Lecture 6

27.10.2025

Today's plan and announcements

- Hour 1: conditions for AI to benefit humanity
- Hour 2: Background for reinforcement learning

- Exercise hour this week
  - Part 1 of python homework - NN for obstacle detection
  - Problem set 3 - on dynamical systems

Conditions for AI to benefit humanity

AI just like any other technology, can have positive and negative impact

Our goal is to think critically about such impacts by stepping back and reflecting on

- Who drives AI research and development?
- What impact it has and will have, and on whom?
- What responsibilities we have to ensure the technology serves humanity?

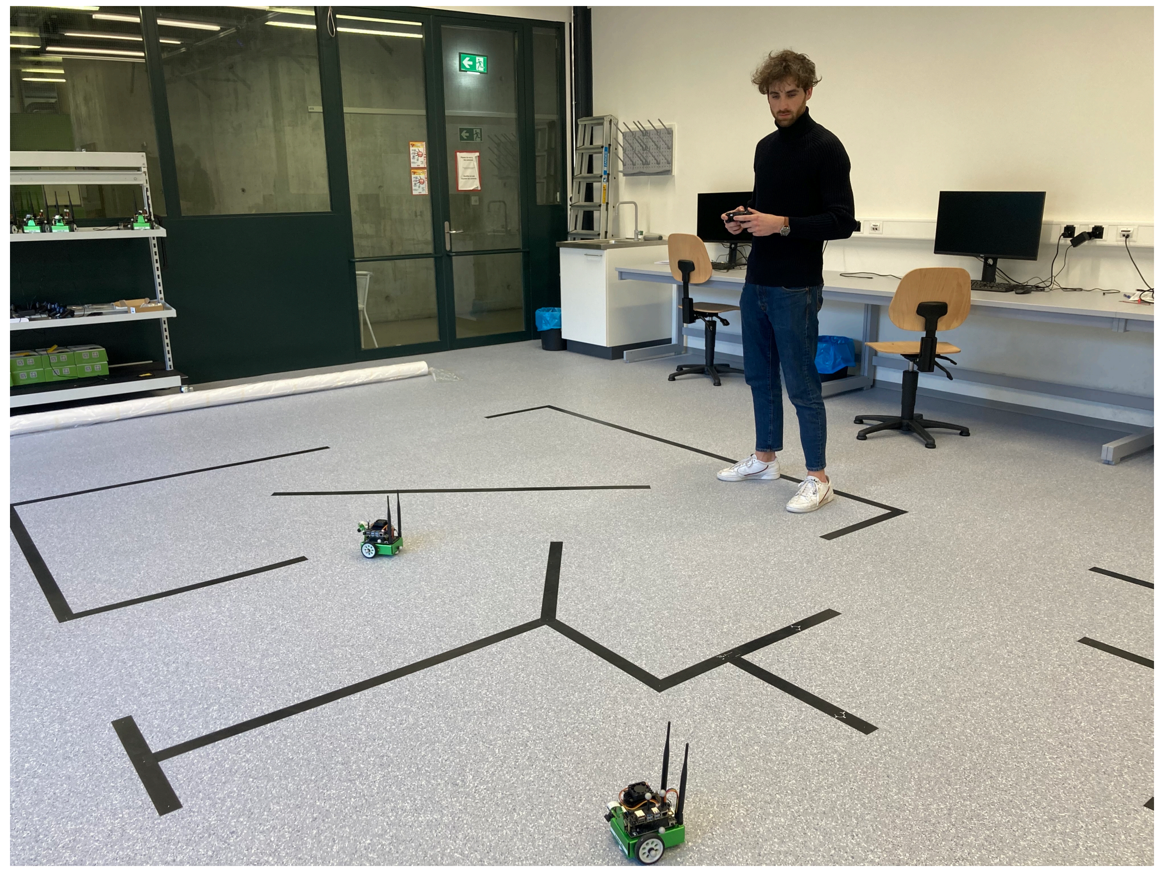
Indicative feedback result

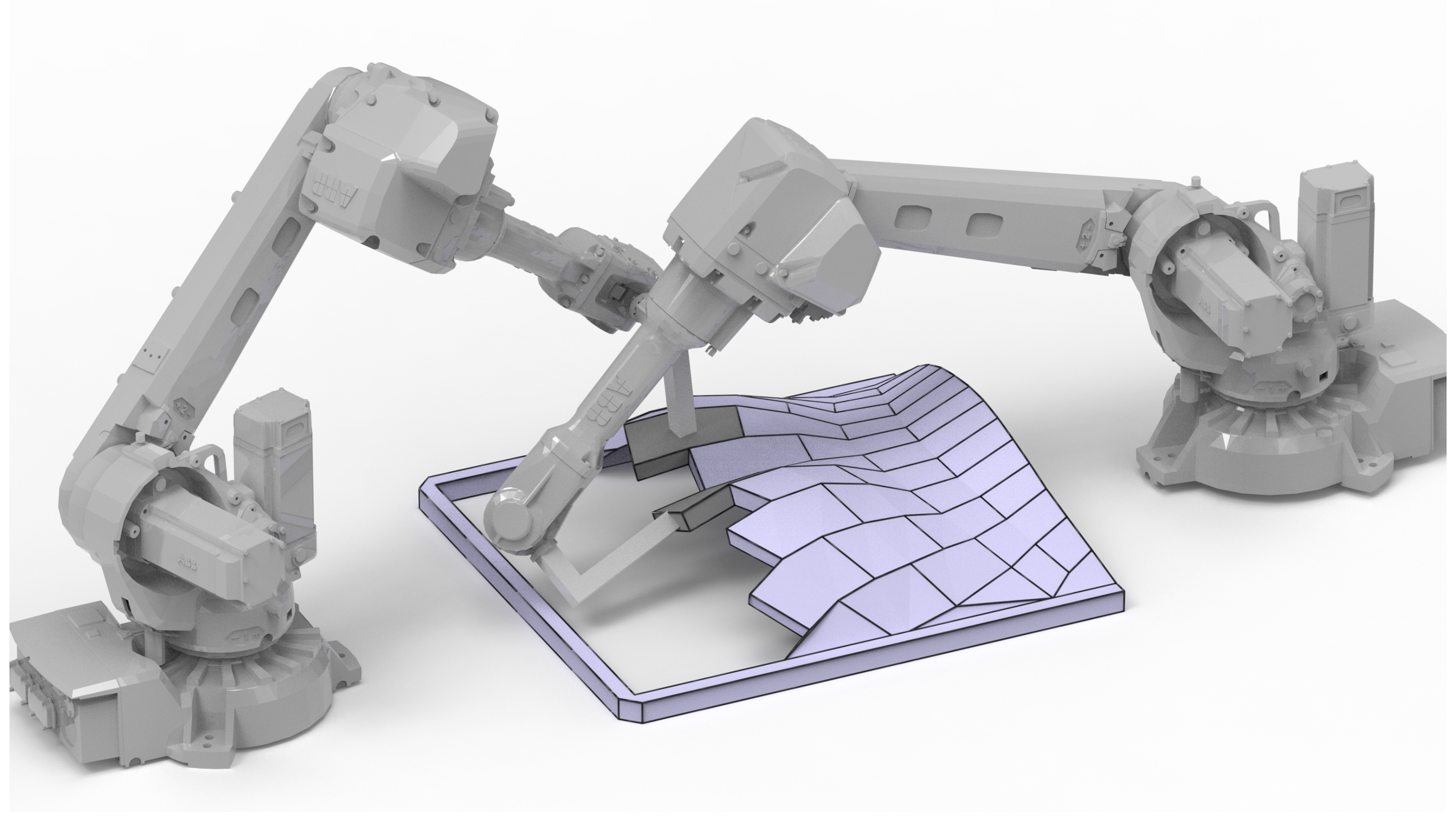
- What you told me
  - The course is interesting and well-structured
  - The Python exercises help connect theory to practice
  - Some of you find the lectures a bit dense and fast
  - A few noted the microphone and voice volume make it hard to follow
  - Some miss the blackboard pace and want more examples or pauses
- What I will do
  - Slow down on key derivations and pause
  - Add more intuitive examples
  - Speak louder and test microphone setup at beginning
  - Keep the structure, problem sets and python exercises

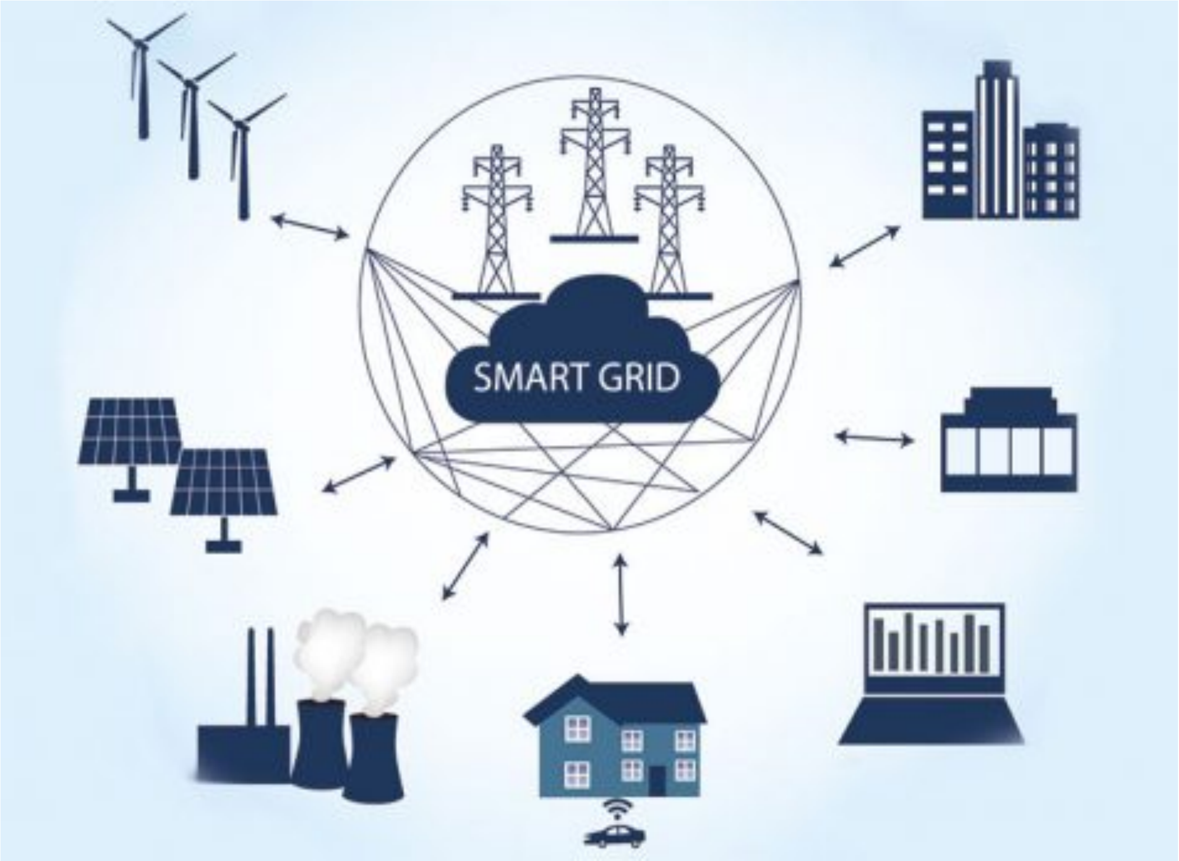
Introduction to reinforcement learning

Dynamical systems

Examples from our research





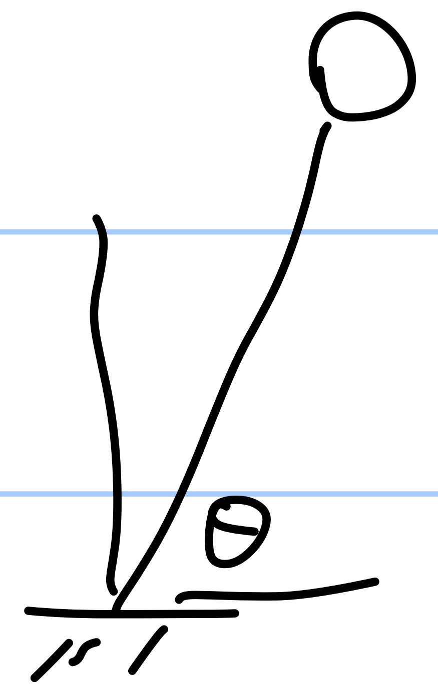


- wind turbine, energy harvesting
- bioengineering / biomechanical
- LLMs

Dynamical systems and differential equations

- Dynamical system: a system that evolves with time according to some evolution rule
  - so far, you have seen this rule being set by differential equations
  - Inverted pendulum

$$\ddot{\theta} = mg \sin \theta$$

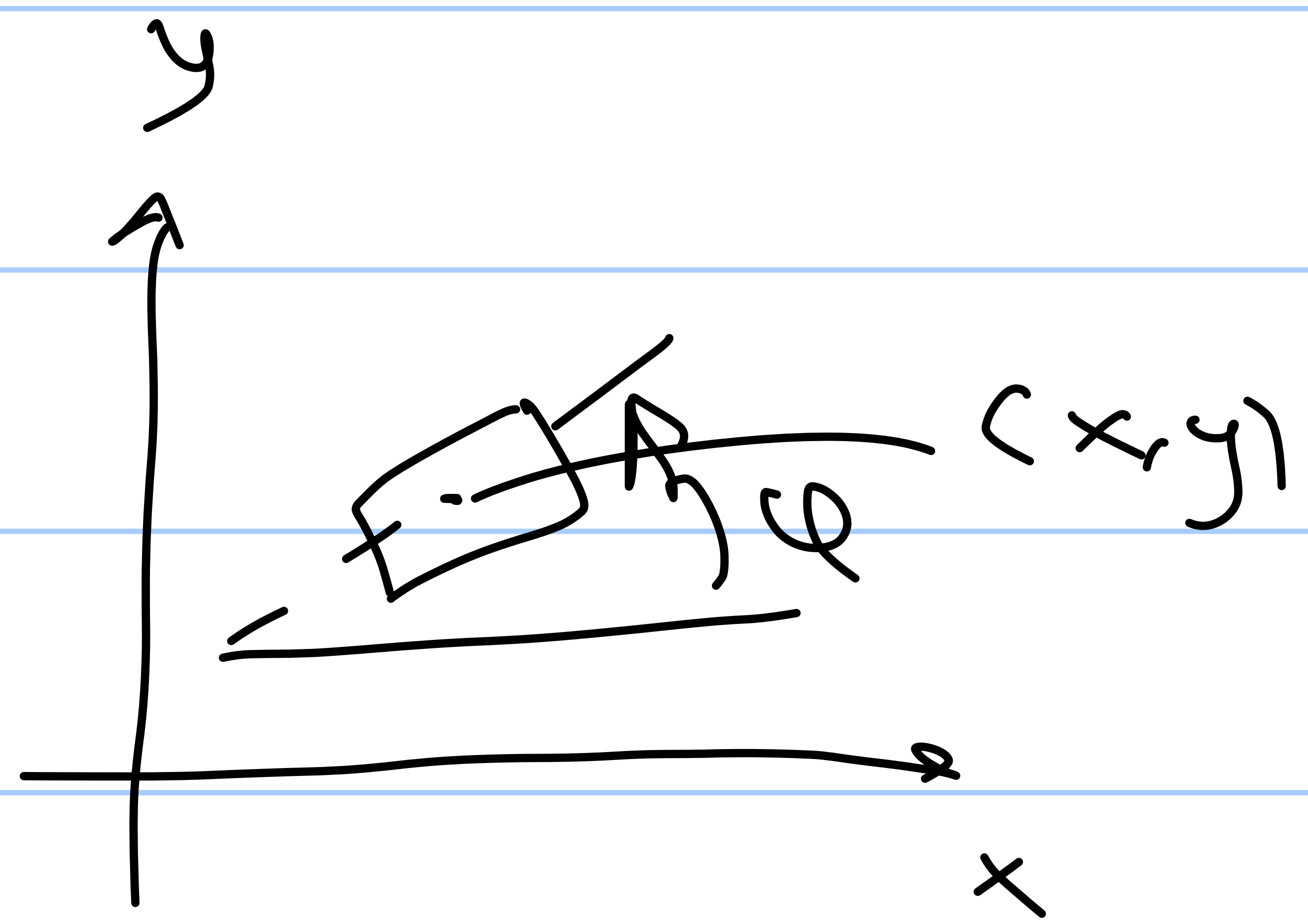


  - Robot unicycle model

V : forward velocity

u : angular velocity

$$\dot{x} = V \cos \varphi$$
$$\dot{y} = V \sin \varphi$$
$$\dot{\varphi} = u$$



EPFL

Dynamical systems in state-space form

- State of a system summarizes all you need about the past in order to predict the future
  - State space S, input space A
- Inverted pendulum

$\ddot{\theta} = mg\sin\theta$

$S = \mathbb{R}^2$

$(\mathbb{R} \times [0, 2\pi))$

$$\left.\begin{aligned} s_1 &= \theta \\ s_2 &= \dot{\theta} \end{aligned}\right\} \longrightarrow \begin{aligned} \dot{s}_1 &= s_2 \\ \dot{s}_2 &= mg\sin(s_1) \end{aligned}$$

- Robot unicycle model

a_1, a_2 : control inputs

$\dot{x} = V\cos\varphi$

$\dot{y} = V\sin\varphi$

$\dot{\varphi} = u$

$$\left.\begin{aligned} s_1 &= x \\ s_2 &= y \\ s_3 &= \varphi \end{aligned}\right\} \quad \begin{aligned} \dot{s}_1 &= a_1\cos s_3 \\ \dot{s}_2 &= a_1\sin s_3 \\ \dot{s}_3 &= a_2 \end{aligned}$$

$S = \mathbb{R}^3 \quad (\mathbb{R}^2 \times [0, 2\pi))$

$A = \mathbb{R}^2$

Dynamical systems control

- Goal: design input for the system to achieve a desired behavior
  - Stability: having bounded deviation from an equilibrium
    - ex: the pendulum stays upright: $(\theta, \dot{\theta}) = (\pi, 0)$; the grid frequency stays at 50Hz
  - Reachability: reaching a desired target point $\exists\, T$ (time) s.t. $s(T) \in G \subset S$
    - ex: robot going to a goal position

Goal set.

  - Safety: staying inside a safe set $\forall t \quad s(t) \in \text{Safe set} \subset S$
    - ex: robot not crashing

Optimal control in continuous time

- Optimal control: formulating the desired objective for our system through a functional

$$\min_{\mu:} \int_0^T \underbrace{c(s(t), \mu(t))}_{\text{running cost}} dt + \underbrace{c_T(s_T)}_{\text{final cost}} \quad \text{subject to} \quad \dot{s}(t) = f(s(t), \mu(t))$$

$[0,T] \to A$

ex: fuel consumption ex: deviation from a desired target

$\mu : [0, T] \to A$ control input

- Optimal control theory: characterizes the function solving the above problem
  - Generally, difficult to find the solution through its theoretical characterization

- <u>Discrete-time</u> <u>stochastic</u> control addresses some of these challenges

Dynamical systems - discretization

- Motivation: most actuators and sensors are implemented digitally over the past 50 years

- Euler discretization: approximate the differentiation by finite difference

dynamical system $\dot{s}(t) = f(s(t), a(t))$

$$\dot{s}(t) = \frac{ds}{dt} \simeq \frac{s(t+\delta) - s(t)}{\delta}$$ finite difference

$$s(t+\delta) = s(t) + \delta f(s(t), a(t))$$

by defining $s_k = s(t + k\delta)$, $a_k = a(t + k\delta)$

$$\boxed{s_{k+1} = f_d(s_k, a_k)}, \text{ where } f_d(s_k, a_k) = s_k + \delta f(s_k, a_k)$$

Discrete-time dynamical system

Euler discretization examples

- Inverted pendulum discretization

$$\dot{s}_1 = s_2$$
$$\dot{s}_2 = mg \sin(s_1)$$

$$s_{1,k+1} = s_{1,k} + \delta s_{2,k}$$

$$s_{2,k+1} = s_{2,k} + \delta mg \sin s_{1,k}$$

where $s_{i,k}$ is state i at time k.

- Unicycle model discretization

exercise.

$$\dot{s}_1 = a_1 \cos s_3$$
$$\dot{s}_2 = a_1 \sin s_3$$
$$\dot{s}_3 = a_2$$

Discrete-time optimal control

- From optimization over functions to optimization over vectors

$$\min \sum_{k=0}^{K-1} c(s_k, a_k)dt + c_K(s_K)$$

, subject to $s_{k+1} = f_d(s_k, a_k)$

- decision variable $a_0, a_1, \ldots, a_{K-1} \in A$

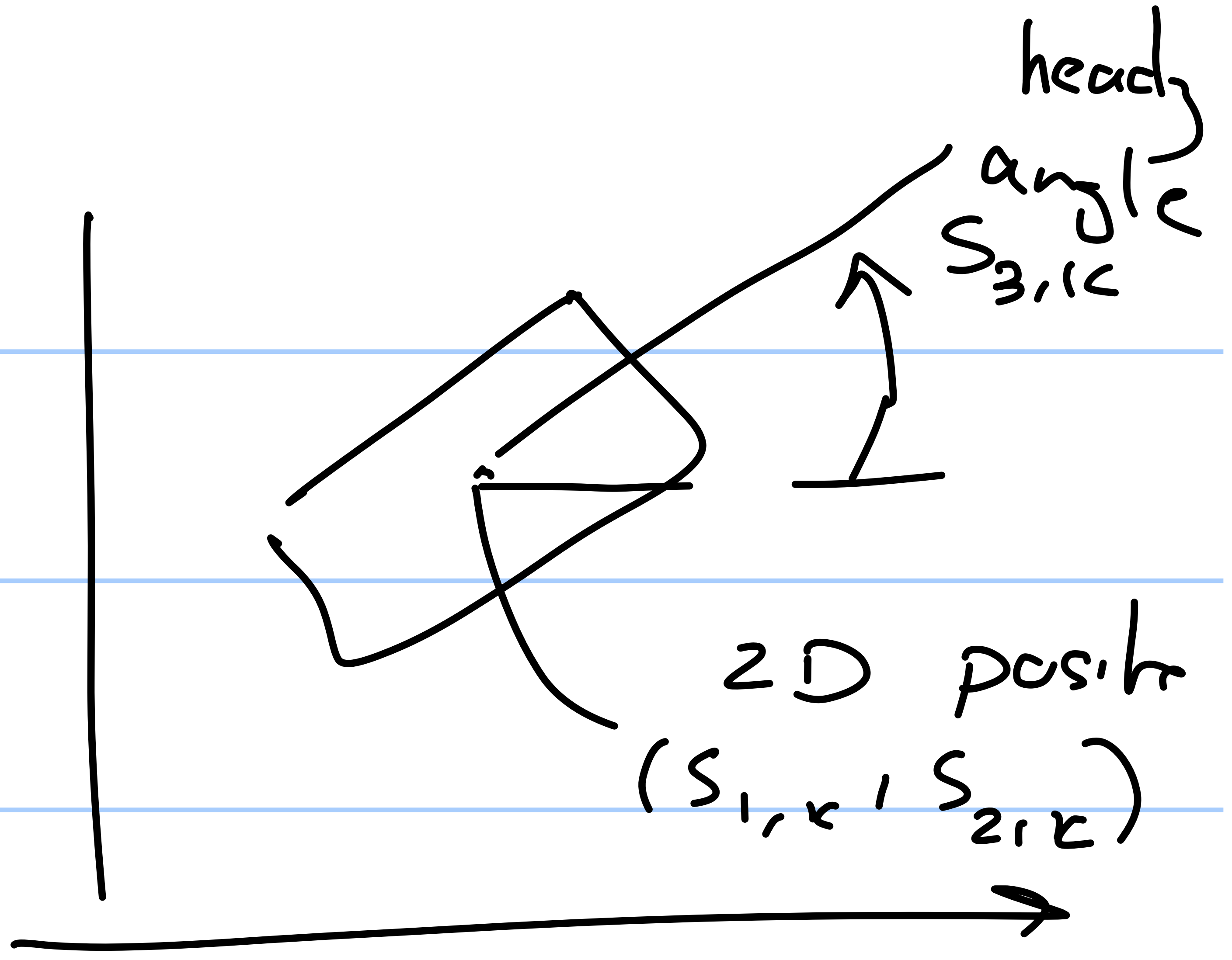
compare to $u_\cdot : [0, T] \rightarrow A$ function

if each $a_k \in \mathbb{R}^m$, then decision variable is in $\mathbb{R}^{Km}$

- What if there is uncertainty in our models?
  - Unknown parameters (friction, mass) in pendulum, wind affecting aircraft flight, etc..
  - Open-loop control $\{a_0, a_1, \ldots, a_{K-1}\}$ doesn't account for model mismatch

Stochastic dynamical systems

- Probabilistic dynamics: $s_{k+1} \sim P(\cdot \mid s_k, a_k)$
  - Example: unicycle model with uncertainty and disturbances



$$s_{1,k+1} = s_{1,k} + \delta\, a_{1,k} \cos(s_{3,k}) + w_{1,k}$$

$$s_{2,k+1} = s_{2,k} + \delta\, a_{1,k} \sin(s_{3,k}) + w_{2,k}$$

$$s_{3,k+1} = s_{3,k} + \delta\, a_{2,k} + w_{3,k}$$

$w_{i,k}$: uncertainty in model or disturbances affecting state i at time k.

ex: $w_{i,k} \sim N(0, \sigma_i^2)$ iid

$$s_{k+1} \sim N(f_d(s_k, a_k), \Sigma), \quad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}$$

Control policy

- Policy is a map from state space to input space, $\pi_k : S \rightarrow A$
- Closed-loop control: at each time k, we observe s_k and apply input $a_k = \pi_k(s_k)$,

<u>example</u> : in unicycle model

$$\pi_k : \mathbb{R}^2 \times [0, 2\pi) \longrightarrow \mathbb{R}^2$$

states : 2D position, heading angle $\quad S = \mathbb{R}^2 \times [0, 2\pi)$

input : forward velocity, angular velocity $\quad A = \mathbb{R}^2$

Stochastic optimal control

$$J(s_0) = \min \mathbb{E}[\sum_{k=0}^{K-1} c(s_k, a_k) + c_K(s_K)]$$

- Dynamic programming

Solving the stochastic optimal control problem

- Dynamic programming
  - Can be used to tractably solve the problem for few special cases (small dimensions, linear dynamics and quadratic costs)
  - For more general cases, need for approximation approaches

- Reinforcement learning: an approximation approach for dynamic programming
  - It can be model-free - useful if model is complex or high dimensional
  - Provable theory for few special cases

Summary

- Reflection: Conditions for AI to benefit humanity
  - What does it mean to benefit humanity? Who defines “benefit”? For whom is this benefit?
  - How would you choose the projects you work on towards the goal of benefiting humanity?

- Dynamical systems
  - State-space representation
  - Continuous-time differential equations and discrete-time difference equations
  - Stochastic dynamics
  - Optimal control

- Your tasks this week
  - Go through Problem set 3
  - Start working on python homework