DIAGNOSIS AND FAULT-TOLERANT CONTROL USING SET-BASED METHODS

Vicenç Puig
Advanced Control Systems (SAC)
Research Group

Réunion GT S3 - February 4th, 2016
Index

1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research

Réunion GT S3 - February 4th, 2016
Model-based Fault Detection

- Model-based fault detection methods rely on the concept of analytical redundancy.

- However, modeling errors and disturbances in complex engineering systems are inevitable, and hence there is a need to develop robust fault detection algorithms.
Robustness in Model-based Fault Detection

- The **robustness** of a fault detection system means that it must be only sensitive to faults, even in the presence of model-reality differences.
Passive Robust Decision-Making using Interval Models

Threshold

Fault Indication

\[ y_k \notin \hat{Y}_k = [\hat{y}_k, \bar{y}_k] \]

Variable de estado X1

Réunion GT S3 - February 4th, 2016
1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Interval Model for FDI (1)

- Consider that the system to be monitored can be described by a general nonlinear model in discrete-time

\[
x(k + 1) = f(x(k), u(k), \theta)
\]
\[
y(k) = g(x(k), u(k), \theta)
\]

- The parameters \( \theta \in \mathbb{R}^m \) are assumed to be unknown but belong to known intervals

\[
\theta_i \in [\underline{\theta}_i, \bar{\theta}_i], \quad i = 1 \ldots m
\]

- An additional equation defining the allowed variance of parameters can be introduced for this purpose:

\[
\theta(k + 1) = \theta(k) + w(k)
\]

where \( |w(k)| \cdot \lambda \).
Interval Model for FDI (2)

- Measurement noise can be taken into account by assuming that the measurements are known to belong to intervals $[y(k)]$, often created by adding an noise term $\sigma$ to the actual measurement $y(k)$, that is,

$$ [y(k)] = [y(k) - \sigma, y(k) + \sigma] $$

- In case uncertain parameters appear linearly with respect to inputs/outputs, the system model will be expressed in regressor form

$$ y(k) = \varphi^T(k)\theta(k) + e(k) $$

- This corresponds to a MA parity equation.
Fault Detection using Direct Image Test

- Considering the uncertainty in parameters $\theta \in \Theta$, the **direct image test** is

$$y(k) \in \left[ \hat{y}(k), \bar{y}(k) \right]$$

Then, no fault is indicated. In other case, a fault is indicated.

- The interval for the estimated output can be determined by

$$\varphi^T(k)\hat{\theta}(k) + \sigma \leq y(k) \leq \varphi^T(k)\bar{\theta}(k) + \sigma$$

where:

$$\hat{\theta}(k) = \arg \min_{\theta \in \mathcal{V}} \varphi^T \theta$$

$$\bar{\theta}(k) = \arg \max_{\theta \in \mathcal{V}} \varphi^T \theta$$
Fault Detection Algorithm using Inverse Test

\[ \exists \theta \in \Theta \mid y(k) - \sigma \leq \varphi^T(k)\theta \leq y(k) - \sigma \]

\[ F_k = \{ \theta \in \mathbb{R}^n : -\sigma \leq y(k) - \varphi(k)^T \theta \leq \sigma \} \]

\[ F_k \cap \Theta_k = \emptyset \]
A zonotope can be thought of as a **Minkowski sum** of a finite set of line segments:

\[ X = p \oplus RB^m = \{ p + Rz : z \in B^m \} \]

A zonotope can also be seen as the linear image of an \( m \)-hypercube in an \( n \)-space.

Réunion GT S3 - February 4th, 2016
Zonotopes (2)

Zonotope Arithmetic

- Sum of two zonotopes:
  \[ X = p \oplus RB^m = (p_1 + p_2) \oplus [R_1 \quad R_2]B^m \]

- Image of a zonotope by a linear application \( L \):
  \[ X = (Lp) \oplus (LR)B^m \]

- Smallest interval box containing a zonotope ("interval hull"):
  \[ \square X = \{ x : |x_i - p_i| \leq \|R_i\|_1 \} \]

- Inverse image of a zonotope by a linear application
- Intersection of two zonotopes
Index

1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Let the model for the state estimator of the monitored system described by a *interval Luenberger observer* formulated as

\[
\hat{x}_{k+1} = A(\theta) \hat{x}_k + B(\theta) u_k + w_k + L(y_k - \hat{y}_k)
\]

\[
\hat{y}_k = C x_k + v_k
\]

This approach is in a half-way between *simulation* and *prediction* approaches.
Let us denote the following sequences from the first time instant to time $k$:

$$\tilde{u}_k = (u_j)^k_{j=0} = (u_0, u_1, \ldots, u_{k-1})$$
$$\tilde{y}_k = (y_j)^k_{j=0} = (y_0, y_1, \ldots, y_k)$$
$$\tilde{w}_k = (w_j)^k_{j=0} = (w_0, w_1, \ldots, w_{k-1})$$
$$\tilde{v}_k = (v_j)^k_{j=0} = (v_0, v_1, \ldots, v_{k-1})$$
$$\tilde{\theta}_k = (\theta_j)^k_{j=0} = (\theta_0, \theta_1, \ldots, \theta_{k-1})$$

The set of estimated states at time $k$ using the interval observer approach is expressed by

$$\hat{X}_k = \left\{ \hat{x}_k \text{ such that } \begin{array}{l}
\hat{x}_{k+1} = A(\theta_k)\hat{x}_k + Bu_k + w_k + L(y(k) - \hat{y}(k))_{j=1}^k \\
\hat{y}_k = Cx_k + v_k_{j=0}^k \\
(\hat{w}_k \in \mathcal{W}, \hat{v}_k \in \mathcal{V}, \theta_k \in \Theta)_{j=0}^k, x_0 \in \mathcal{X} \end{array} \right\}$$
The previous uncertain state set at time $k$ can be computed approximately by admitting the rupture of the existing relations between variables of consecutive time instants:

\[
x_{k+1} = A(\theta)x_k + B(\theta)u_k + w_k + L(y_k - \hat{y}_k)
\]

\[
y_k = Cx_k + v_k
\]

**Algorithm 1: Worst-case State Observer using Set Computations**

1. $\hat{X}_k \leftarrow X_0$
2. for $k = 1$ to $N$ do
3. Compute $\hat{X}_k$
4. Compute $\hat{Y}_k$
5. end for

Réunion GT S3 - February 4th, 2016
Problems of Interval Observers

- When approximating the region of system states using sets several problems should be considered:
  - The *wrapping effect*
  - The preservation of the *parameter time-invariance*
  - The *under/over estimation* of the region

- These problems produce the *propagation of the uncertainty*, deriving in the production of inconsistent, and even, unstable simulations/observations.
The problem of wrapping is related to the use of a crude approximation of the real region of state variables.

At every stage of the simulation/observation, the true region of uncertain states is wrapped into a superset feasible to construct and to represent on a computer.

Because of the overestimation of the a wrapped set is proportional to its radius, a spurious growth of the enclosures can result if the composition of wrapping and mapping is iterated.
Designing the Observer Gain to Avoid the Wrapping Effect

- Given a **non-isotonic interval system**, an interval observer could be designed to fulfil the condition of isotonicity if all the elements of the observer matrix $A_0$ satisfy: $a_{ii}^0 \geq 0$.

- In case of an isotonic observer is designed through appropriate selection of the observer gain, the wrapping effect is not present.

- Consequently, a simple iterative scheme based on a region propagation will work, providing the same results than a trajectory propagation algorithm.

- Moreover, a set-based (**time-varying**) interval observation and a trajectory based (**time-invariant**) interval observation will provide the same interval observation.
Fault Detection using Interval Observers (1)

- **Fault detection test:**
  Given the sequences of measured inputs $\tilde{u}_k$ and outputs $\tilde{y}_k$ of the actual system, a fault is said to have occurred at time $k$ if

  \[ y_k \notin \hat{Y}_k = \left[ \underline{\hat{y}}_k, \bar{\hat{y}}_k \right] \quad \text{or alternatively,} \quad 0 \notin \left[ r_k, r_k \right] = y(k) - \left[ \underline{\hat{y}}_k, \bar{\hat{y}}_k \right] \]

- In case noise in measurements is considered $y_k \in \mathcal{Y}_k = \left[ \underline{y}_k, \bar{y}_k \right]$, a fault is detected at time $k$ if

  \[ \mathcal{Y}_k \cap \hat{Y}_k = \emptyset \]

- **Fault detection** consists in detecting a fault using the previous test given a sequence of measured inputs $\tilde{u}_k$ and outputs $\tilde{y}_k$. 

Réunion GT S3 - February 4th, 2016
Algorithm 2: Fault Detection using Worst-case Observer

1: \( \hat{X}_k \leftarrow X_0 \)
2: for \( k = 1 \) to \( N \) do
3: Compute \( \hat{X}_k \)
4: Compute \( \hat{Y}_k \)
5: if \( \hat{Y}_k \cap Y_k = \emptyset \) then
6: Exit (Fault detected)
7: end if
8: end for

\[
x_{k+1} = A(\theta)x_k + B(\theta)u_k + w_k + K(y_k - \hat{y}_k) \quad y_k = Cx_k + v_k
\]
Invariant Sets and Interval Observers

Interval observer-based FD principle

Invariant set-based FD principle

Réunion GT S3 - February 4th, 2016
Advantages and Disadvantages

Invariant Sets
- Behaviors at steady state
  - Lower fault sensitivity (construct sets off-line)
  - Lower complexity

Interval Observers
- System behaviors at transient and steady state
  - Higher fault sensitivity (estimate sets on-line)
  - Higher complexity
Theoretical FDI Conditions

Theoretical FDI conditions:

\[ 0 \in \tilde{R}_{ij}^{ii} \text{ and } 0 \notin \tilde{R}_{ij}^{ij} \text{ for all } j \neq i \]
Index

1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Set-membership (or Consistency)-based Estimation Principle

- Let us denote the following sequences from the first time instant to time $k$:

\[
\tilde{u}_k = (u_j)_{0}^{k-1} = (u_0, u_1, \ldots, u_{k-1})
\]

\[
\tilde{y}_k = (y_j)_{0}^{k-1} = (y_0, y_1, \ldots, y_{k})
\]

\[
\tilde{w}_k = (w_j)_{0}^{k-1} = (w_0, w_1, \ldots, w_{k-1})
\]

\[
\tilde{v}_k = (v_j)_{0}^{k-1} = (v_0, v_1, \ldots, v_{k-1})
\]

\[
\tilde{\theta}_k = (\theta_j)_{0}^{k-1} = (\theta_0, \theta_1, \ldots, \theta_{k-1})
\]

- The set of estimated states at time $k$ using the set-membership approach is expressed by

\[
X_k = \left\{ x_k \mid \exists \tilde{w}, \tilde{v}, \tilde{\theta}, x_o \text{ such that } \begin{align*}
    x_{k+1} &= A(\theta_k)x_k + B(\theta_k)u_k + w_k \quad j=1 \\
    y_k &= Cx_k + v_k \quad j=0
\end{align*} \right\}
\]
The previous uncertain state set at time $k$ can be computed approximately by admitting the rupture of the existing relations between variables of consecutive time instants.

Two sets are introduced:

- The **set of predicted states** at time $k$ is given by
  \[
  \mathbb{X}_k^p = \{ x_k : A(\theta_{k-1})x_{k-1} + B(\theta_{k-1})u_{k-1} + E y_k + w_{k-1} | x_{k-1} \in \mathbb{X}_{k-1}, \theta_k \in \Theta, w_{k-1} \in \mathbb{W}_{k-1} \}\]

- The **set of consistent states** at time $k$ with measurement is defined as
  \[
  \mathbb{X}_k^{y_k} = \{ x_k : y_k = C x_k + v_k, \theta_k \in \Theta, v_k \in \mathbb{V}_k \}\]
Implementation of Set-membership Estimators (2)

- This allows to write the following algorithm:

```plaintext
Algorithm 1: Set-membership State Estimation using Set Computations

1: \( \mathcal{X}_k^e \leftarrow \mathcal{X}_0 \)
2: \textbf{for } k = 1 \textbf{ to } N \textbf{ do}
3: \hspace{1em} Compute \( \mathcal{X}_k^p \)
4: \hspace{1em} Compute \( \mathcal{X}_k^c \)
5: \hspace{1em} Compute \( \mathcal{X}_k^e = \mathcal{X}_k^p \cap \mathcal{X}_k^c \)
6: \textbf{end for}
```

\[ x_{k+1} = A(\theta)x_k + B(\theta)u_k + w_k \]

Réunion GT S3 - February 4th, 2016
Fault Detection using Set-membership Estimation (1)

- **Fault detection test:**

Given the sequences of measured inputs $\tilde{u}_k$ and outputs $\tilde{y}_k$ of the actual system, a **fault** is said to have occurred at time $k$ if there does not exist a set of sequences $(\tilde{w}_k, \tilde{v}_k, \tilde{\theta}_k)$ which satisfy the nominal system description with initial condition, noise, disturbances and parameters belonging to $(\chi_0, \nu, \omega, \Theta)$, respectively.

- **Fault detection** consists in detecting a fault given a sequence of measured inputs $\tilde{u}_k$ and outputs $\tilde{y}_k$. 
Fault Detection using Set-membership Estimation (2)

Algorithm 2: Fault Detection using Set-membership Estimation

1: \( \mathcal{X}^e_k \leftarrow \mathcal{X}_0 \)
2: for \( k = 1 \) to \( N \) do
3: \( \) Compute \( \mathcal{X}^p_k \)
4: \( \) Compute \( \mathcal{X}^c_k \)
5: \( \) Compute \( \mathcal{X}^e_k = \mathcal{X}^p_k \cap \mathcal{X}^c_k \)
6: \( \) if \( \mathcal{X}^e_k = \emptyset \) then
7: \( \) Exit (Fault detected)
8: \( \) end if
9: \( \) end for

\[
x_{k+1} = A(\theta)x_k + B(\theta)u_k + w_k
\]

\[
\mathcal{X}^e_{k-1} \rightarrow \text{Direct Image} \rightarrow \mathcal{X}^p_k \leftarrow \text{Direct Image} \leftarrow \mathcal{X}^c_k \leftarrow \text{Intersection} \xrightarrow{\text{Inverse Image}} \mathcal{X}^e_k
\]

\[
y_k = Cx_k + v_k
\]

\[
\mathcal{X}^e_k = \mathcal{X}^p_k \cap \mathcal{X}^y_k = \emptyset
\]

Réunion GT S3 - February 4th, 2016
Index

1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Identification for Robust Fault Detection

- One of the key points in model based fault detection is how detection models are estimated.

- In case of set-membership methods, the set for uncertain parameters should be estimated.

- The set for uncertain parameters depend on the way how the uncertain model will be used for fault detection.

- At least two possible types of models can be derived:
  - interval model
  - set-membership or consistency based model
Identification for the Direct Test (1)

Given a set of measurements \( y(k) \) taken in a given interval \( k \in [0, N] \), considering that noise is bounded such that \( y_m(k) \in Y_m(k) \), then a set of model parameters that produces an envelope that cover all measurements (“worst-case approach”):

\[
\Theta = \left\{ \theta \in \Theta \mid \forall y(k) \in Y(k), \forall k \in [0, N] \quad (y(k, \theta) \leq y(k)) \land (y(k) \leq \overline{y}(k, \theta)) \right\}
\]

where at each time \( t \) instant \( k \), model temporal envelope is computed according to:

\[
\underline{y}(t_k) = \min y(t_k, \theta) \quad \overline{y}(t_k) = \max(t_k, \theta)
\]

\( \text{sujeto a :} \quad \theta \in \Theta \)

Réunion GT S3 - February 4th, 2016
Identification for the Direct Test (2)
Identification for the Inverse Test (1)

Given a set of measurements \( y(k) \) taken in a given interval \( k \in [0, N] \), considering that noise is bounded such that \( y_m(t) \in Y_m(t) \), then a set of model parameters that are consistent with model and measurements would be estimated such that ("consistency approach"):

\[
\Theta = \left\{ \theta \in \Theta \mid \exists y(k) \in Y(k), \forall k \in [0, N] \quad y(k) \leq y(k, \theta) \leq \bar{y}(k) \right\}
\]

This set can be computed at each sample time instant \( k \):

\[
\Theta = \left\{ \theta \in \Theta \mid \exists y(k) \in Y(k) \quad y(k) \leq y(k, \theta) \leq \bar{y}(k) \right\}
\]

such that:

\[
\Theta = \bigcap_{k=1}^{N} \Theta_k
\]
Identification for the Inverse Test (2)
1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Fault Tolerant Control

Level 3
Supervision

Level 2
Fault Diagnosis
& Accommodation

Level 1
Control

Réunion GT S3 - February 4th, 2016
Fault tolerant MPC problem

- The solution of a control problem consists on finding a control law in a given set of control laws $\mathcal{U}$ such that the controlled system achieves the control objectives $O$ while its behavior satisfies a set of constraints $C$.
- The solution of the problem is completely defined by the triple: $\langle \mathcal{U}, O, C \rangle$
- In the case of a linear constrained predictive control law:

$$O : \min_{\tilde{u}} J(\tilde{x}, \tilde{u})$$

subject to:

$$C : \begin{cases} x_{k+1} = f(x_k, u_k) \\ u_k \in \mathcal{U} & k = 1, \ldots, N - 1 \\ x_k \in \mathcal{X} & k = 0, \ldots, N \end{cases}$$

where:

$$\mathcal{U} = \left\{ u_k \in \mathbb{R}^m \middle| u_{\text{min}} \leq u_k \leq u_{\text{min}} \right\}$$
$$\mathcal{X} = \left\{ x_k \in \mathbb{R}^n \middle| x_{\text{min}} \leq x_k \leq x_{\text{min}} \right\}$$

$$\tilde{u}_k = (u_j)_0^{k-1} = (u_0, u_1, \ldots, u_{k-1})$$
$$\tilde{x}_k = (x_j)_0^{k-1} = (x_0, x_1, \ldots, x_k)$$
Hybrid MPC Fault-tolerant Control
Preliminary Definitions

- **Definition 1.** The **feasible solution set** is given by
  \[ \Omega = \{ \bar{x}, \bar{u} \mid \left( x_{k+1} = f(x_k, u_k) \right)_{0}^{N-1} \} \]
  and gives the input and state sets compatible with system constraints which originate the set of predictive states.

- **Definition 2.** The **feasible control objectives set** is given by
  \[ J_{\Omega} = \{ J(\bar{x}, \bar{u}) \mid (\bar{x}, \bar{u}) \in \Omega \} \]
  and corresponds to the set of all values of $J$ obtained from feasible solutions.

- **Definition 3.** The **admissible solution set** is given by
  \[ A = \{ (\bar{x}, \bar{u}) \in \Omega_f \mid J(\bar{x}, \bar{u}) \in J_A \} \]
  where $\Omega_f$ corresponds to the feasible solution set of a actuator fault configuration and $J_A$ defined as the admissible control objective set.
Admissibility Evaluation using Set Computations (1)

- The admissibility evaluation using a set computation approach starts obtaining the *feasible solution set* given a set of initial states, the system dynamic and the system operating constraints over $N$.

**Algorithm 1** Computation of $\Omega$

1. $\mathcal{X}_k \leftarrow \mathcal{X}_0$
2. $\Omega_0 \leftarrow \mathcal{X}_0$
3. **for** $k = 1$ to $N$ **do**
4.   $\mathcal{U}_{k-1} \leftarrow \mathcal{U}$
5.   **Compute** $\mathcal{X}^p_k$ from $\mathcal{X}_{k-1}$ and $\mathcal{U}_{k-1}$
6.   **Compute** $\mathcal{X}_k^e = \mathcal{X} \cap \mathcal{X}^p_k$
7.   **Compute** $\mathcal{U}^e_{k-1}$ from $\mathcal{X}^e_k$
8.   $\Omega_k = \mathcal{X}_k^e \times \mathcal{U}^e_{k-1}$
9.   $\mathcal{X}_k \leftarrow \mathcal{X}_k^e$
10. **end for**
11. $\Omega = \bigcup_{k=0}^{N} \Omega_k$

\[ x_{k+1} = f(x_k, u_k) \]

**Intersection**

**Direct Image**

**Inverse Image**
Admissibility Evaluation using Set Computations (2)

- At the same time that the **feasible solution set** is computed $\Omega$, the **feasible control objectives set** $J_\Omega$ at time $k=N$ can be obtained using the following algorithm:

```
Algorithm 2 Computation of $J_\Omega$ using $\Omega_k$
1: $X_k := X_0$
2: $\Omega_0 := X_0$
3: for $k = 1$ to $N$ do
4:   Compute $\Omega_k$ (See Algorithm 1)
5:   Compute $J_{\Omega_k}$ using $\Omega_k = X_k \times U_{k-1}$
6: end for
7: $J_\Omega = \bigcup_{k=0}^N J_{\Omega_k}$
```
Admissibility Evaluation using Constraints Satisfactions (1)

- **Constraints satisfaction problem:**
  "A **constraints satisfaction problem** (CSP) on sets can be formulated as a 3-tuple $H = (V,D,C)$ where:

  - $V = \{v_1, \cdots ,v_n\}$ is a finite set of variables,
  - $D = \{D_1, \cdots ,D_n\}$ is the set of their domains represented by closed sets
  - $C = \{c_1, \cdots ,c_n\}$ is a finite set of constraints relating variables of $V$

- A point solution of $H$ is a $n$-tuple $(v_1,\cdots,v_n) \in D$ such that all constraints $C$ are satisfied.
- The set of all point solutions of $H$ is denoted by $S(H)$. This set is called the global solution set.
- The variable $v_i \in V_i$ is consistent in $H$ if and only if:

$$\forall v_i \in V_i \ \exists (\tilde{v}_1 \in D_1, \cdots ,\tilde{v}_n \in D_n) \ | (\tilde{v}_1, \cdots ,\tilde{v}_n) \in S(H)$$

with $i=1\ldots n$
The admissibility evaluation requires the computation of the admissible solution set:

\[ \Omega = \left\{ (\tilde{x}, \tilde{u}) \left| (x_{k+1} = f(x_k, u_k))^0_{N-1} \right. \right\} \]

Its definition suggests a way of implementation since its mathematical description can be viewed as a constraints satisfaction problem:

**Algorithm 1: Admissibility Evaluation using Constraints Satisfaction**

At each time instant \( k \) over \( N \), the feasible solution set is determined by solving the CSP \( \mathcal{H} = (\mathcal{V}, \mathcal{D}, \mathcal{C}) \) associated with the constraints \( \mathcal{C} \) of the CNMPC problem, where

\[
\mathcal{V} = \{ x_1, x_2, \ldots, x_N, u_1, u_2, \ldots, u_{N-1}, J \}
\]

\[
\mathcal{D} = \{ \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N, \mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_{N-1}, \mathcal{J}_A \}
\]

\[
\mathcal{C} = \left\{ (x_{k+1} = f(x_k, u_k))^0_{N-1}, J(\tilde{x}, \tilde{u}) = \phi(x_N) + \sum_{i=0}^{N-1} \Phi(x_i, u_i) \right\}
\]
Index

1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
The Barcelona Sewer Network

- **Data**
  - Typology: combined
  - Length: 1.650 km
  - Storage capacity: 2.634.124 m³
  - Visitable portion: 55,12%
  - Mean transversal section: 1,8 m²
  - 31 catchment area: 12.326 ha

- **Particularities**
  - Topographic profile: steep slope, gentle at rivers and sea
  - Urban ground: 90% impervious
  - Meteorology: yearly precipitation: 600mm, intensity: up to 150 mm/h in 15 minutes
Barcelona and its Rain

Réunion GT S3 - February 4th, 2016
Solution (1): Detention Tanks
Solution (2): Barcelona’s RTC System

<table>
<thead>
<tr>
<th>ELEMENTS</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain gauges</td>
<td>22</td>
</tr>
<tr>
<td>Water level sensors</td>
<td>119</td>
</tr>
<tr>
<td>Pumping Stations</td>
<td>11</td>
</tr>
<tr>
<td>Gates</td>
<td>23</td>
</tr>
<tr>
<td>Detention Tanks</td>
<td>10</td>
</tr>
</tbody>
</table>

Réunion GT S3 - February 4th, 2016
- **Reduction of the risk of floods**

\[ J_{\text{flood}} = \sum_j \max(o, q_j - \text{lim}_{q_j}) \]

- **Environment protection**

\[ J_{\text{cso}} = \sum_l \text{CSO}_l \]

- **Optimization of the WWTP**

\[ J_{\text{WWTP}} = \sum_i (\text{WWTP}_i - \text{WWTP}_i^*) \]

\[ J = \sum_{k=0}^{N-1} (\alpha J_{\text{flood}}^k + \beta J_{\text{cso}}^k + \gamma J_{\text{WWTP}}^k) \]

\[ c_{\text{so}} \]

\[ \text{CSO}_l^k \] combined sewer overflow volume at site \( l \)

\[ \text{WWTP}_i \] waste water treatment plant flow \( i \)

\( q_j \) flow through sewer \( j \)
Global Control vs Local control

WWTP Volume

CSO Volume

50 % improvement

-18 % reduction

Blue: Local Control
Red: Global Control

Réunion GT S3 - February 4th, 2016
CORAL: MPC tool for Sewer Networks
CORAL Architecture

Réunion GT S3 - February 4th, 2016
Introduction to FDI in Sewer Networks

- In this presentation, the FDI problem of rain gauges and limnimeters of Barcelona’s urban sewer system is addressed.

- Rain gauges and limnimeters are used for the real-time global control of the whole Barcelona network.

- Often these instruments are out of order in rain scenarios when the control system must be fully operative.

- In order to detect and isolate faulty instruments and to reconstruct faulty measurements from data fusion, a fault diagnosis system is necessary.
The Architecture of the FDI System

Réunion GT S3 - February 4th, 2016
Fault Isolation Procedure

**Signature Vector**

\[ s(k) \]

\[ S_i(k) \]

**Hamming Distance**: number of different bits between two binary codes

\[ \text{distHam} = c_1 \oplus c_2 \]

\[ d_i(k) = \text{distHam}(s(k), S_i(k)) \]

\[ d(k) \]

\[ f(k) = \arg \min(d(k)) \]

\[ r_1(k) = y_1(k) - a_{13}y_3(k) - a_{18}y_8(k) \]

\[ r_2(k) = y_2(k) - a_{21}y_1(k) - a_{24}y_4(k) \]

\[ r_3(k) = y_3(k) - a_{31}y_1(k) - a_{37}y_7(k) \]

\[ r_4(k) = y_4(k) - a_{45}y_5(k) - a_{48}y_8(k) \]

**Usual assumptions**

- Single fault and exoneration
- If multiple faults, no compensation

Réunion GT S3 - February 4th, 2016
Enhanced Fault Isolation Scheme

In particular, such interface can be improved taking into account the following information:

- **residual value size**: big violation of the threshold or only a small fault signal activation.
- **residual sensitivity** with respect to a certain fault.
- **time pattern** of fault signal occurrence.
- **order** of fault signal occurrence.

Réunion GT S3 - February 4th, 2016
Interface between Fault Detection and Isolation Modules

- The interface is based on a memory implemented as a table in which events in the residual history are stored:

\[
\phi_i(k) = \begin{cases} 
\frac{(r_i^o(k)/\bar{r}_i^o(k))^4}{1+(r_i^o(k)/\bar{r}_i^o(k))^4} & \text{if } r_i^o(k) \geq 0 \\
-\frac{(r_i^o(k)/\bar{r}_i^o(k))^4}{1+(r_i^o(k)/\bar{r}_i^o(k))^4} & \text{if } r_i^o(k) < 0 
\end{cases}
\]

- For each row, the first column stores the occurrence time \( t_i \), the second one stores, the \( \phi_{i,max} \), and the third one stores the sign of the residual.

- If the fault detection component detects a new fault signal, it updates the memory by filling out the three fields.
Fault Detection and Isolation Interface: *FSM* Matrices

- It is based on the concept of the theoretical *fault signature matrix* (*FSM*) which was introduced by (Gertler, 1998).

- This matrix stores the theoretical binary influence of a given fault $f_j$ (column of *FSM*) on a given residual $r_i(k)$ or equivalently, on a given fault signal $\phi_i(k)$ (row of *FSM*).

- Here, the fault signature matrix concept is generalized since the binary interface is extended taking into account more fault signal properties.

<table>
<thead>
<tr>
<th>Fault Signal Properties</th>
<th>FSM Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>FSM01</td>
</tr>
<tr>
<td>Sign</td>
<td>FSM_sign</td>
</tr>
<tr>
<td>Fault residual sensitivity</td>
<td>FSM_sensit</td>
</tr>
<tr>
<td>Occurrence order</td>
<td>FSM_order</td>
</tr>
<tr>
<td>Occurrence time instant</td>
<td>FSM_time</td>
</tr>
</tbody>
</table>
Limnimeter Modelling (1): “Virtual Reservoir Approach”

- Propagation of flows through sewer pipes can be described by numerical solution of the continuity and momentum Saint-Venant's partial differential equations.

\[
\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = 0
\]

\[
\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \frac{Q^2}{A} \right) + gA \frac{\partial y}{\partial x} - gA(I_0 - I_f) = 0
\]

- Saint-Venant's equations can be used to perform simulation studies but are highly complex to solve in real-time, specially for large scale systems.
Limnimeter Modelling (2):
“Virtual Reservoir Approach”

- The sewerage network is modeled through a simplified graph relating the main sewers and set of virtual and real reservoirs.

- A virtual reservoir is an aggregation of a catchment of the sewage network which approximates the hydraulics of rain, runoff and sewage water retention thereof.

- The hydraulics of virtual reservoirs are:

\[
\frac{dV(t)}{dt} = Q_{in}(t) + I(t)S - Q_{out}(t)
\]

Using Manning’s formula and discretising:

\[
Q_{up}(t) = M_{up}L_{up}(t)
\]

\[
Q_{down}(t) = M_{down}L_{down}(t)
\]

\[
L_{down}(k + 1) = aL_{down}(k) + bL_{up}(k) + cI(k)
\]

Réunion GT S3 - February 4th, 2016
Application Example (1): Modeling Barcelona Sewer Network using Virtual Tanks
Application Example (2): Modeling Barcelona Sewer Network using Virtual Tanks
Application Example: Structure of the Limnimeter Models

- Applying the limnimeter modelling methodology based on “virtual tanks” to the considered sewer network:
  - 12 limnimeters are modelled allowing to compute 12 residuals.
  - Faults affecting 14 limnimeters can be diagnosed.
Application Example:
Fault Scenario affecting $L_7$

- A fault affecting limnimeter $L_7$ occurs at $t_0 = 4000s$. 

*Residual time evolution*
Fault Tolerant Control
Application Example (1)

- Consider the system corresponding to a piece of Barcelona sewer network described by the discrete-time state equations

\[
x_{k+1} = Ax_k + Bu_k + B_p d_k
\]

where:

\[
A = \begin{bmatrix}
1 - \Delta t \beta_1 & 0 & 0 \\
0 & 1 & 0 \\
\Delta t \beta_1 & 0 & 1 - \Delta t \beta_3
\end{bmatrix}
\]

\[
B = \Delta t \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & -1 \\
-1 & -1 & 1
\end{bmatrix}
\]

\[
B_p = \Delta t \begin{bmatrix}
0 & \alpha_2 & 0 \\
0 & 0 & 0 \\
1 & 0 & \alpha_3
\end{bmatrix}
\]
Application Example (2)

- The systems constraints are:

  - **Bounding constraints**: refers to physical restrictions.
    \[
    x_{1,k} \in [0, \infty] \quad u_{1,k} \in [0, 11] \
    x_{2,k} \in [0, 35000] \quad u_{2,k} \in [0, 25] \
    x_{3,k} \in [0, \infty] \quad u_{3,k} \in [0, 7]
    \]

  - **Mass conservation constraints**:
    \[
    d_{1,k} = u_{1,k} + Q_1(k) \\
    q_{x_{1,k}} = u_{2,k} + Q_2(k) \\
    q_{x_{2,k}} \geq u_{3,k}
    \]
Reconfiguration Case

- This case considers actuators completely closed or completely open due to the fault, what would change the admissibility of the obtained actuator fault configurations.

![Graph showing minimum volume over time]

**TABLE I**

<table>
<thead>
<tr>
<th>Fault Location</th>
<th>Min. Volume $[m^3]$</th>
<th>Admissibility Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>1050</td>
<td>--</td>
</tr>
<tr>
<td>Fault in $u_1$</td>
<td>8800</td>
<td>No Admissible</td>
</tr>
<tr>
<td>Fault in $u_2$</td>
<td>52200</td>
<td>No Admissible</td>
</tr>
<tr>
<td>Fault in $u_3$</td>
<td>1050</td>
<td>Admissible</td>
</tr>
</tbody>
</table>

**ADMISSIBILITY CRITERIA:**

$$V_{sea}^f \geq 8V_{sea}^o$$
Accommodation Case

- This case considers that faults produces the reduction of the actuators operating range (for example from 0-100\% to 0-50\%).

**TABLE II**

<table>
<thead>
<tr>
<th>Fault Location</th>
<th>Operation range</th>
<th>Min. Volume $[m^3]$</th>
<th>Admissibility Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fault</td>
<td>---</td>
<td>1050</td>
<td>---</td>
</tr>
<tr>
<td>Fault in $U_1$</td>
<td>0-20%</td>
<td>5200</td>
<td>Admissible</td>
</tr>
<tr>
<td>Fault in $U_1$</td>
<td>0-50%</td>
<td>2300</td>
<td>Admissible</td>
</tr>
<tr>
<td>Fault in $U_2$</td>
<td>0-20%</td>
<td>34000</td>
<td>No Admissible</td>
</tr>
<tr>
<td>Fault in $U_2$</td>
<td>0-50%</td>
<td>15700</td>
<td>No Admissible</td>
</tr>
</tbody>
</table>

**Admissibility Criteria:**

$$V^f_{sea} \geq 8V^o_{sea}$$

Réunion GT S3 - February 4th, 2016
1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Interval Observer Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Conclusions (1)

- This presentation has reviewed the use of set-membership methods in robust fault detection and isolation (FDI) and tolerant control (FTC).

- Alternatively to the statistical methods, set-membership methods use a deterministic unknown-but-bounded description of noise and parametric uncertainty (interval models).

- Using approximating sets to approximate the set of possible behaviours (in parameter or state space), these methods allows to check the consistency between observed and predicted behaviour.

- When an inconsistency is detected a fault can be indicated, otherwise nothing can be stated.
Conclusions (2)

- The same principle has been used to estimate interval models for fault detection and to develop methods for fault tolerance evaluation.

- Finally, same real application of these methods has been used to exemplify the successful uses in FDI/FTC.
Index

1. Introduction
2. Interval Models for Fault Detection
3. Fault Detection using the Worst-case Approach
4. Fault Detection using the Set-membership Approach
5. Identification for Robust Fault Detection
6. Fault-tolerance Evaluation
7. Real Applications
8. Conclusions
9. Further Research
Further Research

- As further research, the set-membership approach could be extended to:
  - extension to non-linear systems via the use of LPV models.
  - deal with the fault isolation and estimation tasks exploiting the set arithmetic concepts
  - adaptive thresholding in the frequency domain
  - better understand the links between the set-membership and interval approach revised in this presentation
  - further extend their application to fault tolerant control as means to specify admissible closed loop behaviours.
Thank you very much for your attention!!!