Fault-Tolerant Control
- Present and Future -

Jan Lunze

Lehrstuhl für Automatisierungstechnik und Prozessinformatik

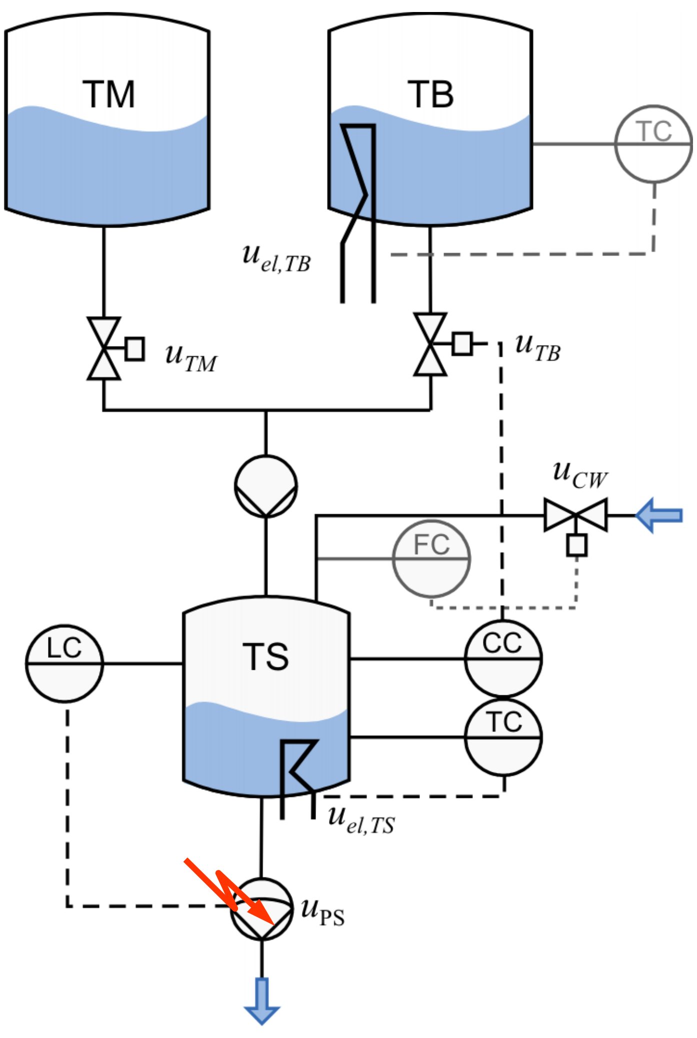
Ruhr-Universität Bochum

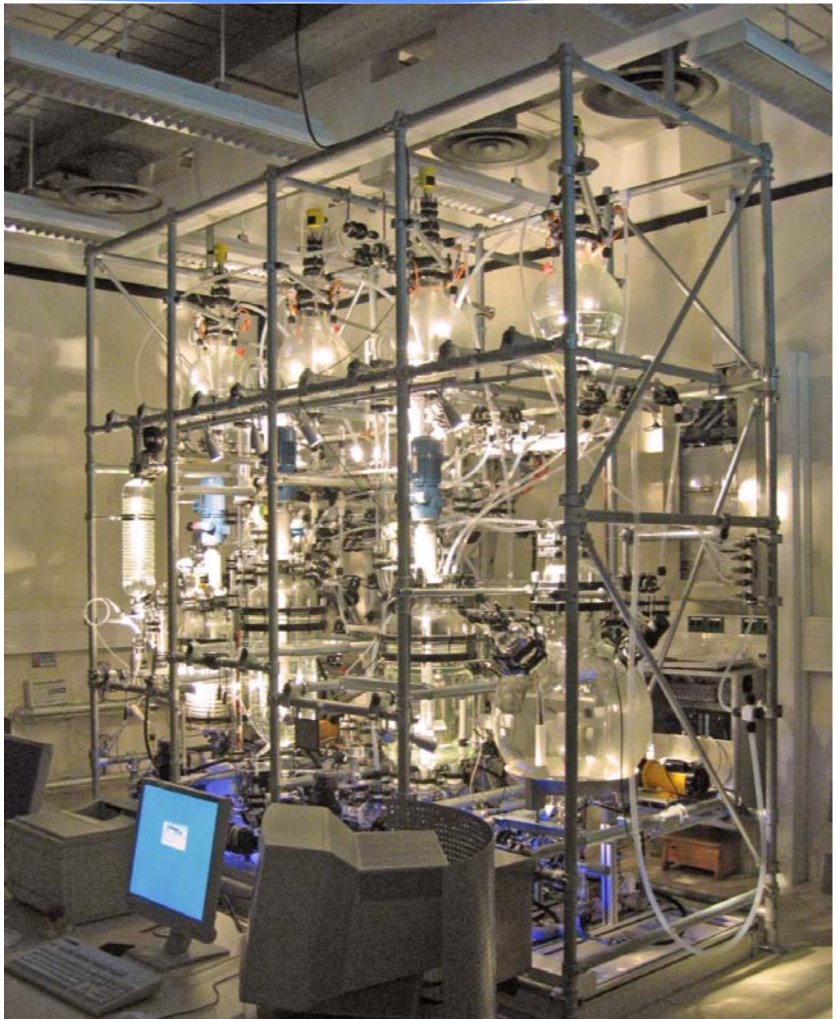
Lunze@atp.rub.de

Outline

1. Introduction to fault-tolerant control
2. Fault diagnosis
3. Controller re-design
4. Fault-hiding approach to control reconfiguration
5. Conclusions and future trends

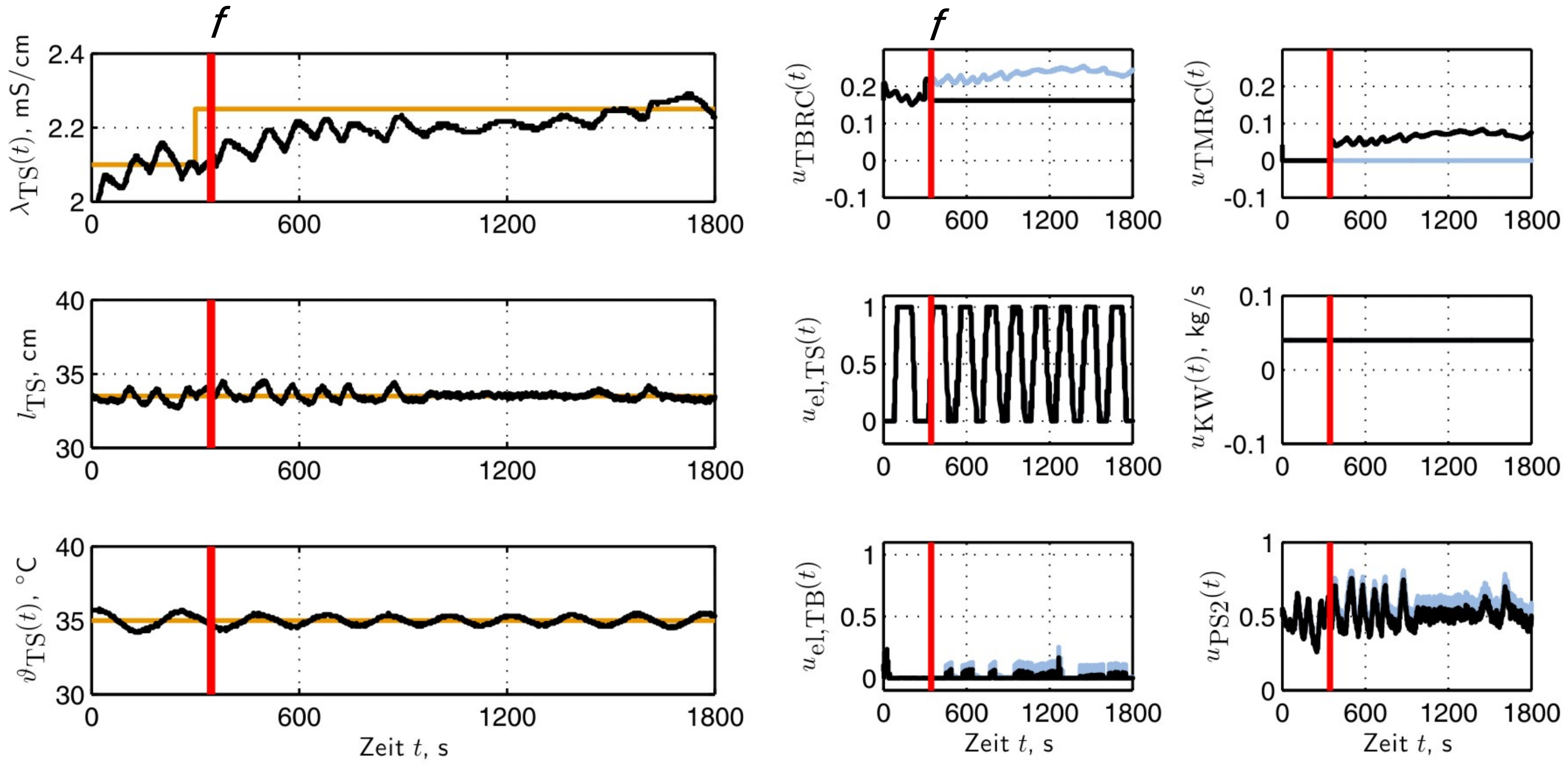
1. Introduction to fault-tolerant control



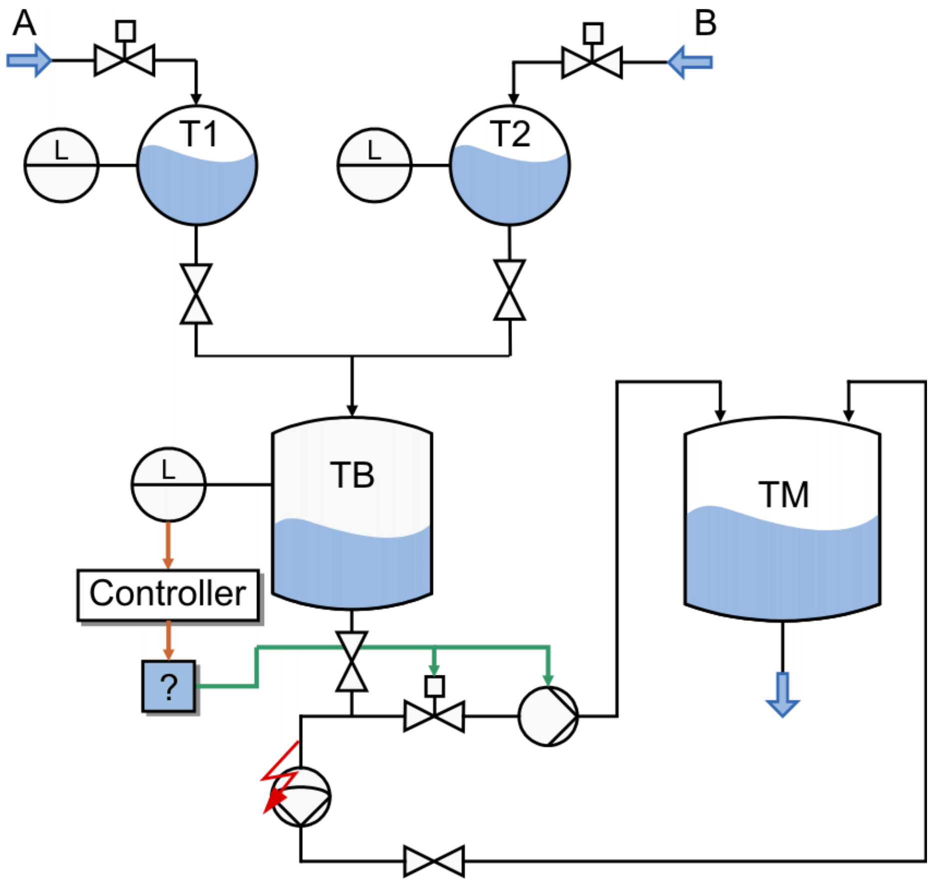


Pilot plant at the Institute ATP

1. Introduction to fault-tolerant control



1. Introduction to fault-tolerant control



Problems to be solved:

- Find the fault
- Select a new control configuration
- Design the controller automatically

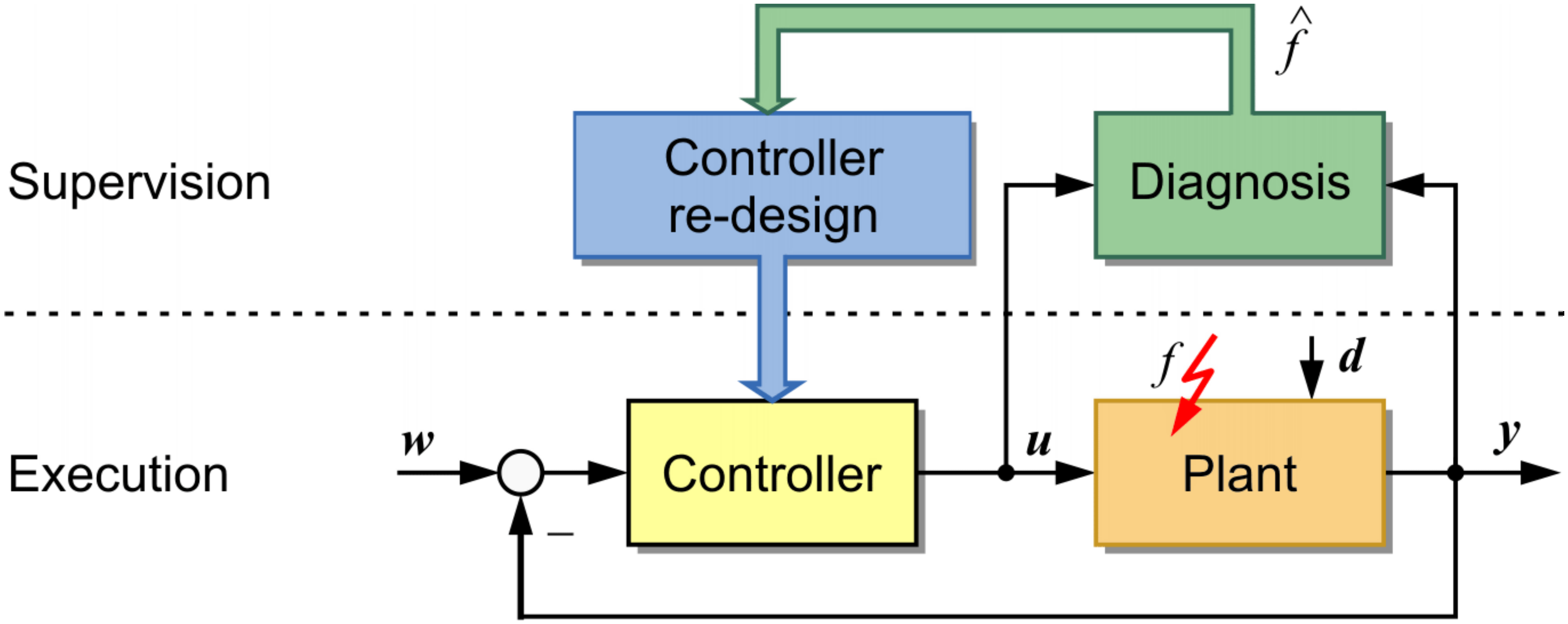
1. Introduction to fault-tolerant control

Fault tolerance

= property of a system to continue to function if components fail

- Graceful degradation of the loop performance
- Enhanced availability instead of sudden shut-down

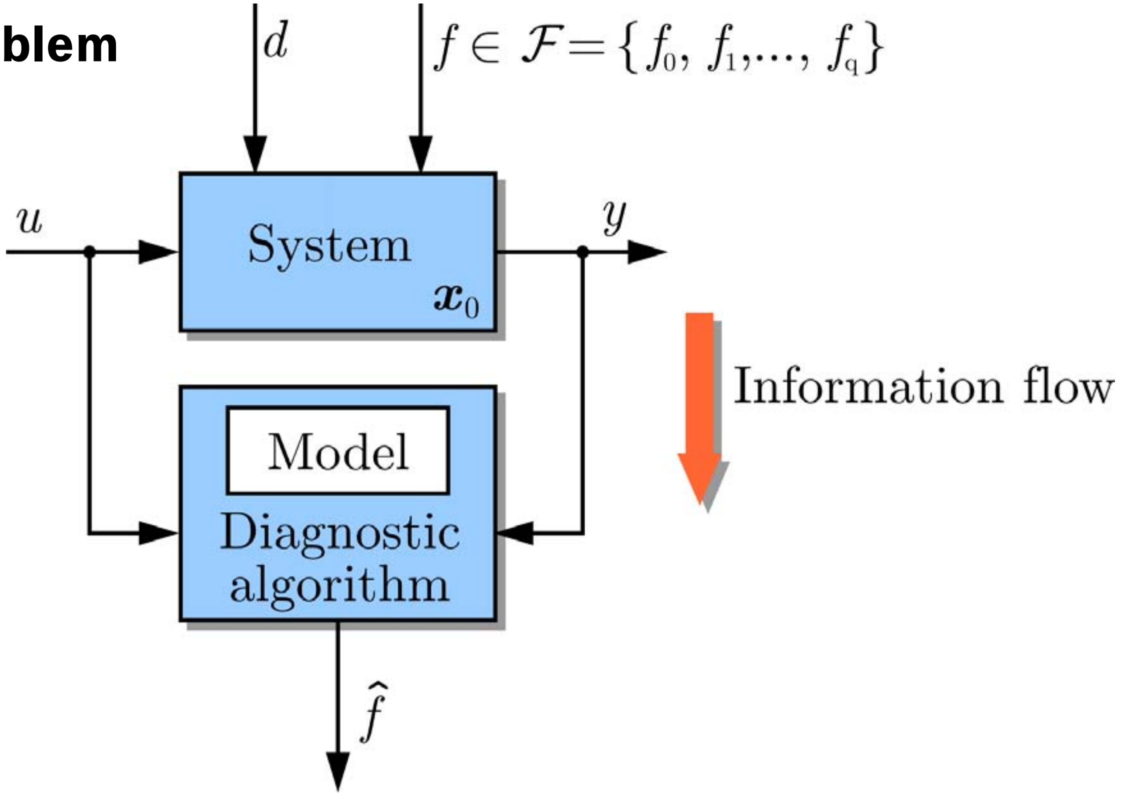
1. Introduction to fault-tolerant control



Blanke, Kinnaert, Lunze, Staroswiecki: *Diagnosis and Fault-Tolerant Control.* (2. Ed.), Springer 2006

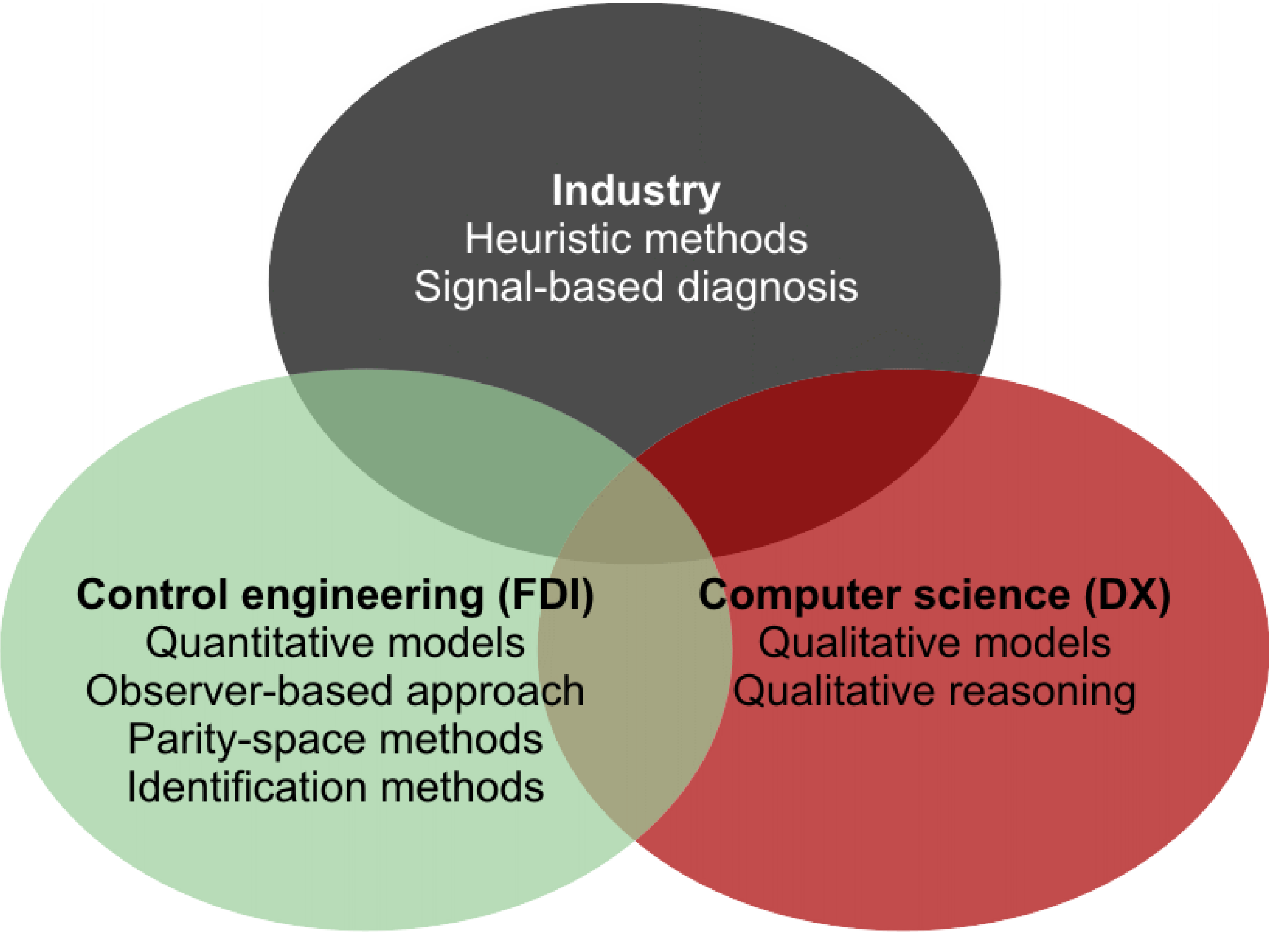
2. Fault diagnosis

Diagnostic problem



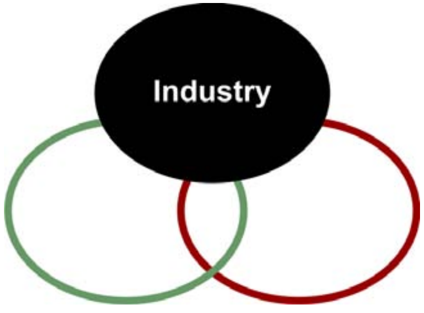
1. **Fault detection:** Does a fault occur?
2. **Fault isolation:** Which component is faulty?
3. **Fault identification:** Which fault occurs?

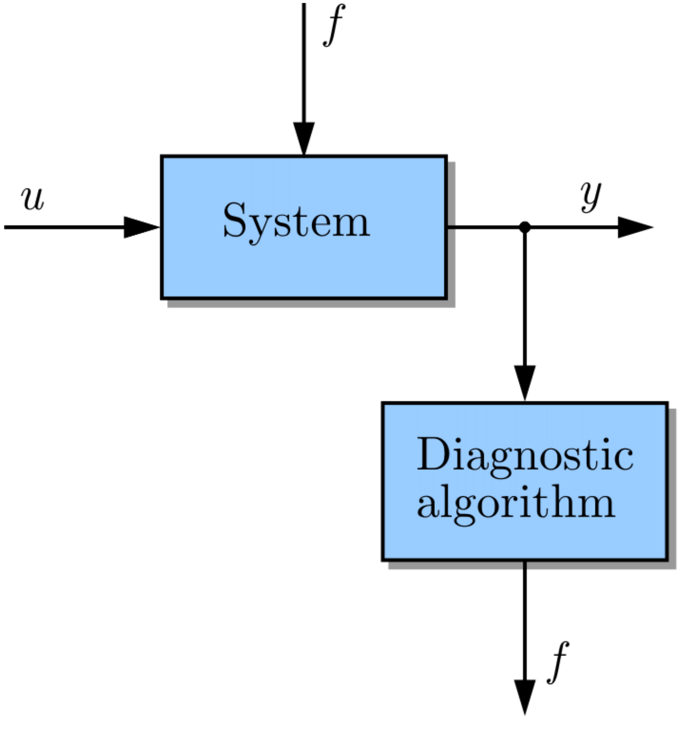
2. Fault diagnosis



2. Fault diagnosis

Signal-based diagnosis

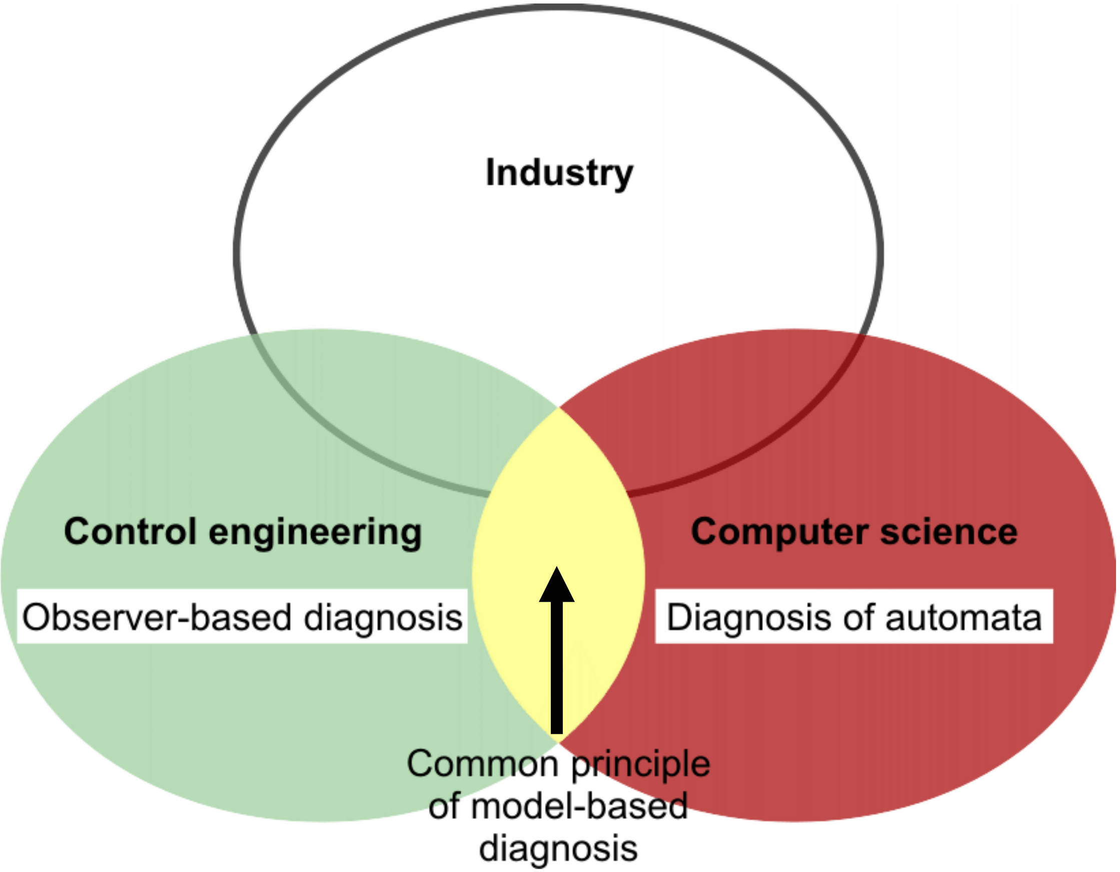




Threshold checking
Alarm handling

- applicable only to stationary processes
- restricted to fault detection

2. Fault diagnosis



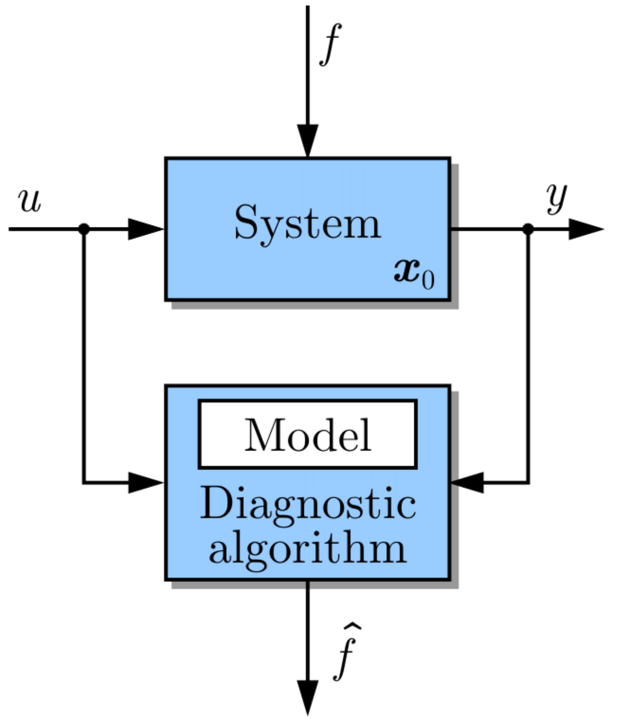
L. Travé-Massuyès et al.: Comparing diagnosability in computer science and discrete-event systems, *SAFEPROCESS 2006*

M.O. Cordier et al.: Conflicts versus analytical redundancy relations. *IEEE Trans. SMC*, October 2004.

2. Fault diagnosis

Consistency-based diagnosis:

A fault changes the I/O-behaviour of a system: $M(f)$



Does the system behave like the model?

- **Fault detection**:

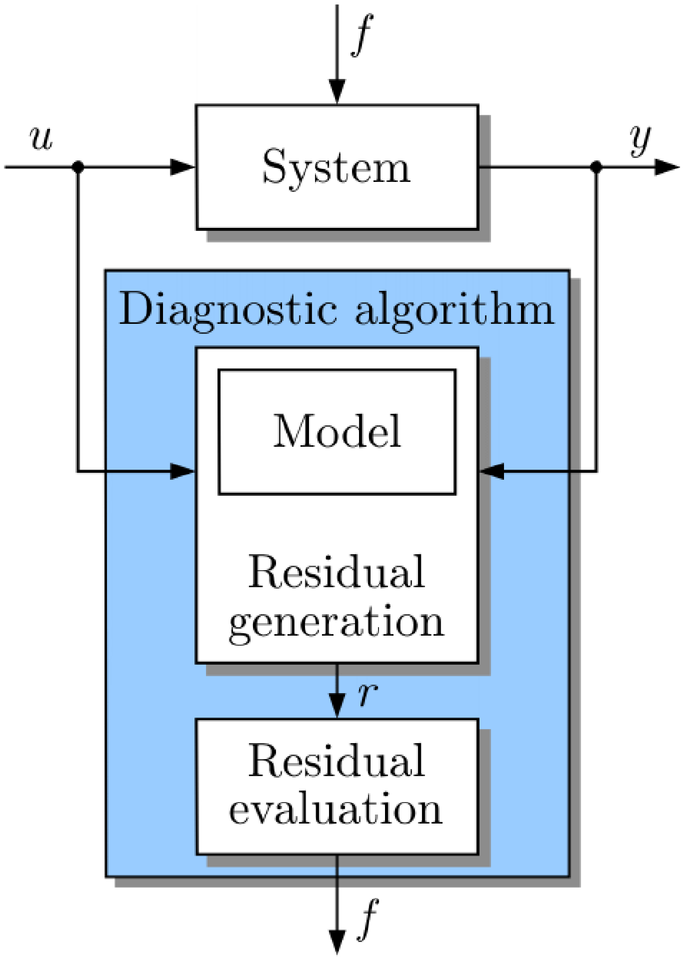
  If the input-output pair $(u(t), y(t))$ is inconsistent with the model $M(f_0)$, a fault has occurred

- **Fault identification**:

  If the input-output pair $(u(t), y(t))$ is consistent with the model $M(f_i)$, the fault f_i may have occurred

2. Fault diagnosis

Consistency-based diagnosis of continuous systems



$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) + \boldsymbol{g}f(t)$$
$$y(t) = \boldsymbol{c}'\boldsymbol{x}(t)$$

Residual with respect to the model $M(f)$:

$$r(f, t) = y(t) - y(f, t)$$

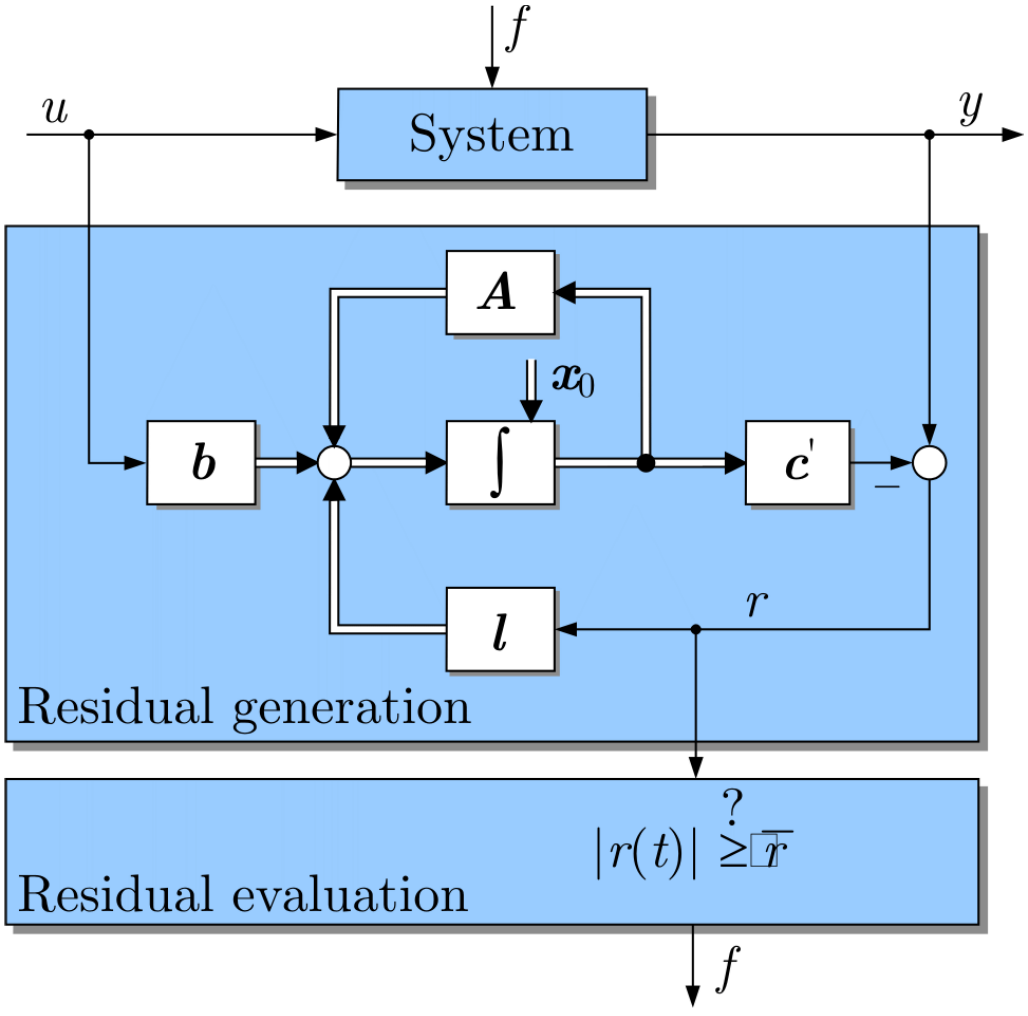
If $|r(f, t)| > \bar{r}$ holds, the I/O-pair is inconsistent with the model of the system subject to fault f

Set of fault candidates:

$$\mathcal{F} = \{f : |r(f, t)| < \bar{r}\}$$

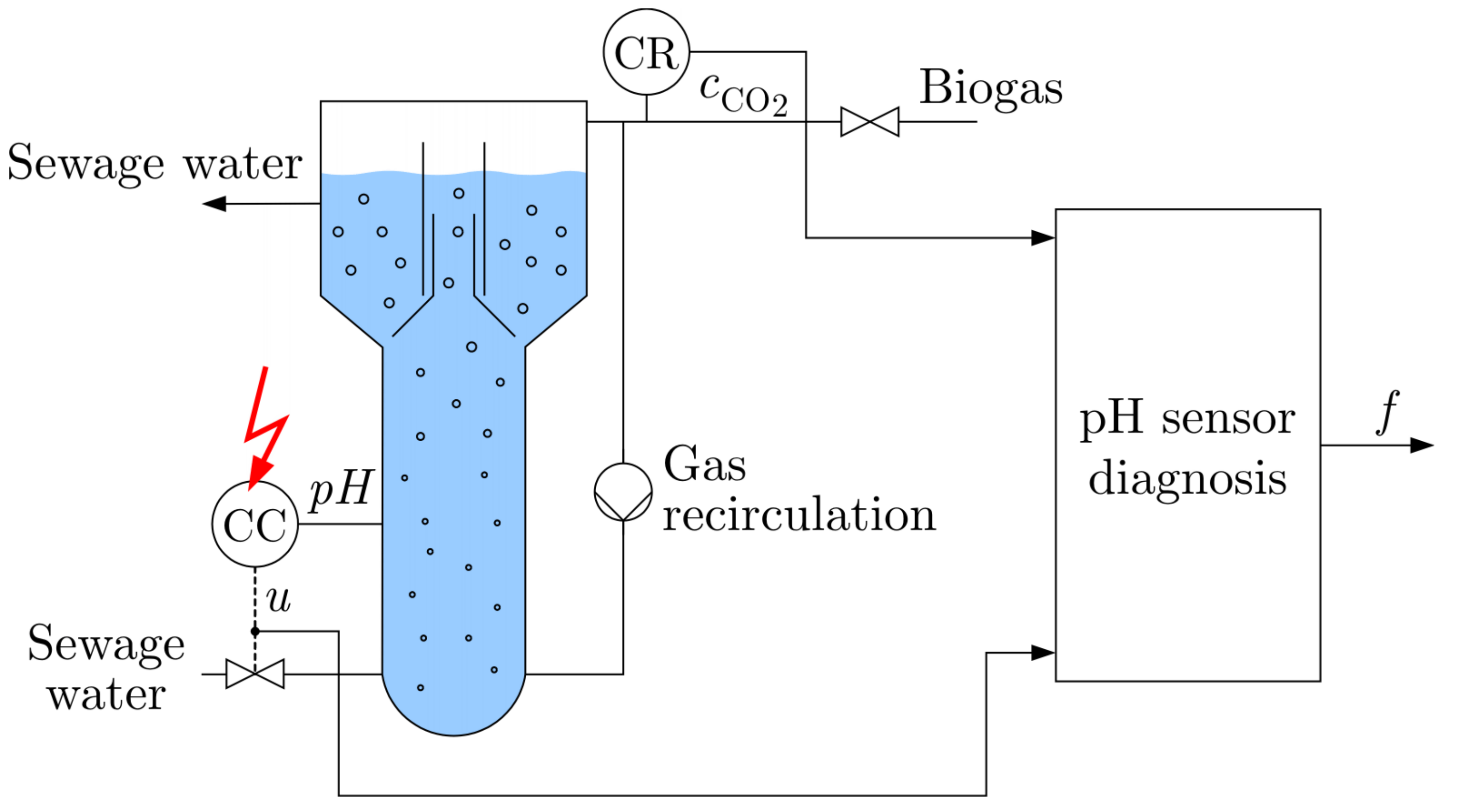
2. Fault diagnosis

Residual generation by means of a state observer



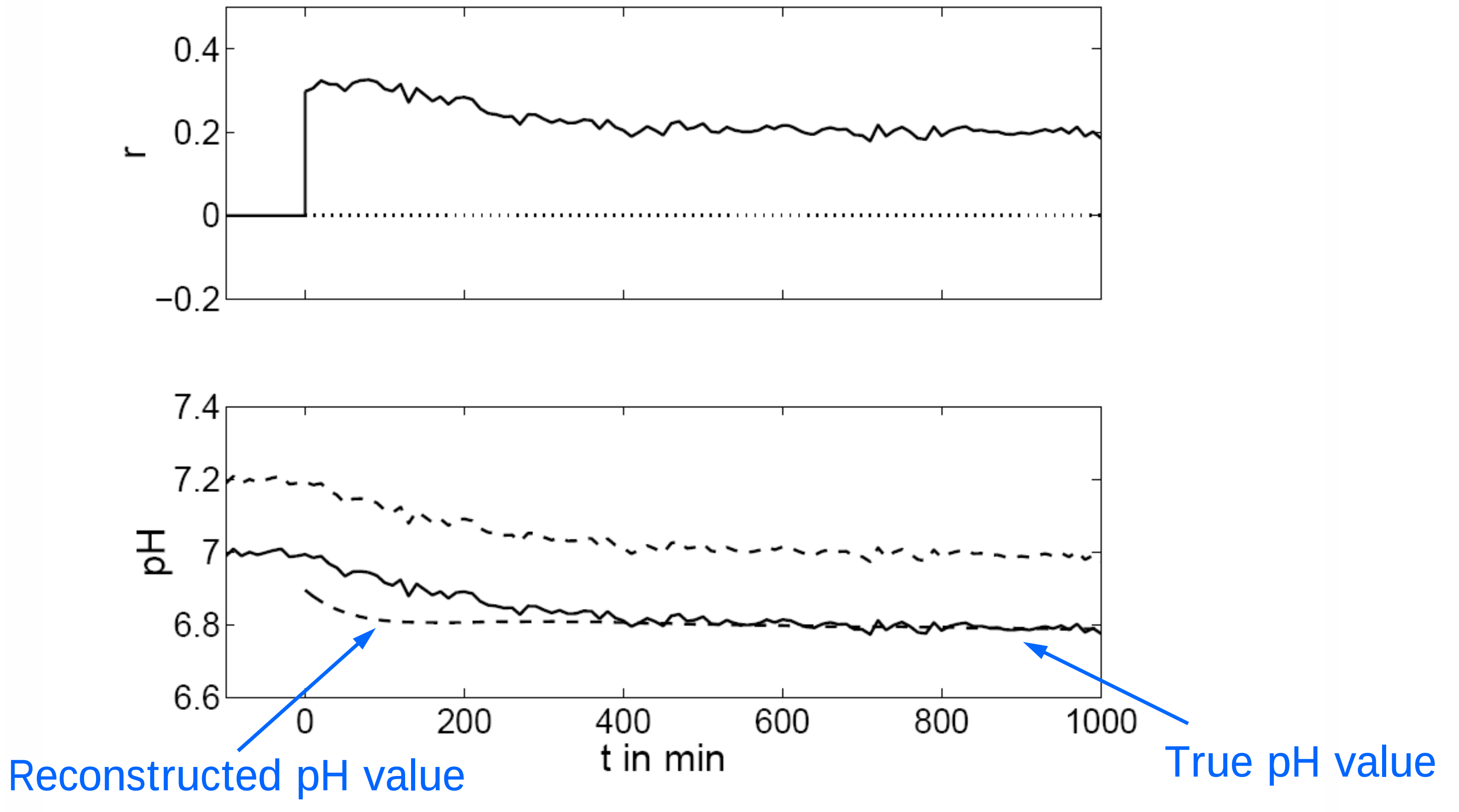
2. Fault diagnosis

Example: Monitoring of a pH sensor



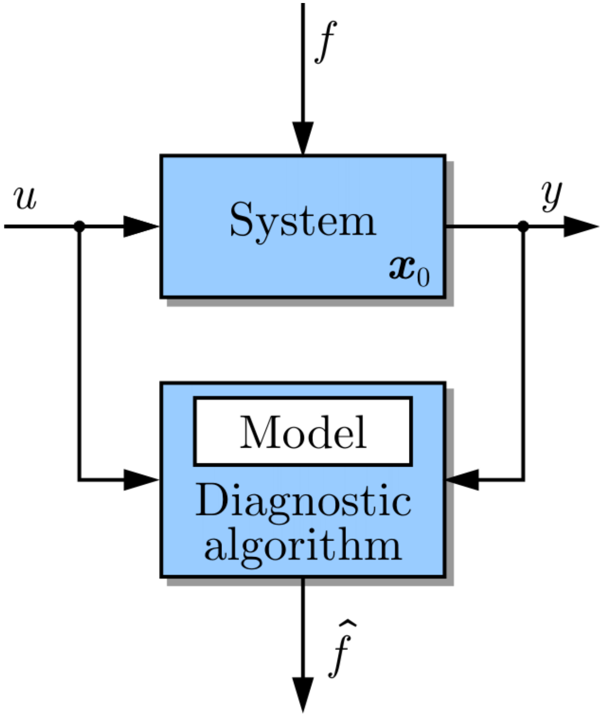
2. Fault diagnosis

Example: Monitoring of a pH sensor

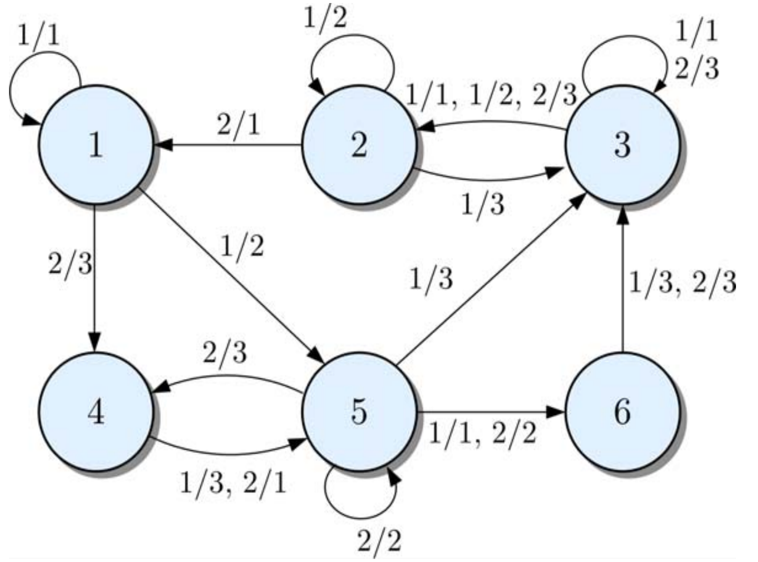


2. Fault diagnosis

Consistency-based diagnosis of discrete-event systems



$$(z(k+1), \boldsymbol{y}(k), z(k), \boldsymbol{u}(k)) \in \mathcal{L}_{\mathrm{n}}, \qquad z(0) = z_0$$

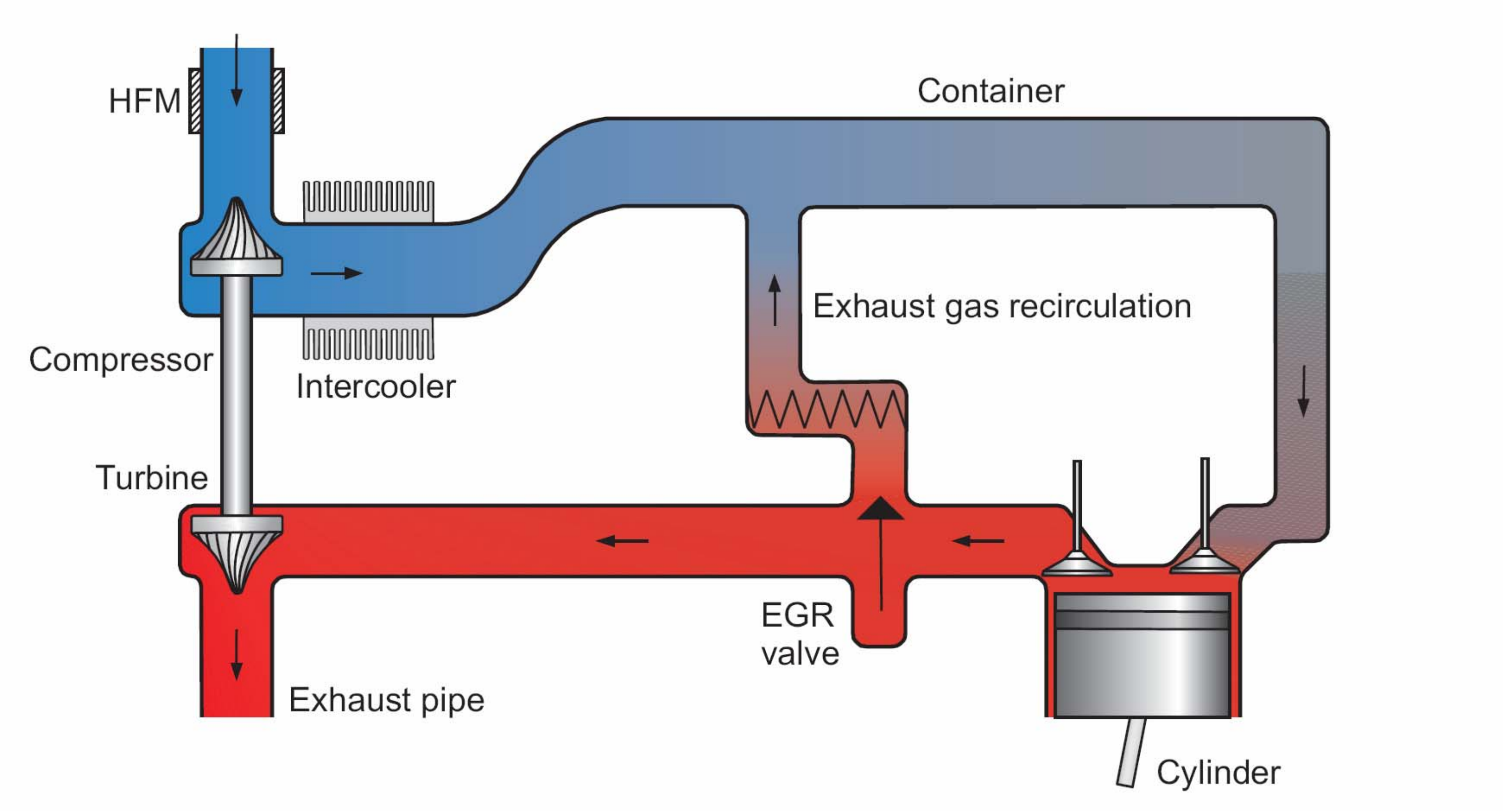


Set of fault candidates:

$$\mathcal{F} = \{f : M(f) \text{ is consistent with } (u, y)\}$$

2. Fault diagnosis

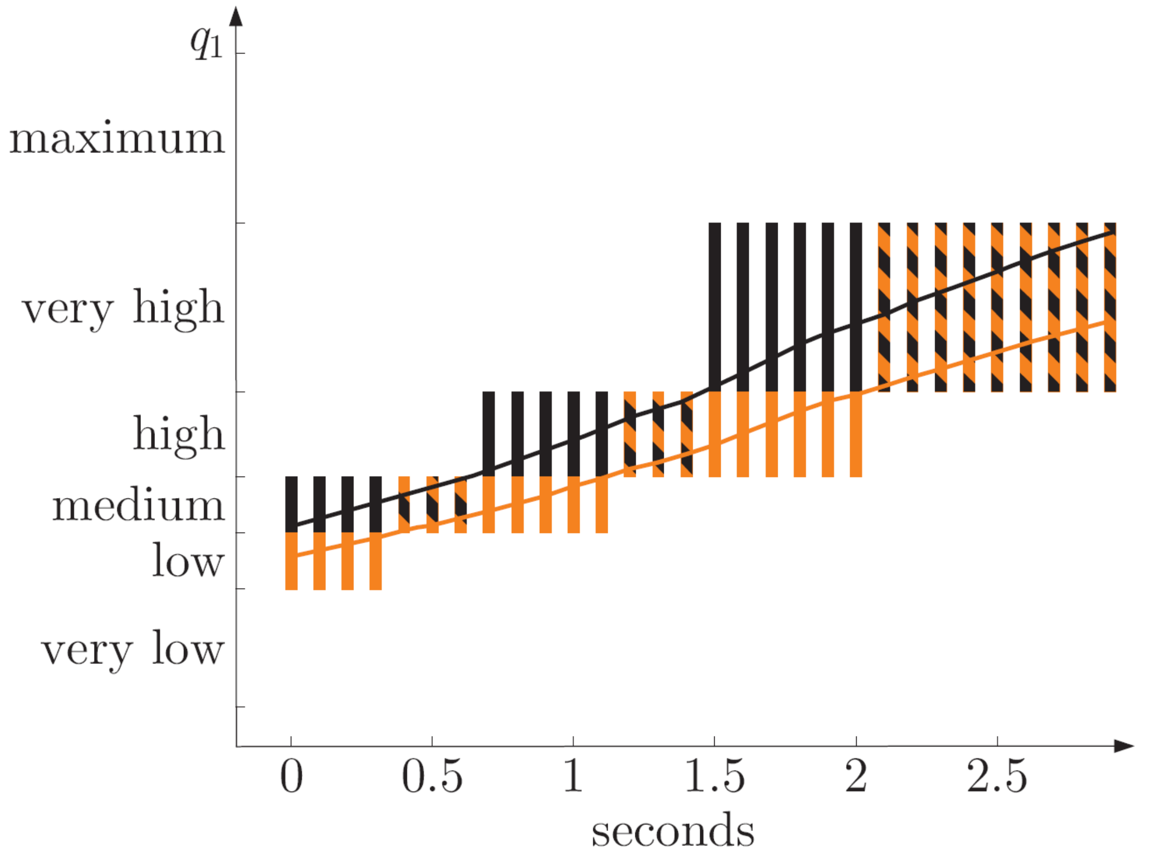
Example: Diagnosis of the air path of a diesel motor



Neidig, Falkenberg, Lunze, Fritz: Qualitative diagnosis of an automotive air path, *ATP international 2005*

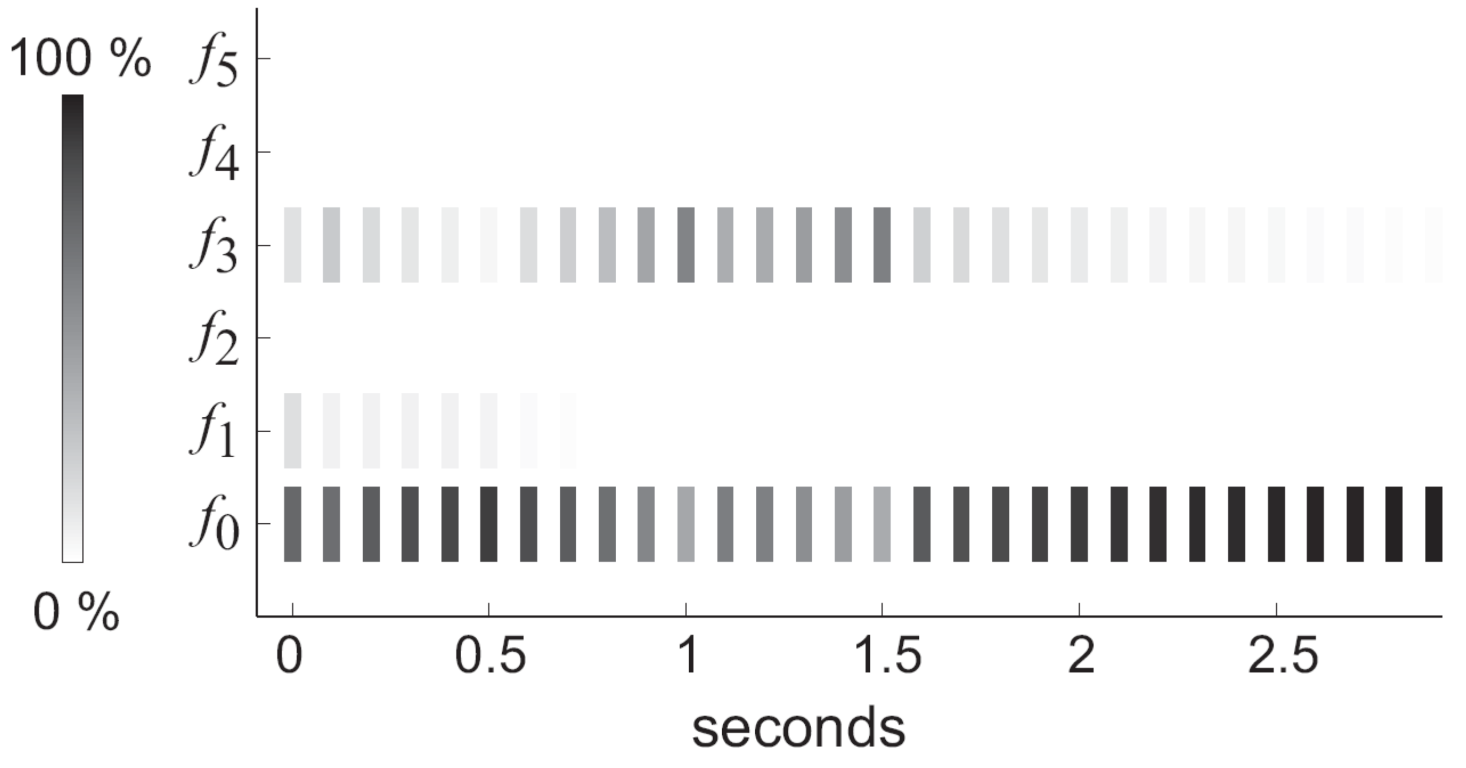
2. Fault diagnosis

Example: Diagnosis of the air path of a diesel motor



2. Fault diagnosis

Example: Diagnosis of the air path of a diesel motor



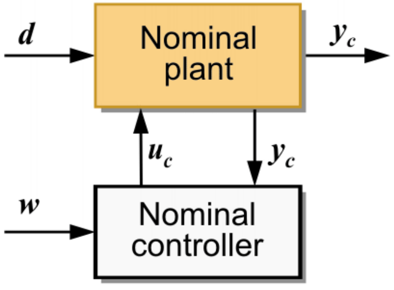
Fault diagnosis

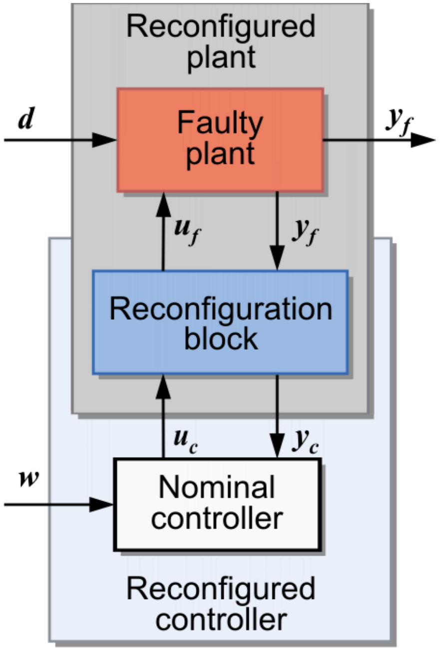
Summary of fault diagnosis

- Fault diagnosis necessitates analytical redundancy (model of the faultless and the faulty system)
- Consistency tests yield the set of fault candidates

4. Fault-hiding approach to control reconfiguration

Fault hiding: The reconfigured plant should have the same properties as the faultless plant





- Stabilisation
- I/O equilibrium:

$$t \to \infty, \forall \bar{\boldsymbol{u}}_c \sigma(t), \bar{\boldsymbol{d}}\sigma(t):$$
$$\boldsymbol{y}_f(t) - \boldsymbol{y}_c(t) \to \boldsymbol{0}$$

- I/O trajectory following:

$$\forall t, \boldsymbol{u}_c(t), \boldsymbol{d}(t):$$
$$\boldsymbol{y}_f(t) - \boldsymbol{y}_c(t) = \boldsymbol{0}$$

Steffen: *Control Reconfiguration of Dynamical Systems*. LNCIS Vol. 320, Springer 2005

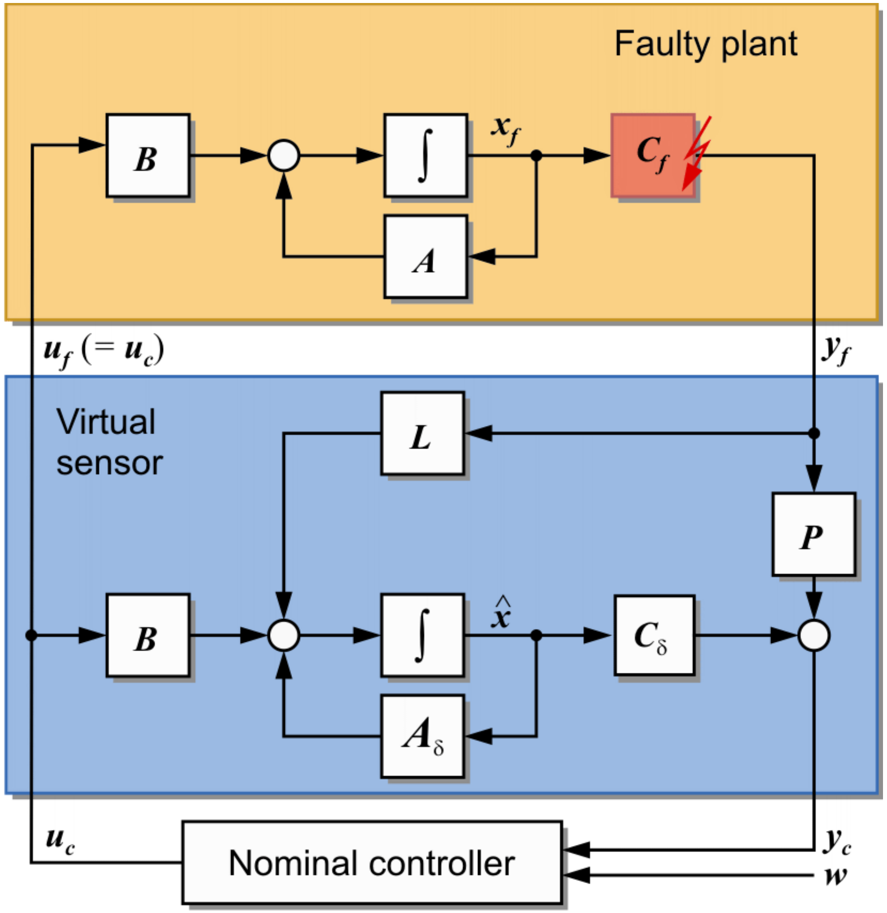
4. Fault-hiding approach to control reconfiguration

Virtual Sensor

$$\hat{\boldsymbol{x}}(t) = \boldsymbol{A}_\delta \hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}_c(t) + \boldsymbol{L}\boldsymbol{y}_f$$
$$\boldsymbol{u}_f(t) = \boldsymbol{u}_c(t)$$
$$\boldsymbol{y}_c(t) = \boldsymbol{C}_\delta \hat{\boldsymbol{x}}(t) + \boldsymbol{P}\boldsymbol{y}_f(t)$$

- Reconfigured plant

$$\boldsymbol{e}(t) = \hat{\boldsymbol{x}}(t) - \boldsymbol{x}_f(t)$$
$$\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{e}}(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_\delta \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{e}(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{u}_c(t)$$
$$\begin{pmatrix} \boldsymbol{y}_c(t) \\ \boldsymbol{y}_f(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{C}_\delta \\ \boldsymbol{C}_f & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{e}(t) \end{pmatrix}$$



- **Separation theorem**
- **The fault-hiding goal is satisfied**

3. Fault-hiding approach to control reconfiguration

Virtual Actuator

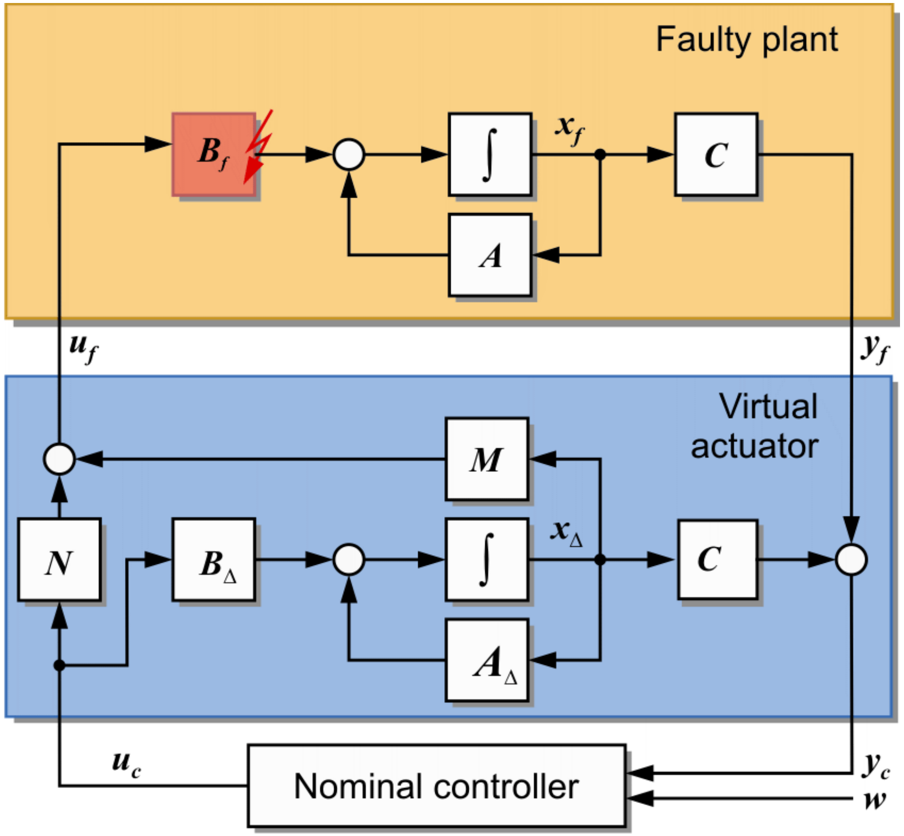
$$\boldsymbol{x}_\Delta(t) = \boldsymbol{A}_\Delta \boldsymbol{x}_\Delta(t) + \boldsymbol{B}_\Delta \boldsymbol{u}_c(t)$$
$$\boldsymbol{u}_f(t) = \boldsymbol{M}\boldsymbol{x}_\Delta(t) + \boldsymbol{N}\boldsymbol{u}_c(t)$$
$$\boldsymbol{y}_c(t) = \boldsymbol{C}\boldsymbol{x}_\Delta(t) + \boldsymbol{y}_f(t)$$

- Reconfigured plant

$$\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}_f(t) + \boldsymbol{x}_\Delta(t)$$

$$\begin{pmatrix} \dot{\tilde{\boldsymbol{x}}}(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & 0 \\ 0 & \boldsymbol{A}_\Delta \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{B}_\Delta \end{pmatrix} \boldsymbol{u}_c(t)$$

$$\begin{pmatrix} \boldsymbol{y}_c(t) \\ \boldsymbol{y}_f(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{C} & 0 \\ \boldsymbol{C} & -\boldsymbol{C} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix}$$



- **Separation theorem**
- **The fault-hiding goal is satisfied**

4. Fault-hiding approach to control reconfiguration

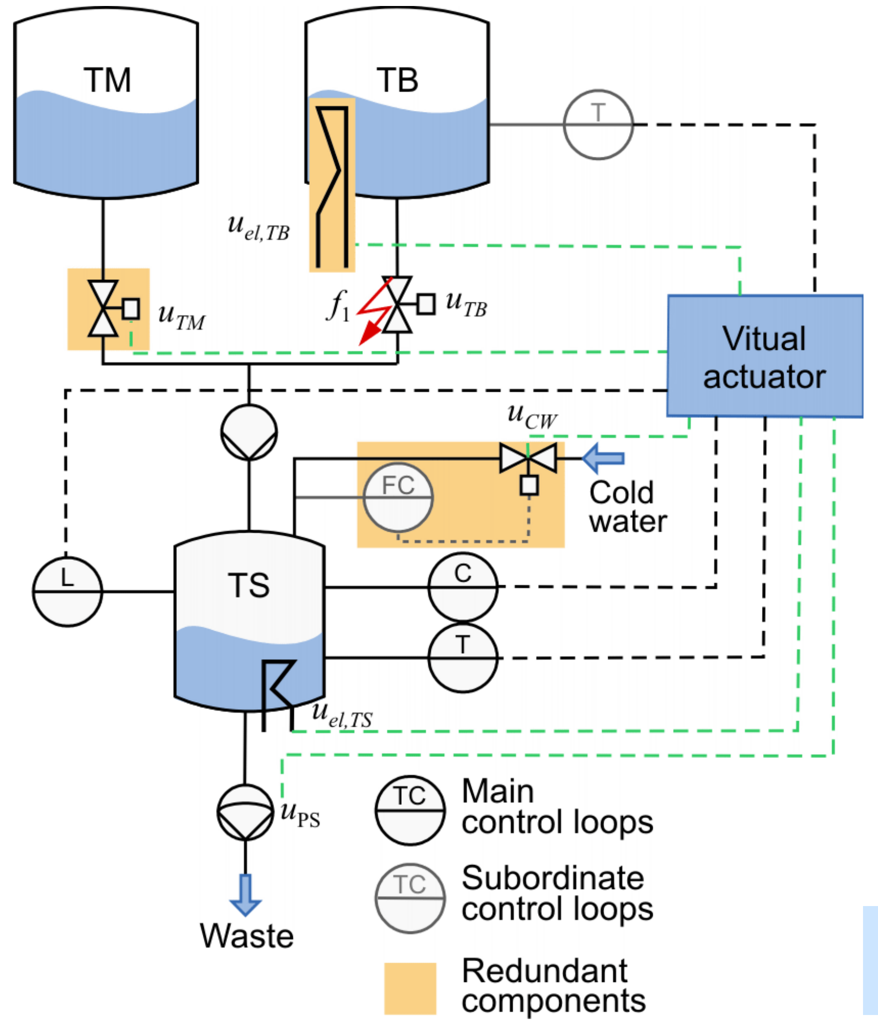
On-line algorithm for reconfiguration after actuator failures

Given: Model $(\boldsymbol{A},\ \boldsymbol{B},\ \boldsymbol{B}_f,\ \boldsymbol{C})$

1. Determine the strongest reachable aim
2. Design of the matrices $\boldsymbol{M},\ \boldsymbol{N}$
3. Apply the virtual actuator to reconfigure the loop

Result: Reconfigured loop, which satisfies the strongest reachable reconfiguration aim

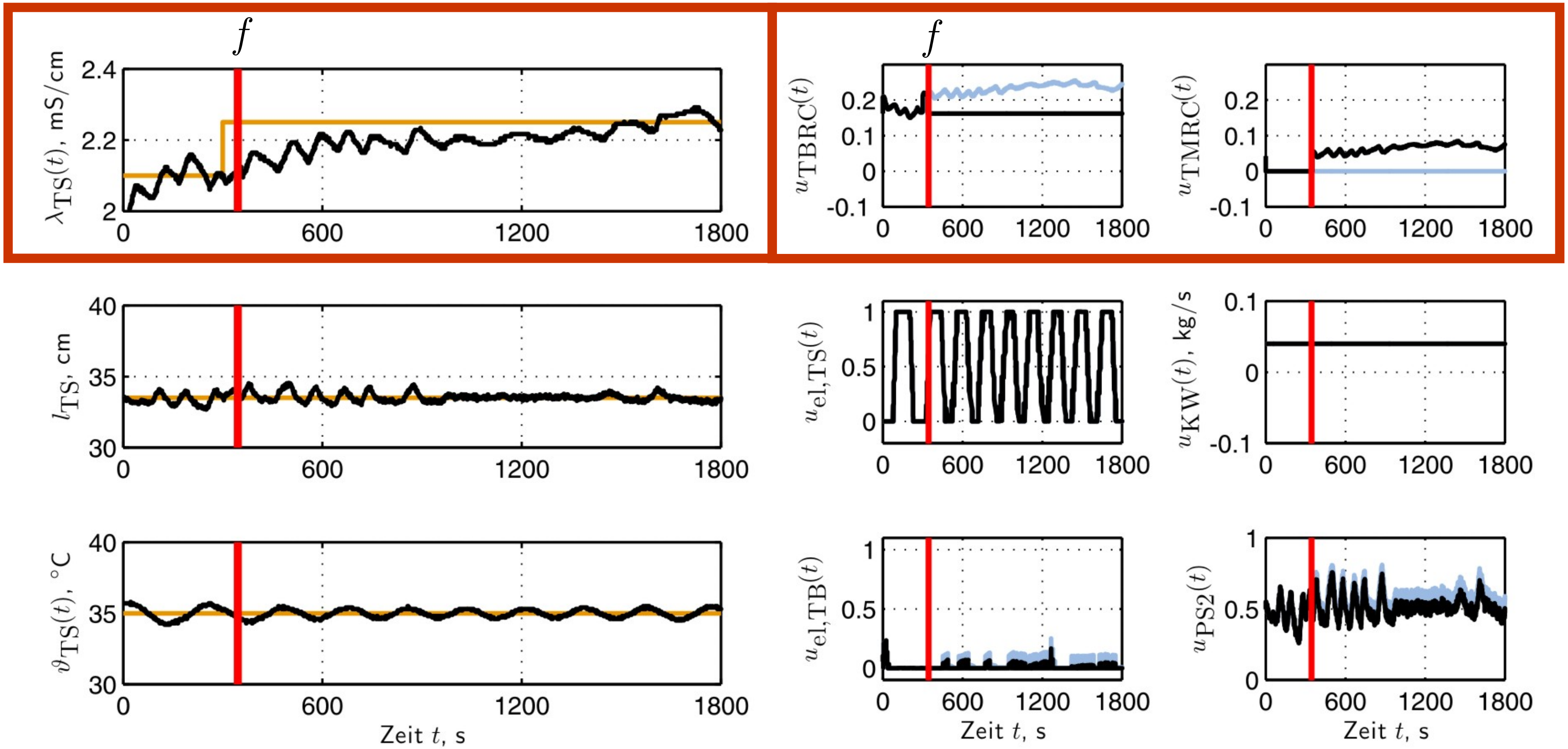
4. Fault-hiding approach to control reconfiguration



Richter, Schlage, Lunze: Control reconfiguration of a thermofluid process, *IET Proceedings*, 2007

4. Fault-hiding approach to control reconfiguration

Control variables **Inputs**



5. Conclusion and future trends

Fault-tolerant control is accomplished in two steps:

1. Fault diagnosis

Based on analytical redundancy (model)
and consistency tests

Established for continuous systems,
Currently in development for discrete-event systems

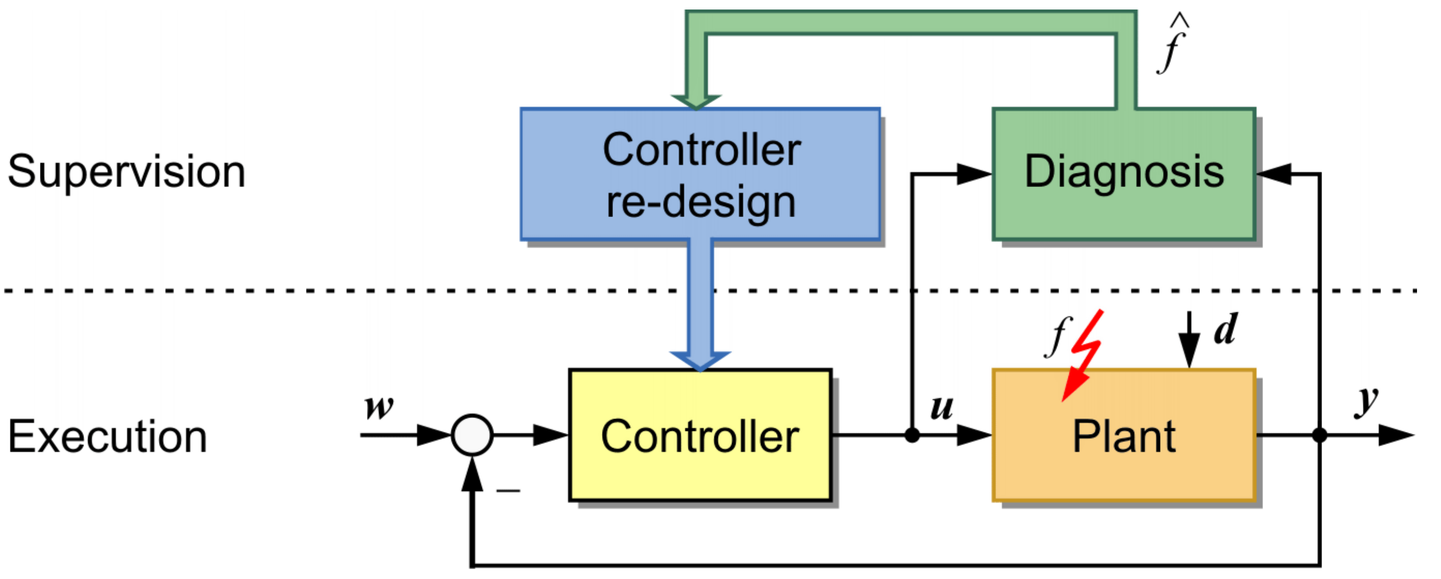
2. Controller re-design

Needs physical redundancy (additional sensors, actuators)

Several approaches exist for fault accomodation and control reconfiguration

5. Conclusion and future trends

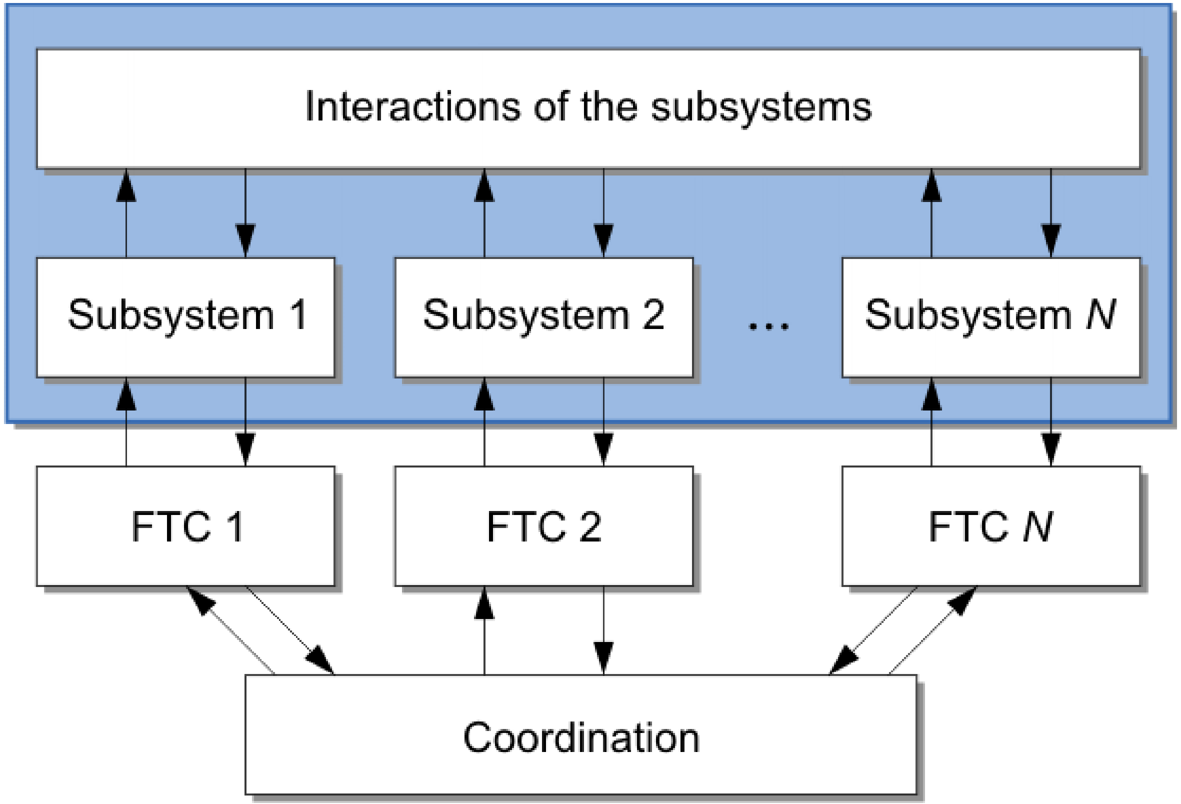
Integration of fault diagnosis and controller re-design



- Time-to-reconfigure?
- How to deal with the uncertainties of the diagnostic result?

5. Conclusion and future trends

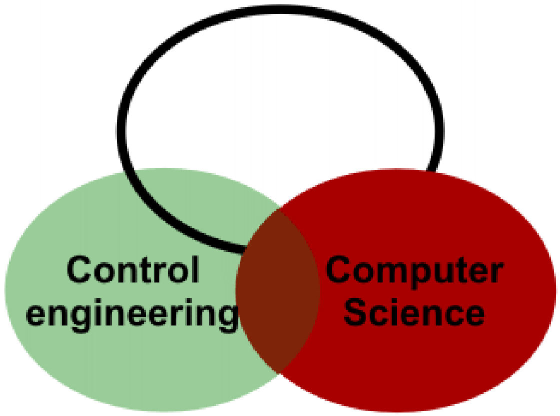
Distributed fault-tolerant control

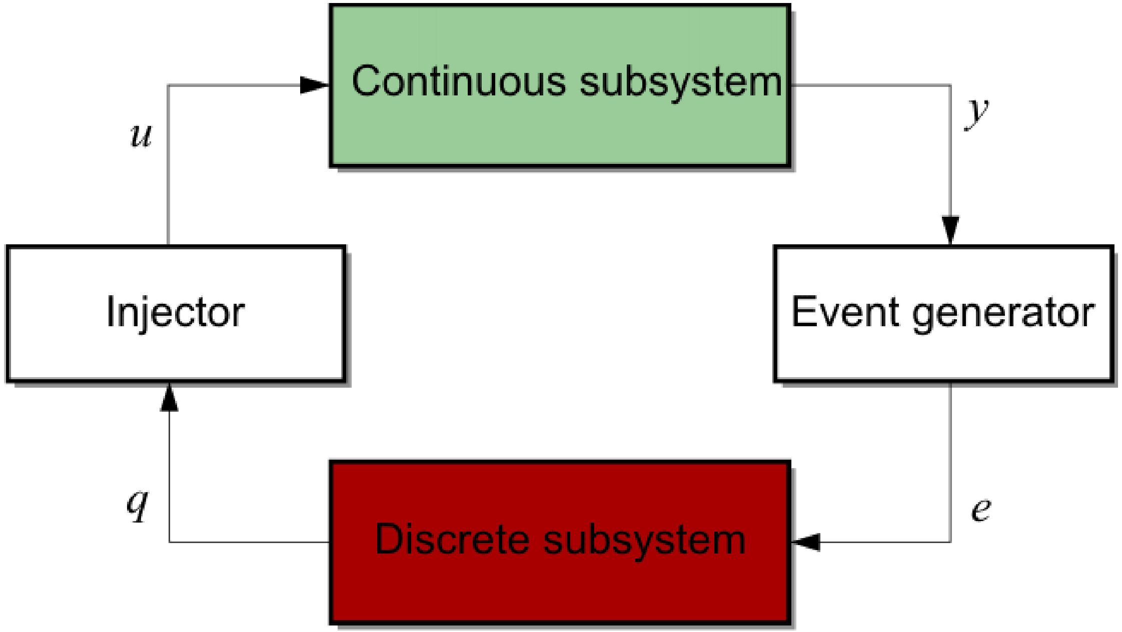


- How to merge the local diagnostic and re-design results?
- How to re-distribute the control functions among the subsystems?

5. Conclusion and future trends

Fault-tolerant control of hybrid systems

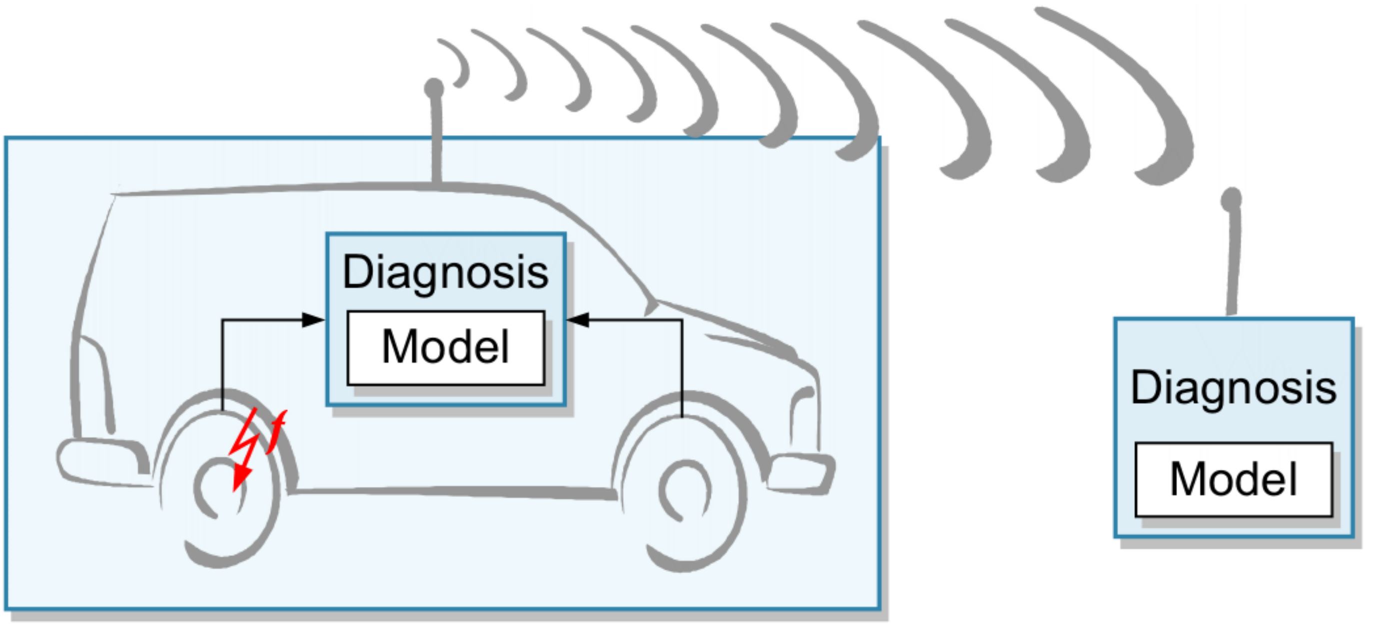




- Use the common foundation for diagnosis of mixed discrete-continuous systems
- How to re-design hybrid controllers?

5. Conclusion and future trends

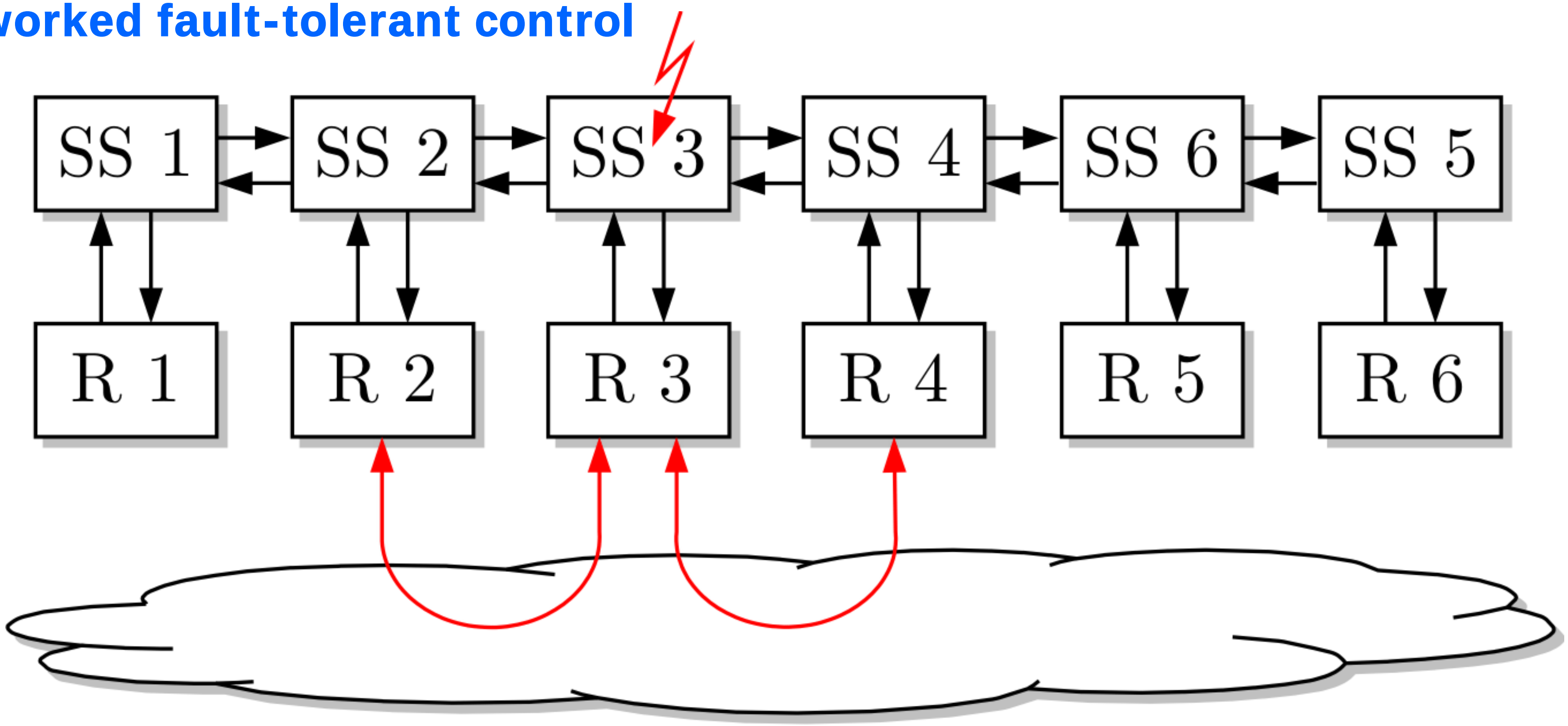
Networked fault-tolerant control



On-board diagnosis vs. Off-board diagnosis

5. Conclusion and future trends

Networked fault-tolerant control



Create a temporary link among decentralised control loops

- Which information is necessary for ensuring fault-tolerance?
- Combine control and communication methods