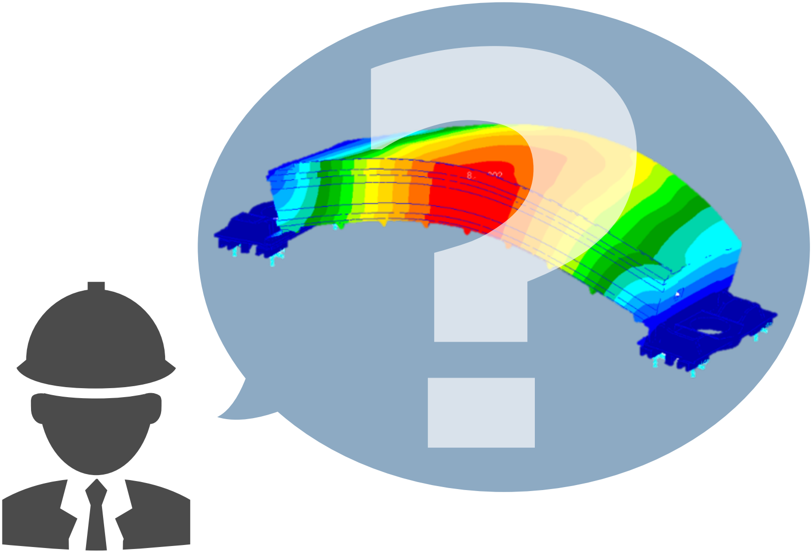

UNCERTAINTY
ROBUST OPTIMIZATION

Uncertainty in Finance

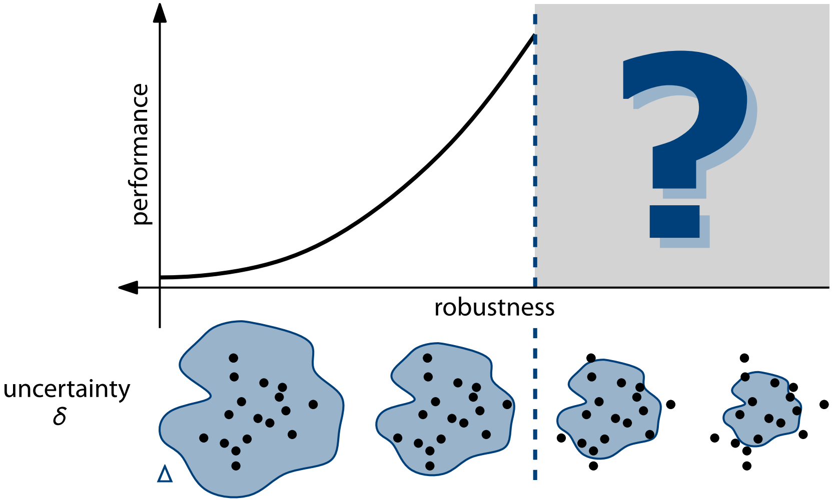
Uncertainty in Finance

Uncertainty in Engineering

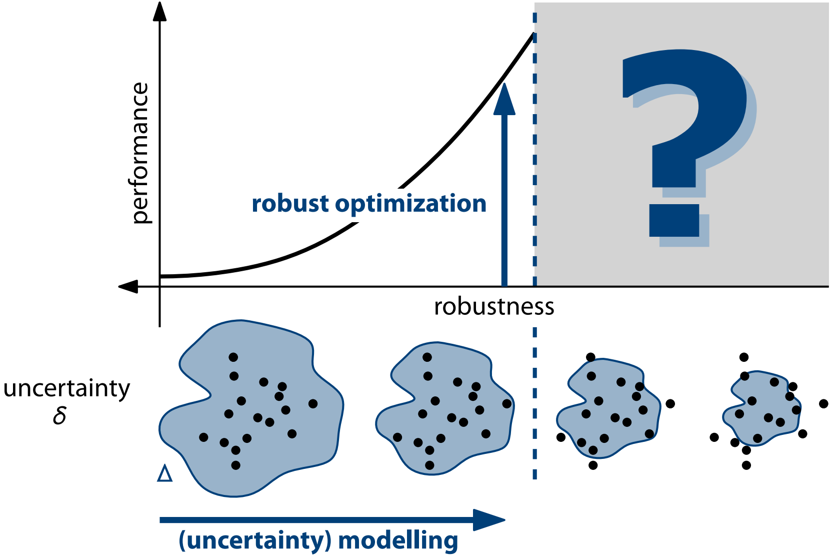
Uncertainty in Engineering



The General Picture



The General Picture



Robust Optimization

Find the solution (portfolio, control, design …),

that is adequate for all possible δ in Δ,

and guarantees the highest performance.

Robust Optimization

Find the solution (portfolio, control, design …),

$\hookrightarrow$ optimization variable x

that is adequate for all possible δ in Δ,

and guarantees the highest performance.

Robust Optimization

Find the solution (portfolio, control, design …),

$\hookrightarrow$ optimization variable x

that is adequate for all possible δ in Δ,

$\hookrightarrow$ constraints $f_i(x, \delta) \geq 0$, for all $\delta \in \Delta$, $i = 1, \ldots, m$

and guarantees the highest performance.

Robust Optimization

Find the solution (portfolio, control, design ...),

$\hookrightarrow$ optimization variable x

that is adequate for all possible δ in Δ,

$\hookrightarrow$ constraints $f_i(x, \delta) \geq 0$, for all $\delta \in \Delta$, $i = 1, \ldots, m$

and guarantees the highest performance.

$\hookrightarrow$ maximize γ subject to $f_0(x, \delta) \geq \gamma$, for all $\delta \in \Delta$

Robust Optimization

Find the solution (portfolio, control, design …),

$\hookrightarrow$ optimization variable x

that is adequate for all possible δ in Δ,

$\hookrightarrow$ constraints $f_i(x, \delta) \geq 0$, for all $\delta \in \Delta$, $i = 1, \ldots, m$

and guarantees the highest performance.

$\hookrightarrow$ maximize γ subject to $f_0(x, \delta) \geq \gamma$, for all $\delta \in \Delta$

$$
\begin{array}{ll}
\underset{x,\gamma}{\text{maximize}} & \gamma \\
\text{subject to} & f_0(x, \delta) \geq \gamma, \quad \text{for all } \delta \in \Delta \\
& f_i(x, \delta) \geq 0, \quad \text{for all } \delta \in \Delta, \\
& \qquad\qquad\qquad\;\; i = 1, \ldots, m
\end{array}
$$

1. *Beyond the original raison d'être*
2. *Solving robust optimization problems*
3. *Results in applications*

Motion Planning

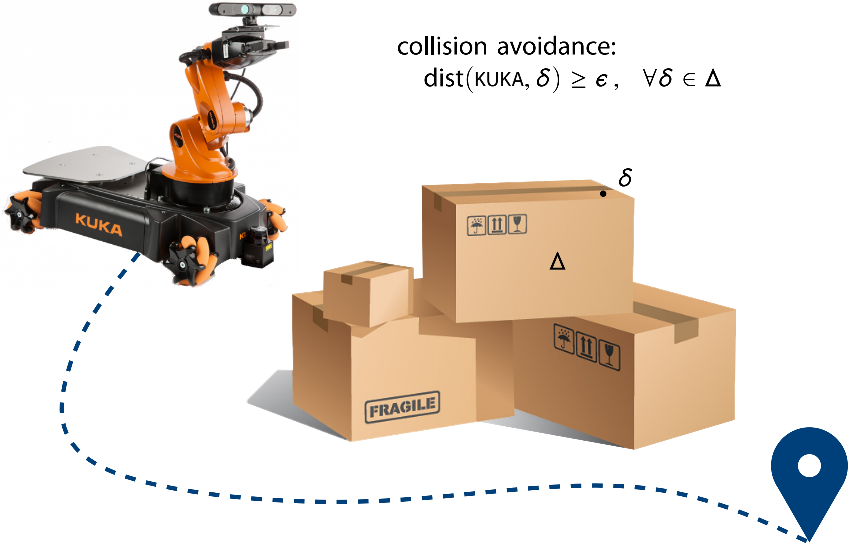
Motion Planning

Motion Planning

Motion Planning

collision avoidance:

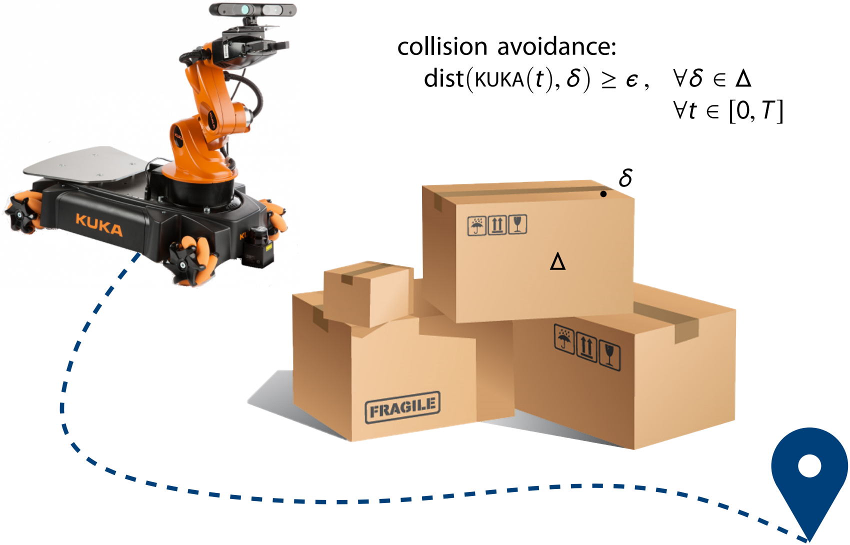
$$\mathrm{dist}(\text{KUKA}, \delta) \geq \epsilon\,, \quad \forall \delta \in \Delta$$



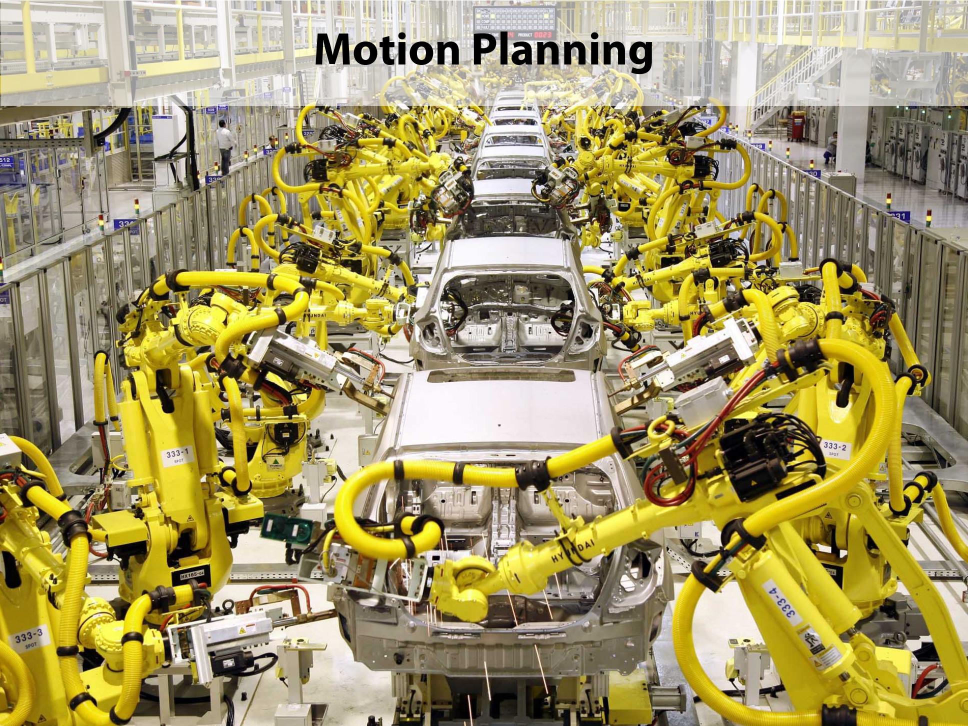
Motion Planning

collision avoidance:

$$\mathrm{dist}(\mathsf{KUKA}(t), \delta) \geq \epsilon\,, \quad \forall \delta \in \Delta$$
$$\forall t \in [0, T]$$



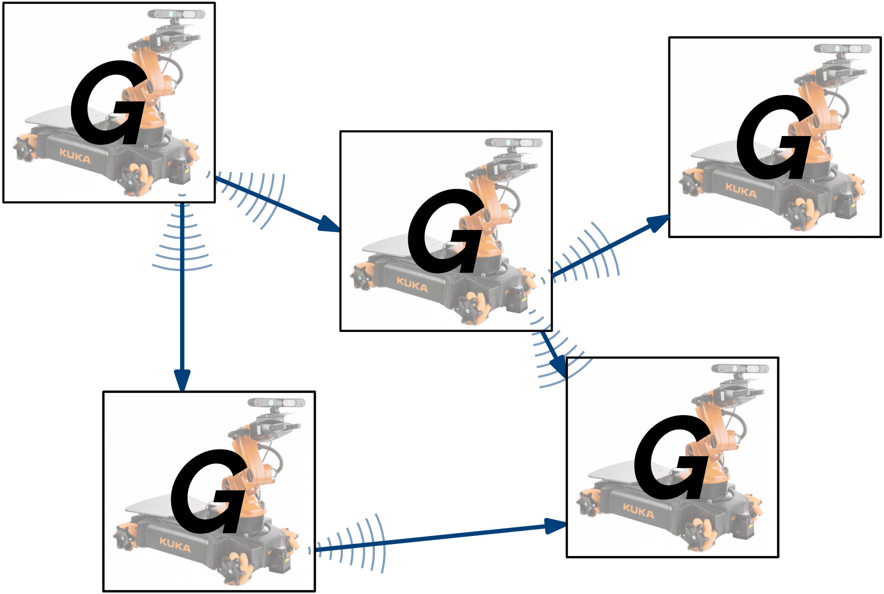
Motion Planning



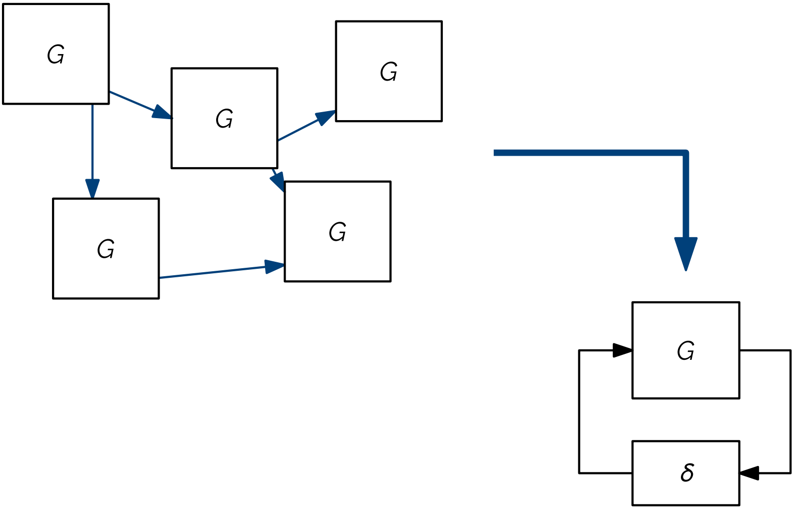
Interconnected Systems

Interconnected Systems

Interconnected Systems



Interconnected Systems



$\Delta \sim$ interconnection

Interconnected Systems

Parametric Programming

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f_0(x, \delta) \\ \text{subject to} \quad & f_i(x, \delta) \geq 0\,, \quad i = 1, \ldots, m \end{aligned}$$

$\longrightarrow$

$x^{\mathrm{opt}}(\delta)$
for $\delta \in \Delta$

Parametric Programming

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f_0(x, \delta) \\ \text{subject to} \quad & f_i(x, \delta) \geq 0\,, \quad i = 1, \ldots, m \end{aligned}$$

$\longrightarrow$

$x^{\text{opt}}(\delta)$
for $\delta \in \Delta$

Trade-off analysis

$\hookrightarrow$ δ: weight on objectives

$$f_0(x, \delta) = \sum_k \delta_k \, f_{0,k}(x)$$

Parametric Programming

$$\begin{array}{lll} \underset{x}{\text{maximize}} & f_0(x,\delta) & \\ \text{subject to} & f_i(x,\delta) \geq 0\,, & i = 1,\ldots,m \end{array} \longrightarrow \begin{array}{c} x^{\text{opt}}(\delta) \\ \text{for } \delta \in \Delta \end{array}$$

Linear parameter varying control

$\hookrightarrow$ δ: measurable parameters affecting the system dynamics

Parametric Programming

$$\begin{array}{ll} \underset{x}{\text{maximize}} & f_0(x,\delta) \\ \text{subject to} & f_i(x,\delta) \geq 0\,, \quad i = 1,\ldots,m \end{array}$$

$$\longrightarrow$$

$x^{\text{opt}}(\delta)$
for $\delta \in \Delta$

Computing approximation $\hat{x}(\delta)$ of $x^{\text{opt}}(\delta)$

$$\hat{x}(\delta) = x_0 + x_1\,\delta + x_2\,\delta^2 \ldots$$

$\hookrightarrow$ solve robust optimization problem to compute coefficients

$$\begin{array}{ll} \underset{x_0,\,x_1,\,\ldots}{\text{maximize}} & \displaystyle\int_\Delta f_0(\hat{x}(\delta),\delta)\,d\delta \\ \text{subject to} & f_i(\hat{x}(\delta),\delta) \geq 0\,, \quad \text{for all } \delta \in \Delta\,, \\ & \qquad\qquad\qquad\qquad i = 1,\ldots,m \end{array}$$

1. ***Beyond the original raison d’être***

2. ***Solving robust optimization problems***

3. ***Results in applications***

The Main Challenge

$$f(x, \delta) \geq 0, \text{ for all } \delta \in \Delta$$

The Main Challenge

$$f_x(\delta) \geq 0, \;\; \text{for all } \; \delta \in \Delta$$

The Main Challenge

$$f_x(\delta) \geq 0\,, \text{ for all } \delta \in \Delta$$

$$\Updownarrow$$

there exists no δ in Δ for which $f_x(\delta) < 0$

The Main Challenge

$$f_x(\delta) \geq 0\,, \ \text{for all } \ \delta \in \Delta$$

$$\Updownarrow$$

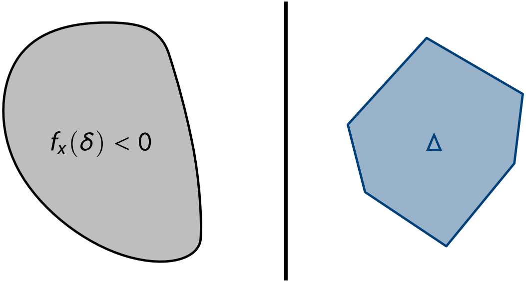
there exists no δ in Δ for which $f_x(\delta) < 0$

$$\Updownarrow$$

$$\{\delta \,|\, f_x(\delta) < 0\} \cap \Delta = \varnothing$$

Few Cases: Tractable Reformulation

Linear separation



Quadratic separation

- S-procedure, Kalman-Yakubovich-Popov lemma …

Otherwise: Relaxations

$$f(x,\delta) \geq 0\,, \quad \text{for all } \delta \in \Delta$$

$$\Uparrow$$

$$\mathcal{F}_{\Delta}(x) \geq 0$$ ***relaxation***

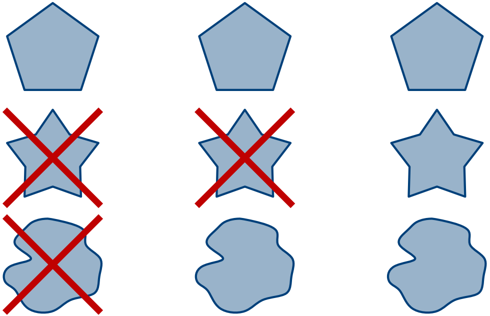
Otherwise: Relaxations

A novel scheme for constructing relaxations

- more general than existing schemes

	Pólya	sum-of-squares	**new**
$f_x(\delta)$	polynomial	polynomial	piecewise polynomial

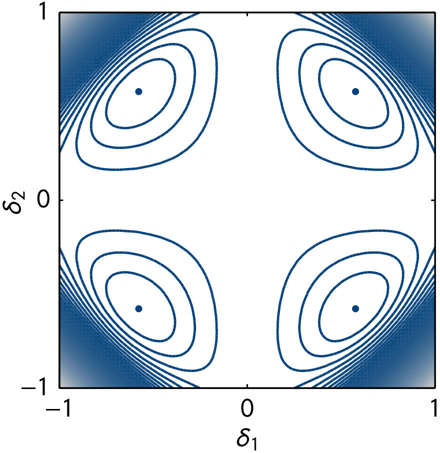
Δ



Otherwise: Relaxations

A novel scheme for constructing relaxations

- more general than existing schemes
- computationally more efficient than existing schemes

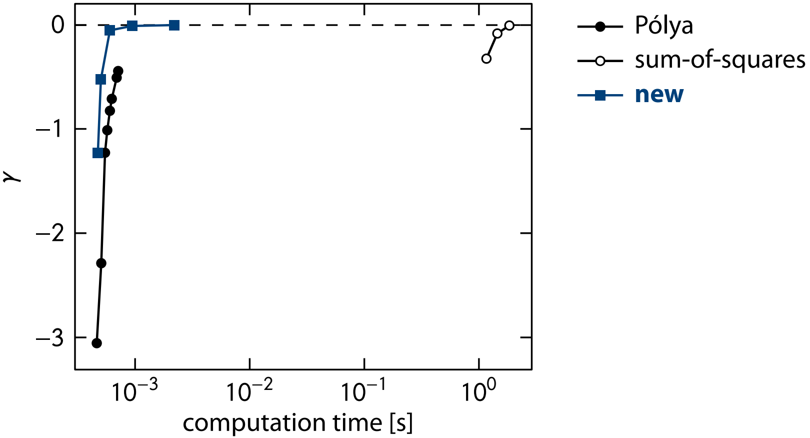


$$\begin{array}{ll} \underset{\gamma}{\text{maximize}} & \gamma \\ \text{subject to} & f(\delta) \geq \gamma\,, \ \text{for all} \ \delta \in \Delta \end{array}$$

Otherwise: Relaxations

A novel scheme for constructing relaxations

- more general than existing schemes
- computationally more efficient than existing schemes



B-spline Relaxations

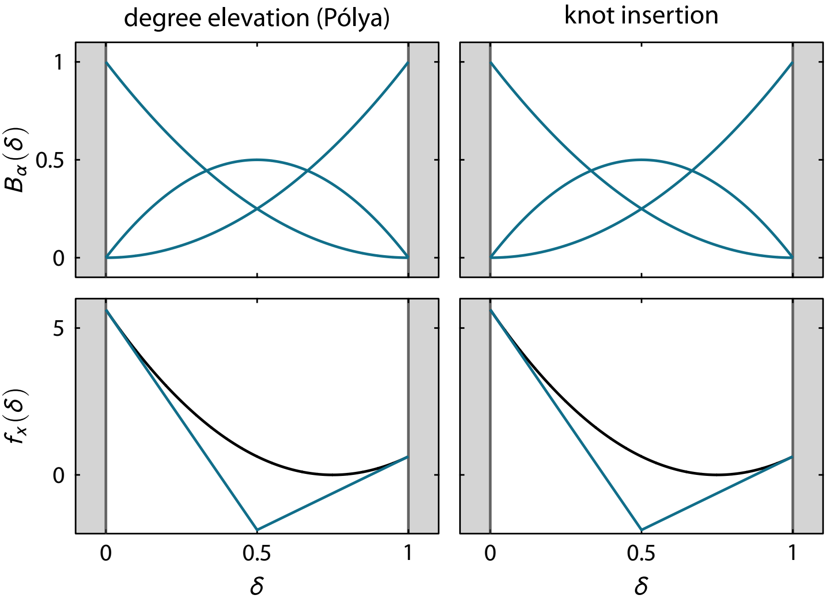
Overall idea

- B-splines: piecewise polynomial basis functions B_α
  - partition of unity on Δ: $\sum_{\alpha \in \boldsymbol{\alpha}} B_\alpha(\delta) = 1, \ \forall \delta \in \Delta$
  - positive on Δ: $B_\alpha(\delta) \geq 0, \ \forall \delta \in \Delta$

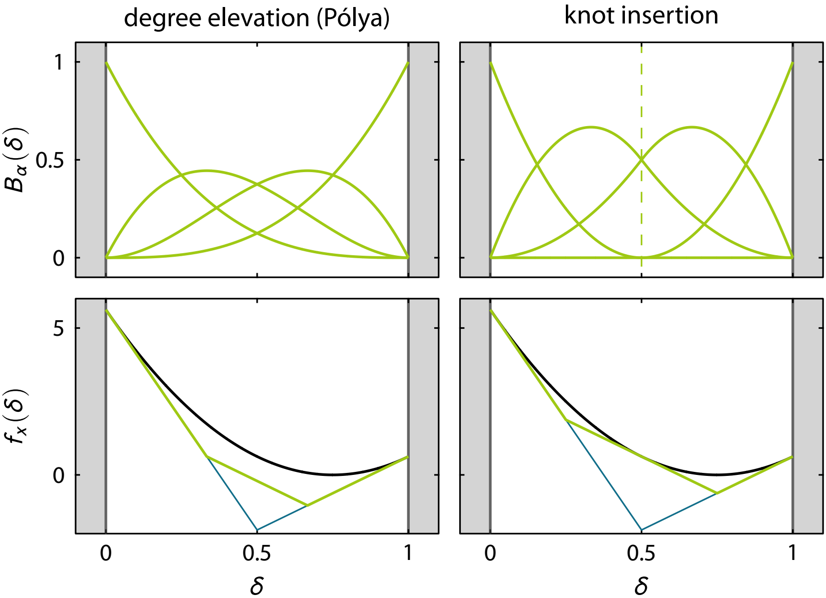
- relaxation: $$f(x, \delta) = \sum_{\alpha \in \boldsymbol{\alpha}} c_\alpha(x) B_\alpha(\delta) \geq 0, \ \ \forall \delta \in \Delta$$
$$\Uparrow$$
$$c_\alpha(x) \geq 0, \ \ \forall \alpha \in \boldsymbol{\alpha}$$

- refinement: higher-dimensional B-spline bases
  - piecewise polynomial of higher degree
  - piecewise polynomial on finer grid

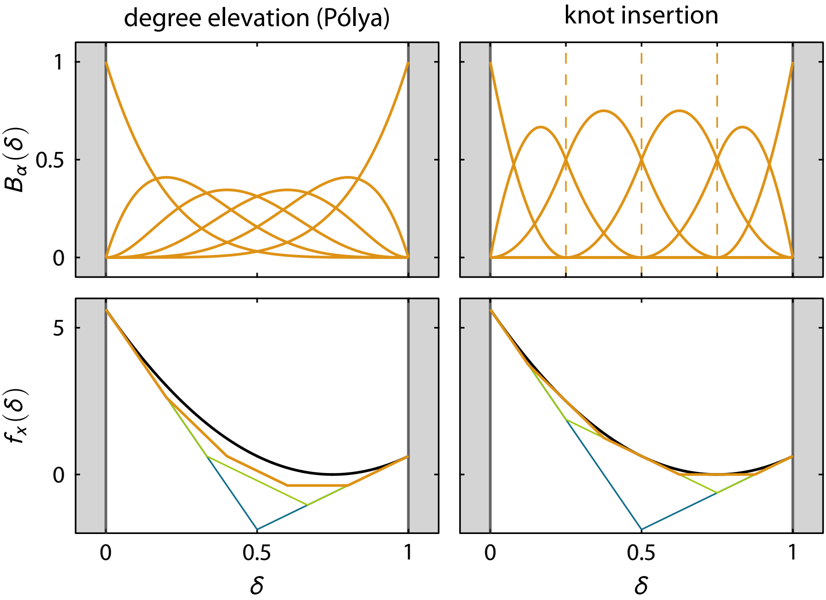
B-spline Relaxations



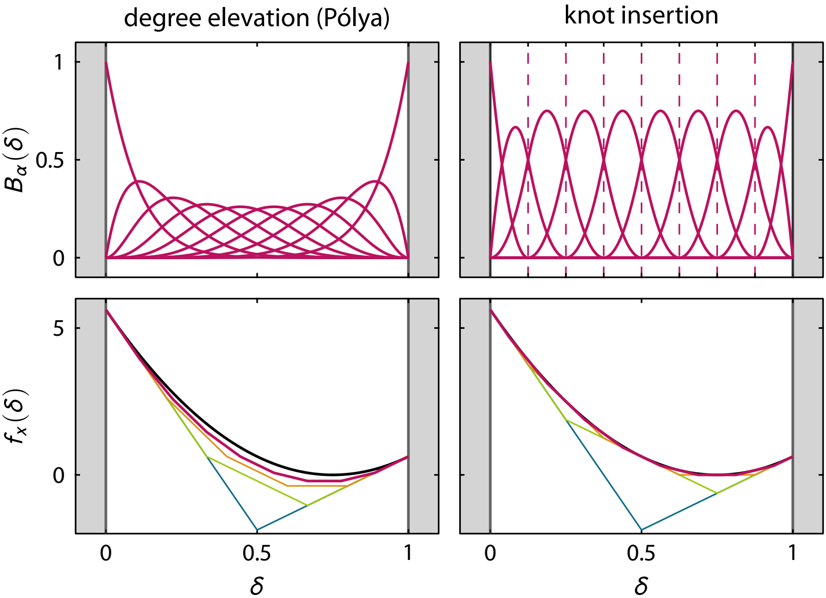
B-spline Relaxations



B-spline Relaxations



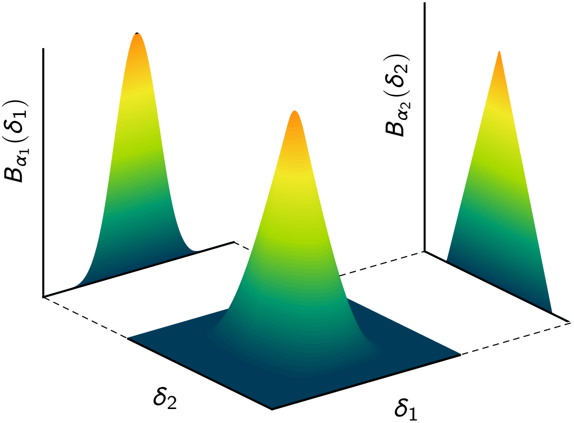
B-spline Relaxations



B-spline Relaxations

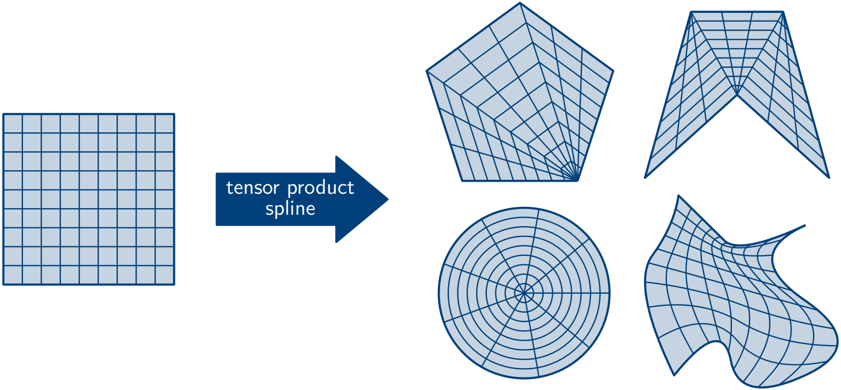
Tensor product B-splines

$$B_{\alpha}(\delta) = B_{\alpha_1}(\delta_1) B_{\alpha_2}(\delta_2)$$



B-spline Relaxations

Towards non-hyperrectangular domains

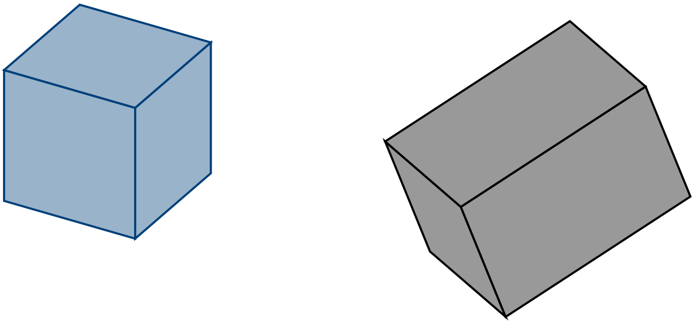


1. *Beyond the original raison d'être*
2. *Solving robust optimization problems*
3. *Results in applications*

Motion Planning

Avoiding collision

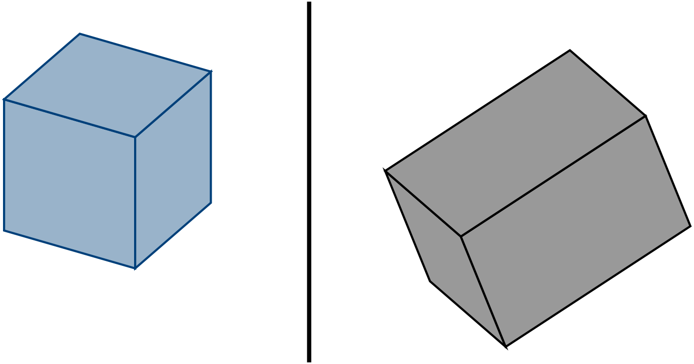
- exact reformulation using linear separation



Motion Planning

Avoiding collision

- exact reformulation using linear separation



Motion Planning

Avoiding collision

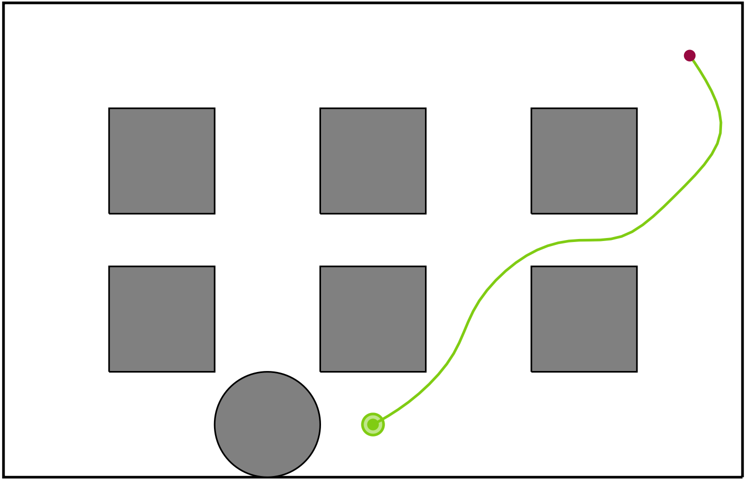
- exact reformulation using linear separation

Enforcing constraints at all times

- B-spline parametrization of motion trajectory
- novel relaxation scheme for efficient constraint satisfaction

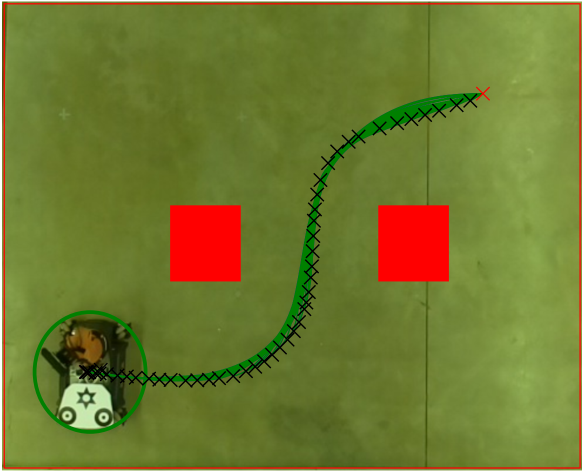
Motion Planning

20 ms for simple kinematic models



Motion Planning

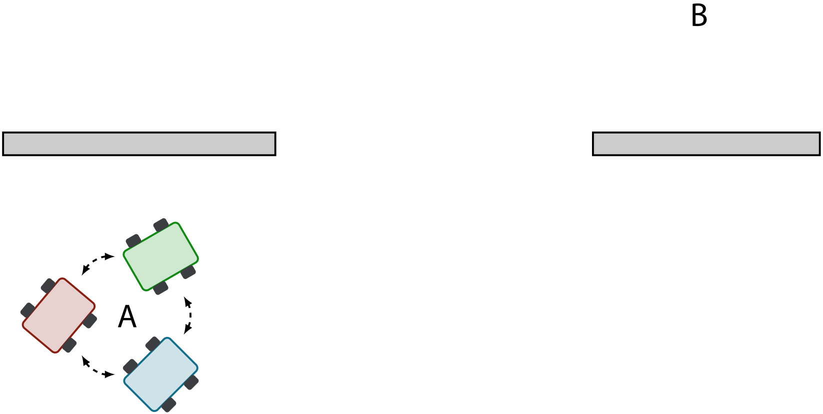
20 ms for simple kinematic models



Motion Planning

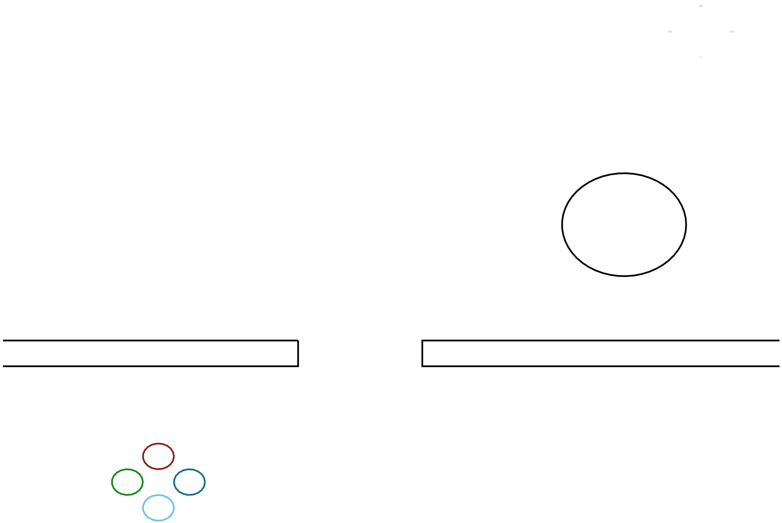
Extension to multi-agent motion planning

- optimization distributed over agents using ADMM
- 1 ADMM iteration per update



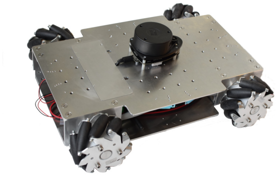
Motion Planning

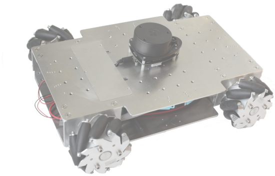
40 ms for simple kinematic models

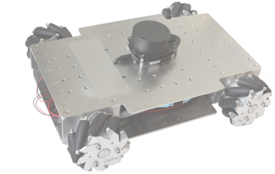


Motion Planning

40 ms for simple kinematic models



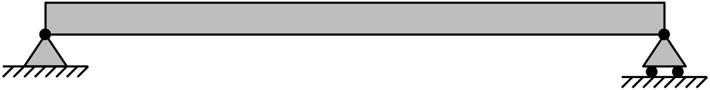




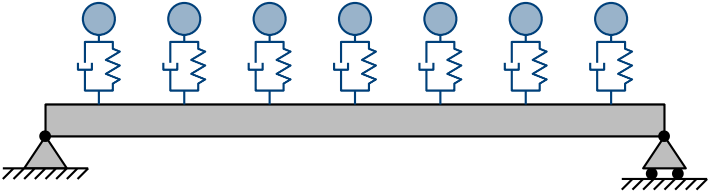
Interconnected Systems

Application in vibro-acoustics

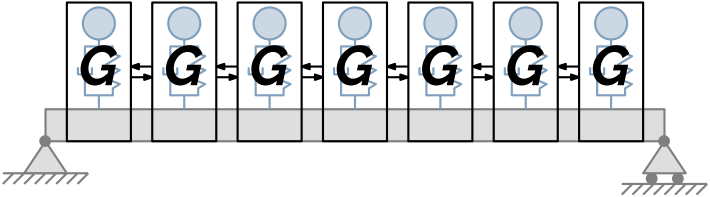
Interconnected Systems



Interconnected Systems

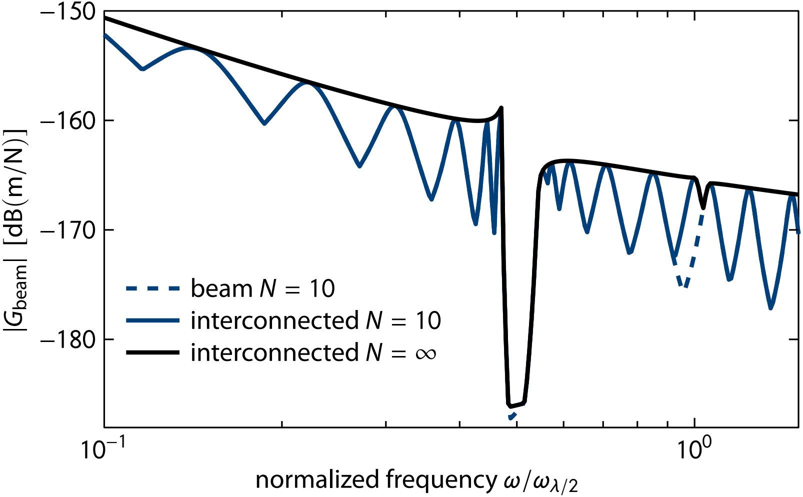


Interconnected Systems



Interconnected Systems

Efficient analysis using quadratic separation



Parametric Programming

A novel B-spline based framework

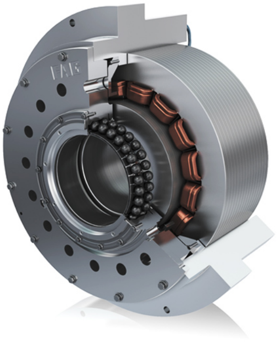
- B-spline parameterized $\hat{x}(\delta)$ for high flexibility
- novel relaxation scheme for low conservatism

Parametric Programming

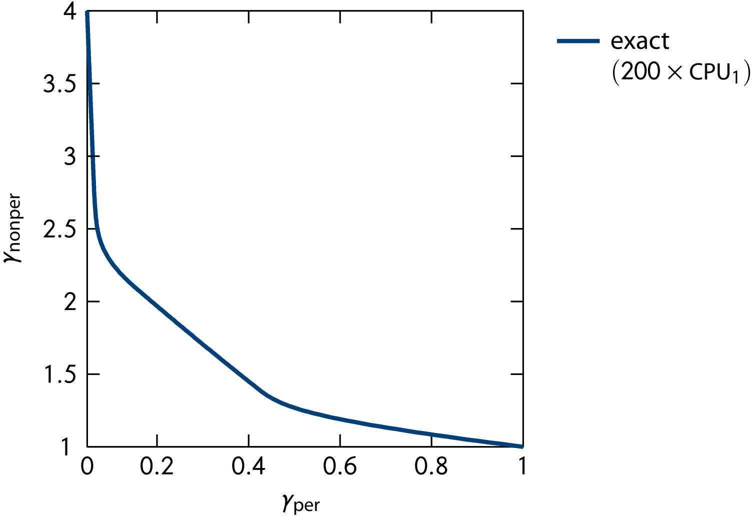
A novel B-spline based framework

- B-spline parameterized $\hat{x}(\delta)$ for high flexibility
- novel relaxation scheme for low conservatism

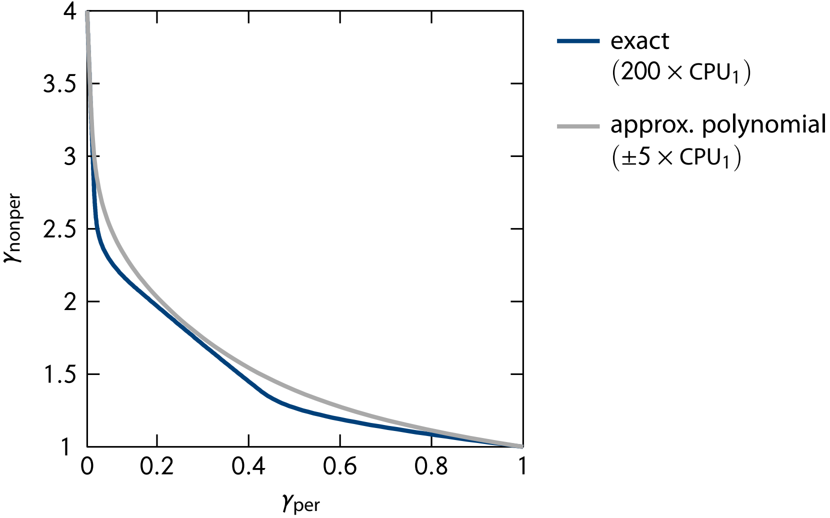
Trade-off analysis in active bearing control



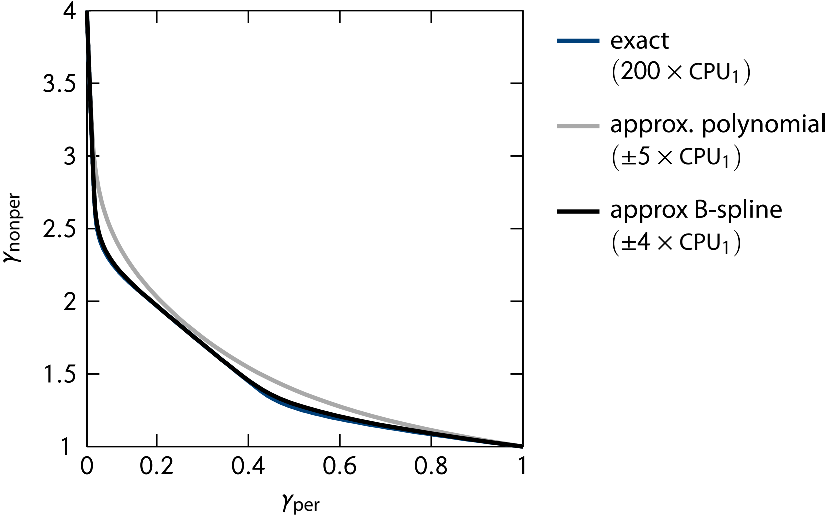
Parametric Programming



Parametric Programming



Parametric Programming



Conclusion

1. ***Robust optimization***
   has many applications in engineering

2. ***General and effective strategy***
   for solving robust optimization problems

3. ***Results in applications***

Thank You!

The MECO research group
Wannes Van Loock, Ruben Van Parys, Tim Mercy

Claus Claeys, Elke Deckers

fwo Research Foundation Flanders
Opening new horizons