ORF522 – Linear and Nonlinear Optimization

1. Introduction

Bartolomeo Stellato - Fall 2020

Meet your classmates!

What is this course about?

The mathematics behind making optimal decisions

What is this course about?

The mathematics behind making optimal decisions

Variables

What is this course about?

The mathematics behind making optimal decisions

Variables

Objective

What is this course about?

The mathematics behind making optimal decisions

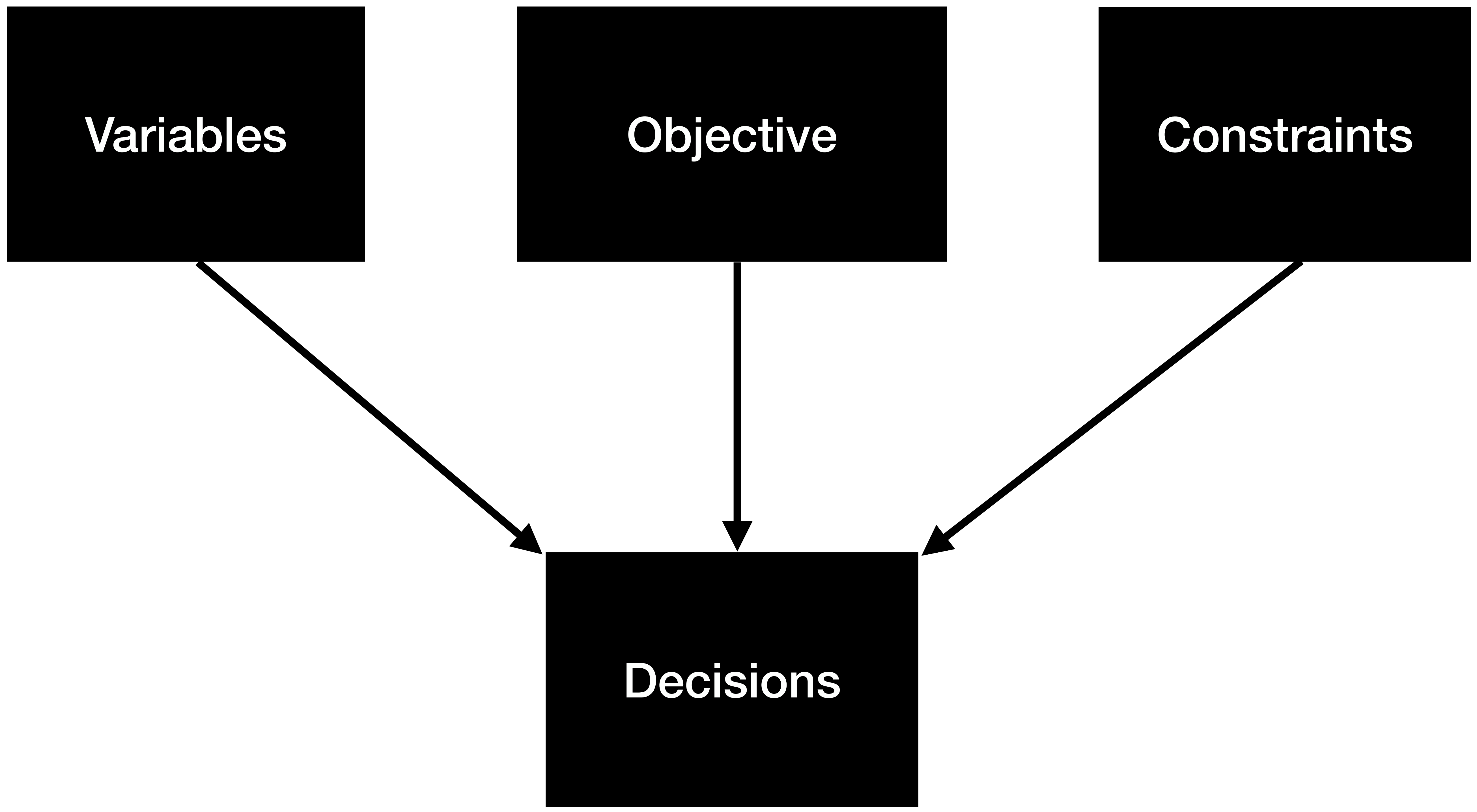
Variables

Objective

Constraints

What is this course about?

The mathematics behind making optimal decisions

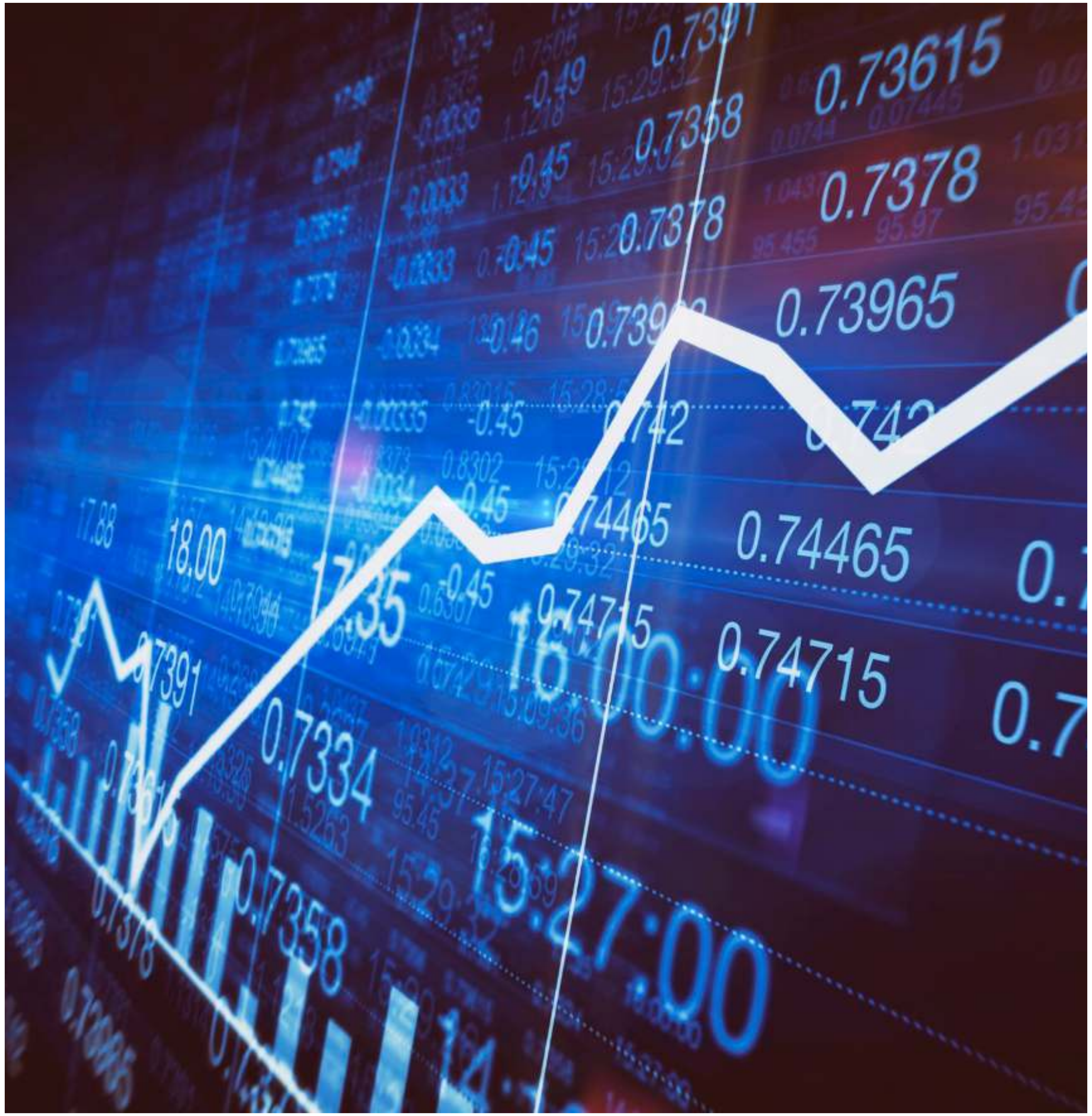


Finance

Variables
Amounts invested in each asset

Constraints
Budget, investment per asset,
minimum return, etc.

Objective
Maximize profit, minus risk



Optimal control

Variables
Inputs: thrust, flaps, etc.

Constraints
System limitations, obstacles, etc.

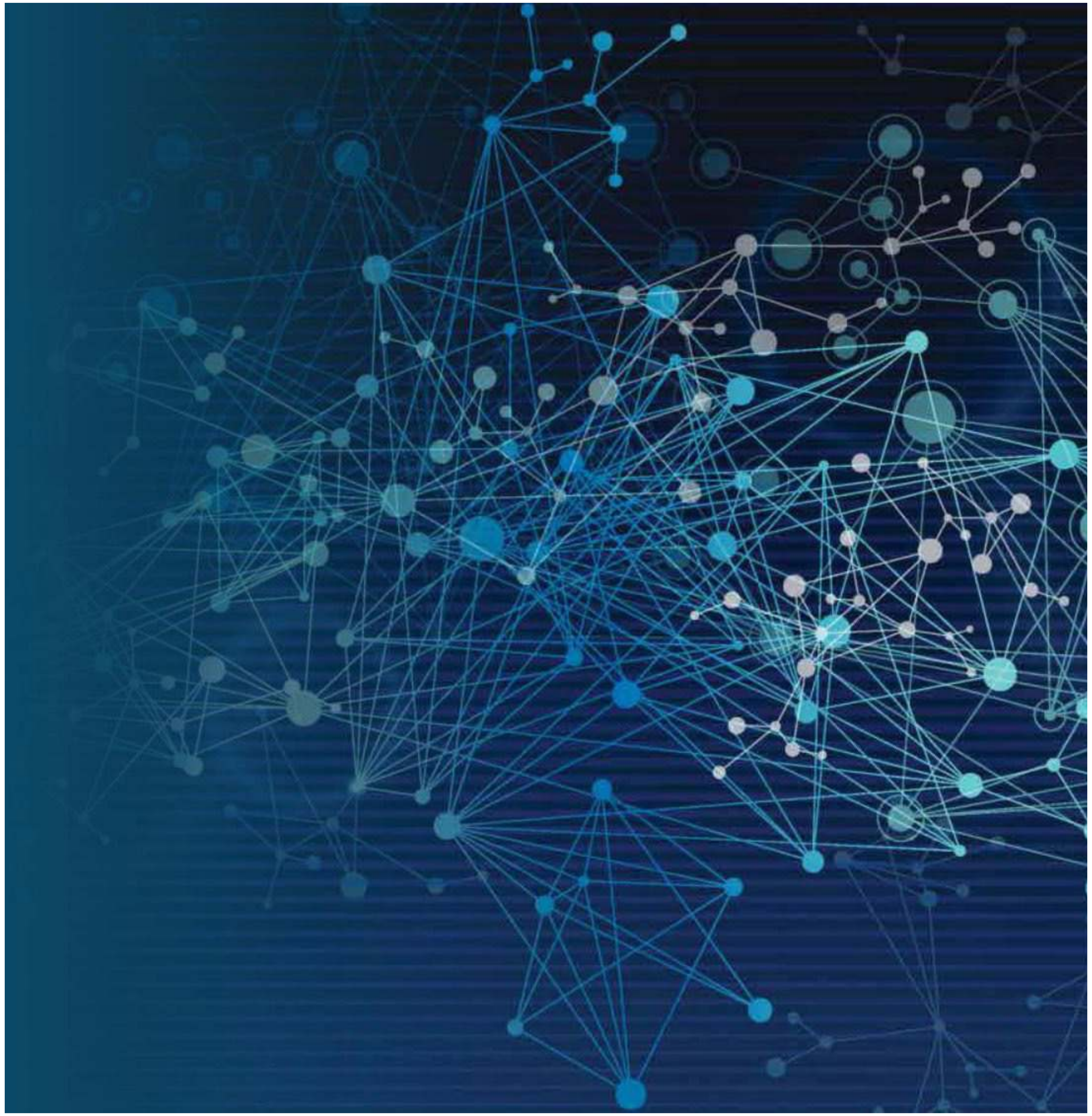
Objective
Minimize distance to target and fuel consumption

Machine learning

Variables
Model parameters

Constraints
Prior information, parameter limits

Objective
Minimize prediction error, plus regularization



Mathematical optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

$x = (x_1, \ldots, x_n)$ Variables

$f : \mathbf{R}^n \to \mathbf{R}$ Objective function

$g_i : \mathbf{R}^n \to \mathbf{R}$ Constraint functions

$x^\star$ Solution/Optimal point

$f(x^\star)$ Optimal value

Most optimization problems cannot be solved

Solving optimization problems

General case → **Very hard!**

Compromises

- Long computation times
- Not finding the solution
  (in practice it may not matter)

Solving optimization problems

General case ⟶ **Very hard!**

Compromises

- Long computation times
- Not finding the solution
  (in practice it may not matter)

Exceptions

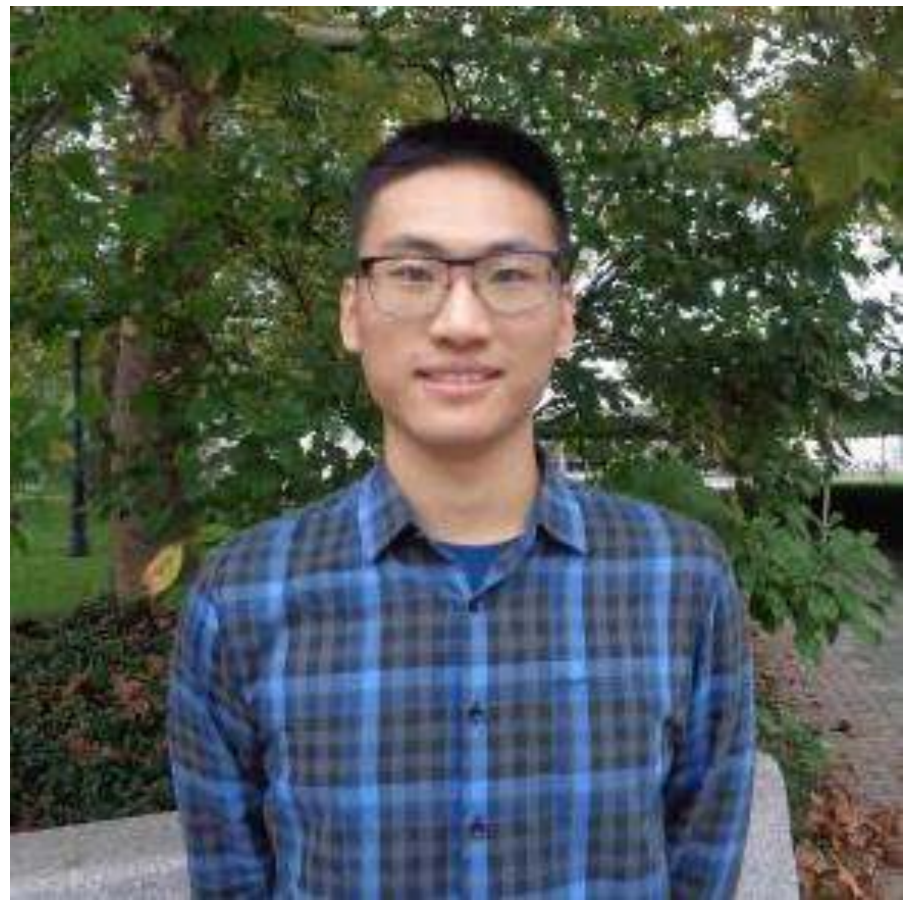
- Linear optimization
- Convex optimization

⟶ **Can be solved very efficiently and reliably**

Meet your teaching staff

Bartolomeo Stellato
I am a Professor at ORFE. I obtained my PhD from Oxford and I was a postdoc at MIT.

email: bstellato@princeton.edu
office hours: Wed 2:00pm-4:00pm EST, at this zoom link
website: stellato.io



Zheng Yu
PhD student at ORFE.

email: zhengy@princeton.edu
office hours: Fri 2:00pm-4:00pm EST, at this zoom link

Today's agenda

- Optimization problems
- History of optimization
- Course contents and information
- A glance into modern optimization

Linear optimization

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1. \ldots, m \end{array}$$

No analytical formula (99% of the time there will be none in this course!)

Efficient algorithms and software we can solve problems with several thousands of variables and constraints

Extensive theory (duality, degeneracy, sensitivity)

Linear optimization

Example: resource allocation

$$
\begin{array}{ll}
\text{maximize} & \sum_{j=1}^{n} c_i x_i \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m \\
& x_i \geq 0, \quad i = 1, \ldots, n
\end{array}
$$

c_i : profit per unit of product i shipped

b_i : units of raw material i on hand

a_{ji} : units of raw material j required to produce one unit of product i

Nonlinear optimization

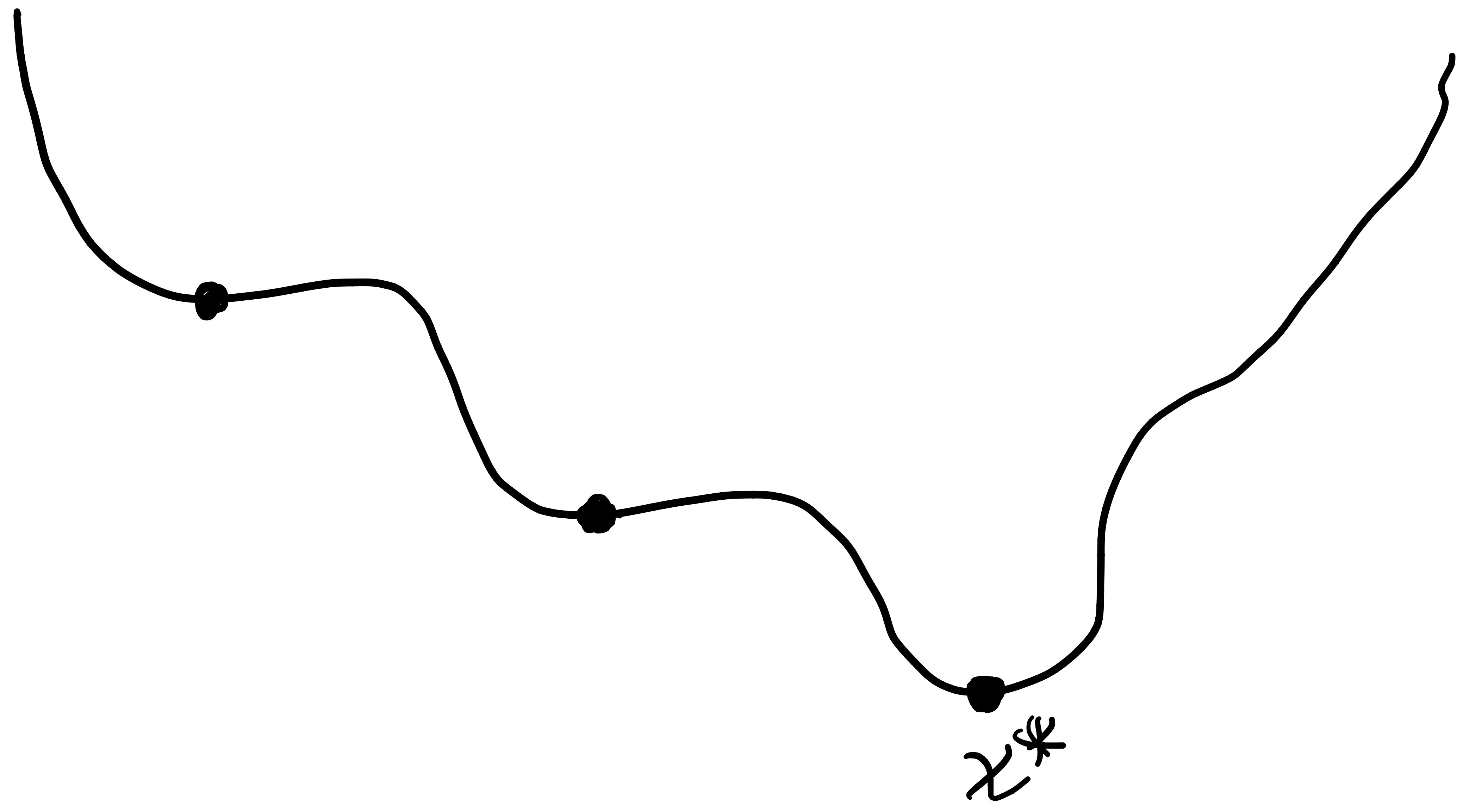
BINARY VARIABLES $x_i \in \{0,1\}$

$\rightarrow x_i(1-x_i)=0$

$$
\begin{array}{ll}
\text{minimize} & f(x) \\
\text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m
\end{array}
$$

Hard to solve in general

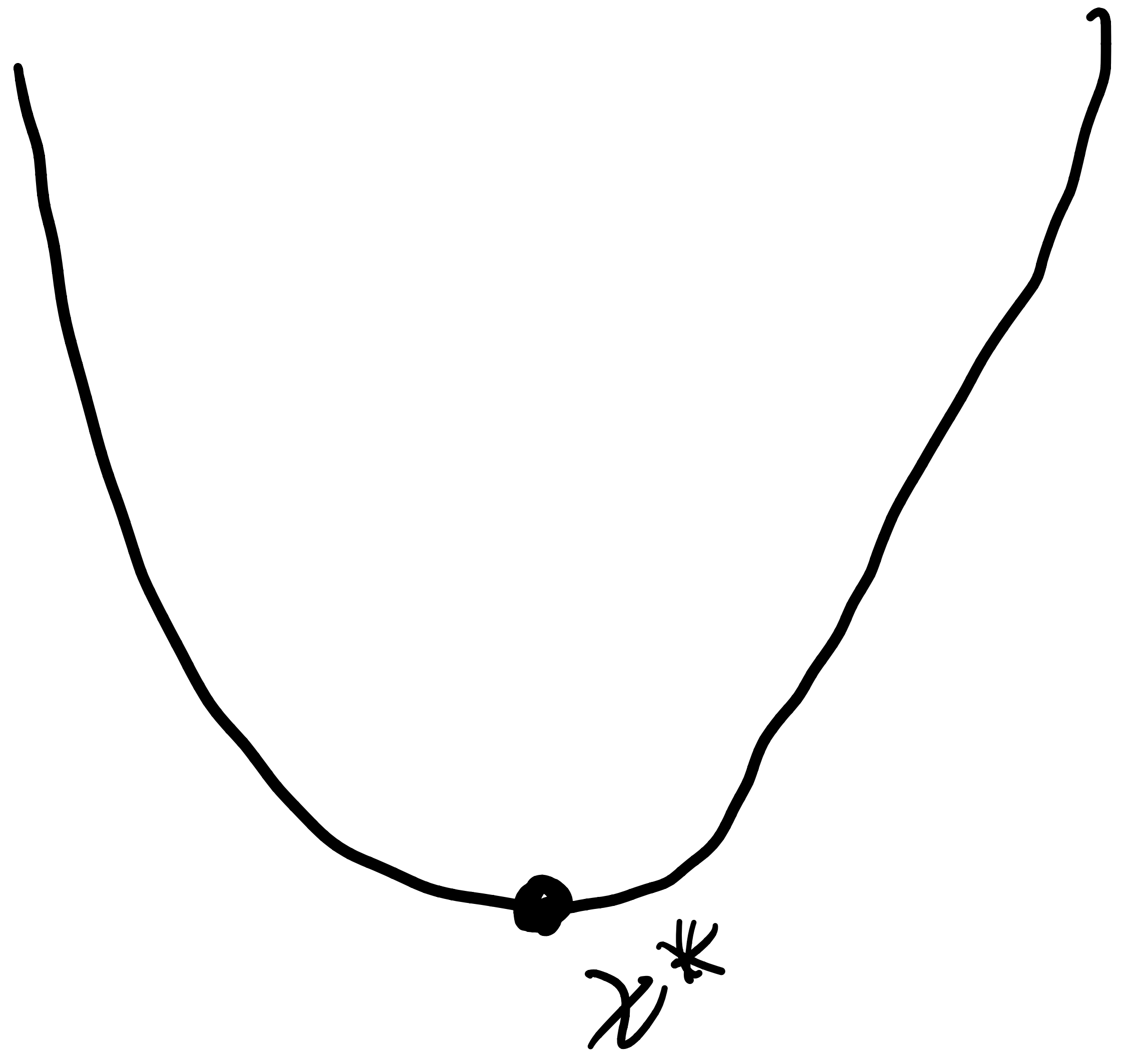
- multiple local minima
- discrete variables $x \in \mathbf{Z}^n$
- hard to certify optimality



Convex optimization

Convex functions

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \dots, m \end{array}$$



All local minima are global!

Efficient algorithms and software

Extensive theory (convex analysis and conic optimization) [ORF523]

Used to solve non convex problems

Prehistory of optimization

Calculus of variations

Fermat/Newton

minimize $f(x),\ x \in \mathbf{R}$

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = 0$$

1670

Euler

minimize $f(x),\ x \in \mathbf{R}^n$

$$\nabla f(x) = 0$$

1755

Lagrange

minimize $f(x)$

subject to $g(x) = 0$

1797

Time

History of optimization

Algorithms

Origin of linear optimization (Kantorovich, Koopmans)	Simplex algorithm (Dantzig)	Ellipsoid method (Khachyan)	Interior-point method for linear optimization (Karmarkar)	Interior-point methods for convex optimization (Nesterov, Nemirovski)	Large-scale optimization
1930s	1947	1970s	1984	1990s	2000s

History of optimization

Algorithms

1930s	1947	1970s	1984	1990s	2000s
Origin of linear optimization (Kantorovich, Koopmans)	Simplex algorithm (Dantzig)	Ellipsoid method (Khachyan)	Interior-point method for linear optimization (Karmarkar)	Interior-point methods for convex optimization (Nesterov, Nemirovski)	Large-scale optimization

Linear vs Nonlinear

History of optimization

Algorithms

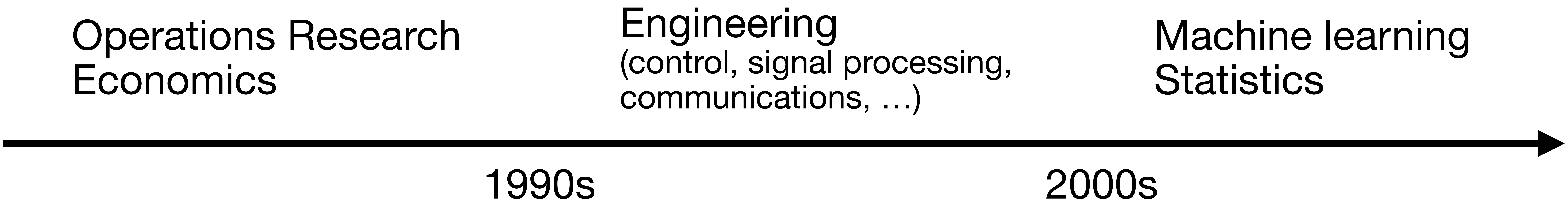
Origin of linear optimization (Kantorovich, Koopmans)	Simplex algorithm (Dantzig)	Ellipsoid method (Khachyan)	Interior-point method for linear optimization (Karmarkar)	Interior-point methods for convex optimization (Nesterov, Nemirovski)	Large-scale optimization
1930s	1947	1970s	1984	1990s	2000s

Linear vs Nonlinear

Convex vs Nonconvex

History of optimization

Applications



What is happening today?

Huge scale optimization

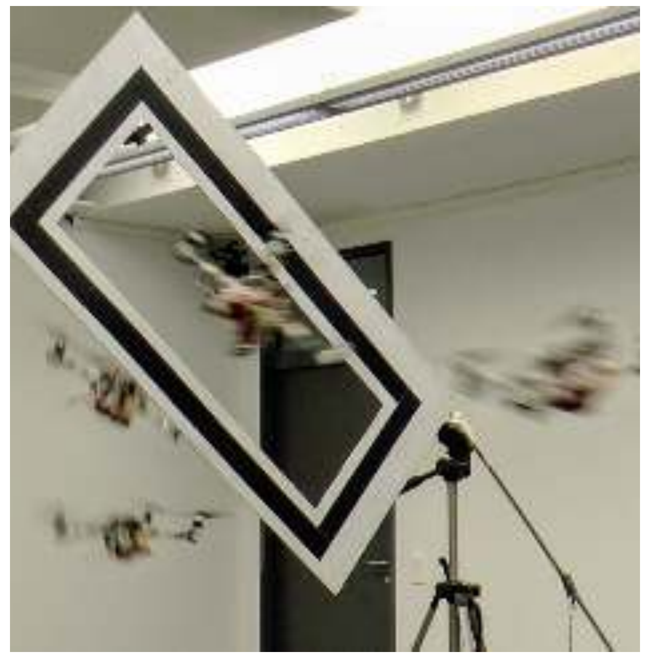
Massive data

+

Massive computations

Real-time optimization

Fast real-time requirements



+

Low-cost computing platforms

What is happening today?

Huge scale optimization

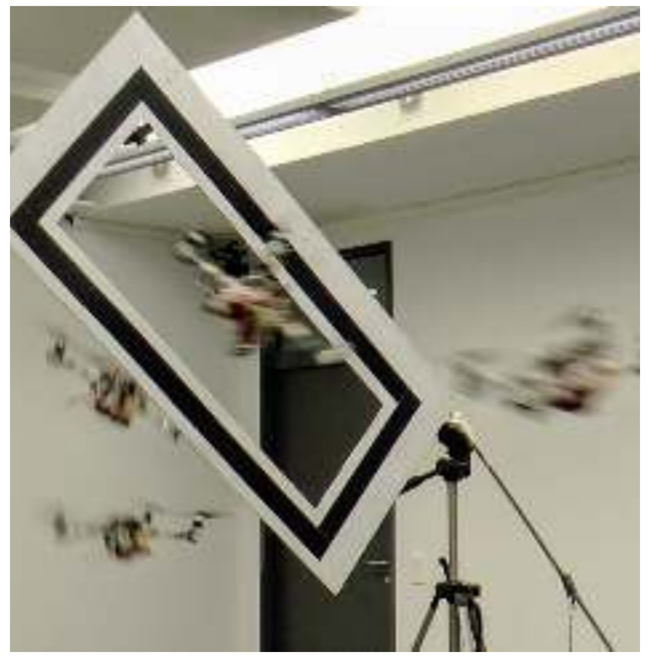
Massive data

+

Massive computations

Real-time optimization

Fast real-time requirements



+

Low-cost computing platforms

Renewed interest in old methods (70s)

- Subgradient methods
- Proximal algorithms

What is happening today?

Huge scale optimization

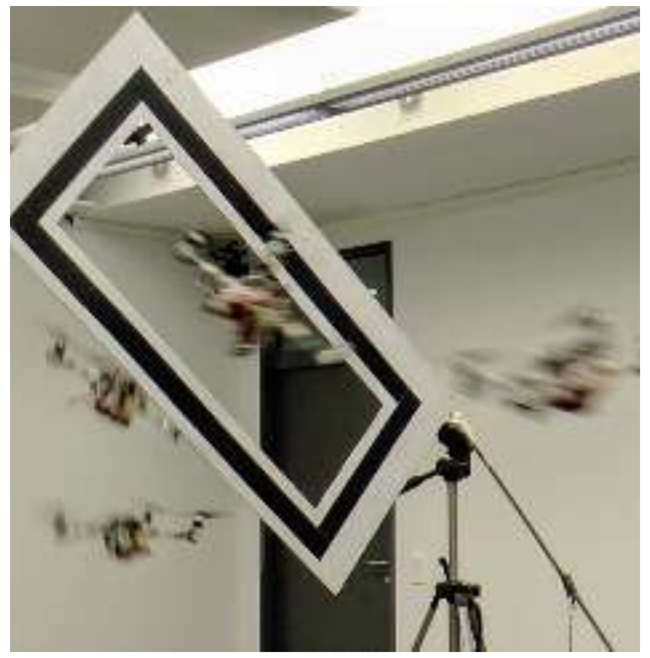
Massive data

+

Massive computations

Real-time optimization

Fast real-time requirements



+

Low-cost computing platforms

Renewed interest in old methods (70s)

- Subgradient methods
- Proximal algorithms

⟶

- Cheap iterations
- Simple implementation

Contents of this course

Linear optimization

- Modelling and applications
- Geometry
- Duality
- Degeneracy
- The simplex method
- Sensitivity analysis
- Interior point methods

Nonlinear optimization

- Modelling and applications
- Optimality conditions
- First-order methods
- Second-order methods

Extensions

- Sequential convex programming
- Branch and bound algorithms
- Data-driven heuristics and algorithm design
- Real-time optimization

Course information

Grading

- **25% Homeworks**
  5 bi-weekly homeworks with coding component. Collaborations are encouraged!

- **25% Midterm**
  90 minutes written exam at home. No collaborations.

- **40% Final**
  Take-home assignment with coding component. No collaborations.

- **10% Participation**
  One question or note on Ed after each lecture.

Course information

10% Participation notes/questions

What?

- Briefly summarize what you learned in the last lecture
- Highlight the concepts that were most confusing/you would like to review.
- Can be anonymous (to your classmates, not to the instructor) or public, as you choose.

Why?

- We will use your ideas to clarify previous lectures, and to improve the course in future iterations.
- You can ask questions you don't feel comfortable asking in class.
- You can use these to gather your thoughts on the previous lecture and solidify your understanding.

Course information

Materials

Prerequisites

- Good knowledge of linear algebra and calculus. For a refresher, we recommend reading Appendices A and C of the S. Boyd and L. Vandenberghe *Convex Optimization* (available **online**).
- Familiarity with Python.

Materials

Lecture slides and readings.

Readings

The following books are useful as reference texts and they are digitally available on Canvas (Reserves):

- R. J. Vanderbei: *Linear Programming: Foundations & Extensions*
- D. Bertsimas, J. Tsitsiklis: *Introduction to Linear Optimization*
- J. Nocedal, S. J. Wright: *Numerical Optimization*

Software (open-source)

CVXPY

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \end{array} \longrightarrow$$

```python
x = cp.Variable(n)
prob = cp.Problem(
    cp.Minimize(c.T@x),
    [A @ x <= b]
)
prob.solve()
print("The optimal value is", prob.value)
print("The solution x is", x.value)
```

Python

Numerical computations on numpy and scipy.

Learning goals

- **Model** your favorite decision-making problems as mathematical optimization problems.
- **Apply** the most appropriate optimization tools when faced with a concrete problem.
- **Implement** optimization algorithms and prove their convergence.

Glance into modern optimization

Huge scale optimization

Dataset with billions of datapoints (x^i, y^i) ⟶ **Goal:** Design predictor $\hat{y}^i = g_\theta(x^i)$

Glance into modern optimization

Huge scale optimization

Dataset with
billions of datapoints (x^i, y^i) ⟶ **Goal:** Design predictor $\hat{y}^i = g_\theta(x^i)$

Optimization problem

minimize $\quad \mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)$

Glance into modern optimization

Huge scale optimization

Dataset with
billions of datapoints (x^i, y^i) ⟶ **Goal:** Design predictor $\hat{y}^i = g_\theta(x^i)$

Optimization problem

Loss

minimize $\mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)$

Glance into modern optimization

Huge scale optimization

Dataset with
billions of datapoints (x^i, y^i) ⟶ **Goal:** Design predictor $\hat{y}^i = g_\theta(x^i)$

Optimization problem

Loss Regularizer

minimize $\mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)$

Glance into modern optimization

Huge scale optimization

Dataset with
billions of datapoints (x^i, y^i) ⟶ **Goal:** Design predictor $\hat{y}^i = g_\theta(x^i)$

Optimization problem

Loss Regularizer

minimize $\mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^{n} \ell(\hat{y}^i, y^i) + \lambda r(\theta)$

Large-scale computing

- Parallel
- Distributed

Glance into modern optimization

Huge scale optimization

Dataset with
billions of datapoints (x^i, y^i) → **Goal:** Design predictor $\hat{y}^i = g_\theta(x^i)$

Optimization problem

Loss Regularizer

minimize $\mathcal{L}(\theta) + \lambda r(\theta) = \sum_{i=1}^n \ell(\hat{y}^i, y^i) + \lambda r(\theta)$

Large-scale computing

- Parallel
- Distributed

Many examples

- Support vector machines
- Regularized regression
- Neural networks

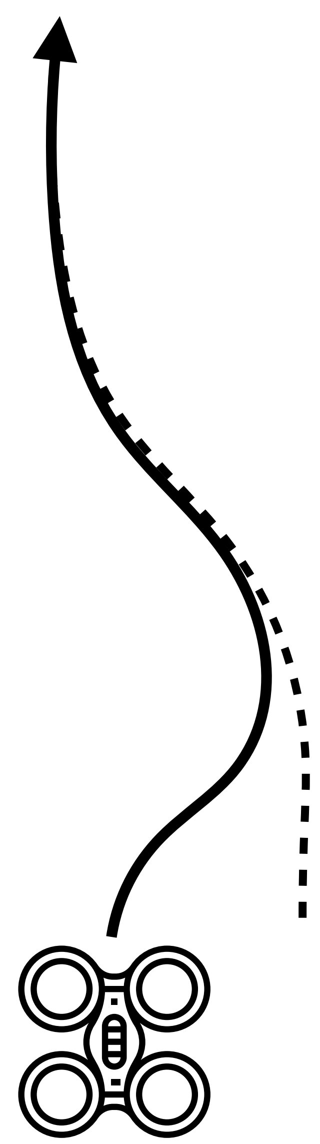
Glance into modern optimization

Real-time optimization

Dynamical system: $x_{t+1} = Ax_t + Bu_t$

$x_t \in \mathbf{R}^n$: state

$u_t \in \mathbf{R}^m$: input



Goal: track trajectory minimize $\sum_t \|x_t - x_t^{\text{des}}\|$

Constraints: inputs $\|u\| \leq U$, states $a \leq x_t \leq b$

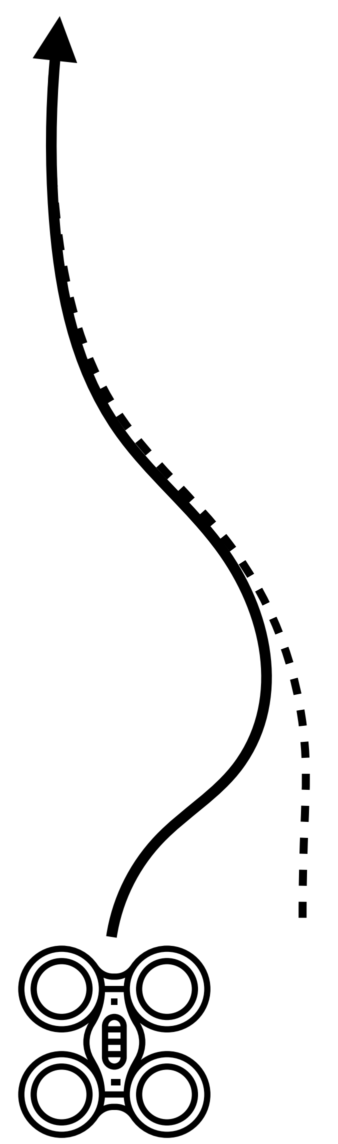
Glance into modern optimization

Real-time optimization

Dynamical system: $x_{t+1} = Ax_t + Bu_t$

$x_t \in \mathbf{R}^n$: state

$u_t \in \mathbf{R}^m$: input



Goal: track trajectory minimize $\sum_t \|x_t - x_t^{\text{des}}\|$

Constraints: inputs $\|u\| \leq U$, states $a \leq x_t \leq b$

1-norm ⟶ ???

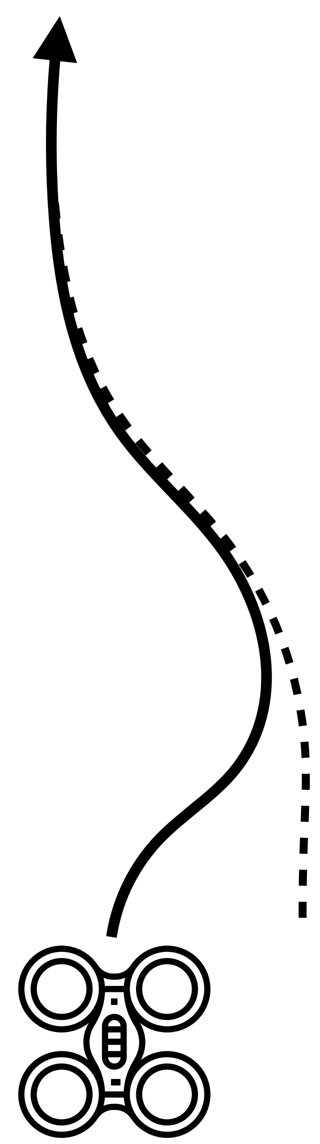
Glance into modern optimization

Real-time optimization

Dynamical system: $x_{t+1} = Ax_t + Bu_t$

$x_t \in \mathbf{R}^n$: state

$u_t \in \mathbf{R}^m$: input



Goal: track trajectory minimize $\sum_t \|x_t - x_t^{\text{des}}\|$

Constraints: inputs $\|u\| \leq U$, states $a \leq x_t \leq b$

1-norm $\longrightarrow$???

∞-norm $\longrightarrow$???

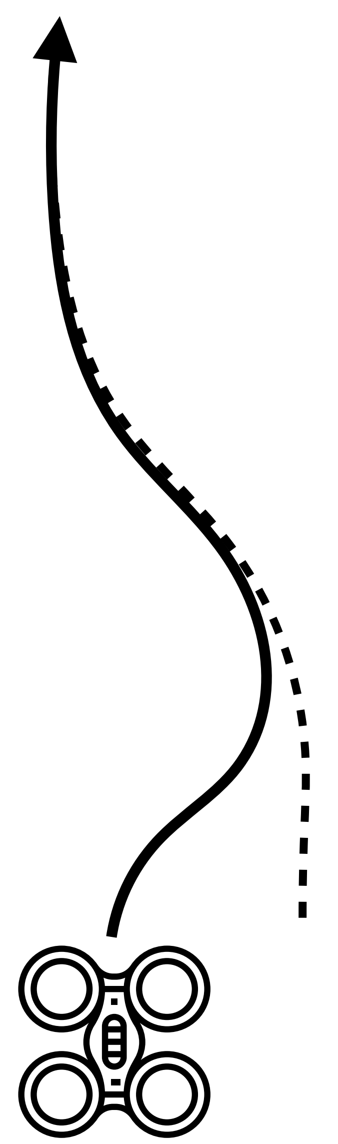
Glance into modern optimization

Real-time optimization

Dynamical system: $x_{t+1} = Ax_t + Bu_t$

$x_t \in \mathbf{R}^n$: state

$u_t \in \mathbf{R}^m$: input



Goal: track trajectory minimize $\sum_t \|x_t - x_t^{\rm des}\|$

Constraints: inputs $\|u\| \leq U$, states $a \leq x_t \leq b$

1-norm
∞-norm → Linear optimization
(more next lecture…)

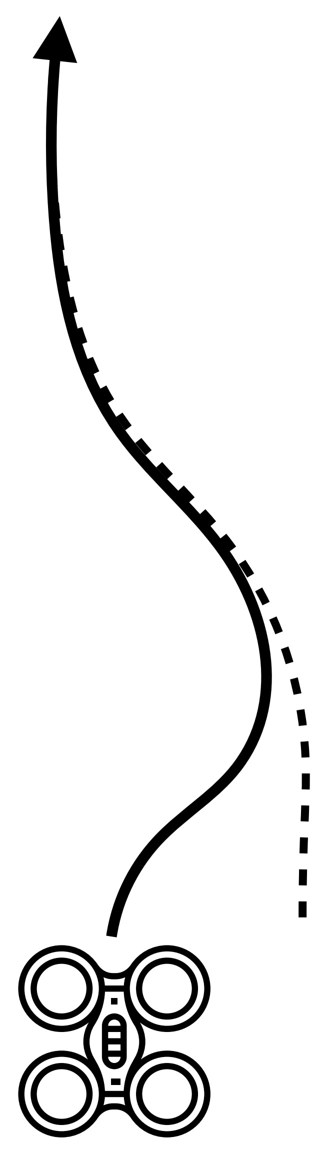
Glance into modern optimization

Real-time optimization

Dynamical system: $x_{t+1} = Ax_t + Bu_t$

$x_t \in \mathbf{R}^n$: state

$u_t \in \mathbf{R}^m$: input



Goal: track trajectory minimize $\sum_t \|x_t - x_t^{\text{des}}\|$

Constraints: inputs $\|u\| \leq U$, states $a \leq x_t \leq b$

1-norm
∞-norm
⟶ Linear optimization
(more next lecture…)

Solve and repeat…..

How fast can we solve these problems?

Next lecture

Linear optimization

- Definitions
- Modelling
- Formulations
- Examples