# 16-745 Optimal Control Lecture 2

Reid Graves

January 21, 2025

## 1 Last time

- Continuous Time Dynamics
- Manipulator Dynamics
- Linear Systems

Won't need to derive these for this class.

## 2 Today

- Equilibria
- Stability
- Discrete Time Dynamics and Simulation

## 3 Equilibria

- A point where the system will "remain at rest"

$$\dot{x} = f(x, u) = 0$$

- Algebraically, roots of the dynamics
- Look at Pendulum again

$$\begin{aligned} \dot{x} &= \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin(\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \dot{\theta} &= 0 \\ \theta &= 0, \pi \end{aligned}$$

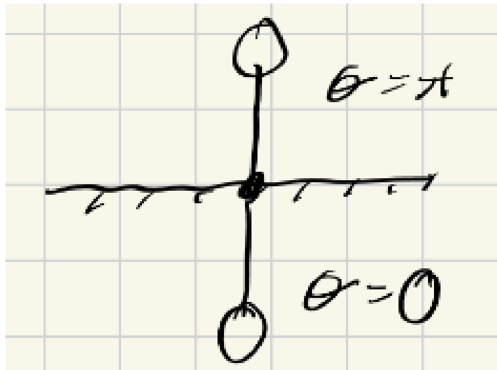


Figure 1: Pendulum

### 3.1 First control Problem

- Can I move the equilibria?

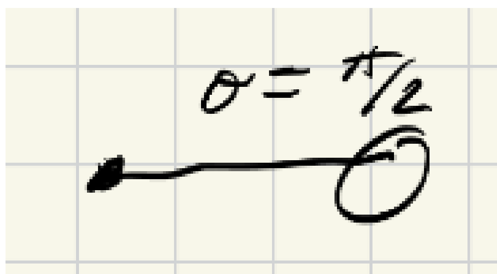


Figure 2: Moving pendulum equilibria

$$\begin{aligned} \theta &= \frac{\pi}{2} \\ \frac{1}{ml^2}u &= \frac{q}{l}\sin\left(\frac{\pi}{2}\right) \\ u &= mgl \end{aligned} \qquad \dot{x} = \begin{bmatrix} \dot{\theta} \\ \frac{-q}{l}\sin\left(\frac{\pi}{2}\right) + \frac{1}{ml^2}u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- In general, we get a root finding problem in u:

$$f(x^*, u) = 0$$

# 4 Stability of Equilibria

- When will we stay ”near” an equilibrium point under perturbations?
- Look at a 1D system (1 dimensional state space) $x \in \mathbb{R}$

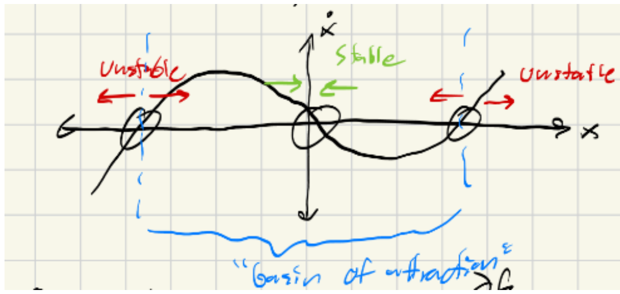


Figure 3: Stability at equilibria graphic

- If $\frac{df}{dx} < 0$, stable
- If $\frac{df}{dx} > 0$, unstable
- Basin of attraction: area between the two unstable points
- In higher dimensions: $\frac{\partial f}{\partial x}$ is a Jacobian Matrix
- Take an Eigendecomposition $\rightarrow$ Decouple into $n$ 1D systems

$$Re\left[eigvals\left(\frac{\partial f}{\partial x}\right)\right] < 0 \rightarrow \text{stable}$$

  Otherwise $\rightarrow$ Unstable

- Pendulum:

$$f(x) = \begin{bmatrix} \dot{\theta} \\ \frac{-q}{l}\sin(\theta) \end{bmatrix} \rightarrow \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ \frac{-q}{l}\cos(\theta) & 0 \end{bmatrix}$$

$$\frac{\partial f}{\partial x}|_{\theta=\pi} = \begin{bmatrix} 0 & 1 \\ \frac{q}{l} & 0 \end{bmatrix}$$

$$eigvals = \pm\sqrt{\frac{q}{l}} \rightarrow \text{unstable}$$

$$\frac{\partial f}{\partial x}_{\theta=0} = \begin{bmatrix} 0 & 1 \\ \frac{-q}{l} & 0 \end{bmatrix} \rightarrow \text{eigvals} = \pm i\sqrt{\frac{q}{l}}$$

$\rightarrow$ Undamped oscillation

- Add damping (e.g. $u = -kd\dot{\theta}$) results in strictly negative real part

## 5 Discrete Time Dynamics

- Motivation:
  - In general, we can't solve $\dot{x} = f(x)$ for $x(t)$
  - Computationally, need to represent $x(t)$ with a set of discrete $x_k$
  - Discrete-time models can capture some effects that continuous ODEs can't (like a brick hitting the ground)

- Explicit Form:

$$x_{k+1} = f_d(x_k, u_k)$$

- Simplest discretization:

$$x_{k+1} = x_k + hf(x_k, u_k)$$

  $h$ is time step, whole right hand side is $f_d(x_k, u_k)$. Whole thing is called **Forward Euler Integration**

- Forward Euler integration adds energy- If you have an oscillatory system, this method always adds energy, causing the system to explode

- Pendulum Sim:

$$l = m = 1$$

### 5.1 Stability of discrete time systems

- In discrete time, dynamics is an iterated map:

$$x_n = f_d(f_d(f_d(f_d \dots f_d(x_0))))$$

- Linearize and apply chain rule:

$$\frac{\partial x_k}{\mathrm{x}_0} = \frac{\partial f_d}{x}|_{x_0} \frac{\partial f_d}{x}|_{x_0} \dots \frac{\partial f_d}{x}|_{x_0} = A_d^N$$

- Assume $x_0 = 0$ is an equilibrium

$$\begin{gathered}
\text{stable} \rightarrow \lim_{k \to \infty} A_d^k x_0 = 0 \quad \forall x_0 \\
\rightarrow \lim_{k \to \infty} A_d^k = 0 \\
\rightarrow |eigvals(A_d)| < 1 \text{ (Inside unit circle)}
\end{gathered}$$

- Pendulum with Forward Euler:

$$x_{k+1} = x_k + hf(x_k)$$

  Right side is $f_d(x_k)$

$$Ad = \frac{\partial f_d}{\partial x} \qquad = I + HA = I + h \begin{bmatrix} 0 & 1 \\ \frac{-q}{l}\cos(\theta) & 0 \end{bmatrix}$$

$$eigvals(A_d|_{\theta \approx 0})$$

- Key takeaway: **Never Use Forward Euler**

- Intuition:

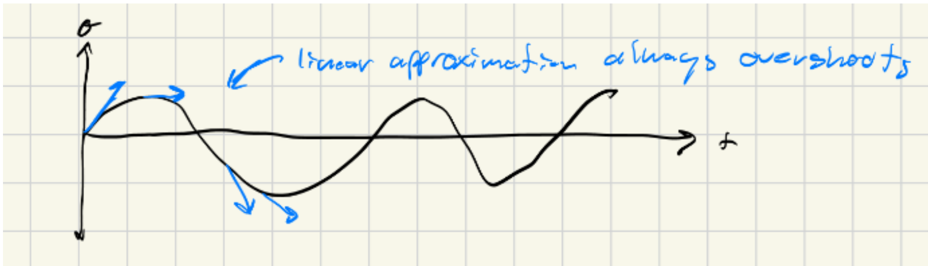


Figure 4: Linear approximator always overshoots

Linear Approximation always overshoots

- Take Aways:
  - Be careful
  - Always sanity check e.g. energy, momentum behavior
  - Never use forward Euler!
- A better explicit integrator:
  - 4th order Runge-Kutta Method
  - RK4 fits a cubic polynomial to $x(t)$ rather than a line- Much better accuracy!
  - Pseudo Code:

$$\begin{aligned} x_{k+1} &= f_d(x_k) \\ k_1 &= f(x_k) \\ k_2 &= f(x_k + \frac{h}{2}k_1) \\ k_3 &= f(x_k + \frac{h}{2}k_2) \\ k_4 &= f(x_k + hk_3 \\ x_{x+1} &= x_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

- take away:
  - Accuracy >> additional compute cost
  - Even "good" integrators have issues
  - Always sanity check