Towards Safe and Reliable CPS: a Learning-based Distributed Fault-Diagnosis Approach

T. Parisini* and M. M. Polycarpou**

* Partially supported by ARTEMIS project “CESAR”
** Partially supported by the ERC Advanced Grant “FAULT-ADAPTIVE”
A few definitions from the literature

- CPS are engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing and communication core embedded in all types of objects and structures in the physical environment.

- CPS usually comprise a network of physically distributed embedded sensors and actuators equipped with computing and communicating capabilities. Although each individual device is fairly inept at monitoring or regulating the physical substratum, the coordinated action of the individual network nodes has the potential for unprecedented capabilities.

- CPSs refer to the next generation of engineered systems that require tight integration of computing, communication, and control technologies to achieve stability, performance, reliability, dependability, fault-tolerance, robustness, and efficiency in dealing with physical systems of many application domains.
Cyber-Physical Systems – a Concept Map

http://CyberPhysicalSystems.org
See authors and contributors.
Monitoring and fault diagnosis of CPSs

Motivations

- Huge recent interest in research and applications into reliable methods for diagnosing faults in complex systems
- High levels of safety, performance, reliability, dependability, and availability are needed in several application domains
- Faults: off-specification production, increased operating costs, chance of line shutdown, danger for humans, detrimental environmental impact, ...
- System errors, component faults and abnormal system operation should be detected promptly and the source and severity of each malfunction should be diagnosed (corrective actions)
- The simultaneous presence of a physical substratum and of a cyber substratum imposes additional challenges in safety-critical applications
Basic definitions and concepts

**Fault**  Undesired change in the system that tends to degrade overall performance (a fault not necessarily represents a failure of a physical component)

**Fault detection**  Binary decision: “either something has gone wrong or everything is fine”

**Fault isolation**  Determination of the source/type of the fault

**Fault diagnosis system**  Procedure used to detect and isolate faults and possibly assess their significance/severity
Safety & Fault-tolerance: a step beyond HW redundancy only

Flight Control Computer COM/MOM Airbus Architecture

- Self-checking computers
- Computer switching in case of fault detection
Model-based analytical redundancy: the very basic idea

**Key design issues:**

- Effects of modelling uncertainties
- Non-conservative diagnosis thresholds

**Bibliography** - several books and papers available on FD based on analytical redundancy concept.

Among others:

- Gertler 1988; Patton and Chen, 2001;
- Blanke, Kinnaert, Lunze and Staroswiecki, 2003; Isermann, 2006
**Large-scale CPSs: why distributed?**

Mainly because of constraints on:

- **Computation power** needed to handle the global dynamic model (model-based approach)
- **Communications resources** needed to convey the information on all the state variables to a single location

Moreover:

- FDI task running on a single computation node is not fault tolerant itself, nor a single node for this task can always be identified
- The physical substratum may be **spatially distributed**: the notion of locality induced by the physical substratum is not necessarily compatible with the notion of locality induced by the network of sensors (*Tabuada, 2006*)

**Layer of networked local monitoring modules**
De/centralised, distributed system

**System:** entity to be monitored against faults

**Centralised**
Possibly large # of sub-systems with global interaction

**Decentralised**
Non-interacting sub-systems

**Distributed**
Sub-systems with local interaction
De/centralised, distributed FD architecture

- **Centralised**: All information conveyed to a single diagnostic system.
- **Decentralised**: No information exchange among the local diagnosers, mirroring the physical interactions.
- **Distributed**: The information exchange mirrors the local physical interactions.
Distributed FDI: *divide et impera*

- Convenient to **decompose** the FDI task in smaller sub-tasks that can run in parallel on different local diagnosers.

- Use of **directed graphs** to match the decomposition structure.

- FDI task decomposition follows from a **decomposition** of the monolithic system **structural graph** and **model**.

- **Consensus** techniques used to monitor **overlapping** parts.

- **Adaptive approximators** used to **learn** on-line **uncertain** parts of the **model** (typically, the interconnection between subsystems).
A simple structural graph

\[ G \triangleq \{N_G, E_G\} \]

- **Nodes** represent state or input components of the monolithic system.
- Two nodes are **connected** if the first one appears in the state equation of the second one.

\[ E_G \triangleq \left\{ (x(i), x(j)) : "x(i) acts on x(j)" \right\} \cup \left\{ (u(i), x(j)) : "u(i) acts on x(j)" \right\} \]
A simple graph decomposition

**Local state variables**
\[ x_1 = [x^{(1)}, x^{(2)}, x^{(3)}]^\top \]
\[ x_2 = [x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}]^\top \]

**Local input variables**
\[ u_1 = u^{(1)} \]
\[ u_2 = u^{(2)} \]

**Interconnection variables**
\[ z_1 = [x^{(4)}, x^{(5)}]^\top \]
\[ z_2 = x^{(2)} \]

**Shared variables**
\[ x^{(3)} \equiv x_1^{(3)} \equiv x_2^{(1)} \]

**Overlap set**
\[ \mathcal{O}_3 = \{1, 2\} \]

![Graph Diagram]
Large-scale CPS “monolithic” model of physical substratum

Nominal model dynamics (healthy mode)

\[ x^+ = \phi(x, u) + \eta(x, u, t) + B(t - T_0)f(x, u) \]

Modelling uncertainty (plant/model mismatch)

Time-evolution of the fault

Deviation to state equation due to a fault/malfunction. Several failure causes (e.g., component-level, sensors, but ... malicious too)
Decomposition

We decompose

\[ x^+ = \phi(x, u) + \eta(x, u, t) + B(t - T_0) f(x, u) \]

as

\[ x_I^+ = \phi_I(x_I, u_I) + g_I(x_I, z_I, u_I) + \beta(t - T_0) f_I(x_I, z_I, u_I) \]

with \( I \in \{1, \ldots, N\} \) and

\[ f_I(x_I, z_I, u_I) \in \mathcal{F}_I = \{f_I^1(x_I, z_I, u_I), \ldots, f_I^{N_I}(x_I, z_I, u_I)\} \]
Types of fault: local

- Example of a **local fault**
- **Green arcs** and **nodes** represent the fault influence
- The **influence set** is a singleton

\[ \mathcal{U} = \{1\} \]

- Only one LFD (the first one) is needed to detect and isolate the fault
Types of fault: distributed

- This is an example of a distributed fault, the fault influence set is
  \[ U = \{2, 3\} \]

- The fault is detected as soon as any LFD locally detects it

- After detection every LFD starts the isolation procedure

- The fault is isolated only if all the LFD belonging to \( U \) succeed in isolating their local component of the fault

The global isolation is possible thanks to the GFD
Distributed FDI Architecture

- **Layer 1:** physical subsystems
- **Layer 2:** a Local Fault Diagnoser (LFD) for each subsystem, using local measurements and exchanging information with neighbors
- **Layer 3:** a Global Fault Diagnoser (GFD) exploiting local fault decisions from LFDs in order to reach a global fault diagnosis decision

"physical"  
"cyber"
Distributed FDI Architecture

\[ \dot{x}_1 = \phi_1(x_1, u_1) + g_1(x_1, z_1, u_1) + \beta(t - T_0)f_1(x_1, z_1, u_1) + N_{F_1} x_1 + 1 \]

\[ \dot{x}_2 = \phi_2(x_2, u_2) + g_2(x_2, z_2, u_2) + \beta(t - T_0)f_2(x_2, z_2, u_2) + N_{F_2} x_2 + 2 \]

\[ \dot{x}_3 = \phi_3(x_3, u_3) + g_3(x_3, z_3, u_3) + \beta(t - T_0)f_3(x_3, z_3, u_3) + N_{F_3} x_3 + 3 \]
Distributed FDI Architecture

Local Fault Detection and Isolation Scheme

Fault detection and approximation estimator

Fault detection decision scheme

Bank of $N^F_2$ fault isolation estimators

Fault isolation decision scheme

Activation

Alarm

Fault that has occurred

$x_2^+ = \phi_2(x_2, u_2) + g_2(x_2, z_2, u_2) + \beta(t - T_0)f_2(x_2, z_2, u_2)$
Local FDI architecture

Local Fault Detection and Isolation Scheme

Fault detection and approximation estimator

Fault detection decision scheme

Bank of $N^{f_I}$ fault isolation estimators

Fault isolation decision scheme

Alarm

Fault that has occurred

$u$

$x$

$x_I^+ = \phi_I(x_I, u_I) + g_I(x_I, z_I, u_I) + \beta(t - T_0)f_I(x_I, z_I, u_I)$
Fault detection and approximation estimator

\[ \hat{x}_I^{(s_I)} = \lambda(\hat{x}_I^{(s_I)} - y_I^{(s_I)}) + \lambda \sum_{J \in O_s} W_s^{(I,J)} \left[ \hat{x}_J^{(s_J)} - \hat{x}_I^{(s_I)} \right] \]

+ \sum_{J \in O_s} W_s^{(I,J)} \left[ \phi_J^{(s_J)}(y_J, u_J) + \hat{g}_J^{(s_J)}(y_J, v'_J, u_J, \hat{\vartheta}_J) \right]'

*consensus on shared variables*

*on-line parametrized adaptive approximation model*

*delays/packets drop-out in information exchanged between neighbouring diagnosers*
**Learning algorithm**

\[
\hat{\vartheta}_I^+ = P_{\hat{\Theta}_I} \left[ \hat{\vartheta}_I + \gamma_I H_I^\top [\epsilon_I^+ - \lambda \epsilon_I] \right]
\]

where:

- \( P_{\hat{\Theta}_I} \) projection operator on compact set \( \hat{\Theta}_I \)
- \( \epsilon_I = y_I - \hat{x}_I^{sI} \) (from \( \hat{x}_I^{+(sI)} = \ldots \))
- \( \gamma_I \) learning rate matrix
- \( H_I^\top = \partial \hat{g}_I / \partial \hat{\vartheta}_I \)
Fault detection

A local detection threshold $\bar{\epsilon}_{\bar{i}}^{0}(t)$ can be designed depending on a number of important quantities like, for example, bounds on local modelling uncertainties, etc.

Fault detected if:

$$\exists \bar{i} \in \{1, \ldots, n\}$$

and

$$\exists \bar{t}$$

such that

$$|\epsilon_{\bar{i}}^{0}(\bar{t})| > \bar{\epsilon}_{\bar{i}}^{0}(\bar{t})$$
Local detection of a fault activates the isolation phase

Fault detection decision scheme

Bank of $N_{FI}$ fault isolation estimators

Fault detection and approximation estimator

Activation

Alarm

Fault that has occurred

$x_I^+ = \phi_I(x_I, u_I) + g_I(x_I, z_I, u_I) + \beta(t - T_0) f_I(x_I, z_I, u_I)$
Local fault isolation

Fault $s$ isolated if:

$$\forall r \in \{1, \ldots, n\} \setminus \{s\}$$

$$\exists \bar{i} \in \{1, \ldots, n\}$$

and

$$\exists t^r \geq T_d$$

such that

$$|\epsilon^r_{\bar{i}}(t^r)| > \mu^r_{\bar{i}}(t^r)$$
Direct Reduction Steel Plant

- Chemical plant for turning iron ore into ~94% pure iron
- Technology born in the ‘70s
- World production rose from 0.7 to 64 Mt/year (currently 6% of total iron production - steadily increasing)
- More economical and environment friendly than blast furnaces (40-60% less CO2)
Direct Reduction Steel Plant

- Typical production: 200 t/hour, worth about 100,000 Euro/hour
- Energy consumption: 600 MW, mainly from natural gas
- Time needed for a stop&start: 3 days
- Economical loss caused by a forced maintenance stop due to a fault: about 6 M Euro
Direct Reduction Steel Plant

EAF No.1 83m
diameter = 10 m
height = 53 m

EAF No.2
4M DR Plant

Courtesy: Tenova-HYL
Simplified layout

- Main component: reduction shaft reactor (height ~ 40 m, diameter ~ 10 m)
- Internal pressure ~ 6 bar, internal temperature ~ 1050°C
- Distributed-parameters, highly nonlinear “multi-physics” system
- Pellet flow + gas flow + heat transfer + chemical reactions
Reactor: Modelling for FD

“Unusual” modelling paradigm

Cell method: discrete formulation of Field Laws:

- discrete equations defined on two staggered grids:
  - first grid: primal cells
  - second grid: dual cells

Inside a cell

- Internal energy: \( U_g \) and \( U_p \)
- Molar concentrations: \( C_g \) and \( C_p \)
- Gas and pellet flow: \( p_g \), \( V_g \), and \( V_p \)

\( g: \) “gas”

\( p: \) “pellet”
Distributed FD of the DRI reactor

**Key point:**

the discrete-cell model “imposes” the decomposition of the large scale system into “strings” of cells between measurement locations
Remarks

What I did not tell you:

- Theoretical results available on distributed fault detectability and isolability [Ferrari et al, IEEE TAC 2012, Boem et al., EJC 2011]

- Fault-tolerant control schemes integrating the FD methodology with reconfigurable controllers are available for local sub-systems

\[ \sigma(t) \quad \sigma_{\text{des}}(t) \]

The extension to the distributed fault-tolerant control problem of CPSs is very challenging. Efforts in this direction are under way.
Concluding remarks

- Safety-critical CSPs: need of effective monitoring and diagnosis methodologies and tools
- Several CPSs are very large-scale spatially distributed systems: need of distributed diagnosis tools with scalability characteristics
- Enormous value comes from exploiting the richness of the ever-increasing amount of available data from sensors (wireless or not)
- On-line approximation/learning: a key enabling factor to achieve effective distributed diagnosis and prognosis