EE 290-005 Integrated Perception, Learning, and Control **Spring 2021**

Lecture 8: Reinforcement Learning and LQR

Scribes: Mihaela Curmei, Haozhi Qi, Aravind Srinivas, Simon Zhai

Presented by: Yibin Li, Kshitij Kulkarni, Jason Zhou, Harry Zhang

8.1 Safe and Data-Efficient Learning for Robotics

By professor Somil Bansal, USC

This work [1] is at the intersection of Learning & Perception and Control Theory.

8.1.1 Main Question

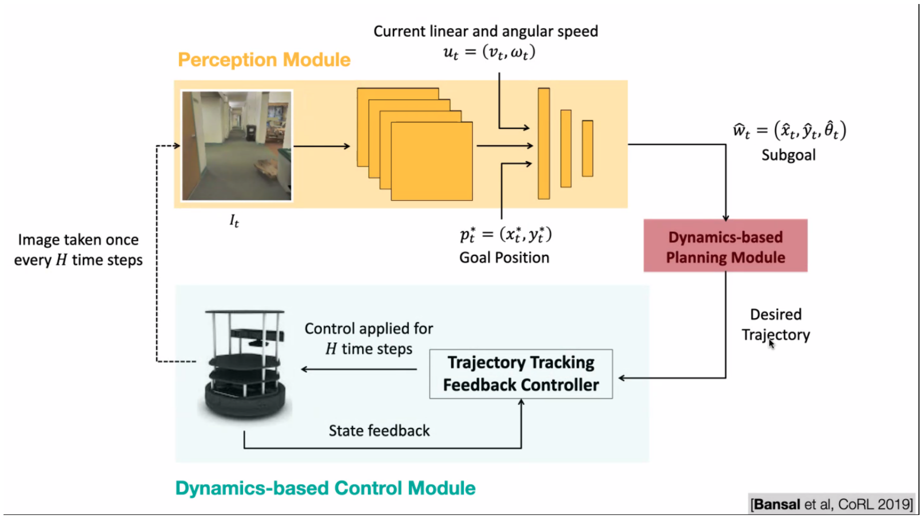
Navigation in unknown environments: *How can a robot with a monocular RGB camera navigate efficiently to a goal state in an unknown environment*

8.1.2 Autonomous driving

Approaches to solve the problem:

- **End-to-End (E2E) Learning.** Pixels to Control actions: Output direct control commands.
  - **Pros:** It can generalize to unknown environments.
  - **Cons:** The sample complexity is very high.
- **Mapping + Planning.** SLAM based approach
  - **Pros:** State of the art in academia and industry.
  - **Cons:** Doesn't contain semantic information (we know that chairs in real world have legs, thus adding learning would be helpful for generalization).

  failure modes: sim to real gap, for example glare on the floor, power lines, branches
- **Learning based Perception + Model based Control (LB-WayPtNav, see Figure 8.1).**
  - Decomposes learning and control and utilize the strengths of each approach;
  - Leads to more modular architectures (e.g., replace robot with quadrotor while keeping perception module unchanged);
  - Provides robustness to sim to real transition (i.e., by data augmentation during training);
  - Can be extended to dynamic environments (i.e., infer from visual cues human behaviour and adapt control outputs to inferred human behaviour)


Current linear and angular speed
$u_t = (v_t, \omega_t)$
Perception Module
$\hat{w}_t = (\hat{x}_t, \hat{y}_t, \hat{\theta}_t)$
Subgoal
I_t
Image taken once every H time steps
$p_t^* = (x_t^*, y_t^*)$
Goal Position
Dynamics-based Planning Module
Desired Trajectory
Control applied for H time steps
Trajectory Tracking Feedback Controller
State feedback
Dynamics-based Control Module

[**Bansal** et al, CoRL 2019]

Figure 8.1: Learning based perception + model based control

8.1.3 Experiments of Learning Based Perception + Model Based Control

- **Dataset:** Stanford 2d-3d dataset, see [2].

- **Training and Testing:** Simulations are conducted in environments derived from scans of real world buildings. Scans from 2 buildings were used to generate training data to train LB-WayPtNav. 185 test episodes (start, goal position pairs) in a 3rd held-out building were used for testing the different methods. Test episodes are sampled to include scenarios such as: going around obstacles, going out of the room, going from one hallway to another.

- **Data Augmentation:** Data augmentation is conducted by applying a variety of random distortions to images, which significantly improves the generalizability of LB-WayPtNav to unseen environments.

- **Results (success rate in terms of reaching the goal):**

  – E2E: 56.16%;

  – LB-WayPtNav: 82.19%

- **Explanation of the Gap:** Pure learning strategies struggle whenever intricate control is required (i.e., narrow corridors).

- **Observations:** Both E2E and LB-WayPtNav improve with more data, and LB-WayPtNav is 10x more data efficient.

8.1.4 Discussion

1. **Q. (Simon Zhai):** “How is the model incorporated in the system?”

   A. (Somil Bansal): There is no model in E2E; in LB-WayPtNav, the model is incorporated via the dynamic-based planning model which constantly output subgoals, and plan the trajectories using the dynamic planning models and the subgoals.

2. **Q. (Frank Chiu):** "What other moving objects did you use? Other robots?"

   A. (Somil Bansal): We tried the approach with moving human, so the model can react to humans moving in the room. We assume that the human does not care about the robot, just the robot needs to react to the human.

3. **Q. (Yi Ma):** "How would it change if you had multiple robots with their own goal"?

   A. (Somil Bansal): Here it is important to model the interaction between robots, we need some kind of game theoretic paradigm to model this interaction.

4. **Q. (Jitendra Malik):** "How to put humans in these environments? Renderings are not realistic; how to to compute realistic behaviour?"

   A. (Somil Bansal): We add 3D mesh and we merge the mapping to get realistic occlusions. In order to model human behaviour we model them via optimal control to incorporate some decision making modules. One issue is that the human is not reactive in the simulated environments.

5. **Q. (Harry Zhang):** "How do you tell what the goal of the robot in real life?"

   A. (Somil Bansal): We have 2D layout and have coordinates. We can do this for short range navigation, but it is challenging for long range navigation.

6. **Comment. (Jitendra Malik):** There are different types of goals: i.e., finding the coordinates, finding the closest chair.

8.2 A tour of Reinforcement Learning from Continuous Control [3]

8.2.1 Motivation and Contribution

- Can we solve continuous control problems via RL (RL typically used for discrete);
- Model based vs model free comparison;
- Strength and weaknesses of RL.

8.2.2 Basic RL/ Control Problem

- The dynamics: $x_{t+1} = f(x_t, u_t, e_t)$.
- The reward function: $R(x_t, u_t)$.
- Optimization:

$$\max \mathbb{E}_{e_t}\left[\sum_{t=0}^{N} R(x_t, u_t)\right], \text{ subject to } x_{t+1} = f(x_t, u_t, e_t). \tag{8.1}$$

- The dynamics

$$\tau_t = (u_1, \ldots, u_{t-1}, x_0, \ldots, x_t). \tag{8.2}$$

- The *decision variables* of the problem is a **policy**: $u_t = \pi_t(\tau_t)$, which uses previous information in the trajectory to compute the control u_t.
- **Goal:** Maximize rewards to obtain a "good" policy.

8.2.3 Types of RL approaches

1. **Model based** System Identification / Supervised Learning

   - Goal:

$$\max \mathbb{E}_{e_t}\left[\sum_{t=0}^{N} R(x_t, u_t)\right] \tag{8.3}$$
$$\text{subject to } x_{t+1} = f(x_t, u_t, e_t), u_t = \pi_t(\tau_t),\ f(x_t, u_t, e_t) \text{ is unknown.}$$

   - A naïve method: find the dynamic model $f(x_t, u_t, e_t)$ by injecting a random probing sequence u_t and measure the response $x_{t+1} \approx \varphi(x_t, u_t) + v_t$.
   - φ could be non-parametric approximation of a neural network, e.g., fit a least square with supervised learning

$$\hat{\varphi} = \arg\min \sum_{t=0}^{N-1} \|x_{t+1} - \varphi(x_t, u_t)\|^2 . \tag{8.4}$$

2. **Model free (Approximate Dynamic Programming and Policy Search):**

   - ADP: We can setup the Q function for Q learning using Bellman's Principle of optimality

$$Q(x, u) = \max \left\{ \mathbb{E}_{e_t}\left[\sum_{t=0}^{N} R(x_t, u_t)\right]\right\}, \tag{8.5}$$
$$\text{subject to } x_{t+1} = f(x_t, u_t, e_t),\ (x_0, u_0) = (x, u).$$

   - Define the terminal Q function to be $Q_N(x, u) = R(x, u)$.
   - The Bellman operator

$$Q_k(x, u) = R(x, u) + \mathbb{E}_e\left[\max_{u'} Q_{k+1}(f(x, u, e), u')\right]. \tag{8.6}$$

   - An optimal policy satisfies

$$\pi_k(\tau_k) = \arg\max Q_k(x_k, u). \tag{8.7}$$

   - For infinite time horizon, introduce the discount factor ($0 < \gamma < 1$):

$$\max \left\{ (1-\gamma)\mathbb{E}_{e_t}\left[\sum_{t=0}^{N} \gamma^t R(x_t, u_t)\right]\right\} \tag{8.8}$$
$$\text{subject to } x_{t+1} = f(x_t, u_t, e_t),\ u_t = \pi_t(\tau_t).$$

   - Then we have the Bellman equation for the discounted case

$$Q_\gamma(x, u) = R(x, u) + \gamma\mathbb{E}_e\left[\max_{u'} Q_\gamma(f(x, u, e), u')\right]. \tag{8.9}$$

8.2.4 Linear Quadratic Regulator LQR

- The simplest class of control problems that exhibit nontrivial results are:

$$\min \mathbb{E}_{e_t}\left[\frac{1}{2}\sum_{t=0}^{N} x_t^\top Q x_t + u_t^\top R u_t + \frac{1}{2} x_{N+1}^\top S x_{n+1}\right] \tag{8.10}$$
$$\text{subject to } x_{t+1} = Ax_t + Bu_t + e_t,\ u_t = \pi_t(\tau_t),\ Q, R, S\text{are PSD.}$$

- Under known dynamics, the optimal policy is linear state feedback:

$$u_t = -K_t x_t \tag{8.11}$$

- When the horizon is infinite, the policy is stationary $u_t = -Kx_p$, where K can be obtained via solving a Ricati equation.

Least Squares Estimation for System Identification

- When A, B are unknown, we can use the least-square estimation:

$$\min_{A,B} \sum_{t=0}^{N-1} \|x_{t+1} - Ax_t - Bu_t\|^2 \tag{8.12}$$

- Note that the least-square estimation may not be robust (e.g., consider the case where A has an unstable eigenvalue).

8.2.5 Discussion

1. **Comment. (Jitendra Malik, Shankar Sastry):** *In the praise of RL.* Separating ID from control only works when the model is simple. Ben Recht says that you can perform system id; however there are system that are more complicated, we might be able to simulate but we might not be able to nicely describe the dynamics-the proof of this is in practice, as we have good simulation environment, who cares about sampling complexity.

   If you know that system is linear, then estimating via LS is not the best way to get accurate estimates.

2. **Comment. (Jitendra Malik):** Manipulation with contact is super hard to operate in the paradigm of system id.

3. **Comment. (Shankar Sastry):** When you have a model, use it! When you don't have a model, use data. Use at least some kind of information about causation (see works of Professor Judea Pearl) – what are inputs, what are outputs in a problem.

4. **Comment. (Yi Ma):** Even if you don't have a model (but have some oracle access), nevertheless adding some kind of even more simple modeling assumption can be useful. So some kind of model based intuition/thinking is often useful.

5. **Comment. (Claire Tomlin):** In economics they try to infer causal behaviour from observational data because RCT are too expensive/unfeasible. They look at very complex behaviours, and there is still a lot of use in adding modeling assumptions

6. **Comment. (Shankar Sastry):** If you mix together inputs and outputs in observational data, can you label the inputs and the outputs? This is a surrogate for causality.

7. **Comment. (Yi Ma):** Recent work on knockoffs (see works of Professor Emannuel Candès) to get at the above problem.

8. **Comment. (Claire Tomlin):** It is interesting to look into algorithmic causality and approaches to scale.

8.3 RL and Control as Probabilistic Inference [4]

This paper makes connections between RL and probabilistic inference on a graphical model. This enables the use of an array of tools for probabilistic graphical models (PGM).

The punchline of the paper is that we can embed a RL problem into a PGM (either as exact probabilistic inference + structured variational inference)

8.3.1 RL as PGM

The RL objective

$$\theta^* \doteq \arg\max_\theta \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim p(s,a|\theta)}[r(s_t,a_t)] \tag{8.13}$$

can be viewed as a PGM in the following fashion:

$$p(\tau) = p(s_1, a_1, \dots, s_T, a_T|\theta) = p(s_1) \prod_{t=1}^{T} \underbrace{p(a_t|s_t,\theta)}_{\text{policy}} \underbrace{p(s_{t+1}|s_t,a_t)}_{\text{state dynamics}} \tag{8.14}$$

Generalize of RL/optimal control to PGM:

- **Deterministic Dynamics:** exact probabilistic inference.
- **Stochastic Dynamics:** structured variational inference.

8.3.2 Benefits of Control as Probabilistic Inference

- Able to model suboptimal behavior
- Able to utilize inference; algorithms to solve control and planning problems;
- Able to rationalize why stochastic behavior may be preferred.

8.3.3 Connection to Bellman Backup

- Define the Q-function and value function as follows:

$$Q(s,a) = \log \beta_t(s_t,a_t),\ V(s_t) = \log \beta_t(s_t). \tag{8.15}$$

- Marginalizing the action, obtain the "soft-max" relationship between V and Q:

$$V(s_t) = \log \int_{\mathcal{A}} \exp(Q(s_t,a_t))da_t \approx \max_{a_t} Q(s_t,a_t). \tag{8.16}$$

- Obtain the Bellman backup (stochastic dynamics):

$$Q(s_t,a_t) = r(s_t,a_t) + \log \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)}[\exp(V(s_{t+1}))]. \tag{8.17}$$

8.3.4 Optimization Objective of a Deterministic Dynamics

- Consider a deterministic dynamics, the optimal trajectory distribution is given by

$$p(\tau) = \left[p(s_1)\prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\right] \exp\left(\sum_{t=1}^{T} r(s_t, a_t)\right), \tag{8.18}$$

  and the trajectory from executing the policy is given by

$$\hat{p}(\tau) \propto \mathbf{1}_{\{p(\tau)\neq 0\}} \prod_{t=1}^{T} \pi(a_t|s_t), \tag{8.19}$$

  where $\mathbf{1}_{\{\cdot\}}$ is the indicator function.

- Next, we want to make $p(\tau)$ and $\hat{p}(\tau)$ as close as possible. Consider the KL-divergence between $p(\tau)$ and $\hat{p}(\tau)$:

$$\begin{aligned}
& - D_{KL}(\hat{p}(\tau)||p(\tau)) \\
=& \mathbb{E}_{\tau\sim\hat{p}(\tau)}\Big[\log p(s_1) + \sum_{t=1}^{T}(\log p(s_{t+1}|s_t, a_t) + r(s_t, a_t)) \\
& - \log p(s_1) - \sum_{t=1}^{T}(\log p(s_{t+1}|s_t, a_t) + \log\pi(a_t|s_t))\Big] \\
=& \mathbb{E}_{\tau\sim\hat{p}(\tau)}\left[\sum_{t=1}^{T} r(s_t, a_t) - \log\pi(a_t|s_t)\right] \\
=& \sum_{t=1}^{T}\mathbb{E}_{\tau\sim\hat{p}(\tau)}\left[r(s_t, a_t) - \log\pi(a_t|s_t)\right] \\
=& \sum_{t=1}^{T}\left\{\mathbb{E}_{(s_t,a_t)\sim\hat{p}(s_t,a_t)}\left[r(s_t, a_t)\right] + \mathbb{E}_{s_t\sim\hat{p}(s_t)}[\mathcal{H}(\pi(a_t|s_t))]\right\},
\end{aligned} \tag{8.20}$$

  where $\mathcal{H}(\cdot)$ is the entropy function.

8.3.5 Optimization Objective of a Stochastic Dynamics

In the stochastic case, both initial state distribution and the state transition distribution depend on the optimal variables, so the KL-divergence is given as

$$- D_{KL}(\hat{p}(\tau)||p(\tau)) = \mathbb{E}_{\tau\sim\hat{p}(\tau)}\left[\log p(s_1) + \sum_{t=1}^{T} r(s_t, a_t) + \log p(s_{t+1}|s_t, a_t)\right] + \mathcal{H}(\hat{p}(\tau)) \tag{8.21}$$

8.3.6 The Recursive case

In the recursive case, the objective can be rewritten as

$$\begin{aligned}
& \mathbb{E}_{(s_t,a_t)\sim\hat{p}(s_t,a_t)}[r(s_t, a_t) - \log\pi(a_t|s_t)] + \mathbb{E}_{(s_t,a_t)\sim\hat{p}(s_t,a_t)}[\mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)}V(s_{t+1})] \\
=& \mathbb{E}_{s_t\sim\hat{p}(s_t)}\left[-D_{KL}\left(\pi(a_t|s_t)\middle|\middle|\frac{\exp(Q(s_t, a_t))}{\exp(V(s_t))}\right) + V(s_t)\right],
\end{aligned} \tag{8.22}$$

accordingly, we define

$$\begin{aligned} Q(s_t, a_t) &= r(s_t, a_t) + \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)[V(s_{t+1})]} \\ V(s_t) &= \log \int_{\mathcal{A}} \exp(Q(s_t, a_t)) da_t. \end{aligned} \tag{8.23}$$

8.3.7 Connection to Structured Variational Inference

In structured variational inference, the goal is to approximate some distribution $p(y)$ with another (potentially simpler) distribution $q(y)$. Typically, $q(y)$ is taken to be some tractable factorized distribution, such as a product of conditional distributions connected in a chain or tree, which lends itself to tractable exact inference. In our case, we aim to approximate $p(\tau)$, given by

$$p(\tau) = \left[p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t) \right] \exp\left(\sum_{t=1}^{T} r(s_t, a_t) \right) \tag{8.24}$$

via the distribution

$$q(\tau) = q(s_1) \prod_{t=1}^{T} q(s_{t+1}|s_t, a_t) q(a_t|s_t). \tag{8.25}$$

8.3.8 Maximum Entropy Policy Gradients

In terms of maximizing the entropy, the objective function is written as

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_1,a_t)\sim q(s_t,a_t)}[r(s_t, a_t) + \mathcal{H}(q_\theta(a_t|s_t))] \tag{8.26}$$

and the gradient is:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_{t=1}^{T} \nabla_\theta \mathbb{E}_{(s_t,a_t)\sim q(s_t,a_t)}[r(s_t, a_t) + \mathcal{H}(q_\theta(a_t|s_t))] \\ &= \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim q(s_t,a_t)} \left[\nabla_\theta \log q_\theta(a_t|s_t) \left(\sum_{t'=t}^{T} r(s_{t'}, a_{t'}) - \log q_\theta(a_{t'}|s_{t'}) - 1 \right) \right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim q(s_t,a_t)} \left[\nabla_\theta \log q_\theta(a_t|s_t) \left(\sum_{t'=t}^{T} r(s_{t'}, a_{t'}) - \log q_\theta(a_{t'}|s_{t'}) - b(s_{t'}) \right) \right] \end{aligned} \tag{8.27}$$

8.3.9 Maximum Entropy Actor-Critic Algorithms

A simple and straightforward approach is to represent them with parameterized functions $Q_\phi(s_t, a_t)$ and $V_\psi(s_t)$, with parameters ϕ and ψ, and optimize the parameters to minimize a squared error objectives:

$$\begin{aligned} \mathcal{E}(\phi) &= \mathbb{E}_{(s_t,a_t)\sim q(s_t,a_t)} \left[\left(r(s_t, a_t) + \mathbb{E}_{q(s_{t+1}|s_t,a_t)}[V_\psi(s_{t+1})] - Q_\phi(s_t, a_t) \right)^2 \right] \\ \mathcal{E}(\psi) &= \mathbb{E}_{s_t\sim q(s_t)} \left[\left(\mathbb{E}_{a_t\sim q(a_t|s_t)}[Q_\phi(s_t, a_t) - \log q(a_t|s_t)] - V_\psi(s_t, a_t) \right)^2 \right]. \end{aligned} \tag{8.28}$$

8.3.10 Related Approaches

- Boltzmann Exploration
- Entropy Regularization
- Variational Policy Search and Expectation Maximization
- KL-Divergence Constraints for Policy Search

8.3.11 Conclusion

This paper discusses how the maximization of a reward function in Markov decision process can be formulated as an inference problem in a particular graphical model, and how a set of update equations similar to the well-known value function dynamic programming solution can be recovered as the direct consequence of applying structured variational inference to this graphical model.

The classical maximum expected reward formulation emerges as a limiting case of this framework, while the general case corresponds to a maximum entropy variant of reinforcement learning or optimal control, where the optimal policy not only aims to maximize the expected reward, but also aims to maintain high entropy.

8.3.12 Discussion

1. **Q. (Yi Ma):** "Why does the entropy of the policy pop out? What is the intuition behind this?"

   A. (Harry Zhang): Mathematically it turns out that minimizing KL divergence is equivalent to optimizing reward as well as entropy, thus encouraging 'good' behaviour as well as exploration.

References

[1] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Conference on Robot Learning*, pp. 420–429, PMLR, 2020. 8-1

[2] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543, 2016. 8-2

[3] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019. 8-3

[4] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018. 8-6