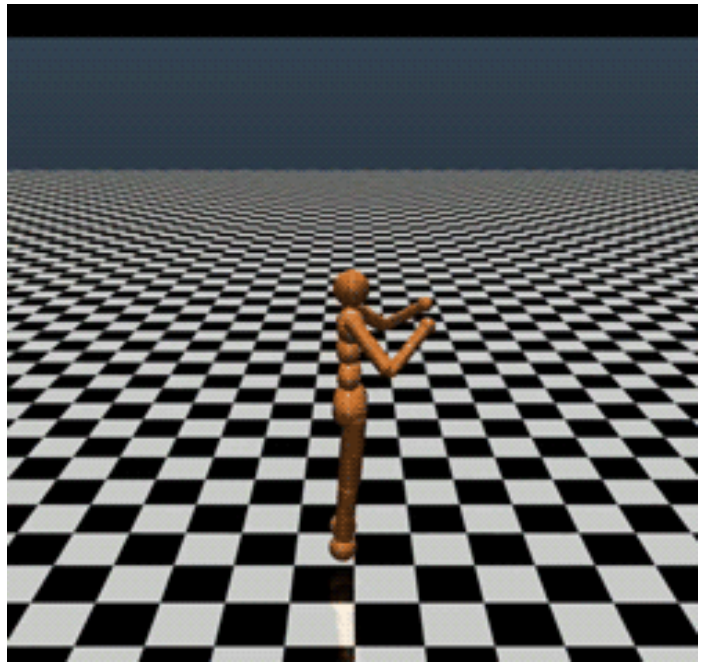
Beating Atari champions[2]





[2] Mnih et al. *Playing Atari with Deep Reinforcement Learning*, arXiv preprint, 2013.

Make a Robot Walk [3]



[3] Schulman et al. *Trust Region Policy Optimization*, arXiv preprint, 2017.

A graphical representation

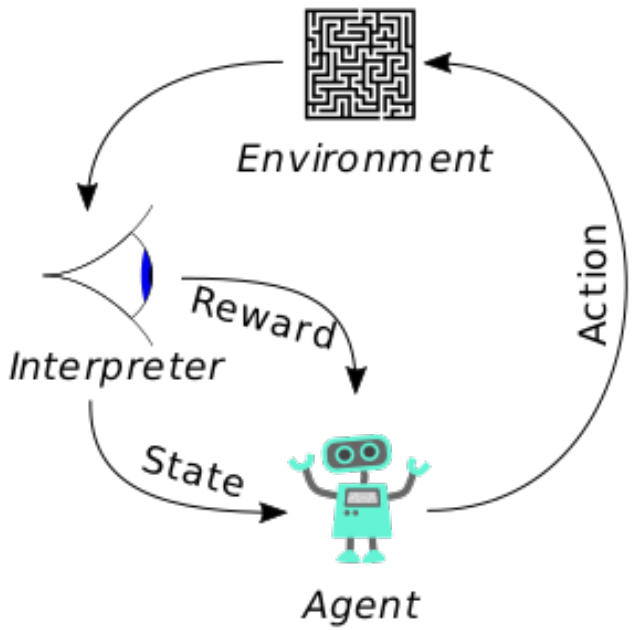


Photo Credit: @en.wikipedia.org/wiki/Reinforcement_learning

Optimality or adaptivity or both?

Optimal Design

- Offline
- Model-base

Adaptive Design

- Online
- Non-optimal
- Model-free

Reinforcement Learning: Optimality + Adaptivity [4].

[4] Lewis et al. *Reinforcement learning and feedback control*, IEEE Control Systems Magazine, 2012.

Markov Decision Process (MDP)

- States
- Actions
- Transition probability
- Reward
- Discount factor

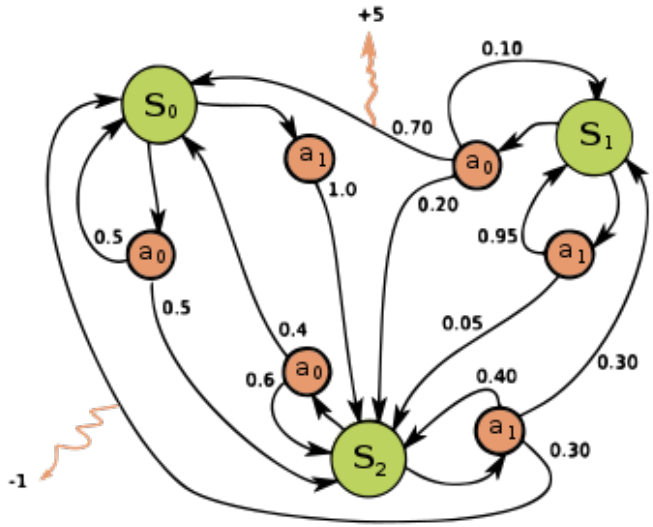


Photo Credit: @ https://en.wikipedia.org/wiki/Markov_decision_process

RL for Continuous-time Systems

Optimal control problem

- Agent

$$u = k(x).$$

- Environment

$$\dot{x} = f(x) + g(x)u = F(x, u).$$

- Average value function

$$V_a(x(t), u(t)) = \lim_{T \to \infty} \frac{1}{T} \int_t^{t+T} r(x(\tau), u(\tau)) d\tau.$$

Optimal average cost

$$\lambda^*.$$

Optimal control problem

Optimize the value function

$$V_a(x(t), u(t)) = \lim_{T \to \infty} \frac{1}{T} \int_t^{t+T} r(x(\tau), u(\tau)) d\tau.$$

with respect to

$$\dot{x} = f(x) + g(x)u = F(x, u).$$

Do I sell optimal control problem again with the fancy name RL???

Why to use RL at all?

- No analytical solution in general
- Difficult to solve
- Computation is offline
- Exact knowledge of dynamics is required

The key to RL

Bellman Principle of Optimality

&

Finding Fixed point equations!

$$\lambda^* = \min_u [r(x(t), u(t)) + \nabla V_\infty^{*T} F(x, u)].$$

Integral Reinforcement Learning (IRL)

Bellman in integral form [5] [6]

$$V^*(x(t)) = \min_u \int_t^{t+T} r(x(\tau), u(\tau)) - \lambda^* d\tau + V^*(x(t+T))).$$

No dynamics is involved.

[5] Vrabie et al. *Adaptive optimal control for continuous-time linear systems based on policy iteration*, Automatica, 2009

[6] Adib Yaghmaie et al. *Reinforcement Learning for Average Cost Optimal Control Problem with an Application in Tracking*, TAC, Under revision, 2018.

Methods to solve Bellman equation

- Monte-Carlo
- Temporal Difference Learning
- Least squares Temporal Difference Learning
- Their variations/combinations

Algorithm 1 Policy Iteration Algoritm

1: **Initialize:** $u^{(0)}$, $k = 0$.
2: **repeat**
3: **Step 1:** Policy evaluation

$$V_\infty^{(k)}(x(t)) = \int_t^{t+\Delta T} Q(x) + u^{(k)T} R u^{(k)} d\tau + V_\infty^{(k)}(x(t+\Delta T)) - \lambda^{(k)} \Delta T.$$

4: **Step 2:** Policy improvement

$$u^{(k+1)} = -\frac{1}{2} R^{-1} g^T \nabla V_\infty^{(k)}.$$

5: **until** Convergence

Simulation results

Tracking problem

- System

$$\dot{x}_s = f_s(x_s) + g_s(x_s)u.$$

- Reference

$$\dot{x}_r = f_r(x_r).$$

- Design u such that

$$x_s - x_r \to 0.$$

Solution to tracking problem

Theorem 3.1 *If a feedback controller u_{fb} stabilizes the system and*

$$f_r(x_r) = f_s(x_r) + g_s(x_r)u_{ff}(x_r),$$

then the following controller solves the tracking problem

$$u(x_r, x_r) = u_{ff}(x_r) + u_{fb}(x_s - x_r).$$

System Model

System [6]

$$\dot{x}_s = \begin{bmatrix} 0 & 1 \\ -5 & -0.5 \end{bmatrix} x_s + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

Reference

$$\dot{x}_r = \begin{bmatrix} 0 & 1 \\ -5 & 0 \end{bmatrix} x_r, \quad x_{r0} = [0, 0.5\sqrt{5}]^T.$$

Tracking problem

$$\lim_{t\to\infty} e = \lim_{t\to\infty} (x_s - x_r) \to 0.$$

[6] Adib Yaghmaie et al. *Reinforcement Learning for Average Cost Optimal Control Problem with an Application in Tracking*, TAC, Under revision, 2018.

Define $x = [e^T, x_r^T]^T$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5 & -0.5 & 0 & -0.5 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -5 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u.$$

Running cost

$$r(x, u) = x^T \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + u^2.$$

Analytical solution

$$u = k_{fb}e + k_{ff}x_r.$$

Offline solution-Feedback gain

$$ARE(\begin{bmatrix} 0 & 1 \\ -5 & -0.5 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, 1)$$

Feedback controller

$$u_{fb}(e) = k_{fb}e = \begin{bmatrix} -6.18 & -10.11 \end{bmatrix} e.$$

Offline solution-Feedforward gain

Tracking condition

$$f_r(x_r) = f_s(x_r) + g_s(x_r)u_{ff}(x_r).$$

Feedforward controller

$$u_{ff}(x_r) = k_{ff}x_r = \begin{bmatrix} 0 & 0.5 \end{bmatrix} x_r.$$

Offline solution-Tracking controller

$$u = \begin{bmatrix} -6.18 & -10.11 \end{bmatrix} e + \begin{bmatrix} 0 & 0.5 \end{bmatrix} x_r$$

RL framework

Optimize the value function

$$V_a(x(t), u(t)) = \lim_{T \to \infty} \frac{1}{T} \int_t^{t+T} e^T Q e + u^2 d\tau.$$

with respect to

$$\begin{bmatrix} \dot{e} \\ \dot{x}_r \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5 & -0.5 & 0 & -0.5 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -5 & 0 \end{bmatrix} \begin{bmatrix} e \\ x_r \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u$$

which is unknown.

RL framework

Value function $V_\infty^* = w^T \Phi(x)$

$$\Phi(x) = [x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2,\ x_2x_3,\ x_2x_4,\ x_3^2,\ x_3x_4,\ x_4^2].$$

Optimal controller

$$u^* = -\frac{1}{2} R^{-1} g^T \nabla \Phi w.$$

RL solution and comparison with offline method

Analytical	$w^* =$	[116.14	12.36	#	#
		10.11	#	#	#
		#	#]		
	$K^* =$	[-6.18	-10.11	0	0.5]
	$\lambda^* =$	0.1563			
RL Algorithm	$\hat{w}^{(9)} =$	[116.14	12.36	-4.51	-0.06
		10.11	0.034	-0.96	0
		-0.12	0]		
	$K^{(9)} =$	[-6.18	-10.11	-0.01	0.48]
	$\hat{\lambda}^{(9)} =$	0.1559			

State and control

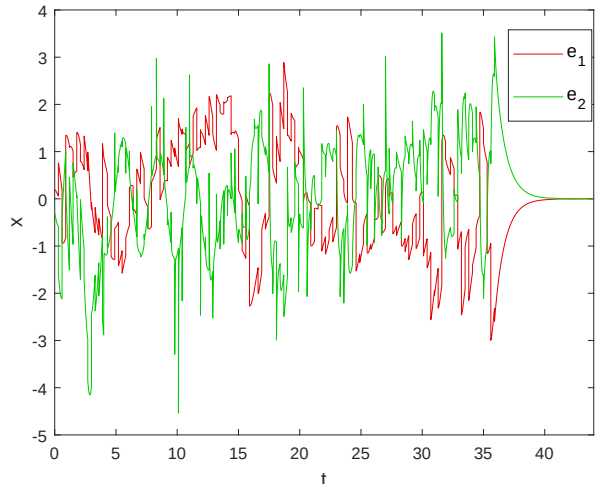


Figure: state e

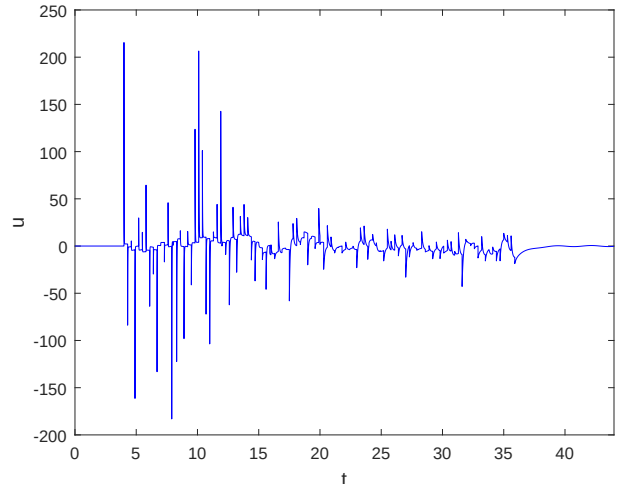


Figure: Control

Weights

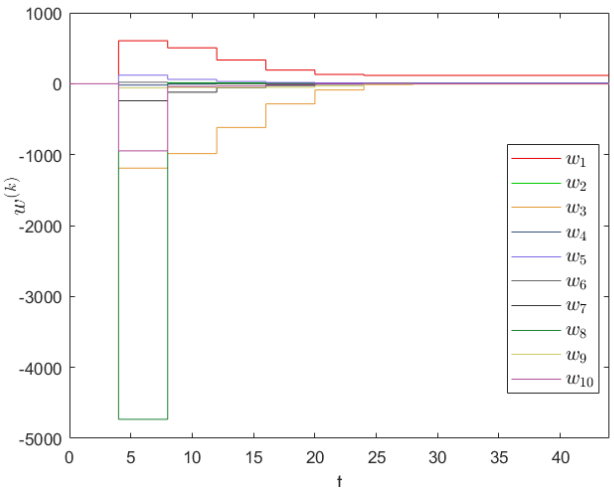


Figure: The weight $w^{(k)}$

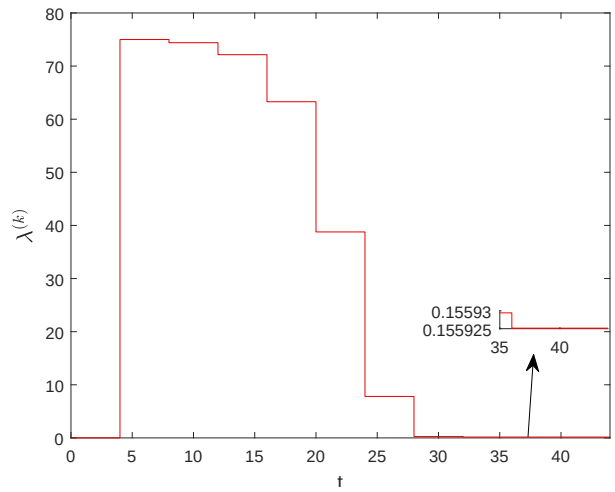
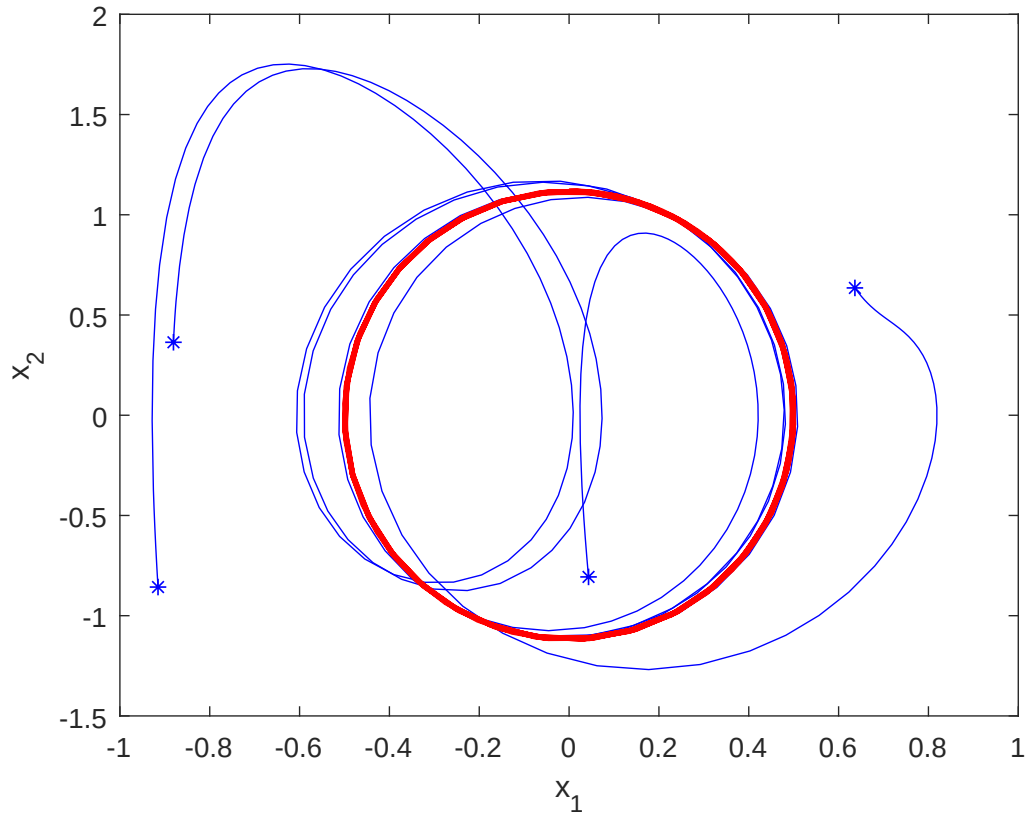


Figure: The average cost $\lambda^{(k)}$

Tracking from random initial condition



Open Problems

- Stability of dynamic system
- Really model-free?
- Deep RL in control?

Farnaz Adib Yaghmaie

farnaz.adib.yaghmaie@liu.se

www.liu.se

LINKÖPING UNIVERSITY