# Distributed Fault Detection and Isolation with Imprecise Network Models

Iman Shames,André H. Teixeira, Henrik Sandberg, Karl H. Johansson

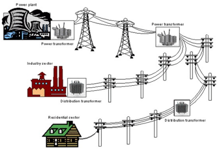
KTH, Stockholm, Sweden

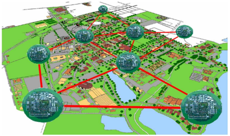
29 June, 2012, American Control Conference

# Networked Systems

Examples of Networked Systems:

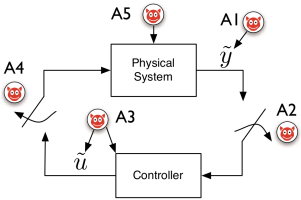
- Power Generation and Distribution Networks.
- Water Networks.
- Sensor Networks.
- Networked Industrial Processes

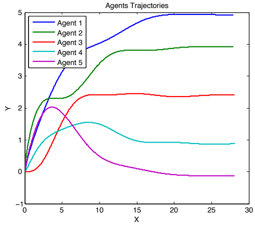


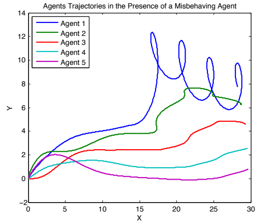



# Motivation: Distributed Systems Strengths and Vulnerabilities

- Solved many problems:
  - Modularity.
  - Distributed computation.
  - Easier monitoring.
  - No single point of failure.
  - ...




- Raised concerns:
  - Spatial distribution.
  - More components to fail.
  - More entry points for malicious agents.

# Today's Outline

# Model-based Fault Detection and Isolation

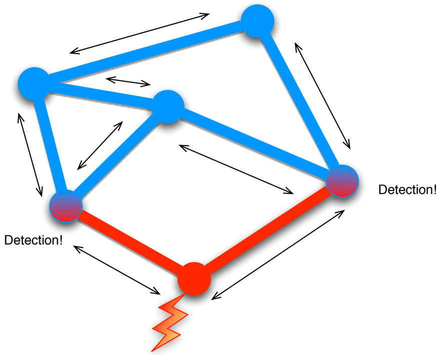
## Problem Description

### Problem

*How to detect and isolate the fault distributedly where the exact model of the network is not known?*

### Solution

- *Each node $i$ detect a fault in $j \in \mathcal{N}_i$ via (1) local measurements, and (2) knowing the network. The result is known.*
- *Applications to two sets of problems are highlighted.*



# Outline

# Model-based Fault Detection and Isolation

## Problem Description

- There are $N$ nodes and each Node $i$:

$$\dot{\mathbf{x}}_i(t) = \mathbf{R}_i(\mathbf{x}_i(t), \mathbf{x}_{i_1}(t), \ldots, \mathbf{x}_{i_{|\mathcal{N}_i|}}(t))$$
$$= \mathbf{R}_i(\mathbf{y}_i(t))$$

  - $i_j \in \mathcal{N}_i$
  - $\mathbf{y}_i(t) = [\mathbf{x}_i(t), \mathbf{x}_{i_1}(t), \ldots, \mathbf{x}_{i_{|\mathcal{N}_i|}}(t)]$: measurements available to $i$

- A faulty node $j$:

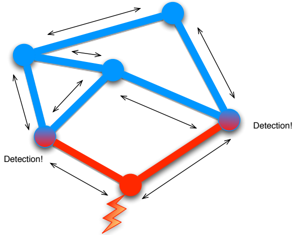
$$\dot{\mathbf{x}}_j(t) = \mathbf{R}_j(\mathbf{x}_j(t), \mathbf{x}_{i_1}(t), \ldots, \mathbf{x}_{i_{|\mathcal{N}_j|}}(t)) + \mathbf{f}_j(t)$$
$$= \mathbf{R}_j(\mathbf{y}_j(t)) + \mathbf{f}_j(t)$$

  - Unknown fault signal: $\mathbf{f}_j(t)$

- Network:

$$\dot{\mathbf{x}}(t) = \mathbf{R}(\mathbf{x}(t)) + \mathbf{b}\mathbf{f}_j(t)$$

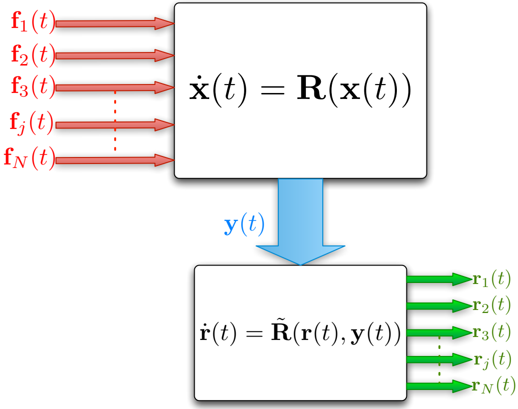
  - $\mathbf{x}(t) = [\mathbf{x}_1^\top(t), \ldots, \mathbf{x}_N^\top(t)]^\top$.
  - $\mathbf{R}(\cdot) = [\mathbf{R}_1^\top(\cdot), \ldots, \mathbf{R}_N^\top(\cdot)]^\top$.
  - $\mathbf{b}$: a vector of all zero except for the entries corresponding to $j$.



# Outline

# Solution Idea

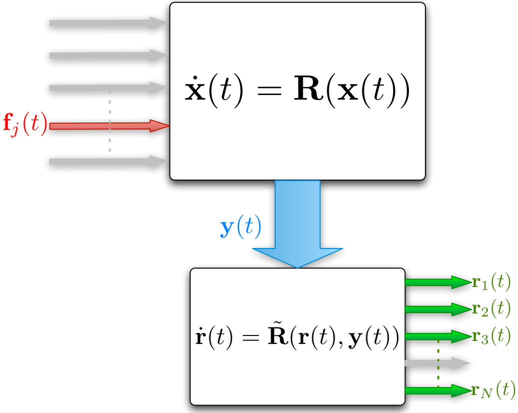
## Centralized Solution




- Find a residual generator, $\tilde{\mathbf{R}}(\cdot)$ s.t. $\|\mathbf{r}_j(t)\| \leq \epsilon$ iff $\mathbf{f}_k(t) = 0$ for all $k \neq j$.
- The value $\mathbf{r}_j(t)$ is called a residual.

# Solution Idea

## Centralized Solution




- Find a residual generator, $\tilde{\mathbf{R}}(\cdot)$ s.t. $\|\mathbf{r}_j(t)\| \leq \epsilon$ iff $\mathbf{f}_k(t) = 0$ for all $k \neq j$.
- The value $\mathbf{r}_j(t)$ is called a residual.

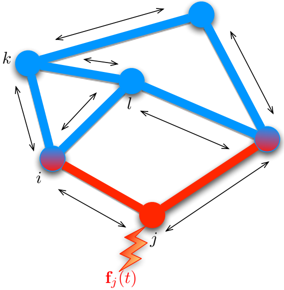
# Solution Idea

## Distributed Solution

### Distributed FDI

It is desired:

1. Node $i$ only measures $\mathbf{y}_i(t) = [\mathbf{x}_i(t), \mathbf{x}_j(t), \mathbf{x}_k(t), \mathbf{x}_l(t)]$.
2. Node $i$ generates $\mathbf{r}^i_j(t)$, $\mathbf{r}^i_k(t)$, and $\mathbf{r}^i_l(t)$ s.t. $\mathbf{r}^i_j(t) \leq \epsilon$, $\mathbf{r}^i_k(t), \mathbf{r}^i_l(t) > \epsilon$ iff $j$ is faulty.




It is possible to generate such residuals for two important applications.

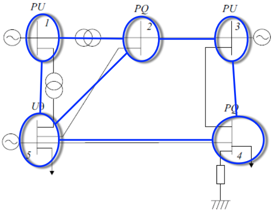
# Distributed Fault Detection and Isolation

## Application: Power Networks

- Active power flow on a loss-less distribution grid.
- Each bus has dynamics given by the ”swing equation“:

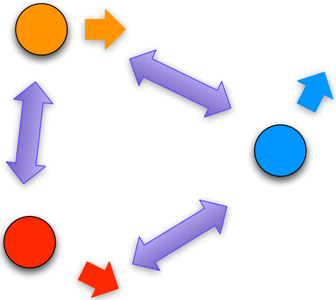
$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i = -\sum_{j\in N_i} w_{ij}\sin(\delta_i - \delta_j) + P_{mi}$$

- $\delta_{ij} = \delta_i - \delta_j$ is small, thus $\sin(\delta_i - \delta_j) \approx \delta_i - \delta_j$.
  - consider $\delta_i$ and $\dot{\delta}_i(t)$ to be states of each bus.
  - Stacking all the states: $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{P}_m$.



# Distributed Fault Detection and Isolation

## Application: Robotic Networks



$$u = \sum_{j \in N_i} w_{ij} \left[ (\xi_j - \xi_i) + \gamma \left( \zeta_j - \zeta_i \right) \right]$$

$\dot{\xi}(t) = \zeta \quad \xi$ : Position

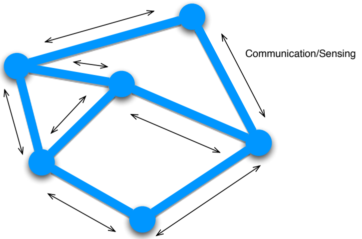
$\dot{\zeta}(t) = u \quad \zeta$ : Velocity

# Distributed Fault Detection and Isolation

## Network Models

Consider $N$ agents

$$\dot{\xi}_i(t) = \zeta_i(t)$$
$$\dot{\zeta}_i(t) = u_i(t),$$




Application 1:

$$u_i(t) = -\frac{d_i}{m_i}\zeta_i(t) + \sum_{j\in N_i} \frac{w_{ij}}{m_i}\left(\xi_j(t) - \xi_i(t)\right)$$

Application 2:

$$u_i(t) = \sum_{j\in N_i} w_{ij}\left[\left(\xi_j(t) - \xi_i(t)\right) + \gamma\left(\zeta_j(t) - \zeta_i(t)\right)\right]$$

# Distributed Fault Detection and Isolation

## Network Models

Set $\mathbf{x}(t) = [\xi_1(t), \cdots, \xi_N(t), \zeta_1(t), \cdots, \zeta_N(t)]^\top$

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t)$$

Application 1:

$$A = \begin{bmatrix} 0_N & I_N \\ -\bar{M}\mathcal{L} & -\bar{D}\bar{M} \end{bmatrix}$$

$$\bar{M} = \text{diag}\left(\frac{1}{m_1}, \cdots, \frac{1}{m_N}\right)$$

$$\bar{D} = \text{diag}\left(d_1, \cdots, d_N\right)$$

Application 2:

$$A = \begin{bmatrix} 0_N & I_N \\ -\mathcal{L} & -\gamma\mathcal{L} \end{bmatrix},$$

$$y_i(t) = C_i\mathbf{x}(t)$$

$\mathcal{L}$: Laplacian.
Fault at agent $j$:

$$\begin{bmatrix} \dot{\xi}_j(t) \\ \dot{\zeta}_j(t) \end{bmatrix} = \begin{bmatrix} \xi_j(t) \\ \zeta_j(t) \end{bmatrix} + f_j(t)$$

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{b}f_j(t)$$

# Distributed Fault Detection and Isolation

## Unknown Input Observer

To generate $\mathbf{r}(t)$ at each of the nodes we use UIOs.

### Definition (UIO)

A state observer is an unknown input observer (UIO) if the state estimation error approaches zero asymptotically, regardless of the presence of an unknown input.

### Theorem

*The necessary and sufficient conditions for a UIO to exist for the above system in are:*

$$\textit{rank}\,(C_i\mathbf{b}) = \textit{rank}\,(\mathbf{b}) = 1, \quad \textit{rank}\left(\begin{bmatrix} sI_{2N} - A & \mathbf{b} \\ C_i & 0_{\tilde{N}_i \times 1} \end{bmatrix}\right) = 2N + 1$$

*for all* $Re(s) \geq 0$*.*

# Distributed Fault Detection and Isolation

## Model-based Fault Detection and Isolation: Sensing Requirements

- Suppose double integrator dynamics.
- For a given $\mathbf{b}$ (fault distribution vector), it is required to sense (have an "appropriate" $C_i$) such that

$$\text{rank}\left(C_i\mathbf{b}\right) = \text{rank}\left(\mathbf{b}\right) = 1$$

$$\text{rank}\left(\begin{bmatrix} sI_{2N} - A & \mathbf{b} \\ C_i & 0_{\tilde{N}_i \times 1} \end{bmatrix}\right) = 2N + 1$$

for all $Re(s) \geq 0$.

**Theorem**

*There exists a $C_i$ corresponding to local measurements at each note $i$ such that the above conditions are satisfied for both protocols.*

**Remark**

Using $|\mathcal{N}_i|$ UIO at $i$ appropriate residuals can be generated such that faults in $\mathcal{N}_i$ can be isolated and detected.

1. Global Model Knowledge, 2. Computationally Expensive.

# Outline

# Distributed Fault Detection and Isolation

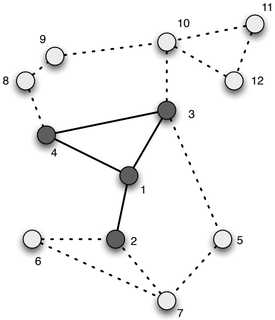
## Model-based Fault Detection and Isolation: Imprecise Network Models

- Exact network model with no fault: $\dot{r}_j^i(t) = F r_j^i(t)$ eas.
- Inexact model, except for one-hop neighbours of $i$:

$$\dot{r}_j^i(t) = F r_j^i(t) + \Delta(t)$$

.

- For the fault free case: $\Delta(t)$ eas.
  - But not for the faulty case.
  - No $r_i^j(t)$ in the network goes to zero.
  - Isolation becomes impossible . . . not quite.

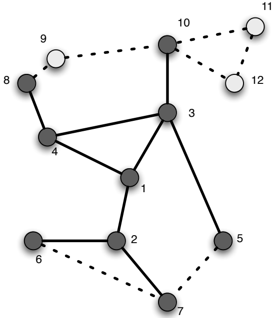


# Distributed Fault Detection and Isolation

## Model-based Fault Detection and Isolation: Imprecise Network Models

### Assumption

- *Only the proximity graph of $i$ is known.*
- *$i$ can measure the states of its two-hop neighbours.*




### Theorem

*Consider the distributed control system with a fault in node $k \in \mathcal{N}_i$ and measurements satisfying above assumption. There exists a UIO for node $i$ that enables to detect and isolate a fault in $k$.*

# Outline

Problem Description

Solution Idea

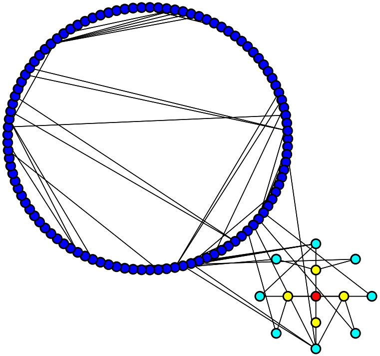
D-FDI with Imprecise Network Models

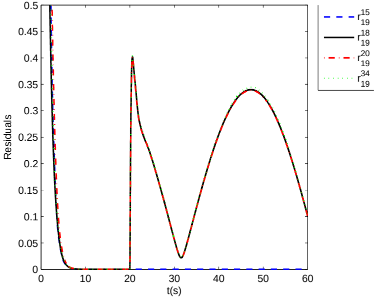
**Simulation**

Concluding Remarks and Future Steps

# Distributed Fault Detection and Isolation

## Imprecise Model: Simulation






89% reduction in the dimension of the observers.

FDI is achieved.

# Outline

# Concluding Remarks and Future Steps

Concluding Remarks:

- Existence of observers for two major linear control laws for double integrator agents.
- Having position or velocity measurements from neighbours, we always can construct an observer at each of the nodes.
- Imprecise interconnection models were handled.
- The dimensions of the observers were decreased.

Near-Future Steps:

- Classification of observable components of a network.
- Replacing network components with their approximate models. .

-Thanks
–The End
—Questions?