Linear Quadratic Regulator

Malintha Fernando
Department of Intelligent Systems Engineering
Indiana University

*** Examples and codes have been adopted from the book "Data Driven Science & Engineering - Machine Learning, Dynamical Systems, and Control" by Steve Brunton and Nathan Kutz.**

Outline

- Recap
- Controllability of Linear Systems
- Pole Placement
- Linear Quadratic Regulator
- Inverted Pendulum Example

Control Laws

- A control law is a set of parameters that is used to regulate the control input for a system.

- Consider the control law $u = K(r - x)$ for a linear dynamical system. Here, r is the desired state of the system.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}(\mathbf{r} - \mathbf{x})$$
$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}\mathbf{r} - \mathbf{B}\mathbf{K}\mathbf{x}$$
$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}\mathbf{K}\mathbf{r}$$

- Here, A, B are inherent to the system. Thus, by choosing K appropriately, we can bring the system to stabilization, aka to r.

Linearizing Nonlinear Systems

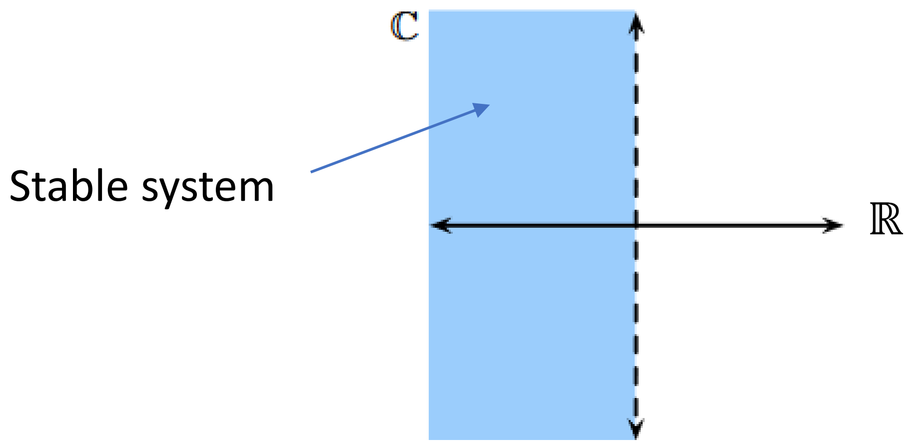
- We can obtain the linear dynamical system matrices from a nonlinear differential equations system as below.
- Evaluate the two matrices at the fixed point $(\bar{x}, \bar{u})$ to get the *numerical* linearized matrices at any point.

$$A = \frac{df}{dx} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \ldots \\ . & . & \frac{\partial f_n}{\partial x_m} \\ . & . & \end{bmatrix}\right|_{\bar{x}} \qquad B = \frac{df}{du} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots \\ \frac{\partial f_2}{\partial u_2} & \frac{\partial f_2}{\partial u_2} & \ldots \\ . & . & \frac{\partial f_n}{\partial u_l} \\ . & . & \end{bmatrix}\right|_{\bar{u}}$$

- In trajectory following, the fixed point could be any *“close enough”* point in the given **trajectory**.

Stability of Linear Systems

- We can analyze the A matrix to check if a system is stable at a fixed point.
- For any **continuous time** dynamical system,
  - If the all the **eigenvalues** of A has negative real parts → system is stable at the fixed point.
  - If any of the **eigenvalues** has positive real parts → unstable at the fixed point.



Controllability

- The ability to place the eigenvalues of a closed loop system with some chosen gain matrix K.
- The controllability matrix $\mathcal{C}$ takes the form,

$$\mathcal{C} = \begin{bmatrix} \mathrm{B} & \mathrm{AB} & \mathrm{A}^2\mathrm{B} & \cdots & \mathrm{A}^{n-1}\mathrm{B} \end{bmatrix}$$

- If $\mathcal{C}$ is **full rank** (if it has n **linearly independent columns**), then the system is controllable.
- If the system is controllable, the <u>eigenvalues can be placed arbitrarily anywhere</u>.

Remark *: Rank of a matrix refers to the <u>maximal number of linearly independent columns</u> of a matrix. If the number of <u>linearly independent columns is equal to the number of columns</u>, it is a full rank matrix.*

Controllability

- Consider the following linear system.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\dot{x} = Ax + Bu$$

- Is this system controllable?

Example

- Consider the following linear system.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\dot{x} = Ax + Bu$$

- Is this system controllable?

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix} = [B \ AB]$$

- This is not a full rank matrix. Thus, the system is not controllable.

Reachability

- If a system is controllable, it also satisfies the reachability condition.
- This means, it is possible to steer the system to any arbitrary state in finite time with some actuation signal.

Pendulum Example

- Nonlinear differential equation:

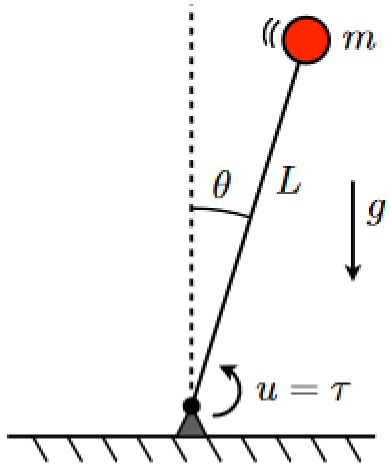
$$\ddot{\theta} = -\frac{g}{L}\sin(\theta) + u.$$

- First order set of equations:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad \Longrightarrow \quad \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{L}\sin(x_1) + u \end{bmatrix}$$

- By doing the partial differentiation we get,

$$\frac{\mathbf{df}}{\mathbf{dx}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L}\cos(x_1) & 0 \end{bmatrix}, \qquad \frac{\mathbf{df}}{\mathbf{du}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

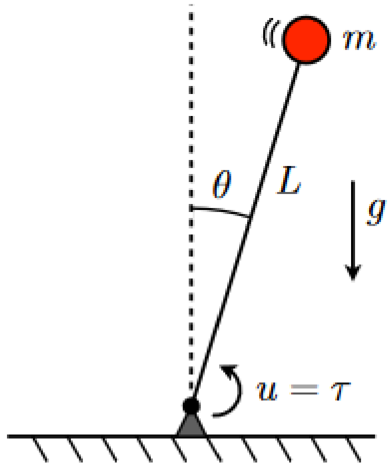


Pendulum Example

- Now evaluate this system at the **fixed points**. Since we have two fixed points, this will yield two A matrices.

$$A_D = \begin{bmatrix} 0 & 1 \\ g/L & 0 \end{bmatrix} \qquad A_U = \begin{bmatrix} 0 & 1 \\ -g/L & 0 \end{bmatrix}$$

- Substituting g, L we will get the corresponding linear system near the fixed points.



$$A_D = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad A_U = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Pole Placement

- Let's consider the control law $u = -K(x)$.
- After applying this control, our system becomes,

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \underbrace{(\mathbf{A} - \mathbf{B}\mathbf{K})}_{\text{New } A \text{ Matrix}}\mathbf{x}.$$

- Choose $K = [5, 5]$ for the system at the upper fixed point.

$$\dot{x} = [A - BK]x$$

$$\dot{x} = \left[\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 5 & 5 \end{bmatrix}\right]x = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix}x$$

- New eigenvalues (<u>Poles</u>) for the matrix becomes $\lambda = -2, \lambda = -3$.
  This means we can stabilize this system using this control law at an unstable fixed point!!!.

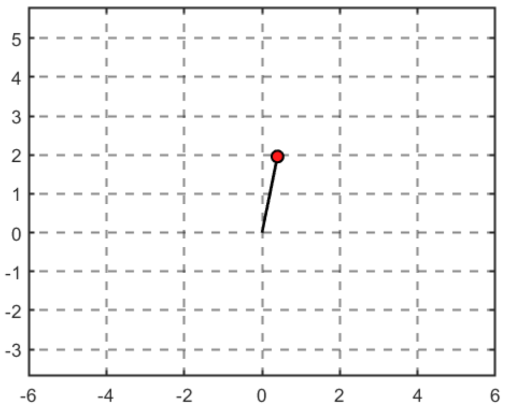
Pole Placement

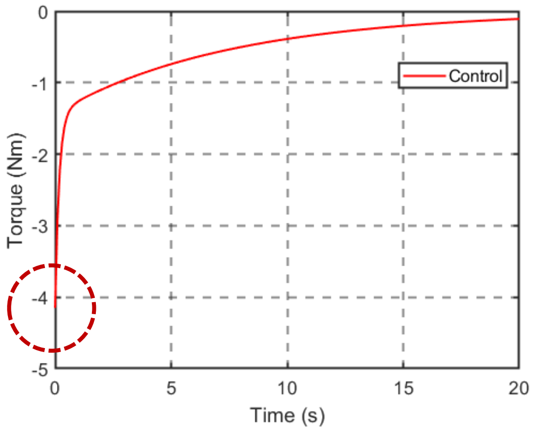
- If the system is **controllable**, we can place its poles anywhere on the complex plane.
- Let's find the gain matrix K for poles of our choice.
- Let $A_u = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, B = [0,1]^T$ and some arbitrarily chosen **stable** poles $p = [-5 \ -6]$.
- We can use Matlab's "`place`" command to find the corresponding gain matrix K.

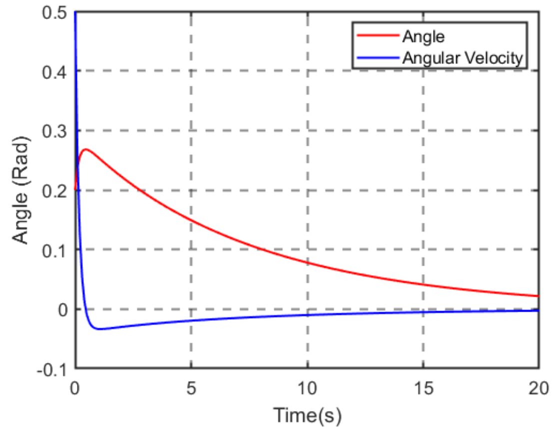
```
Au = [0 1; -1 0];
B = [0; 1];
p = [-5 -6];
k = place(Au, B, p);

>> k = [29.0 11.0]
```

Inverted Pendulum

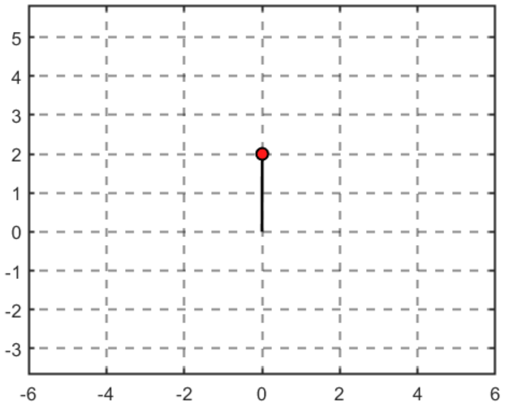


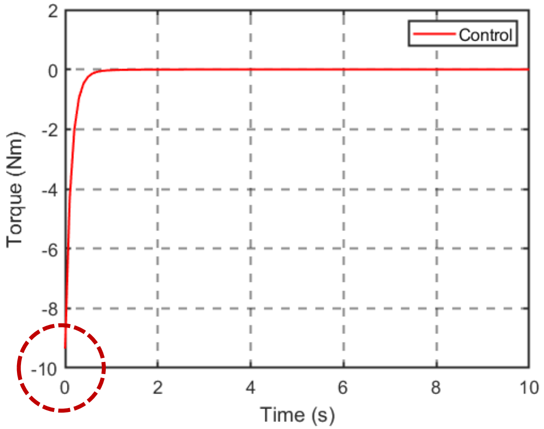


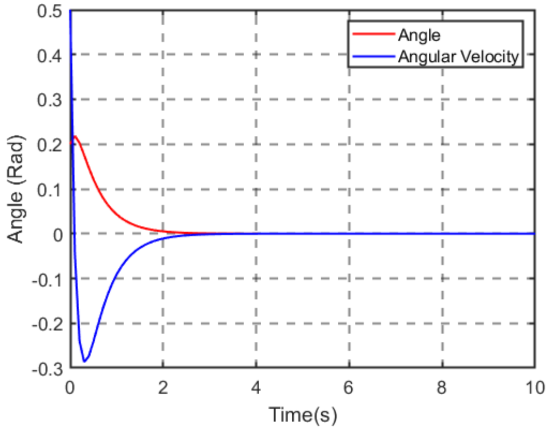


Initial state = [0.2, 0.5].

← Slow Convergence
$\lambda = [-1.5 \ \ -4.5]$







Desired state = [0, 0].

← Aggressive Convergence
$\lambda = [-3.5 \ \ -6.5]$

What is the Best Pole Placement?

- Too small gains can cause the system to
  - converge much slower,
  - jitter in the presence of noise.

- Too aggressive placements can cause the system to
  - exceed maximum control amplitudes requiring much advanced actuators,
  - use more energy.

- We need to arbitrate a balance between these competing goals.

Linear Quadratic Regulator (LQR)

- An optimal control mechanism for **linear dynamical systems** by optimizing a **quadratic cost function**.
- The canonical form for the cost function takes form

$$J = x^T Q x + u^T R u$$

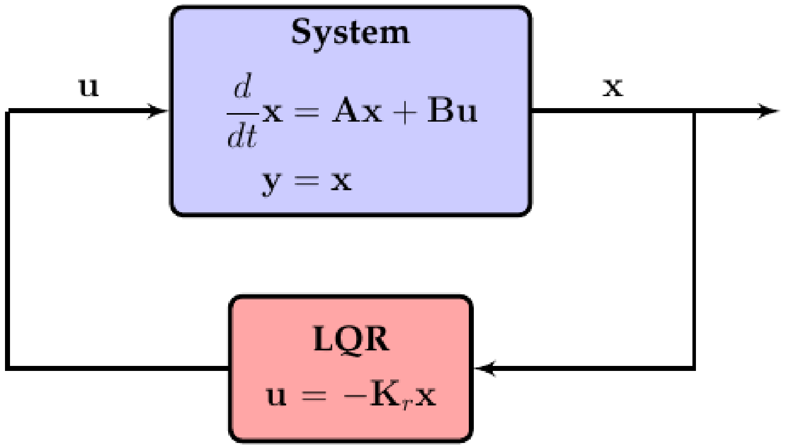
where, $Q \succcurlyeq 0$ and $R \succ 0$ are two diagonal weight matrices.

- The weights lets us balance the contributions of the state error and control amount.
- This is a **full state optimal control** algorithm.

$$y = x \text{ and } C = I, D = 0$$
$$y = Cx + Du$$

Linear Quadratic Regulator (LQR)

- The objective is to minimize the cost and **steer the system to the stability at the desired fixed point**.
- The output of the LQR is a gain matrix K that can be used to calculate the optimal control with the control law $u = -K(x)$.
- Here, x is measured from the new coordinate system whose origin is the fixed point.



Objective Function

- Formally, K is the solution to the mathematical optimization problem,

$$\boxed{\begin{aligned} &minimize\ J(\tau) = \int_0^{\infty} x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)\, d\tau \\ &subject\ to\ \dot{x} = Ax + Bu \\ &u = -Kx \end{aligned}}$$

- This has a closed form solution in the form of the algebraic *Riccati* equation.

$$K = R^{-1}BX$$

$$AX + XA - XBR^{-1}BX + Q = 0$$

- Numerical solutions to this equation is implemented in many programming languages.

Stabilizing the Inverted Pendulum with LQR

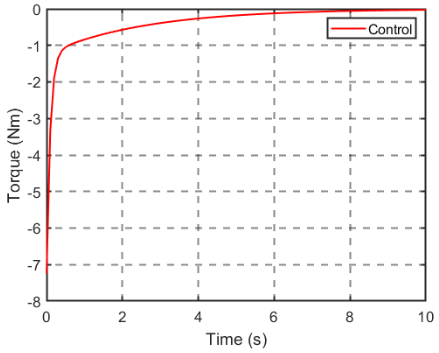
Python equivalent:
https://python-control.readthedocs.io/en/0.9.0/generated/control.lqr.html

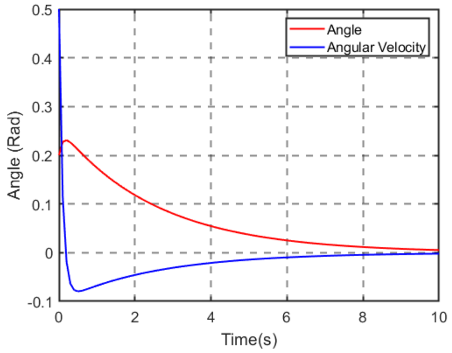
LQR Function

Dynamics integration

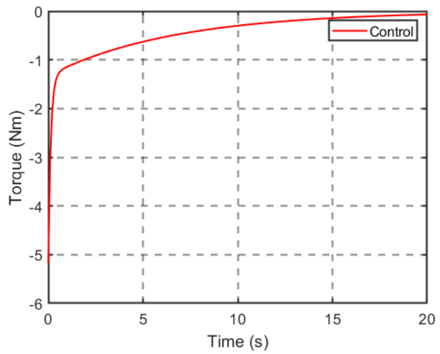
```
Au = [0 1; -1 0];
B = [0; 1];
Q = 100*eye(2);
R = 1;
k = lqr(Au, B, Q, R);
[t,y] = ode45(@(t,y)pend_function(y,m,L,d,g, -k*(y - fixed)),tspan,y0);
```

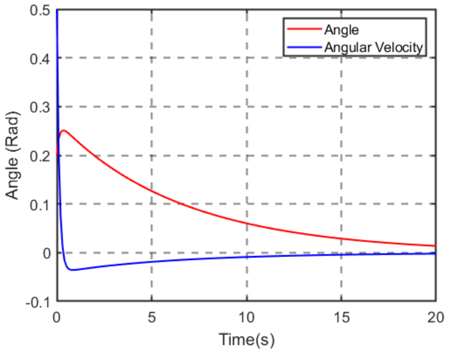
$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$





$R = 1$ (less control penalty)





$R = 2$ (more control penalty)

LQR Gist

- To control any nonlinear dynamical system with LQR:
    - Identify the fixed points.
    - Identify the dimensions of Q and R matrices and construct them with desired weights.
    - Linearize the nonlinear system around the fixed points.
    - Identify A and B matrices by evaluating the system around the fixed points.
    - Obtain the gain matrix K by using the `LQR(A, B, Q, R)` function.
    - Simulate the system using the control law $u = -Kx$.

- In trajectory tracking, use the waypoints as the fixed points. After reaching one waypoint, use the next one as the fixed point and repeat the procedure.