ISRCS 2011

Supervisory control of a pilot-scale cooling loop

Kris Villez, Venkat Venkatasubramanian, Humberto Garcia, Craig Rieger

PURDUE ENGINEERING
Think impact.™

Laboratory for Intelligent Process Systems

INL

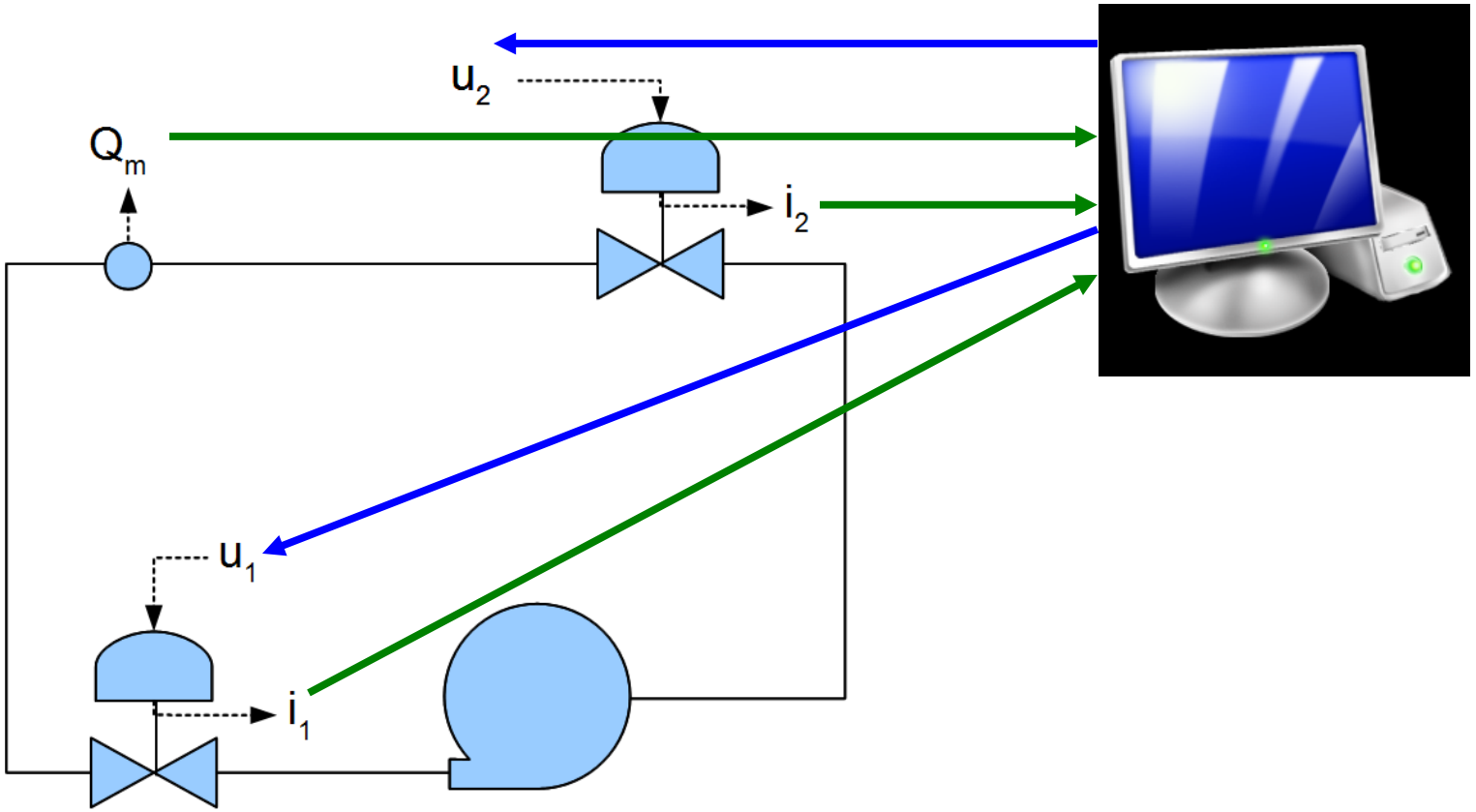
Overview

- Background
- Pilot-scale plant
- Problem statement
- Method
- Results
  - In simulation
  - In pilot-scale plant
- Future

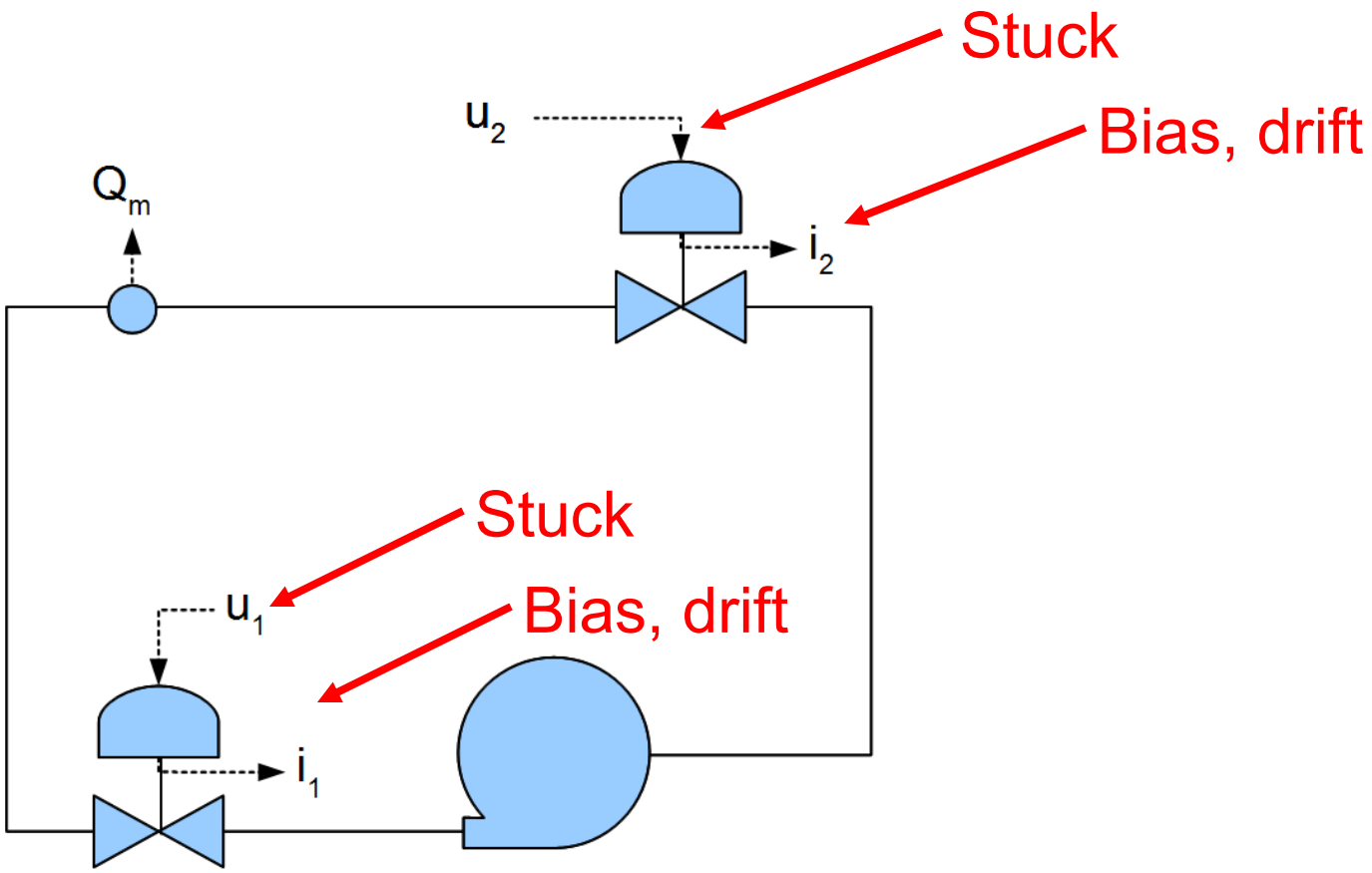
Background

- Supervisory control
  - Changing conditions:
    - Internal: System/process conditions
    - External:
      - Demand changes (known)
      - Changes in environment (unknown)
  - This talk: internal
    - Faults in sensors and actuators
    - Different locations in system
    - Different types
      - Bias/drift/stuck
  - Challenge:
    - Fault introduced at unknown time
    - Recover control, performance

Pilot-scale setup



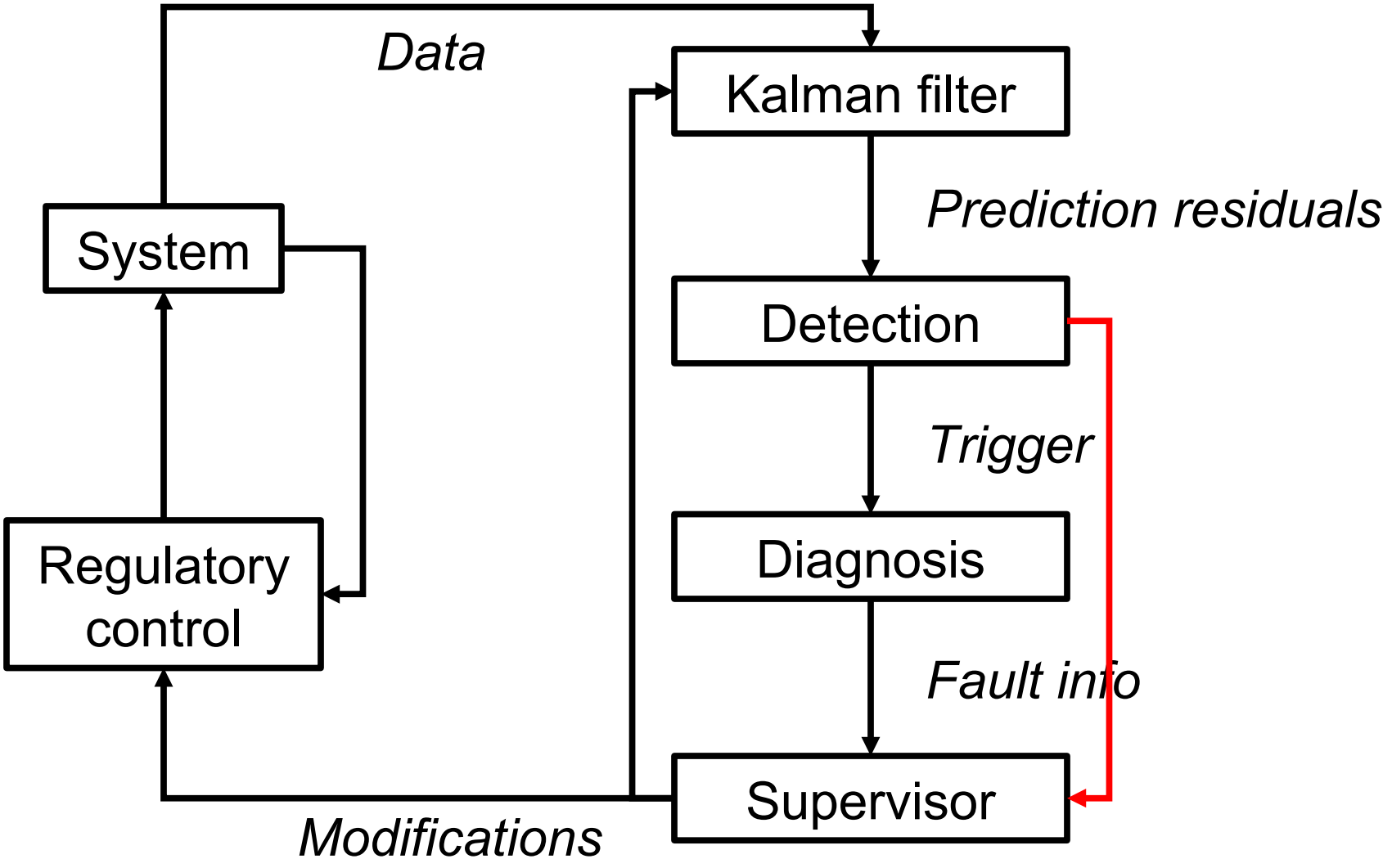
Pilot-scale setup



Problem statement

- Control problem
  - Make flow close to setpoint
  - With given valves
  - Which may be faulty/failing
- To solve:
  - Enable fault identification
  - Enable fault location/type
  - Accommodate by proper action

Method


Data
Kalman filter
Prediction residuals
System
Detection
Trigger
Regulatory control
Diagnosis
Fault info
Supervisor
Modifications

Method

- Detailed valve models needed for
  - Kalman filter
  - Diagnosis
- Valves are subject to
  - Delay
  - Hysteresis
- By design
  - Prevent blocking
  - Disconnect motor and vane axles reduces wear and tear
- Discrete input signals:
  - -1 / 0 / 1 : close / stay / open

Method

Kalman filter

Valve prediction:

$$x_{m,p} = x_{m,s} + s \cdot u$$

$$x_{i,p} = min\,[\,max\,(x_{m,p} - g_m,\, x_{i,s}),\, x_{m,p} + g_m\,)\,]$$

$$x_{v,p} = min\,[\,max\,(x_{i,p} - g_i,\, x_{v,s}),\, x_{i,p} + g_i\,)\,]$$

Valve update:

$$r_i = x_{i,p} - y_i$$

$$x_{m,s} = x_{m,p} - K \cdot r_i$$

$$x_{i,s} = x_{i,p} - K \cdot r_i$$

$$x_{v,s} = x_{v,p} - K \cdot r_i$$

Covariance pred/updates: assume input noise only at motor axle

Method

Fault detection:

$$\Sigma_{i,k} r_{i,k}^2 \geq \chi^2_{crit}$$

Fault diagnosis - fit different models

Bias:	$y_f = y_t + b$	$\mid t \geq t_f$
Drift:	$y_f = y_t + d \cdot (t - t_f)$	$\mid t \geq t_f$
Stuck:	$u_f = u_t(t-1)$	$\mid t \geq t_f$

Optimize parameters b / d and t_f for each location (valve) and fault type (max. likelihood)

→ b/d : analytical solution conditional to t_f

→ t_f : grid search […, t-10, t-9, …]

Select max. likelihood scenario

Method

Supervisor:

1. Fault-Tolerant Control
   - Bias (diagnosis) → correct signal
   - Drift (diagnosis) → correct signal
2. Reconfigurable control

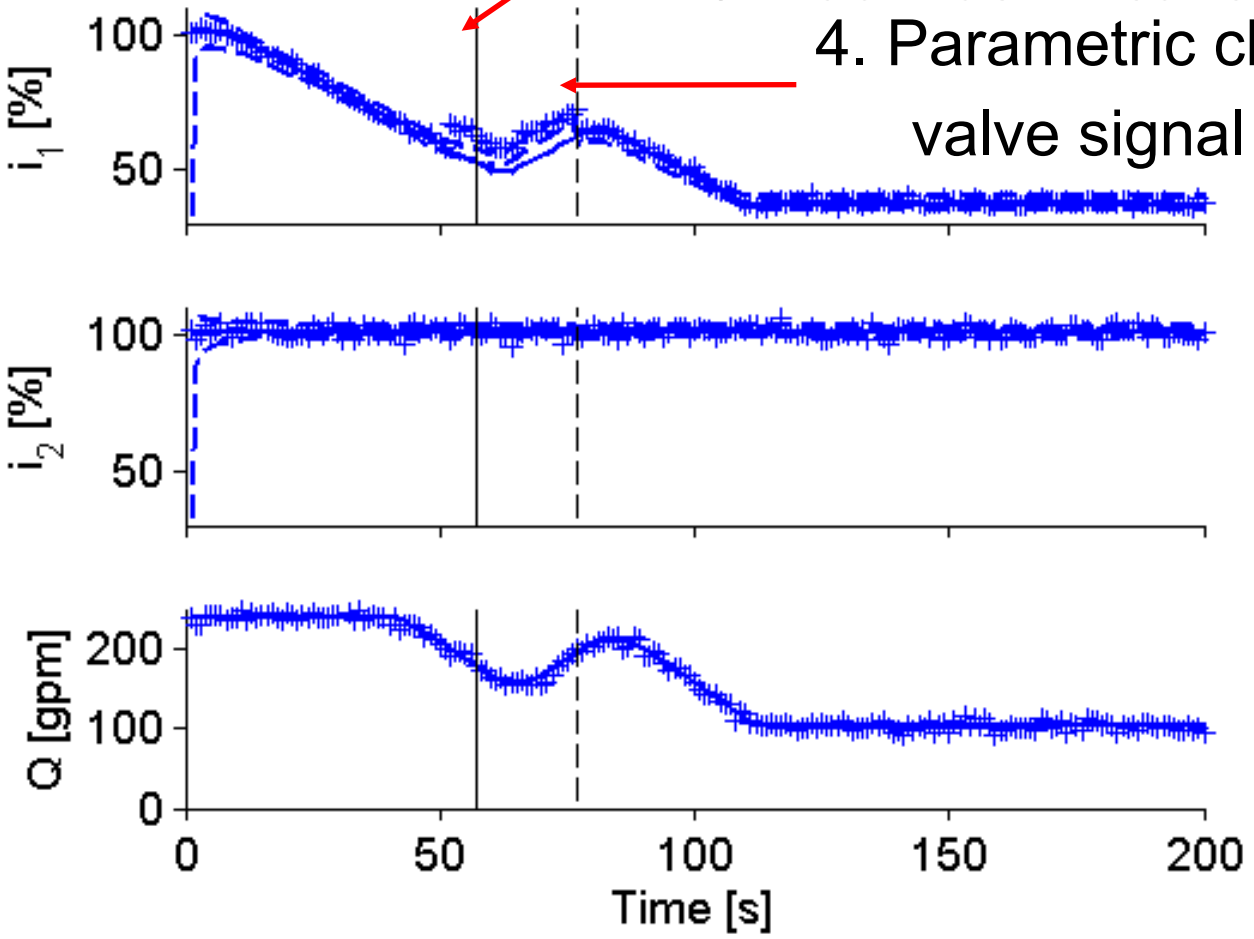
   Stuck valve (diagnosis) → best among available valves
3. Pro-active component

   Fault confirmation (no diagnosis yet) → open all valves

Typical result

1. Fault introduction
2. Fault confirmation
3. Fault identification
4. Parametric change:

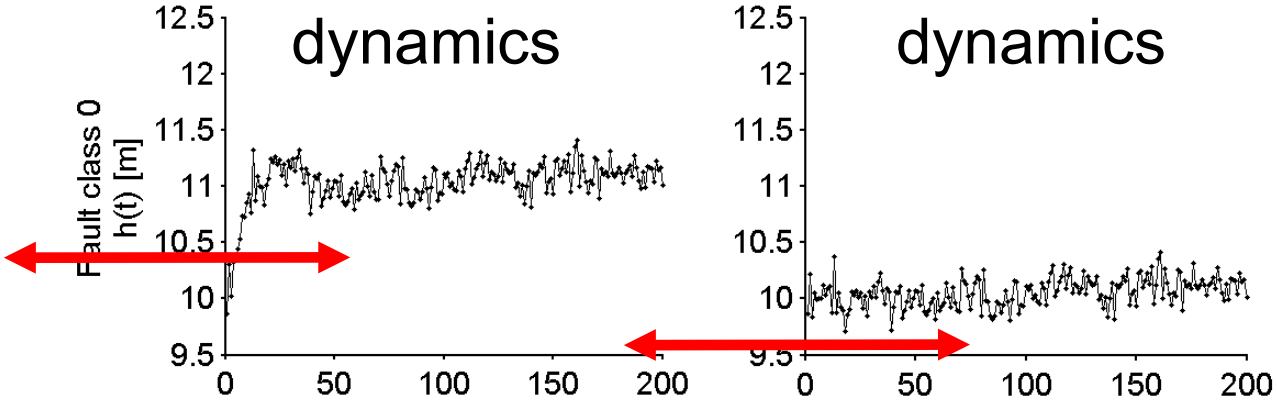
valve signal correction



Detection result

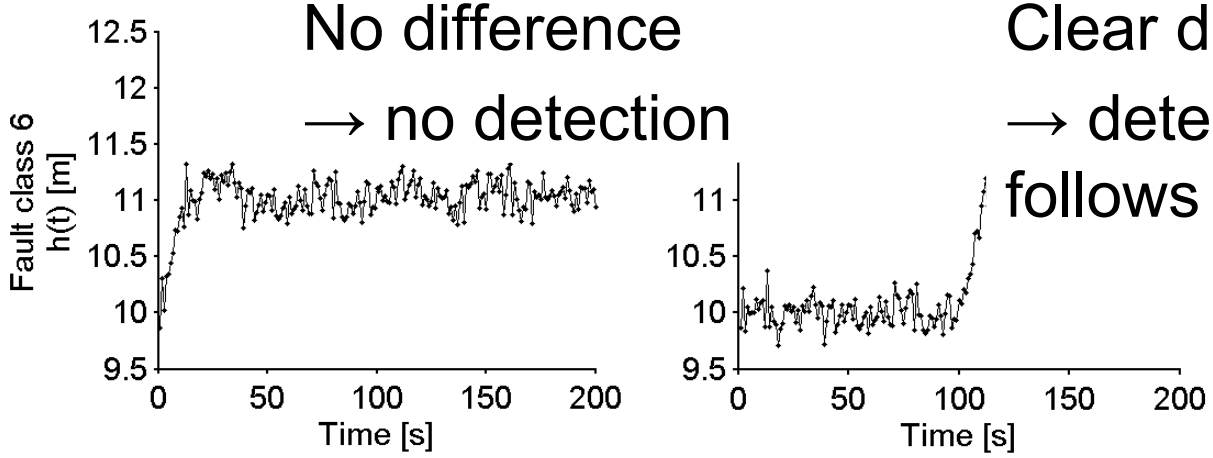
No control dynamics

With control dynamics

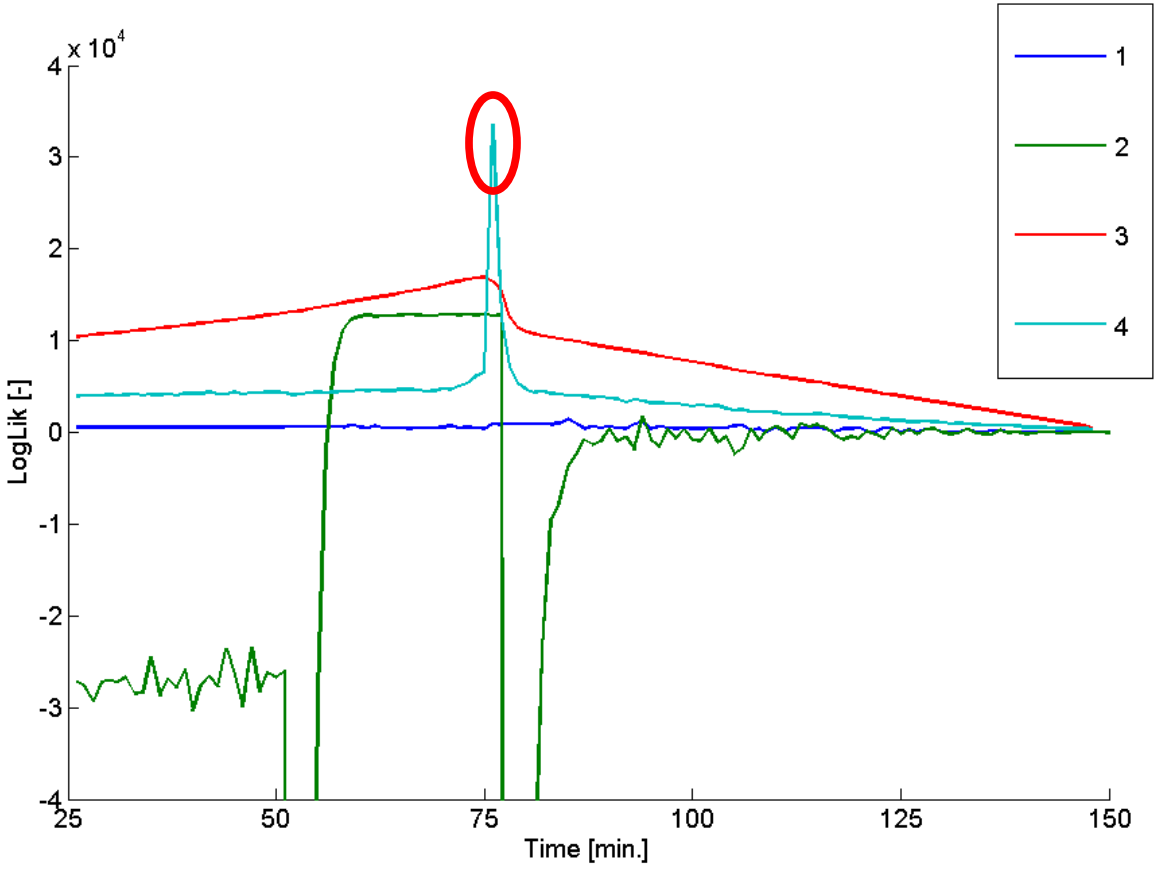


No difference → no detection

Clear difference → detection follows

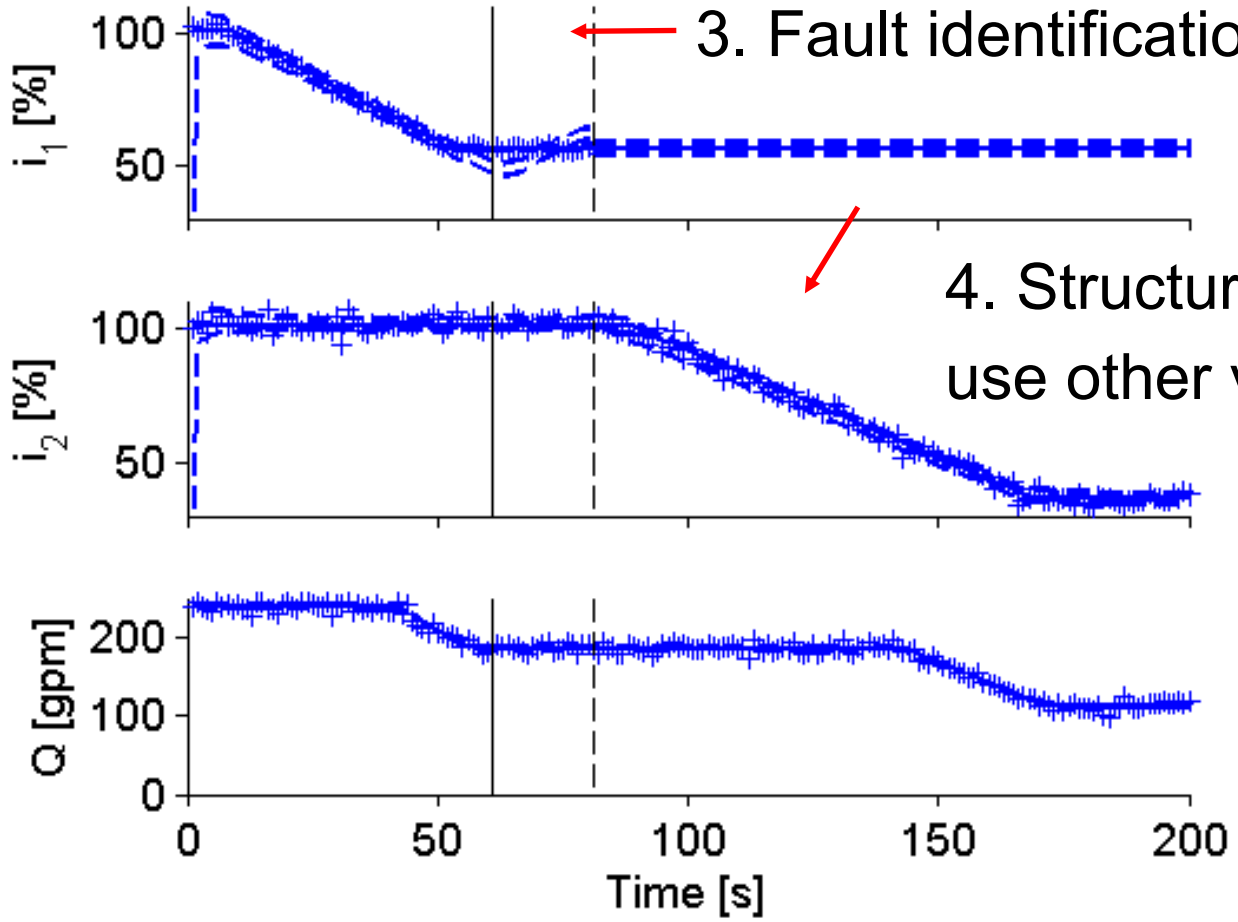


Diagnostic result



Typical result 2

1. Fault introduction
2. Fault confirmation
3. Fault identification
4. Structural change: use other valve



Additional observations

Method allows

- Multiple faults in one location
  - Additional parameters → uncertainty
  - No loss of degrees of freedom
- Single failure in each sensor/actuator
  - Loss of degrees of freedom

Conclusions and perspectives

- Reconfigurable control and fault-tolerant control
  - Reactive to current fault
  - Allow to retain degrees of freedom
- Active component in supervisory control
  - “Safety mode”
  - Generates information-rich data

Future:

- include “unknown” class of faults
- real-life testing (INL)
- larger systems