State Estimation of Uncertain Non Linear Systems
A Bounding Approach based on Zonotopes

Christophe COMBASTEL

école nationale supérieure de l'électronique
et de ses applications

Outline

- Introduction to zonotopes and state bounding
- The observation algorithm (bounded errors)
- Application (bioreactor)
- Conclusion & future prospects

Introduction

- **Observation:**

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**

Domain representation

- **Solutions based on domain approximations:**
  - **Orthotope (i.e. interval vector or aligned boxes)**
  - **Parallelotope**
  - **Ellipsoïd**
  - **Zonotope**
  - **Polytope (+simplification)**

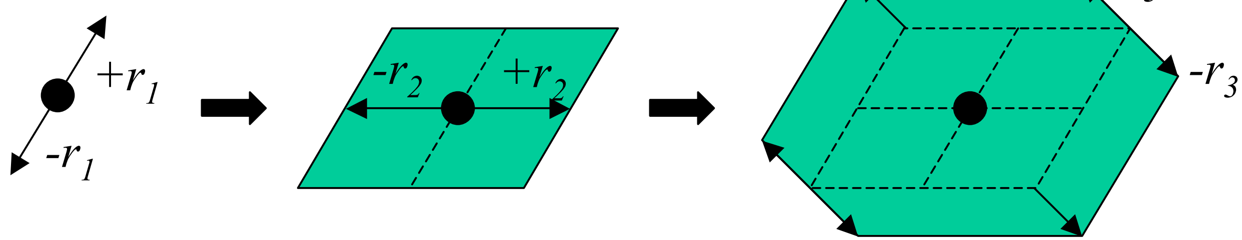
Zonotopes : definitions

- **Zonotope = Minkowski sum of straight line segments**

$$[Z] = [S_1] + \cdots + [S_p]$$

$$[S_i] = c_i + r_i.\square \qquad \square = [-1;1]$$

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



- **Zonotope = Linear image of a <u>*p*-hypercube</u> in a *n*-space (abstract space)**

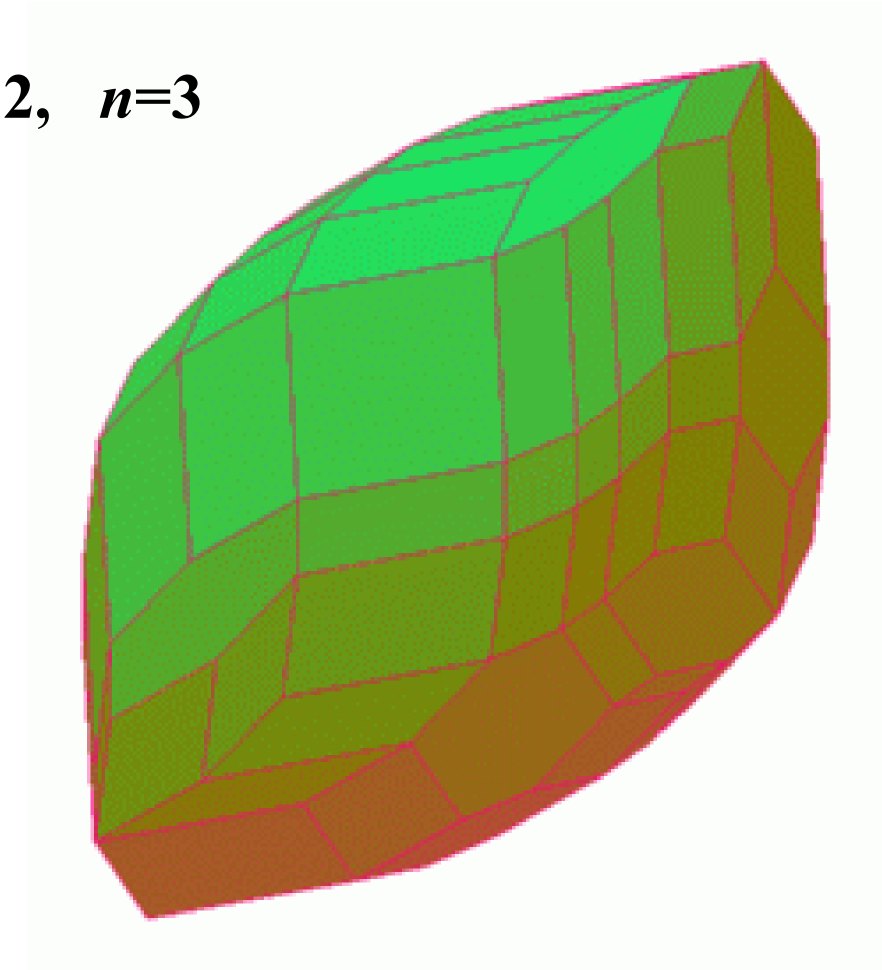
$$[Z] = c + \underbrace{R.\square^p}$$

$\blacklozenge R$

$$[Z] \subset \Re^n \qquad c \in \Re^n \qquad R \in \Re^{n \times p}$$

Zonotopes : an example

p=12, *n*=3



http://www.decatur.de/wolfgang/zono/index.html

(Centered) Zonotopes : Properties

- **Sum of two zonotopes :**

$$\Diamond R_1 + \Diamond R_2 = \Diamond [R_1\ R_2]$$

➔ **matrix concatenation**

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

$$L.(\Diamond R) = \Diamond (L.R)$$

➔ **matrix product**

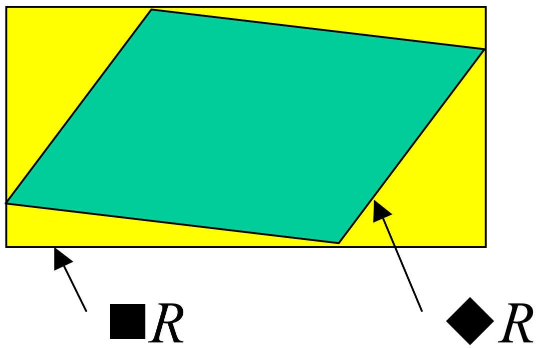
- **Smallest « box » containing a zonotope (« interval hull »):**

$$\blacksquare R = \Diamond rs(R)$$

➔ **Norm of each line vector**

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

(the dashed lines in the matrix mark row and column i)



Problem formulation n°1

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

$$x_{k+1} = A_k.x_k + B_k.u_k + E_k.v_k$$

$$y_k = C_k.x_k + D_k.u_k + F_k.w_k$$

$$v_k \in \square^n \qquad w_k \in \square^m \qquad \square = [-1;1]$$

- **Known:** $A_k, B_k, C_k, D_k, E_k, F_k,\quad u_k, y_k,\quad [x_0] \;/\; x_0 \in [x_0]$
- **Unknown:** x_k, v_k, w_k
- **Goal:** $[x_k]$, **smallest domain such that** $x_k \in [x_k]$ **is guaranteed.**

Problem formulation n°2

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

$$\dot{x}(t) = f(x(t), u(t), v(t))$$
$$y(t) = g(x(t), u(t), w(t))$$

$v(t) \in \square^r$

$w(t) \in \square^m$

$\square = [-1;1]$

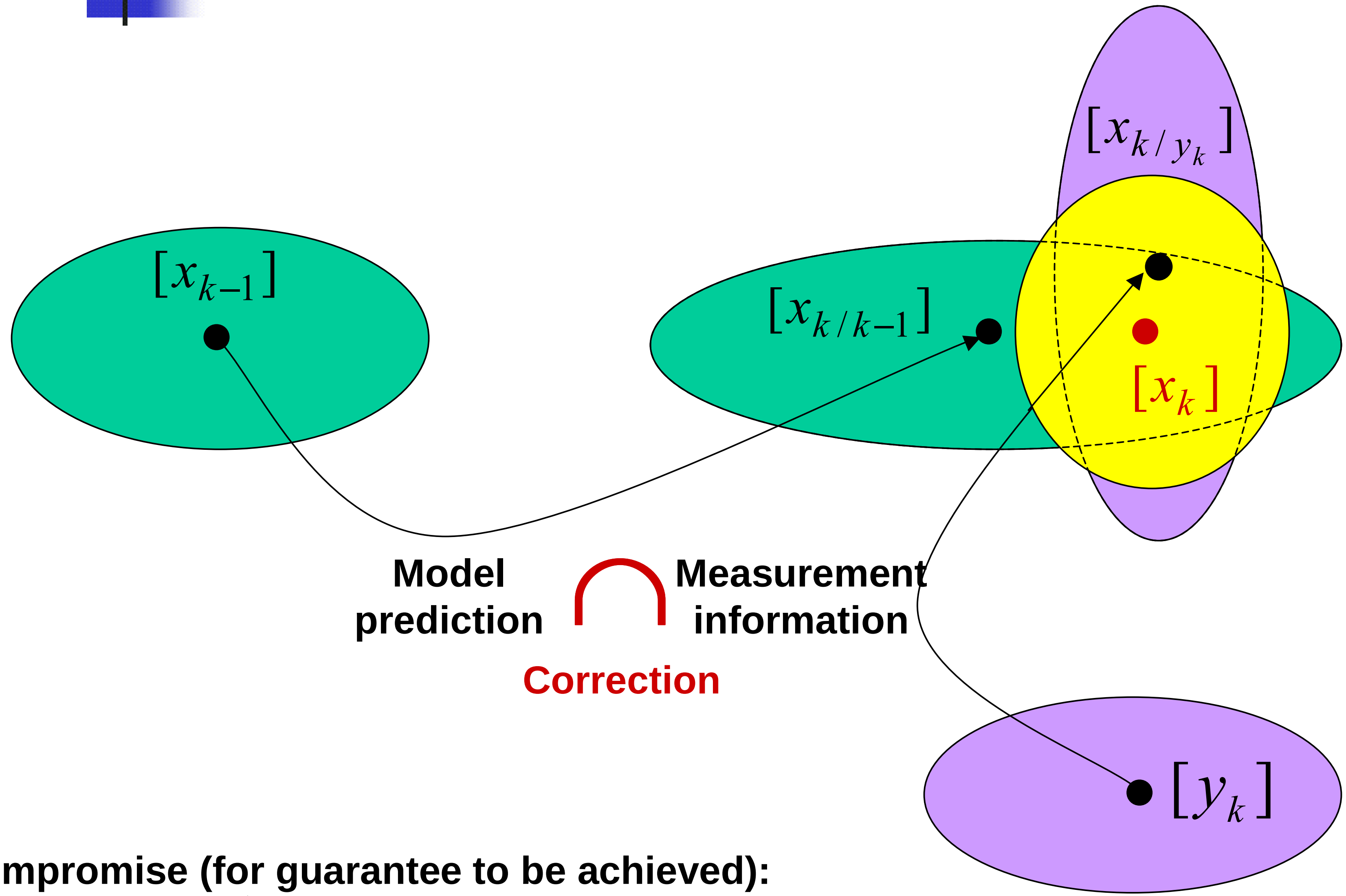
- **Sampling:**
  - **Notations:** $x_k = x(kT_s), \ldots$
  - **Zero order hold:** $u(t) = u_k, \quad t \in [kT_s \,;\, (k+1)T_s[$

- **Known:** $[x_0] \;/\; x(0) \in [x_0]$ $\quad u_k, y_k$ $\quad \begin{cases} f(.) \in \mathcal{C}^2, \text{ local. Lipschitz} \\ g(.) \in \mathcal{C}^2 \end{cases}$

- **Unknown:** $x(t), v(t), w(t)$

- **Goal:** $[x_k]$, **(smallest) domain such that** $x_k \in [x_k]$ **is guaranteed.**

Bounded error state estimation



Compromise (for guarantee to be achieved):
↑ exactness ⇔ ↑ complexity ⇔ ↓ outer approximation

Main steps of the observation algorithm

Initialization ($[x_{1/0}]$ is computed from $[x_0]$)

For k=1 to k_{max}

System at time k: u_k, y_k

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

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

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

End

Outline

- Introduction to zonotopes and state bounding
- The observation algorithm (bounded errors)
- Application (bioreactor)
- Conclusion & future prospects

Main steps of the observation algorithm

Initialization ($[x_{1/0}]$ is computed from $[x_0]$)

For k=1 to k_{max}

System at time k: u_k, y_k

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

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

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

End

One step prediction

Prediction:

- **From:** $[x_k] = c_k + \blacklozenge R_k$ $\quad x_k \in [x_k]$
- **And:** $\dot{x}(t) = f(x(t), u(t), v(t))$ $\quad v(t) \in [-1;+1]^n$
- **Compute:** $[x_{k+1/k}] = c_{k+1/k} + \blacklozenge R_{k+1/k}$ $\quad x_{k+1} \in [x_{k+1/k}]$

How ?

- **Discretization**
- **Linearization**
- **Inclusion of the linearization error**
- **Inclusion of the discretization error**

Discretization

$$\dot{x}(t) = f(x(t), u(t), v(t))$$

- **Discretization: Euler (for the sake of clarity)**

$$x_{k+1} = x_k + f(x_k, u_k, v_k).T_s + ed_k$$ ← **Discretization error at time *k***

- **Inclusion:**

$$x_{k+1} \in [x_k] + f([x_k], u_k, [v_k]).T_s + [ed_k]$$

$$[x_k] = c_k + \blacklozenge R_k$$
$$[v_k] = [-1;+1]^r$$
$$[ed_k] = cd_k + \blacklozenge Rd_k$$ ← **Inclusion of the discretization error at time *k***

$$x_{k+1} \in \left\{ c_k + R_k.s + f(c_k + R_k.s, u_k, v).T_s + [ed_k],\ \ s \in [-1;+1]^p, v \in [-1;+1]^r \right\}$$

Linearization

$$x_{k+1} \in \left\{ c_k + R_k.s + f(c_k + R_k.s, u_k, v).T_s + [ed_k], \quad s \in [-1;+1]^p, v \in [-1;+1]^r \right\}$$

Linearization of *f* around (c_k,u_k,0):

$$f(c_k + R_k.s, u_k, v) = f(c_k, u_k, 0) + L_k.R_k.s + M_k.v + el_k(R_k.s, v)$$

$$L_k = \left. \frac{\partial f(x,u,v)}{\partial x} \right|_{\substack{(x,u,v)= \\ (c_k,u_k,0)}} \qquad M_k = \left. \frac{\partial f(x,u,v)}{\partial v} \right|_{\substack{(x,u,v)= \\ (c_k,u_k,0)}}$$

$$[el_k] = cl_k + \blacklozenge Rl_k$$ ← Inclusion of the linearization error at time *k*

Prediction:

$$x_{k+1} \in [x_{k+1/k}] = c_{k+1/k} + \blacklozenge R_{k+1/k}$$

$$c_{k+1/k} = c_k + f(c_k, u_k, 0).T_s + cl_k.T_s + cd_k$$

$$R_{k+1/k} = [(I + L_k.T_s).R_k \quad M_k.T_s \quad Rl_k.T_s \quad Rd_k]$$

Inclusion of the linearization error

- **With no loss of generality (*):**

$$\begin{bmatrix} \vdots \\ f_q(c_k + R_k.s, u_k) \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ f_q(c_k, u_k) \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ f_q^{[1]}(c_k, R_k.s, u_k) \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ el_{q,k} \\ \vdots \end{bmatrix} \qquad q = 1\ldots n$$

$f(c_k + R_k.s, u_k)$ — **Constant** (wrt s) — **Linear** — el_k

$$el_{q,k} = \underbrace{f_q^{[2]}(c_k, R_k.s, u_k)}_{\textbf{Quadratic}} + \underbrace{f_{q[3]}(c_k, R_k.s, u_k)}_{\textbf{Higher order terms}} \qquad s \in [-1;+1]^p$$

- **Goal:**

$$[el_k] = \left(c_k^{[2]} + \blacklozenge R_k^{[2]}\right) + \left(c_{[3]k} + \blacklozenge R_{[3]k}\right)$$

(*) When f depends on v, $c_k \leftarrow [c_k\ 0]^T$, $R_k \leftarrow [R_k\ 0;\ 0\ I]$, $s \leftarrow [s\ v]^T$.

Inclusion of quadratic terms

- **Quadratic form related to the Taylor dev. of f_q around (c_k,u_k):**

$$f_q^{[2]}(c_k, R_k.s, u_k) = (R_k s)^T . Q_q(c_k, u_k) . (R_k s)$$

$n \times n$ matrix obtained from a formal calculus software

$$= s^T . \underline{Q}_{q,k} . s \qquad \underline{Q}_{q,k} = R_k^T . Q_q(c_k, u_k) . R_k$$

$$= \sum_i \underline{Q}_{q,k,ii} s_i^2 + \sum_{i<j} (\underline{Q}_{q,k,ij} + \underline{Q}_{q,k,ji}) s_i s_j$$

$$s_i^2 \in [0;+1] = \frac{1}{2} + \frac{[-1;+1]}{2} \qquad s_i s_j \in [-1;+1]$$

Inclusion of quadratic terms

- **Result:**

$$[f^{[2]}(c_k, R_k.s, u_k)] \subseteq c_k^{[2]} + \blacklozenge R_k^{[2]} \qquad c_k^{[2]} = \begin{bmatrix} \vdots \\ c_{q,k}^{[2]} \\ \vdots \end{bmatrix} \quad R_k^{[2]} = \begin{bmatrix} \vdots \\ R_{q,k}^{[2]} \\ \vdots \end{bmatrix}$$

$$c_{q,k}^{[2]} = trace(\underline{Q}_{q,k})/2$$

$$R_{q,k}^{[2]} = [\cdots \underbrace{\underline{Q}_{q,k,ii}/2}_{\forall i} \cdots \mid \cdots \underbrace{(\underline{Q}_{q,k,ij} + \underline{Q}_{q,k,ji})}_{\forall i<j} \cdots]$$

- **An illustrative example:**

$$\begin{cases} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} f_1(x_1,x_2) \\ f_2(x_1,x_2) \end{bmatrix} = \begin{bmatrix} -x_1 + (x_1x_2)/2 \\ -x_2 - x_1x_2 \end{bmatrix} \\ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} . \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \end{cases} \Rightarrow$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -3 & -1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} . + \begin{bmatrix} s_1^2 + s_1s_2 \\ -2.s_1^2 - 2.s_1s_2 \end{bmatrix}$$

$$\boxed{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \begin{bmatrix} 1/2 \\ -2 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1/2 & 1 \\ -3 & -1 & -1 & -2 \end{bmatrix} . \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}}$$

- **Quadratic terms disappear**
- **Linear dependence between similar quadratic terms is kept**

Inclusion of higher order terms

- **Rest expressed as a non linear function of linear forms:**

$$f_{q[3]}(c_k, R_k.s, u_k) = \widetilde{f}_{q[3]}(c_k, \psi(c_k, u_k).R_k.s, u_k)$$

Matrix (formal calculus can be used)
No general methodology at present (application dependent)

> **Goal: take linear dependence into account as much as possible before interval inclusion**

$$f_{q[3]}(c_k, R_k.s, u_k) \in \widetilde{f}_{q[3]}(c_k, \blacksquare(\psi(c_k, u_k).R_k), u_k)$$

- **Result (use of interval arithmetic):**

$$[f_{[3]}(c_k, R_k.s, u_k)] \subseteq c_{[3]k} + \blacklozenge R_{[3]k}$$

Main steps of the observation algorithm

Initialization ($[x_{1/0}]$ is computed from $[x_0]$)

For k=1 to k_{max}

System at time k: u_k, y_k

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

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

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

End

Correction

- **Correction: taking measurements into account**

$$y_k = g(x_k, u_k, w_k) \qquad \begin{cases} x_k \in [x_{k+1/k}] = c_{k+1/k} + \blacklozenge R_{k+1/k} \\ w_k \in [w_k] = [-1;+1]^m \end{cases}$$

- **Linearization + inclusion of the related error (like in prediction)**

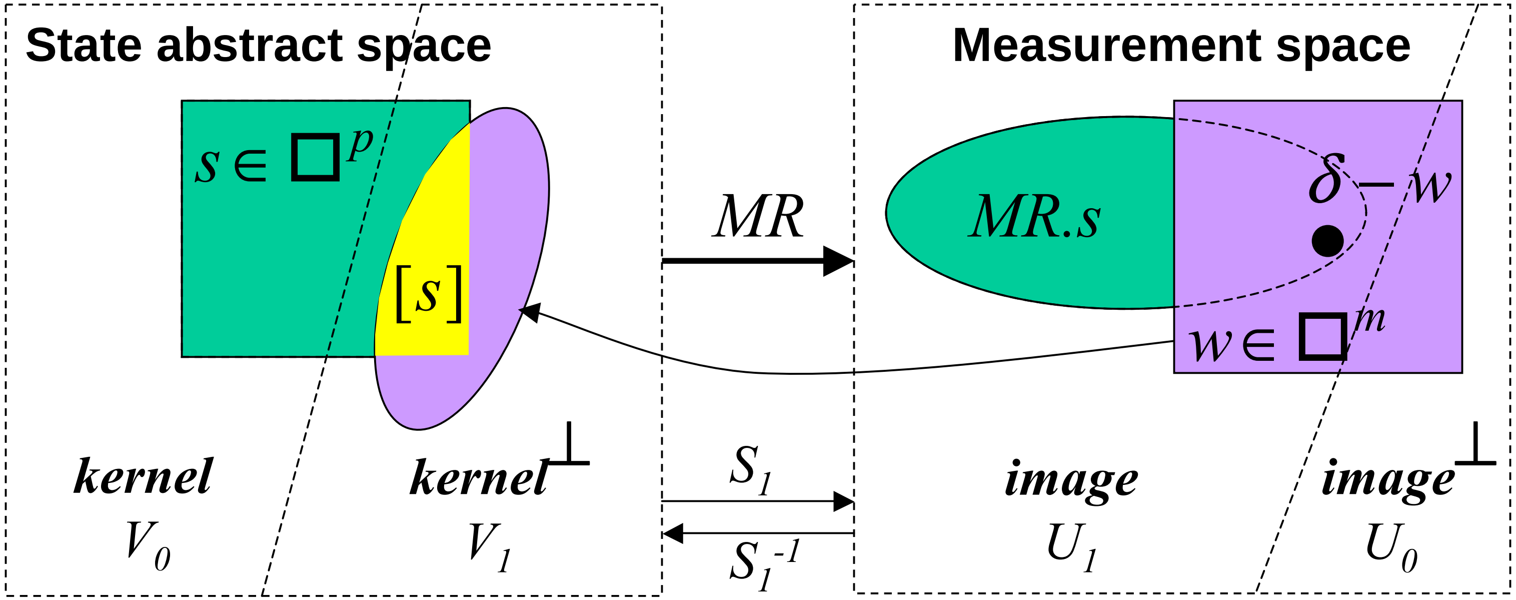
$$y_k = d_k + \underline{C}_k.s + \underline{F}_k.\underline{w}, \quad s \in [-1;+1]^p, \underline{w} \in [-1;+1]^{\underline{m}}$$

➔ **Correction in the linear case can be applied.**

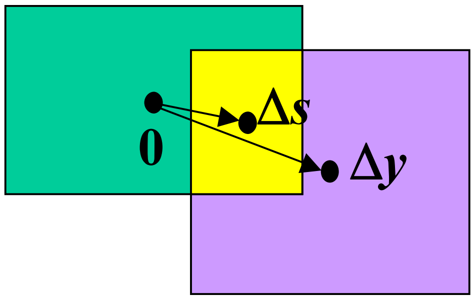
$$y_k = C_k.x_k + D_k.u_k + F_k.w_k$$

Correction using measurements

- **Principle:**



- **In the subspace influenced by the measurement (V_1):**



Center of the corrected domain: depends on uncertainties

Main steps of the observation algorithm

Initialization ($[x_{1/0}]$ is computed from $[x_0]$)

For k=1 to k_{max}

System at time k: u_k, y_k

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

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

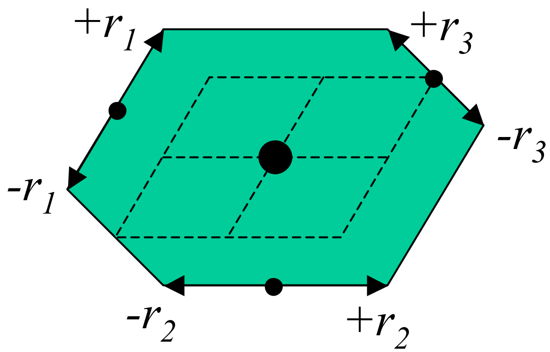
$[x_{k+1/k}] \leftarrow$ **Reduction**($[x_{k+1/k}]$)

End

Reduction

- **Zonotopes:**

$$\lozenge R = \lozenge[r_1 \ \cdots \ r_i \ \cdots \ r_p] \Rightarrow$$



(p=3)

- **Choose the zonotope complexity (max. *nd* segments)**
- **Sort on decreasing Euclidian norm:** $\|r_i\| \geq \|r_{i+1}\|$
- **Reduction:**

$$Red(\lozenge R) = \lozenge[\text{first (nd-n) col.}] + \blacksquare[\text{other col.}]$$

$$Red(\lozenge R) = \lozenge[\text{first (nd-n) col.} \ \ \text{n col.}]$$

Outline

- Introduction to zonotopes and state bounding
- The observation algorithm (bounded errors)
- Application (bioreactor)
- Conclusion & future prospects

Application

- **Model of a bio-reactor in fed-batch mode (Dochain, 2001):**

$$S \rightarrow X \qquad F_{out} = 0$$

$$\begin{cases} u = D = F_{in} / V \\ \begin{bmatrix} \dot{X} \\ \dot{S} \\ \dot{V} \\ \dot{M} \\ \dot{K} \end{bmatrix} = \dot{x} = f(x,u) = \begin{bmatrix} (\mu - D).X \\ -\mu.X / Y_{xs} + D.(S_{in} - S) \\ D.V - F_{out} \\ 0 \\ 0 \end{bmatrix} \\ y = S + F.w \qquad w \in [-1;+1] \end{cases}$$

Law of Monod:

$$\mu = \mu(K, M, S) = \frac{M.S}{K + S}$$

State extension: parameter estimation

$[x(0)]$ **: Initial state (and parametric) uncertainty**

S (resp. ***X***) : concentration of the substrate (resp. biomass).
D : dilution rate. ***V*** : volume. S_{in} : input concentration of the substrate.
Y_{xs} : efficiency coefficient for the growth of the biomass on the substrate.
μ : specific growth rate of the biomass. F_{in}, F_{out}: input and output flow.

Inclusion of the linearization error

- **Higher order terms:**

$$f_{[3]}(c_k, R_k.s, u_k) = \begin{bmatrix} (\mu X)_{[3]}(c_k, R_k.s) \\ -(\mu X)_{[3]}(c_k, R_k.s)/Y_{xs} \\ 0_{3\times 1} \end{bmatrix}$$

$$\mu X(x) = \frac{M.S.X}{K+S}$$

- **Inclusion of $\mu X_{[3]}$:** $[\mu X_{[3]}(c_k, R_k.s)] = c_{I,k} + \blacklozenge R_{I,k}$

- **Result:** $[f_{[3]}(c_k, R_k.s, u_k)] \subseteq c_{[3]k} + \blacklozenge R_{[3]k}$

$$c_{[3]k} = \begin{bmatrix} c_{I,k} \\ -c_{I,k}/Y_{xs} \\ 0_{3\times 1} \end{bmatrix} \qquad R_{[3]k} = \begin{bmatrix} R_{I,k} \\ -R_{I,k}/Y_{xs} \\ 0_{3\times 1} \end{bmatrix}$$

Inclusion of the linearization error

- **Inclusion of $\mu X_{[3]}$:**

$$\mu X(x+\delta x) = \frac{(M+\delta M).(S+\delta S).(X+\delta X)}{(K+\delta K)+(S+\delta S)} \qquad x = \begin{bmatrix} X \\ S \\ V \\ M \\ K \end{bmatrix}$$

- **Expression of $\mu X_{[3]}$ as a non linear function of linear forms:**

$$\mu X_{[3]}(x,\delta x) = \frac{(\Psi_1(x).\delta x).(\Psi_2(x).\delta x).(\Psi_3(x).\delta x)}{(\Psi_4(x).x)^3.(\Psi_4(x).(x+\delta x))}$$

Linear forms:

$$\Psi_1(x) = \begin{bmatrix} 0 & K & 0 & 0 & -S \end{bmatrix}$$

$$\Psi_2(x) = \begin{bmatrix} (K+S) & -X & 0 & 0 & -X \end{bmatrix}$$

$$\Psi_3(x) = \begin{bmatrix} 0 & -M & 0 & (K+S) & -M \end{bmatrix}$$

$$\Psi_4(x) = \Psi_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- **Interval inclusion:**

$$\begin{cases} x = c_k \\ \delta x = R_k.s \end{cases} \Rightarrow \quad I_{i,k} = \blacksquare(\Psi_i(c_k).R_k)$$

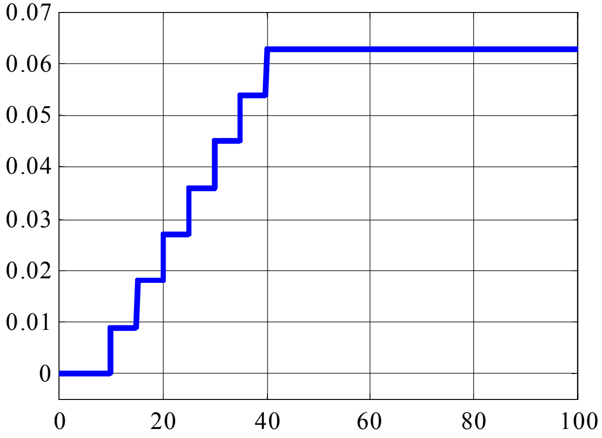
$$[\mu X_{[3]}(c_k, R_k.s)] = \frac{I_{1,k}.I_{2,k}.I_{3,k}}{(\Psi_4.c_k)^3.(\Psi_4.c_k + I_{4,k})}$$

Inclusion of the discretization error

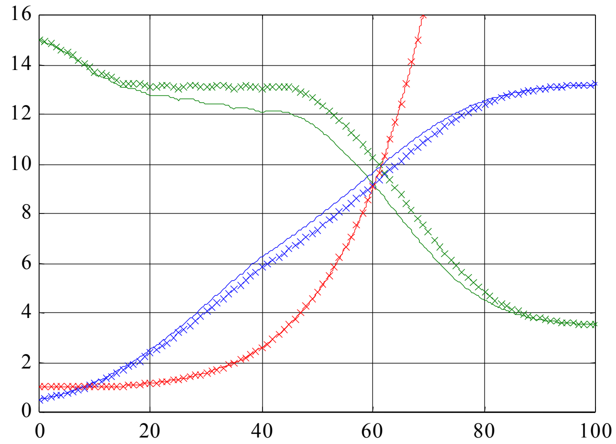
- **Euler scheme:** $ed_k = x_{k+1} - x_k - f(x_k, u_k, v_k).T_s$

- **Empirical evaluation (simulation):**

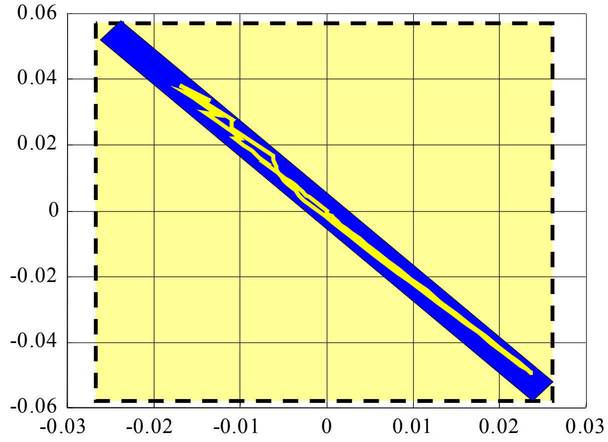
$u(t) = D(t)$



$X(t)$, X_k, $S(t)$, S_k, $V(t)$, V_k



(edX_k, edS_k)



- **Dependency is taken into account, but not the dynamic evol.**

$$[ed_k] = cd_k + \blacklozenge Rd_k \qquad cd_k = 0 \qquad Rd_k = \begin{bmatrix} 0.025 & 0.00125 & 0_{1\times 3} \\ -0.055 & 0.003 & 0_{1\times 3} \\ 0_{3\times 1} & 0_{3\times 1} & 10^{-5}.I_{3\times 3} \end{bmatrix}$$

Simulation results: context

- **Initial (extended) state domain:**

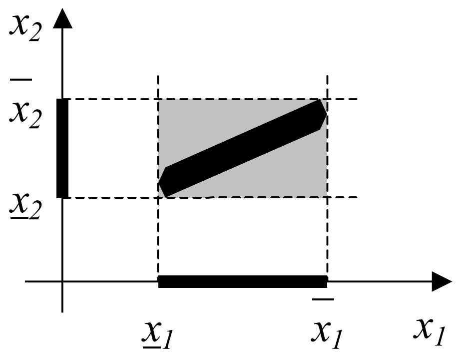
$$[x_0] = [\underbrace{[0;20]}_{X(0)}, \underbrace{[0;20]}_{S(0)}, \underbrace{[0.5;2]}_{V(0)}, \underbrace{[0.085;0.105]}_{M(0)}, \underbrace{[1.9;2.4]}_{K(0)}]^T$$

- **Measurement noise:**

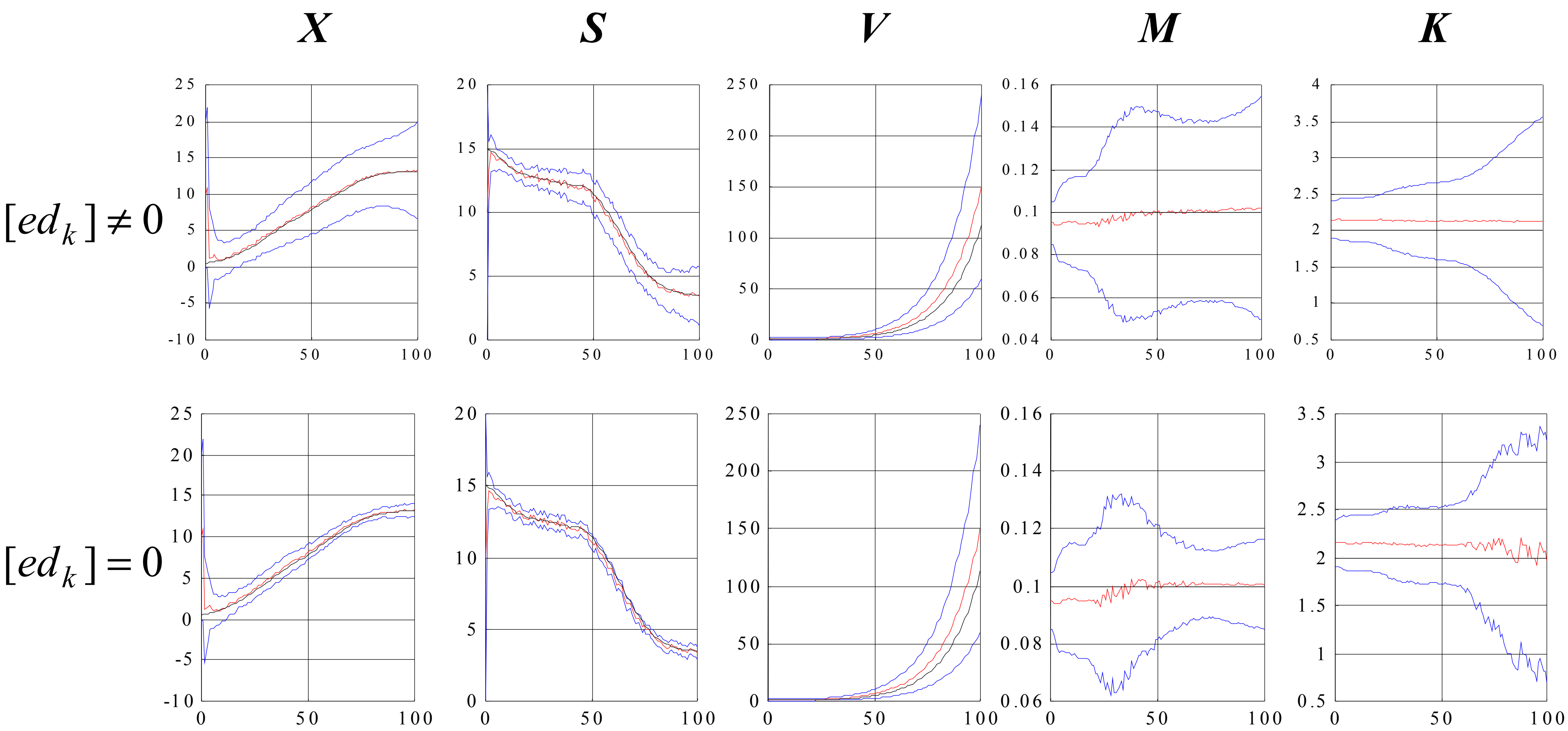
  w **is a uniform random noise (same bounds as the interval)**

- **Zonotope complexity:** $d = 25$

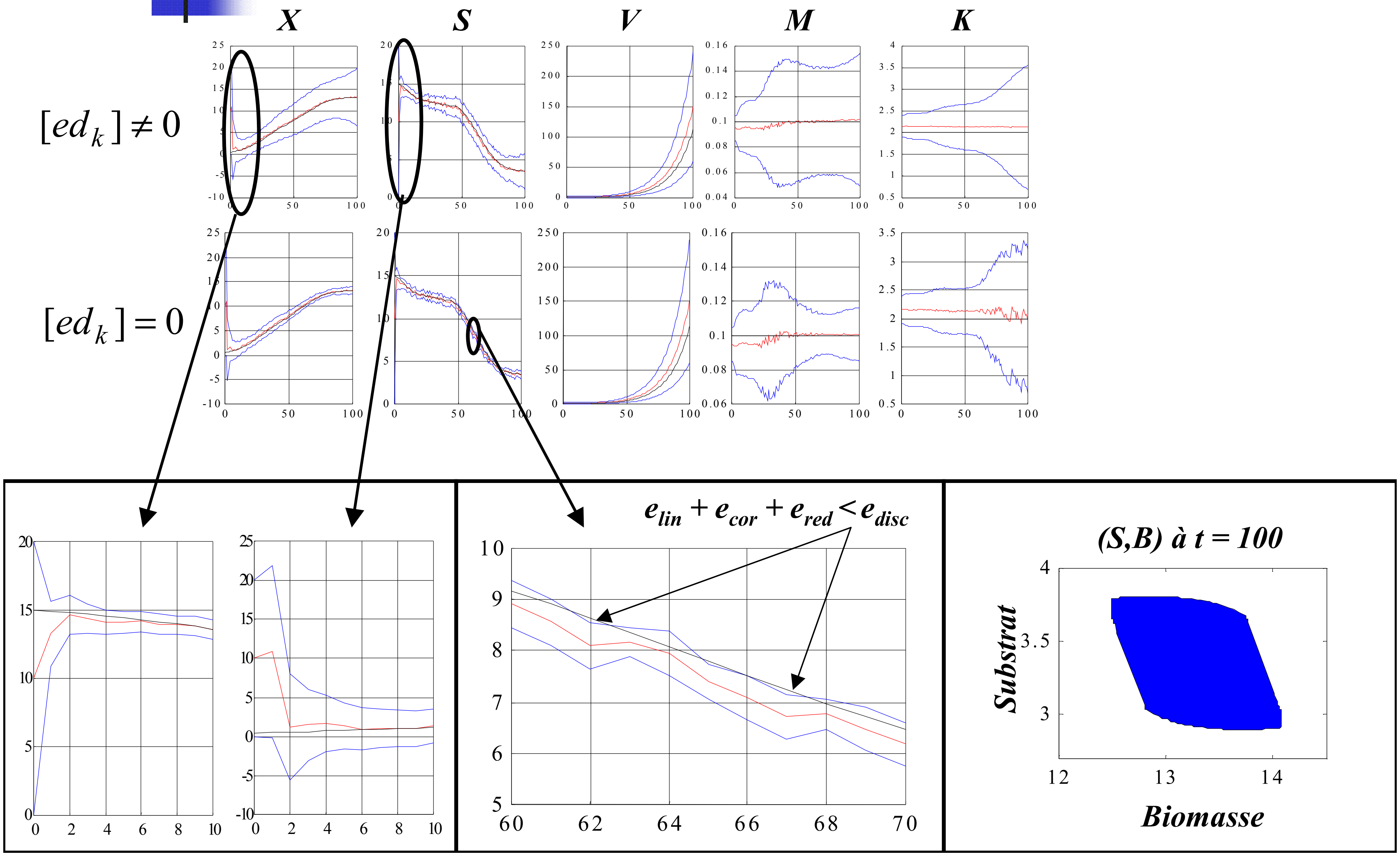
- **Computation of state bounds → Interval hull:**



Non linear observation and parameter estimation



Non linear observation and parameter estimation



Conclusion & Future prospects

Conclusion:

- Zonotopes → An alternative to other domain representations
- Wrapping effect → Parameterization of the domain complexity
- Observer handling dependency in uncertain non-linear systems with no iterative optimization, no bissection, no facets or vertices enum.
- From linear to non-linear sys. → Analogy with (extended) Kalman filter
- Need for a refined evaluation of uncertainties → Precision of estimation

Future prospects:

- Dynamic inclusion of the discretization error
- Formal calculus → Automatic procedures to design the observer
- Quantifying the pessimism
- Design and evaluation of other correction approaches