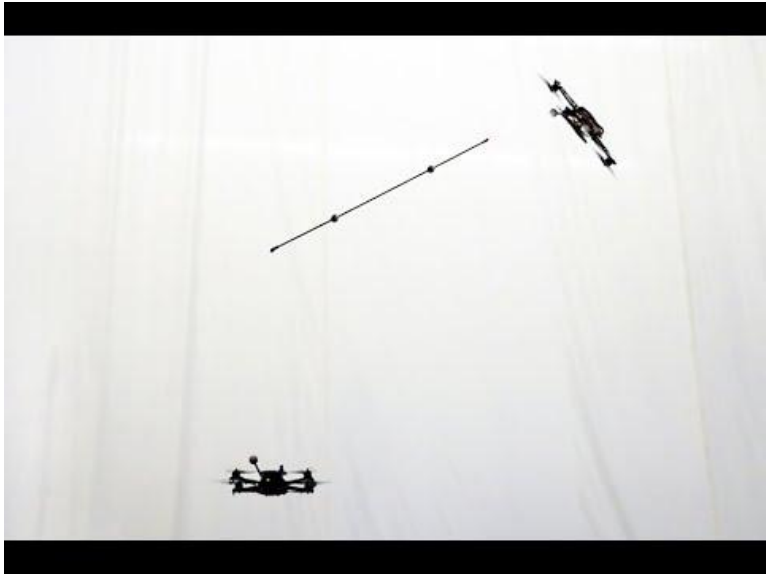
Swing up and balancing of an inverted pendulum on a 2-D quadrotor

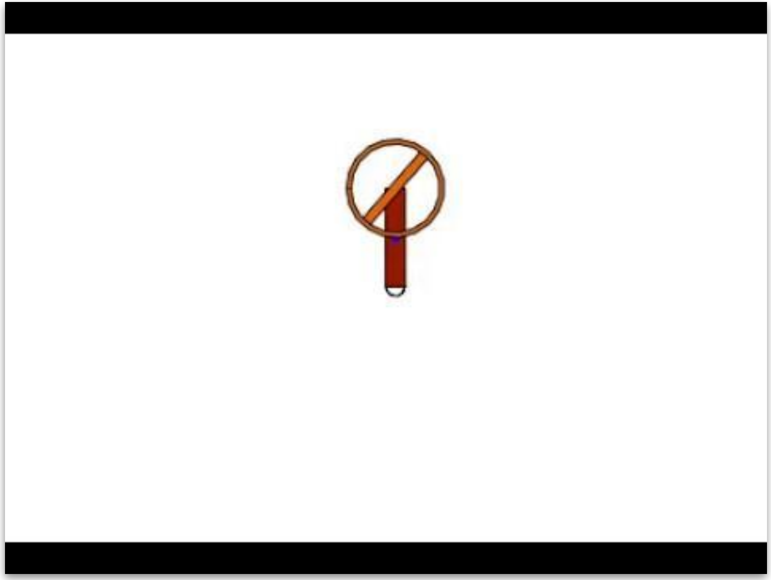
Gilhyun Ryou, ghryou@mit.edu
Seong Ho Yeon, syeon@mit.edu

Inspiration



Quadcopter Pole Acrobatics
(ETH - Raffaello D'Andrea Group)

+



PSet3 : Inertial Pendulum Swing-Up

Can we swing up and stabilize a single/double pendulum on drone?

System Modeling of 2-D quadrotor

Mb = 1 kg
lb = 0.2 m

M1 = 1 kg
l1 = 0.2 m

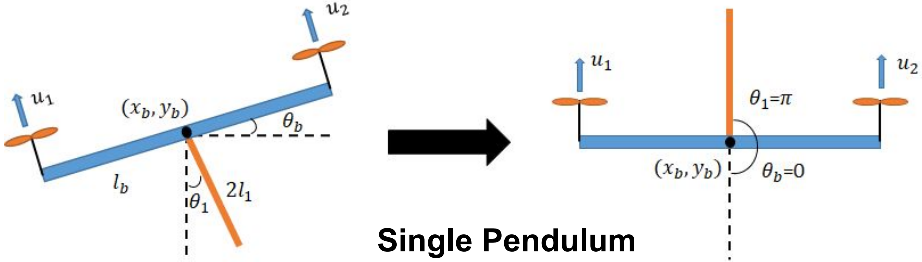
M2 = 1kg
l2 = 0.2m

g = 10 m/s^2

No hysteresis on thrust
Thrust is bidirectional

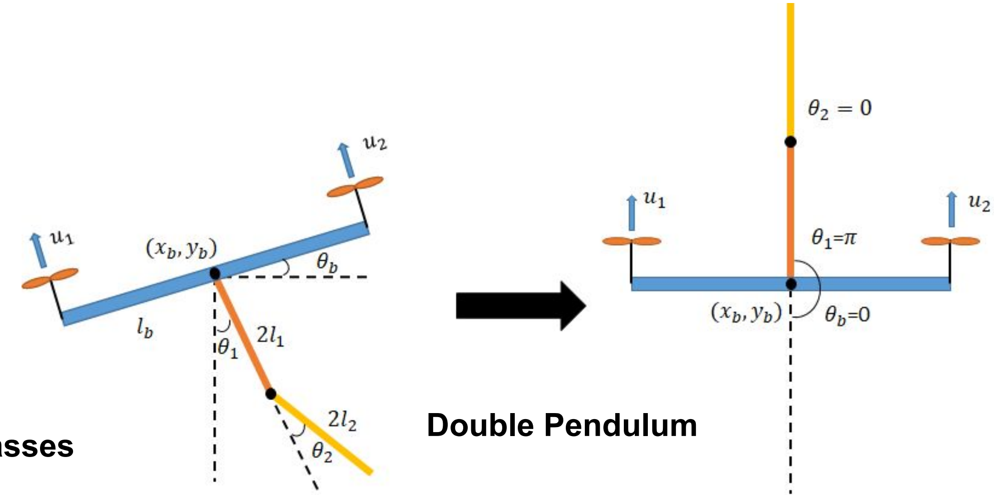
Body is vertically symmetric
(Allowed body flipping)

No friction/contact between masses



Single Pendulum

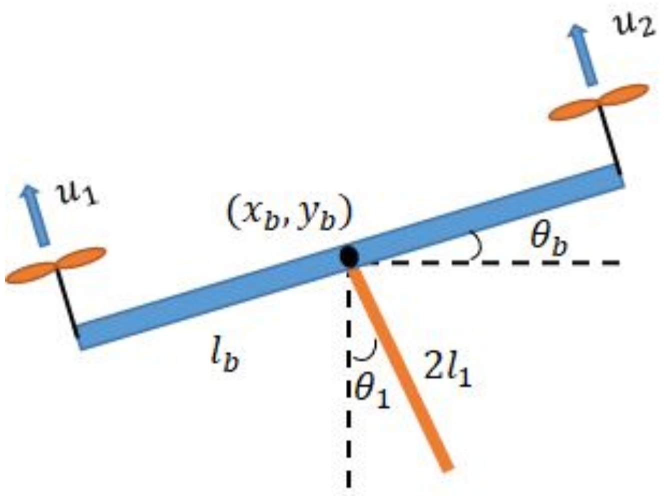
u_max = 20 N
(1 x total weight)



Double Pendulum

u_max = 45 N
(1.5 x total weight)

Single Pendulum Dynamics Analysis



$$\mathcal{L} = T - V$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = F_i$$

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q, \dot{q}) = \mathcal{T}_g(q) + \mathcal{B}(q)u$$

$$T = \frac{1}{2}m_b(\dot{x_b}^2 + \dot{y_b}^2) + \frac{1}{2}m_1(\dot{x_1}^2 + \dot{y_1}^2) + \frac{1}{2}I_b\dot{\theta_b}^2 + \frac{1}{2}I_1\dot{\theta_1}^2$$

$$V = m_b g y_b + m_1 g y_1 + m_2 g y_2$$

$$x_1 = x_b + l_1 \sin\theta_1, \quad y_1 = x_b - l_1 \cos\theta_1$$

$$I_b = \frac{1}{3}m_b l_b^2, \quad I_1 = \frac{1}{3}m_1 l_1^2$$

$$q = [x_b, y_b, \theta_b, \theta_1]^T$$

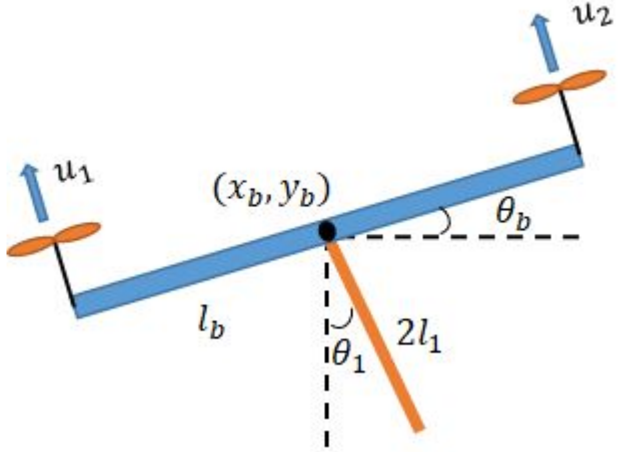
Single Pendulum Dynamics Analysis

$$\mathcal{M}(q) = \begin{bmatrix} m_b + m_1 & 0 & 0 & m_1 l_1 \cos\theta_1 \\ 0 & m_b + m_1 & 0 & m_1 l_1 \sin\theta_1 \\ 0 & 0 & I_b & 0 \\ m_1 l_1 \cos\theta_1 & m_1 l_1 \sin\theta_1 & 0 & I_1 + m_1 l_1^2 \end{bmatrix}$$

$$\mathcal{C}(q, \dot{q}) = \begin{bmatrix} -m_1 l_1 \sin\theta_1 \dot{\theta_1}^2 & m_1 l_1 \cos\theta_1 \dot{\theta_1}^2 & 0 & 0 \end{bmatrix}^T$$

$$\mathcal{T}_g(q) = \begin{bmatrix} 0 & -(m_b + m_1)g & 0 & -m_1 l_1 g \sin\theta_1 \end{bmatrix}^T$$

$$\mathcal{B}(q) = \begin{bmatrix} -\sin\theta_b & -\sin\theta_b \\ \cos\theta_b & \cos\theta_b \\ -l_b & l_b \\ 0 & 0 \end{bmatrix}$$



Double Pendulum Dynamics Analysis

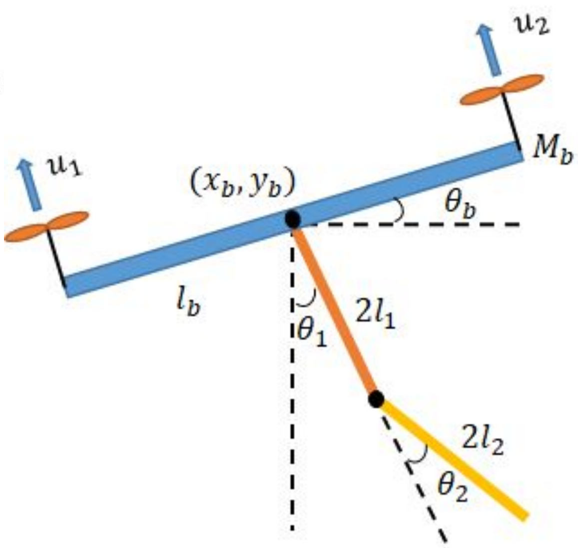
$$T = \frac{1}{2}m_b(\dot{x_b}^2 + \dot{y_b}^2) + \frac{1}{2}m_1(\dot{x_1}^2 + \dot{y_1}^2) + \frac{1}{2}m_2(\dot{x_2}^2 + \dot{y_2}^2) + \frac{1}{2}I_b\dot{\theta_b}^2 + \frac{1}{2}I_1\dot{\theta_1}^2 + \frac{1}{2}I_2\dot{\theta_2}^2$$

$$V = m_b g y_b + m_1 g y_1 + m_2 g y_2$$

$$x_1 = x_b + l_1 \sin\theta_1, \quad y_1 = x_b - l_1 \cos\theta_1$$

$$x_2 = x_b + 2l_1 \sin\theta_1 + l_2 \sin\theta_2, \quad y_2 = x_b - 2l_1 \cos\theta_1 - l_2 \cos\theta_2$$

$$q = [x_b, y_b, \theta_b, \theta_1, \theta_2]^T$$



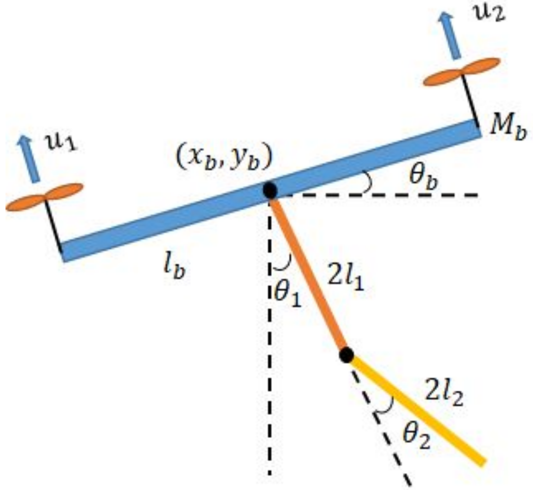
Double Pendulum Dynamics Analysis

$$\mathcal{M}(q) = \begin{bmatrix} m_b + m_1 + m_2 & 0 & 0 & (m_1 + 2m_2) l_1 \cos\theta_1 & m_2 l_2 \cos\theta_2 \\ 0 & m_b + m_1 + m_2 & 0 & (m_1 + 2m_2) l_1 \sin\theta_1 & m_2 l_2 \sin\theta_2 \\ 0 & 0 & I_b & 0 & 0 \\ (m_1 + 2m_2) l_1 \cos\theta_1 & (m_1 + 2m_2) l_1 \sin\theta_1 & 0 & I_1 + m_1 l_1^2 + 4 m_2 l_1^2 & 2 m_2 l_1 l_2 \cos(\theta_1 - \theta_2) \\ m_2 l_2 \cos\theta_2 & m_2 l_2 \sin\theta_2 & 0 & 2 m_2 l_1 l_2 \cos(\theta_1 - \theta_2) & I_2 + m_2 l_2^2 \end{bmatrix}$$

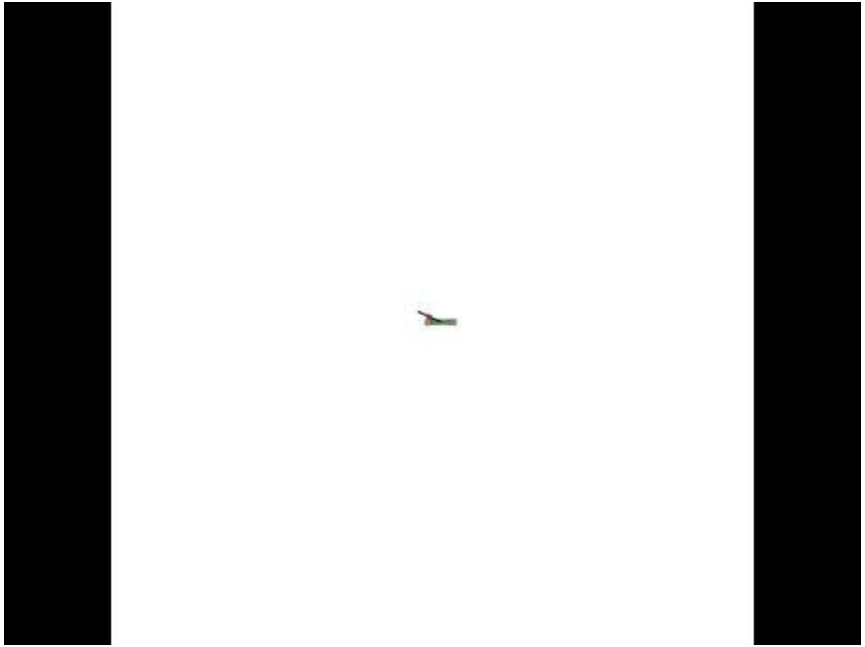
$$\mathcal{C}(q, \dot{q}) = \begin{bmatrix} -(m_1 + 2m_2) l_1 \sin\theta_1 \dot{\theta_1}^2 - m_2 l_2 \sin\theta_2 \dot{\theta_2}^2 \\ (m_1 + 2m_2) l_1 \cos\theta_1 \dot{\theta_1}^2 + m_2 l_2 \cos\theta_2 \dot{\theta_2}^2 \\ 0 \\ 2 m_2 l_1 l_2 \sin(\theta_1 - \theta_2) \dot{\theta_2}^2 \\ -2 m_2 l_1 l_2 \sin(\theta_1 - \theta_2) \dot{\theta_1}^2 \end{bmatrix}$$

$$\mathcal{T}_g(q) = \begin{bmatrix} 0 & -(m_b + m_1 + m_2) g & 0 & -(m_1 + 2m_2) g l_1 \sin\theta_1 & -m_2 g l_2 \sin\theta_2 \end{bmatrix}^T$$

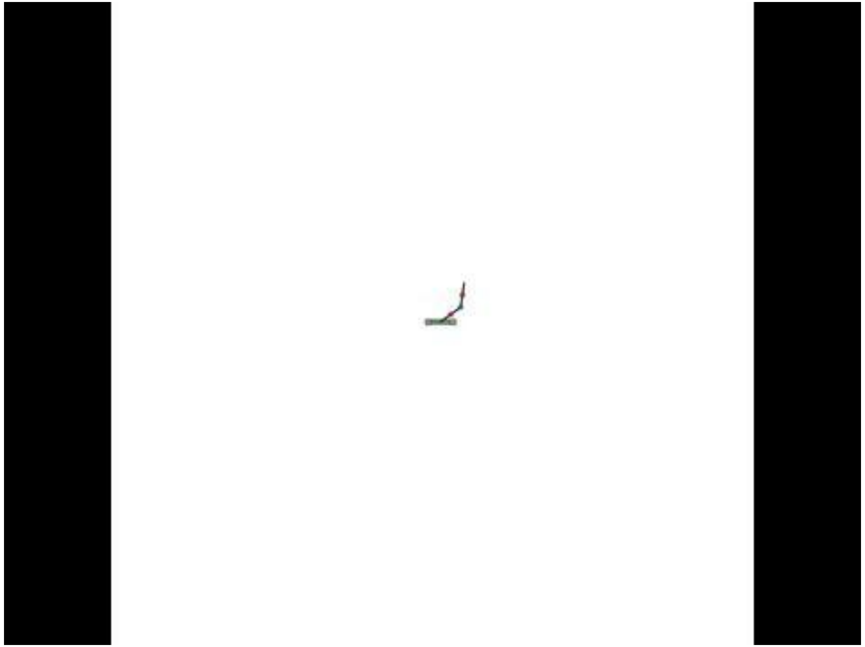
$$\mathcal{B}(q) = \begin{bmatrix} -\sin\theta_b & -\sin\theta_b \\ \cos\theta_b & \cos\theta_b \\ -l_b & l_b \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$



Simulator Construction and Dynamics Verification



Single Pendulum
Constant Input u1, u2 (= drone weight)



Double Pendulum
Constant Input u1, u2 (= drone weight)

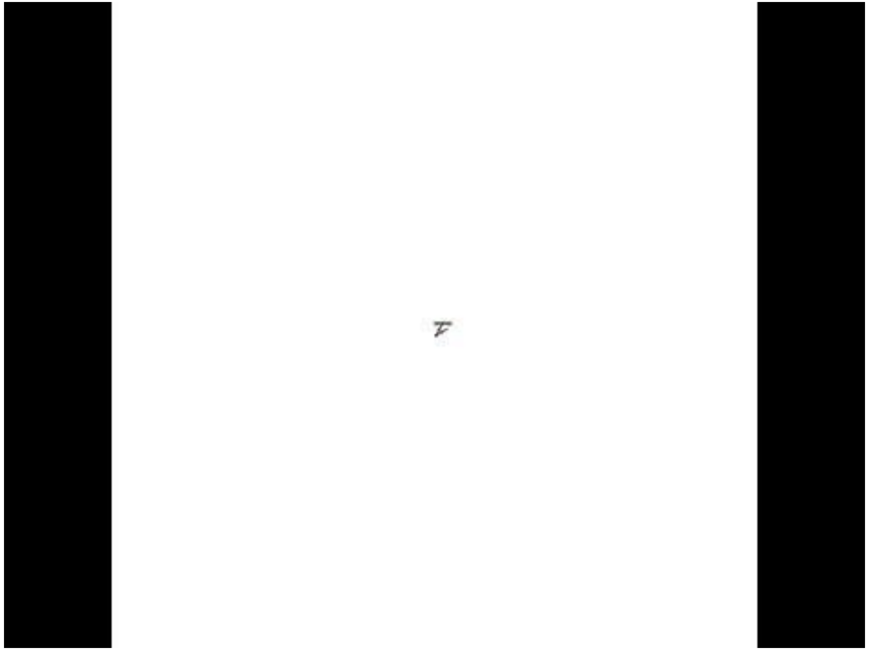
Simulator is constructed based on the dynamics on PyDrake environment (same framework as PSET 3,4)

Control frequency = 200Hz, Simulation frequency = 1kHz

Simulator Construction and Dynamics Verification

Single Pendulum

Dynamics verification with Demonstration of failed controller



Double Pendulum

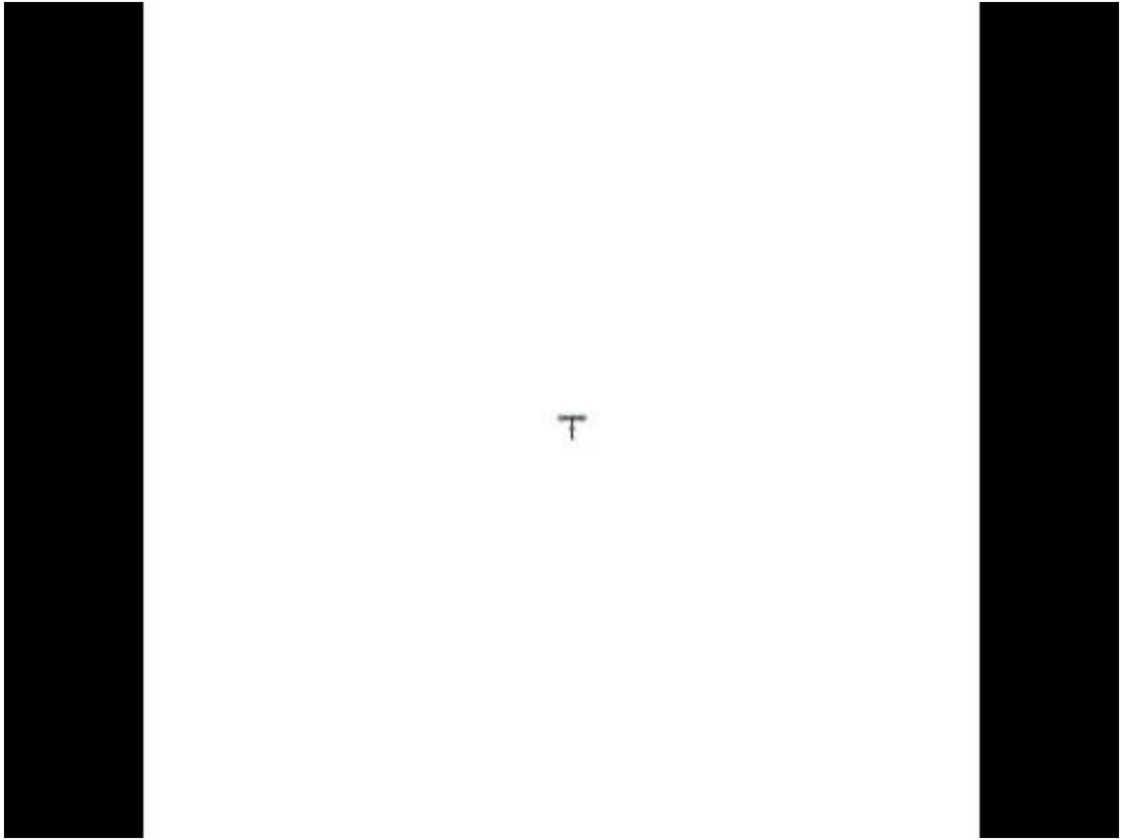
Dynamics verification with Demonstration of failed controller

Result Summary : Single Pendulum

Swing-Up trajectory controller based on dynamics intuition

+

Stabilization with LQR

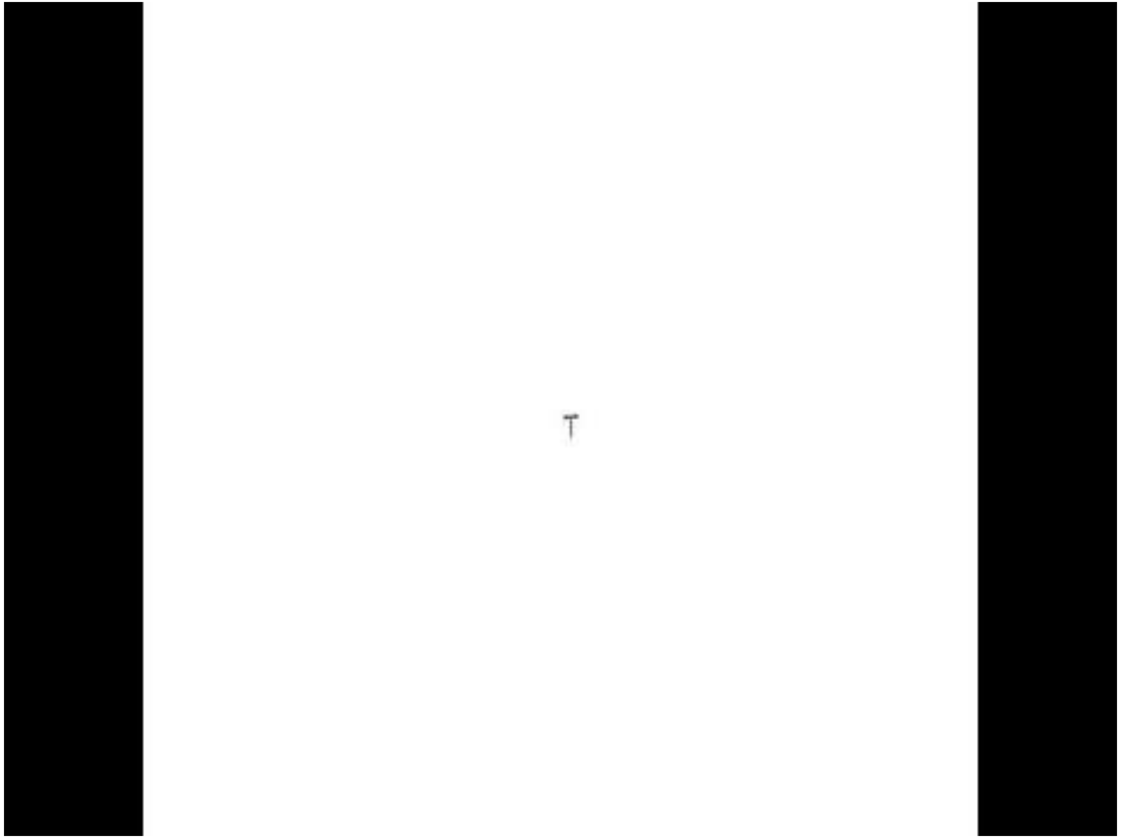


Result Summary : Double Pendulum

Swing-Up trajectory controller based on dynamics intuition

+

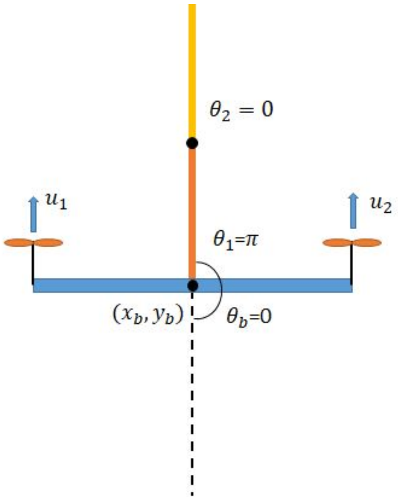
Stabilization with LQR



LQR formulation

General Linearization of Dynamics around any (x, u) numerically using MATLAB

Solve Riccati equation on PyDrake



$$\bar{q} = q - q_f, \quad \dot{\bar{q}} = \dot{q} - \dot{q}_f, \quad \ddot{\bar{q}} = \ddot{q} - \ddot{q}_f, \quad \bar{u} = u - u_f$$

$$x = [\bar{q} \;\; \dot{\bar{q}}]^T$$

$$\ddot{q} = \mathcal{M}(q)^{-1}\big(- \mathcal{C}(q, \dot{q}) + \mathcal{T}_g(q) + \mathcal{B}(q)u \big)$$

$$\ddot{\bar{q}} \approx \Psi_0 \bar{q} + \Psi_1 \dot{\bar{q}} + \Psi_2 \bar{u}$$

$$\dot{x} \approx \begin{bmatrix} 0 & I \\ \Psi_0 & \Psi_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ \Psi_3 \end{bmatrix} \bar{u}$$

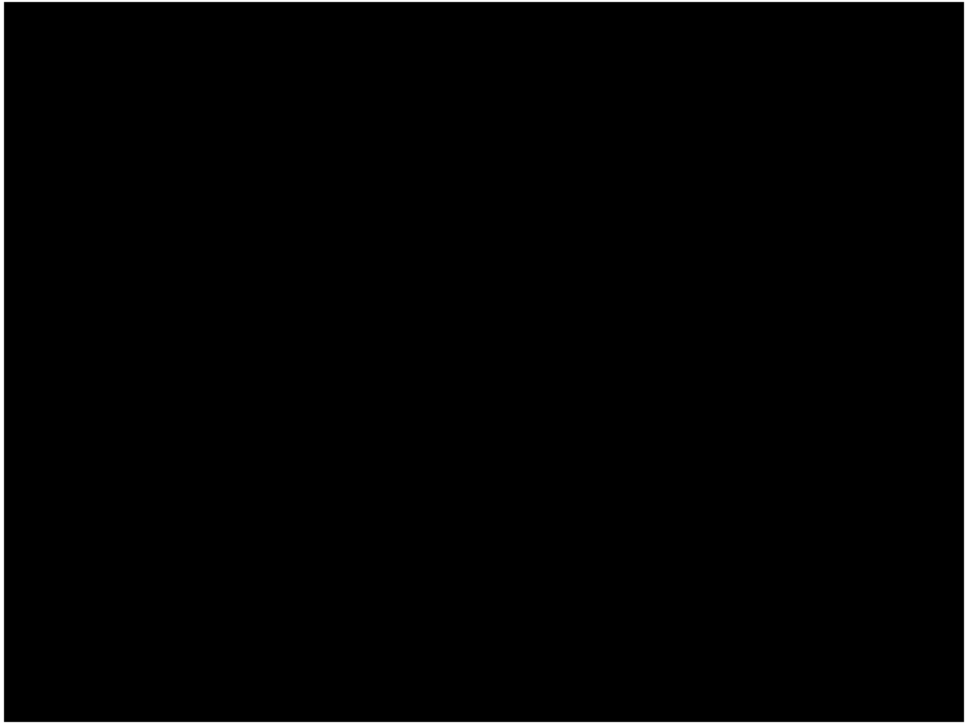
$$\dot{x} \approx Ax + B\bar{u}$$

$$\dot{x} = Ax + B\bar{u}$$

$$u = u_f - Kx$$

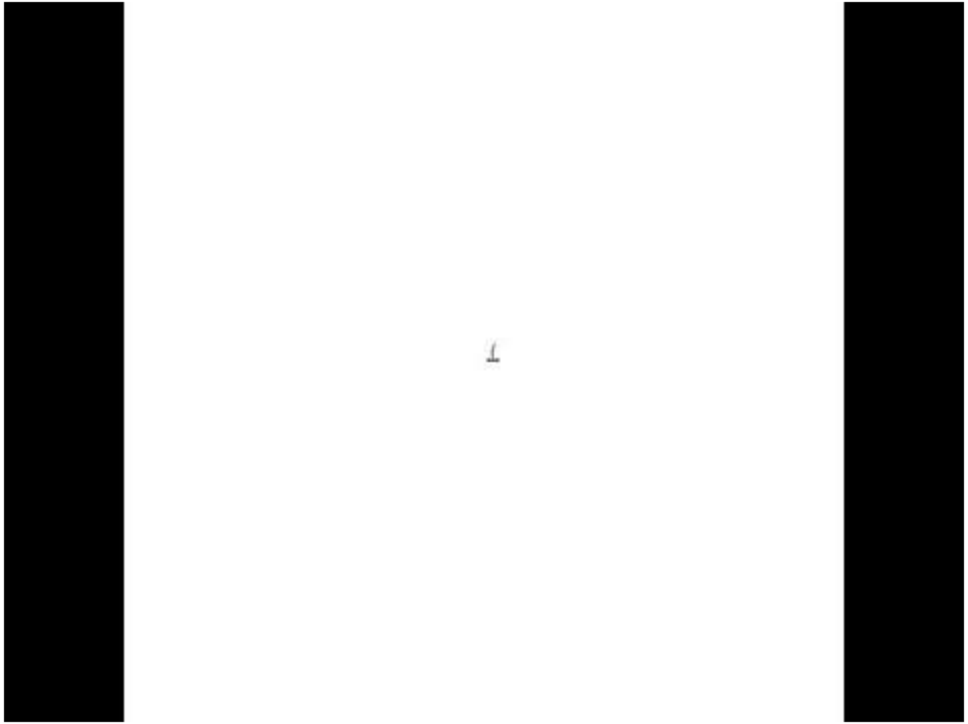
Single Pendulum LQR result 1 - Stabilization

Stabilization with LQR



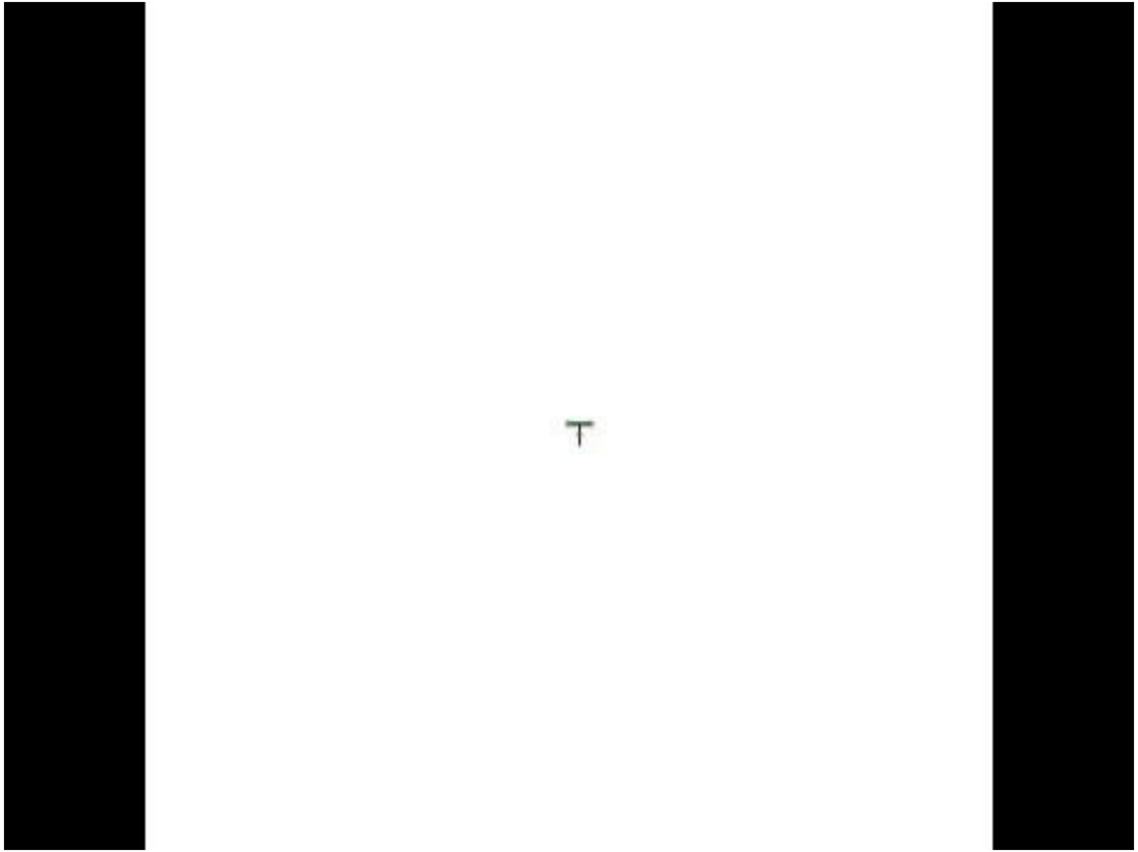
Double Pendulum LQR result 1 - Stabilization

Stabilization with LQR



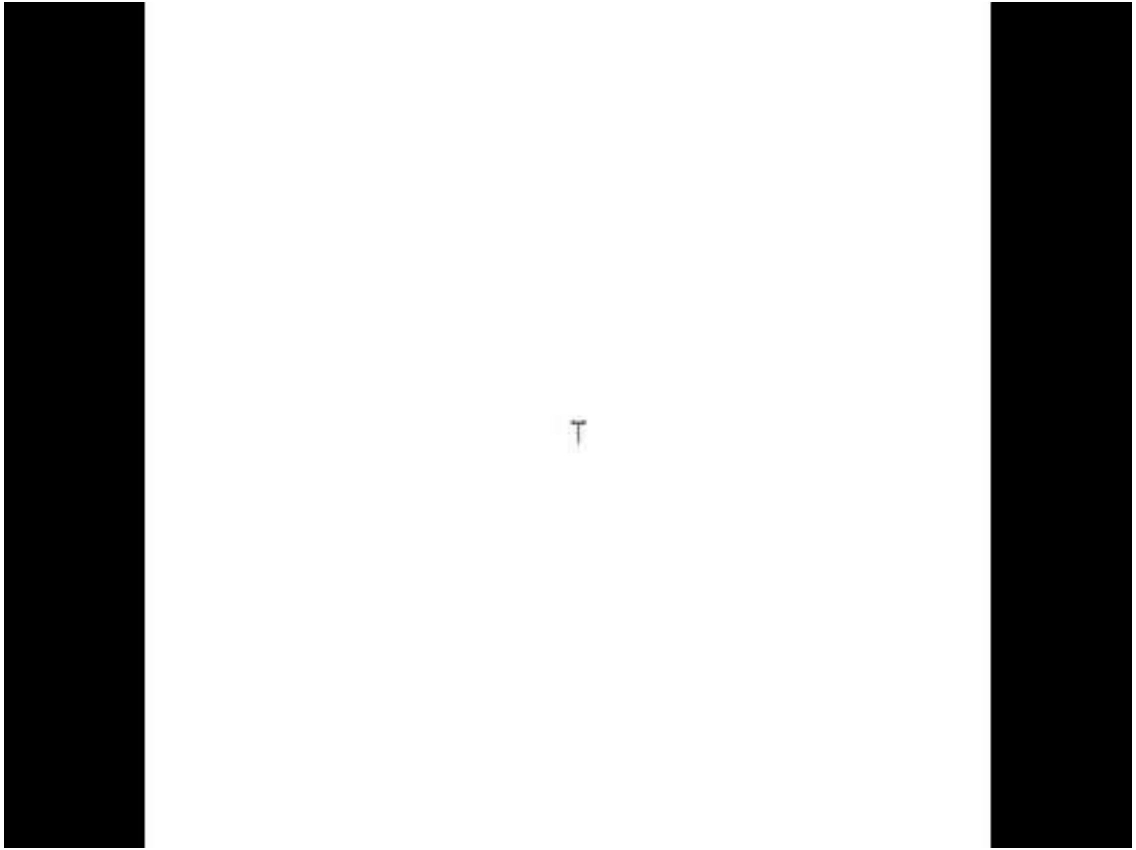
Single Pendulum (unexpected) LQR result 2

Swing-up with LQR ???

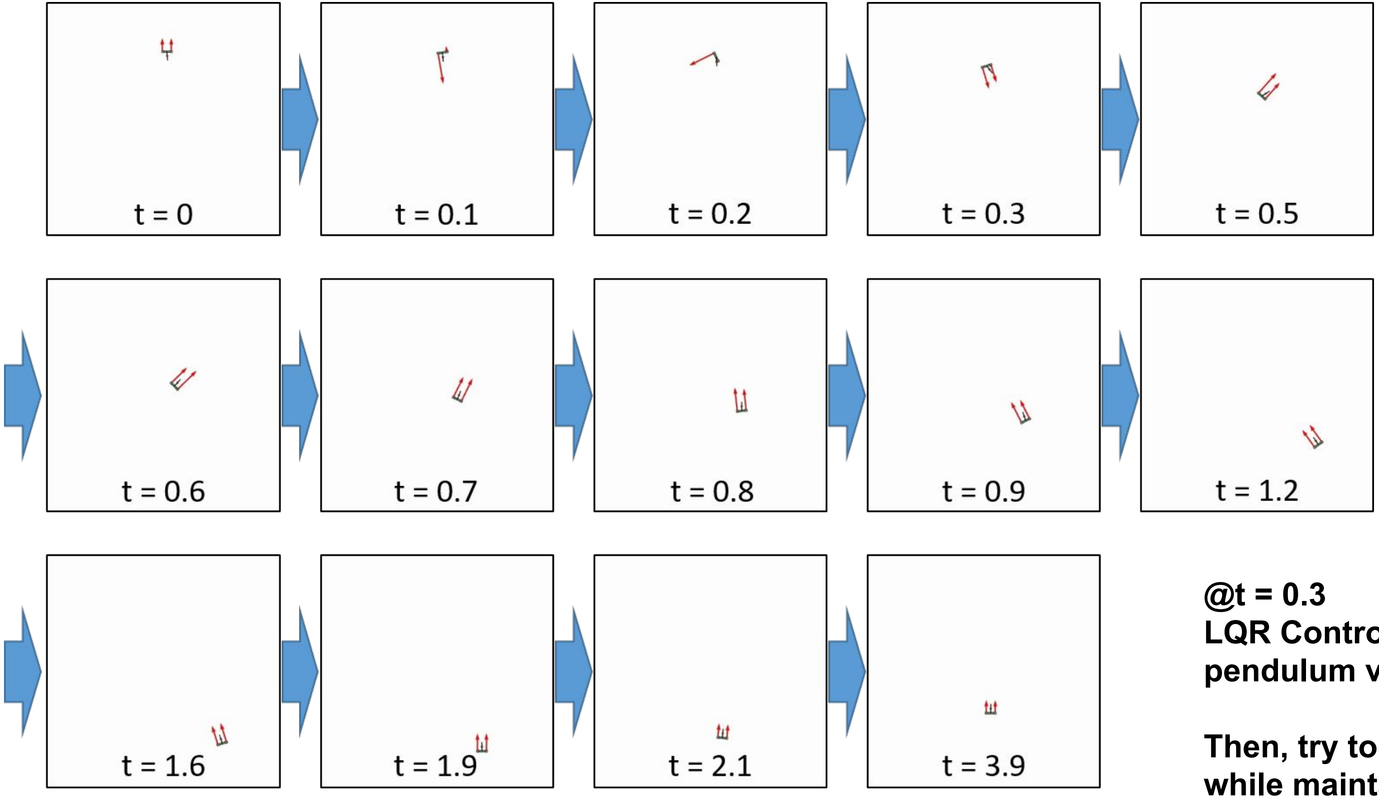


Double Pendulum (unexpected) LQR result 2

Swing-up with LQR ???



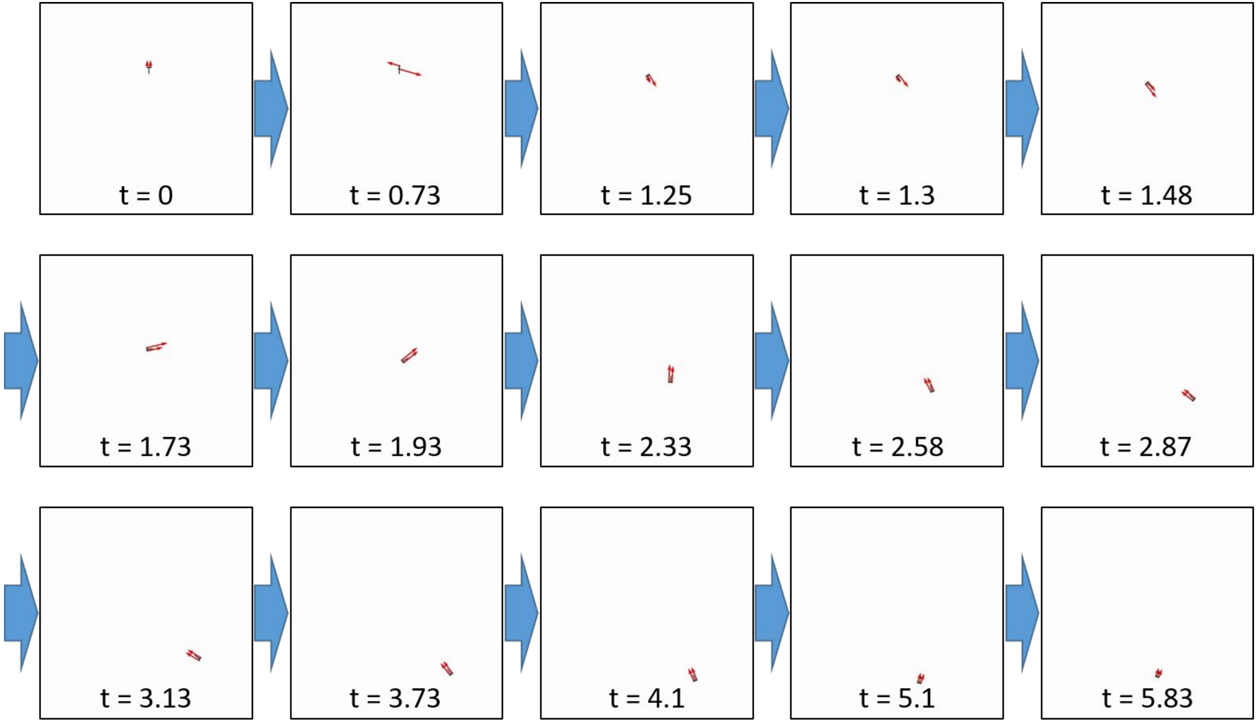
Intuition from Swing-up via LQR



@t = 0.3
LQR Controller tries to set pendulum vertical toward body

Then, try to stabilize body while maintaining pendulum vertical

Intuition from Swing-up via LQR (double)

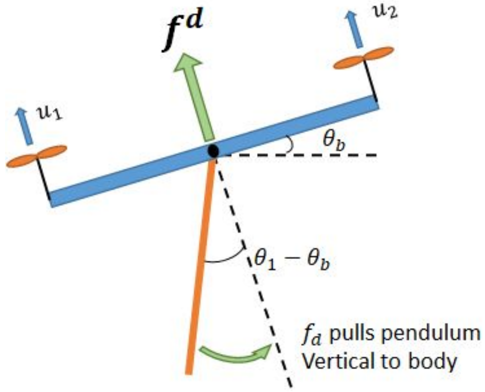


@t = 1.25
LQR Controller tries to set pendulum vertical toward body

Then, try to stabilize body while maintaining pendulum vertical

Swing-up Formulation

$\theta_b - \theta_1 \rightarrow 0$



$\theta_1 \rightarrow \theta_{1f}$

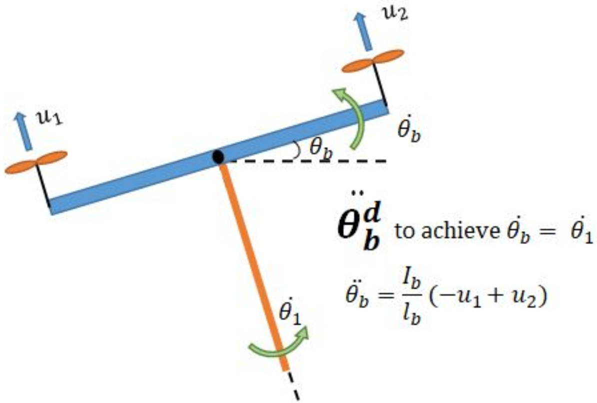
$\dot{\theta}_1 \rightarrow 0$

$$f^d = \left(\ddot{x}_b^d \hat{x} + \ddot{y}_b^d \hat{y}\right) = -K_1(\theta_1 - \theta_{1f}) - K_2\dot{\theta}_1$$

$$f^d \cos\theta_b = \ddot{y}_b = (u_1 + u_2)\cos\theta_b - g$$

$$u_1 + u_2 = f^d + \frac{g}{\cos\theta_b}$$

$\dot{\theta}_b - \dot{\theta}_1 \rightarrow 0$



$$\ddot{\theta}_b^d = -K_3(\dot{\theta}_b - \dot{\theta}_1)$$

$$\ddot{\theta}_b^d = \ddot{\theta}_b = \frac{l_b}{I_b}(-u_1 + u_2)$$

$$-u_1 + u_2 = \frac{I_b}{l_b}\ddot{\theta}_b^d$$

$$u_1 = \frac{1}{2}\big(- K_1(\theta_1 - \theta_{1f}) - K_2\dot{\theta}_1 + \frac{g}{\cos\theta_b} + K_3\frac{I_b}{l_b}(\dot{\theta}_b - \dot{\theta}_1)\big)$$

$$u_2 = \frac{1}{2}\big(- K_1(\theta_1 - \theta_{1f}) - K_2\dot{\theta}_1 + \frac{g}{\cos\theta_b} - K_3\frac{I_b}{l_b}(\dot{\theta}_b - \dot{\theta}_1)\big)$$

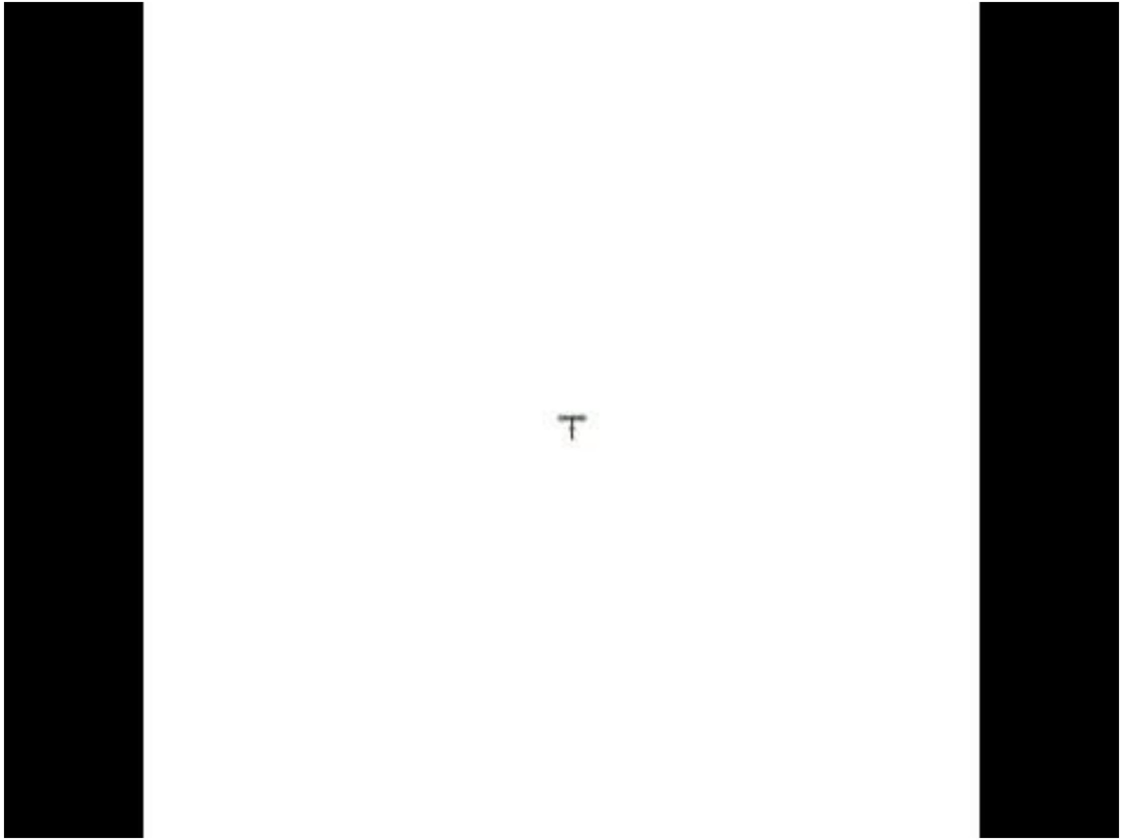
Switching from Swing-up → LQR
@ x with minimum J = x'Sx

Result Summary : Single Pendulum

Swing-Up trajectory controller based on dynamics intuition

+

Stabilization with LQR

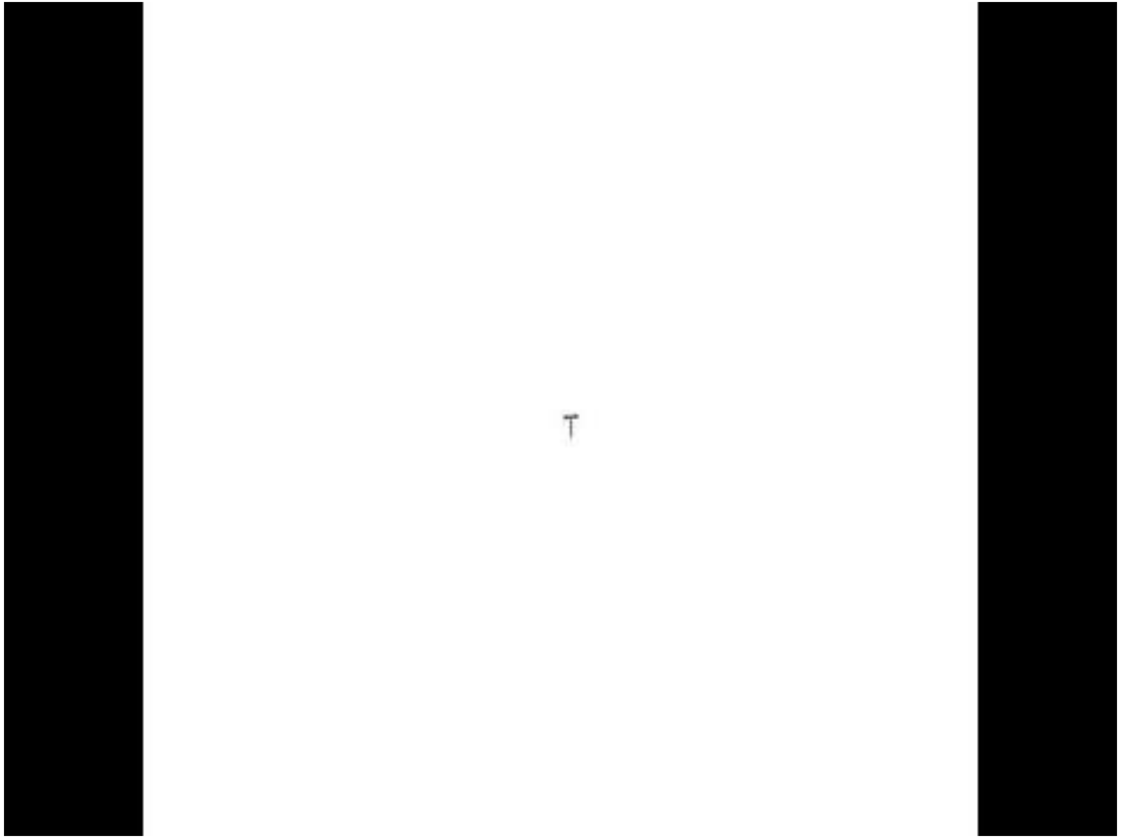


Result Summary : Double Pendulum

Swing-Up trajectory controller based on dynamics intuition

+

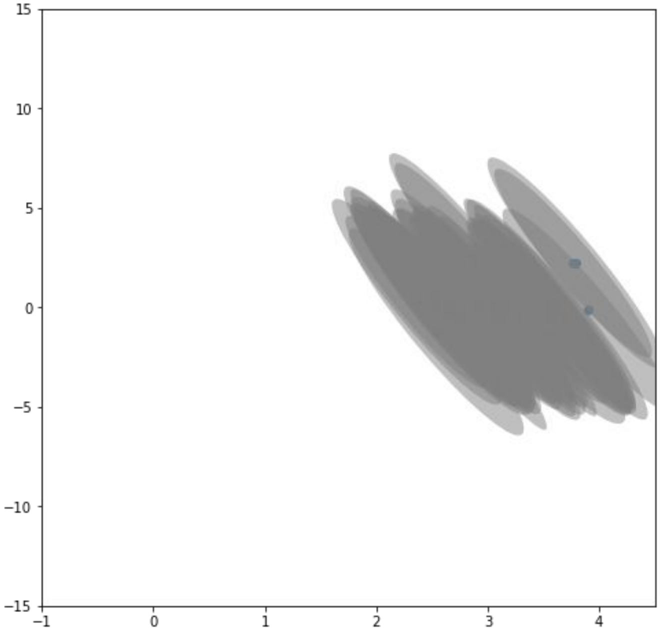
Stabilization with LQR



Other approaches and (our) failure

iLQR, LQR-Trees, …

- Large dimension of state space
- Discontinuity during motion planning ($0 = 2\pi$)
- Complicate dynamics
- Numerical issue on non-linear optimization



Questions?