Data-driven and Learning-based Control

2nd hands-on session

Erica Salvato

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Table of Contents

What we know so far?

Given:

- a DT time-invariant dynamical system $x^{(k+1)} = f\left(x^{(k)}, u^{(k)}\right)$
- a task
- a performance index

We can compute the optimal controller $\pi^*\left(x^{(k)}\right)$ by solving an optimal control problem defined as follows:

$$\pi^* = \arg\max_{\pi} J \qquad \text{if } h = \text{Objective reward function}$$

$$\pi^* = \arg\min_{\pi} J \qquad \text{if } h = \text{Objective cost function}$$

s.t.:

$$x^{(k+1)} = f\left(x^{(k)}, u^{(k)}\right)$$

What we know so far?

1 A brief recap

By exploiting Bellman's principle of optimality we can solve it by relying on

- Dynamic programming (bottom-up approach)
- Bellman's optimality equations
  - value iteration (forward approach)
  - policy iteration (forward approach)

What we know so far?

By exploiting Bellman's principle of optimality we can solve it by relying on

- Dynamic programming (bottom-up approach)
- Bellman's optimality equations
  - value iteration (forward approach)
  - policy iteration (forward approach)

However, if we are interested in a state-feedback controller, we must have access to state measures.

What we know so far?

By exploiting Bellman's principle of optimality we can solve it by relying on

- Dynamic programming (bottom-up approach)
- Bellman's optimality equations
  - value iteration (forward approach)
  - policy iteration (forward approach)

However, if we are interested in a state-feedback controller, we must have access to state measures.

Furthermore, if we are interested in reaching a state from an initial state within a fixed number of steps, we need the terminal state to be reachable.

In the special case of linear dynamical systems, we can check the observability and reachability properties of the system:

Necessary and sufficient condition for full observability

If n is the state size of the dynamical system. Necessary and sufficient condition for a system to be **fully observable** is that

$$\operatorname{rank}\left\{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\right\} = n$$

What we know so far?

Necessary and sufficient condition for full reachability

If n is the state size of the dynamical system. Necessary and sufficient condition for a system to be **fully reachable** is that

$$\text{rank}\left\{\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}\right\} = n$$

Necessary and sufficient condition for full reachability

If n is the state size of the dynamical system. Necessary and sufficient condition for a system to be **fully reachable** is that

$$\text{rank}\left\{\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}\right\} = n$$

Sometimes we don't need the full reachability, but we can also accept that only the terminal state can be reached in H steps.

What we know so far?

1 A brief recap

If we want to verify that a state x_{des} is reachable in H steps starting from an initial condition $x^{(0)} = x_{\text{ini}}$, we have to:

1. Check that the H-steps reachability matrix has rank H
$$\text{rank}\left\{\begin{bmatrix} B & AB & \dots & A^{H-1}B \end{bmatrix}\right\} = H$$
It ensures that there exists a subset of $\mathcal{X}$ containing states reachable in H steps
2. Solve the following:
$$x_{\text{des}} - A^H x_{\text{ini}} = \sum_{j=0}^{H-1} A^{H-(j+1)} B u^{(j)} = \begin{bmatrix} B & AB & \dots & A^{H-1}B \end{bmatrix} \begin{bmatrix} u^{(H-1)} \\ u^{(H-2)} \\ \vdots \\ u^{(0)} \end{bmatrix}$$
That admits solution when $x_{\text{des}} - A^H x_{\text{ini}} \in \text{Im}\left(\begin{bmatrix} B & AB & \dots & A^{H-1}B \end{bmatrix}\right)$

What we know so far?

Moreover, we can define infinite horizon LQR problems.
Find $\pi^*\left(x^{(k)}\right)$ solution of:

$$\arg\min_{\pi} \sum_{k=0}^{\infty} x^{(k)^\top} Q x^{(k)} + u^{(k)^\top} R u^{(k)}$$

s.t.:

$$x^{(k+1)} = Ax^{(k)} + Bu^{(k)},$$

$$x^{(0)} = x_{\text{ini}},\ x^{(\infty)} \to 0$$

What we know so far?

1 A brief recap

Moreover, we can define infinite horizon LQR problems.
Find $\pi^* \left(x^{(k)}\right)$ solution of:

$$\arg\min_{\pi} \sum_{k=0}^{\infty} x^{(k)^\top} Q x^{(k)} + u^{(k)^\top} R u^{(k)}$$

s.t.:

$$x^{(k+1)} = Ax^{(k)} + Bu^{(k)},$$

$$x^{(0)} = x_{\text{ini}},\ x^{(\infty)} \to 0$$

Whose solution is given by the **Riccati equations**

$$A^\top PA - P + Q - A^\top PB \left(R + B^\top PB\right)^{-1} B^\top PA = 0$$

Then $K = \left(R + B^\top PB\right)^{-1} B^\top PA$

What we know so far?

We can also define finite horizon LQR problems.
Find $\pi^*\left(x^{(k)}\right) = -K_k x^{(k)}$ solution of:

$$\arg\min_{\pi} \; x^{(H)^\top} Q_H x^{(H)} + \sum_{k=0}^{H-1} x^{(k)^\top} Q x^{(k)} + u^{(k)^\top} R u^{(k)}$$

s.t.:

$$x^{(k+1)} = A x^{(k)} + B u^{(k)},$$

$$x^{(0)} = x_{\text{ini}},\; x^{(H)} = 0$$

What we know so far?

We can also define finite horizon LQR problems.
Find $\pi^*\left(x^{(k)}\right) = -K_k x^{(k)}$ solution of:

$$\arg\min_{\pi} \; x^{(H)^\top} Q_H x^{(H)} + \sum_{k=0}^{H-1} x^{(k)^\top} Q x^{(k)} + u^{(k)^\top} R u^{(k)}$$

s.t.:

$$x^{(k+1)} = Ax^{(k)} + Bu^{(k)},$$
$$x^{(0)} = x_{\text{ini}}, \; x^{(H)} = 0$$

Whose solution is given by the **Riccati recursion**

$$P_k = A^\top P_{k+1} A - P_{k+1} + Q - A^\top P_{k+1} B \left(R + B^\top P_{k+1} B\right)^{-1} B^\top P_{k+1} A$$

starting from $P_H = Q_H$. Then $K_k = \left(R + B^\top P_{k+1} B\right)^{-1} B^\top P_{k+1} A$

What we know so far?

However, LQR works only if some assumptions are satisfied:

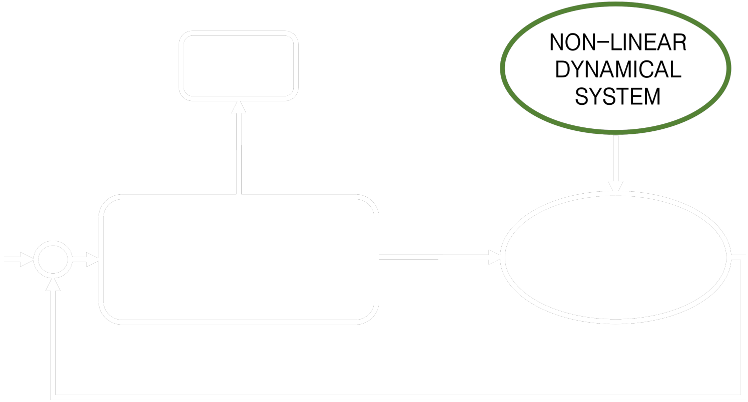
$$\mathbf{Q} = \mathbf{Q}^\top,\ \mathbf{Q} \geq \mathbf{0}$$

$$\mathbf{Q_H} = \mathbf{Q_H}^\top,\ \mathbf{Q_H} \geq \mathbf{0}$$

$$\mathbf{R} = \mathbf{R}^\top,\ \mathbf{R} > \mathbf{0}$$

- **The state is accessible** or at least $(A,\ C)$ observable
- **The state $\mathbf{x}^{(\mathbf{H})} = \mathbf{0}$ must be reachable in $\mathbf{H}$ steps**, in the better case $(A,\ B)$ fully reachable

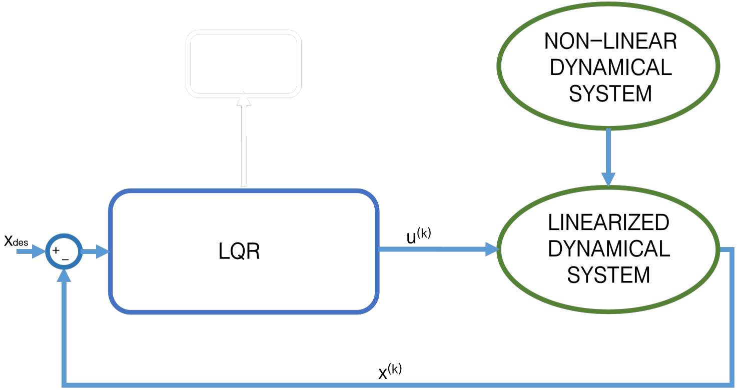
What if we wanted to apply LQR to non-linear systems?



What we know so far?

1 A brief recap

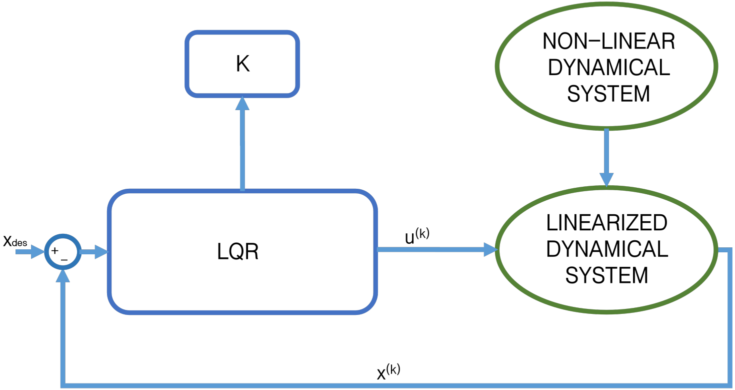
What if we wanted to apply LQR to non-linear systems?



What we know so far?

1 A brief recap

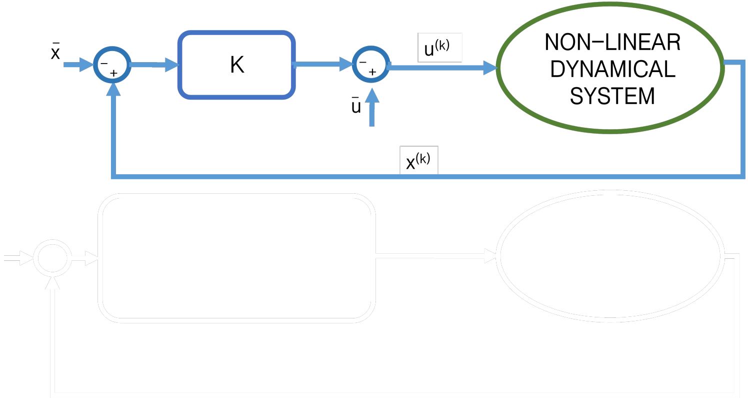
What if we wanted to apply LQR to non-linear systems?



What we know so far?

1 A brief recap

What if we wanted to apply LQR to non-linear systems?



What we know so far?

Or apply the iLQR:

1. Record a state-input trajectory of H steps from the non-linear system $\left\{\bar{x}^{(k)}, \bar{u}^{(k)}\right\}_{k=0}^{H-1}$

What we know so far?

1 A brief recap

Or apply the iLQR:

1. Record a state-input trajectory of H steps from the non-linear system $\left\{\bar{x}^{(k)}, \bar{u}^{(k)}\right\}_{k=0}^{H-1}$

2. Linearize the non-linear system around the trajectory

$$\delta x^{(k+1)} = A(k)\delta x^{(k)} + B(k)\delta u^{(k)}$$

where $A(k) = \left[\frac{\partial f}{\partial x}\right]_{x=\bar{x}^{(k)},u=\bar{u}^{(k)}}$, $B(k) = \left[\frac{\partial f}{\partial u}\right]_{x=\bar{x}^{(k)},u=\bar{u}^{(k)}}$, while $\delta x^{(k)} = x^{(k)} - \bar{x}^{(k)}$ and $\delta u^{(k)} = u^{(k)} - \bar{u}^{(k)}$ are the variations of $x^{(k)}$ and $u^{(k)}$ from their nominal values.

The cost term then becomes

$$\left(x^{(k)} + \delta x^{(k)}\right)^\top Q \left(x^{(k)} + \delta x^{(k)}\right) + \left(u^{(k)} + \delta u^{(k)}\right)^\top R \left(u^{(k)} + \delta u^{(k)}\right)$$

What we know so far?

3. Compute the second-order Taylor series expansion of the cost around the nominal trajectory thus obtaining a quadratic approximation

$$\hat{x}^{(k)^\top} Q\widehat{(k)}\hat{x}^{(k)} + \hat{u}^{(k)^\top} R\hat{u}^{(k)}$$

   where $\hat{x}^{(k)}$ and $\hat{u}^{(k)}$ is the homogeneous transformation of the time-variant linearization of the non-linear system

What we know so far?

1 A brief recap

4. Solve an LQR problem:
   Find $\pi^*\left(\hat{x}^{(k)}\right) = -K_k\hat{x}^{(k)}$ solution of:

$$\arg\min_{\pi} \sum_{k=0}^{\infty} \hat{x}^{(k)}{}^{\top} \hat{Q}(k)\hat{x}^{(k)} + \hat{u}^{(k)}{}^{\top} R\hat{u}^{(k)}$$

   s.t.:

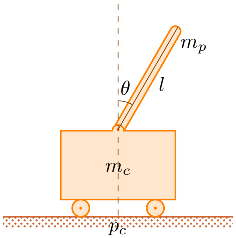
$$\hat{x}^{(k+1)} = \begin{bmatrix} A(k) & 0 \\ 0 & 1 \end{bmatrix} \hat{x}^{(k)} + \begin{bmatrix} B(k) \\ 0 \end{bmatrix} \hat{u}^{(k)}$$

   where $\hat{x}^{(k)} = \begin{bmatrix} x^{(k)} - \bar{x}^{(k)} \\ 1 \end{bmatrix}$ and $\hat{u}^{(k)} = u^{(k)} - \bar{u}^{(k)}$.

   The solution is a controller K_k which returns $u^{(k)} = -K_k\left(x^{(k)} - \bar{x}^{(k)}\right) + \bar{u}^{(k)}$

Table of Contents

2 A case study



$$\ddot{\theta} = \frac{g\sin(\theta) + \cos(\theta)\left[\frac{-F - m_p\, l\, \dot{\theta}^2 \sin(\theta)}{m_c + m_p}\right] - \frac{\mu_p \dot{\theta}}{m_p\, l}}{l\left[\frac{4}{3} - \frac{m_p\, cos^2(\theta)}{m_c + m_p}\right]}$$

$$\ddot{p}_c = \frac{F + m_p\, l\left[\dot{\theta}^2 \sin(\theta) - \ddot{\theta}\cos(\theta)\right]}{m_c + m_p}$$

Table of Contents

Hands-on outline

Given the cart and pole system previously defined:

1. Clearly define an optimal control problem for pole balancing

Expectation: Clearly and formally define an optimal control problem able to balance the pole

Hands-on outline

Given the cart and pole system previously defined:

1. Clearly define an optimal control problem for pole balancing

Expectation: Clearly and formally define an optimal control problem able to balance the pole

2. Is the LQR applicable in its simple form on the system? If not what is missed?

Expectation: Formally justify your answer. In the case of a negative answer formally define all the steps you need to perform in order to apply the LQR.

Hands-on outline

3. Design an LQR for the linearized version of the cart and pole obtained in the previous hands-on. Justify all the design choices and perform all the required checks

Expectation: Create a code able to compute the LQR on the linearized cart and pole system obtained in the 1st hands-on. Verify all the assumptions and try different design choices.

Hands-on outline

3 2nd hands-on session

3. Design an LQR for the linearized version of the cart and pole obtained in the previous hands-on. Justify all the design choices and perform all the required checks

Expectation: Create a code able to compute the LQR on the linearized cart and pole system obtained in the 1st hands-on. Verify all the assumptions and try different design choices.

4. Simulate the linearized cart and pole system controlled by an LQR. Plot the significant behaviors

Expectation: Create a code able to simulate the LQR obtained in the previous point on the linearized cart and pole system. Justify by evidence the better design choices adopted

5. Simulate the non-linear cart and pole system controlled by the previously obtained LQR. Plot the significant behaviors

Expectation: Create a code able to simulate the LQR obtained in the previous point on the non-linear cart and pole system.

Hands-on outline

5. Simulate the non-linear cart and pole system controlled by the previously obtained LQR. Plot the significant behaviors

Expectation: Create a code able to simulate the LQR obtained in the previous point on the non-linear cart and pole system.

6. Comment results obtained in the two simulations

Expectation: Comment on and justify differences in the simulation based on the theory covered in class.

- Jupyter Notebooks: solve discrete-time algebraic Riccati equation (DARE).

```
from scipy import linalg as la
p=la.solve_discrete_are(a, b, q, r)
```

Functionalities

- Jupyter Notebooks: solve discrete-time algebraic Riccati equation (DARE).

```python
from scipy import linalg as la
p=la.solve_discrete_are(a, b, q, r)
```

- Matlab Livescripts: solve discrete-time algebraic Riccati equation (DARE).

```matlab

[K,P] = dlqr(A,B,Q,R,Q_N);
```

Questions’ time!

UNIVERSITÀ DEGLI STUDI DI TRIESTE