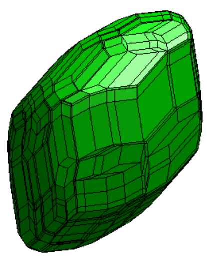
Rencontre du CIRM: Modélisation, Optimisation et Analyse statique
(6-10 décembre 2010)

Quelques Algorithmes de Calcul d'Enveloppes à base de Zonotopes



Christophe COMBASTEL

ENSEA

Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

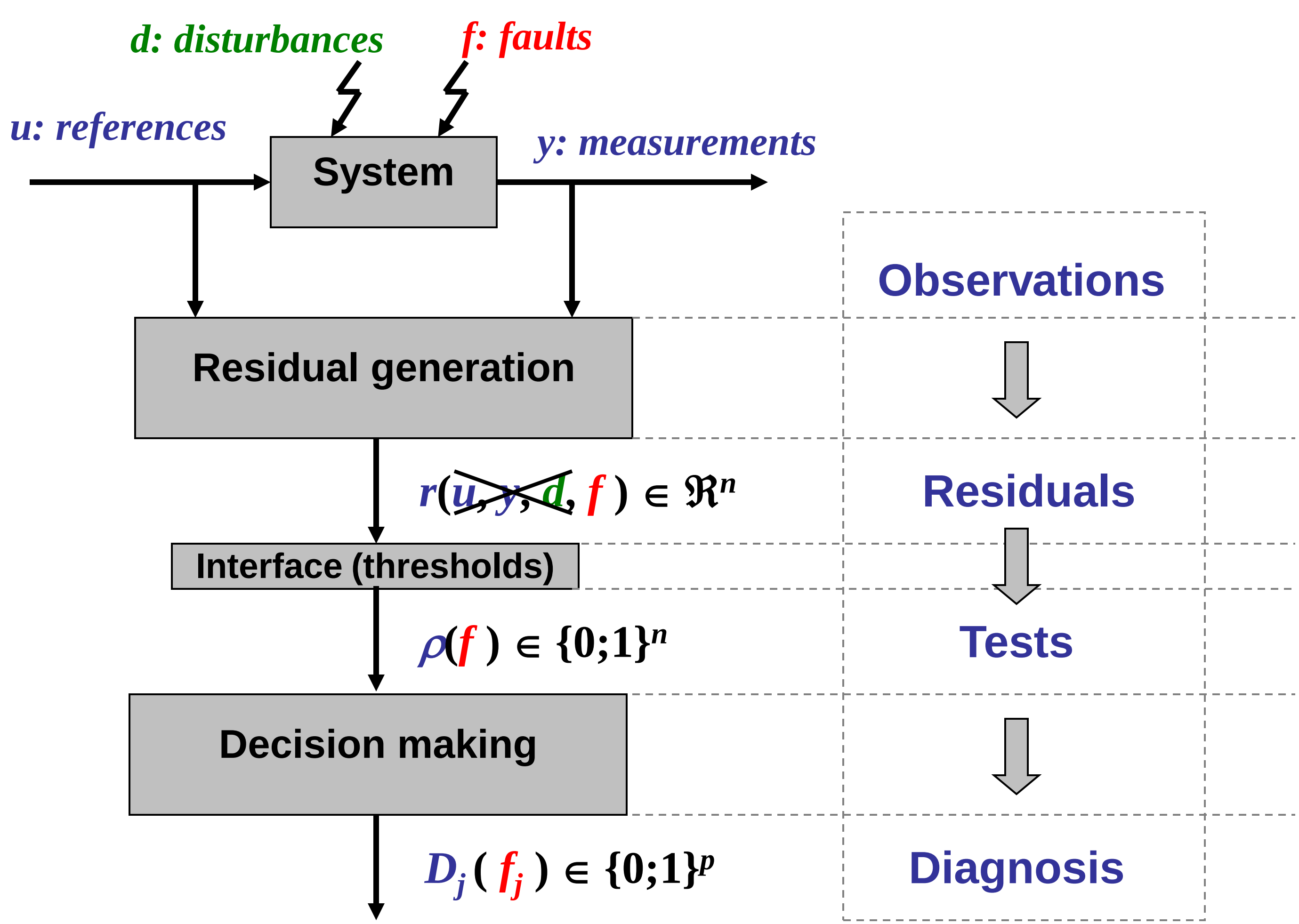
4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

« FDI » usual scheme

- **Fault diagnosis = Detection, Isolation, Identification of faults**



Known value
Unknown value
Tested value
Decoupling

$$\begin{cases} \dot{x} = A.x + B.u + E.d + F.f \\ y = C.x + D.u + G.d + H.f \end{cases}$$

Observers,
Parity space,…

	f_1	f_2	f_3
ρ_1	1	1	1
ρ_2	1	0	1
ρ_3	1	1	0

Uncertainties

- **Observation and Fault diagnosis :**

Data (*corrupted*)	+	**Knowledge** (*partial*)	➔	**Information** (*imprecise*)

Measurement (*noise*)	+	**Model** (*modeling errors*, *disturbances*)	➔	**State** (?)

- **Classification according to how uncertainties are dealt with:**
  - **Not explicit** **(Luenberger observers)**
  - **Stochastic context** **(Kalman filters)**
  - **Deterministic context** ➔ **Set-membership approaches**

Key Points of Set-Membership Approaches

- Computing domains (related to the modes to be tested)

- Interesting to solve the choice of thresholds (from the specifications about uncertainty bounds, tolerances, etc…)

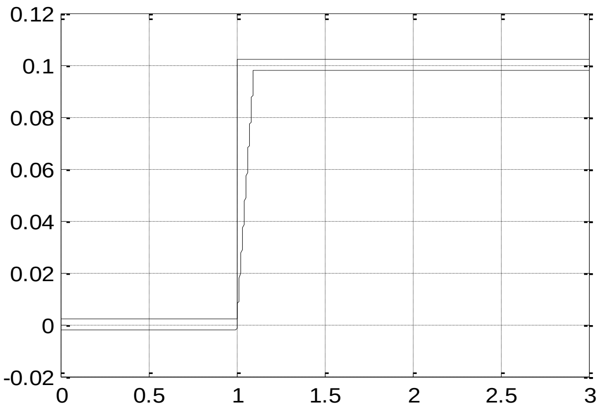
- Advantage: logically sound interface between the specifications and the decisions

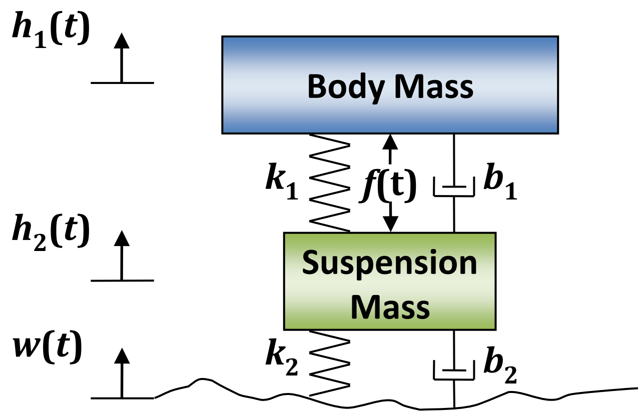
- Difficulty: efficient computation of the propagation of uncertainties in dynamical systems (fast computations, low conservatism).

Verified Model-based Design (vehicle suspension)

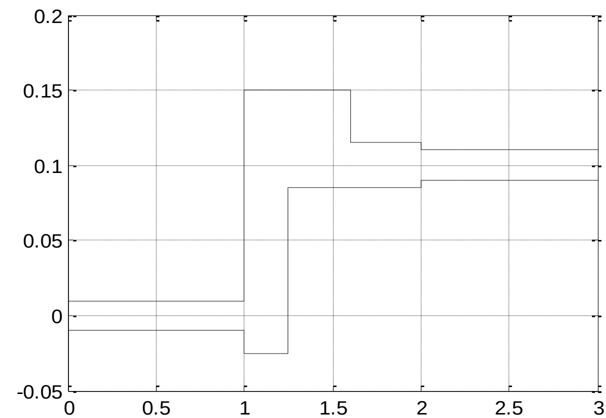
- **Computation of “guaranteed” envelopes in the design process:**

$[w^-(t), w^+(t)]$





$[\sigma^-(t), \sigma^+(t)]$



$$\dot{x}(t) = Ax(t) + \phi(t)$$
$$\phi(t) = u(t) + Z(t)s(t)$$
$$s(t) \in [-1, +1]^p, \ \ \forall t \in \mathbb{R}^+$$

- **Monte-Carlo simulations ➔ Inner approximations**
- **Verification of safety properties ➔ Need for outer approximations…**

… to achieve a full coverage of the specified scenarios.

Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

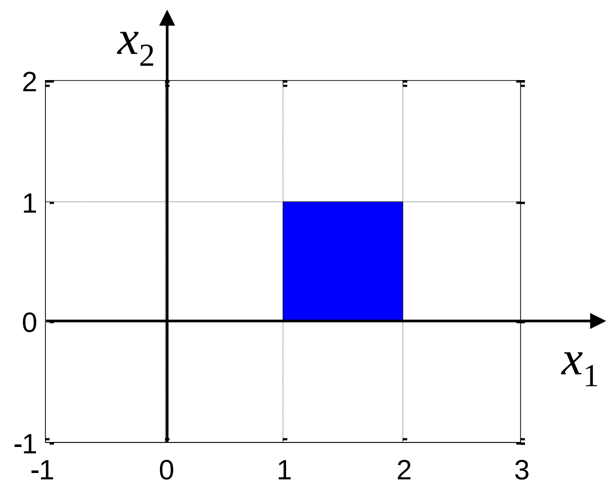
Interval arithmetic

- **Domain:** $x \in [x]$
- **Interval:** $[x] = [x_m, x_M] = \{\, x \mid x_m \leq x \leq x_M \,\}$
- **Usual operators:** $\forall \bullet \in \{+,-,\times,/\},\ [x] \bullet [y] = \{\, x \bullet y \mid x \in [x], y \in [y] \,\}$

$$[x] + [y] = [x_m + y_m,\ x_M + y_M]$$
$$[x] \times [y] = [min(x_m y_m,\ x_m y_M,\ x_M y_m,\ x_M y_M),\ max(x_m y_m,\ x_m y_M,\ x_M y_m,\ x_M y_M)]$$

- **Interval vector = aligned box**

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \begin{bmatrix} [1,2] \\ [0,1] \end{bmatrix} \Rightarrow$$



The dependency problem

- **The natural interval extension is an inclusion function …**
- **… but an often (very) pessimistic one:**

- **Example:** $x \in [-1;+1]$,

Implicitly assumed to be independent, even though depending on the same variable: *x*

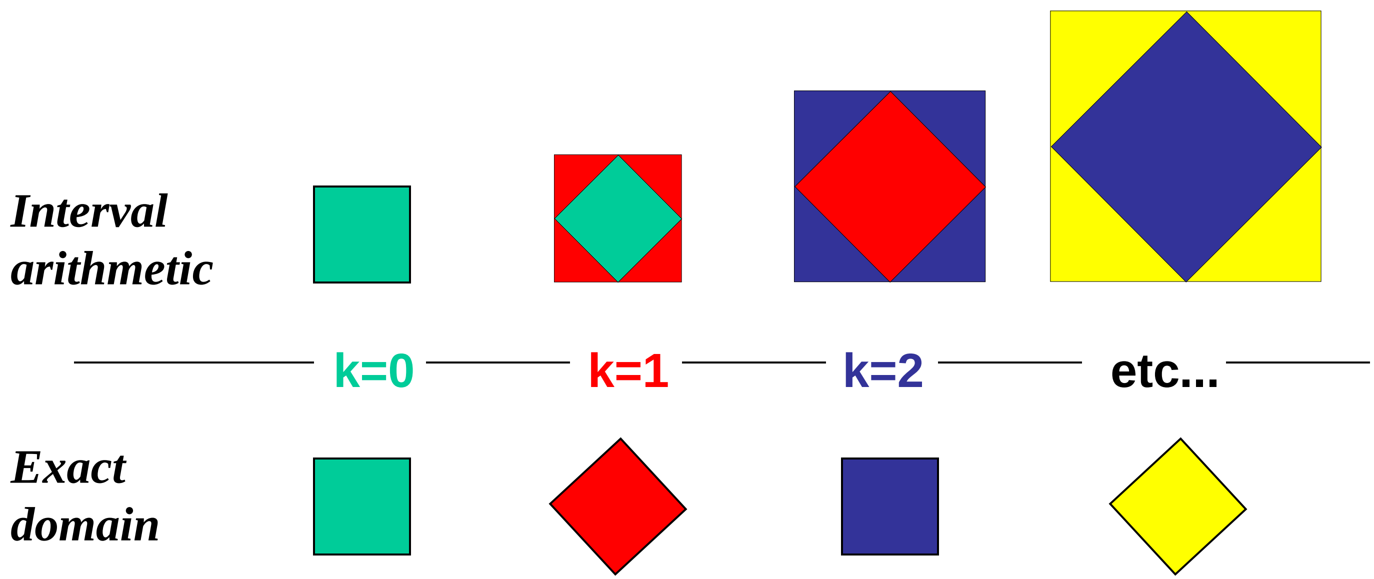
$$f(x) = x - x\ (= \mathbf{0}) \in [-1;+1] - [-1;+1] = \mathbf{[-2;+2]}$$

$$g(x) = x \times x\ (\geq \mathbf{0}) \in [-1;+1] \times [-1;+1] = \mathbf{[-1;+1]}$$

- **Conclusion:**
  - **Multi-occurrence of uncertain variables often involves pessimism**
  - **Dependency relations not taken into account ➔ Pessimism**

Wrapping effect

- **Interval arithmetic directly applied to a dynamical system:**
- **« The » usual example :** $\boxed{x_{k+1} = R.x_k}$ $\quad R = \begin{bmatrix} \cos(45°) & -\sin(45°) \\ \sin(45°) & \cos(45°) \end{bmatrix}$



- **Keep as much information on dependencies as possible**
- **Exact domain: Image of a unit hypercube by a linear application (zonotope)**

$$[x_k^{exact}] = \left\{ x = R^k s, \quad s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \in [-1,+1]^2 \right\}$$

Problem formulation for a first prediction algorithm

- **Model of the system (enclosing the real behavior):**

$$x_{k+1} = A_k.x_k + B_k.u_k + E_k.v_k$$
$$y_k = C_k.x_k + D_k.u_k + F_k.w_k$$

- **Bounded initial state set :** $x_0 \in c_0 + Z(R_0)$

- **Bounded uncertain inputs :** $v_k \in [-1,+1]^q$

  $w_k \in [-1,+1]^m$

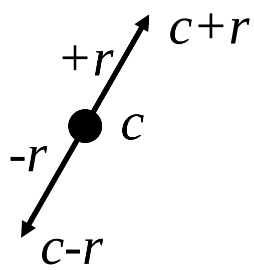
- **Goal: a « good » outer approximation of the …**
  - **… reachable state set, $[x_k]$**
  - **… reachable output set, $[y_k]$**

- **Remarks: continuous/sampled, prediction/correction**

Some preliminaries

- **Minkowski sum:** $\boxed{[x]+[y]=[z]=\{z=x+y\ /\ x\in[x], y\in[y]\}}$

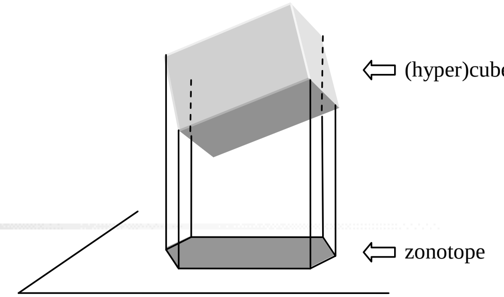
- **Line segment in $\Re^n$:** $\boxed{c+r[-1;+1]}=\{x=c+rs,\quad s\in[-1;+1]\}$



$c\in\Re^n$ **: center**

$r\in\Re^n$ **: radius (length + direction)**

Zonotope: Definition(s)



- **(Centered) Zonotope = Linear image of a *p*-hypercube in an *n*-space**

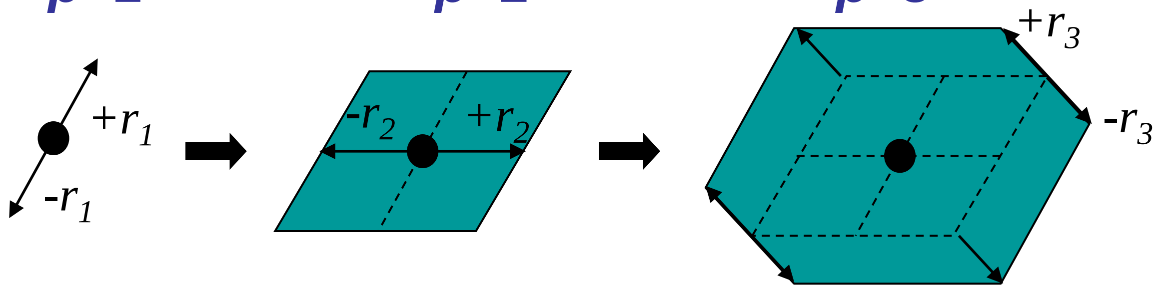
$$\boxed{Z(R) = \left\{ x = R.s, \quad s \in [-1,+1]^p \right\}} \qquad R = [\cdots r_i \cdots] \in \Re^{n \times p}$$

$$Z(R) \subset \Re^n$$

- **Zonotope = Minkowski sum of *p* straight line segments in $\Re^n$:**

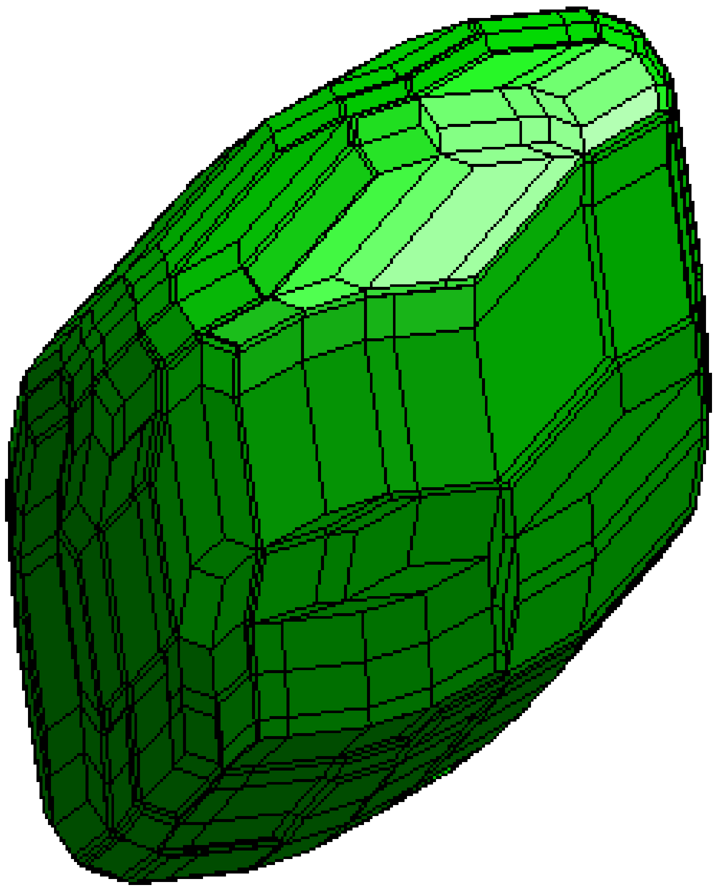
$$c + Z(R) = \underbrace{(c_1 + r_1[-1;+1])}_{\text{Segment}[S_1]} + \cdots + \underbrace{(c_i + r_i[-1;+1])}_{\text{Segment}[S_i]} + \cdots + \underbrace{(c_p + r_p[-1;+1])}_{\text{Segment}[S_p]}$$

- **Example (*n*=2):** ***p*=1** ***p*=2** ***p*=3**



$$c = c_1 + \cdots + c_p \in \Re^n$$

Zonotopes: "emerald" example (!)



***n*=3, *p*=30**

(Centered) Zonotope Properties

- **Sum of two zonotopes :**

$$\boxed{Z(R_1) + Z(R_2) = Z([R_1\ R_2])}$$

➔ **Matrix concatenation**

- **Image of a zonotope by a linear application *L* :**

$$\boxed{L(Z(R)) = Z(LR)}$$

➔ **Matrix product**

- **Smallest box enclosing a zonotope (« interval hull »):**

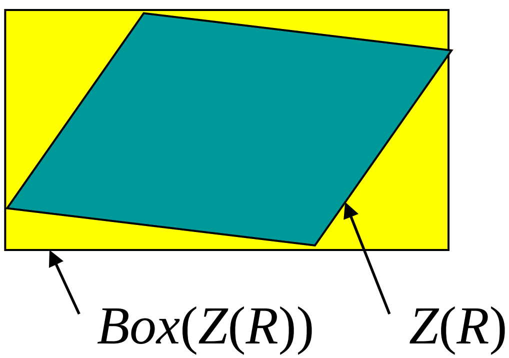
$$\boxed{Box(Z(R)) = Z(b(R))}$$

➔ **1-norm of each line vector**

$$b(R) = \begin{bmatrix} * & & 0 \\ & \ddots & \\ 0 & & * \end{bmatrix} \qquad b(R)_{ii} = \sum_{j=1}^{p} \left| R_{ij} \right|$$

$$i$$

$$b(R) = diag(|R|\mathbf{1})$$



Set-Membership Computations

- **Computation of the reachable set** [Kühn 98]**:**

$$\boxed{\begin{aligned} x_{k+1} &= A_k.x_k + B_k.u_k + E_k.v_k \\ y_k &= C_k.x_k + D_k.u_k + F_k.w_k \end{aligned}}$$

$v_k \in [-1;+1]^q$

$w_k \in [-1;+1]^m$

$x_k \in [x_k] = c_k + Z(R_k)$
(hyp: true at k=0)

- **Recursive algorithm to compute $[x_k]$ and $[y_k]$ (only prediction) :**

$$x_{k+1} \in [x_{k+1}] = c_{k+1} + Z(R_{k+1})$$

$$\begin{cases} c_{k+1} = A_k c_k + B_k u_k \\ R_{k+1} = [A_k R_k \quad E_k] \end{cases}$$

Reduction of the zonotope complexity: $R_k = Red_q(R_k)$ $\Rightarrow$ **Outer approx.**

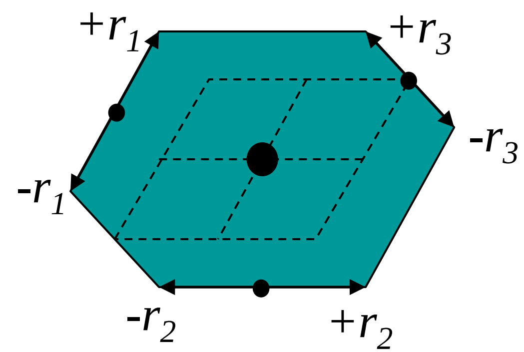
$$y_k \in [y_k] = c_{y,k} + Z(R_{y,k})$$

$$\begin{cases} c_{y,k} = C_k c_k + D_k u_k \\ R_{y,k} = [C_k R_k \quad F_k] \end{cases}$$

Reduction of the Zonotope Complexity

- **Zonotopes:**

$$Z(R) = Z([r_1 \ \cdots \ r_i \ \cdots \ r_p]) \Rightarrow$$



(p=3)

- **Choose the zonotope complexity (*q* segments maxi.)**
- **Sort columns on decreasing Euclidian norm:** $\|r_i\| \geq \|r_{i+1}\|$
- **Reduction (**[Kühn 98], [Combastel 03]**):**

$$Red_q(R) = [\ r_1 \ \ldots \ r_{q\text{-}n} \ \ b([r_{q\text{-}n+1} \ \ldots \ r_p]) \ ]$$

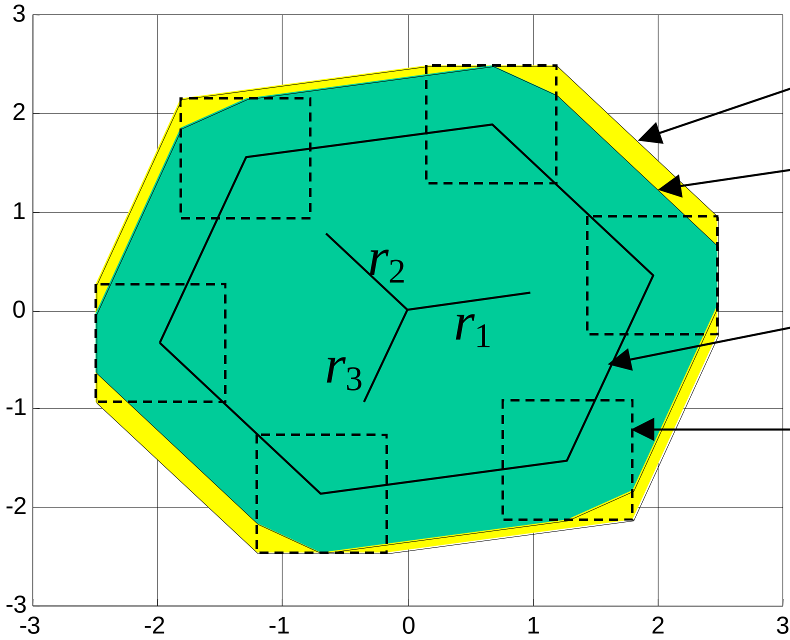
$$\boxed{Z(Red_q(R)) = Z([r_1 \ \ldots \ r_{q\text{-}n}]) + Box(Z([r_{q\text{-}n+1} \ \ldots \ r_p]))}$$

***q-n* segments**
(large segments are kept)

***n* segments**
(reduction of small segments)

Reduction : Example

- **Reduction of a 6-zonotope into a 5-zonotope (*n*=2):**



$Z(Red_5(R)) = [r_1\ r_2\ r_3\ b([r_4\ r_5\ r_6])]$

$Z(R) = [r_1\ r_2\ r_3\ r_4\ r_5\ r_6]$

$Z([r_1\ r_2\ r_3])$ (« Large » segments)

$Box(Z([r_4\ r_5\ r_6]))$ (« Small » segments)

- **Remarks:**
  - **Other sorting criterion (**[Girard 05]**):** $\|r_i\|_1 - \|r_i\|_\infty$
  - **Difficulty to quantify the effect of sorting to obtain theoretical errors…**

From Prediction/Reduction to Observation

Initialization ($[x_0] \leftarrow$ domain s.t. $x_0 \in [x_0]$)

For k=1 to k_{max}

System at time k: u_k, y_k

$[x_{k/k-1}] \leftarrow$ **Prediction**$([x_{k-1}])$

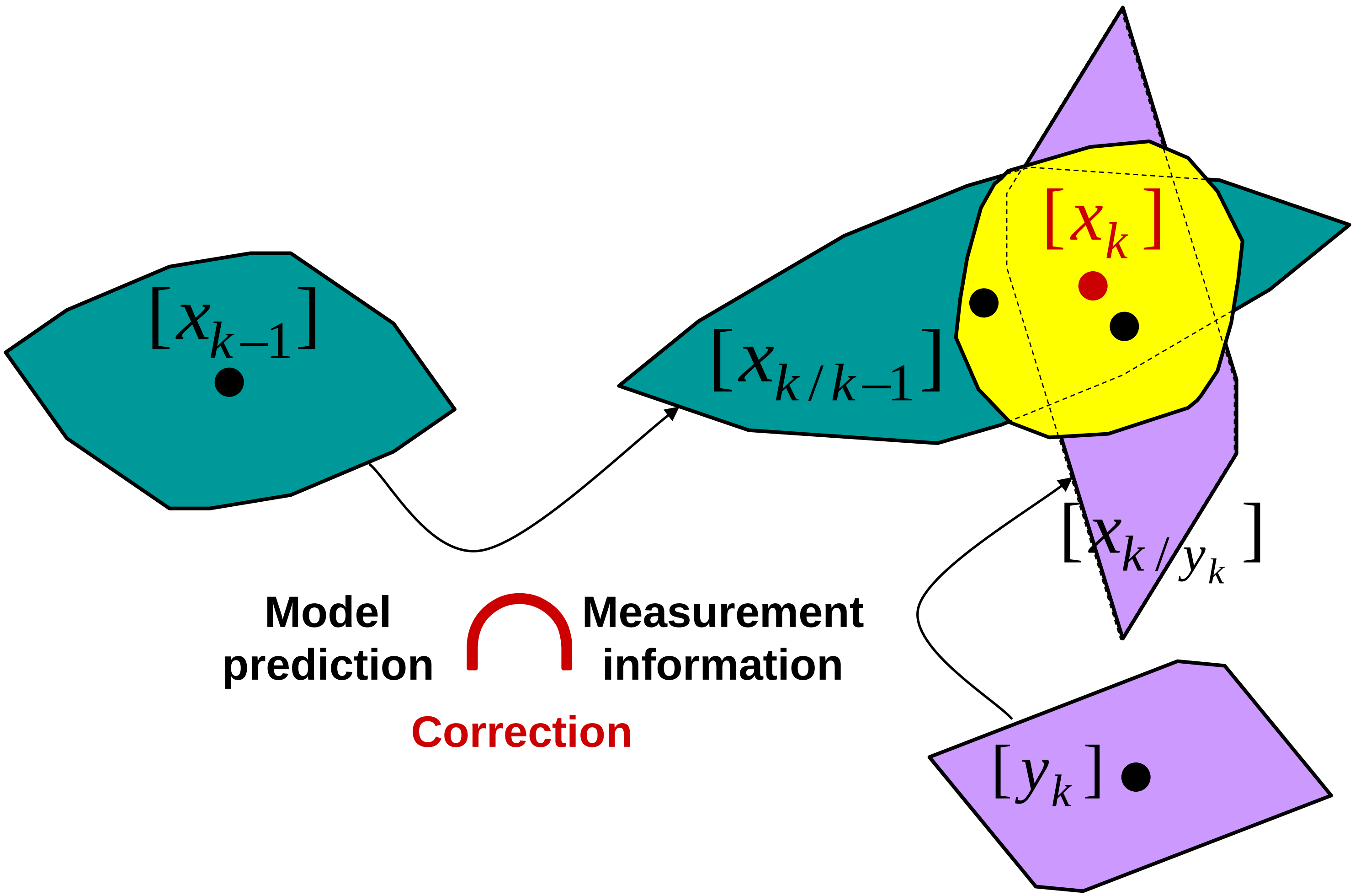
$[x_{k/k-1}] \leftarrow$ **Reduction**$([x_{k/k-1}])$

$[x_k] \leftarrow$ **Correction**$([x_{k/k-1}], y_k, u_k)$

End

basic prediction algorithm

Bounded error state estimation



Compromise (for guarantee to be achieved):

↑ exactness ⇔ ↑ complexity ⇔ ↓ outer approximation

Correction

- **Correction: Outer approx. of the intersection between 2 zonotopes**
- **Singular value decomposition:** $M = USV^T$
- **From simple matrix operations (sums, products,...):**

$$\boxed{[s] = c_s + Z(R_s)}$$

$$[x_k] = [x_{k/k}] = c_{k/k-1} + R_{k/k-1}[s]$$

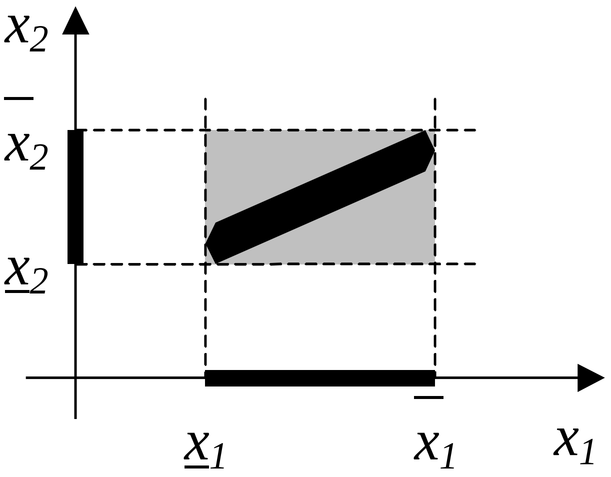
- **Corrected domain:**

$$\boxed{[x_k] = [x_{k/k}] = (c_{k/k-1} + R_{k/k-1}c_s) + Z(R_{k/k-1}R_s)}$$

Center update *Shape update*

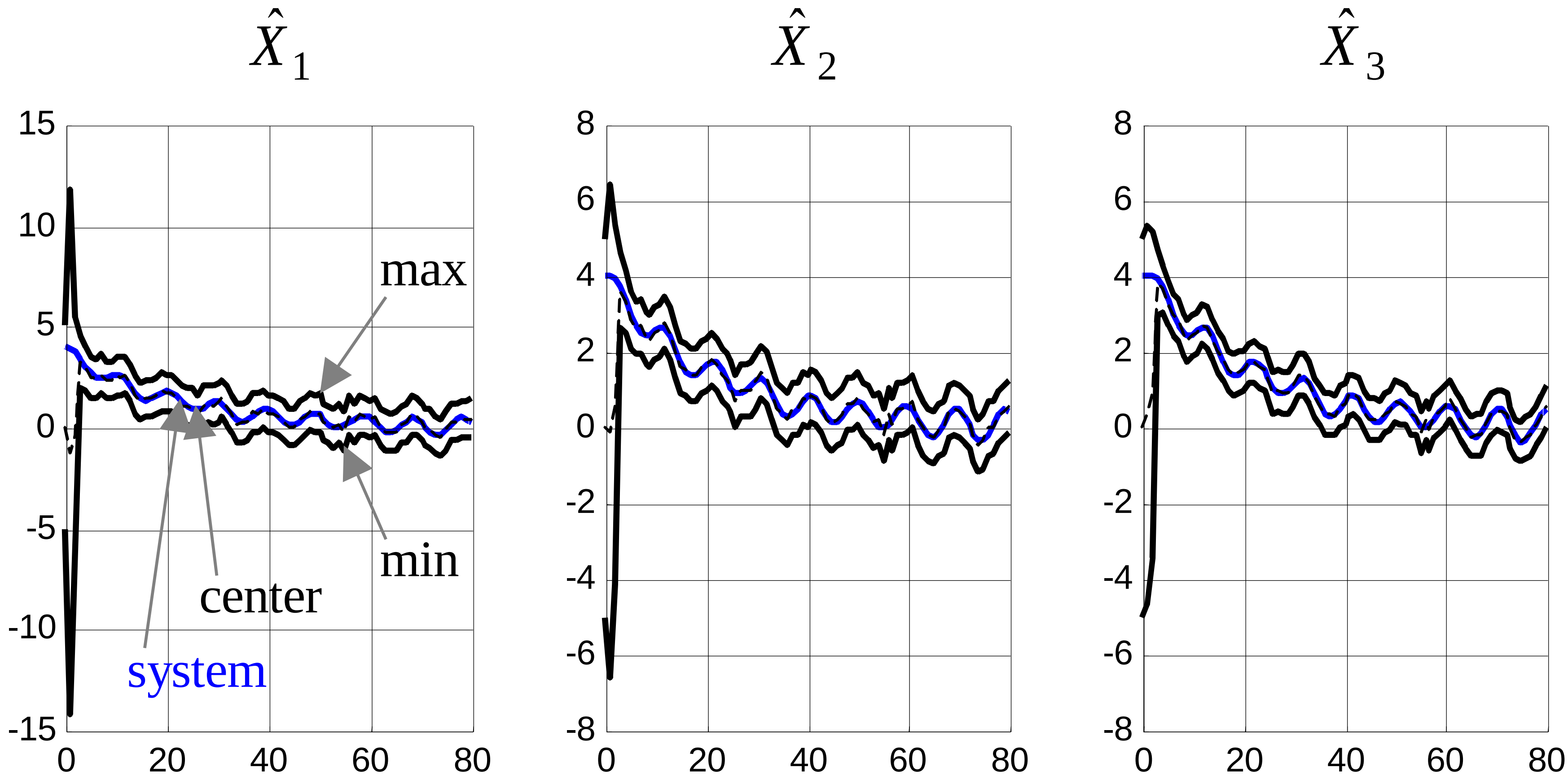
Simulations (academic examples)

- Computation of state bounds ➔ Interval hull:



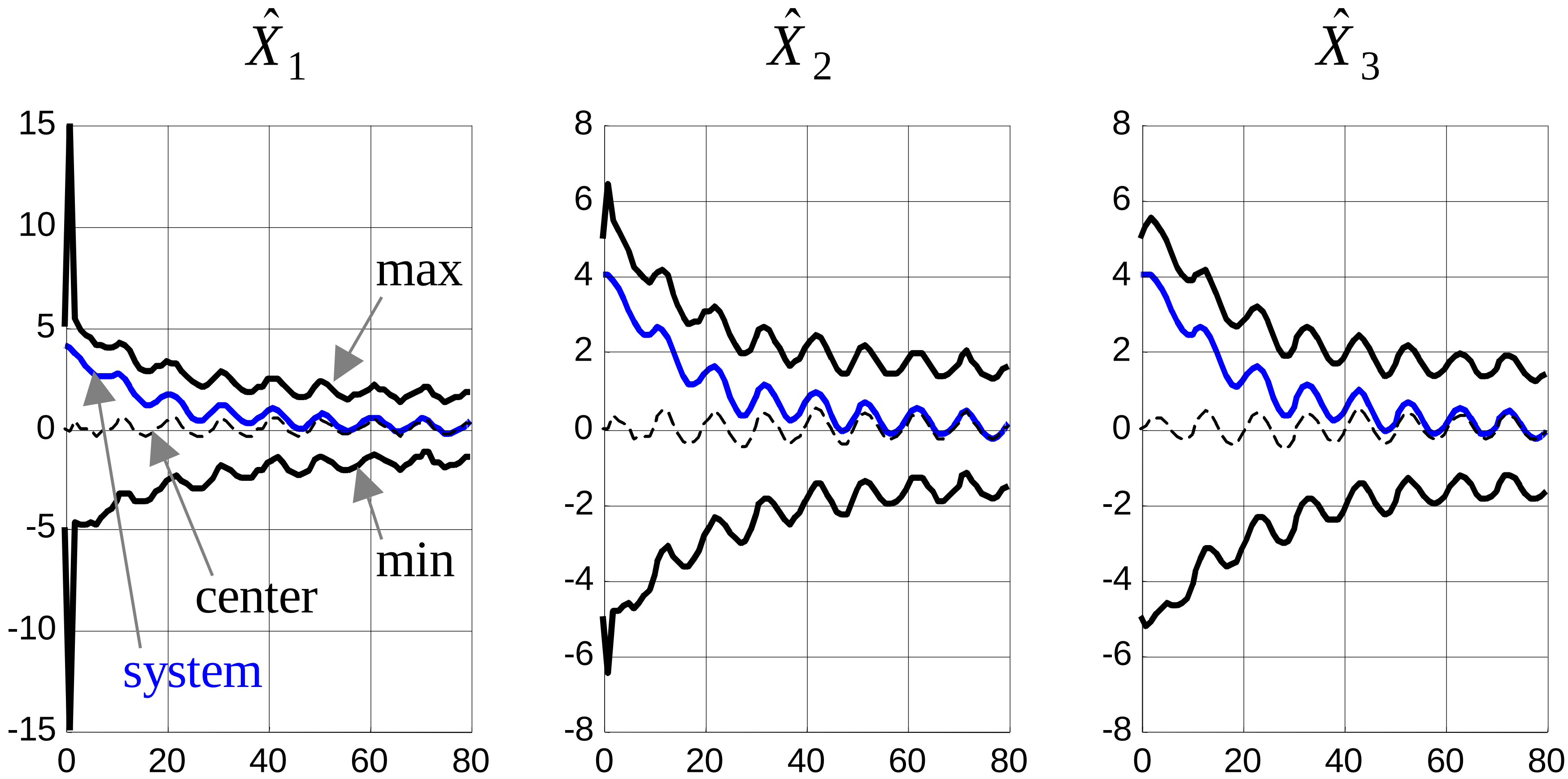
- Application to 3 cases with various observability properties

Case 1: Observable



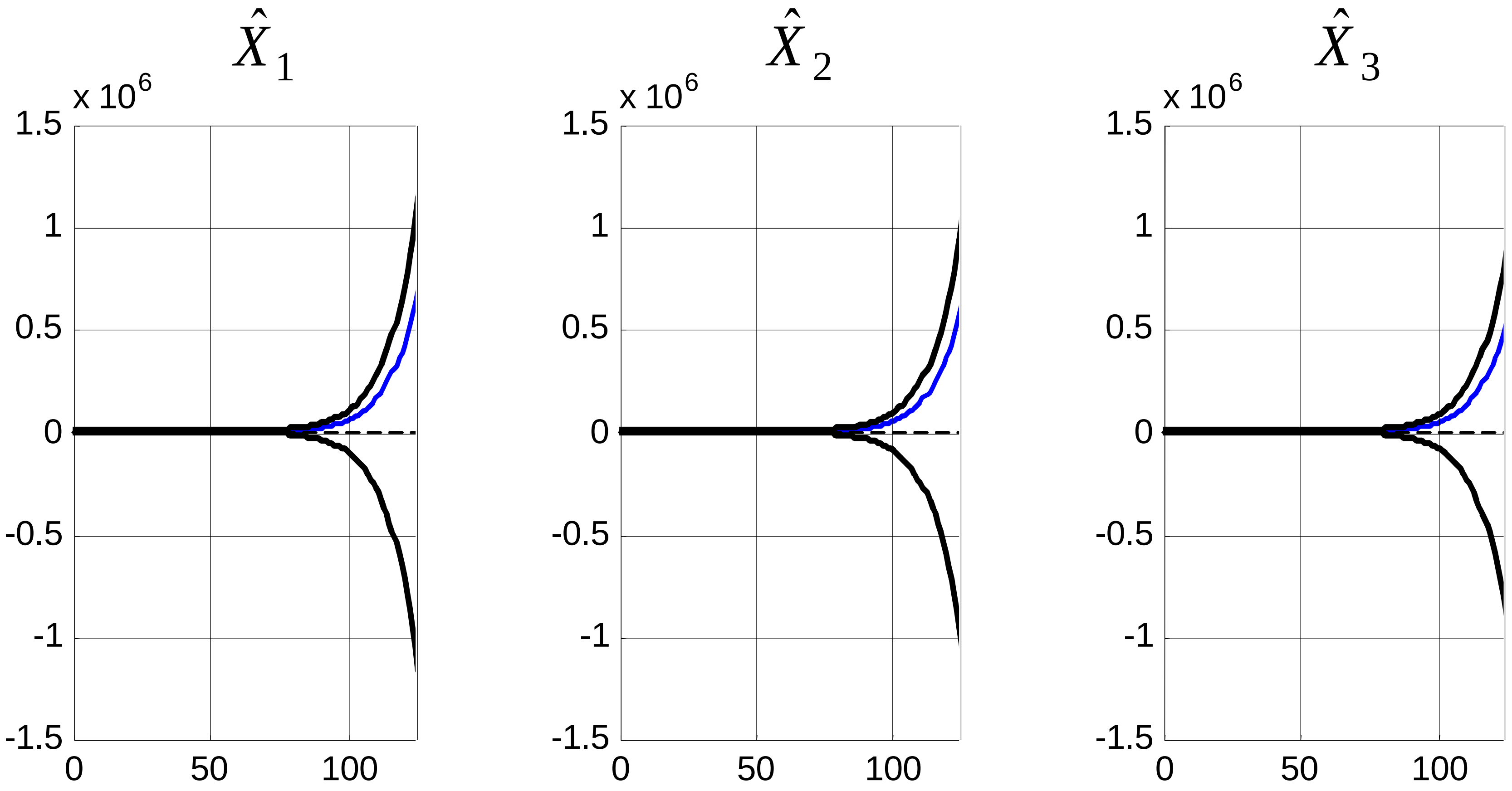
$$G_1(z) = \frac{(z-2)}{(z-0.95)} \cdot \frac{(z-0.7)}{(z-z_0).(z-\bar{z}_0)}$$

Case 2: Non observable, Detectable



$$G_2(z) = \frac{(z-2)}{\cancel{(z-0.95)}} \cdot \frac{\cancel{(z-0.95)}}{(z-z_0).(z-\bar{z}_0)}$$

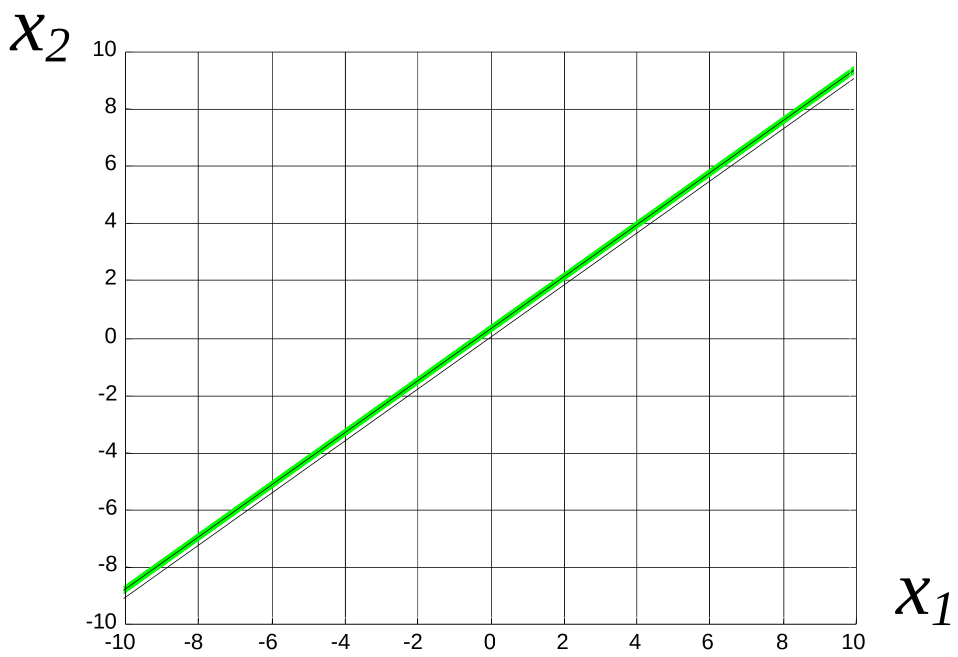
Case 3: Non observable, Non detectable



$$G_3(z) = \frac{(z-2)}{\cancel{(z-1.1)}} \cdot \frac{\cancel{(z-1.1)}}{(z-z_0).(z-\bar{z}_0)}$$

Case 3: Non observable, Non detectable

- **Projection of the zonotope in the plane (x_1,x_2):**



Zoom, k=30

$$D_{NO}=[0.631\ \ 0.574\ \ 0.522]^T$$

➔ **Domain growth in the direction of the non obs. space**

A basic extension to parametric uncertainties

$$x_{k+1} = \begin{bmatrix} 0 & -0.5 \\ 1 & 1+0.3\alpha_k \end{bmatrix} . x_k + 0.02 . \begin{bmatrix} -6 \\ 1 \end{bmatrix} . v_k$$

$$y_k = \begin{bmatrix} -100 & 10 \end{bmatrix} . x_k + 0.02 . w_k$$

[El Ghaoui, 2001]

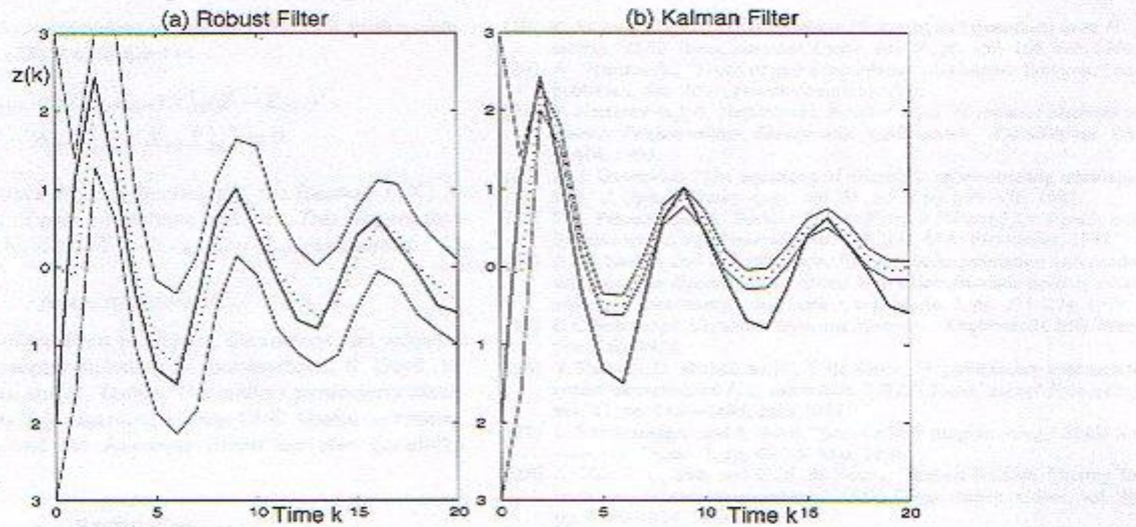
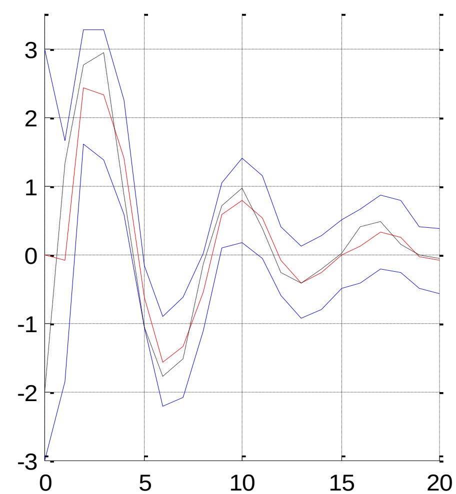


Fig. 1. Estimation of $z(k)$ using (a) the robust deterministic filter and (b) a standard Kalman filter. The thick lines represent $z(k)$, the dotted lines represent the central estimates, the solid lines represent the bounds on the estimates [ellipsoidal projections for (a), and 3σ confidence regions, for (b)].



Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

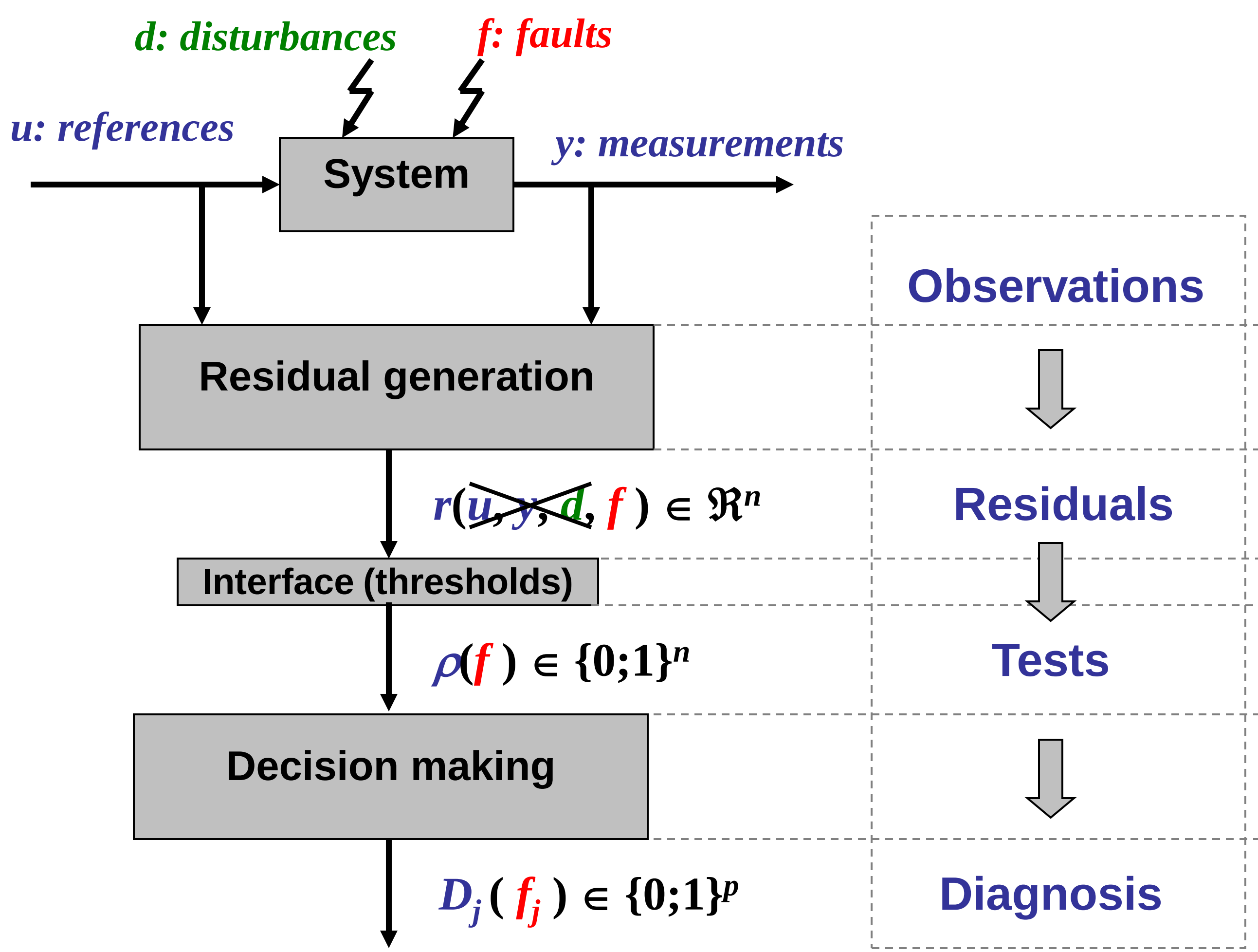
4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

Analytical Redundancy and FDI : Usual Scheme

- **Fault diagnosis = Detection, Isolation, Identification of faults**



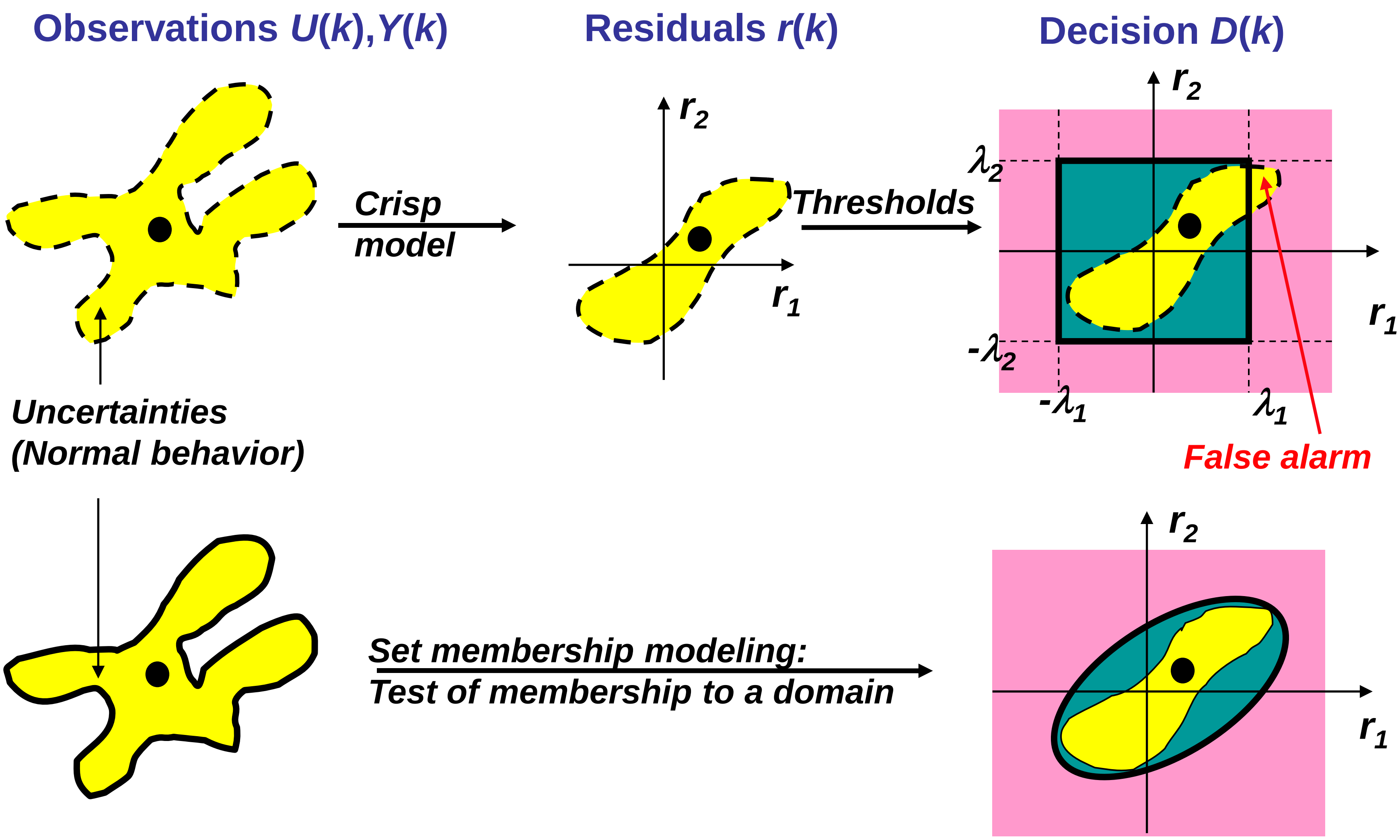
Known value
Unknown value
Tested value
Decoupling

$$\begin{cases} \dot{x} = A.x + B.u + E.d + F.f \\ y = C.x + D.u + G.d + H.f \end{cases}$$

Observers,
Parity space,…

	f_1	f_2	f_3
ρ_1	1	1	1
ρ_2	1	0	1
ρ_3	1	1	0

Thresholds for Fault Detection

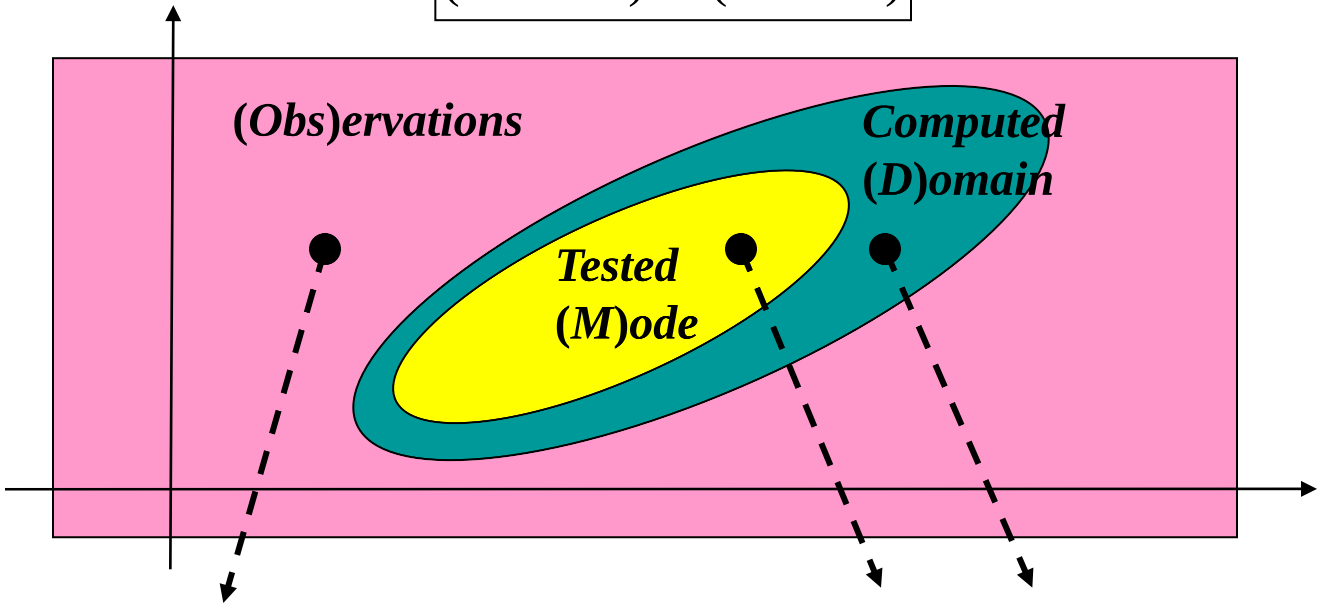


refined/rough out-bounding ➔ good/bad sensitivity (guaranteed robustness)

Set-membership approaches and diagnosis

- **Compute domains (related to the modes to be tested):**
  - **Guaranteed inclusion :** $M \subset D$

$$\boxed{(Obs \in M) \Rightarrow (Obs \in D)}$$



Inconsistency ➔ Certainty

$Obs \notin D$ $\quad$ $Obs \notin M$

Consistency ➔ No certainty

$Obs \in D$ $\quad$ $Obs \in M$ **?** $Obs \notin M$

➔ **Logically sound continuous / discrete interfaces**

➔ **Bounded uncertainties: No limit on the number of perfect decoupling**

Observers and set-membership computations

- **Given a sampled (LTV) residual generator…**

$$\boxed{\begin{cases} z_{k+1} = M_k.z_k + N_k^u.u_k + N_k^y.y_k + N_k^v.v_k \\ r_k = P_k.z_k + Q_k^u.u_k + Q_k^y.y_k + Q_k^v.v_k \end{cases}}$$

$$y_k \in [y_k]$$
$$v_k \in [-1,+1]^p$$
$$z_0 \in [z_0]$$

- **…compute the reachable output set $[r_k]$**
  - **On-line ➔ Adaptive threshold**
  - **Off-line ➔ Choice of a fixed threshold**

- **Example: residual generator based on an adaptive observer**

Class of Systems under study

- **Goal : Computing bounds from the residuals obtained from an adaptive observer** [Zhang, et al, 2001], [Zhang, et al, 2002], [Guyader and Zhang, 2003]

- **System model:**

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k + E_k w_k + f_k \\ y_k = C_k x_k + F_k v_k \end{cases} \qquad f_k = \Psi_k \theta_k$$

A_k, B_k, C_k : Time-varying matrices

$w_k \in [-1;+1]^n$, $v_k \in [-1;+1]^m$: Bounded errors

$f_k \in \Re^n$: Influence of faults ($\Psi_k \in \Re^{n\times p}$, $\theta_k \in \Re^p$)

- **Specification of the fault-free behavior:**

$\theta_{k+1} = \theta_k + G_k e_k$: Admissible parameter variations where $e_k \in [-1;+1]^p$

$SysOK \Rightarrow \theta_k \in [-\varepsilon;+\varepsilon]$ $\Leftrightarrow$ $\theta_k \notin [-\varepsilon;+\varepsilon] \Rightarrow \neg SysOK$

Adaptive Observer

- **Adaptive observer : Estimation of the state (x_k) and some parameters (θ)** [Guyader and Zhang, 2003]

$$
\begin{cases}
\Gamma_{k+1} = (A_k - K_k C_k)\Gamma_k + \Psi_k \\
\hat{\theta}_{k+1} = \hat{\theta}_k + \mu_k \Gamma_k^T C_k^T (y_k - C_k \hat{x}_k) \\
\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + \Psi_k \hat{\theta}_k + K_k (y_k - C_k \hat{x}_k) + \Gamma_{k+1}(\hat{\theta}_{k+1} - \hat{\theta}_k)
\end{cases}
$$

$K_k \in \Re^{n\times m}$: Matrix sequence designed so that $\Phi_k = A_k - K_k C_k$ is exponentially stable
$\Gamma_k \in \Re^{n\times p}$: Linear filtering of Ψ_k
$\mu_k \in \Re^{+}$: Parameter adaptation gain

Theorem 1: Residual Generator
[Guyader and Zhang, 2003]

- **Residual Generator**:

$$\begin{cases} \eta_{k+1} = (A_k - K_k C_k)\eta_k + E_k w_k - K_k F_k v_k - \Gamma_{k+1} G_k e_k \\ \tilde{\theta}_{k+1} = (I - \mu_k \Gamma_k^T C_k^T C_k \Gamma_k)\tilde{\theta}_k - \mu_k \Gamma_k^T C_k^T C_k \eta_k - \mu_k \Gamma_k^T C_k^T v_k + G_k e_k \end{cases}$$

- **The residuals are:**

$$\tilde{\theta}_k = \theta_k - \hat{\theta}_k$$
$$\eta_k = x_k - \hat{x}_k - \Gamma_k \tilde{\theta}_k$$

- **The residual generator is a LTV system with bounded inputs and:**

$$[\theta_k] = \underbrace{\hat{\theta}_k + [\tilde{\theta}_k]}_{\text{Set allowing to detect inconsistencies (FDI)}}$$

Theorem 2: Convergence of the Adaptive Observer

[Guyader and Zhang, 2003]

IF:

1. $A_k, C_k, \Psi_k, K_k, \mu_k, w_k, v_k, e_k$ are all bounded and $\Phi_k = A_k - K_k C_k$ is exponentially stable,

2. $\mu_k > 0$ is small enough so that $\| \sqrt{\mu_k} C_k \Gamma_k \| \leq 1$
where $\| \bullet \|$ is the spectral norm (the largest singular value) of a matrix,

3. $\exists$ a constant $\alpha > 0$ and an integer $L > 0$ such that:

$$\forall k \geq 0, \quad \frac{1}{L} \sum_{i=k}^{k+L-1} \mu_i \Gamma_i^T C_i^T C_i \Gamma_i \geq \alpha I$$

(Input excitation)

THEN:

The residuals $\tilde{\theta}_k = \theta_k - \hat{\theta}_k$ and $\eta_k = x_k - \hat{x}_k - \Gamma_k \tilde{\theta}_k$ are bounded.

Link between the Observer and the Set-Membership Computations

- **Denoting:** $\underline{A}_k = \begin{bmatrix} A_k - K_k C_k & 0 \\ -\mu_k \Gamma_k^T C_k^T C_k & (I - \mu_k \Gamma_k^T C_k^T C_k \Gamma_k) \end{bmatrix}$ $\underline{E}_k = \begin{bmatrix} E_k & -K_k F_k & -\Gamma_{k+1} G_k \\ 0 & -\mu_k \Gamma_k^T C_k^T F_k & G_k \end{bmatrix}$

$$z_k = \begin{bmatrix} \eta_k \\ \tilde{\theta}_k \end{bmatrix} \qquad \underline{w}_k = \begin{bmatrix} w_k \\ v_k \\ e_k \end{bmatrix} \qquad \underline{C}_k = \begin{bmatrix} 0 & I \end{bmatrix}$$

- **… the residual generator can be rewritten as a sampled LTV system:**

$$\begin{cases} z_{k+1} = \underline{A}_k z_k + \underline{E}_k \underline{w}_k \\ \theta_k = \underline{C}_k z_k + \hat{\theta}_k \end{cases} \Rightarrow \boxed{\theta_k \in [\theta_k]}$$

$$\Downarrow$$

$$\boxed{[\theta_k] \cap [-\varepsilon; +\varepsilon] \Rightarrow \neg SysOK}$$

Set-Membership Computations (Basic Prediction Algorithm)

- … the residual generator can be rewritten as a sampled LTV system:

$$\begin{cases} z_{k+1} = \underline{A}_k z_k + \underline{E}_k \underline{w}_k \\ \theta_k = \underline{C}_k z_k + \hat{\theta}_k \end{cases}$$

$\underline{w}_k \in [-1,1]^m$

$z_k \in [z_k] = c_k + Z(R_k)$
(hyp: true at k=0)

Adaptive Observer

- Recursive algorithm to compute $[\theta_k]$:

$$z_{k+1} \in [z_{k+1}] = c_{k+1} + Z(R_{k+1})$$

$$\begin{cases} c_{k+1} = \underline{A}_k c_k \\ R_{k+1} = [\underline{A}_k R_k \quad \underline{E}_k] \end{cases}$$

Reduction of the zonotope complexity:

$$R_k = Red_q(R_k)$$

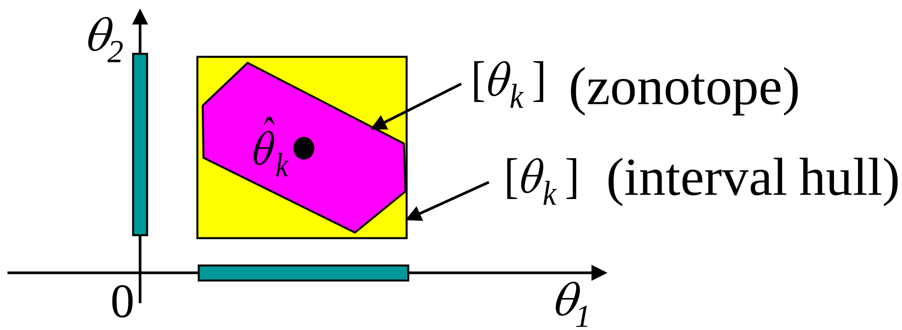
$$\theta_k \in [\theta_k] = c_{\theta,k} + Z(R_{\theta,k})$$

$$\begin{cases} c_{\theta,k} = \underline{C}_k c_k + \hat{\theta}_k \\ R_{\theta,k} = \underline{C}_k R_k \end{cases}$$

Decision

- **The decision is based on:**

$$\boxed{\theta_k \in Box([\theta_k]) = c_{\theta,k} + Z(b(R_{\theta,k}))}$$

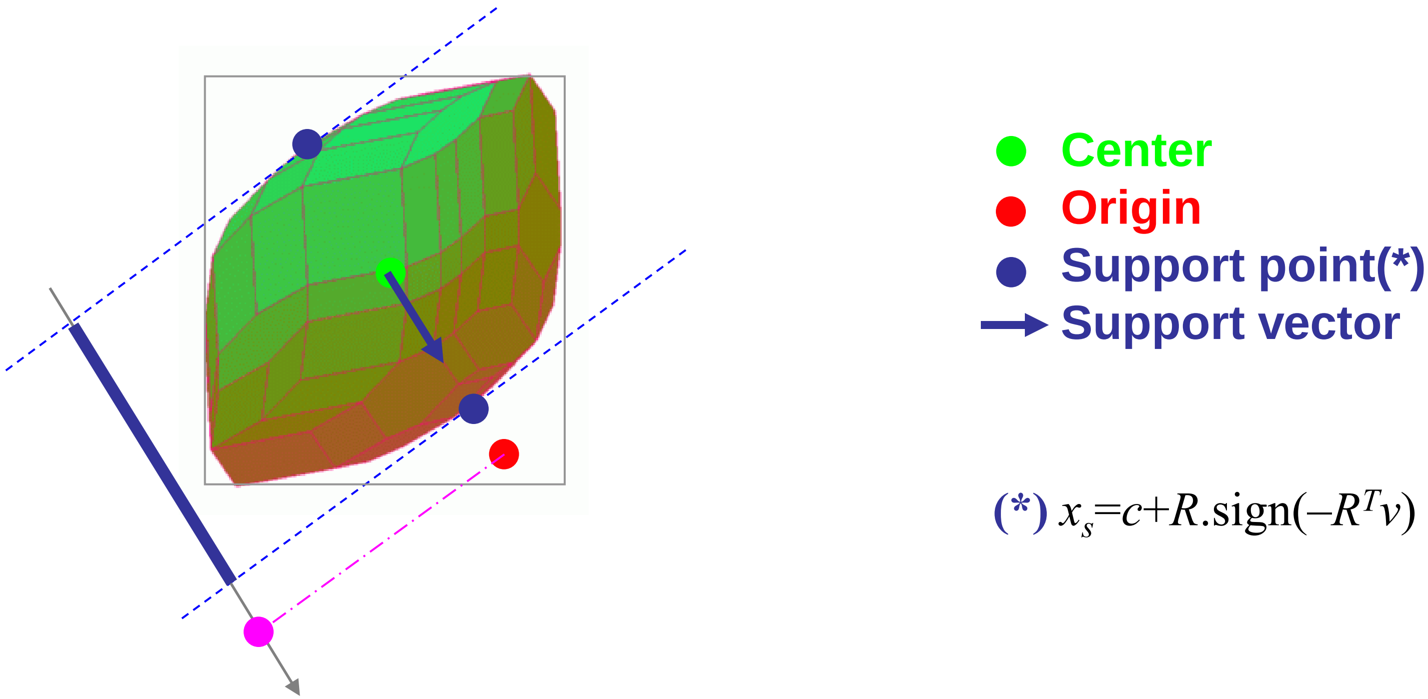


- **The (non-) membership of 0 to [θ_k] allows to detect and to isolate the faults**
- **The domain [θ_k] allows to identify the faults (with quantified uncertainties).**

- **Remark: refined decision (↑ computation load) ➔ collision detection**

Remark: collision detection & zonotopes

- Collision detection = $\boxed{\text{Testing if } 0 \in c + Z(R)}$



(*) $x_s = c + R.\mathrm{sign}(-R^T v)$

- Algorithm: ISA-GJK [Van Den Bergen, 1999], among others.

- Link with fault detection: when the origin is out of the "residual" domain, an inconsistency is detected (i.e. a fault has occurred).

Example: Satellite Model

Continuous time model (linearized satellite model):

$$\begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & 0 & 1 \\ 0 & -2\omega & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \theta_1 & 0 \\ 0 & 0 \\ 0 & \theta_2 \end{bmatrix} \cdot \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \\ \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} \end{cases}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r-1 \\ \dot{r} \\ \varphi - \omega.t \\ \dot{\varphi} - \omega \end{bmatrix} \begin{matrix} \text{radial position} \\ \text{radial speed} \\ \text{angular position} \\ \text{angular velocity} \end{matrix}$$

r : radius of (circular) orbit (normalized to 1)
φ : rotation angle
ω : nominal angular velocity

Discrete time model (T_s = 0.1s):

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k + E_k w_k + f_k \\ y_k = C_k x_k + F_k v_k \end{cases} \qquad f_k = \Psi_k \theta_k$$

Example: Satellite Model

- **Sampled model:**

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k + E_k w_k + f_k & f_k = \Psi_k \theta_k \\ y_k = C_k x_k + F_k v_k \end{cases}$$

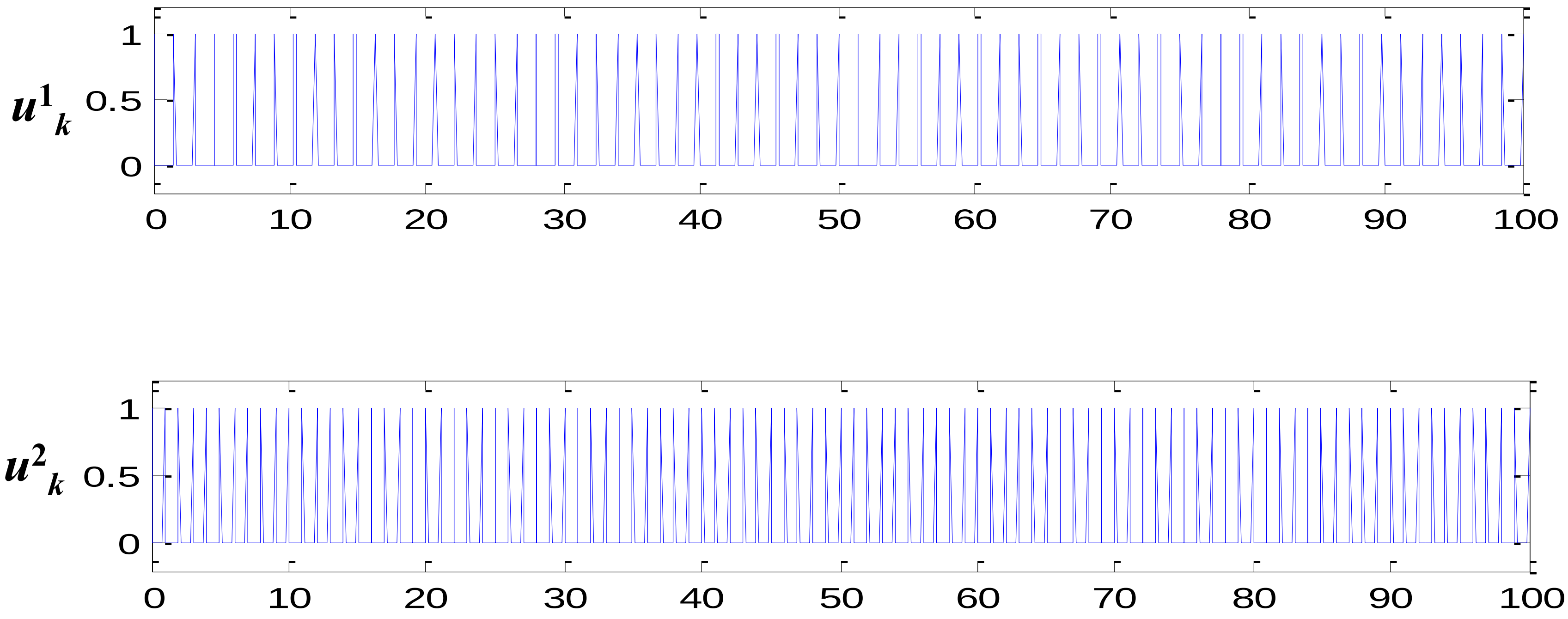
$$B_k = H.\begin{bmatrix} 0 & 0 \\ g_1 & 0 \\ 0 & 0 \\ 0 & g_2 \end{bmatrix} \quad u_k = \begin{bmatrix} u_k^1 \\ u_k^2 \end{bmatrix} \quad f_k = \Psi_k \theta_k = H\begin{bmatrix} 0 & 0 \\ u_k^1 & 0 \\ 0 & 0 \\ 0 & u_k^2 \end{bmatrix}.\begin{bmatrix} \theta_{1,k} \\ \theta_{2,k} \end{bmatrix} \quad C_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad x_k = \begin{bmatrix} r-1 \\ \dot{r} \\ \varphi - \omega.t \\ \dot{\varphi} - \omega \end{bmatrix}_k$$

$$g_1 = 1, \quad g_2 = 1.5$$

$$A_k = \begin{bmatrix} 1 & 0.1 & 0 & 3.49\times10^{-6} \\ 3.66\times10^{-8} & 1 & 0 & 6.98\times10^{-5} \\ -4.25\times10^{-14} & -3.49\times10^{-6} & 1 & 0.1 \\ -1.28\times10^{-12} & -6.98\times10^{-5} & 0 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 0.1 & 0.005 & 0 & 1.16\times10^{-7} \\ 1.83\times10^{-9} & 0.1 & 0 & 3.49\times10^{-6} \\ -1.06\times10^{-15} & -1.16\times10^{-7} & 0.1 & 0.005 \\ -4.25\times10^{-14} & -3.49\times10^{-6} & 0 & 0.1 \end{bmatrix}$$

- **Uncertain inputs:** $E_k = 10^{-5} I_4$, $F_k = 10^{-2} I_2$,

Inputs Excitation



Simulation context

- **"Real system":** $x_0 = [1, 0, 0, 3.49\ \ 10^{-4}]^T$
- **Adaptive observer:**

$$\Gamma_0 = 0_{4\times 2}$$
$$\hat{x}_0 = [0.1, 0, 0, 3.49\ \ 10^{-5}]^T$$
$$\hat{\theta}_0 = [-0.8, 0.8]^T$$

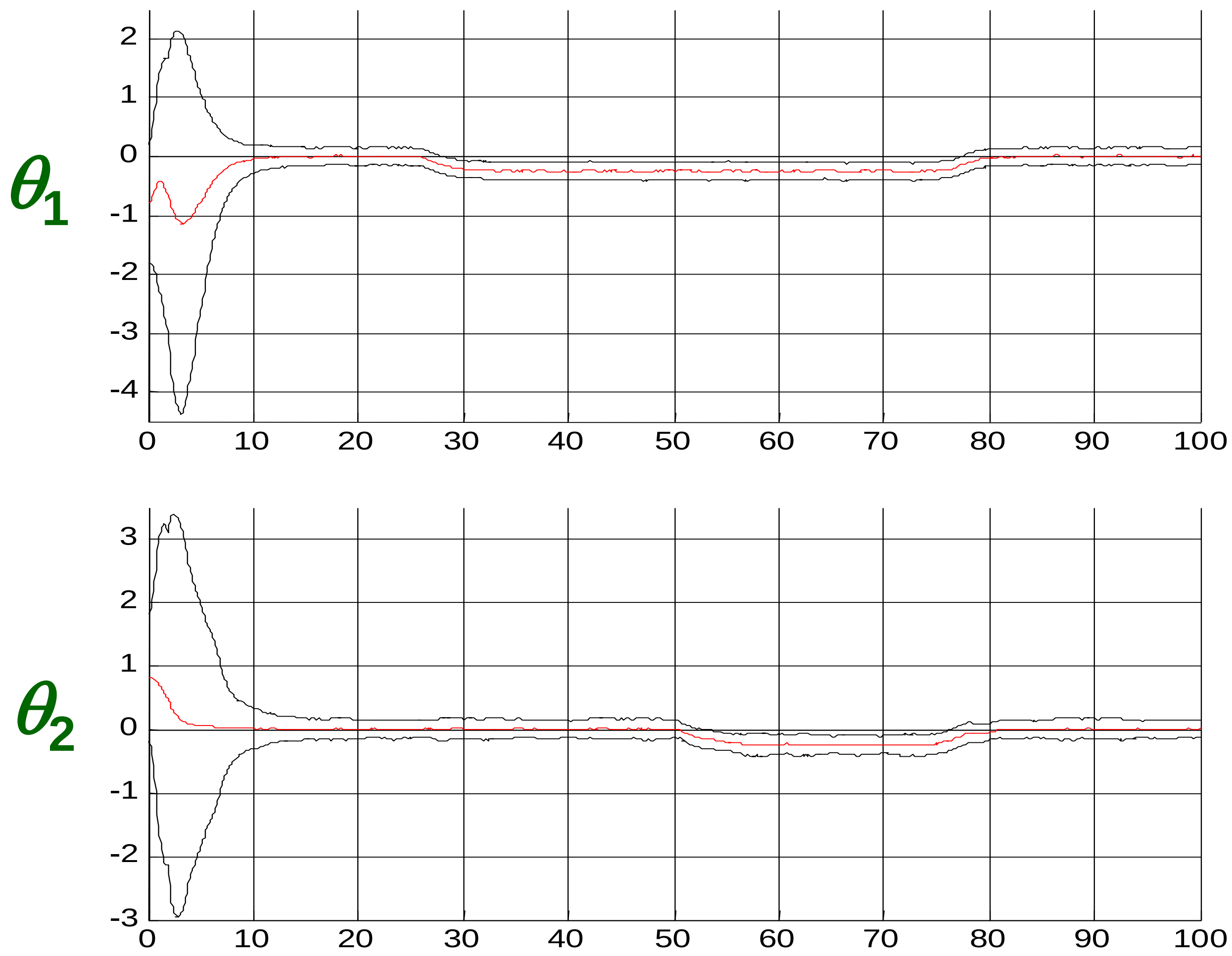
$$\mu_k = 4 \qquad K_k = \begin{bmatrix} 0.1412 & 4.93\times 10^{-6} \\ 0.0932 & 5.26\times 10^{-5} \\ -4.93\times 10^{-6} & 0.1412 \\ -5.26\times 10^{-5} & 0.0932 \end{bmatrix}$$

- **Residual evaluator:**

$$\left.\begin{aligned} c_0 &= [0, 0, 0, 0, 0, 0]^T, \\ Z_0 &= I_6, \end{aligned}\right\} [z_0] = c_0 + Z(R_0) = [-1, +1]^6$$
$$q = 40$$

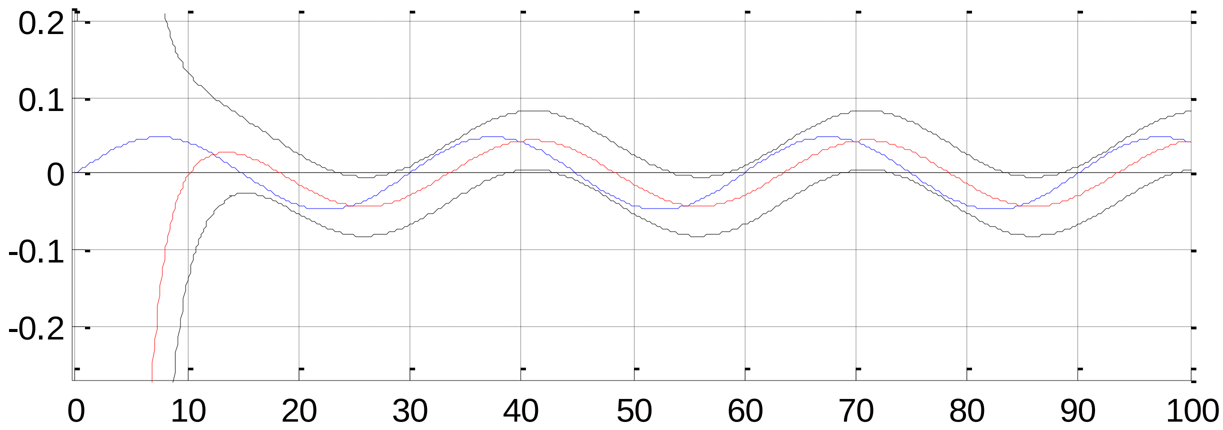
Initial domain for states and parameters

Temporal Evolution of the Intervals bounding the Fault Parameters



Here: no admissible fault-free parameter variations (G_k=0 and ε=0)

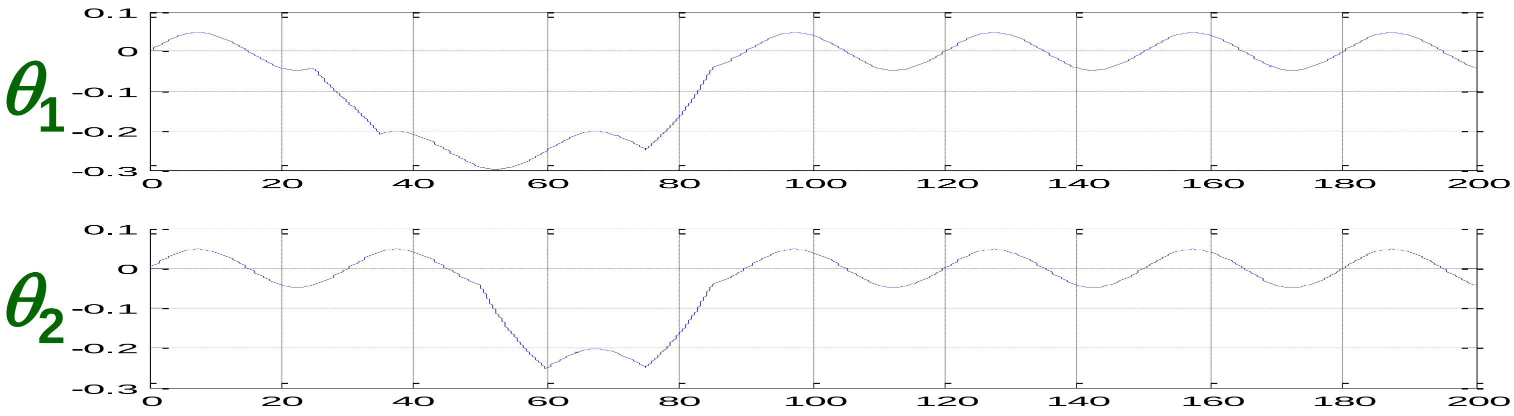
Fault-free Parameter Variations



Real value, estimated value and envelope of $\theta_{1,k}$ under variations consistent with the fault-free specifications

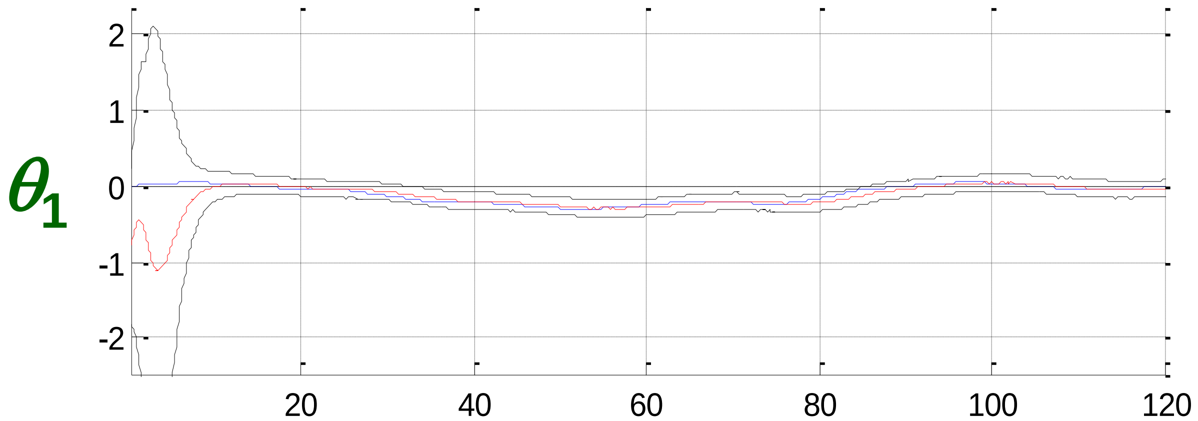
($G_k = 10^{-3} I_2$ and $\varepsilon = 0.05[1\ 1]^T$)

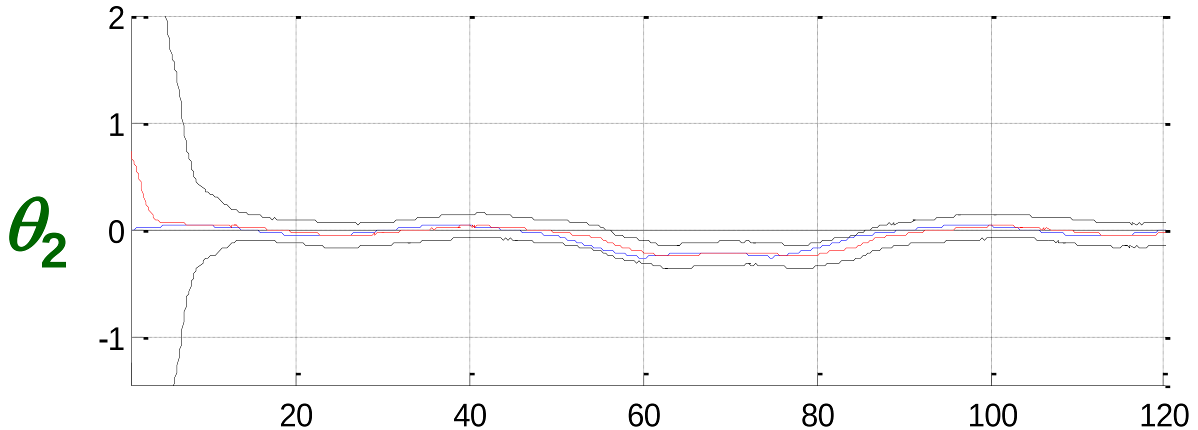
A Scenario of Parameter Variations



Combination of faulty and fault-free parameter variations

Temporal Evolution of the Intervals bounding the Fault Parameters



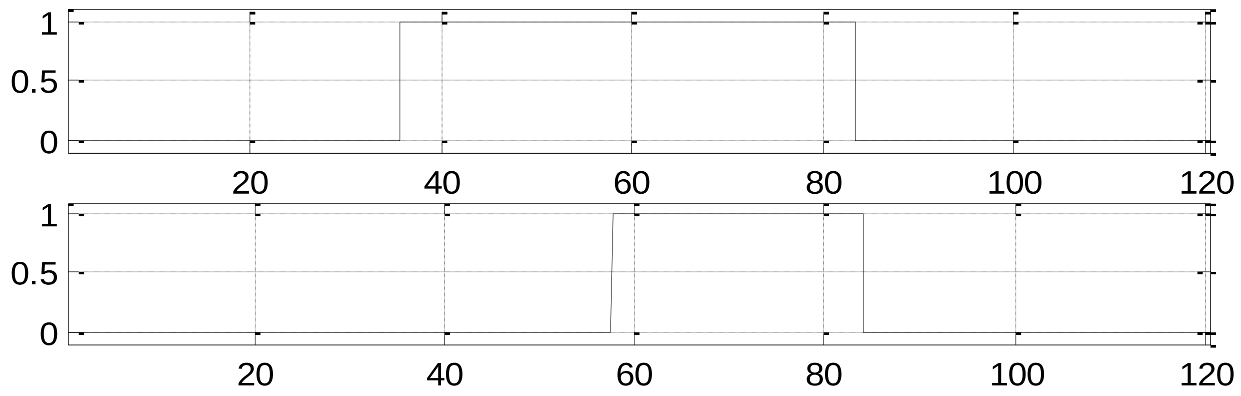


Temporal evolution of the intervals bounding the fault parameters
Top: [θ_1]. Bottom: [θ_2].

Diagnostic decision

Fault 1 (θ_1)

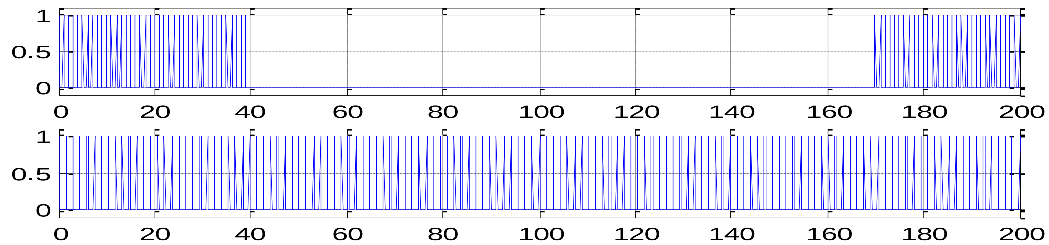
Fault 2 (θ_2)



Fault Detection and Isolation

Loss of Input Excitation

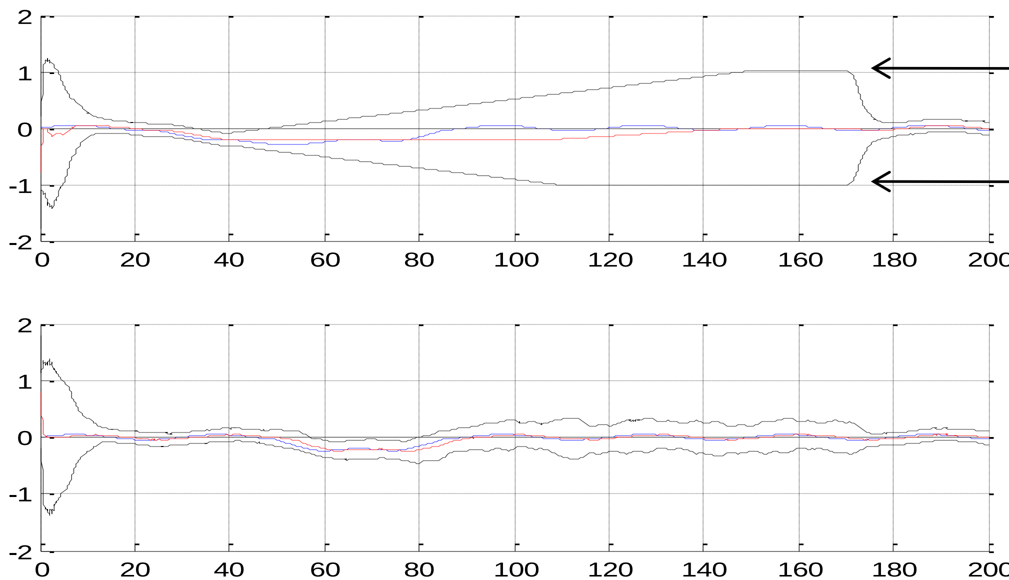
Inputs



Estimated Parameters (Faults)

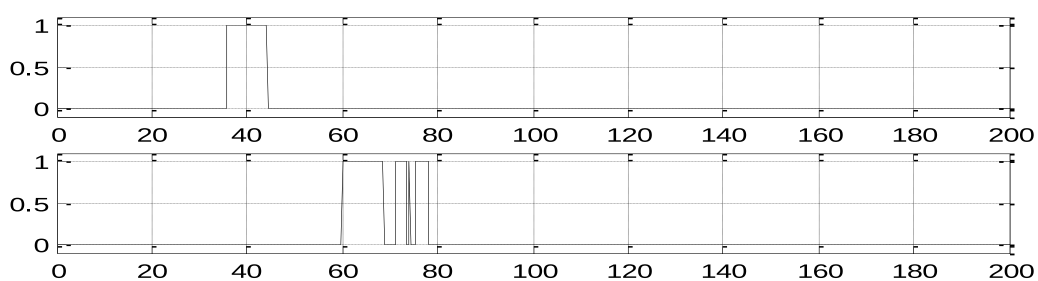
θ_1

θ_2



Intersection with physical bounds

Decision



Adaptive observer and zonotopes for FDI

- **Main features:**
  - Adaptive observer for joint state-parameter estimation
  - Set-membership residual evaluation based on zonotopes computations
  - Combination of both: Fault detection, isolation and identification are achieved with a single observer and in a guaranteed way (up to the model validity).
  - The case of multiple and intermittent fault is naturally handled (satellite example)
  - Admissible fault-free parameter variations are taken into account
  - The estimated parameter set remains consistent even in the case of a lack of input excitation

Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

Problem formulation with parametric uncertainties

- **Model of the system:**

$$\begin{cases} x_{k+1} = A_{\theta,k}x_k + B_{\theta,k}u_k + E_{\theta,k}v_k \\ x_0 \in \{c_0\} \oplus Z(R_0) \subset \Re^n, \quad v_k \in [-1;+1]^r, \quad \theta \in [-1;+1]^q \\ A_{\theta,k} \in [A]_\theta = A_c + A_1\theta_1 + \ldots + A_q\theta_q + [-A_r;+A_r] \\ B_{\theta,k} \in [B]_\theta = B_c + B_1\theta_1 + \ldots + B_q\theta_q + [-B_r;+B_r] \\ E_{\theta,k} \in [E]_\theta = E_c + E_1\theta_1 + \ldots + E_q\theta_q + [-E_r;+E_r] \end{cases}$$

- **Main features:**
  - **Bounded initial state set**
  - **Bounded uncertain inputs**
  - **Both constant and sampled-time varying parametric uncertainties**
- **Goal: Computation of an outer approximation of $[x_k]$**
- **Remarks: no output equation (not an issue), prediction only**

Definition of *afm* objects

- An *afm* object is an affine matrix function of some parameters (θ) with an interval remainder term:

Math. definition of $[M]_\theta \subset \Re^{n\times p}$ **:**

$$[M]_\theta = \{ M_\theta \mid M_\theta = M_c + M_1\theta_1 + \ldots + M_q\theta_q + M_R \wedge M_R \in [-M_r ;+M_r] \}$$

Constructor method:

$$[M]_\theta = afm(M_c, \{M_1, \ldots, M_q\}, M_r)$$

- *afm* objects address the "dependency problem" by preserving some affine dependencies (as well as some inclusion properties)

A scalar example: $(\theta \in [-1;+1])$

$x(\theta) = 3+\theta,$ $\quad [x] - [x] = 3+[-1;+1]-3-[-1;+1] = \mathbf{[-2;+2]}$

$[x]_\theta = afm(3,\{1\},0),$ $\quad [x]_\theta - [x]_\theta = 3+\theta-3-\theta = afm(0,\{0\},0) = \mathbf{\{0\}}$

- **Remark: main idea: affine arithmetic** [Stolfi] **applied to matrices.**

Operators overload with *afm* objects

- ***afm*** **operators are designed to satisfy an inclusion property:**

<u>Inclusion property</u>: The *afm* remainder terms are computed such that the non linear terms of θ are always enclosed provided $\theta \in$ [-1;+1]^q:

$$[R]_\theta = [M]_\theta \bullet [N]_\theta \wedge \theta \in [-1;+1]^q \Rightarrow \{\, M_\theta \bullet N_\theta \mid M_\theta \in [M]_\theta,\, N_\theta \in [N]_\theta \,\} \subset [R]_\theta$$

- **Sum:** $[M]_\theta + [N]_\theta = afm(M_c+N_c, \{M_1+N_1, \ldots, M_q+N_q\}, M_r+N_r)$
- **Concat.:** $[[M]_\theta\, [N]_\theta] = afm([M_c\; N_c], \{[M_1\; N_1], \ldots, [M_q\; N_q]\}, [M_r\; N_r])$
- **Product:** *Less direct extension (see next slide).*

A scalar example (continued): $x(\theta)\times x(\theta) = (3+\theta)^2 = 9 + 6\theta + \theta^2,$

$$\Big\downarrow\; T(x(.)\times x(.),\, \theta)\;:\;\text{simplif. + enclosure}$$

$$\theta \in [-1;+1] \;\Rightarrow\; x(\theta)\times x(\theta) \in [x]_\theta \times [x]_\theta = 9.5 + 6\theta + [-0.5;+0.5]$$

- **Now, we are equipped to address matrix polynomial functions of θ!**

Operators overload with *afm* objects

- **Sum:** $[M]_\theta + [N]_\theta = afm(M_c+N_c, \{M_1+N_1, \ldots, M_q+N_q\}, M_r+N_r)$

- **Concat.:** $[[M]_\theta\,[N]_\theta] = afm([M_c\ N_c], \{[M_1\ N_1], \ldots, [M_q\ N_q]\}, [M_r\ N_r])$

- **Product:**

$$[M]_\theta[N]_\theta = afm\left(M_cN_c + \frac{1}{2}\sum_{i=1}^{q} M_iN_i, \right.$$
$$\{(M_1N_c + M_cN_1), \ \ldots\ , (M_qN_c + M_cN_q)\},$$
$$M_r\left(|N_c| + \sum_{i=1}^{q}|N_i|\right) + \left(|M_c| + \sum_{i=1}^{q}|M_i|\right)N_r + M_rN_r +$$
$$\left. \frac{1}{2}\sum_{i=1}^{q}|M_iN_i| + \sum_{\substack{(i,j)\in\{1\ldots q\}^2 \\ i<j}} |M_iN_j + M_jN_i| \right)$$

- **Other operators: not implemented yet but extensions are possible**

Inclusion in an interval matrix

- **Notations:** $C \pm R = [C\text{-}R; C\text{+}R]$ (interval matrix in centered form)

$$[M]_\theta = \{ M_\theta \mid M_\theta = M_c + M_1\theta_1 + \ldots + M_q\theta_q + M_R \wedge M_R \in [-M_r ;+M_r] \}$$

$\rightarrow$ Interval matrix **depending** on θ

$$[M_\theta] = \mu([M]_\theta) = M_c + (0\pm|M_1|) + \ldots + (0\pm|M_1|) + (0\pm M_{\mathrm{r}})$$

$\rightarrow$ Interval matrix **independent** from θ

- ***mid*** **and** ***rad*** **operators for interval matrix inclusion:**

$$mid([M_\theta]) = M_c$$
$$rad([M_\theta]) = |M_1| + \ldots + |M_q| + M_r$$

- **Related inclusion property:**

$$\theta \in [-1;+1]^q \Rightarrow [M]_\theta \subset [M_\theta]$$
$$[M_\theta] = mid([M_\theta]) \pm rad([M_\theta]) = afm(mid([M_\theta]), \{\}, rad([M_\theta]))$$

Problem formulation with *afm* objects

- **Model of the system:**

$$
\begin{cases}
x_{k+1} = A_{\theta,k}x_k + B_{\theta,k}u_k + E_{\theta,k}v_k \\
\\
x_0 \in \{c_0\} \oplus Z(R_0) \subset \Re^n, \quad v_k \in [-1;+1]^r, \quad \theta \in [-1;+1]^q \\
\\
A_{\theta,k} \in [A]_\theta = afm(A_c, \{A_1 \ldots A_q\}, A_r) \\
B_{\theta,k} \in [B]_\theta = afm(B_c, \{B_1 \ldots B_q\}, B_r) \\
E_{\theta,k} \in [E]_\theta = afm(E_c, \{E_1 \ldots E_q\}, E_r)
\end{cases}
$$

- **Main idea of this work:**
  - **Keep the general structure of the basic prediction algorithm**
  - **Replace real vector/matrix operators by operators over *afm* objects in order to preserve (constant) parameter dependencies (operator overload)**

- **Goal: Computation of an outer approximation of $[x_k]$**

Parameterized Families of Zonotopes (PFZ)

- **Illustration of the main idea and consequences:**

$$\begin{aligned} c_{k+1} &= A_k c_k + B_k u_k \\ R_{k+1} &= [A_k R_k \quad E_k] \\ R_k &= Red_m(R_k) \end{aligned} \quad \Rightarrow \quad \begin{aligned} [c]_{\theta,k+1} &= [A]_\theta [c]_{\theta,k} + [B]_\theta u_k \\ [R]_{\theta,k+1} &= [[A]_\theta [R]_{\theta,k} \quad [E]_\theta] \\ [R]_{\theta,k+1} &= Red_m([R]_{\theta,k+1}) \end{aligned}$$

Computation:	**From real matrices…**	**…to parameter dependent interval matrices**
Semantic:	**From zonotopes…**	**…to Parameterized Families of Zonotopes**

- **Parameterized Family of (centered) Zonotopes ($\mathbf{PFZ_c}$):**

$$Z(R) = \{Rs \mid s \in [-1;+1]^p\} \quad \Rightarrow \quad Z([R]_\theta) = \{Z(R) \mid R \in [R]_\theta\}$$

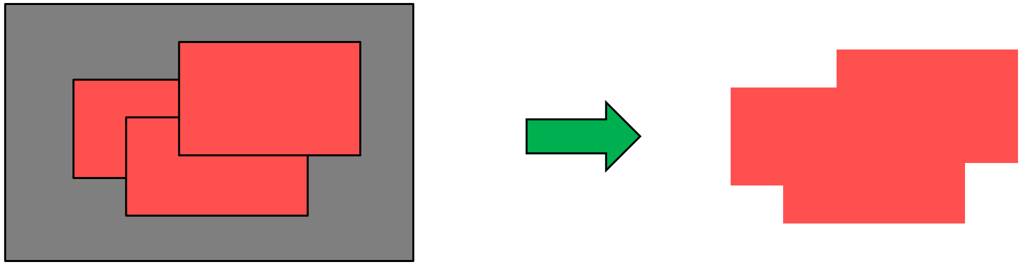
From sets… **…to sets of sets (SOS !)**

$\underbrace{\text{…to sets of sets (SOS !)}}_{\text{parameter dependent}}$

- **Remark: The notion of family of zonotopes with no parametric dependency has first been introduced in** [Alamo, 05]**.**

Operators related to SOS

- **Unary union operator:** $\cup\Theta = \bigcup_{S\in\Theta} S$ **returns a set enclosing all the elements in a set of set**



- **Element by element singleton inclusion operator:**

$$\sigma : a \rightarrow \{a\}$$

$$\sigma : \{a, b, \ldots\} \rightarrow \{\{a\}, \{b\}, \ldots\}$$

Case 1: the operand is not a set

Case 2: the operand is a set

- **Remark: set invariance by (∪σ):** $S = \cup(\sigma(S))$

PFZ : definition and properties

- **(Minkowski) sum of two PFZ_c :**

$$\theta \in [-1;+1]^q \Rightarrow \boxed{Z([P]_\theta) \oplus Z([Q]_\theta) = Z([[P]_\theta \ [Q]_\theta])}$$

- **"Linear" image of a PFZ_c :**

$$\theta \in [-1;+1]^q \Rightarrow \boxed{[L]_\theta Z([R]_\theta) \subset Z([L]_\theta [R]_\theta)}$$

- **Parameterized Family of Zonotopes (PFZ):**

$$\boxed{\sigma([c]_\theta) \oplus Z([R]_\theta)}$$

- **Enclosure properties:** **need for some developments ! (next slides)**

PFZ Properties

- **Enclosure of a (non parameterized) family of zonotopes (**[Alamo,05]**):**

$$\forall B \in (\mathrm{R}^+)^{n\times p},\ \cup(Z(0\pm B)) = \{x \mid x = Ms \wedge M \in [-B,+B] \wedge s \in [-1;+1]^p\} \subset Z(b(B))$$

- **Parameterized set enclosing all the zonotopes in a PFZ:**

$$\cup(\sigma([c]_\theta) \oplus Z([R]_\theta)) = \{\ x \mid x = c + Rs \wedge c \in [c]_\theta \wedge R \in [R]_\theta \wedge s \in [-1;+1]^p\}$$

- **Geometric Zonotope enclosure of a PFZ:**

$$\cup\{\ \cup(\sigma([c]_\theta) \oplus Z([R]_\theta)) \mid \theta \in [-1;+1]^q\} \subset \{\underline{c}\} \oplus Z(\underline{R})$$
$$\underline{c} = c_c,\quad \underline{R} = [c_1\ \ldots\ c_q\ \ R_c\ \ b([c_r\ \ |R_1| + \ldots + |R_q| + R_r])]$$

An intermediate step of the proof is:

$$\subset \{c_c\} \oplus Z([c_1\ \ldots\ c_q]) \oplus (0\pm c_r) \oplus Z(R_c) \oplus \cup Z(0\pm(|R_1| + \ldots + |R_q| + R_r))$$

- **Geometric Box enclosure (interval hull) of a PFZ$_c$:** *(corollary)*

$$\cup\{\ \cup Z([R]_\theta)\ \mid\ \theta \in [-1;+1]^q\}\ \subset\ Z(b([mid([R_\theta])\ \ rad([R_\theta])]))$$

Reduction operator for a PFZ

- **Reduction algo.:** ***m*** **= maximum number of columns after reduction**

Input: $[R]_\theta = afm(R_c, \{R_1, \ldots, R_q\}, R_r) \subset \Re^{n\times p}$

If $p \le m$ Then

$\quad Red_m([R]_\theta) = [R]_\theta$

Else

$\quad K = |mid([R_\theta])| + rad([R_\theta])$

$\quad L = 1_{1\ \ n}(K * K)$

$\quad I$ = indices of the $(m\text{-}n)$ largest elements in L

$\quad J = \{1 \ldots p\} \setminus I$

$$Red_m([R]_\theta) = [\ ([R]_\theta)_{\bullet,I}\ \ afm(b([mid([R_\theta]_{\bullet,J})\ \ rad([R_\theta]_{\bullet,J})]), \{\}, 0)\ ]$$

- **Inclusion property:** $\forall R_\theta \in [R]_\theta,\ Z(R_\theta) \subset Z(Red_m(R_\theta))$

- **Corollary:** $\cup(Z([R]_\theta)) \subset \cup(Z(Red_m([R]_\theta)))$

Computation of the reachable sets

- **Algorithm for envelope computation under parametric uncertainties:**

Initialization:

$$[c]_\theta = afm(c_0, \{\}, 0)$$
$$[R]_\theta = afm(R_0, \{\}, 0)$$
$$(\underline{c}_0, \underline{R}_0) = bz([c]_\theta, [R]_\theta) \qquad (\text{here} : x_0 \in \{\underline{c}_0\} \oplus Z(\underline{R}_0))$$

For k=0 to (k_{max}-1),

$$[c]_\theta = [A]_\theta [c]_\theta + [B]_\theta u_k$$
$$[R]_\theta = [\ [A]_\theta [R]_\theta \quad [E]_\theta\ ]$$
$$[c]_\theta = afm(c_c, \{c_1, \ldots, c_q\}, 0)$$
$$[R]_\theta = [afm(R_c, \{R_1, \ldots, R_q\}, 0) \quad afm(b([c_r \ \ R_r]), \{\}, 0)]$$
$$[R]_\theta = Red_m([R]_\theta)$$
$$(\underline{c}_{k+1}, \underline{R}_{k+1}) = bz([c]_\theta, [R]_\theta) \qquad (\text{here} : x_{k+1} \in \{\underline{c}_{k+1}\} \oplus Z(\underline{R}_{k+1}))$$

Function $(\underline{c}, \underline{R}) = bz([c]_\theta, [R]_\theta)$

$$\underline{c} = c_c$$
$$\underline{R} = [c_1 \ \ldots \ c_q \ \ R_c \ \ b([c_r \ \ |R_1|+\ldots+|R_q|+R_r])]$$

Computation of the reachable sets

- **Comments about the proof of the inclusion property:**

$$x_k \in \cup\{ \cup(\sigma([c]_\theta) \oplus Z([R]_\theta)) \mid \theta \in [-1;+1]^q\}$$

Geometric inclusion: true, but not a good starting point for the proof because the explicit dependency on θ is lost.

$$\boxed{x_{\theta,k} \in \cup(\sigma([c]_{\theta,k}) \oplus Z([R]_{\theta,k})}$$

In order to take the dependency on θ (central term in *afm*) into account in the proof.

$$\cup(\sigma([c]_\theta) \oplus Z([R]_\theta)) \subset \cup(\sigma(afm(c_c,\{c_1, \ldots, c_q\},0)) \oplus Z([afm(R_c,\{R_1, \ldots, R_q\},0)\ \ afm(b([c_r\ \ R_r]),\{\},0)]))$$

$$\cup(\sigma([c]_\theta) \oplus Z([R]_\theta)) \subset \cup(\sigma([c]_\theta) \oplus Z(Red_m(R_\theta)))$$

$$\boxed{x_{\theta,k+1} \in \cup(\sigma([c]_{\theta,k+1}) \oplus Z([R]_{\theta,k+1})}$$

$$x_{k+1} \in \{ x_{\theta,k+1} \mid \theta \in [-1;+1]^q\} \subset \{c_{k+1}\} \oplus Z(R_{k+1})$$

Geometric inclusion: byproduct of the iteration update.

Mass-Spring Example

- Dynamic system: 3 bodies and 5 springs
- 8 uncertain parameters (q=8): Viscous friction and Springs stiffness

$$x_{k+1} = A_\theta x_k + B_\theta u_k \qquad \theta \in [-1;+1]^q$$

$$A(\theta) = \begin{bmatrix} 1 & \frac{1}{10} & 0 & 0 & 0 & 0 \\ -1-\frac{1}{80}\theta_4-\frac{1}{100}\theta_5-\frac{1}{400}\theta_8 & \frac{7}{10}+\frac{3}{100}\theta_1 & \frac{2}{5}+\frac{1}{100}\theta_5 & 0 & \frac{1}{10}+\frac{1}{400}\theta_8 & 0 \\ 0 & 0 & 1 & \frac{1}{10} & 0 & 0 \\ \frac{2}{5}+\frac{1}{100}\theta_5 & 0 & -\frac{7}{10}-\frac{1}{100}\theta_5-\frac{3}{400}\theta_6 & \frac{7}{10}-\frac{3}{100}\theta_2 & \frac{3}{10}+\frac{3}{400}\theta_6 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{10} \\ \frac{1}{10}+\frac{1}{400}\theta_8 & 0 & \frac{3}{10}+\frac{3}{400}\theta_6 & 0 & -\frac{3}{5}-\frac{3}{400}\theta_6-\frac{1}{200}\theta_7-\frac{1}{400}\theta_8 & \frac{7}{10}-\frac{3}{100}\theta_3 \end{bmatrix}$$

$$= A_c + A_1\theta_1 + \ldots + A_8\theta_8 \qquad (A_r=0)$$

$$B(\theta) = [0 \quad 0.1 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$= B_c \qquad (B_1 = \ldots = B_8 = B_r = 0)$$

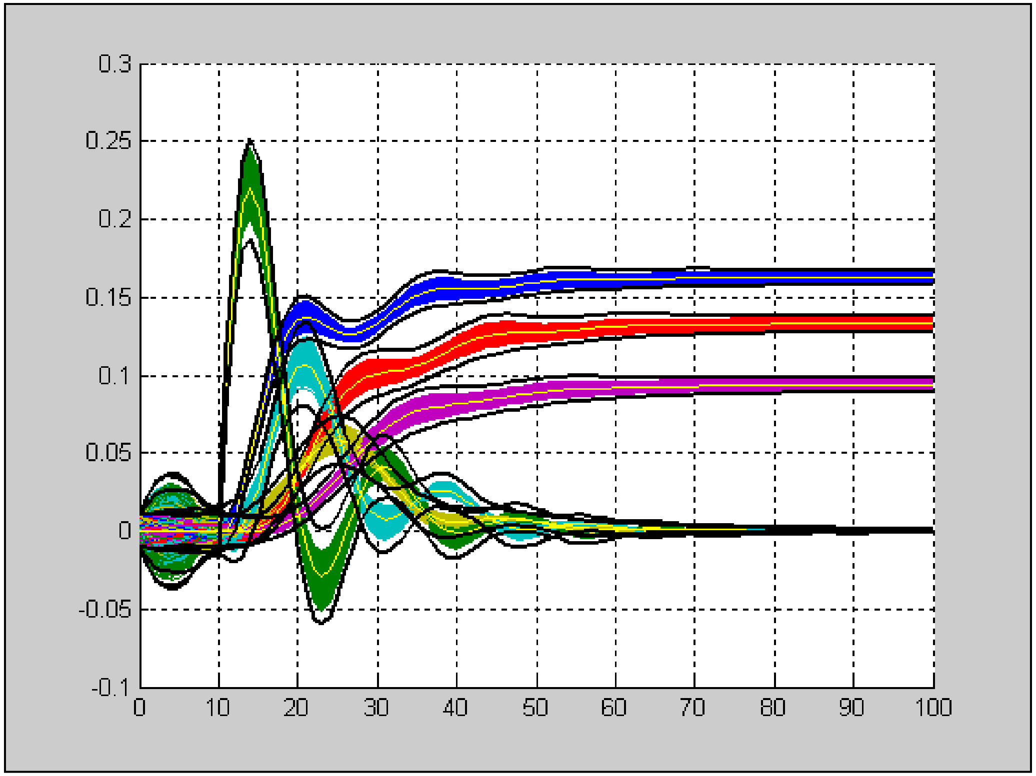
Table 1. Computation times (100 steps, Matlab implementation)

MCS (2500 simul.)	*SBS* (*Reduc*$_{200}$)	*SBNS* (*Reduc*$_{200}$)	*SBS*$_{50}$ (*Reduc*$_{50}$)
10.5 s	1.7 s	0.5 s	1.1 s

Mass-Spring Example

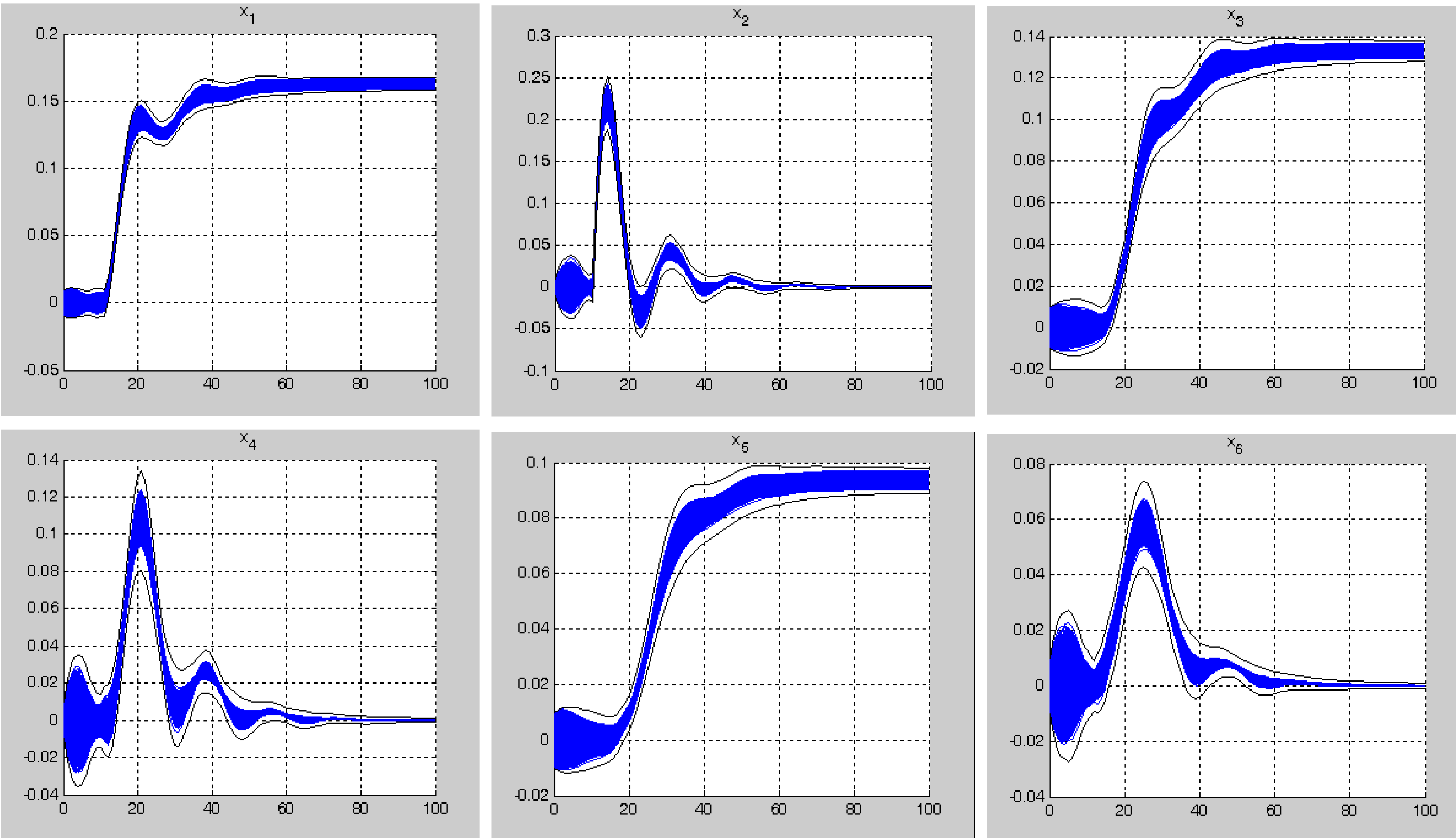
- **Dynamic system: 3 bodies and 5 springs:**

« Comparison » : Monte-Carlo (2500 simul., 10.5s) / Zonotopes (1 simul., 1.7s)

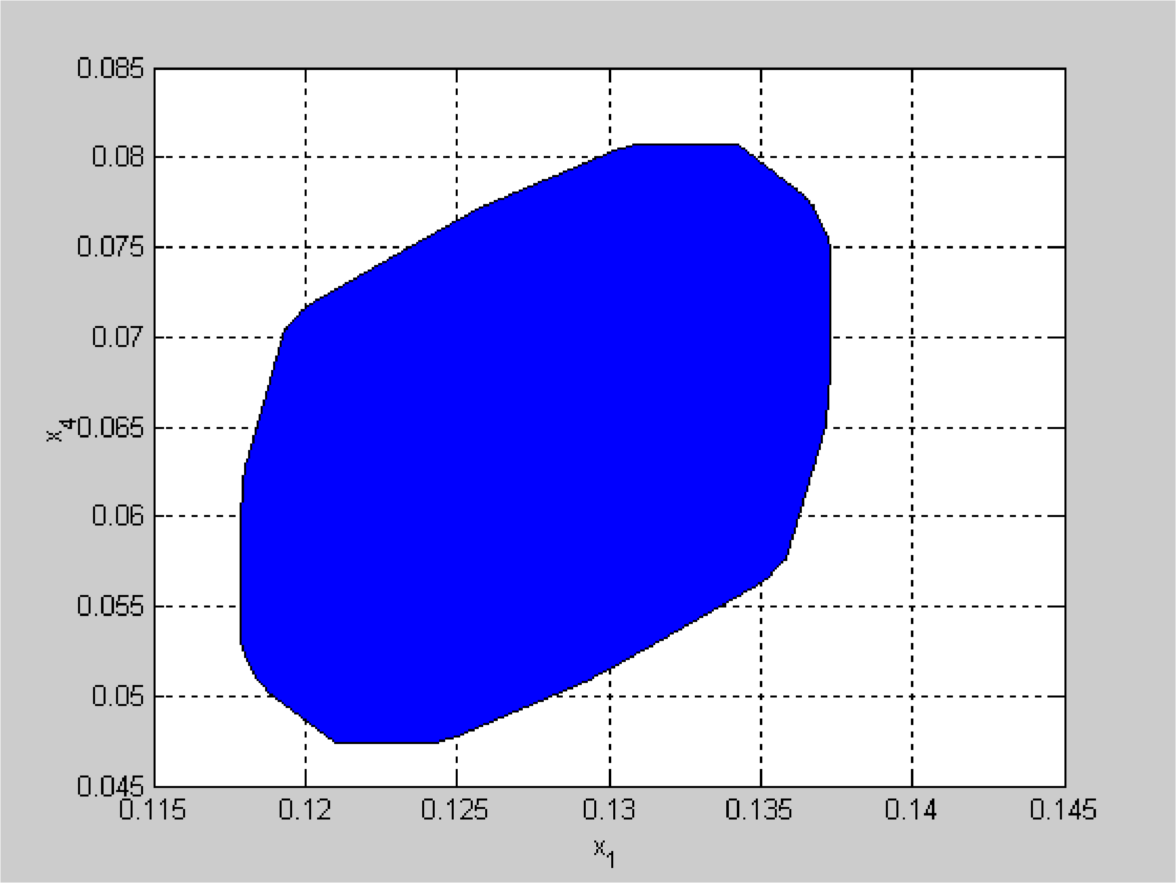


Guaranteed reachable set for continuous-time linear dynamic systems with an uncertain initial state and with parametric uncertainties (±10% viscous friction, ±2.5% stiffness).

Mass-Spring Example



Mass-Spring Example

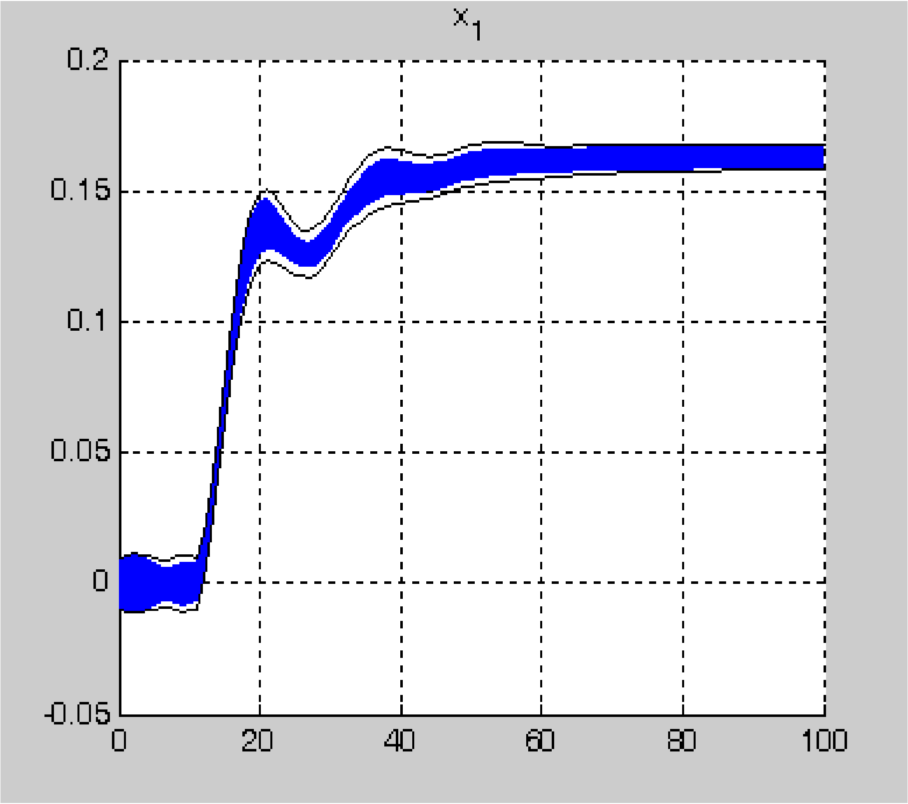


State bounds computed with structured parametric uncertainties: zonotope enclosing (x_1,x_4) at k=25

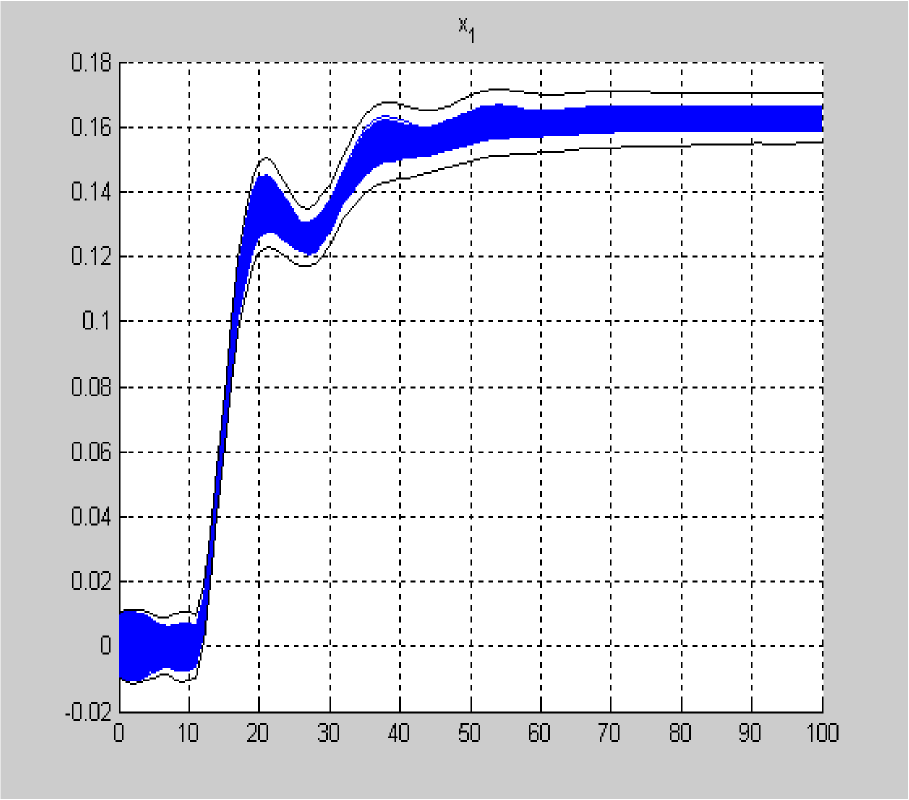
Mass-Spring Example

- **Influence of the reduction operator: Tradeoff between Pessimism and Computational load**

Red_{200}



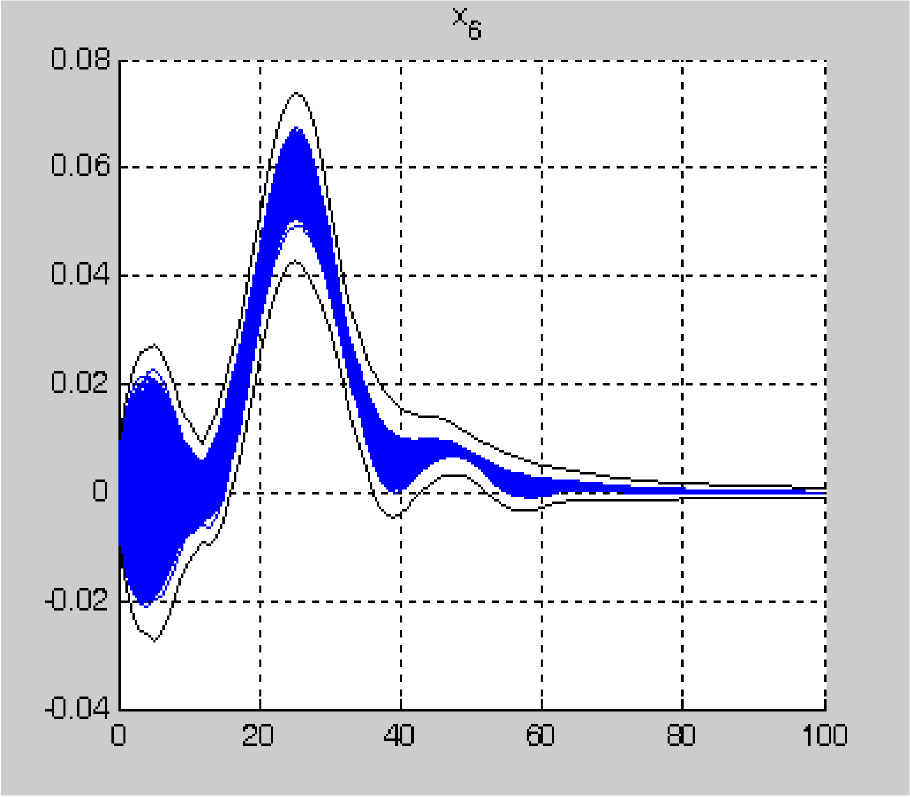
Red_{50}



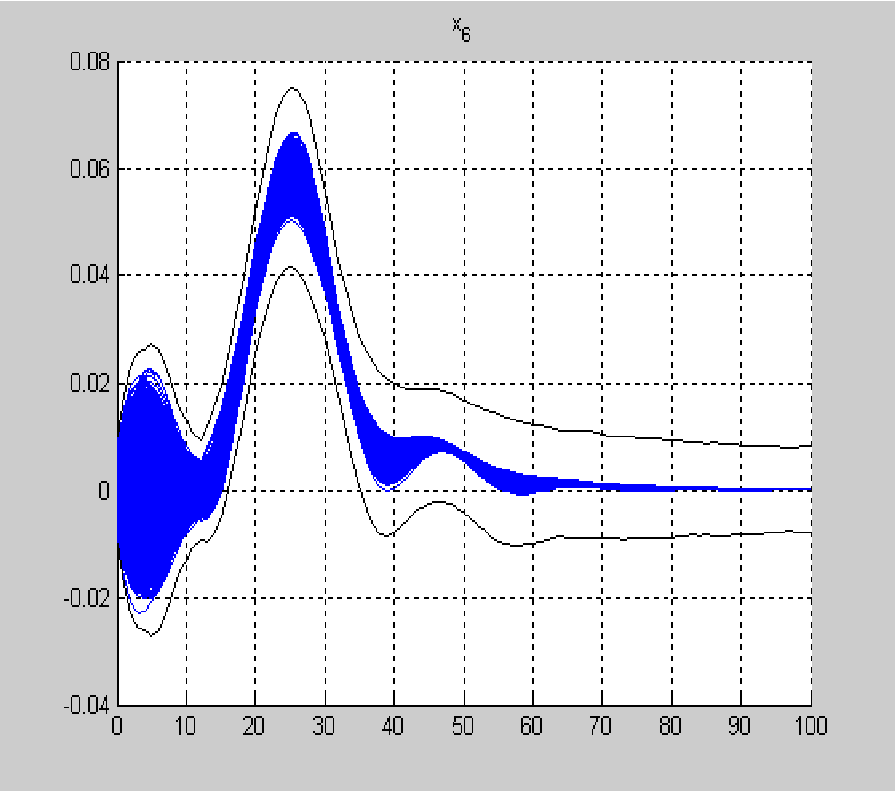
Mass-Spring Example

- **Influence of the reduction operator: Tradeoff between Pessimism and Computational load**

Red_{200}



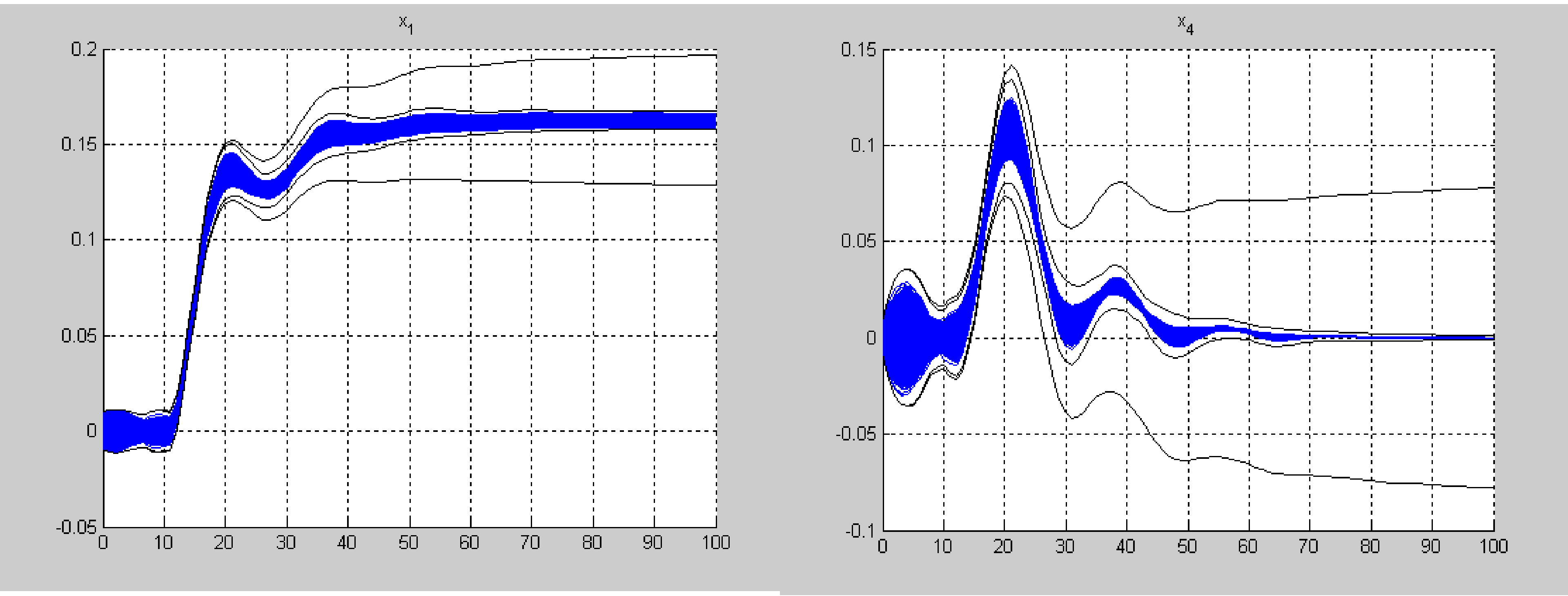
Red_{50}



Mass-Spring Example

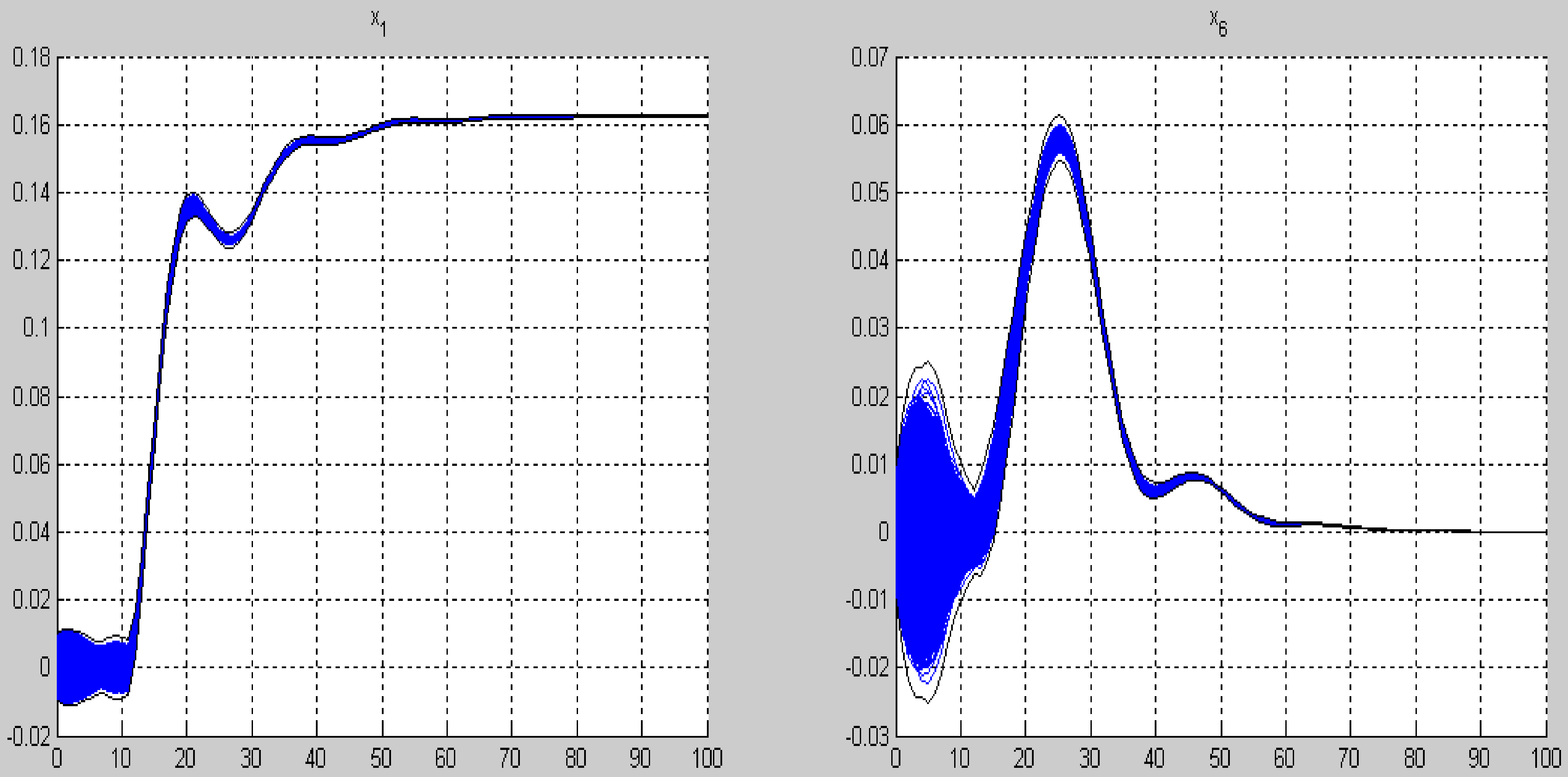
- **Interest in preserving parametric dependencies:**

PFZ versus Interval Matrices



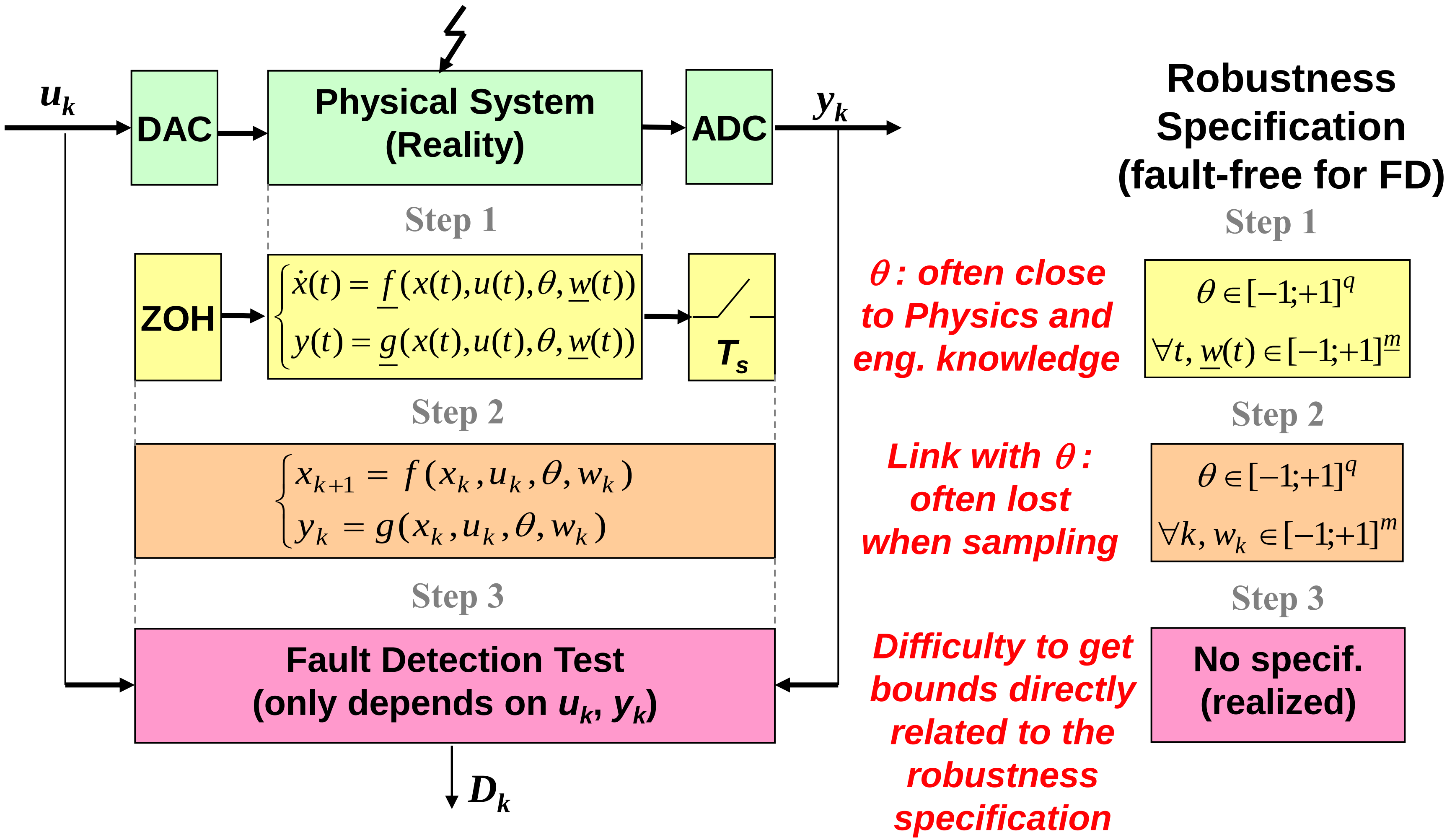
Mass-Spring Example

- **10 times smaller parametric uncertainties:**



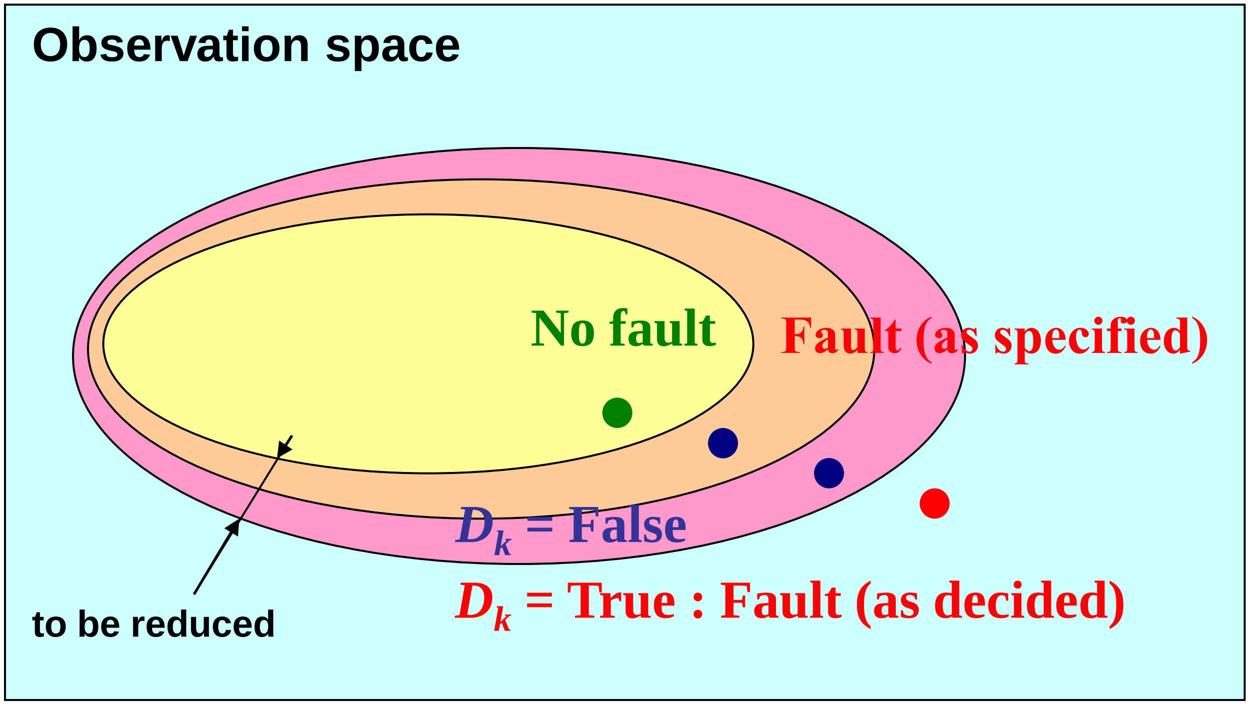
A usual design process for Fault Detection (based on a continuous-time knowledge model)

- Fault Detection design process: a sequence of modeling tasks:



Requirements for a logically consistent decision

- The set of observations (defined implicitly or explicitly) at step *s*+1 should be an outer approximation of the set at step *s* :



Robustness Specification (fault-free for FD)

Step 1

$\theta \in [-1;+1]^q$

$\forall t, \underline{w}(t) \in [-1;+1]^m$

Step 2

$\theta \in [-1;+1]^q$

$\forall k, w_k \in [-1;+1]^m$

Step 3

No specif. (realized)

- Then, no false alarm…

… provided the initial specif. (i.e. model) is valid

- Guaranteed process from the initial modeling to the final decision

Outline of the proposed approach

Physical System (Reality)

- **Step 1 : Problem formulation and system modeling**

Continuous LTI model with polynomial dependencies on θ and meas. noise

- **Step 2 : Model transformation using *afm* objects**

Continuous LTI model with affine dependencies on θ and meas. noise

- **Step 3 : Guaranteed discretization preserving affine dependencies on θ**

Discrete LTI model with affine dependencies on θ and meas. noise

- **Step 4 : Design of a set-membership test**

(Parity-like) fault detection test based on zonotopes and collision detection

- **Step 5 : Application to an ore crushing and classification process**

Problem formulation

Problem formulation and system modeling:

$$System\_OK \;\Rightarrow\; \begin{cases} \dot{x}(t) = \underline{A}(\theta).x(t) + \underline{B}(\theta).u(t) \\ y(t) = \underline{C}(\theta).x(t) + \underline{D}(\theta).u(t) + \underline{F}(\theta).w(t) \\ \\ \theta \in [-1;+1]^q \\ w(t) \in [-1;+1]^m, \\ x(t) \in R_x.[-1;+1]^n \\ \\ \forall t \in [kTs\,;(k+1)Ts[,\; u(t) = u(kTs) \end{cases}$$

- Continuous time LTI system
- $\underline{A}(.)$: matrix polynomial function (of θ), idem for the others
- Bounded measurement noise

- **Remark:** $Parametric\_fault \;\Rightarrow\; \exists i \in \{1,\ldots,q\},\; \theta_i \notin [-1;+1]$

Transformation of the initial model

- *afm* operators are used to automatically build *afm* enclosures of the initial model matrices and put them into a simplified form:

$$\dot{x}(t) = \underline{A}(\theta).x(t) + \underline{B}(\theta).u(t)$$

$$y(t) = \underline{C}(\theta).x(t) + \underline{D}(\theta).u(t) + \underline{F}(\theta).w(t)$$

$\underline{A}(.)$ **is a (matrix) polynomial function (idem for** B**, …)**

$$\dot{x}(t) = \underline{A}_\theta.x(t) + \underline{B}_\theta.u(t)$$

$$y(t) = \underline{C}_\theta.x(t) + \underline{D}_\theta.u(t) + \underline{F}_\theta.w(t)$$

$$\theta \in [-1;+1]^q \Rightarrow \boxed{\underline{A}_\theta \in [\underline{A}]_\theta = T(\underline{A}(.),\theta)}$$

$[\underline{A}]_\theta$ **=** ***affmat*** **object with affine dependencies on** $\boldsymbol{\theta}$ **(idem for** B**, …)**

- where $T(\underline{A}(.),\theta)$ consists in computing $\underline{A}(\theta)$ with *afm* operators and elementary *afm* operands (like $[\theta_i]_\theta = afm(0,\{0,\ldots,0, 1, 0,\ldots 0\},0) = \{\theta_i\}$)

Guaranteed discretization

- **Discretization with guaranteed enclosure (as long as $\theta \in [-1;+1]^q$) and preserving affine dependencies on the initial parameter vector θ:**

$$\dot{x}(t) = \underline{A}_\theta . x(t) + \underline{B}_\theta . u(t)$$
$$\underline{A}_\theta \in [\underline{A}]_\theta, \quad \underline{B}_\theta \in [\underline{B}]_\theta$$
$$\forall t \in [kT_s; (k+1)T_s[, \ u(t) = u(kT_s)$$

$$\Rightarrow$$

$$x_{k+1} = A_\theta . x_k + B_\theta . u_k$$
$$A_\theta \in [A]_\theta, \quad B_\theta \in [B]_\theta$$

$$A_\theta = \exp(\underline{A}_\theta T_s) \qquad B_\theta = \int_0^{T_s} \exp(\underline{A}_\theta t) \underline{B}_\theta dt$$

- **From a Taylor expansion of the matrix exp function (**IFAC Safeprocess'2009's paper**):**

$$[A]_\theta = \sum_{i=0}^{r} \frac{[\underline{A}]_\theta^i}{i!} T_s^i + \frac{[\underline{A}]_\theta^{r+1}}{(r+1)!} \boxed{\exp([\underline{A}][0;T_s])} T_s^{r+1}$$

$$[B]_\theta = \left(\sum_{i=0}^{r-1} \frac{[\underline{A}]_\theta^i}{(i+1)!} T_s^{i+1} + \frac{[\underline{A}]_\theta^r}{(r+1)!} \boxed{\exp([\underline{A}][0;T_s])} T_s^{r+1} \right) [\underline{B}]_\theta$$

exp function of an interval matrix [Shieh, *et al.*, 1996] + any interval matrix can be represented as an *afm* object.

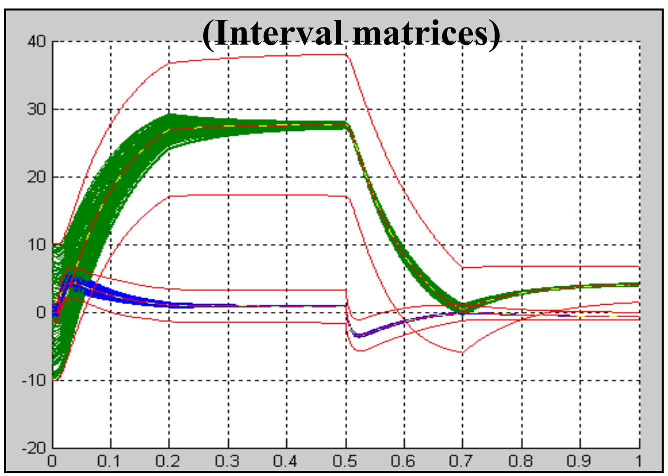
- **Then, direct computations using *afm* sums and products !**

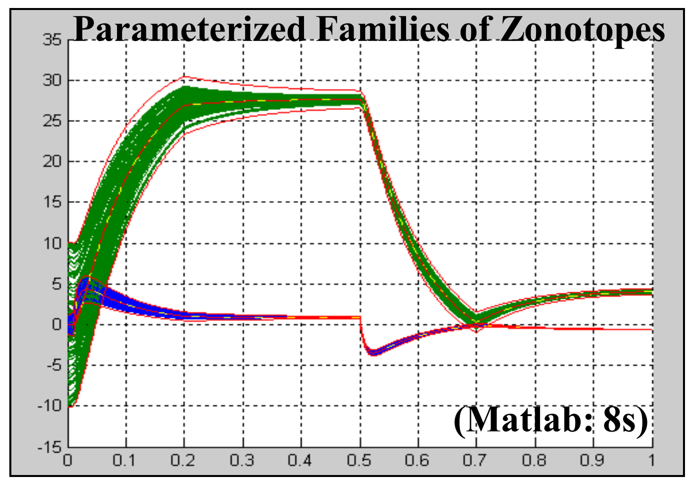
Propagation of uncertainties in dynamical systems…

- « Guaranteed » enveloppe computation:

$$\dot{x}(t) = A(\theta).x(t) + B(\theta).u(t) + E(\theta).v(t) \qquad v(t) \in [-1;+1]^p \qquad \theta \in [\theta]$$

Example: DC Motor : $A(\theta) = \begin{bmatrix} -\frac{R}{L} & -\frac{k}{L} \\ \frac{k}{J} & -\frac{f}{J} \end{bmatrix} \quad B(\theta) = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \quad E(\theta) = \begin{bmatrix} 0 \\ -\frac{C_{r,\max}}{J} \end{bmatrix}$





Guaranteed reachable set for continuous-time linear dynamic systems with an uncertain initial state and with parametric uncertainties (±5% bounds for *L*, *R*, *J*, *f*).

- Useful for <u>threshold selection</u>,
- Verification with a <u>full coverage</u>, without testing each possible scenario (infinite set when considering input and/or parametric uncertainties)

Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

Introduction

- **Computation of the reachable (state or output) set :**
  - **Verification of safety properties (using worst-case simulation)**
  - **Predictive control with input constraints**
  - **Fault Diagnosis : (adaptive) thresholds**

- **Propagation of uncertainties (initial states, inputs, …):**
  - **Stochastic context (assumptions about probability density functions…)**
  - **Deterministic context → Set-membership approaches**

- **In this work: discrete-time linear dynamical systems with:**
  - **Bounded inputs AND**
  - **Bounded slew-rate (inputs slope)**

Bounded inputs AND Bounded slew-rate

- Model of the system : $x_{k+1} = A.x_k + E.v_k$
- Bounded initial state set : $x_0 \in Z(R_0)$
- Bounded uncertain input … : $\forall k,\ \ v_k \in [-M;+M]$ «Absolute» bounds
- … with bounded slew-rate : $\forall k,\ \ v_{k+1} - v_k \in [-g;+g]$ «Relative» bounds

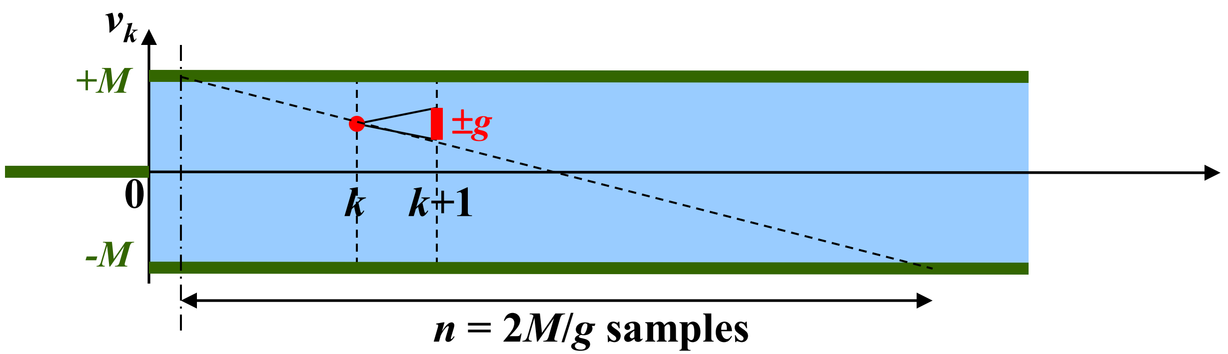
- Assumption : M is a multiple of g

- Goal : Outer approx. of $[x_k]$

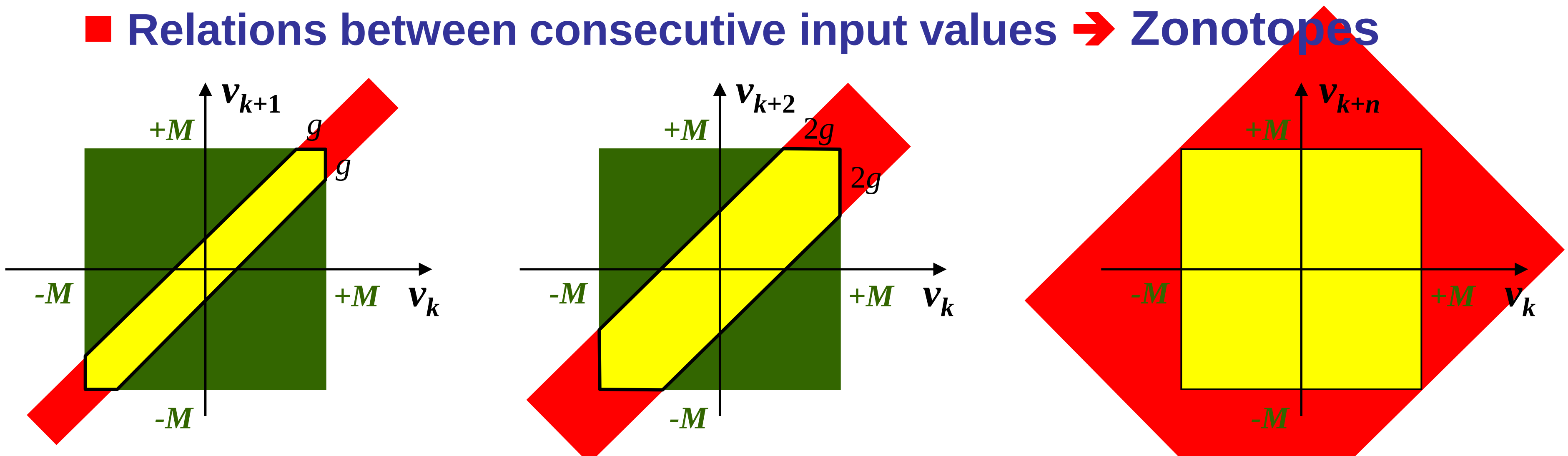
- Straightforward extensions :
  - **Time-varying discrete-time systems (*A*$_k$, *E*$_k$)**
  - **Multiple inputs (superposition theorem)**

Illustration of possible inputs

- Bounded uncertain input … : $\forall k,\ v_k \in [-M;+M]$ «Absolute» bounds
- … with bounded slew-rate : $\forall k,\ v_{k+1} - v_k \in [-g;+g]$ «Relative» bounds
- Time domain :



- Relations between consecutive input values ➔ Zonotopes



Modeling of the dependency relations

- **Rewriting of the uncertain input:**

$$\in[-1;+1] \qquad n=\frac{2M}{g}$$

$$v_k=\left(\frac{g}{2}\right)\left(s_{1,k}+\cdots+s_{i,k}+\cdots+s_{n,k}\right) \quad \Longrightarrow \quad v_k\in[-M;+M]$$

$$v_{k+1}=\left(\frac{g}{2}\right)\left(s_{1,k+1}\cdots+s_{i,k+1}+\cdots+s_{n,k+1}\right) \quad \Longrightarrow \quad v_{k+1}\in[-M;+M]$$

- **The « absolute » bounds satisfy the requirements, …**
- **… but not the « relative » bounds :**

$$[-1;+1]-[-1;+1]=[-2;+2] \quad \Longrightarrow \quad v_{k+1}-v_k\in[-2M;+2M]$$

Modeling of the dependency relations

- **Additional relation between uncertain variables:** $\boxed{s_{i,k+1} = s_{i+1,k}}$

$$\in [-1;+1] \qquad n = \frac{2M}{g}$$

$$v_k = \left(\frac{g}{2}\right)\left(\boxed{s_{1,k}} + \cdots + s_{i,k} + \cdots + s_{n,k}\right) \quad \Rightarrow \quad v_k \in [-M;+M]$$

$$v_{k+1} = \left(\frac{g}{2}\right)\left(s_{1,k+1} \cdots + s_{i,k+1} + \cdots + \boxed{s_{n,k+1}}\right) \quad \Rightarrow \quad v_{k+1} \in [-M;+M]$$

- **Then,**

$$v_{k+1} - v_k \in \left(\frac{g}{2}\right)\left(\boxed{s_{n,k+1}} - \boxed{s_{1,k}}\right) \quad \Rightarrow \quad v_{k+1} - v_k \in [-g;+g]$$

- **The « absolute » AND « relative » bounds satisfy the requirements**

Reformulation of the problem

$$x_{k+1} = A.x_k + E.v_k \longleftarrow v_k = \left(\frac{g}{2}\right)\left(s_{1,k} + \cdots + s_{i,k} + \cdots + s_{n,k}\right)$$

$$\boxed{s_{i,k+1} = s_{i+1,k}}$$

- **New model (with additional dependency relations):**

$$\begin{bmatrix} x_{k+1} \\ s_{1,k+1} \\ \vdots \\ s_{n-1,k+1} \\ s_{n,k+1} \end{bmatrix} = \left[\begin{array}{c|cccc} A & E\dfrac{g}{2} & E\dfrac{g}{2} & \cdots & E\dfrac{g}{2} \\ \hline & 0 & 1 & 0 & 0 \\ 0 & \vdots & \ddots & \ddots & 0 \\ & \vdots & & \ddots & 1 \\ & 0 & \cdots & \cdots & 0 \end{array}\right] . \begin{bmatrix} x_k \\ s_{1,k} \\ \vdots \\ s_{n-1,k} \\ s_{n,k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} . s_{n+1,k} \qquad \forall k, \quad s_{n+1,k} \in [-1;+1]$$

$$\begin{bmatrix} x_0 \\ s_{1\ldots n,0} \end{bmatrix} \in Z\left(\begin{bmatrix} R_0 & 0 \\ 0 & I_n \end{bmatrix}\right)$$

- **New goal: Reachable state set of a system with bounded inputs**

➔ **The previous algorithm can be used ! ...**

Reduction preserving some dependencies

- … and optimized with a slight modification of the reduction step…

$$R_{k+1} = \boldsymbol{Red}(R_{k+1}) \quad \Longrightarrow \quad \begin{array}{l} R_{k+1} = [\ S_{k+1} \quad T_{k+1}\ ] \\ R_{k+1} = [\ \boldsymbol{Red}(S_{k+1}) \quad T_{k+1}\ ] \end{array}$$

$\longleftrightarrow$

(n+1) segments

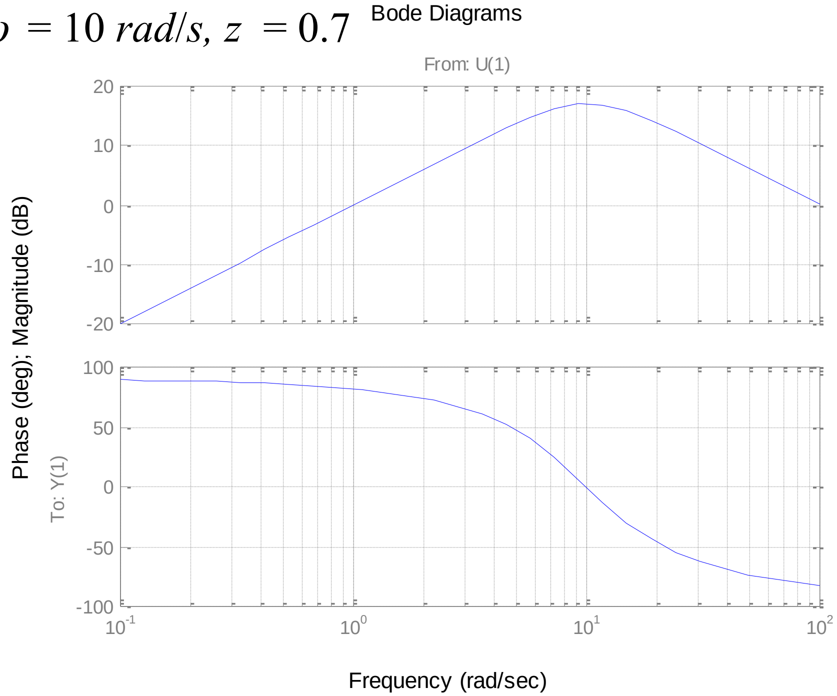
➔ Additional dependencies are not disturbed by the reduction step

Simulation example

- **Discretization of a 2nd order pass-band filter :**

$$F_c(p) = \frac{p}{\frac{p^2}{\omega^2} + 2z\frac{p}{\omega} + 1} \quad \overset{T_s = 0.05\ s}{\Longrightarrow} \quad \begin{aligned} x_{k+1} &= A.x_k + E.v_k \\ y_k &= C.x_k \end{aligned}$$

$\omega = 10\ rad/s,\ z = 0.7$

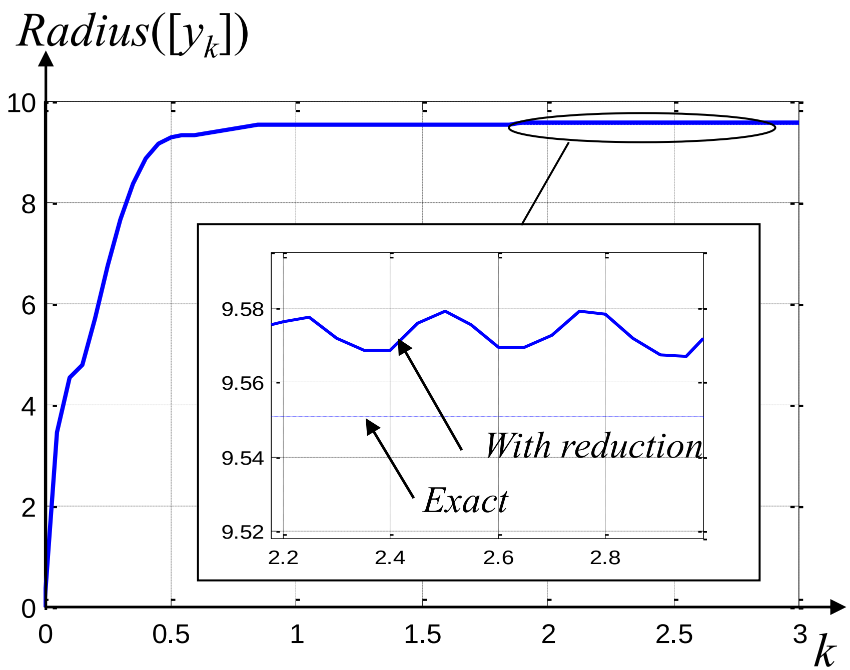


$$A = \begin{bmatrix} 1.3205 & -0.24829 \\ 2 & 0 \end{bmatrix} \quad E = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.7245 & -0.86226 \end{bmatrix}$$

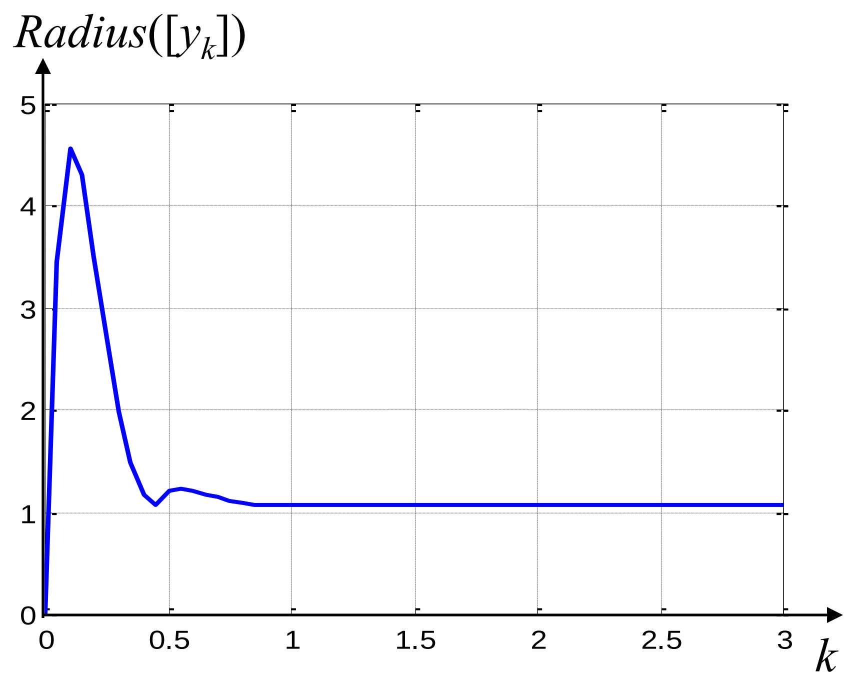
Simulation results

Bounded input only



$v_k \in [-1;+1]$

Bounded input AND Bounded slew-rate



$v_k \in [-1;+1]$

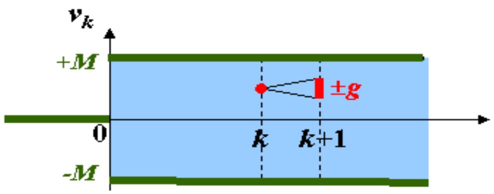
$v_{k+1} - v_k \in [-0.05;+0.05]$

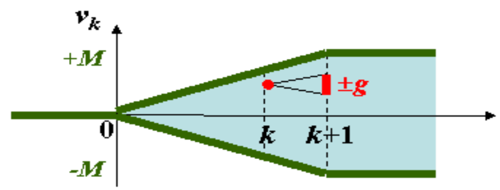
$M = 1$

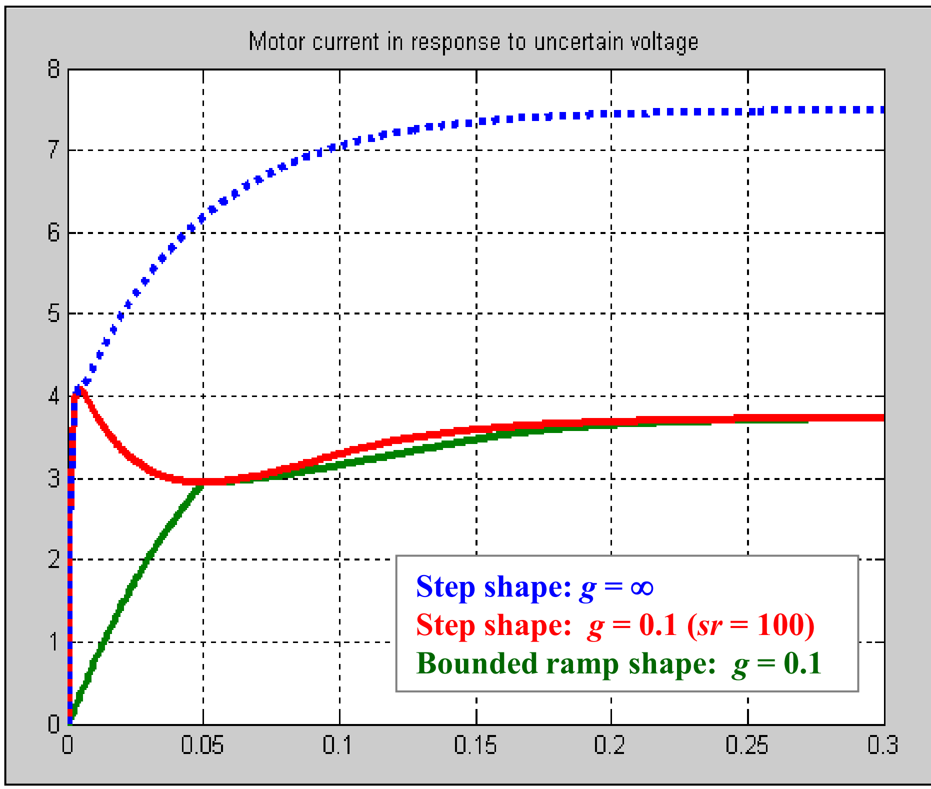
$M = 1;\quad g = 0.05;\quad n = 40$

DC Motor Example

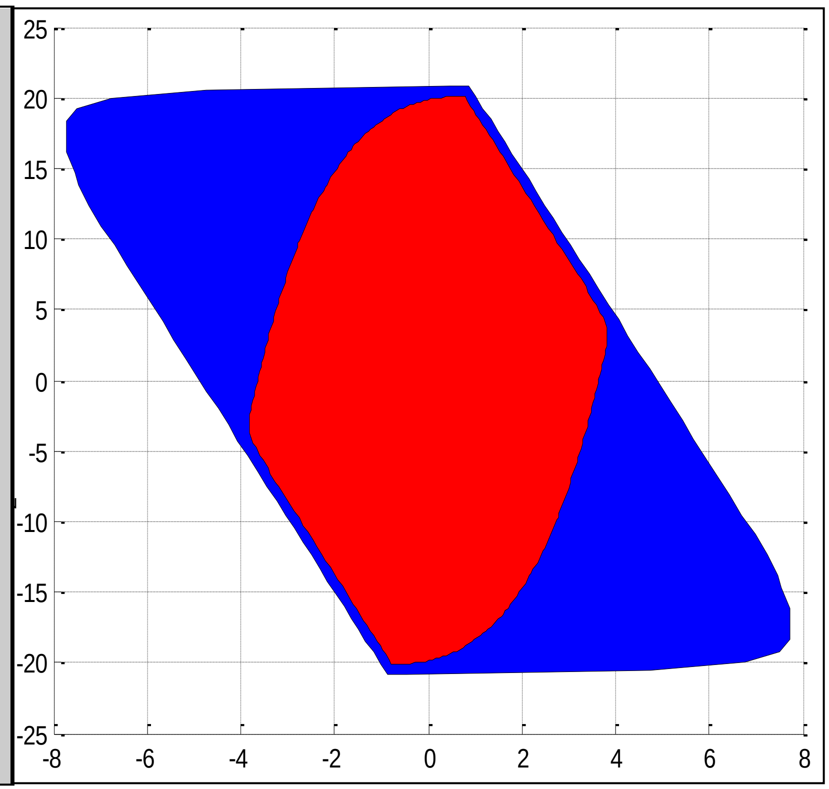
- Current (and speed) envelopes for different voltage specifications:







$M = 5; \quad T = 0.001$



(*Current, Speed*)

Outline

1) Introduction

2) Zonotopes: definition, properties, basic prediction algorithm

3) Application to fault diagnosis (using an adaptive observer)

4) Dealing with parametric uncertainties

5) Dealing with bounded inputs & bounded slew-rate

6) Conclusion

Conclusion and Future work

Conclusion:

- Efficient modeling and propagation of dependency relations between uncertain variables ➔ key point to ↓ wrapping effect
- Zonotopes: ability to propagate large uncertainties within possibly (sampled-)time varying linear dynamics
- Parameterized Families of Zonotopes
- Preservation of parameter dependencies under sampling
- Links between Verification and (model-based) Fault Diagnosis

Future work:

- "Reasonable" theoretical error bounds and reduction…
- Interval observers
- Guaranteed inclusion of non-linear dynamics
- Dealing with guards in Hybrid Dynamical Systems
- Specifications of uncertain inputs and parameters + Model validation

The End