VARIABLE ROBUSTNESS CONTROL:

PRINCIPLES and ALGORITHMS

Marco C. Campi

Simone Garatti

thanks to :

Simone Garatti

Giuseppe Calafiore

Maria Prandini

Algo Care’

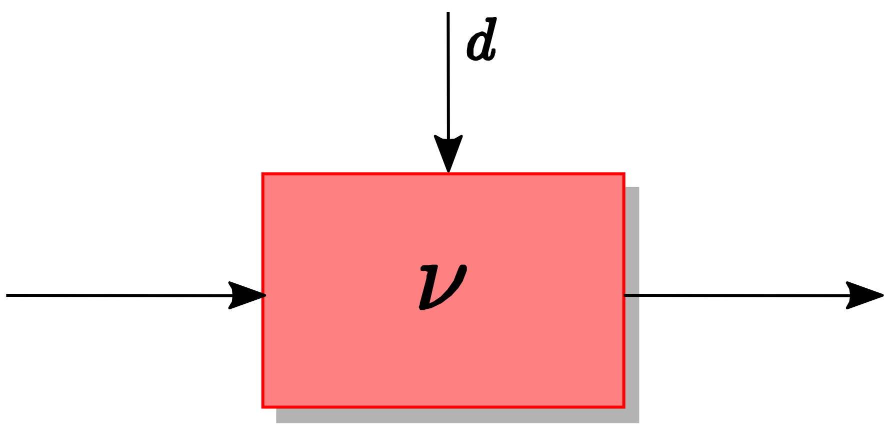
PART I: Principles

Optimization

- controller synthesis
- noise compensation
- prediction
- ⋮

→ optimization program

Uncertainty



Uncertain Optimization Program

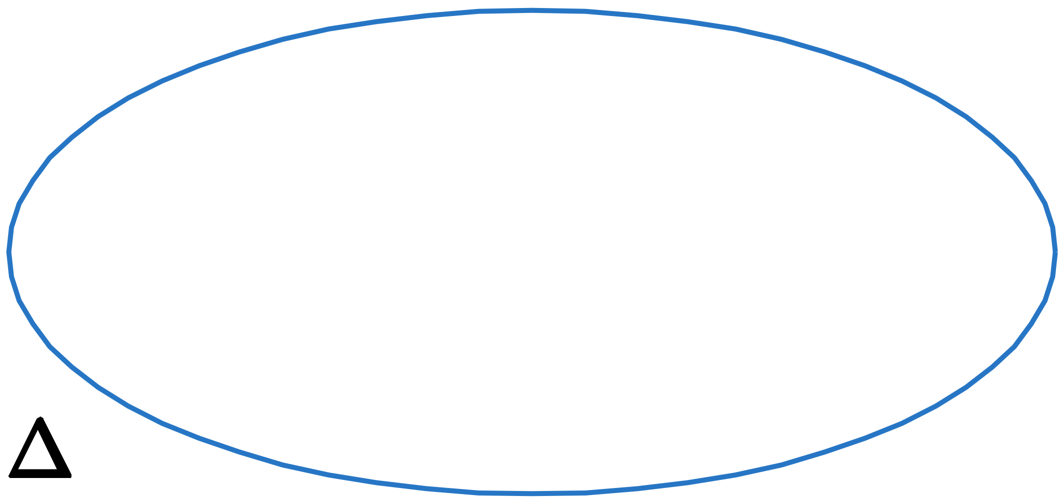
U-OP: $\min_{\theta} \ell(\theta, \delta), \qquad \delta \in \Delta$

Uncertain Optimization Program

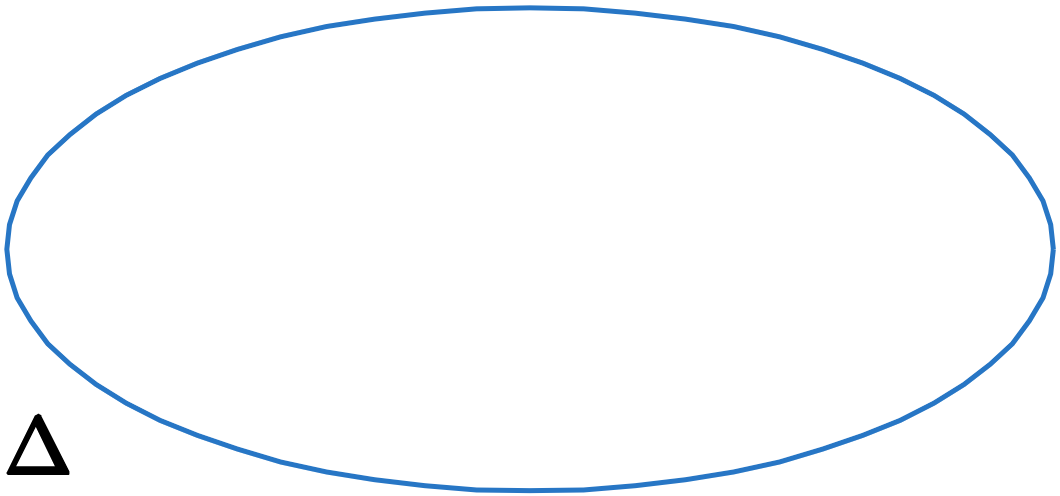
U-OP: $\min_{\theta} \ell(\theta, \delta), \qquad \delta \in \Delta$

not well-defined

Uncertainty

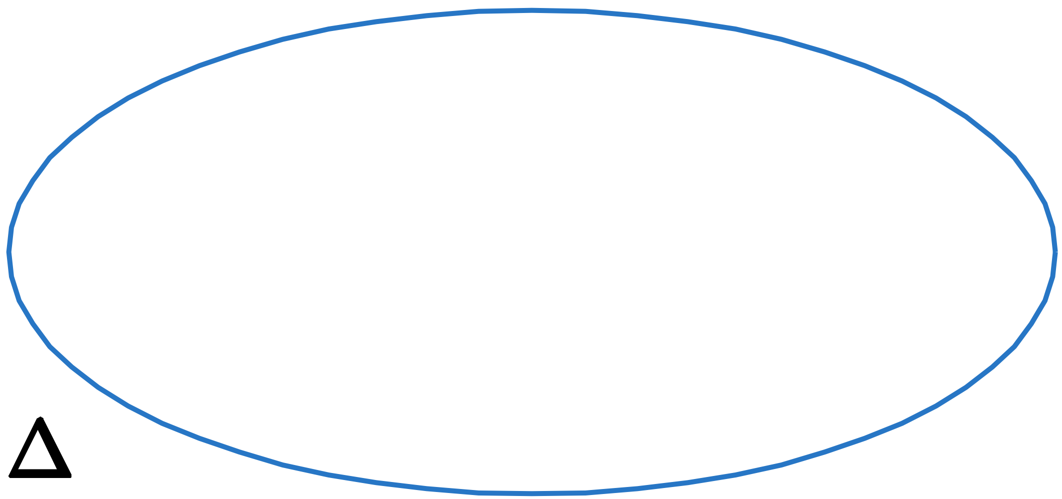


Uncertainty



$$\min_{\theta}\left[\max_{\delta\in\Delta}\ell(\theta,\delta)\right] \qquad \text{(worst-case approach)}$$

Uncertainty

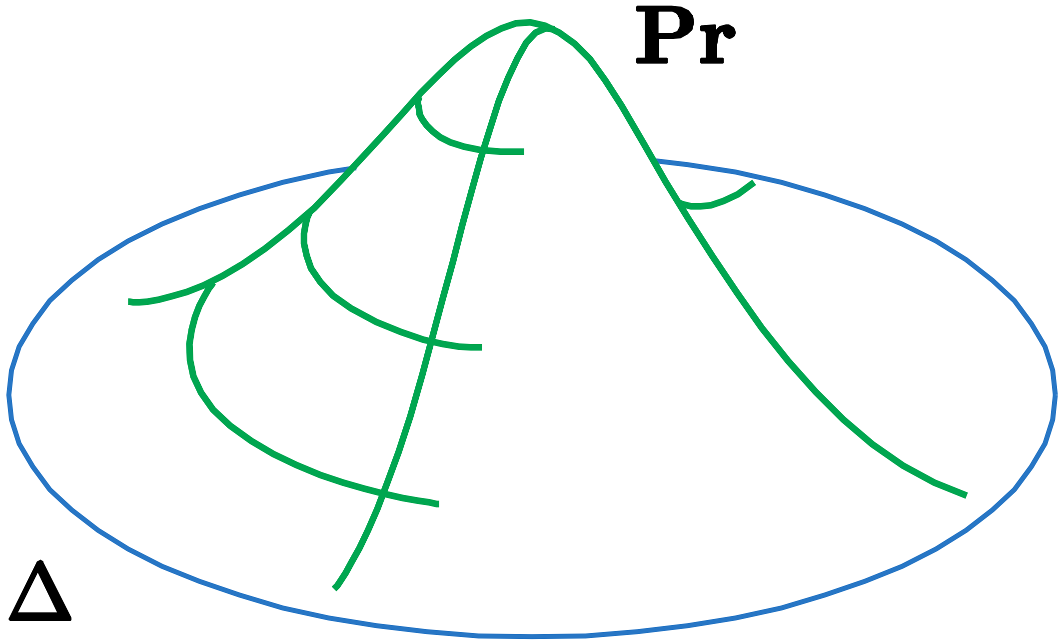


$$\min_{\theta} \left[\max_{\delta \in \Delta} \ell(\theta, \delta) \right] \qquad \text{(worst-case approach)}$$

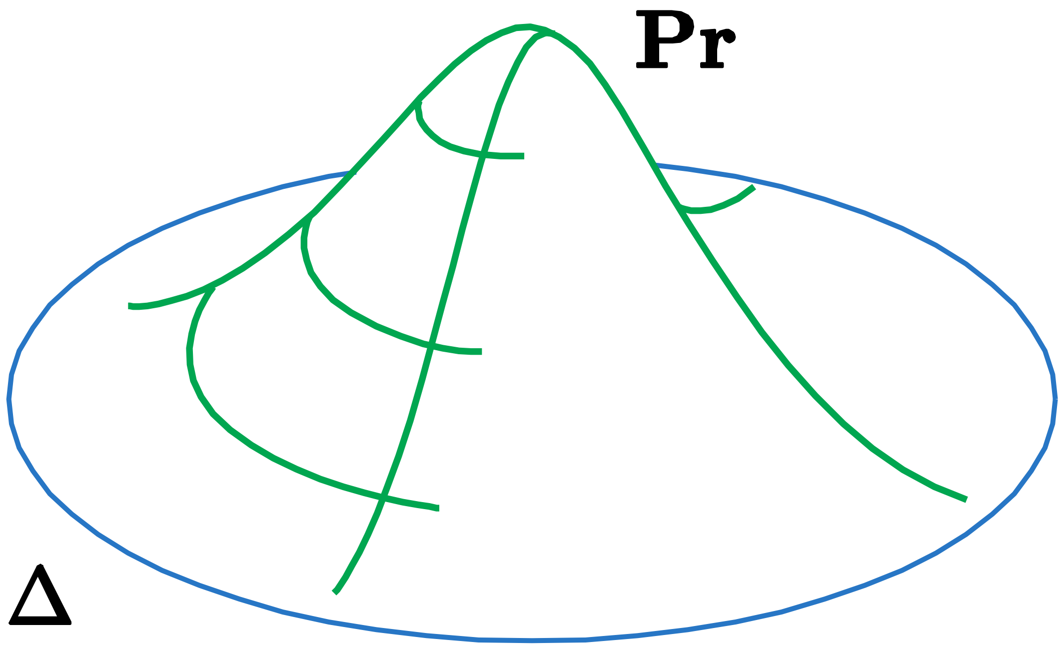
H_∞ theory

[J.C. Doyle, 1978], [G. Zames, 1981]

Probabilistic uncertainty

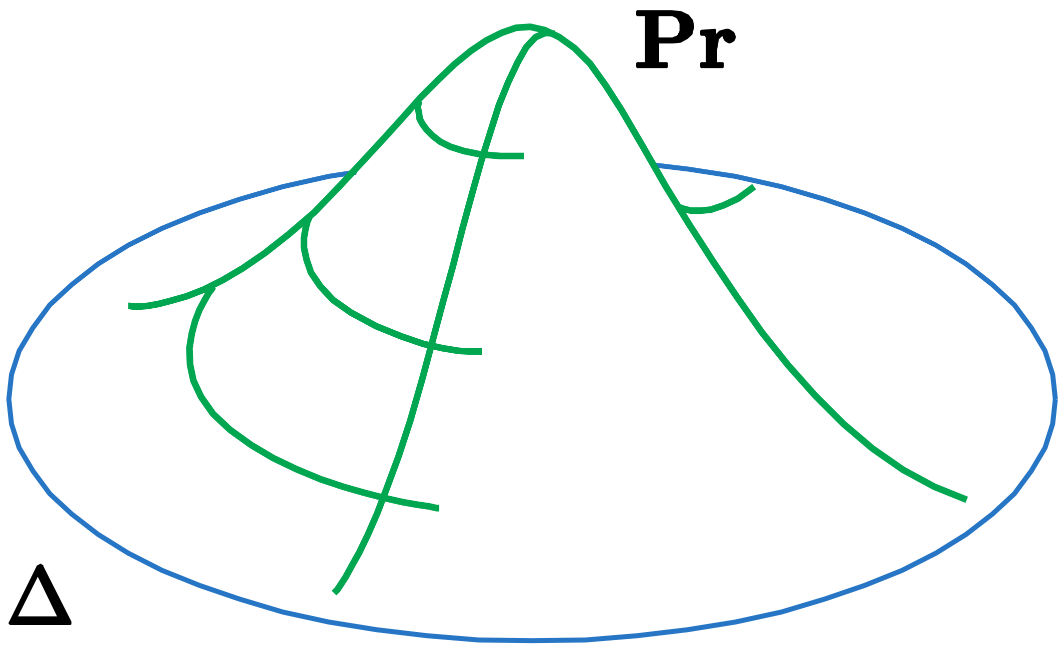


Probabilistic uncertainty



$$\min_{\theta} E_{\Delta}\left[\ell(\theta,\delta)\right] \qquad \text{(average approach)}$$

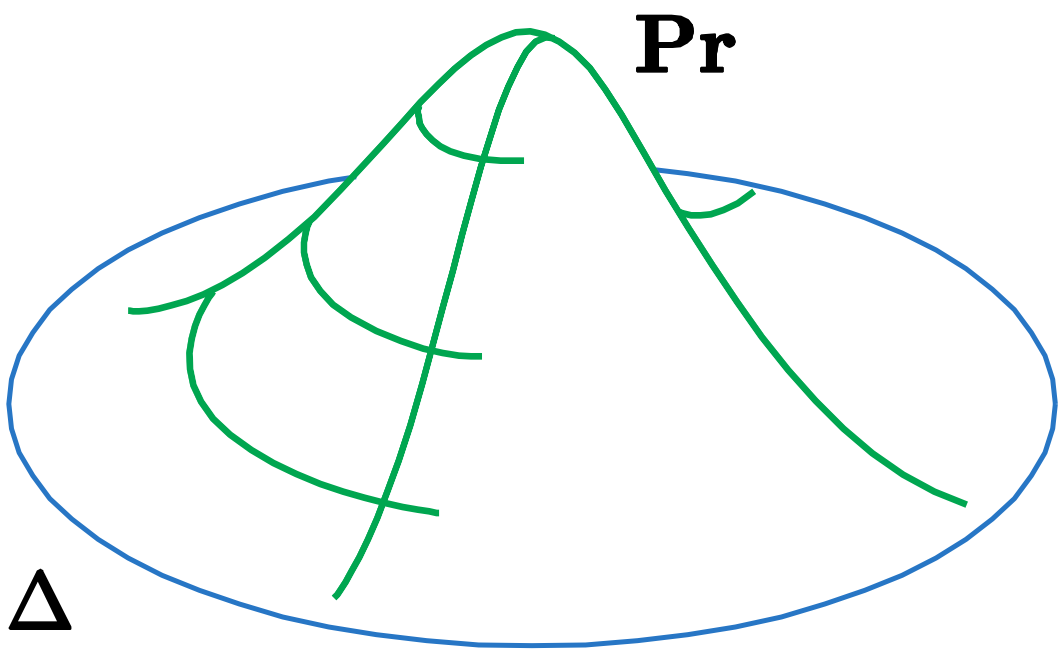
Probabilistic uncertainty



$\min_{\theta} E_{\Delta}\left[\ell(\theta, \delta)\right]$ (average approach)

stochastic control: $E_{\Delta}[\sum_t x_t^T Q x_t + u_t^T R u_t]$

Probabilistic uncertainty

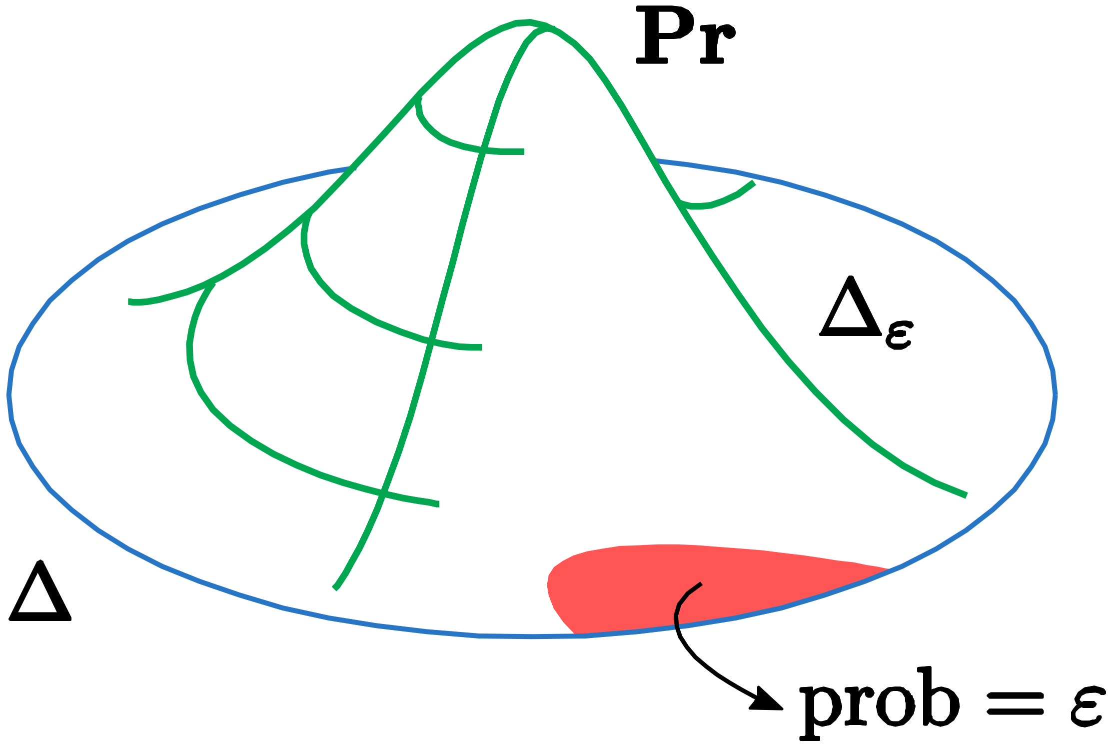


$$\min_{\theta} E_{\Delta}\left[\ell(\theta,\delta)\right] \qquad \text{(average approach)}$$

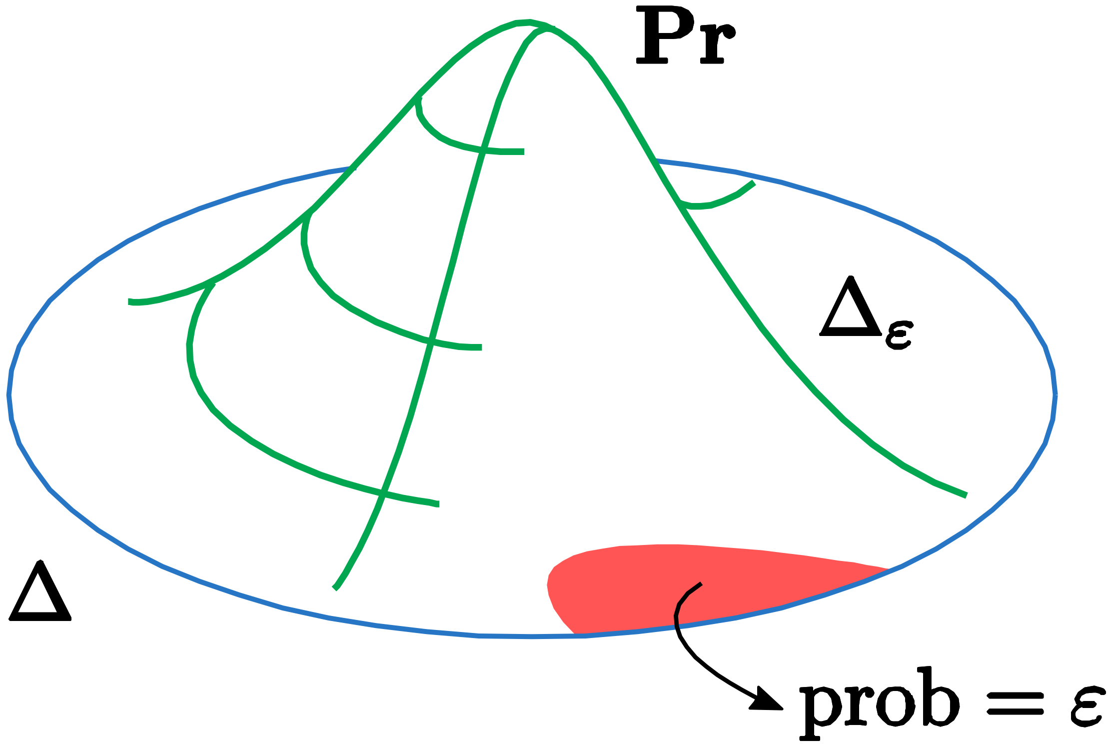
stochastic control: $E_{\Delta}[\sum_t x_t^T Q x_t + u_t^T R u_t]$

structural uncertainty: [M. Vidyasagar, 1998]

Probabilistic uncertainty



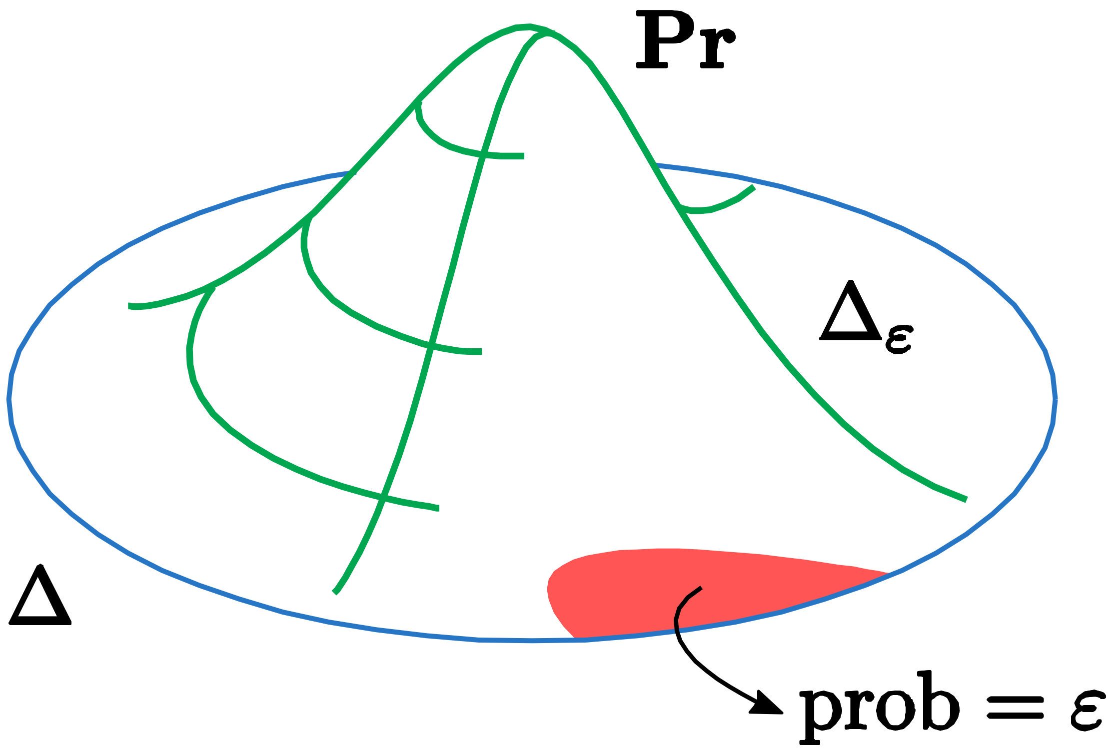
Probabilistic uncertainty



R.F. Stengel, L.R. Ray, B.R. Barmish, C.M. Lagoa …

R. Tempo, E.W. Bai, F. Dabbene, P.P. Khargonekar, A. Tikku, …

Probabilistic uncertainty



$$\min_{\theta} \left[\max_{\delta \in \Delta_\epsilon} \ell(\theta, \delta) \right]$$

$Pr(\Delta_\epsilon) = 1 - \epsilon$ (chance-constrained approach)

Probabilistic uncertainty

chance-constrained approach:

[A. Charnes, W.W. Cooper, and G.H. Symonds, 1958]

Probabilistic uncertainty

chance-constrained approach:

[A. Charnes, W.W. Cooper, and G.H. Symonds, 1958]

almost neglected by the systems
and control community:

(i) tradition;

(ii) lack of algorithms.

Probabilistic uncertainty

chance-constrained approach:

[A. Charnes, W.W. Cooper, and G.H. Symonds, 1958]

almost neglected by the systems
and control community:

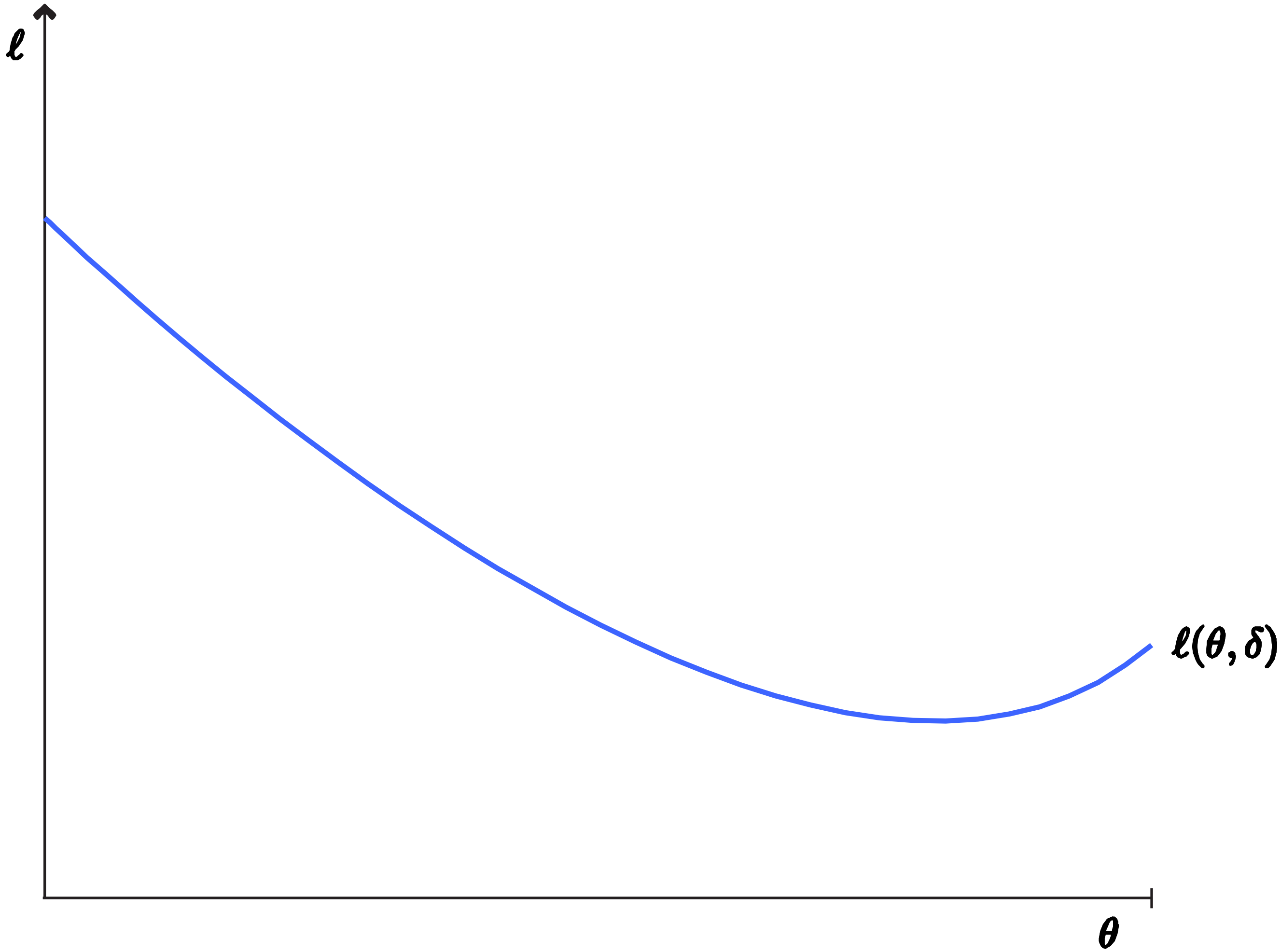
(i) tradition;

(ii) lack of algorithms.

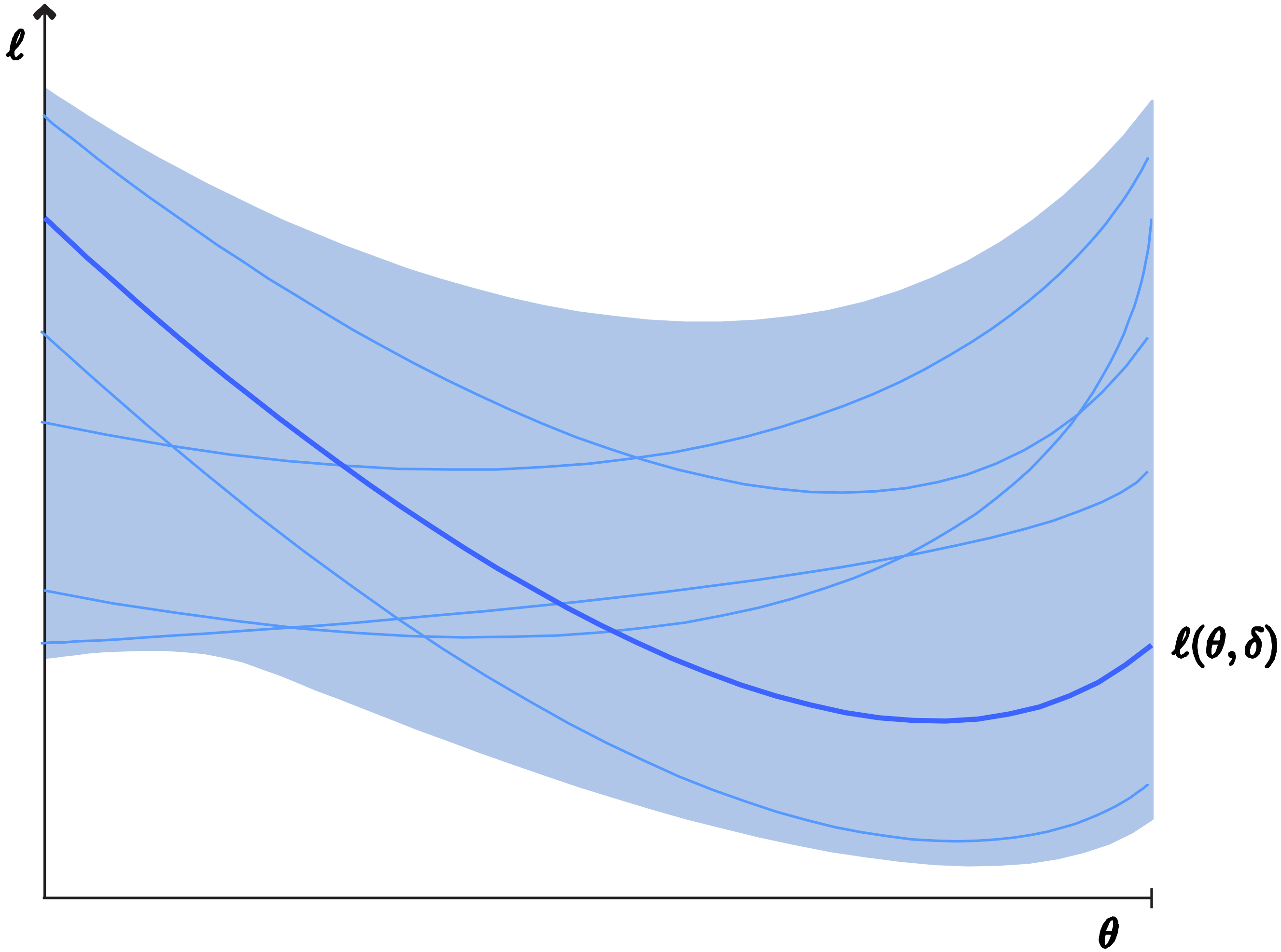
GOALS: **1. excite interest in the chance-constrained approach**

2. provide algorithmic tools

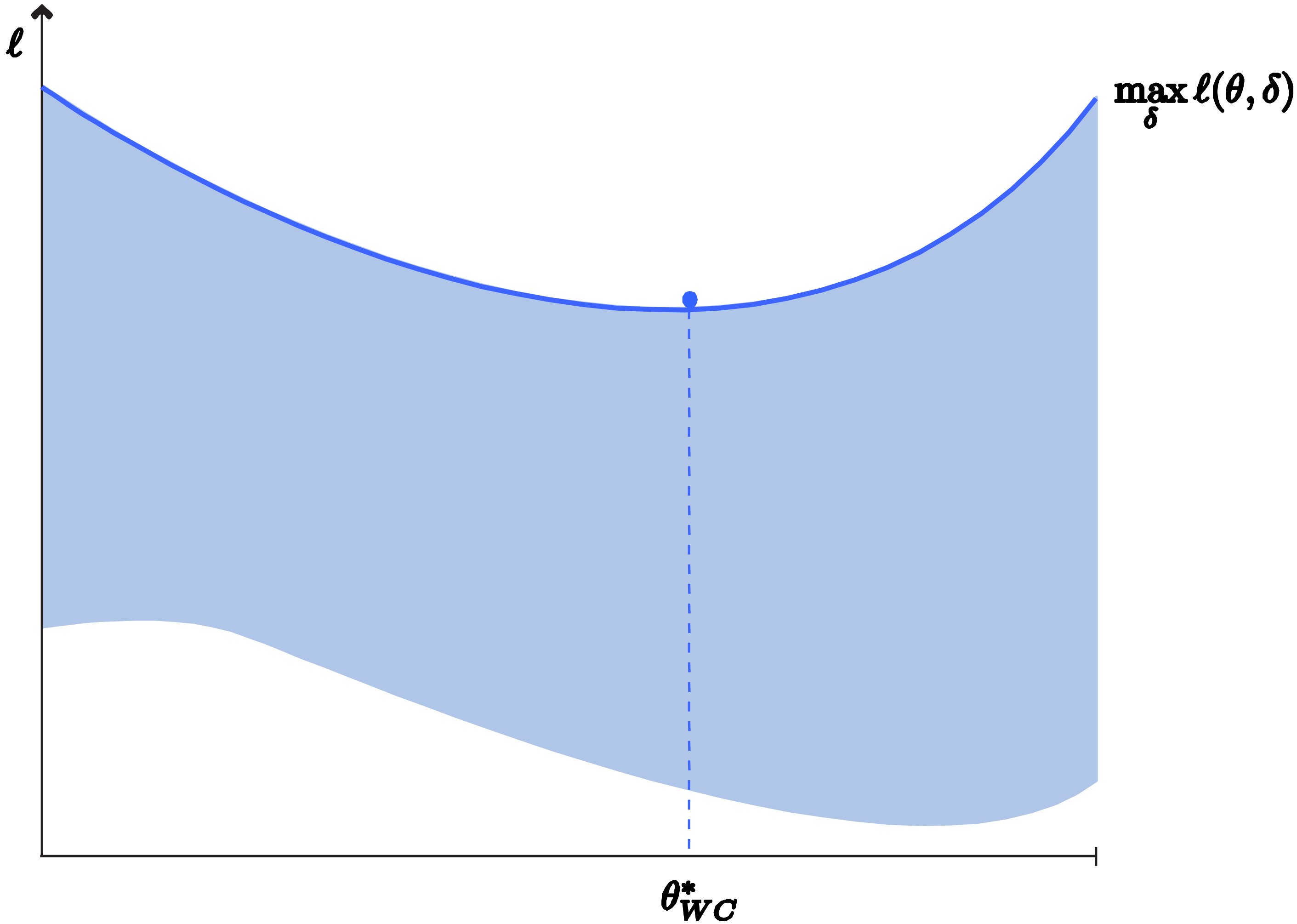
a look at optimization in the $\theta-\ell$ space



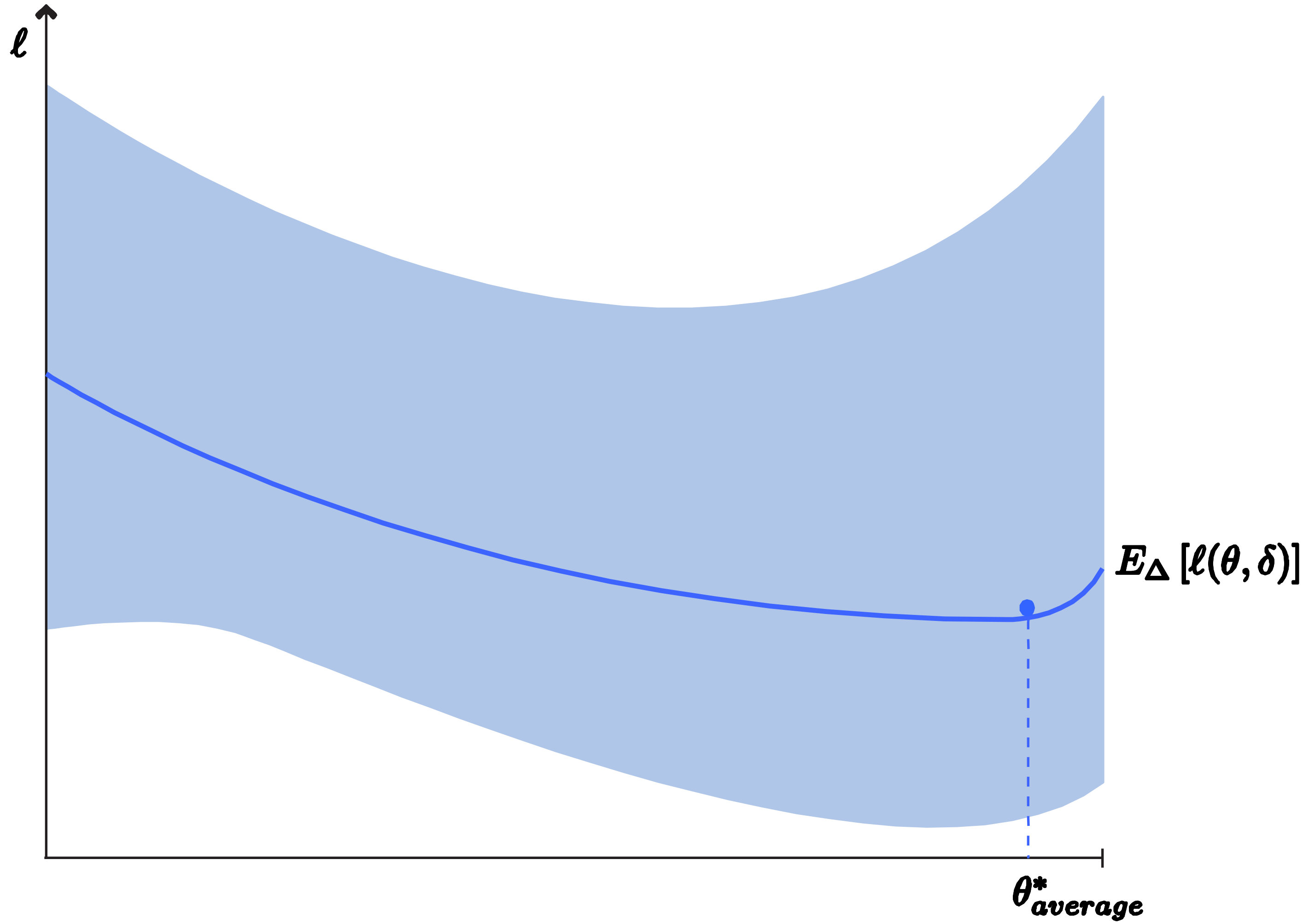
performance cloud



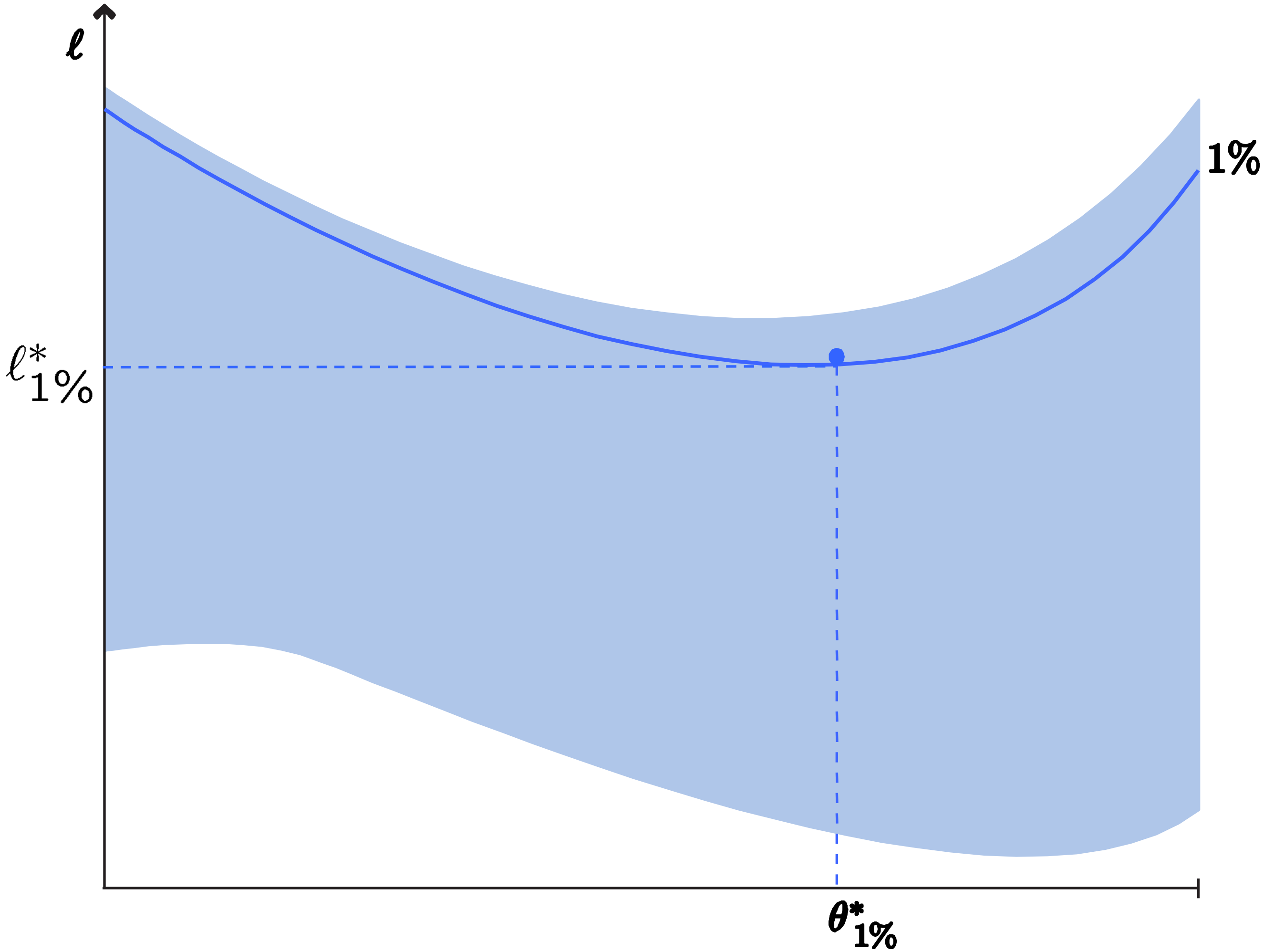
worst-case



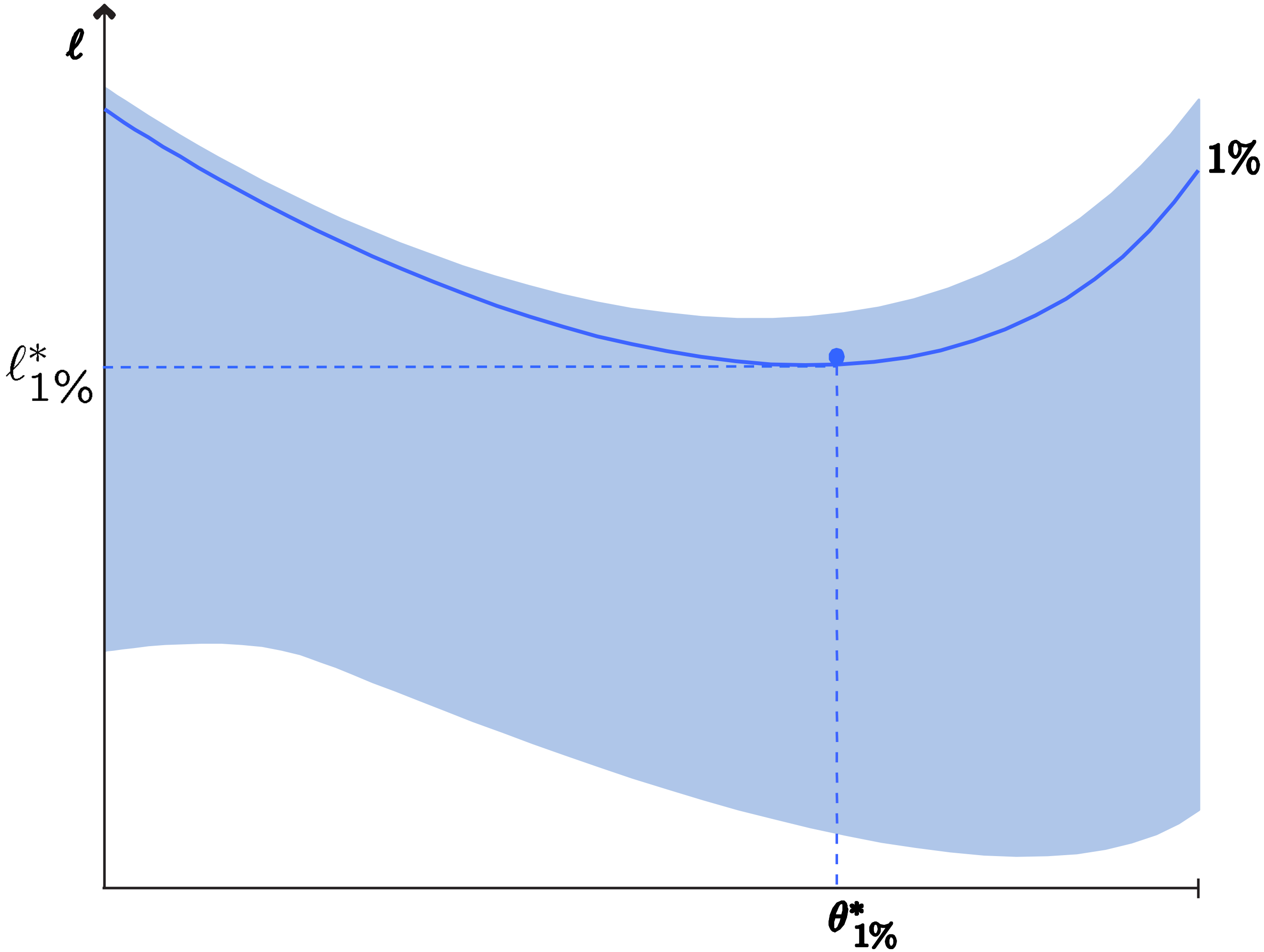
average



chance-constrained approach

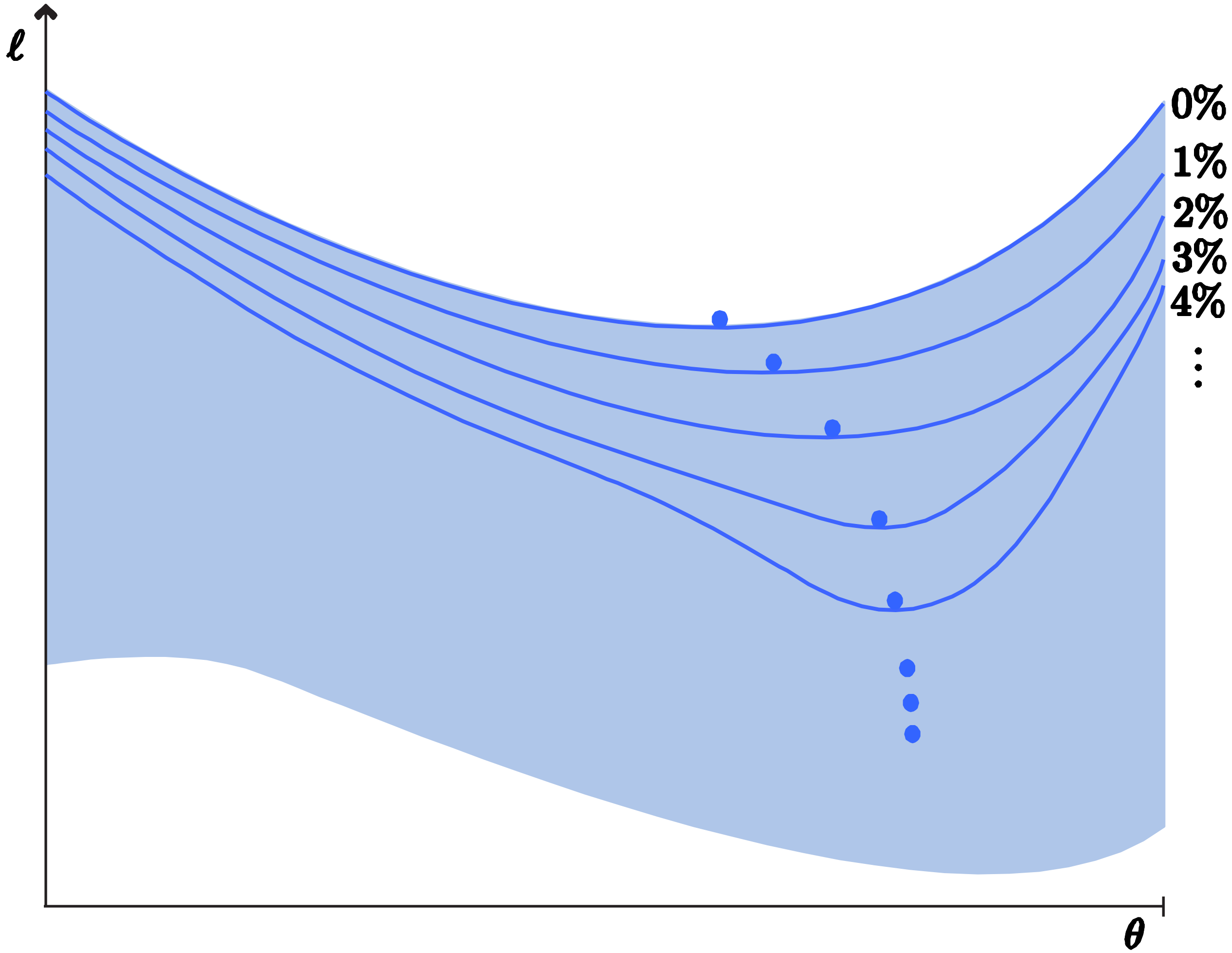


chance-constrained approach

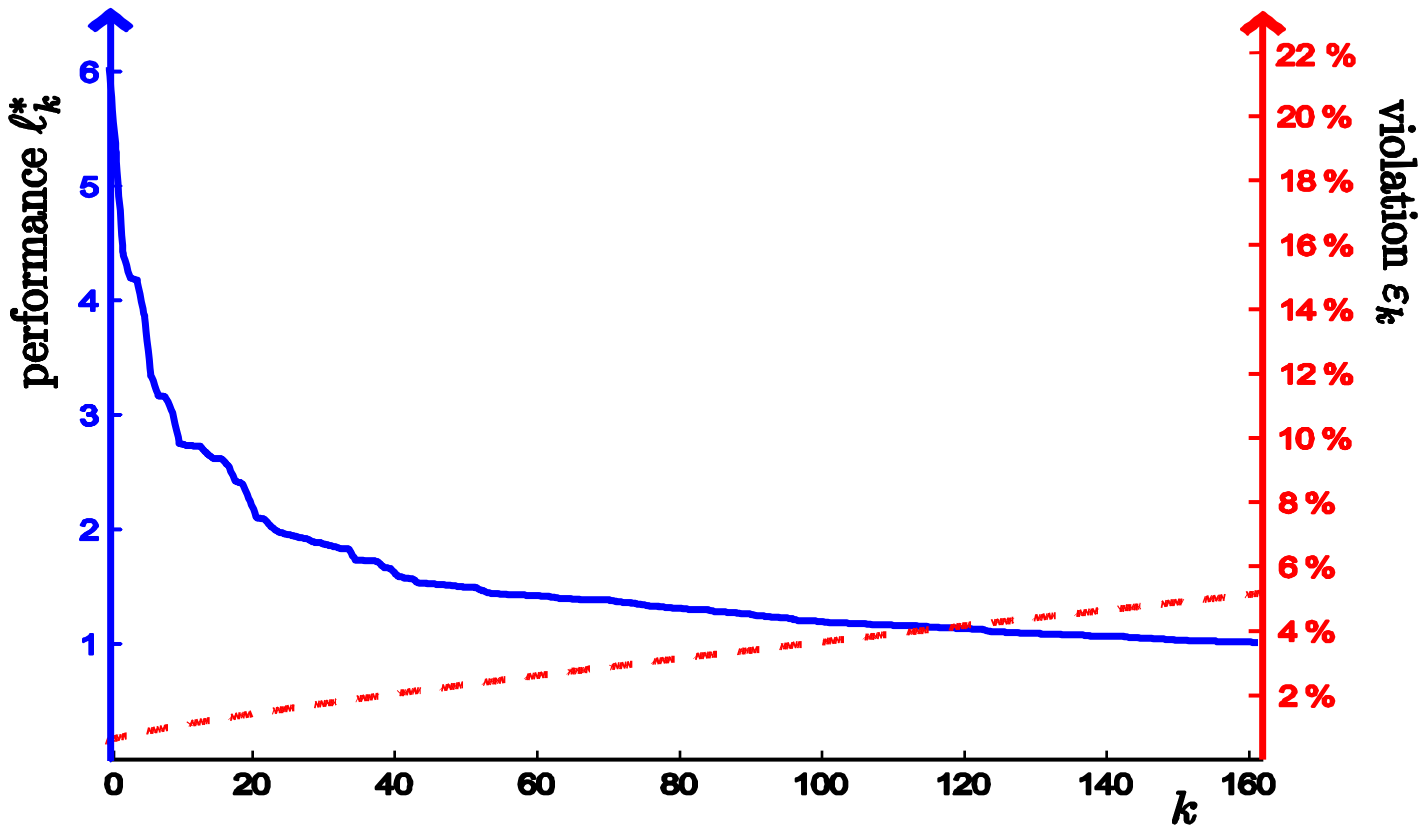


very hard to solve!

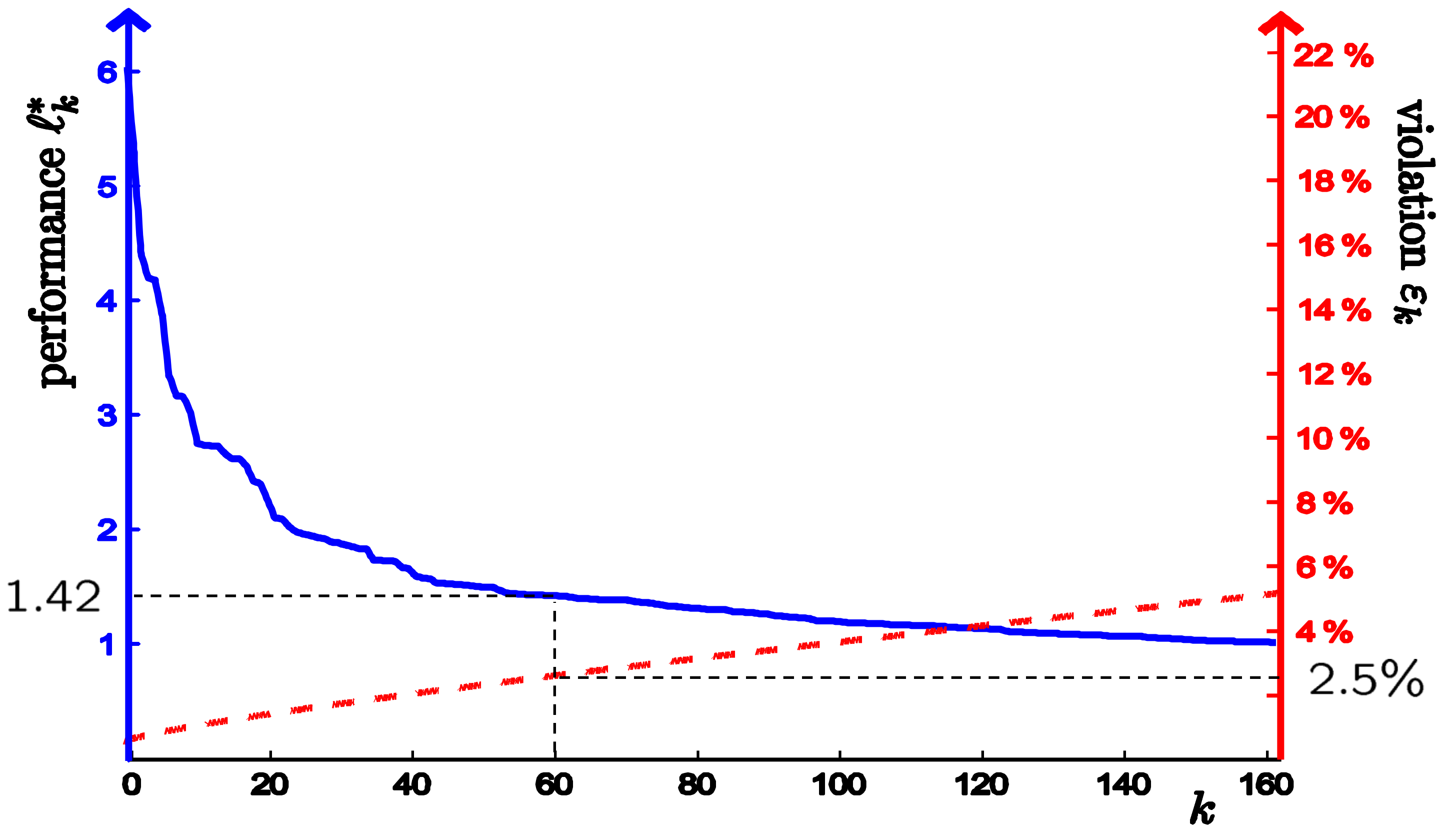
VRC – Variable Robustness Control



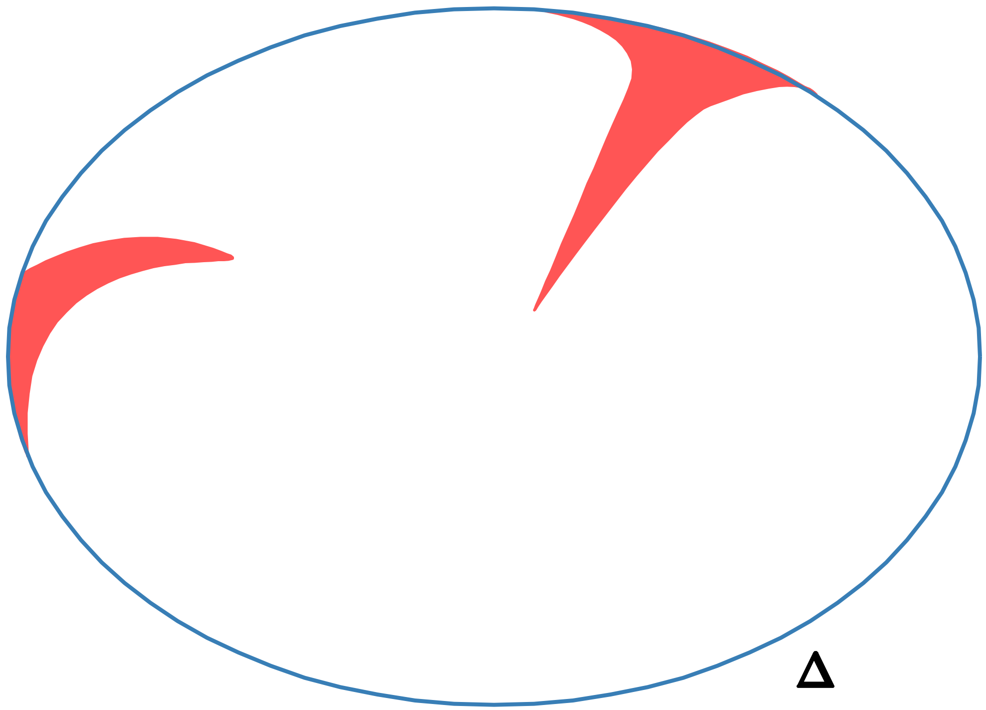
performance - violation plot



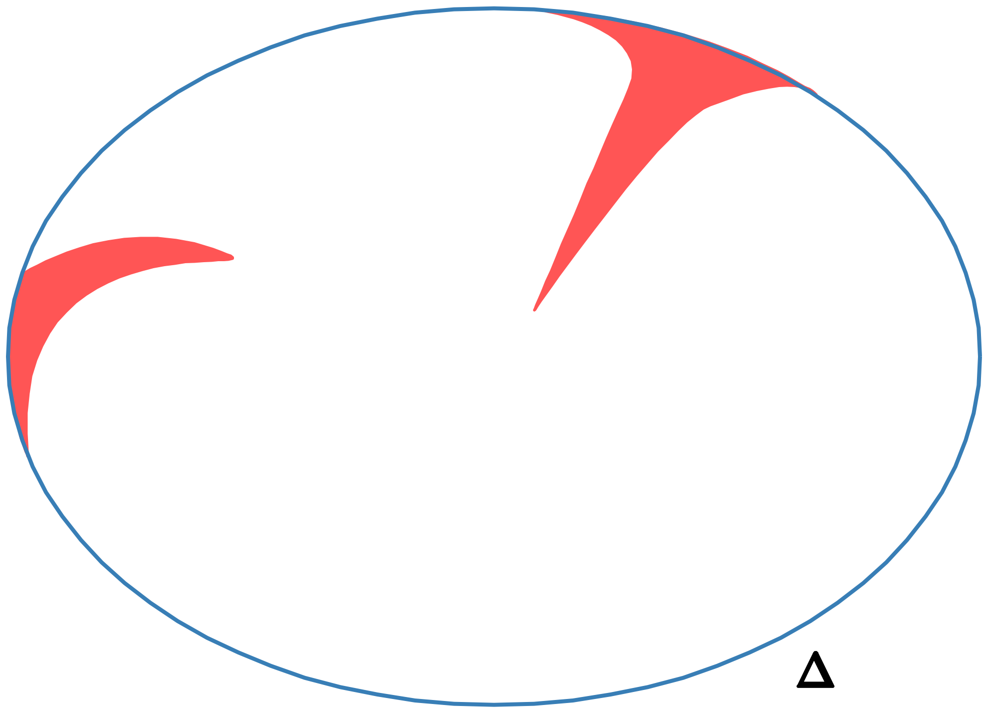
performance - violation plot



icicle geometry [C.M. Lagoa & B.R. Barmish, 2002]



icicle geometry [C.M. Lagoa & B.R. Barmish, 2002]

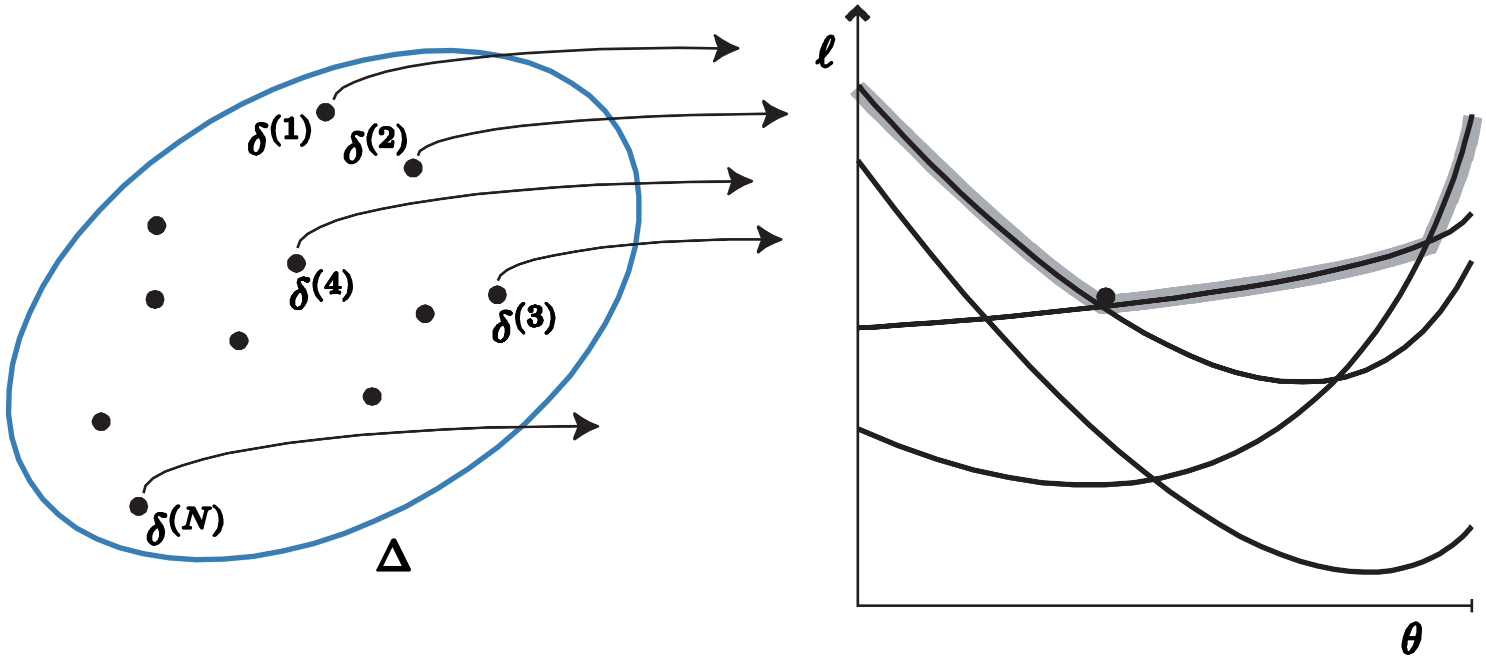


… let the problem speak

PART II: Algorithms

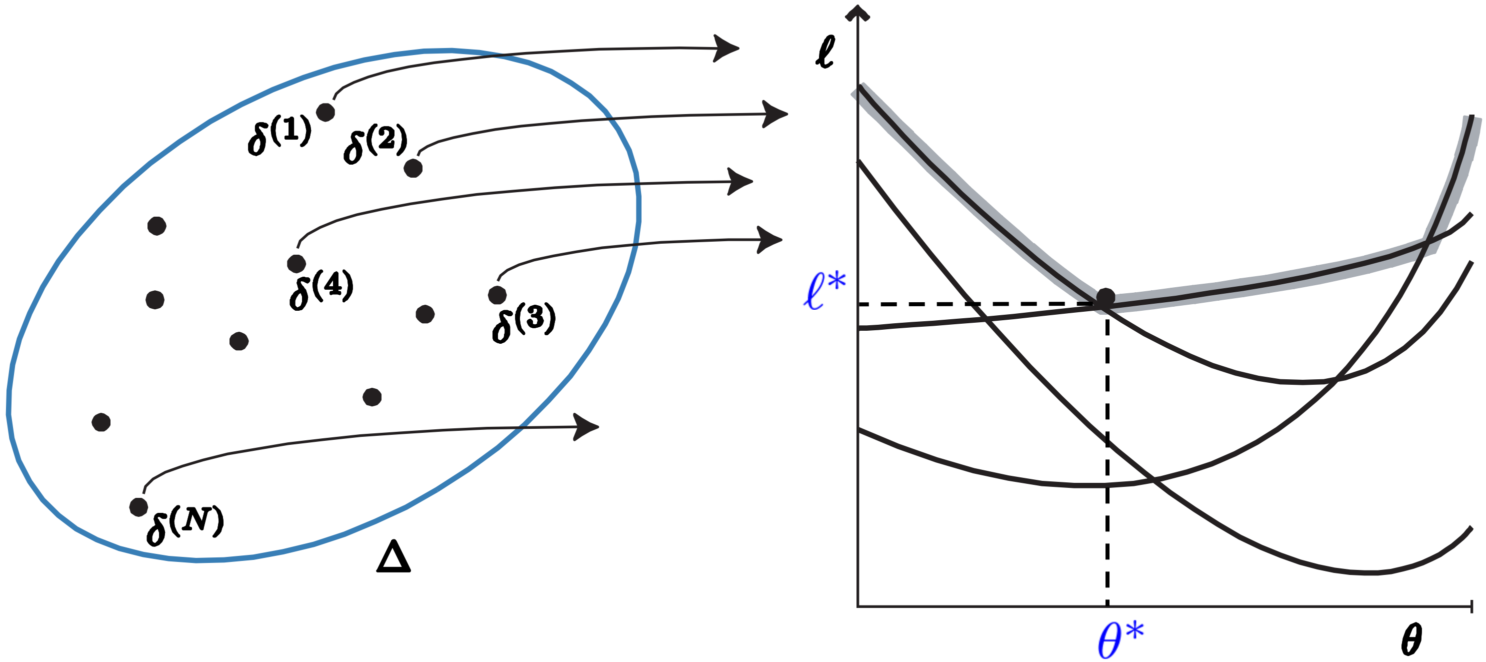
(convex case)

The "scenario" paradigm



[G. Calafiore & M. Campi, 2005, 2006]

The "scenario" paradigm



SP_N = scenario program

- SP_N is a standard finite convex optimization problem

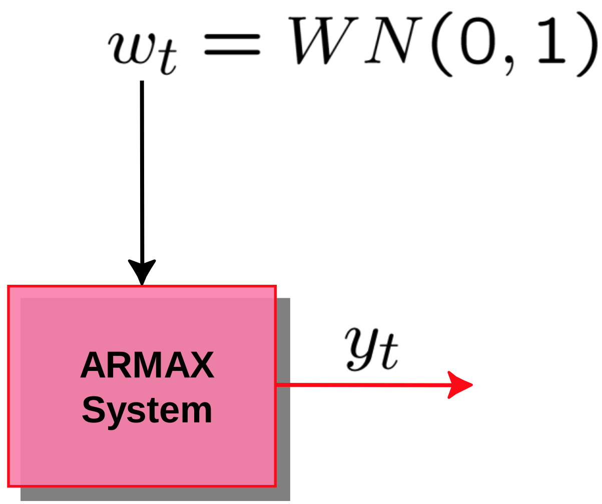
[G. Calafiore & M. Campi, 2005, 2006]

Fundamental question:

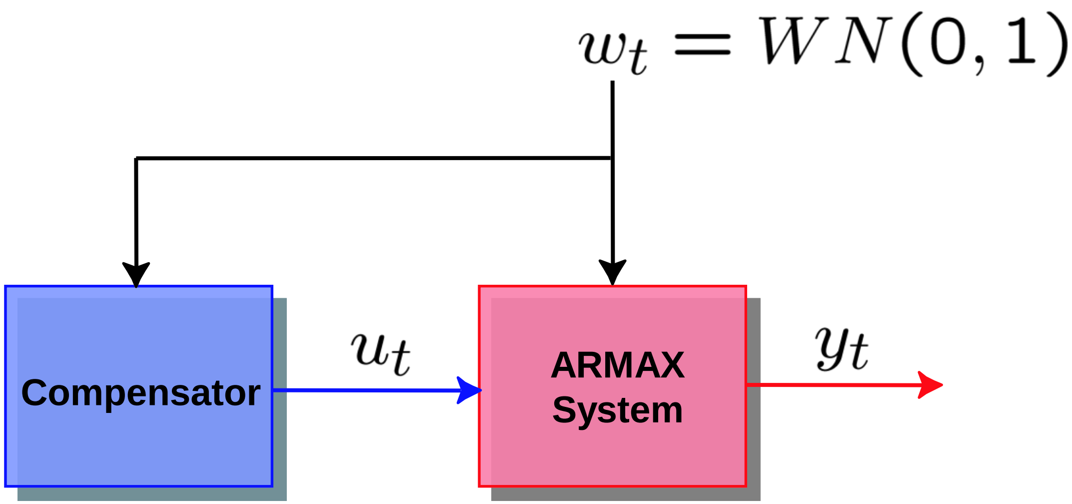
how robust is ℓ^* ?

Example: feedforward noise compensation

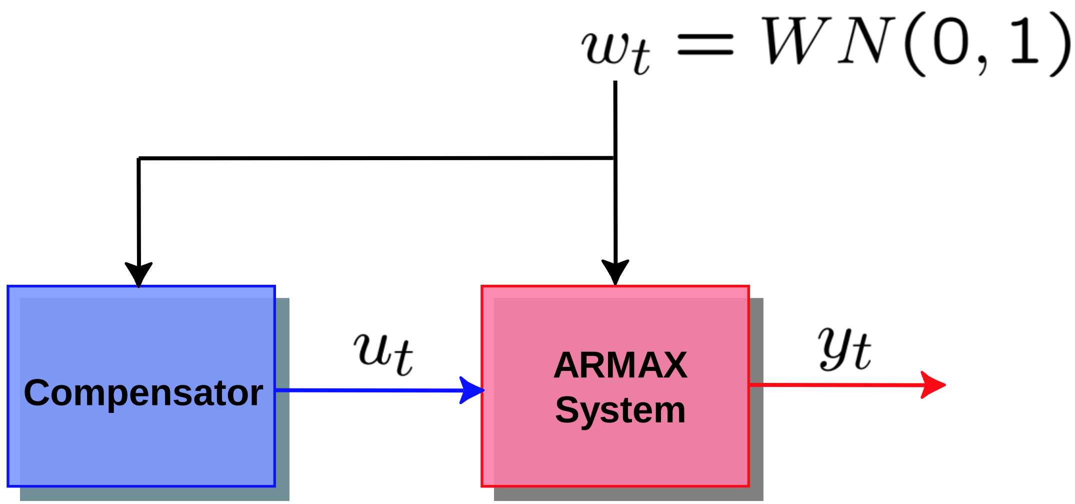
Example: feedforward noise compensation



Example: feedforward noise compensation

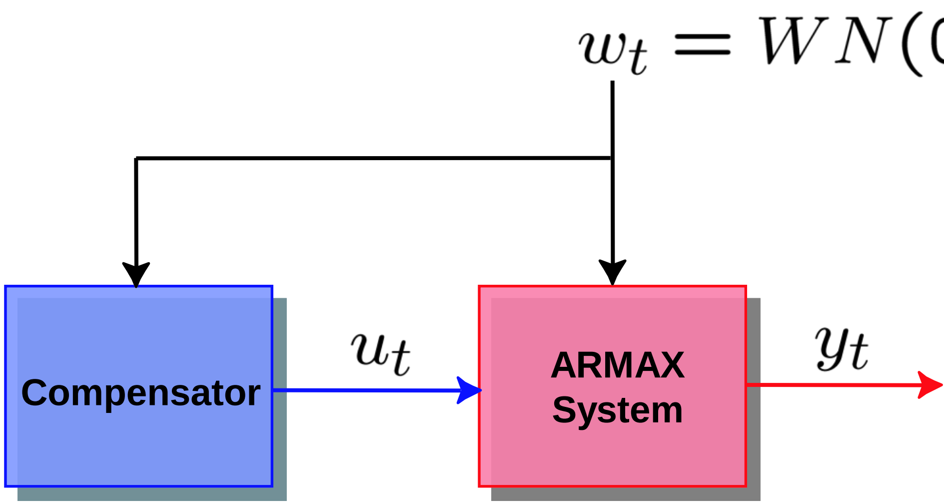


Example: feedforward noise compensation



Objective: reduce the effect of noise

Example: feedforward noise compensation



ARMAX System:

$$y_{t+1} = ay_t + bu_t + cw_t + dw_{t-1}$$

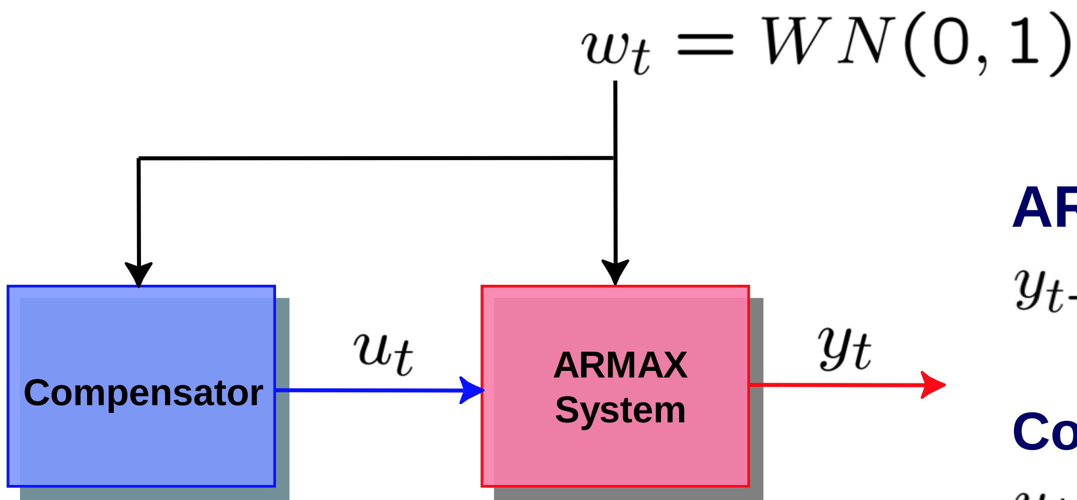
Compensator:

$$u_t = k_1 w_t + k_2 w_{t-1}$$

Goal:

$$\min var[y_t]$$

Example: feedforward noise compensation



ARMAX System:

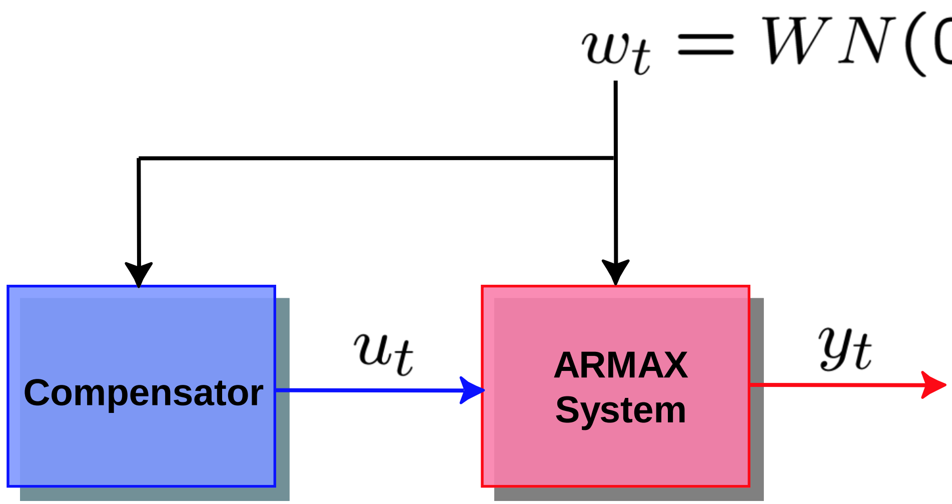
$$y_{t+1} = ay_t + bu_t + cw_t + dw_{t-1}$$

Compensator:

$$u_t = k_1 w_t + k_2 w_{t-1}$$

$$\min_{k_1,k_2} var[y_t] = \frac{(c+bk_1)^2+(d+bk_2)^2+2a(c+bk_1)(d+bk_2)}{1-a^2}$$

Example: feedforward noise compensation



ARMAX System:

$$y_{t+1} = ay_t + bu_t + cw_t + dw_{t-1}$$

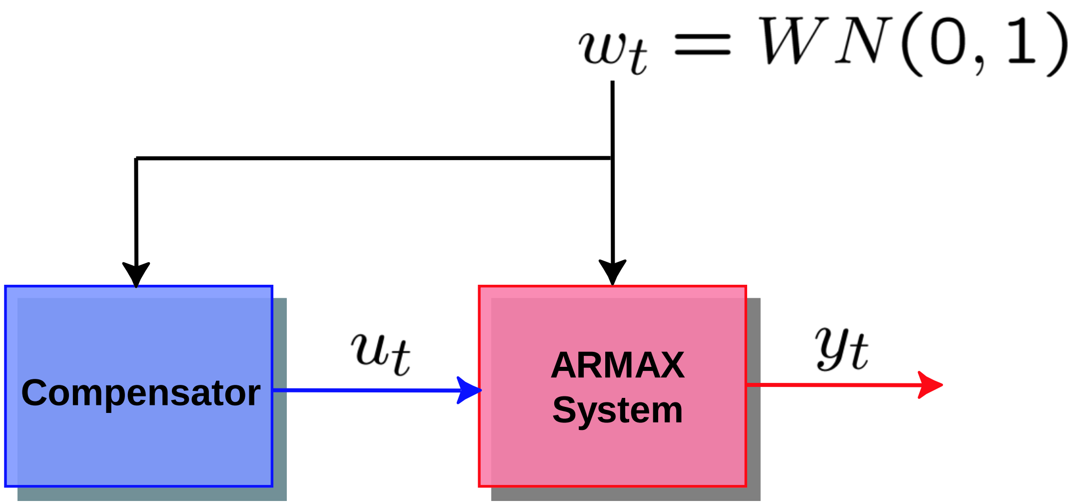
Compensator:

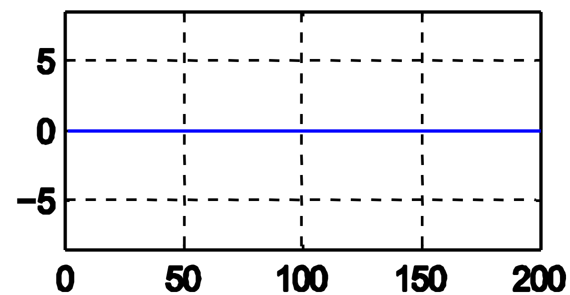
$$u_t = k_1 w_t + k_2 w_{t-1}$$

$$\min_{k_1,k_2} var[y_t] = \frac{(c+bk_1)^2+(d+bk_2)^2+2a(c+bk_1)(d+bk_2)}{1-a^2}$$

Easy: $k_1 = -\frac{c}{b} \quad k_2 = -\frac{d}{b} \quad \Rightarrow \quad var[y_t] = 0$

Example: feedforward noise compensation





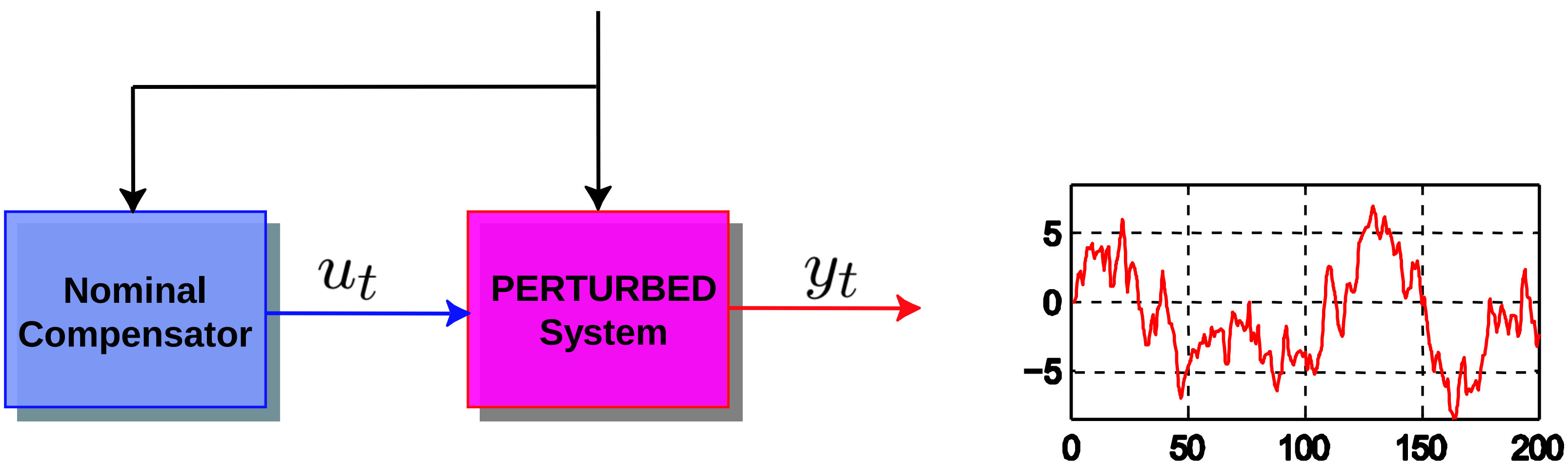
Example: feedforward noise compensation

system parameters unknown: $a, b, c, d \in \Delta$

Example: feedforward noise compensation

system parameters unknown: $a, b, c, d \in \Delta$

$$w_t = WN(0,1)$$



Example: feedforward noise compensation

scenario approach:

sample: $a_i, b_i, c_i, d_i \in \Delta, \quad i = 1, 2, \ldots, N;$

solve:

$$\min_{k_1,k_2} \left[\max_i \frac{(c_i + b_i k_1)^2 + (d_i + b_i k_2)^2 + 2a_i(c_i + b_i k_1)(d_i + b_i k_2)}{1 - a_i^2} \right]$$

Fundamental question:

how robust is ℓ^* ?

Fundamental question:

how robust is ℓ^* ?

that is: how guaranteed is ℓ^* against all $\delta \in \Delta$

Fundamental question:

how robust is ℓ^* ?

that is: how guaranteed is ℓ^* against all $\delta \in \Delta$

from the "visible" to the "invisible"

Theorem (with S. Garatti - G. Calafiore)

Fix $\epsilon \in (0,1)$ (robustness parameter)
$\beta \in (0,1)$ (confidence parameter)

If $N \geq N(\epsilon, \beta) \doteq \frac{2}{\epsilon}\left(\ln\frac{1}{\beta} + n_\theta\right)$,
then,
with probability $\geq 1 - \beta$,
ℓ^* is ϵ-level robust.

Theorem (with S. Garatti - G. Calafiore)

Fix $\epsilon \in (0,1)$ (robustness parameter)

If $N \geq N(\epsilon,\beta) \doteq \frac{2}{\epsilon}\left(\ln\frac{1}{\beta} + n_\theta\right)$,
then,

ℓ^* is ϵ-level robust.

Theorem (with S. Garatti - G. Calafiore)

Fix $\epsilon \in (0,1)$ (robustness parameter)
$\beta \in (0,1)$ (confidence parameter)

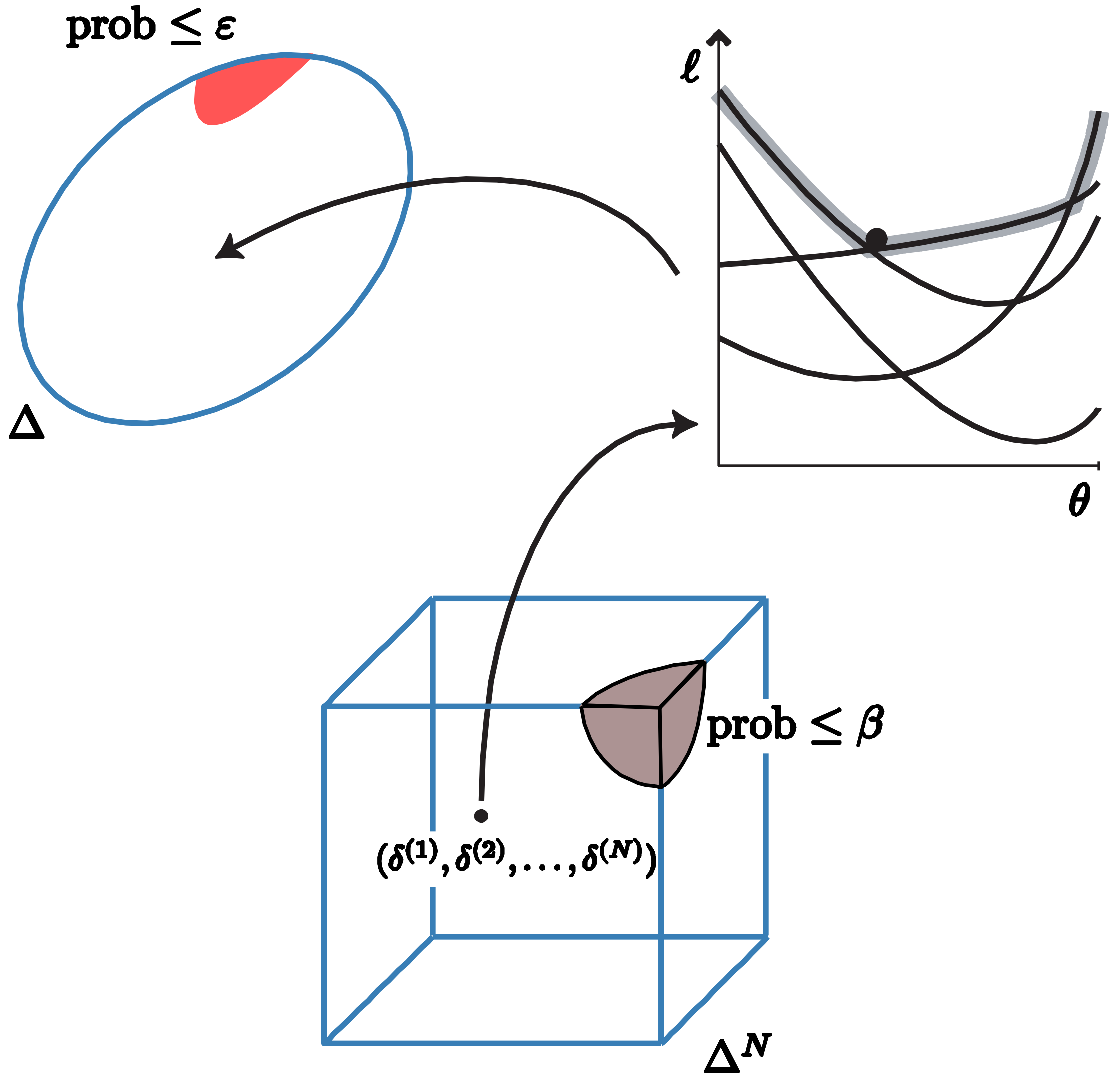
If $N \geq N(\epsilon,\beta) \doteq \frac{2}{\epsilon}\left(\ln\frac{1}{\beta} + n_\theta\right)$,
then,
with probability $\geq 1-\beta$,
ℓ^* is ϵ-level robust.

Theorem (with S. Garatti - G. Calafiore)

Fix $\epsilon \in (0, 1)$ (robustness parameter)

If $N \geq N(\epsilon) \doteq \frac{2}{\epsilon}(7 \ln 10 + n_\theta)$,
then,

ℓ^* is ϵ-level robust.



prob $\leq \varepsilon$
Δ
ℓ
θ
prob $\leq \beta$
$(\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(N)})$
Δ^N

Comments

generalization $\longrightarrow$ need for structure

Good news: the structure we need is only convexity

… more comments

$$N = \frac{2}{\epsilon}\left(\ln\frac{1}{\beta} + n_\theta\right)$$

- *N* often tractable by standard solvers
- *N* easy to compute
- *N* independent of *Pr*
- permits to address problems otherwise intractable

Ex: feedforward noise compensation

Example: feedforward noise compensation

$$\min_{k_1,k_2} var[y_t] = \frac{(c+bk_1)^2+(d+bk_2)^2+2a(c+bk_1)(d+bk_2)}{1-a^2}$$

Example: feedforward noise compensation

$$\min_{k_1,k_2} var[y_t] = \frac{(c+bk_1)^2+(d+bk_2)^2+2a(c+bk_1)(d+bk_2)}{1-a^2}$$

$$\Delta = \{a,b,c,d \;\; : \;\; a = \frac{3.5\sigma_1^2 - 0.2}{3\sigma_1^2 + 0.3} \cdot (0.32\sigma_1 + 0.6),$$

$$b = 1 + \frac{\sigma_1 \sigma_2^2}{10},$$

$$c = \frac{-0.01 + (\sigma_1 + \sigma_2^2)^2}{0.02 + (\sigma_1 + \sigma_2^2)^2} \cdot \left(1 - \frac{(\sigma_1 - 1)(\sigma_2 - 1)}{2}\right),$$

$$d = \frac{0.05}{0.025 + (\sigma_1 + \sigma_2 - 2)^2},$$

$$(\sigma_1, \sigma_2) \in [-1, 1]^2\}.$$

Example: feedforward noise compensation

$\varepsilon = 0.005 \quad \beta = 10^{-7} \quad \Rightarrow \quad N = 5427$

Example: feedforward noise compensation

$$\varepsilon = 0.005 \quad \beta = 10^{-7} \quad \Rightarrow \quad N = 5427$$

sample: $a_i, b_i, c_i, d_i \in \Delta, \quad i = 1, 2, \ldots, 5427;$

solve:

$$\min_{k_1,k_2} \left[\max_i \frac{(c_i + b_i k_1)^2 + (d_i + b_i k_2)^2 + 2a_i(c_i + b_i k_1)(d_i + b_i k_2)}{1 - a_i^2} \right]$$

Example: feedforward noise compensation

$\varepsilon = 0.005 \quad \beta = 10^{-7}$ ⇒ $N = 5427$

sample: $a_i, b_i, c_i, d_i \in \Delta, \quad i = 1, 2, \ldots, 5427;$

solve:

$$\min_{k_1,k_2} \left[\max_i \frac{(c_i + b_i k_1)^2 + (d_i + b_i k_2)^2 + 2a_i (c_i + b_i k_1)(d_i + b_i k_2)}{1 - a_i^2} \right]$$

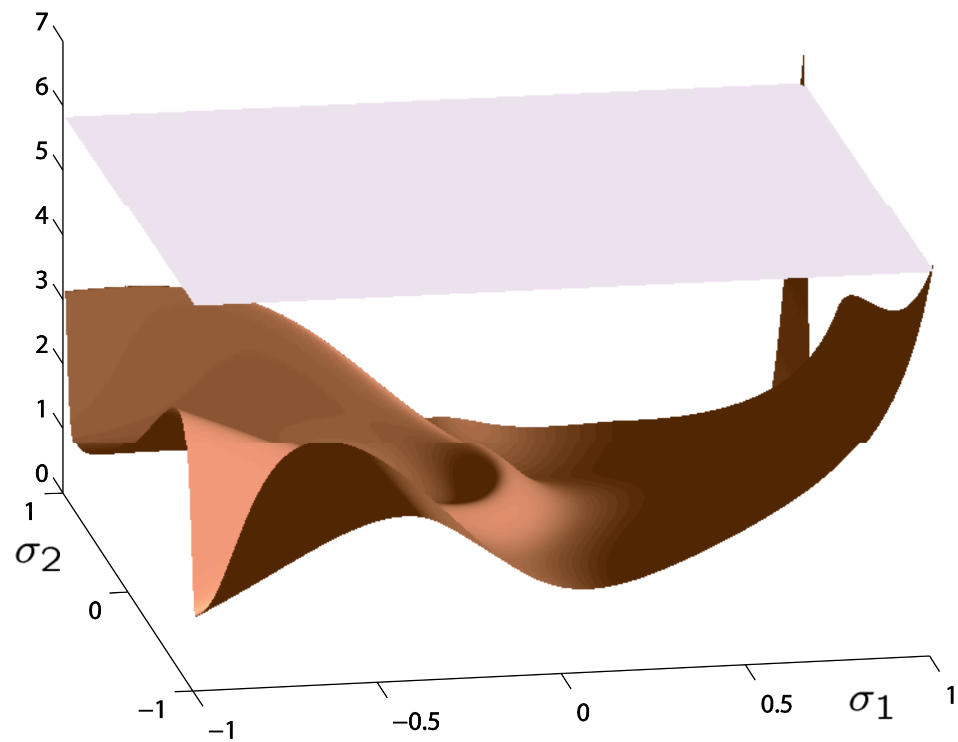
⇒ $k_1^* = -0.9022, \quad k_2^* = -0.9028, \quad \ell^* = 5.8$

Example: feedforward noise compensation

$\ell^* = 5.8$ ⇨ **Output variance below 5.8 for all plants but a small fraction (ε = 0.5%)**

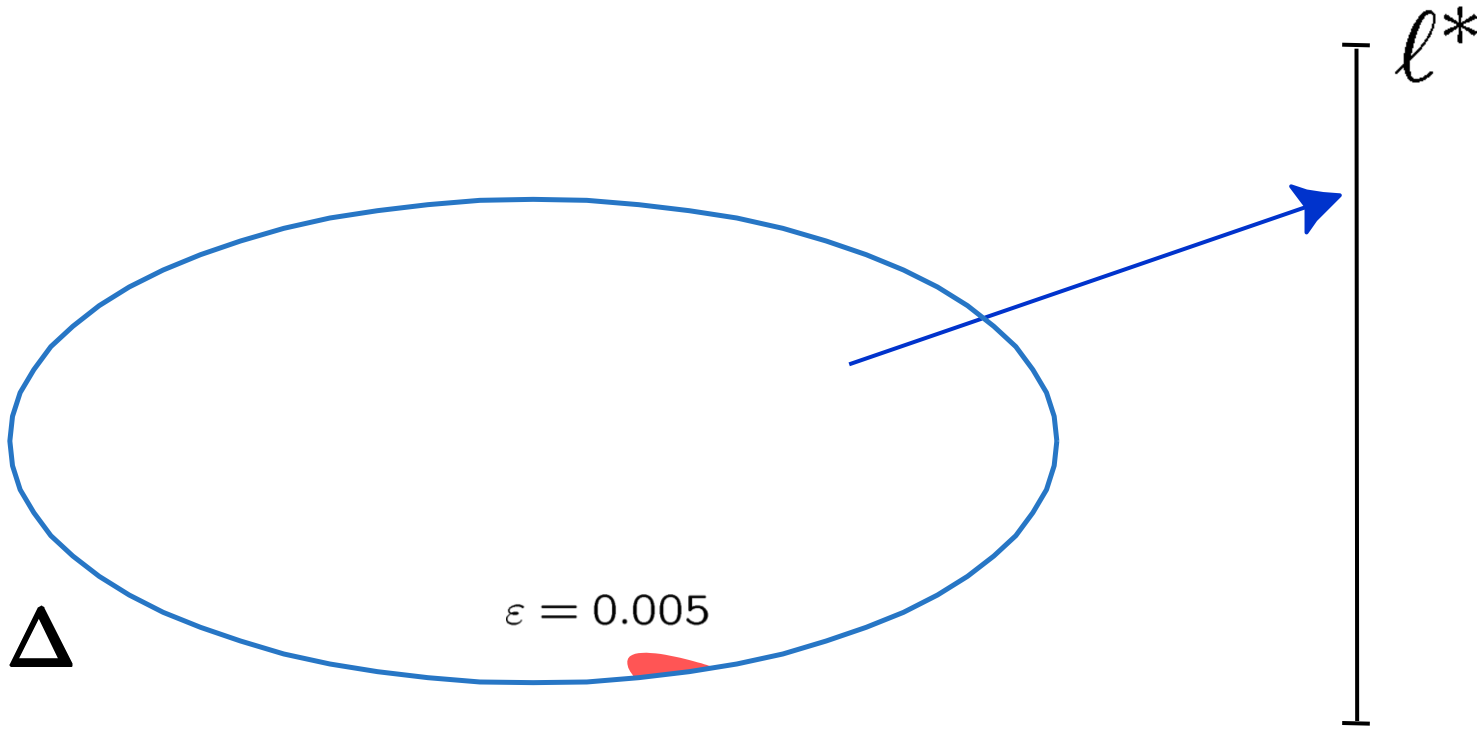
Example: feedforward noise compensation

$\ell^* = 5.8$ → **Output variance below 5.8 for all plants but a small fraction (ε = 0.5%)**

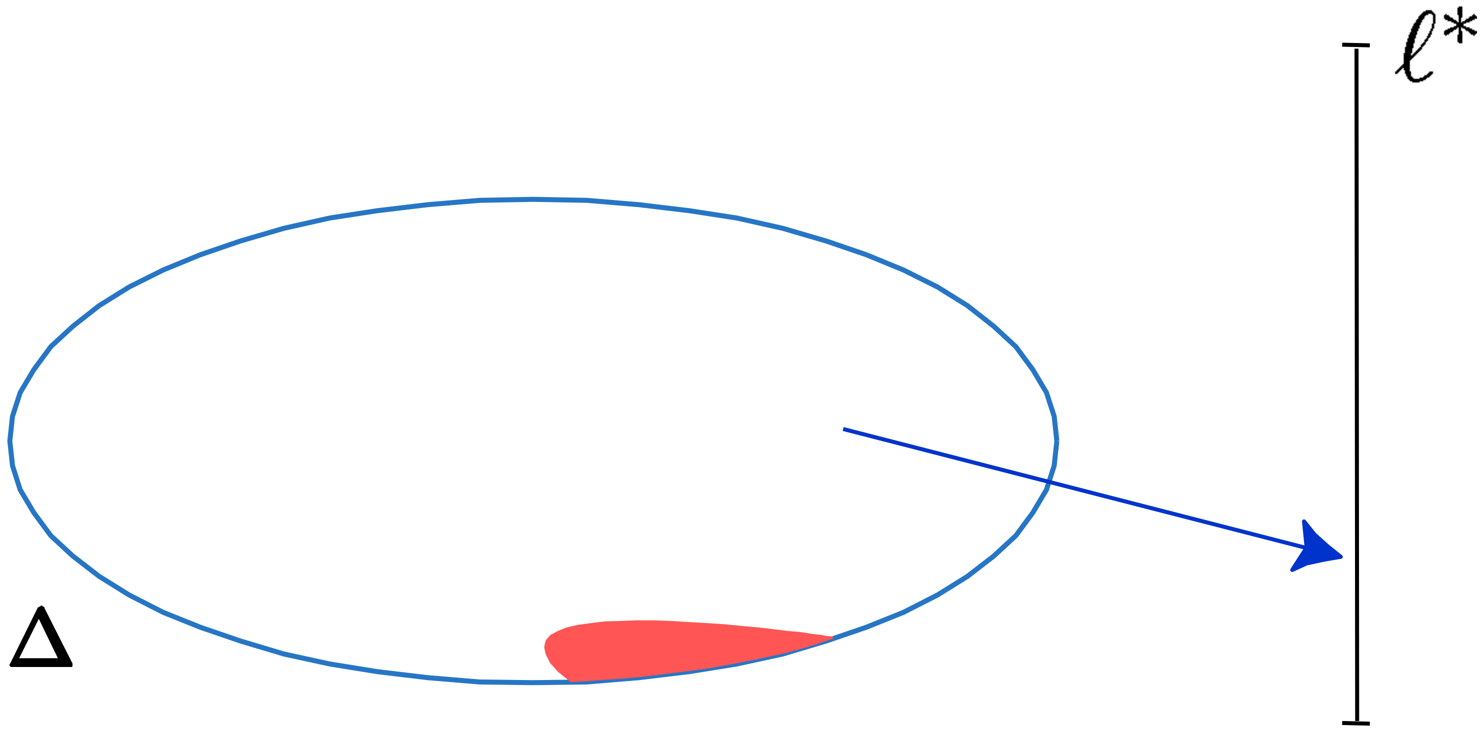


performance profile

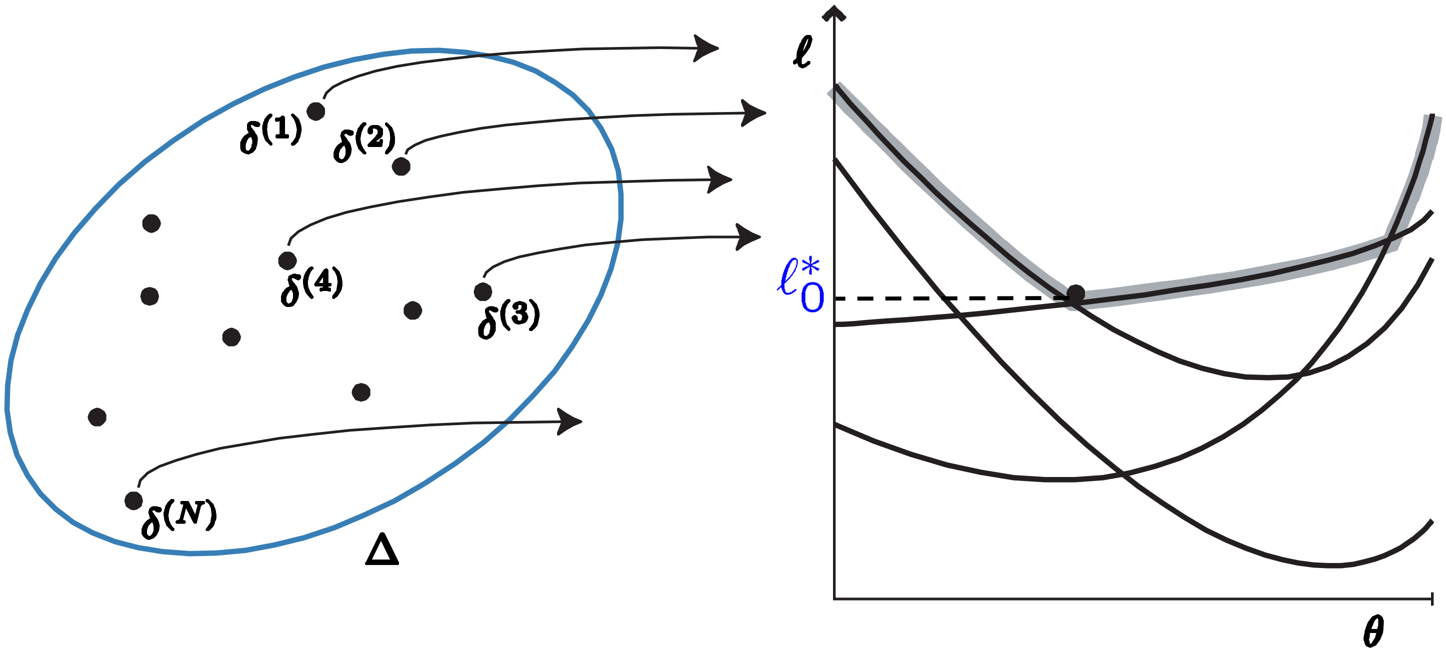
Variable Robustness Control



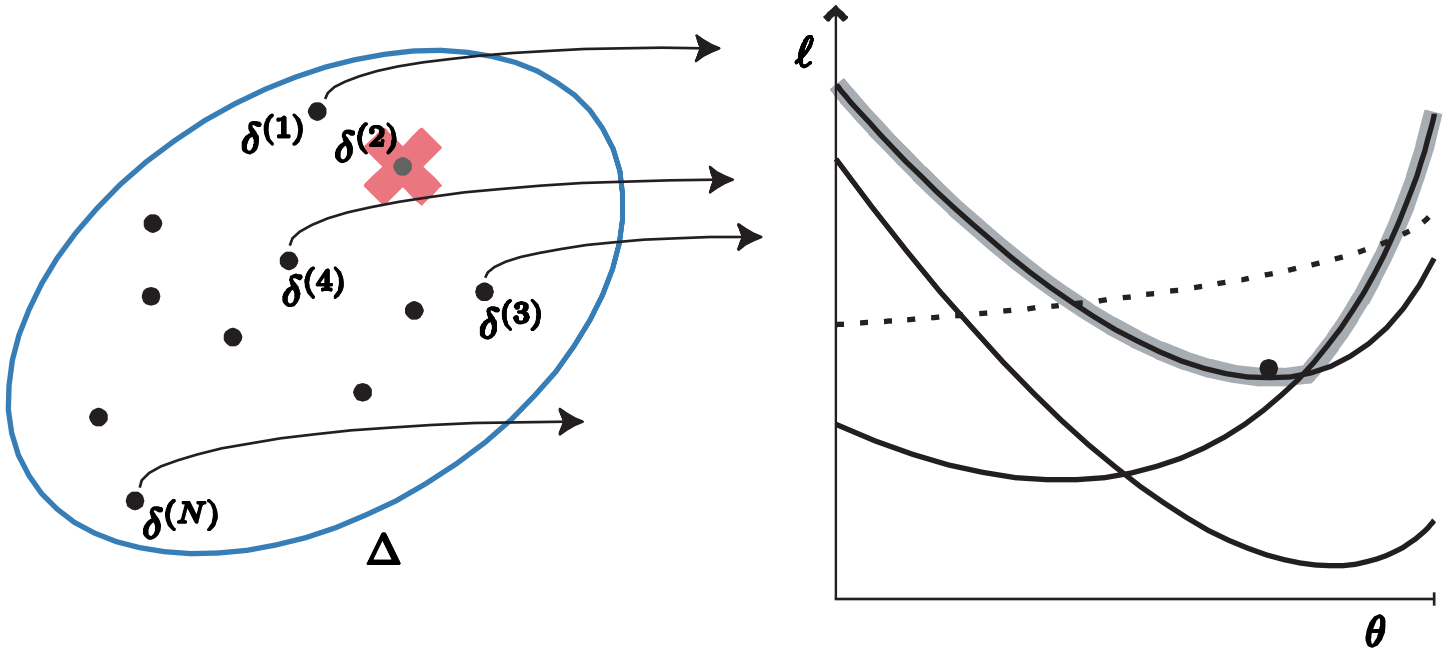
Variable Robustness Control



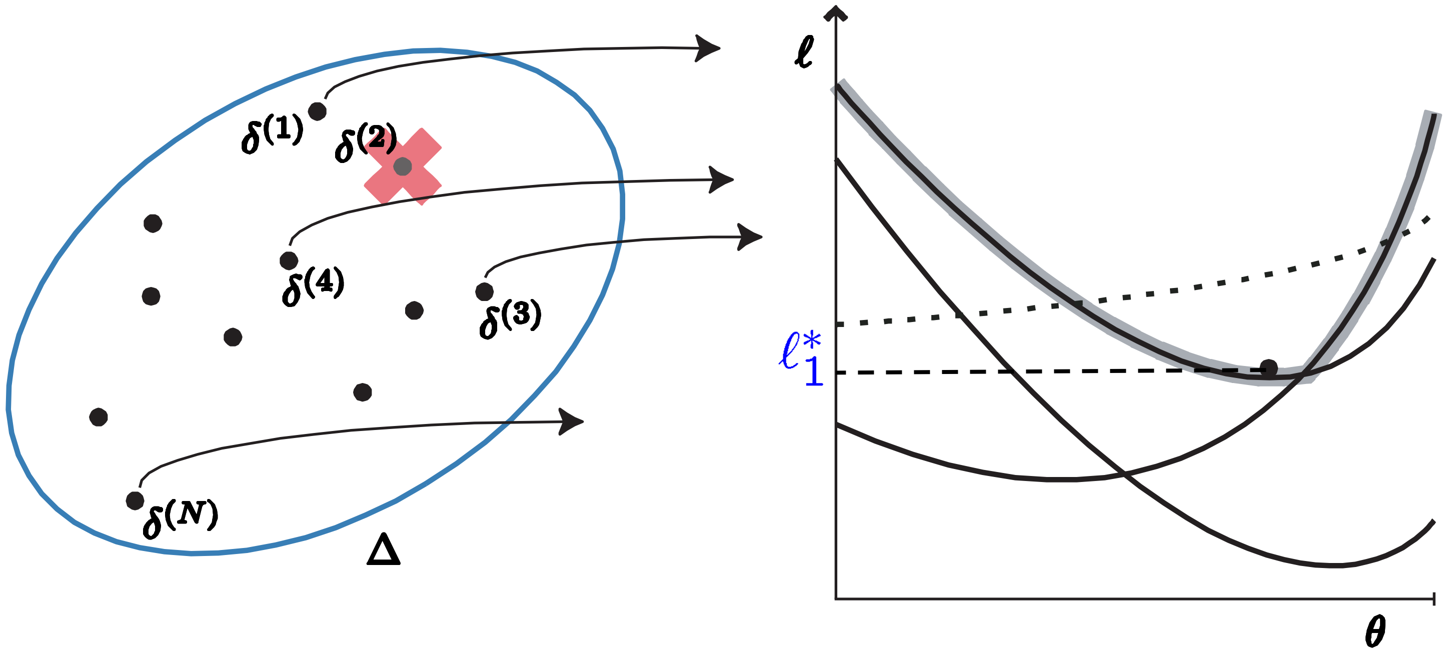
Variable Robustness Control



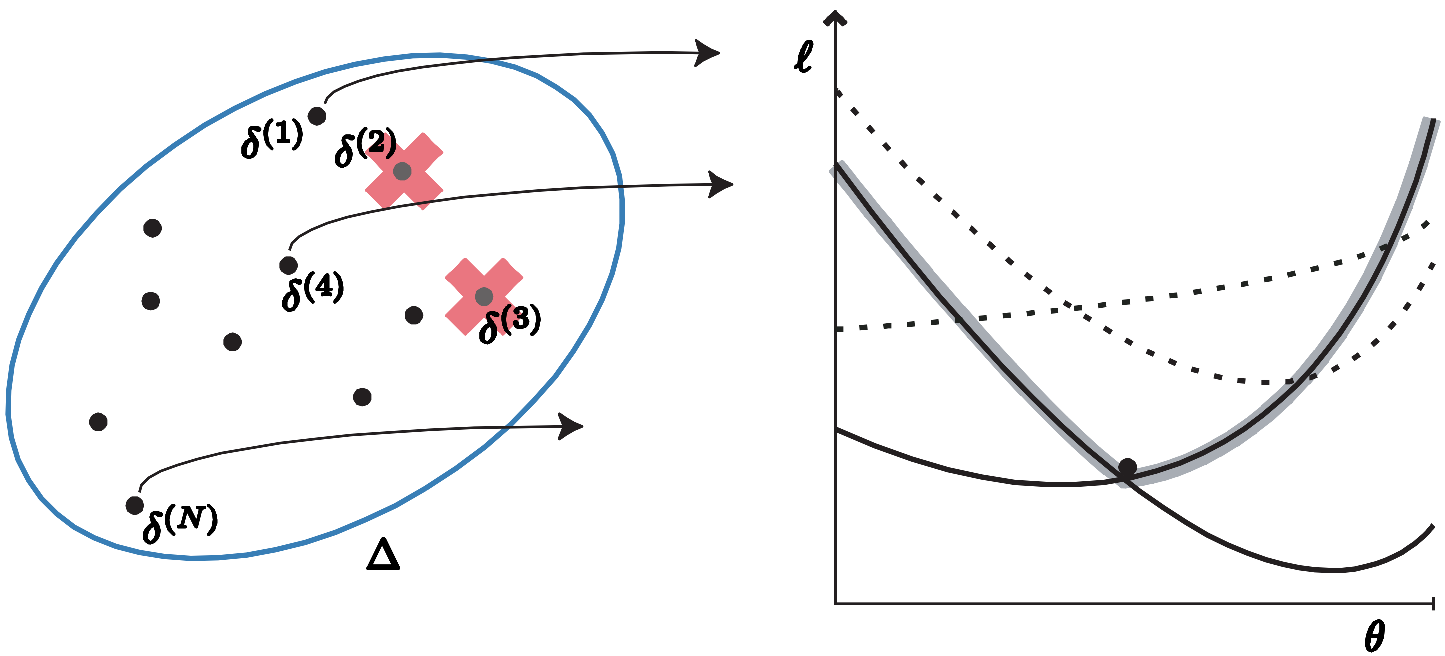
Variable Robustness Control



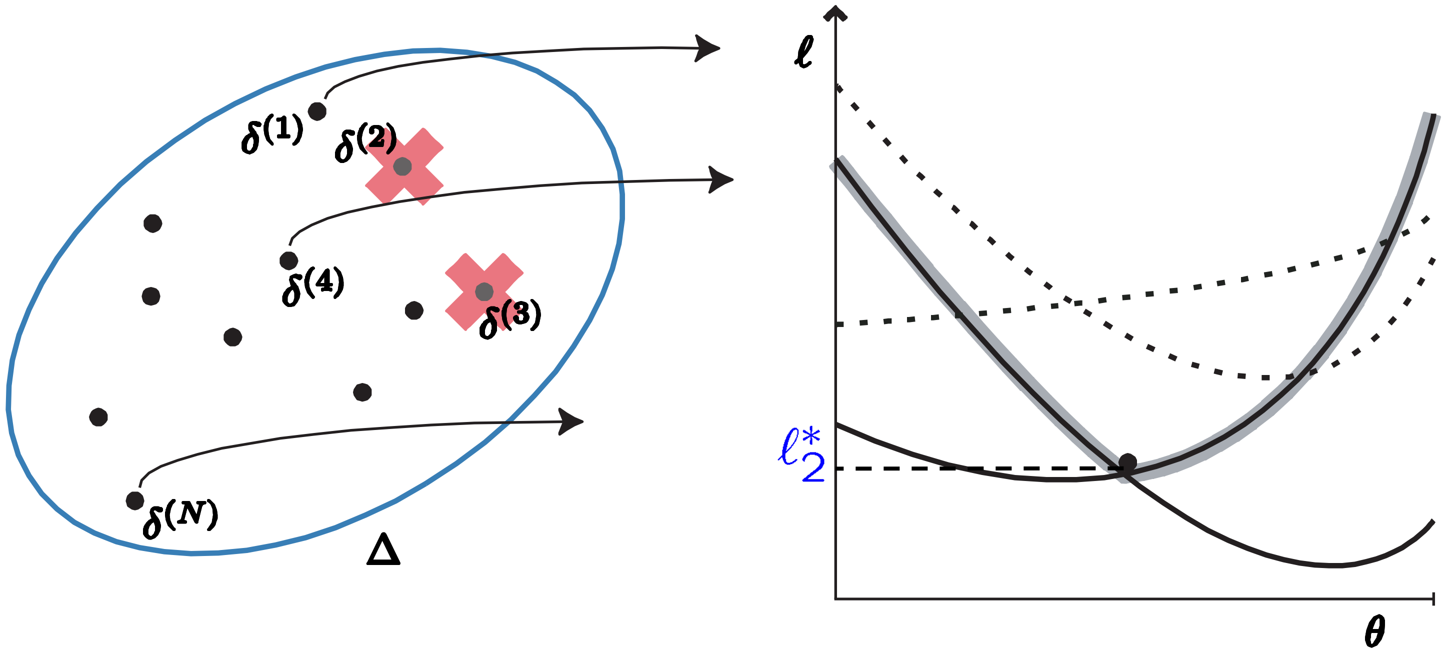
Variable Robustness Control

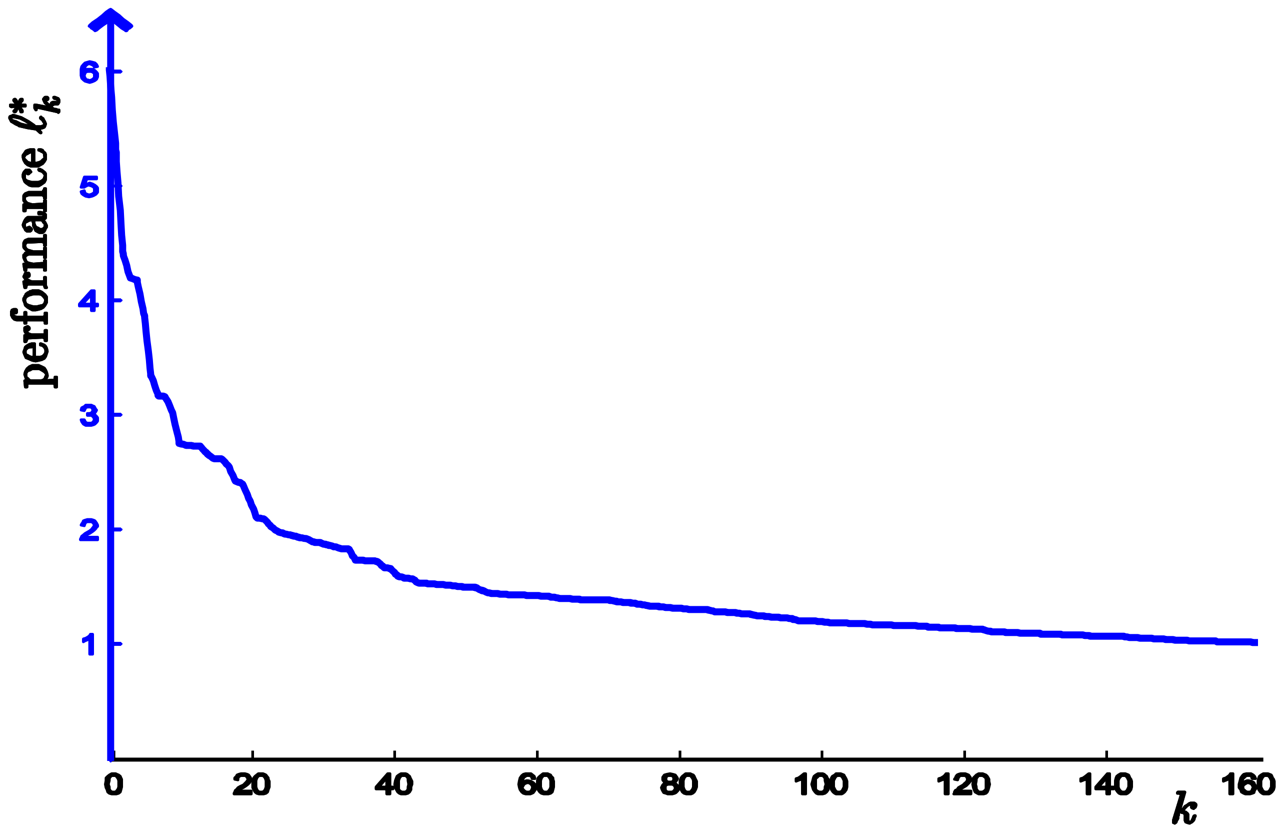


Variable Robustness Control



Variable Robustness Control




performance ℓ_k^*
6
5
4
3
2
1
0
20
40
60
80
100
120
140
160
k

Theorem (with S. Garatti)

$$N \geq N(\epsilon, \beta) \doteq \frac{2}{\epsilon}\left(\ln \frac{1}{\beta} + n_\theta\right).$$

Then, ℓ_k^* is ϵ_k-level robust where:

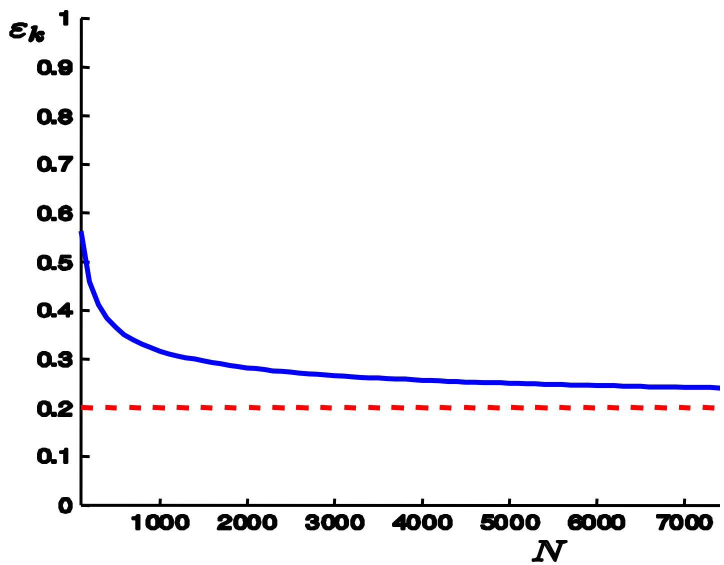
$$\epsilon_k = \frac{k}{N} + O\left(\frac{1}{\sqrt{N}}\right)$$

Theorem (with S. Garatti)

$$N \geq N(\epsilon, \beta) \doteq \frac{2}{\epsilon}\left(\ln\frac{1}{\beta} + n_\theta\right).$$

Then, ℓ_k^* is ϵ_k-level robust where:

$$\epsilon_k = \frac{k}{N} + O\left(\frac{1}{\sqrt{N}}\right)$$



Comments

- the result does not depend on the algorithm for eliminating k constraints

Comments

- the result does not depend on the algorithm for eliminating k constraints

… do it greedy

Comments

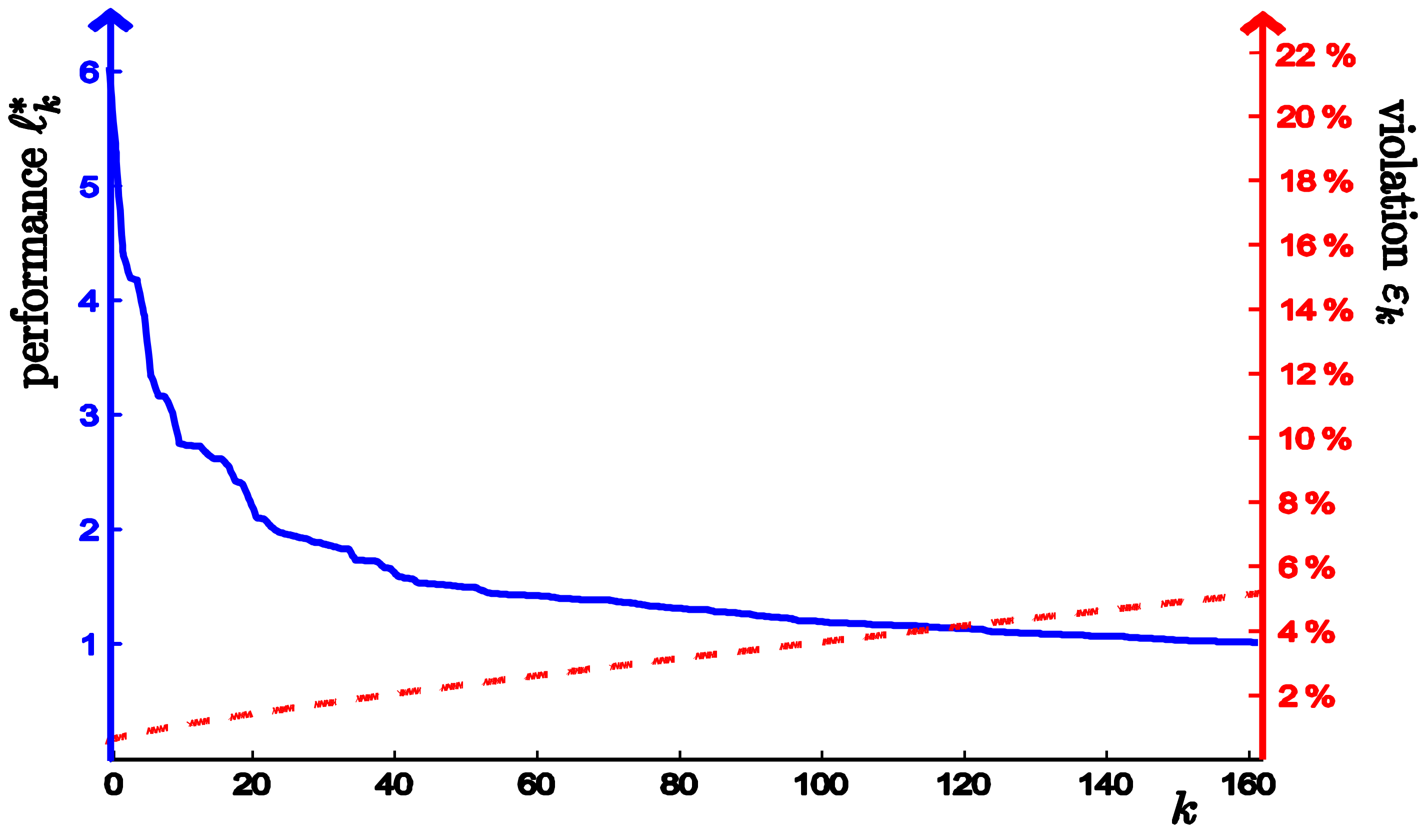
- the result does not depend on the algorithm for eliminating k constraints

  … do it greedy

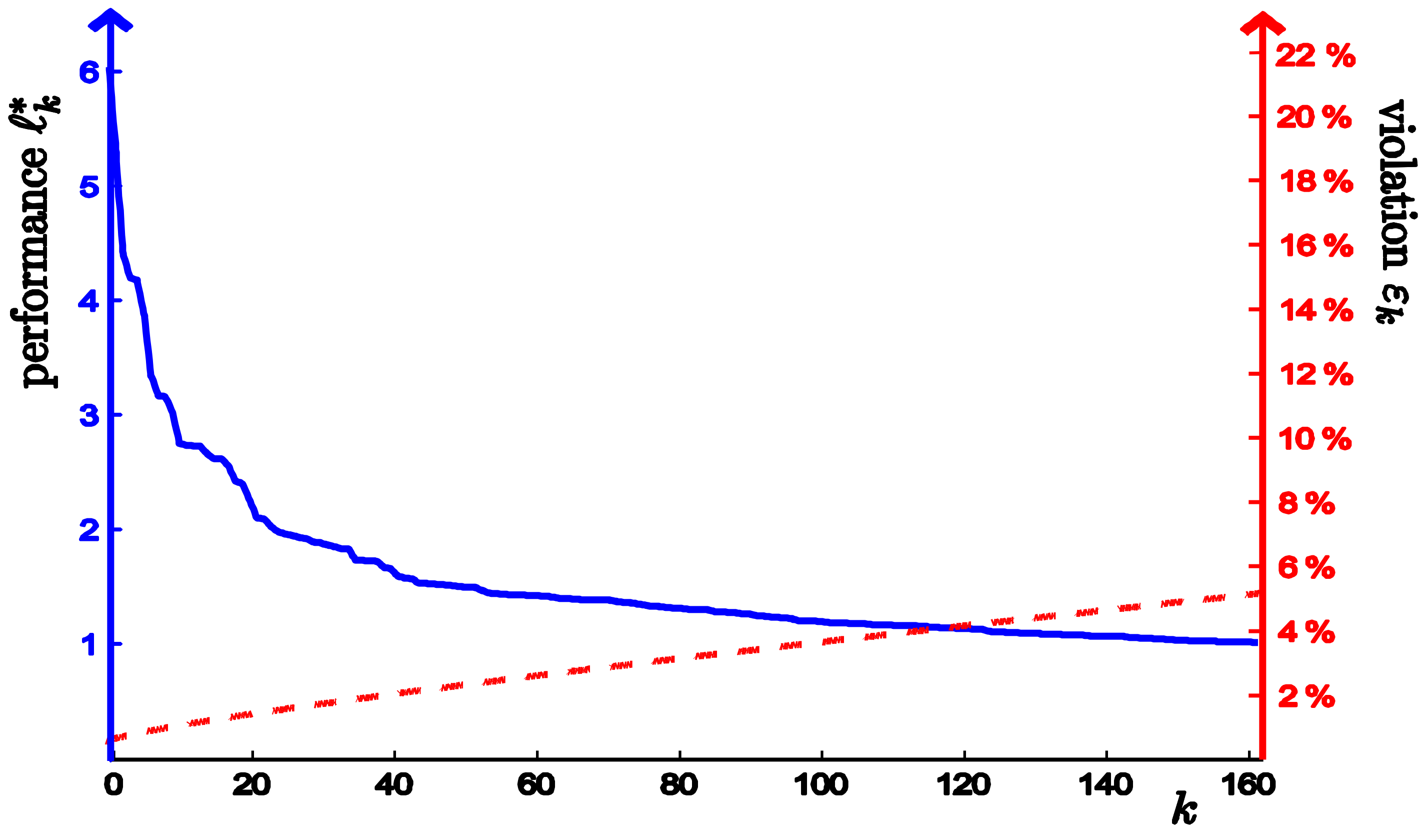
- value can be inspected

  violation probability is guaranteed by the theorem

performance - violation plot



Example: feedforward noise compensation



Example: feedforward noise compensation

sample: $a_i, b_i, c_i, d_i \in \Delta, \quad i = 1, 2, \ldots, 5427;$

solve:

$$\min_{k_1,k_2} \left[\max_i \frac{(c_i + b_i k_1)^2 + (d_i + b_i k_2)^2 + 2a_i(c_i + b_i k_1)(d_i + b_i k_2)}{1 - a_i^2} \right]$$

Example: feedforward noise compensation

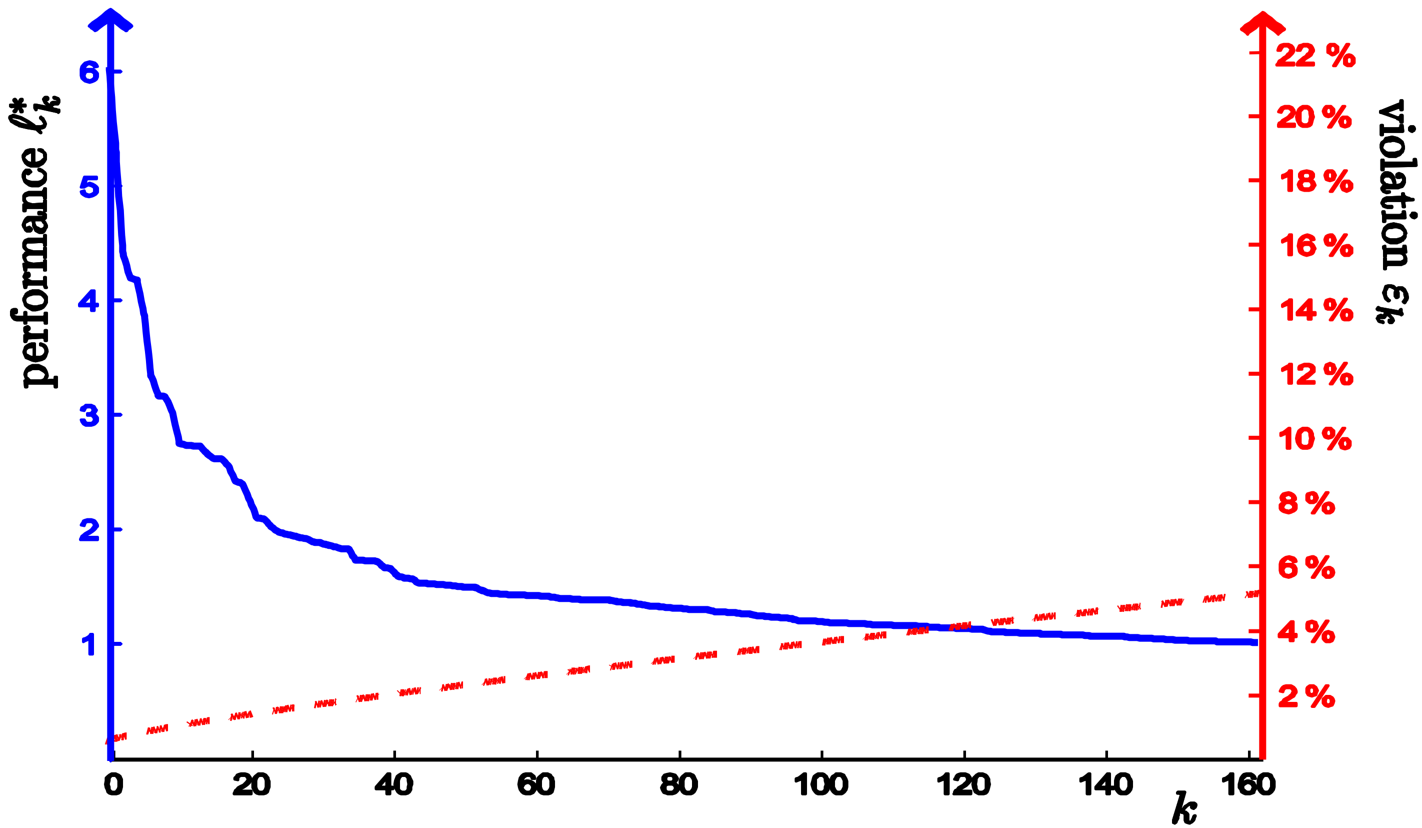
sample: $a_i, b_i, c_i, d_i \in \Delta, \quad i = 1, 2, \ldots, 5427;$

solve:

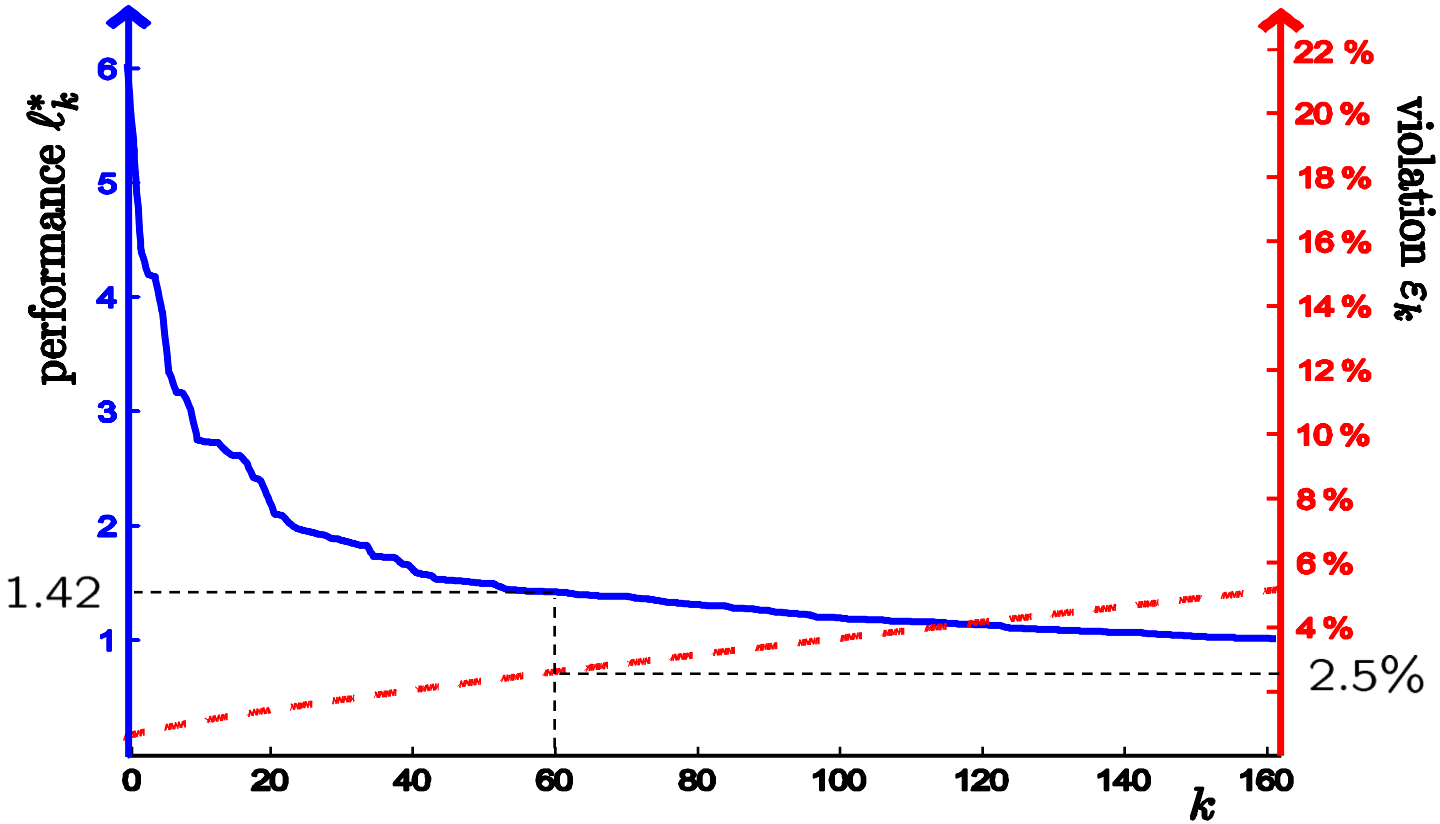
$$\min_{k_1,k_2} \left[\max_i \frac{(c_i + b_i k_1)^2 + (d_i + b_i k_2)^2 + 2a_i(c_i + b_i k_1)(d_i + b_i k_2)}{1 - a_i^2} \right]$$

⇨ eliminate $k = 1, 2, \ldots$ constraints

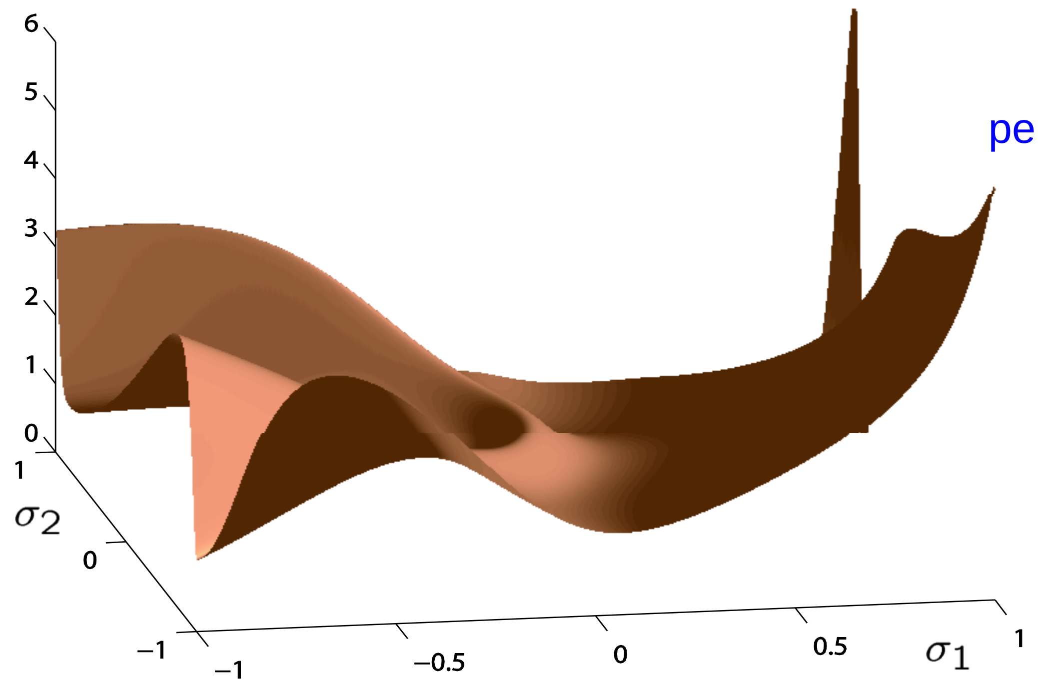
Example: feedforward noise compensation



Example: feedforward noise compensation

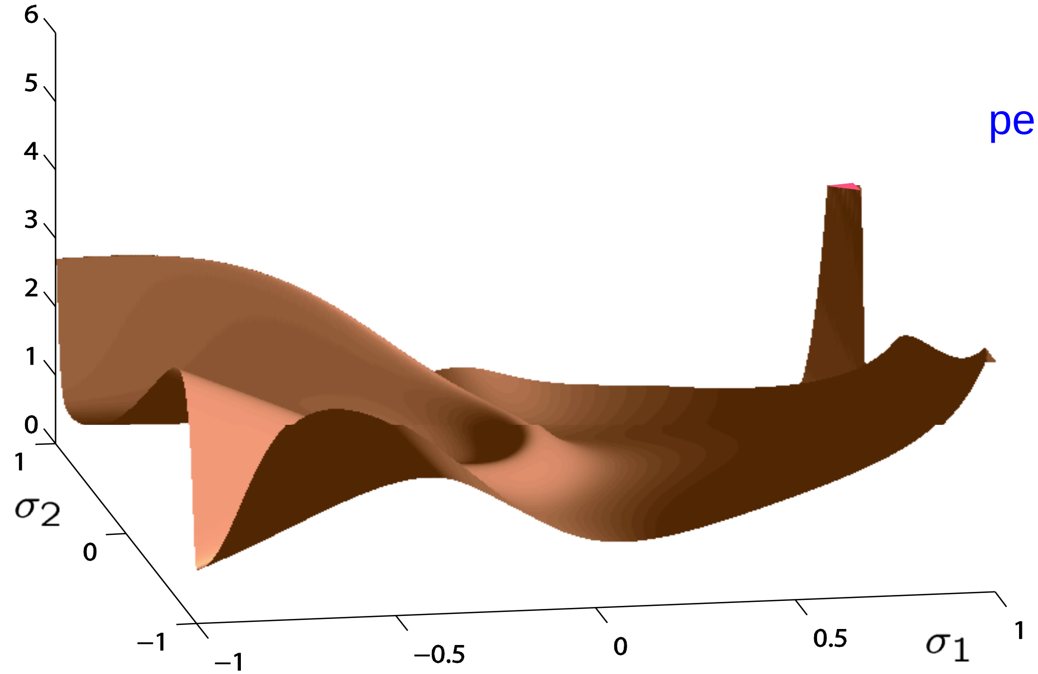


Example: feedforward noise compensation



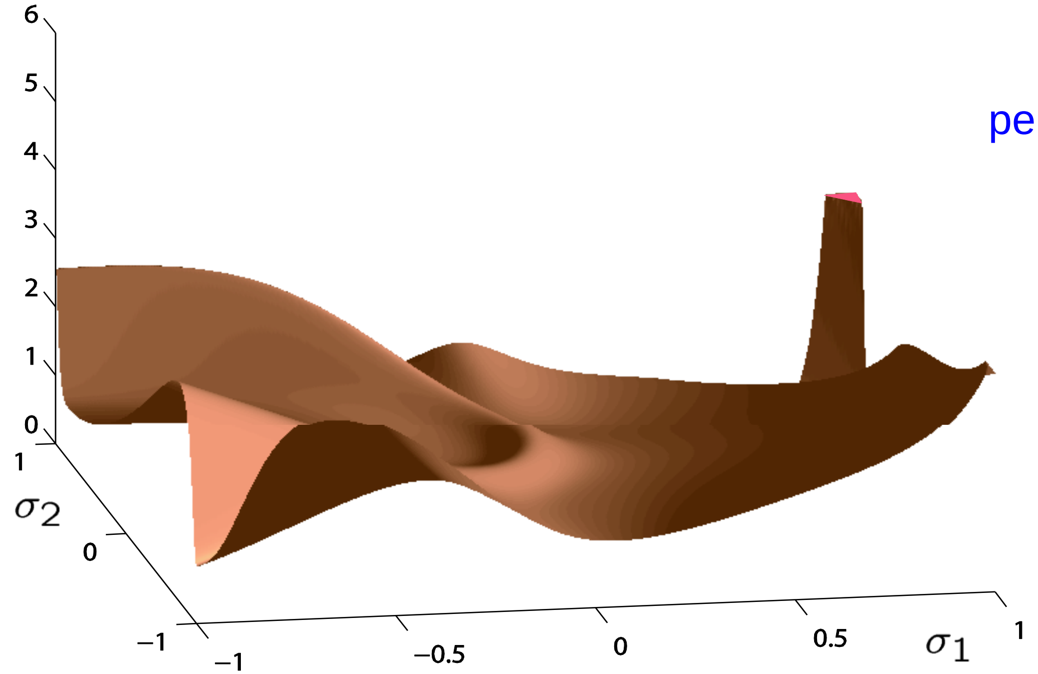
performance profile

Example: feedforward noise compensation



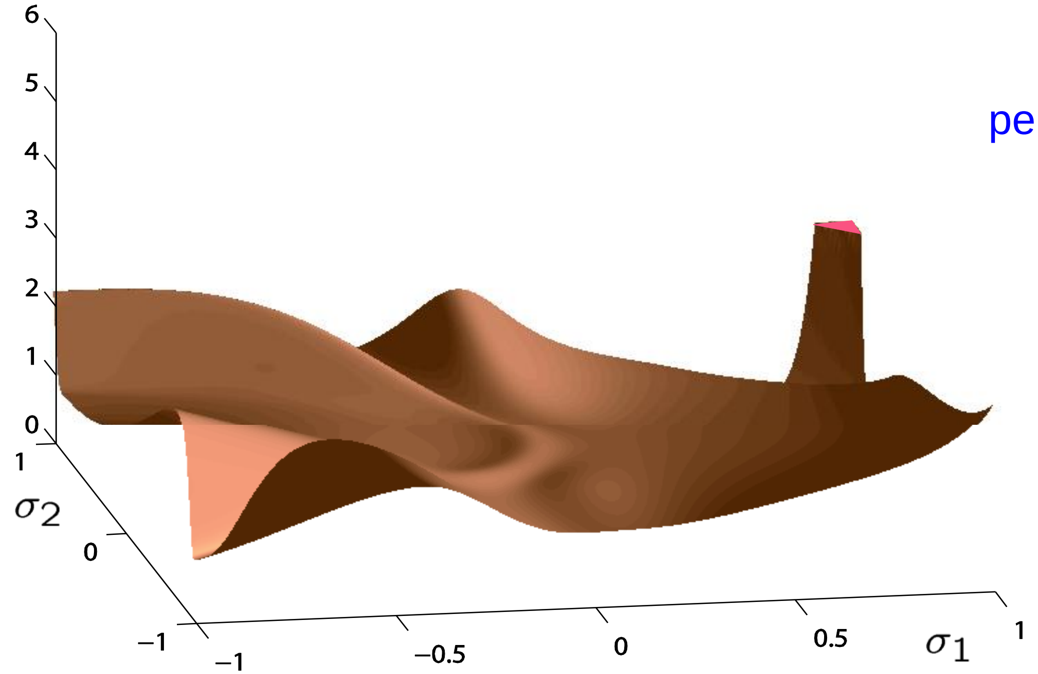
performance profile

Example: feedforward noise compensation



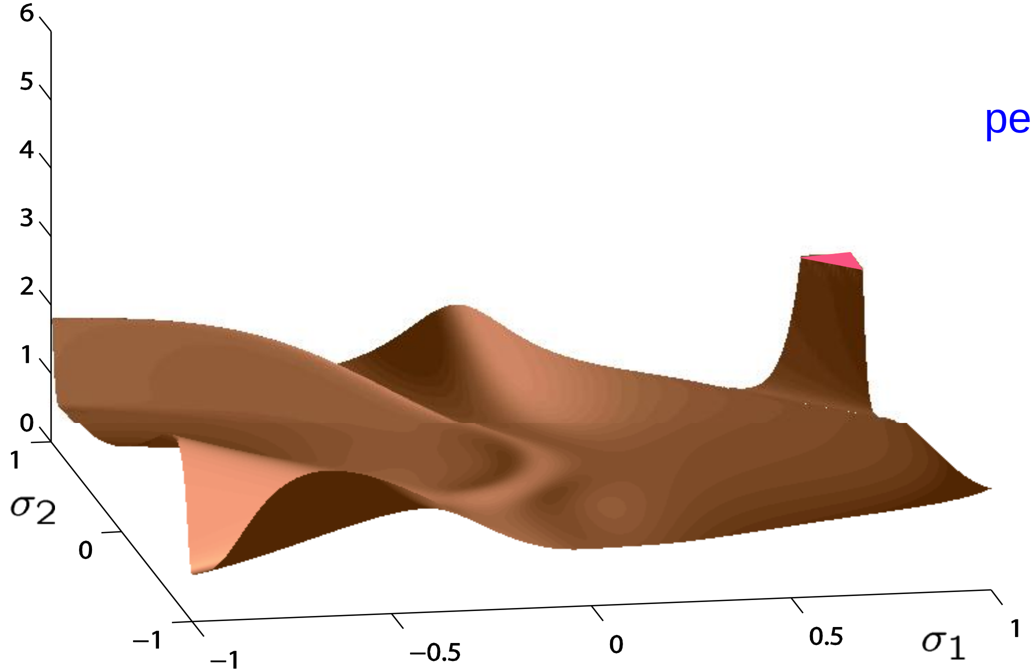
performance profile

Example: feedforward noise compensation



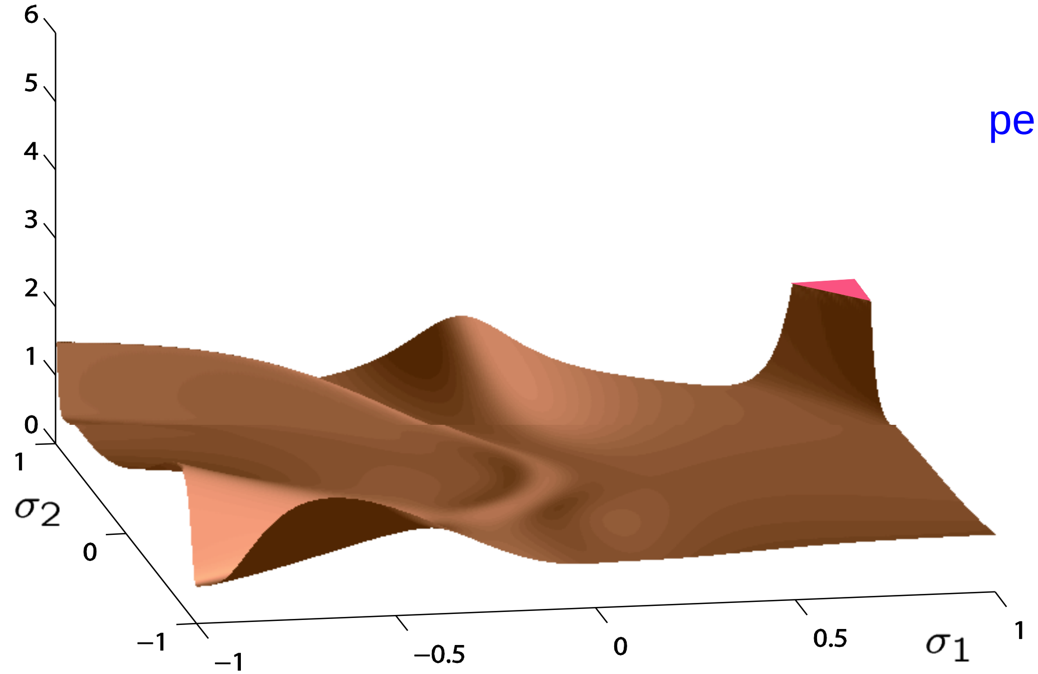
performance profile

Example: feedforward noise compensation



performance profile

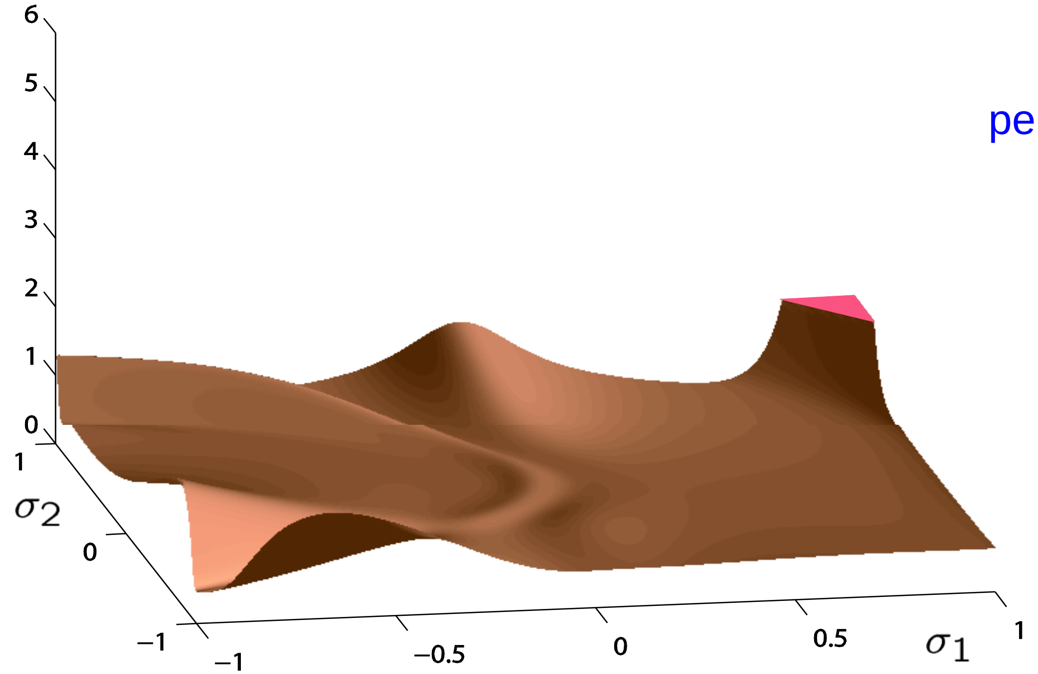
Example: feedforward noise compensation



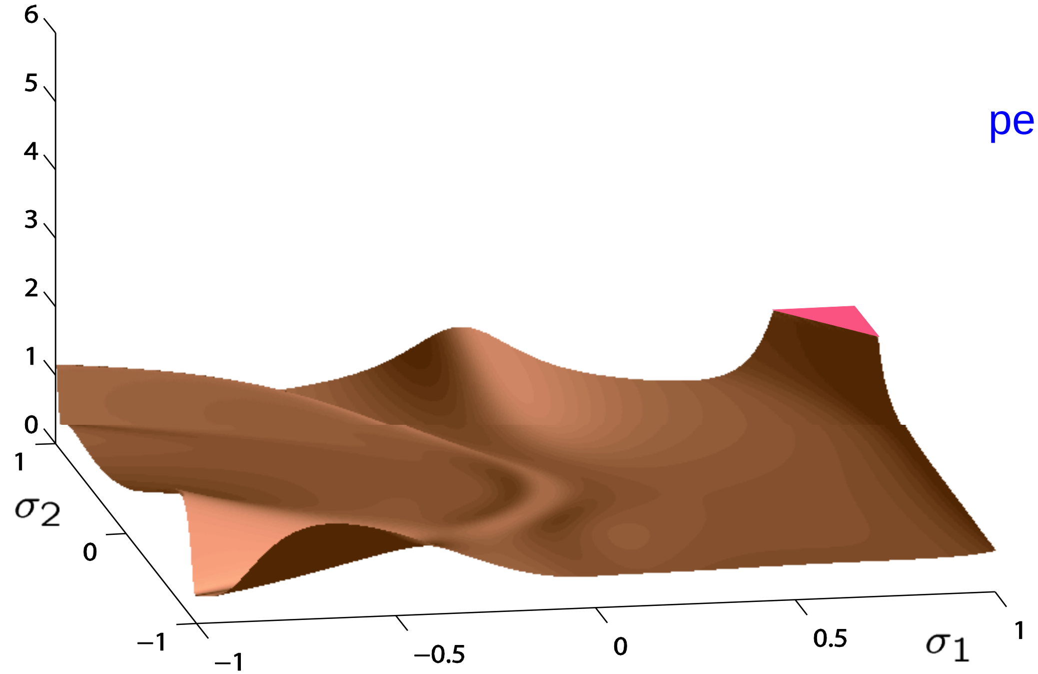
performance profile

Example: feedforward noise compensation

performance profile

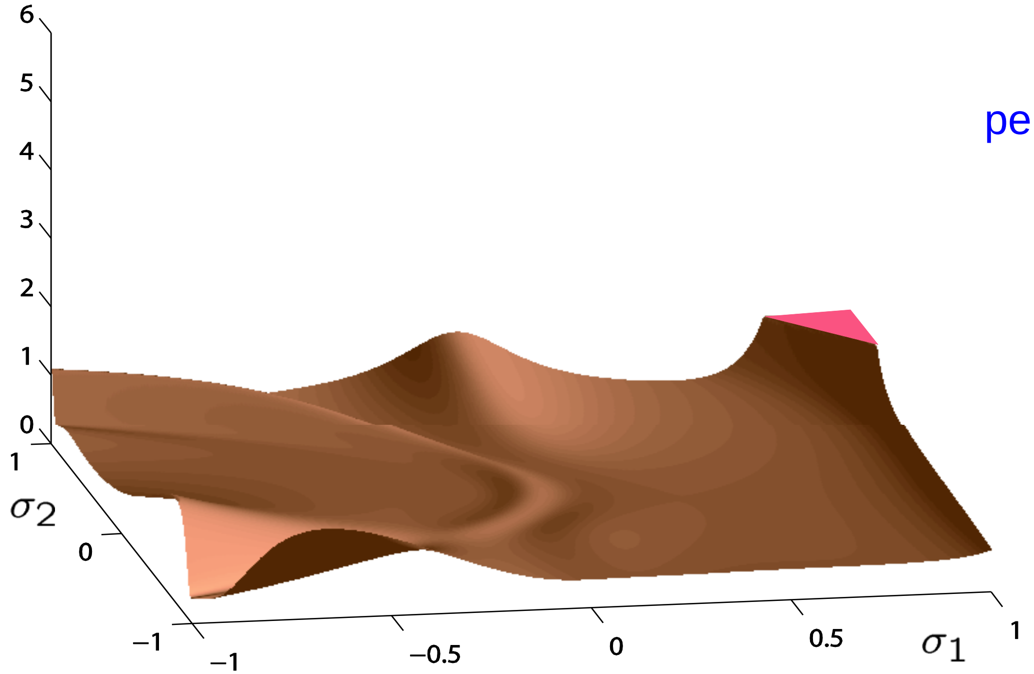


Example: feedforward noise compensation



performance profile

Example: feedforward noise compensation



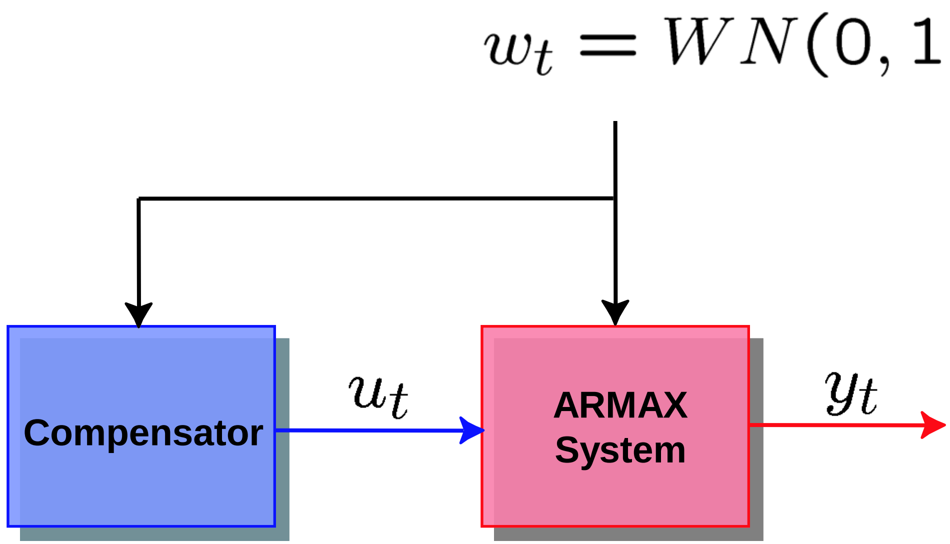
performance profile

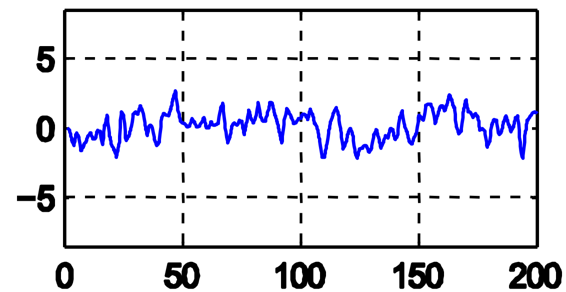
$k = 60$

$l^*_{60} = 1.42$

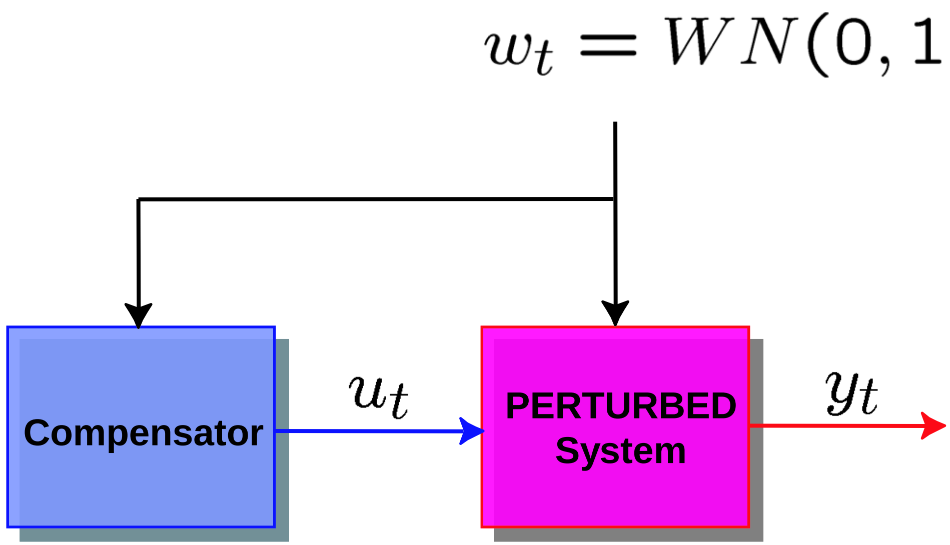
$\epsilon_{60} = 2.5\%$

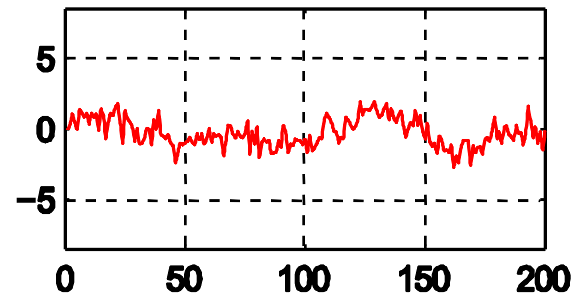
Example: feedforward noise compensation





Example: feedforward noise compensation





Conclusions

The VRC approach is a very general tool to trade robustness for performance

Conclusions

The VRC approach is a very general tool to trade robustness for performance

It is based on a solid and deep theory, but its practical use is very simple

Conclusions

The VRC approach is a very general tool to trade robustness for performance

It is based on a solid and deep theory, but its practical use is very simple

Applications in:

- prediction
- robust control
- engineering
- finance

⋮

REFERENCES

M.C. Campi and S. Garatti.
Variable Robustness Control: Principles and Algorithms.
Proceedings MTNS, 2010.

M.C. Campi and S. Garatti.
The Exact Feasibility of Randomized Solutions of Uncertain Convex Programs.
SIAM J. on Optimization, 19, no.3: 1211-1230, 2008.

G. Calafiore and M.C. Campi.
Uncertain Convex Programs: randomized Solutions and Confidence Levels.
Mathematical Programming, 102: 25-46, 2005.

G. Calafiore and M.C. Campi.
The Scenario Approach to Robust Control Design.
IEEE Trans. on Automatic Control, AC-51: 742-753, 2006.