ORF522 – Linear and Nonlinear Optimization

23. Data-driven Decision-Making and the Role of Optimization

Bartolomeo Stellato — Fall 2024

Today’s lecture

[ISA, Ch 1 and 3]

Data-driven optimization under uncertainty

- Scenario approach
- Portfolio optimization

What’s left out there?

The role of optimization

Data-driven robust optimization

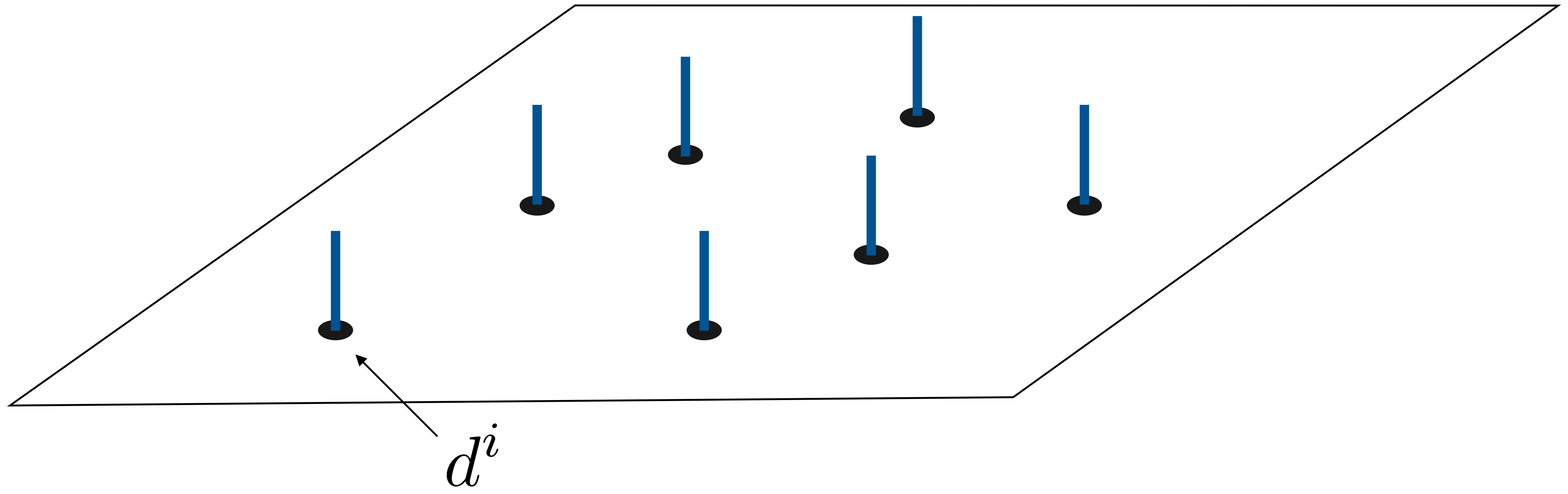
Issues with traditional uncertainty set construction

Probability distributions $\mathbf{P}$ are never observed in practice

Data is observed in practice

All we have is the empirical distribution

$$\hat{\mathbf{P}}^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{d^i}$$



Are there better ways to model the uncertainty
that still lead to tractable formulations?

Using data to build uncertainty sets

Let $u \in \mathbf{R}^p$ be a random variable, we write a *chance constrained program* as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \mathbf{P}(g(u,x) \leq 0) \geq 1 - \epsilon \end{array}$$

dataset

$$D = \{d^i\}_{i=1}^N$$

how can we build uncertainty sets?

data-driven probabilistic guarantees

$$\mathbf{P}^N(\mathbf{P}(g(u, x^\star_D) \leq 0) \geq 1 - \epsilon) \geq 1 - \beta$$

product distribution

data-driven solution

probability of constraint satisfaction

Scenario approach

Direct sample approximation

discretized uncertainty set

$$\begin{array}{ll}\text{minimize} & f(x)\\ \text{subject to} & g(u,x) \leq 0, \quad \forall u \in \mathcal{U} = D = \{d^i\}_{i=1}^N\end{array}$$

scenario reformulation

sampled problem

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

→

randomized solution

$x_D^\star$

benefits

- no assumptions on distribution
- tractable if not too many samples

challenges

- random reformulation
- solution can violate chance constraints

Data-driven probabilistic guarantees

Assumptions

- Convex problem
- Unique solution

$$\mathbf{P}^N(\mathbf{P}(g(u, x_D^\star) \leq 0) \geq 1 - \epsilon) \geq 1 - \sum_{i=1}^{n-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$$

cumulative distribution function of beta distribution $\mathcal{B}(n, N-n+1)$

How many samples do we need to have at least $1 - \beta$?

Introduction to the scenario approach, Campi, Garatti

Sample requirements

$$
\begin{aligned}
\sum_{i=0}^{n-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} &= 2^{n-1} \sum_{i=0}^{n-1} \binom{N}{i} \frac{\epsilon^i}{2^{n-1}} (1-\epsilon)^{N-i} \\
&\le 2^{n-1} \sum_{i=0}^{n-1} \binom{N}{i} \left(\frac{\epsilon}{2}\right)^i (1-\epsilon)^{N-i} \\
&\le 2^{n-1} \sum_{i=0}^{N} \binom{N}{i} \left(\frac{\epsilon}{2}\right)^i (1-\epsilon)^{N-i} \\
&= 2^{n-1} \left(\frac{\epsilon}{2} + (1-\epsilon)\right)^N \\
&= 2^{n-1} \left(1 - \frac{\epsilon}{2}\right)^N \\
&\le 2^{n-1} \exp\left(-\frac{\epsilon}{2} N\right)
\end{aligned}
$$

binomial theorem →

↑ $(1-\epsilon/2) \le \exp(-\epsilon/2)$

Therefore

$$2^{n-1} \exp\left(-\frac{\epsilon}{2} N\right) \le \beta$$

$$\iff \quad (n-1)\log 2 - \frac{\epsilon}{2} N \le \log \beta$$

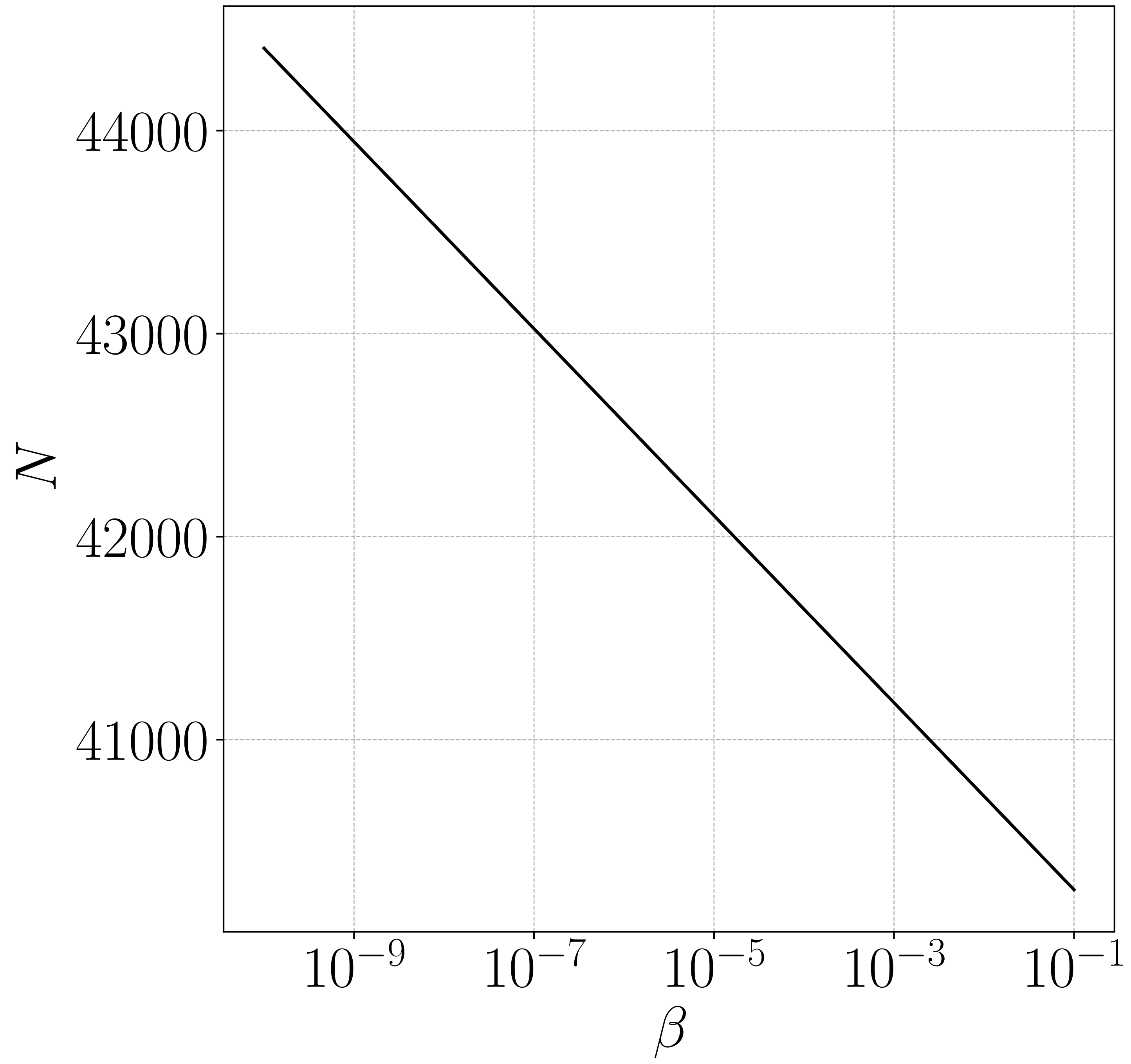
$$\iff \quad N \ge \frac{2}{\epsilon}\left(\log\frac{1}{\beta} + (d-1)\log 2\right)$$

$(\log(2) < 1)$ $$\iff \quad N \ge \frac{2}{\epsilon}\left(\log\frac{1}{\beta} + n - 1\right)$$

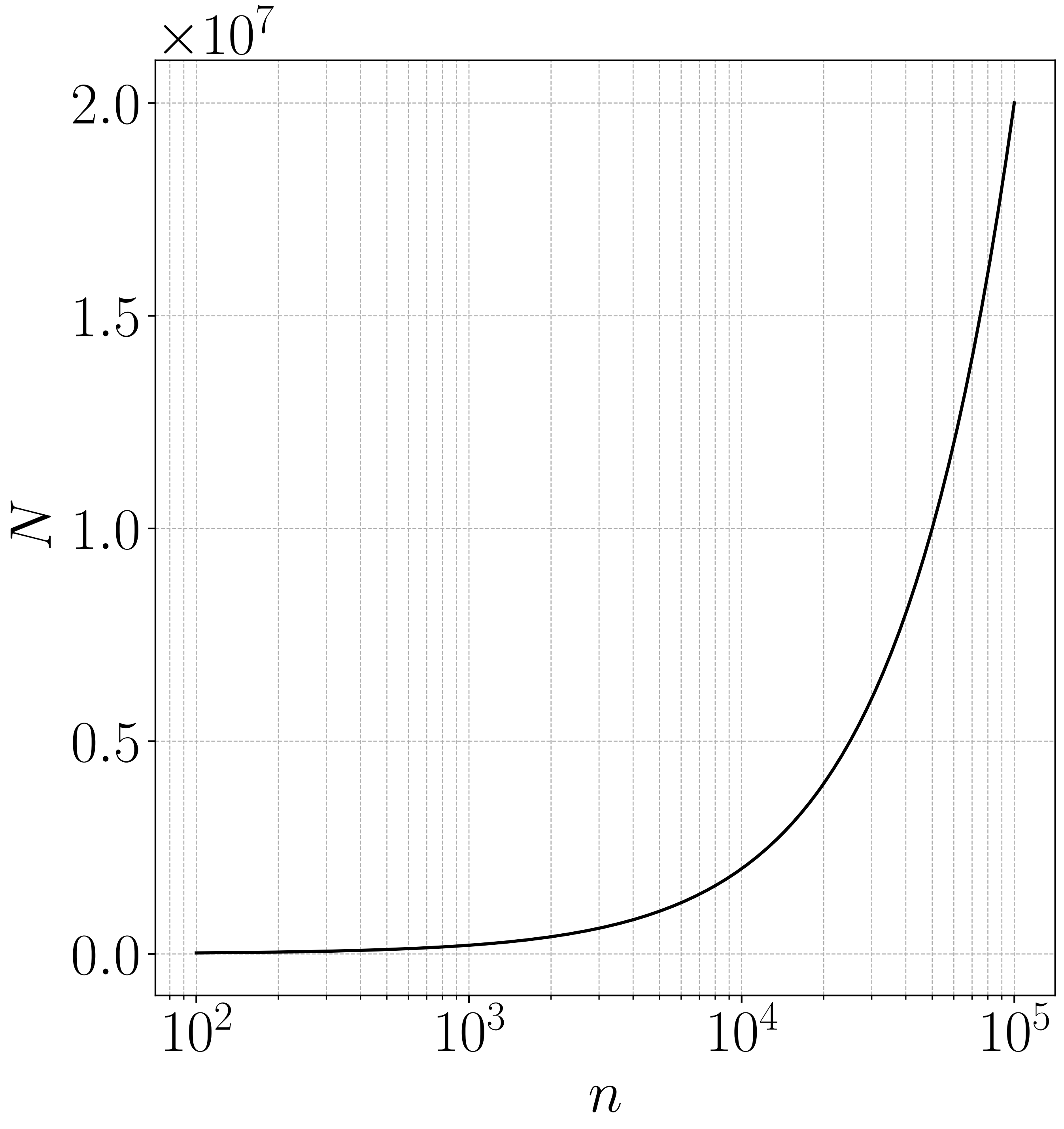
■

Number of samples required

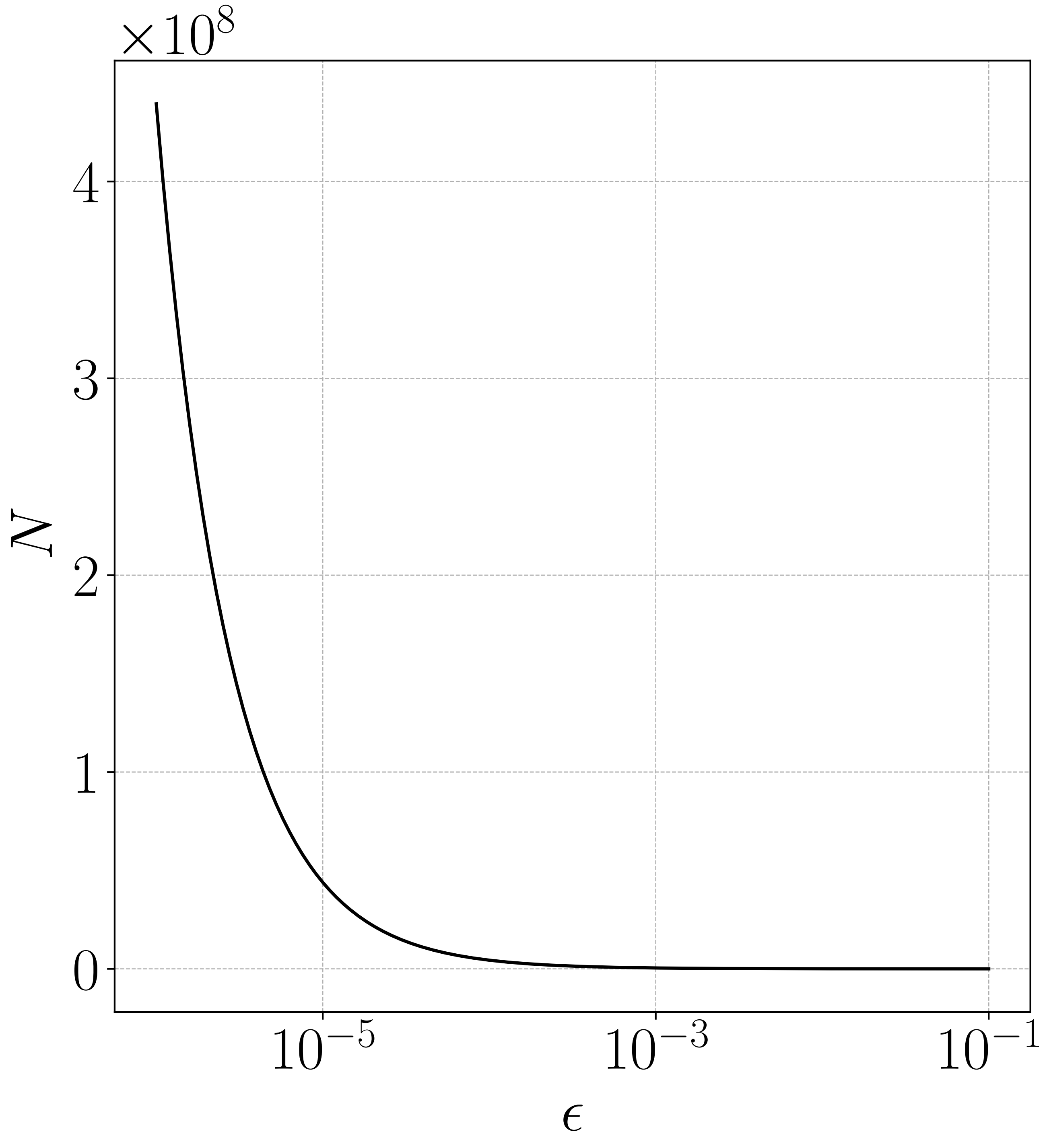
$$N \geq \frac{2}{\epsilon}\left(\log\left(\frac{1}{\beta}\right) + n - 1\right)$$



good



bad



ugly

Portfolio optimization example

asset allocations $x \in \mathbf{R}^n$

uncertain returns $r \in [\ell, u]$ with mean μ

optimization problem

$$\begin{array}{ll} \text{maximize} & \mu^T x \\ \text{subject to} & \mathbf{P}(r^T x \leq \alpha) \leq \epsilon \\ & \mathbf{1}^T x = 1 \\ & x \geq 0 \end{array}$$

expected returns

loss risk constraint

unwanted return level

scenario approach reformulation

$$\begin{array}{ll} \text{maximize} & \mu^T x \\ \text{subject to} & (r^i)^T x \geq \alpha, \quad i = 1, \dots, N \\ & \mathbf{1}^T x = 1 \\ & x \geq 0 \end{array}$$

Portfolio optimization strategies

returns

$$\mu_i = 1.05 + \frac{3(n-i)}{10n} \quad (\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n)$$

$$|r_i - \mu_i| \leq u_i = 0.05 + \frac{n-i}{2n}, \qquad u_n = 0$$

n-th asset is cash
(guaranteed 5% return)

baselines

- nominal minimizer $x^\star_{\text{nom}} = e_1$
- conservative minimizer $x^\star_{\text{con}} = e_n$
- robust minimizer $x^\star_{\text{ro}}$
- scenario minimizer $x^\star_{\text{sce}}$

Portfolio optimization results

$$\mathbf{P}(r^T x \leq \overset{1.1}{\overset{\downarrow}{\alpha}}) \leq \overset{10^{-2}}{\overset{\swarrow}{\epsilon}}$$

scenario approach samples

$$\beta = 10^{-9},\ n = 200 \ \Rightarrow N = 43,945$$

violation probability

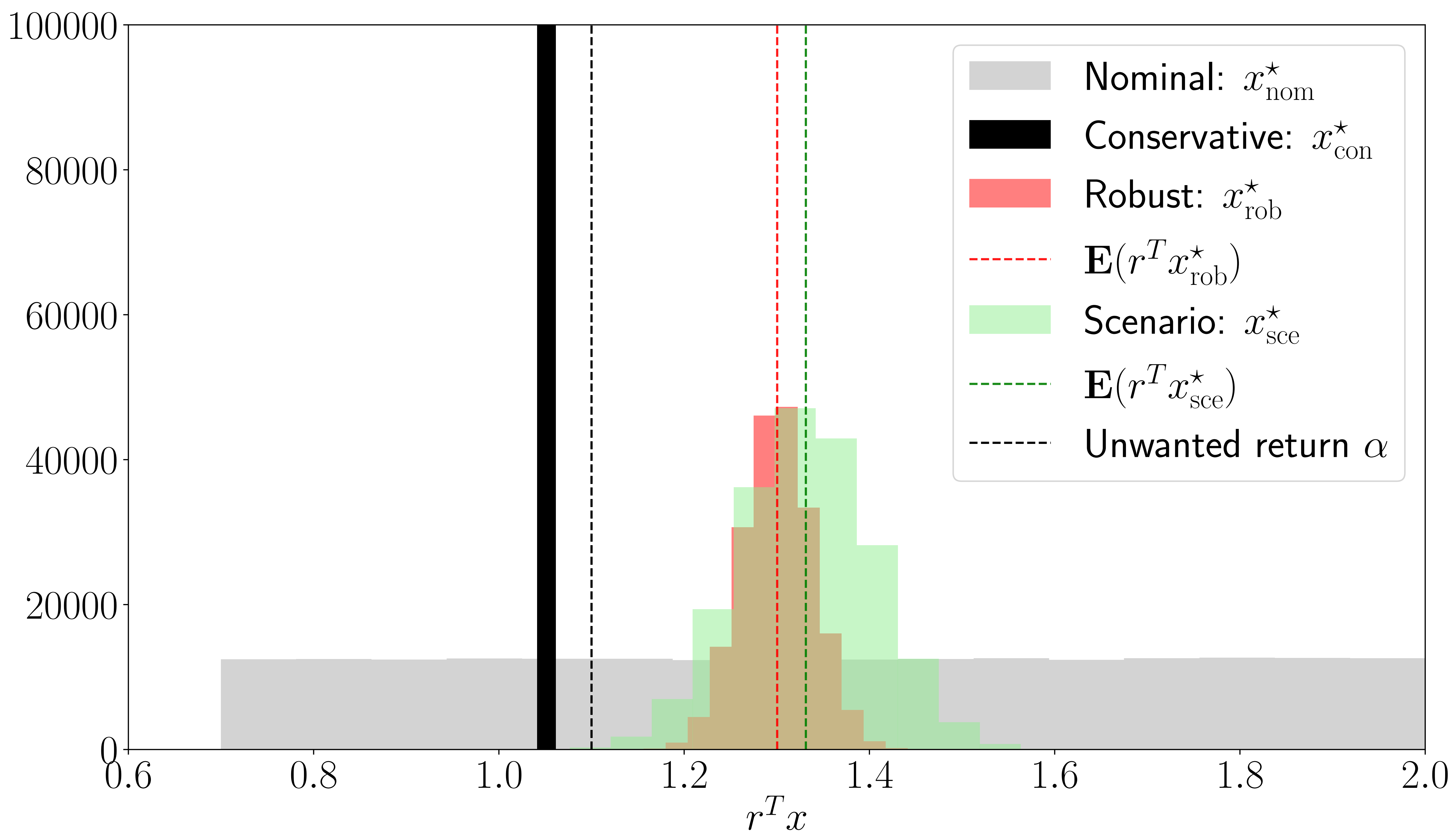
$x^\star_{\text{rob}}$	0.0
$x^\star_{\text{sce}}$	0.00073
ϵ	0.01

→ scenario approach is less conservative

solution time

$x^\star_{\text{rob}}$	1.6 ms
$x^\star_{\text{sce}}$	41.8 s

→ scenario approach is computationally much more expensive



Issues with scenario approach

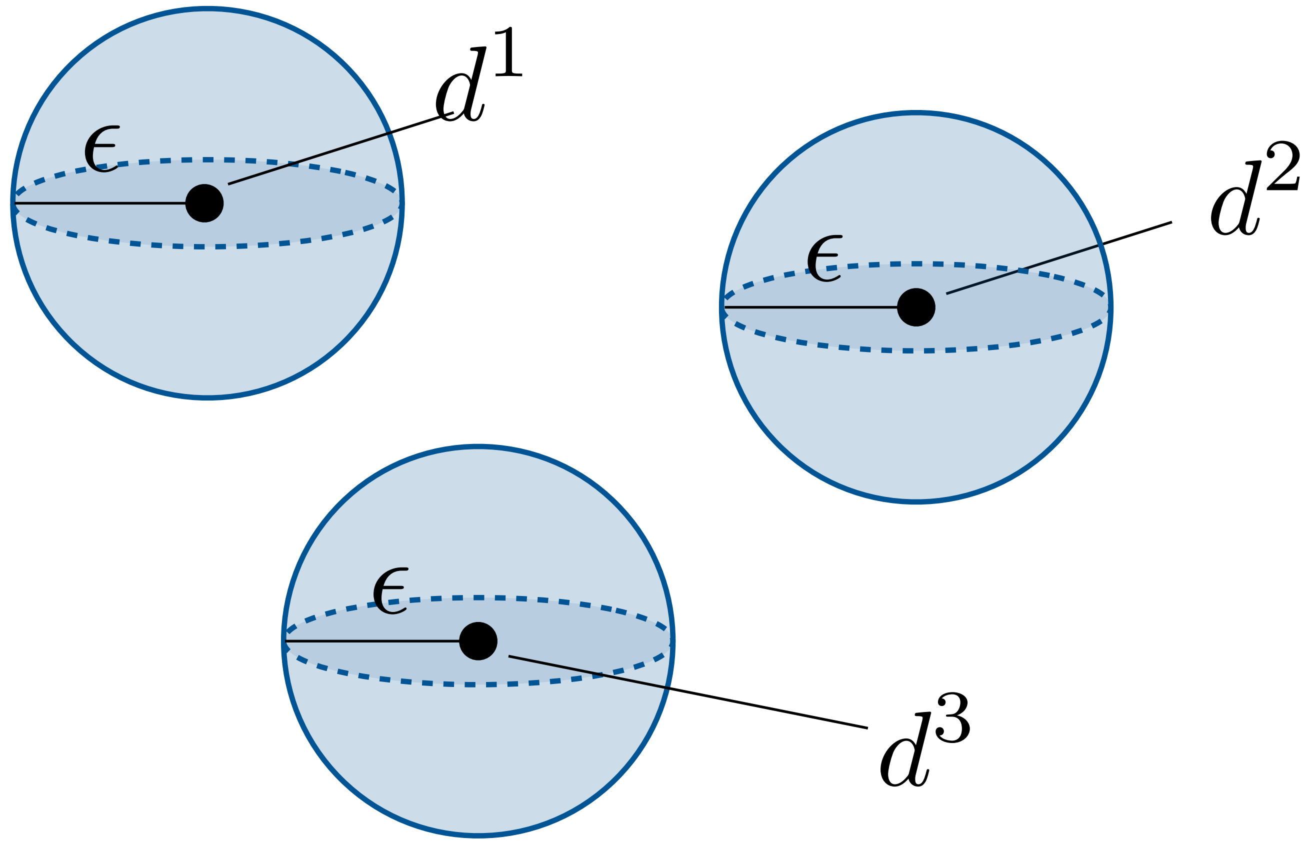
- Requires *many* samples
  - Solvers may require a long time to give a solution
  - You may not be able to sample so much

    (e.g., in finance you get a sample per trading day ~250/year. We would need 200 years of trading history to collect 50,000 samples)

↓

can we construct sets that
require fewer samples?

Allowing perturbations around samples



$$\mathcal{U} = \{u = (v^1, \dots, v^N) \mid \|v^i - d^i\| \le \epsilon, \quad i = 1, \dots, N\}$$

We can pick ϵ to ensure probabilistic guarantees *(with light-tailed distributions)*

$$\mathbf{P}^N(\mathbf{P}(g(u, x^\star_D) \le 0) \ge 1 - \epsilon) \ge 1 - \beta$$

$$\epsilon_N(\beta) = \begin{cases} \left(\frac{\log(c_1/\beta)}{c_2 N}\right)^{1/p} & N \ge \frac{\log(c_1/\beta)}{c_2} \\ \left(\frac{\log(c_1/\beta)}{c_2 N}\right)^{1/a} & N < \left(\frac{\log(c_1/\beta)}{c_2}\right) \end{cases}$$

active research area!

A Data-Driven Approach to Multistage Stochastic Linear Optimization, Bertsimas, Shtern, Sturt

Mean Robust Optimization, Wang, Becker, Van Parys, Stellato

What’s left out there?

What we did not cover in nonlinear optimization

Second-order methods: High accuracy on small/medium-scale data

- Newton’s method
- Quasi-Newton (BFGS, L-BFGS)
- Interior-point methods for nonlinear optimization (IPOPT)

Stochastic gradient methods

- Stochastic gradient descent
- Variance reduction methods
- Deep learning optimizers

→ Covered in
ELE539: Optimization for Machine Learning

Optimization in data science

- Compressed sensing
- Matrix completion
- Many more…

→ covered in
ORF525: Statistical Foundations of Data Science

What we did not cover in convex optimization?

More in details on convex analysis

Conic optimization

- Second-order cone programming
- Semidefinite programming
- Sum-of-squares optimization

→ Covered in ORF523: Convex and Conic Optimization

Convex relaxations of NP-hard problems

The role of optimization

Optimization as a surrogate for real goal

Very often, optimization is not the actual goal

The goal usually comes from practical implementation (new data, real dynamics, etc.)

Real goal is usually encoded (approximated) in cost/constraints

Optimization problems are just models

> “All models are wrong, some are useful.”
>
> — George Box

Implications

- Problem formulation does not need to be “accurate”
- Objective function and constraints “guide” the optimizer
- The model includes parameters to tune

We often do not need to solve most problems to extreme accuracy

Portfolio

Optimization problem

$$\begin{array}{ll}\text{maximize} & \mu^T x - \gamma x^T \Sigma x \\ \text{subject to} & \mathbf{1}^T x = 1 \\ & x \geq 0\end{array}$$

Goal
Optimize backtesting performance

Uncertain returns

p_t random variable:
mean μ, covariance Σ

→

Backtesting performance

(sum over all past realizations)

- Total returns
- Cumulative risk (quadratic term)

Model fitting

Training data

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N \longrightarrow$$

Optimization problem

$$\underset{w}{\text{minimize}} \quad f_{\text{train}}(w) = \sum_{(x_i, y_i) \in \mathcal{D}_{\text{train}}} \ell(y_i, h_w(x_i))$$

Goal
Optimize test performance

Test data
(unknown)

$$\mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=1}^N \longrightarrow$$

Test performance

$$f_{\text{test}}(w) = \sum_{(x_i, y_i) \in \mathcal{D}_{\text{test}}} \ell(y_i, h_w(x_i))$$

Control

Optimization problem
(control policy)

→

$$\begin{aligned} \phi(\bar{x}) = \quad & \underset{u}{\text{argmin}} && \sum_{t=0}^{T-1} \ell(x_t, u_t) \\ & \text{subject to} && x_{t+1} = f(x_t, u_t) \\ &&& x_0 = \bar{x} \\ &&& x_t \in \mathcal{X}, \quad u_t \in \mathcal{U} \end{aligned}$$

Goal:
Optimize closed-loop performance

Real dynamics

$x_{t+1} = f(x_t, u_t, w_t)$
w_t uncertainty

→

Control input

$u_t = \phi(x_t)$

→

Closed-loop performance

$$J = \sum_{t=0}^{\infty} \ell(x_t, u_t)$$

Quadcopter control

Low accuracy works well

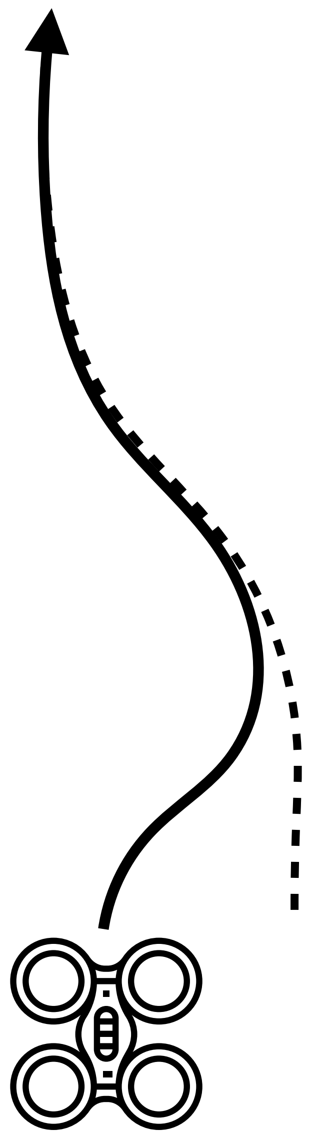
Quadcopter example

Linearized dynamics $x_{t+1} = Ax_t + Bu_t + w_t$

$x_t \in \mathbf{R}^{12}, \quad u_t \in \mathbf{R}^4$

Input and state constraints

$x_t \in [\underline{x}, \overline{x}], \quad u_t \in [\underline{u}, \overline{u}]$



Goal: track trajectory minimize $\sum_t \|x_t - x_t^{\text{des}}\|_2^2 + \gamma \|u_t\|_2^2$

Closed loop simulation

Simulated dynamics $x_{t+1} = Ax_t + Bu_t + w_t$

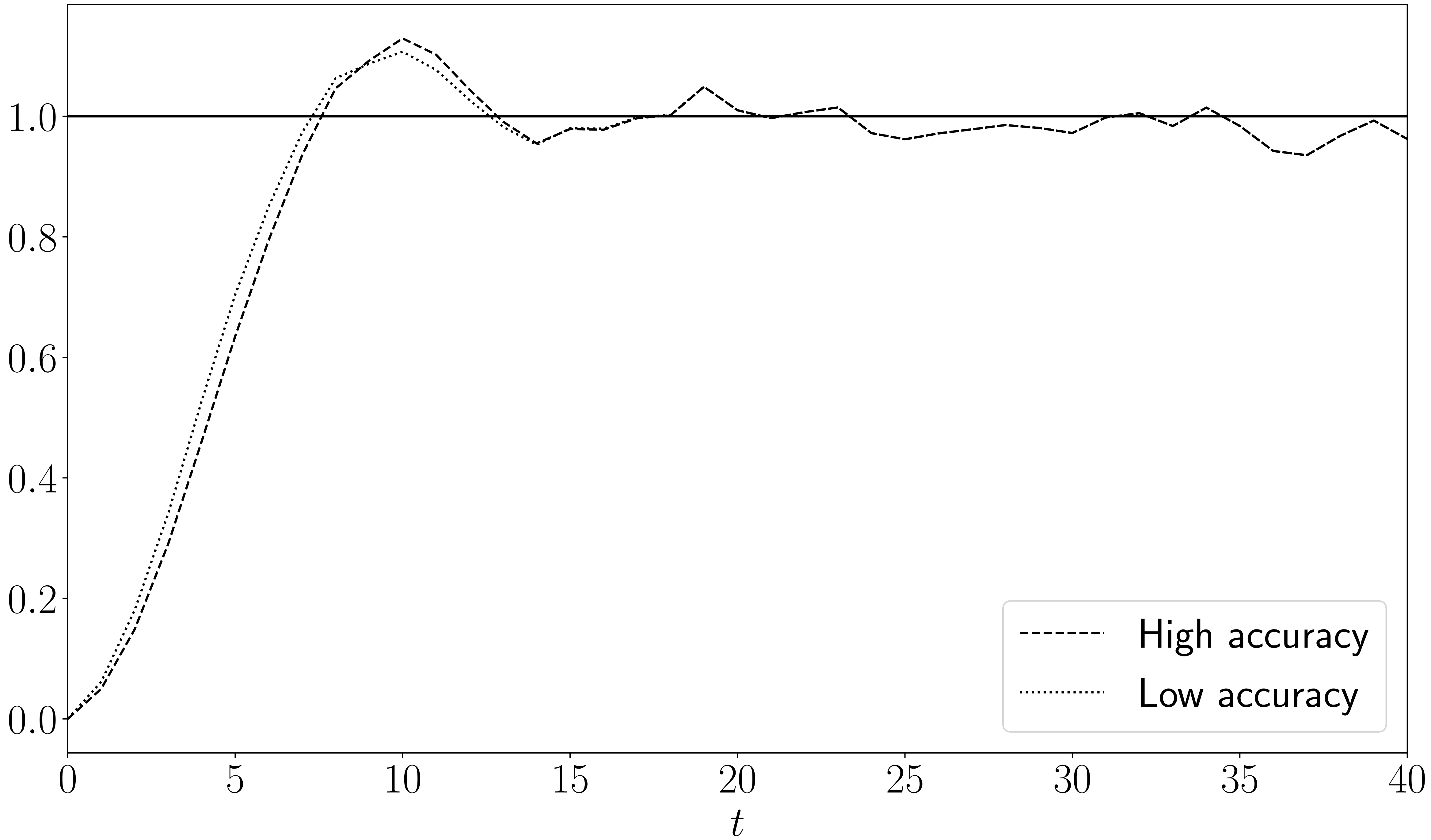
random variable (nonlinearities, disturbances, etc.)

Quadcopter control

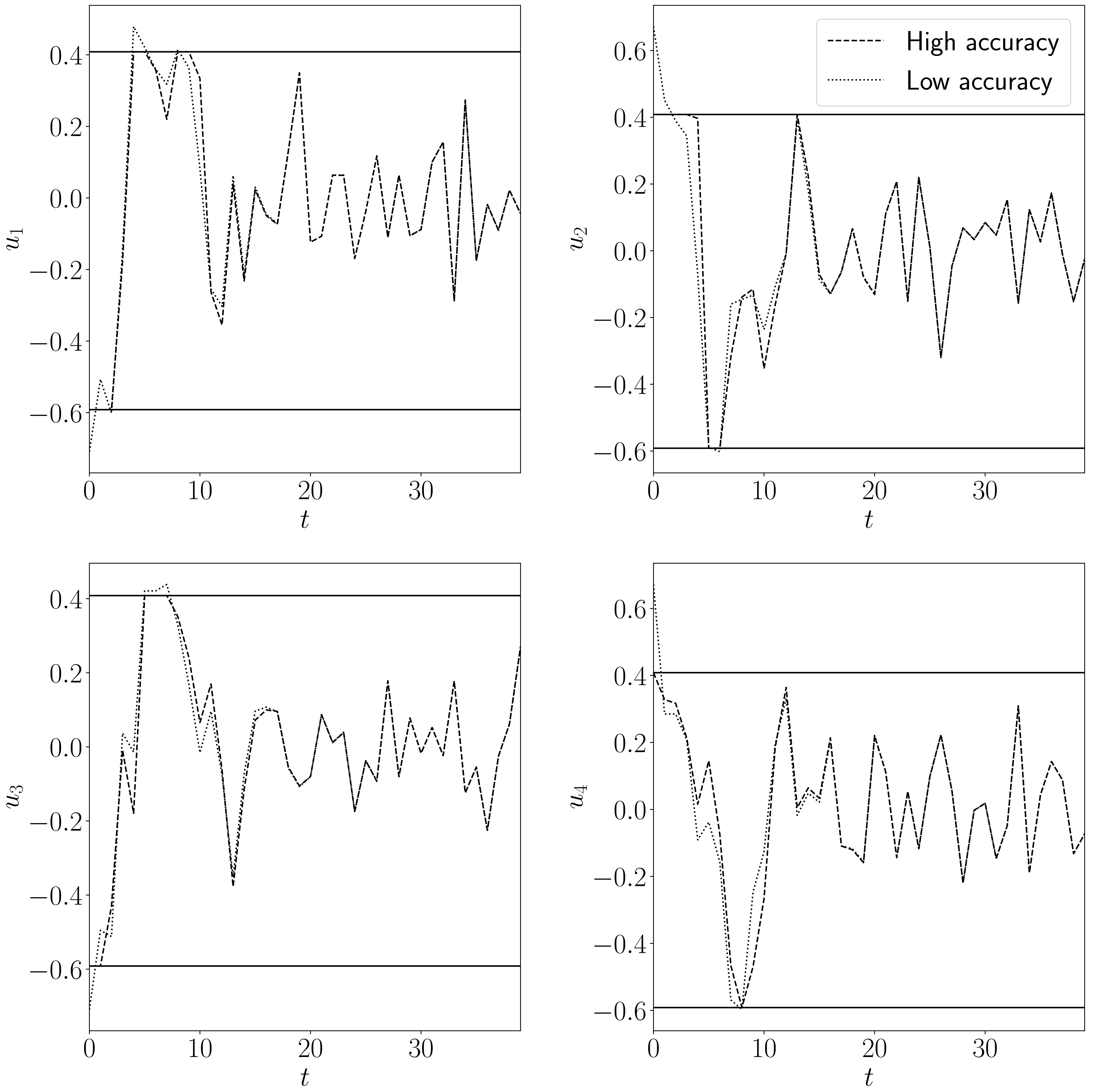
Closed-loop behavior with OSQP solver

- Low accuracy: $\epsilon = 0.1$
- High accuracy: $\epsilon = 0.0004$

Altitude reference tracking



Control effort



Conclusions

In ORF522, we learned to:

- **Model decision-making problems** across different disciplines as mathematical optimization problems.
- **Apply the most appropriate optimization tools** when faced with a concrete problem.
- **Implement** optimization algorithms and **prove** their **convergence.**
- **Understand** the limitations of optimization

Optimization cannot solve all our problems

It is just a mathematical model

But it can help us making better decisions

Thank you!

Bartolomeo Stellato