Automated Design of Fault Diagnosis Systems for Dependable Complex Systems

Carl Svärd, Scania AB, Södertälje
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- Case Study: Automotive Engine Fault Diagnosis
- Summary and Conclusions
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- Case Study: Automotive Engine
- Summary and Conclusions
Importance of Fault Diagnosis

- Low exhaust emissions
- High vehicle safety
- High vehicle uptime
- Efficient repair

- Regulations
- Vehicle Life-Cycle Cost
Fault diagnosis is essential for dependability!
Different Fault Diagnosis Activities

On-board diagnosis

On-board fault accommodation

Off-board diagnosis

- Uptime
- Emissions
- Safety
- Repair
- Fuel Economy
- Driveability

Legislative On-Board Diagnosis
Off-Board Diagnosis
On-Board Fault Accommodation
On-Board Fault Diagnosis

- Performed in ECUs as the vehicle operates on the road.
- OBD-legislations require detection and isolation of emission critical faults.
- ISO 26262 may pose similar requirements regarding safety critical faults.
Off-Board Fault Diagnosis

- Performed off-board by mechanic with additional computer support.
- Diagnosis combined with decision-theretic troubleshooting.
- Potentially gives better and more precise diagnosis results.
On-Board Fault Accommodation

- Performed on-board in ECUs.
- Aim is to prevent faults from developing into critical failures by taking appropriate actions.
- Typical action is reconfiguration of the control system.
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- Case Study: Automotive Engine Fault Diagnosis
- Summary and Conclusions
Common Fault Accommodation Strategy

- Centralized
- Many dependencies
- Non-modular
- Non-scalable

Not suitable for large, complex, systems!
Alternative Fault Accommodation Strategy

- De-Centralized
- Few dependencies
- Modular
- Scalable

[Diagram showing three systems labeled A, B, and C, each with blocks for Fault Detection and Isolation and Fault Accommodation.]
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- Case Study: Automotive Engine Fault Diagnosis
- Summary and Conclusions
Fault Diagnosis System
Detection Tests

- **Input**: observations (often subset)
- **Output**: binary detection result
- **Common approaches**:
  - Limit checking
  - Hardware redundancy
  - Residuals

\[ d_i = \tau_i (y) = T_i (R_i (y)) = \begin{cases} 1 & \text{if } \lambda_i (r_i) > J_i \\ 0 & \text{if } \lambda_i (r_i) \leq J_i \end{cases} \]
Fault Isolation

- Numerous approaches exist, e.g., AI, Bayesian, etc.
- Here: consistency based diagnosis and structured tests.

Tests: \[ \{\tau_1, \tau_2, \tau_3\} \]
Faults: \[ \{f_1, f_2, f_3\} \]

\[
\begin{array}{c|ccc}
\tau_1 & f_1 & f_2 & f_3 \\
\hline
\tau_1 & 1 & 1 & 1 \\
\tau_2 & 1 & 1 & 1 \\
\tau_3 & 1 & 1 & 1 \\
\end{array}
\]

\[
D_1 = \{f_2, f_3\} \quad D_2 = \{f_1, f_3\} \quad D_3 = \{f_1, f_2, f_3\}
\]

\[
D = D_1 \cap D_2 \cap D_3 = \{f_2, f_3\} \cap \{f_1, f_3\} \cap \{f_1, f_2, f_3\} = \{f_3\}
\]
Detection Tests Based on Residuals

- **Residual**: signal ideally zero in the no-fault case.

- **Residual generator**: creates residual using a model of the system and measurements.

- **Residual evaluator**: evaluates behavior of residual by means of thresholding of test statistic.
Residual

Test Statistic
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- Case Study: Automotive Engine Fault Diagnosis
- Summary and Conclusions
Design of Fault Diagnosis Systems

- Design of a complete diagnosis system is an intricate and complex task.

- Optimal solution requires substantial engineering effort, and well-defined requirements regarding, e.g.,
  - Robustness
  - Faults to diagnose
  - Behavior of system
Automated Design Methodology

- Supports continuous improvements
- Effective and systematic design
- Contributes to higher quality
Automated Design Methodology

- Requirements and Data
- Design of Residual Generators
- Residual Generators
- Design of Residual Evaluators
- Residual Evaluators
- Evaluation
- Data

Model
Design of Residual Generators

- Two-Step Approach:
  1) Create large set of candidate residual generators.
  2) Select and realize most suitable residual generators.

- The method used for realization, i.e., construction, of residual generators influences both steps.
Realization of Residual Generators

- Sequential residual generation
  1) Compute variables in the model by sequential computations.
  2) Evaluate redundant equation as residual.

- Why?
  - Can be automated to high extent.
  - Tractable and successful for complex models in real applications.

\[ \omega_t = \frac{P_t \eta_m - P_c}{J_t \omega_t} \]
\[ \dot{T}_{em} = \frac{R_e T_{em}}{\rho_{em} V_{em} c_{ve}} (W_{in} c_{ve} (T_{em,in} - T_{em}) + R_e (T_{em,in} W_{in} - T_{em} W_{out})) \]
\[ \dot{p}_{em} = \frac{R_e T_{em}}{V_{em}} (W_{co} - W_{egr} - W_t + \Delta W_{em}) + \frac{R_e}{V_{em} c_{ve}} (W_{in} c_{ve} (T_{em,in} - T_{em}) + R_c (T_{em,in} W_{in} - T_{em} W_{out})) \]
\[ P_{amb} = y_{p_{amb}} \]
\[ P_{bc} = P_{amb} \]
\[ x_{vgt} = u_{vgt} \]
\[ T_{em,in} = T_{amb} + (T_c - T_{amb}) \exp \left( -\frac{h_{tot} \pi d_{pipe} l_{pipe} n_{pipe}}{W_{co} c_{pc}} \right) \]
\[ W_{egr} = \frac{(\dot{p}_{im} V_{im} - R_s T_{im} W_{th} + W_{ei} R_t T_{im})}{R_s T_{im}} \]
\[ P_c = \frac{W_c c_{pa} T_{bc}}{\eta_c} \left( \Pi_c^{1-1/\gamma_a} - 1 \right) \]
\[ r = y_{p_{em}} - p_{em} \]
Candidate Residual Generators

- A candidate residual generator is a subset of the model from which a residual generator may be constructed.

- For construction of a sequential residual generator, a Minimal Structurally Over-determined (MSO) subset is necessary.

  An MSO set is a candidate sequential residual generator

- An MSO set corresponds to a minimal redundant subset of the model.
- There are efficient algorithms that finds all MSO sets in a large model.
Selecting Residual Generators

- Requirements for the selection:
  i. Single fault isolability.
  ii. As few residual generators as possible.

- Selection performed with greedy search:
  - Select the best candidate in each iteration until solution is complete.

- **Best**: The one which isolates most faults  
  - (cf. ii)

- **Complete**: All faults are isolable  
  - (cf. i)
Design of Residual Evaluators

- Residual evaluators based on a comparison of:
  - Current residual distribution, estimated online.
  - No-fault residual distribution, estimated off-line using no-fault training data.

- Comparison done with the Generalized Likelihood Ratio (GLR) Test.

- Diagnostic test constructed by thresholding the GLR test statistic.
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- **Case Study: Automotive Engine Fault Diagnosis**
- Summary and Conclusions
Aim of Case Study

- Use automated design methodology to design complete fault diagnosis system for an automotive diesel engine.
- No adaption of methods of tuning of obtained diagnosis system.
Automotive Engine System
Requirements on Diagnosis System

- Use existing hardware
- Detect and isolate small faults
- On-board implementation
- Robustness
- Systematic design
Some Characterizing Properties of System

- Few sensors
- Many operating modes
- Highly interconnected
- Complex model
Design Challenges

- Fault decoupling
- Model complexity
- Uncertainties
Handling of Design Challenges

- Model Complexity
- Fault Decoupling

- Uncertainties
Appliance of Design Methodology

- **Input:**
  - Model of engine
    - Non-linear DAE
    - 46 equations
  - Diagnosis requirement
    - 9 faults
  - No-fault data

- **Output:**
  - 8 residual generators
  - 8 residual evaluators
Obtained Residual Generators

- 8 residual generators (dynamic systems)
- Implemented in Matlab/Simulink
- Sensitive to different subset of faults
Obtained Residual Evaluators

- 8 residual evaluators
- Utilizes estimated probability distributions of the residuals
- Implemented in Matlab
Fault Isolation

- Residual generators sensitive to different faults
- Fault isolation done by matching residual responses with the fault signature matrix
Fault Detection Illustration
Fault Detection Illustration

No Response

Response

Response

Response
Outline

- Why Fault Diagnosis?
- Fault Accommodation
- Fault Diagnosis
- Automated Design of Fault Diagnosis Systems
- Case Study: Automotive Engine Fault Diagnosis
- Summary and Conclusions
Fault diagnosis essential for dependability of complex technical systems, e.g., automotive systems.

Design of diagnosis systems is an intricate and complex task. Optimal solution requires excessive knowledge.

An automated design methodology supports continuous improvements, is effective, systematic, and contributes to higher quality.

A brief overview of an automated methodology for design of fault diagnosis systems has been given. Its applicability has been illustrated by means of an automotive engine case study.
Questions?

Contact: carl.svard@scania.com