Robotic Systems II

Lecture 1: Dynamics, Discretization and Integrators

Konstantinos Chatzilygeroudis - costashatz@upatras.gr

Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

Logistics

- **Lectures**:
  - 10 Lectures
  - 1 Recitation Lecture
  - 1-2 Seminars
  - 10 Lab Exercises
- **Examination**:
  - 4 Homeworks **(40% of total grade)**
  - Oral Exam **(60% of total grade)**
- **Office Hours**:
  - **Tue & Thu** *(09:00-11:00)*
  - 24/7 by email (`costashatz@upatras.gr`, subject: *ECE_RSII_AM*)
- **Material and Announcements**

Laboratory of Automation & Robotics

What is this course about?

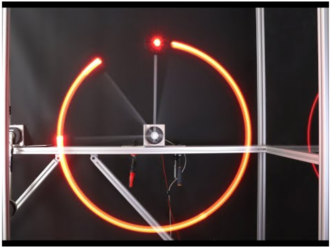
We could name the course "Applied Optimal Control":

- Optimization
- Control of Linear Systems (LQR)
- Model-Predictive Control
- Linearization of Non-Linear Systems
- Trajectory Optimization (Direct Collocation)
- Optimal Control on Manifolds
- Differential Dynamic Programming
- Locomotion, Contacts, Complex Robots
- **We focus on implementation/robotics applications. You are required to write code!** Not a theory course!

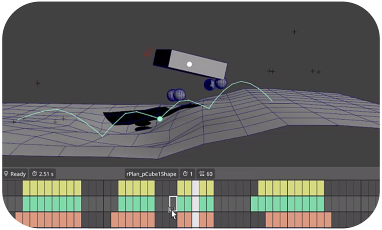
Lectures/Lab Format

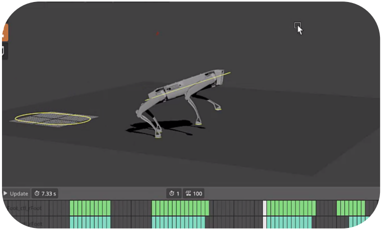
- **3-hour lectures:**
  - A lot of theory! Hints for personal reading!
  - Live code examples + analyses!
  - Ask questions please!
- **2-hour lab exercises:**
  - Do not count for final grade! **You need to participate in 60%**!
  - Small code examples that you need to fill
  - You need to deliver code even if it doesn't run
  - Ask questions, experiment!

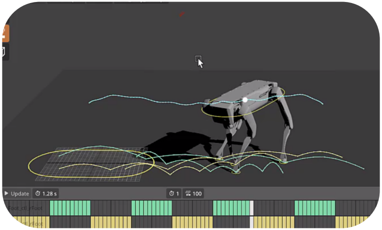
Why study optimal control?

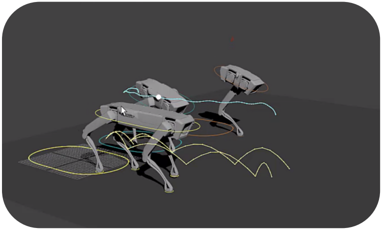


What will you be able to achieve?









Credits: Ragdoll Dynamics (Imbalance Ltd)

Quick Dynamics Review

Continuous Dynamics:

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

$\boldsymbol{x} \in \mathbb{R}^n$: "State space"

$\boldsymbol{u} \in \mathbb{R}^m$: "Control/Input space"

f : "Dynamics Function"

In robotics we usually have:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{v} \end{bmatrix}$$

$\boldsymbol{q}$: "Configuration"

$\boldsymbol{v}$: "Velocity"

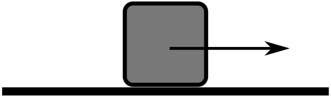
Double Integrator:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}$$

$$\boldsymbol{u} = K\ddot{\boldsymbol{q}}$$

where

$$\boldsymbol{q} = \begin{bmatrix} x \end{bmatrix} \in \mathbb{R}$$
$$\dot{\boldsymbol{q}} = \begin{bmatrix} v \end{bmatrix} = \begin{bmatrix} \dot{x} \end{bmatrix} \in \mathbb{R}$$
$$\boldsymbol{u} = \begin{bmatrix} K\ddot{x} \end{bmatrix} \in \mathbb{R}$$



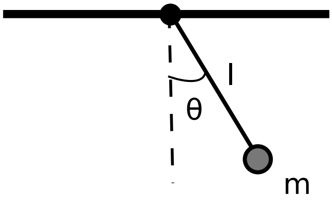
Dynamics:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \frac{\boldsymbol{u}}{K} \end{bmatrix}$$

Pendulum:

$$\boldsymbol{x} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$

$$\boldsymbol{u} = \text{“actuator torque”} \in \mathbb{R}$$



Dynamics:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix}$$

$$= \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta + \frac{1}{ml^2}\boldsymbol{u} - b\dot{\theta} \end{bmatrix}$$

where g is the gravity, m the point mass, b is a damping coefficient, and l the length of the pole.

Control Affine Systems:

$$\dot{\boldsymbol{x}} = \underbrace{f_o(\boldsymbol{x})}_{\text{“drift”}} + \underbrace{\boldsymbol{B}(\boldsymbol{x})}_{\text{“input Jacobian”}} \boldsymbol{u}$$

Many systems can be written like this!

- **Double Integrator:**

$$f_o(\boldsymbol{x}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ 0 \end{bmatrix}, \boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ \frac{1}{K} \end{bmatrix}$$

- **Pendulum:**

$$f_o(\boldsymbol{x}) = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix}, \boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$$

Manipulator Equation Reminder:

$$\underbrace{\boldsymbol{M}(\boldsymbol{q})}_{\text{“Mass Matrix”}}\dot{\boldsymbol{v}} + \underbrace{\boldsymbol{C}(\boldsymbol{q},\boldsymbol{v})}_{\text{“Coriolis/Gravity Forces”}} = \boldsymbol{B}(\boldsymbol{q})\underbrace{\boldsymbol{u}}_{\text{“Usually }\boldsymbol{\tau}\text{”}} + \underbrace{\boldsymbol{F}_{\text{ext}}}_{\text{“External forces”}}$$

Velocity Kinematics:

$$\dot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{v}$$

and thus,

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{G}(\boldsymbol{q})\boldsymbol{v} \\ \boldsymbol{M}^{-1}(\boldsymbol{q})\Big(\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{F}_{\text{ext}} - \boldsymbol{C}(\boldsymbol{q},\boldsymbol{v})\Big) \end{bmatrix}$$

Manipulator Equation Examples

Double Integrator:

$$\boldsymbol{M}(\boldsymbol{q}) = K, \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{v}) = \boldsymbol{0}, \boldsymbol{B}(\boldsymbol{q}) = \boldsymbol{I}, \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{I}$$

Pendulum:

$$\boldsymbol{M}(\boldsymbol{q}) = ml^2, \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{v}) = mgl\sin\theta + bml^2\dot{\theta}, \boldsymbol{B}(\boldsymbol{q}) = \boldsymbol{I}, \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{I}$$

This is basically a re-write of Euler-Langrange equations of Kinematic and Potential Energy.

Equilibria

- **Point(s) where the system will remain at rest (i.e. not moving):** $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0}$.
- **Algebraically:** roots of the dynamics function.
- **Uncontrolled double integrator:**

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{0} \end{bmatrix}$$

$$\dot{\boldsymbol{q}} = 0$$

whe(n/r)ever we have zero velocity!

Equilibria (2)

- **Uncontrolled Pendulum:**

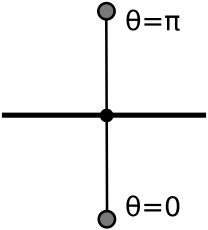
$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- **Uncontrolled Pendulum:**

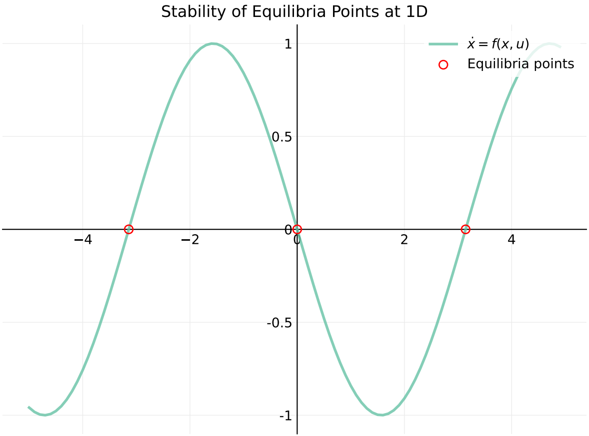
$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\dot{\theta} = 0$$

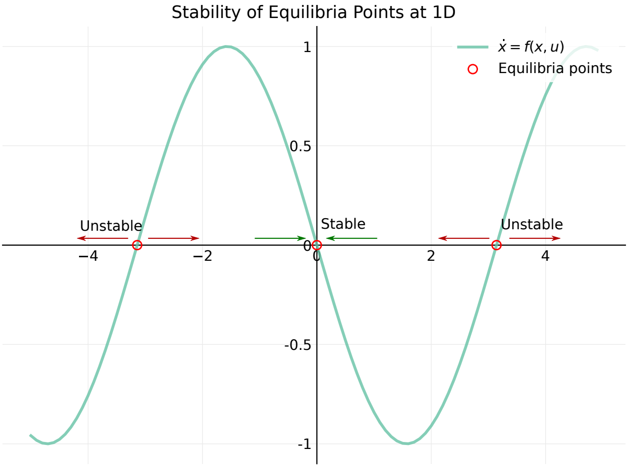
$$\theta = 0, \pm\pi$$



Informal Definition: Do we stay "close" to the equilibrium point under perturbations?



Stability of Equilibria Points (2)



Stability of Equilibria Points (3)

- **In 1-D:** $\frac{df}{d\boldsymbol{x}} < 0 \rightarrow$ "Stable", $\frac{df}{d\boldsymbol{x}} > 0 \rightarrow$ "Unstable"
- **In N-D:** $\frac{df}{d\boldsymbol{x}}$ is a Jacobian matrix:
  - We do an eigenvalue decomposition,
  - if **all** real-parts of the eigenvalues are negative, $\mathrm{Re}[\mathrm{eig}(\frac{df}{d\boldsymbol{x}})] < 0$, then "Stable",
  - if **any** real-part of the eigenvalues is positive, $\mathrm{Re}[\mathrm{eig}(\frac{df}{d\boldsymbol{x}})] > 0$, then "Unstable".

Stability of Equilibria Points - Pendulum Example

Let's see first for $\theta = \pi$:

$$f(\boldsymbol{x}) = \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix}$$

Stability of Equilibria Points - Pendulum Example

Let's see first for $\theta = \pi$:

$$f(\boldsymbol{x}) = \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l}\cos\theta & -b \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}}\Big|_{\theta=\pi} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -b \end{bmatrix}$$

$$\text{eig}(\frac{df}{d\boldsymbol{x}}) = -\frac{b}{2} \pm \sqrt{\frac{0.25b^2 l + g}{l}}$$

Stability of Equilibria Points - Pendulum Example

Let's see first for $\theta = \pi$:

$$f(\boldsymbol{x}) = \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l}\cos\theta & -b \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}}\Big|_{\theta=\pi} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -b \end{bmatrix}$$

$$\text{eig}(\frac{df}{d\boldsymbol{x}}) = -\frac{b}{2} \pm \sqrt{\frac{0.25b^2 l + g}{l}}$$

Unstable if $\sqrt{\frac{0.25b^2l+g}{l}} > \frac{b}{2}$(true for realistic cases $g = 9.81$, $l \leqslant 1$, $b \leqslant 0.5$)

Now let's see for $\theta = 0$:

$$f(\boldsymbol{x}) = \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l}\cos\theta & -b \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}}|_{\theta=0} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l} & -b \end{bmatrix}$$

$$\text{eig}(\frac{df}{d\boldsymbol{x}}) = -\frac{b}{2} \pm \sqrt{\frac{0.25b^2 l - g}{l}}$$

Now let's see for $\theta = 0$:

$$f(\boldsymbol{x}) = \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin\theta - b\dot{\theta} \end{bmatrix}$$

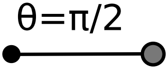
$$\frac{df}{d\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l}\cos\theta & -b \end{bmatrix}$$

$$\frac{df}{d\boldsymbol{x}}|_{\theta=0} = \begin{bmatrix} 0 & 1 \\ \frac{-g}{l} & -b \end{bmatrix}$$

$$\text{eig}(\frac{df}{d\boldsymbol{x}}) = -\frac{b}{2} \pm \sqrt{\frac{0.25b^2 l - g}{l}}$$

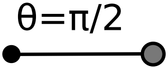
$\text{eig}(\frac{df}{d\boldsymbol{x}}) = -\frac{b}{2} \pm i\sqrt{\frac{0.25b^2 l - g}{l}}$ for realistic cases $(g = 9.81, l \leqslant 1, b \leqslant 0.5)$
and thus the equilibrium is **Stable!!**

Alright! Let's "see" some action! How can we "move" the equilibrium to another point? Let's try $\theta = \frac{\pi}{2}$.



First Control Problem!

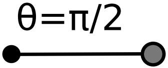
Alright! Let's "see" some action! How can we "move" the equilibrium to another point? Let's try $\theta = \frac{\pi}{2}$.



$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin(\frac{\pi}{2}) + \frac{1}{ml^2}\boldsymbol{u} - b\dot{\theta} \end{bmatrix} = \mathbf{0}$$

First Control Problem!

Alright! Let's "see" some action! How can we "move" the equilibrium to another point? Let's try $\theta = \frac{\pi}{2}$.



$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{-g}{l}\sin(\frac{\pi}{2}) + \frac{1}{ml^2}\boldsymbol{u} - b\dot{\theta} \end{bmatrix} = \boldsymbol{0}$$

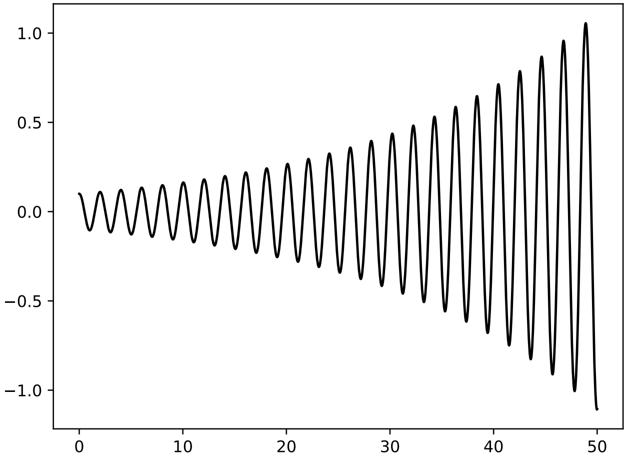
Thus,

$$\frac{1}{ml^2}\boldsymbol{u} = \frac{g}{l}\sin(\frac{\pi}{2}) \Rightarrow \boldsymbol{u} = mgl$$

- **Why do we care about discretization?**
  - We can't "solve" $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ in the general case,
  - Computers cannot handle continuous t in representing $\boldsymbol{x}(t)$,
  - Sometimes discrete time dynamics can describe better the situation (e.g. contact-based dynamics).
- **How?**

Discretization of Dynamical Systems

- **Why do we care about discretization?**
  - We can't "solve" $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ in the general case,
  - Computers cannot handle continuous t in representing $\boldsymbol{x}(t)$,
  - Sometimes discrete time dynamics can describe better the situation (e.g. contact-based dynamics).
- **How?**
  - $\boldsymbol{x}_{k+1} = f_{\text{discrete}}(\boldsymbol{x}_k, \boldsymbol{u}_k)$
  - **Euler Integration:** $\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{x}_k + \underbrace{f(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\text{"Continuous Dynamics"}} \underbrace{dt}_{\text{"Timestep"}}}_{f_{\text{discrete}}(\boldsymbol{x}_k, \boldsymbol{u}_k)}$


1.0
0.5
0.0
−0.5
−1.0
0
10
20
30
40
50

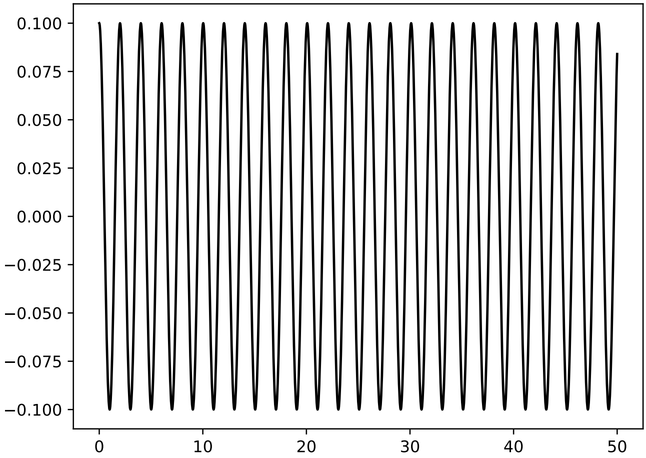
Runge Kutta 4th Order (RK4)

- **Do not use Euler Integration!**
- **What else?**
  - **Runge Kutta 4th Order (RK4):**

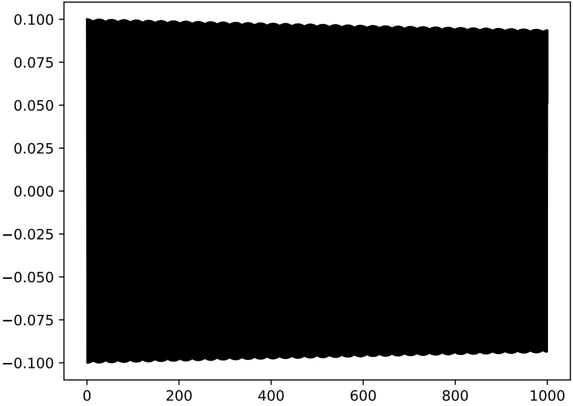
$$
\begin{aligned}
\boldsymbol{f}_1 &= f(\boldsymbol{x}_k, \boldsymbol{u}_k) \\
\boldsymbol{f}_2 &= f(\boldsymbol{x}_k + \boldsymbol{f}_1\frac{dt}{2}, \boldsymbol{u}_k) \\
\boldsymbol{f}_3 &= f(\boldsymbol{x}_k + \boldsymbol{f}_2\frac{dt}{2}, \boldsymbol{u}_k) \\
\boldsymbol{f}_4 &= f(\boldsymbol{x}_k + \boldsymbol{f}_3 dt, \boldsymbol{u}_k) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \frac{dt}{6}\Big(\boldsymbol{f}_1 + 2\boldsymbol{f}_2 + 2\boldsymbol{f}_3 + \boldsymbol{f}_4\Big)
\end{aligned}
$$

  - RK4 basically fits a cubic polynomial.

RK4 - Pendulum Code/Simulation



The simulation is damped!



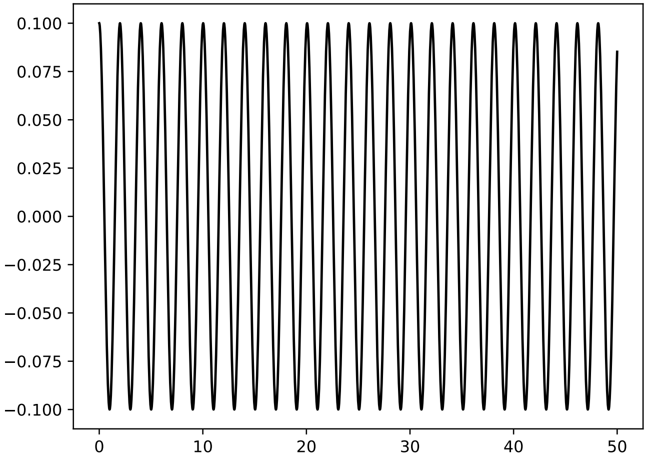
Semi-Implicit Euler Integration

- **RK4 is nice, but not perfect!**
- **What else?**
  - **Semi-Implicit Euler Integration:**

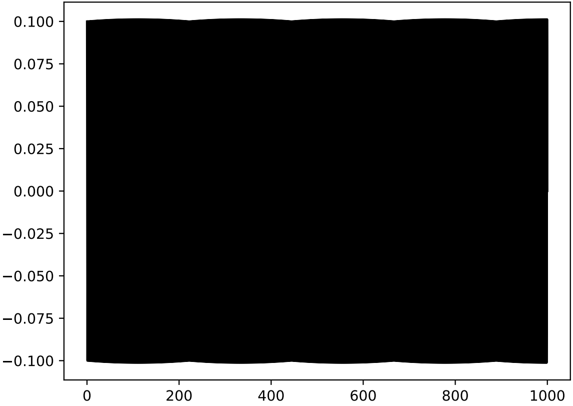
$$\begin{bmatrix} \dot{\boldsymbol{q}}_k \\ \dot{\boldsymbol{v}}_k \end{bmatrix} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$$
$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \dot{\boldsymbol{v}}_k dt$$
$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \boldsymbol{v}_{k+1} dt$$

  - We basically update first the velocity of the system, and then use the $k+1$ velocity to update the position part.

Semi-Implicit Euler - Pendulum Code/Simulation



Undamped simulation! Semi-Implicit Euler preserves energy!



Implicit Integrators

So far we have been discussing explicit integrators. Implicit integrators are defined as:

$$f_{\text{discrete}}(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k, \boldsymbol{u}_k) = \mathbf{0}$$

We solve this as a root-finding problem. We know $\boldsymbol{x}_k, \boldsymbol{u}_k$ and we need to solve for $\boldsymbol{x}_{k+1}$.

Simplest form is "Backward Euler":

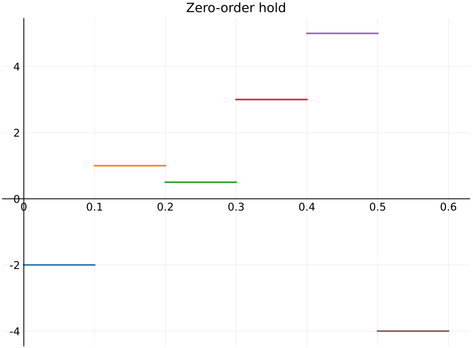
$$f_{\text{discrete}}(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k - f(\boldsymbol{x}_{k+1}, \boldsymbol{u}_k)dt = \mathbf{0}$$

Integrators - Summary

- **Do not use Forward Euler!**
- RK4 is the "industry standard", but comes with damped behavior (unrealistic)
- Implicit methods are more accurate, but require optimization!
- Semi-Implicit Euler is a middle ground (not always good enough!)
- Be careful! Test! Test! Test!

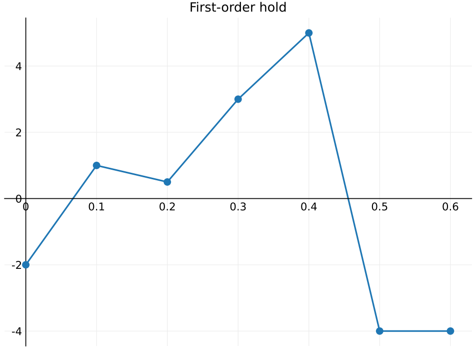
How can we discretize the control inputs?

- **“Zero-order hold”**



How can we discretize the control inputs?

- **"First-order hold"**



- **Higher order polynomials/Splines!**

Control Affine Linear Systems

Time Varying Linear Systems:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u}$$

Time Invariant Linear Systems:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

Also:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{d}$$

They are VERY important for controlling robotic systems and control in general!

Integration of Linear Systems

- All integrators that we have seen above work for any $f(\cdot, \cdot)$,
- **Can we do better if we know the system is linear?**

- All integrators that we have seen above work for any $f(\cdot, \cdot)$,
- **Can we do better if we know the system is linear?**

Let's start with the easiest case (1D):

$$\begin{aligned} \dot{x} = \frac{dx}{dt} &= \alpha x \\ \int \frac{1}{x} dx &= \int \alpha dt \\ \ln x &= \alpha t + C \\ x &= e^{\alpha t} x_0 \end{aligned}$$

Integration of Linear Systems (2)

What happens in higher dimensions?

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$
$$\boldsymbol{x} = e^{\boldsymbol{A}t}\boldsymbol{x}_0$$

$e^{\boldsymbol{A}t}$ is the matrix exponential.

Integration of Linear Systems (2)

What happens in higher dimensions?

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$
$$\boldsymbol{x} = e^{\boldsymbol{A}t}\boldsymbol{x}_0$$

$e^{\boldsymbol{A}t}$ is the matrix exponential. We can write (why?):

$$\boldsymbol{x}_{k+1} = e^{\boldsymbol{A}dt}\boldsymbol{x}_k$$

We have analytically discretized a linear system! No need for explicit/implicit integrators!

We usually call $\boldsymbol{A}_d = e^{\boldsymbol{A}dt}$ a state transition matrix: $\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k$.

Stability of Linear Systems

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$

is stable if all $\mathrm{Re}[\mathrm{eig}(\boldsymbol{A})] < 0$.

What about the discretized version?

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$

is stable if all $\mathrm{Re}[\mathrm{eig}(\boldsymbol{A})] < 0$.

What about the discretized version?

$$\boldsymbol{x}_{k+1} = e^{\boldsymbol{A}dt}\boldsymbol{x}_k = \boldsymbol{A}_d\boldsymbol{x}_k$$

is stable if all $|\mathrm{eig}(\boldsymbol{A})| < 1$.

How can we use the above to infer if my discretized system is stable?

Stability of Non-Linear Discrete Systems?

How can we use the above to infer if my discretized system is stable? For the general case, we have:

$$\boldsymbol{x}_{k+1} = f_{\text{discrete}}(\boldsymbol{x}_k)$$

How can we make this linear to study its stability?

Stability of Non-Linear Discrete Systems?

How can we use the above to infer if my discretized system is stable? For the general case, we have:

$$\boldsymbol{x}_{k+1} = f_{\text{discrete}}(\boldsymbol{x}_k)$$

How can we make this linear to study its stability? We can, but only around an equilibrium point $\boldsymbol{x}_0$:

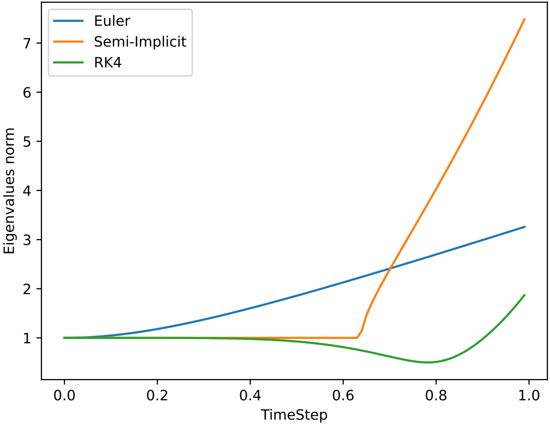
$$\boldsymbol{x}_{k+1} = \underbrace{\frac{\partial f_{\text{discrete}}}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_0}}_{\boldsymbol{A}_{d_{\boldsymbol{x}_0}}} \boldsymbol{x}_k$$

So now we have:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{d_{\boldsymbol{x}_0}} \boldsymbol{x}_k$$

and we can do the linear case analysis here.

Stability of Integrators: Pendulum - Code Example



How can we integrate the following system?

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

How can we integrate the following system?

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

- We assume zero order hold and thus $\dot{\boldsymbol{u}} = \boldsymbol{0}$,

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix}$$

- So now we can write:

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_{k+1} \end{bmatrix} = e^{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} dt} \begin{bmatrix} \boldsymbol{x}_{k} \\ \boldsymbol{u}_{k} \end{bmatrix}$$

Integration of Controlled Systems (2)

We write:

$$\begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{B}_d \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = e^{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} dt}$$

And thus:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d \boldsymbol{x}_k + \boldsymbol{B}_d \boldsymbol{u}_k$$

Thank you

- **Any Questions?**
- **Office Hours**:
  - **Tue & Thu** *(09:00-11:00)*
  - 24/7 by email (`costashatz@upatras.gr`, subject: *ECE_RSII_AM*)
- **Material and Announcements**

Laboratory of Automation & Robotics