DESIGN OF BUILT-IN TESTS FOR ROBUST ACTIVE FAULT DETECTION AND ISOLATION OF DISCRETE FAULTS IN UNCERTAIN SYSTEMS

William T. Hale*, Kyle A. Palmer, Matthew D. Stuber, and George M. Bollas

*Ph.D. Candidate – Chemical & Biomolecular Engineering

School of Engineering – University of Connecticut

Phone: +1 (860) 486-6580
Email: william.hale@uconn.edu

American Control Conference, Milwaukee, WI, June 29, 2018
System Health Diagnostics in Safety-Critical Fields

Improving fault diagnostics leads to increased test rigor and guarantees to safety

Goal of system health diagnostics

- High accuracy
- Quick resolutions

Tradeoff exists between test complexity and allotted time

- Cost
- Safety
Active Model-Based Fault Detection and Isolation

Active model-based FDI methods provide accurate, low cost diagnostics

FDI methods

- Data-based
- Model-based

Passive FDI

- Analysis of system health during standard operation using measurements

Active FDI

- Incorporation of interruptive auxiliary input signals to improve system health analysis
Impact of Uncertainty on Maintenance

Costly no fault found, false alarm, and non-detection events occur frequently during maintenance

Uncertainty negatively impacts system health diagnostics

- Major cause of false alarms and no fault founds
- Increases cost and maintenance time and decreases safety

No fault found events (NFFs) increase the potential of inserting a faulty system back into operation

- 30-50% of LRUs removed for maintenance in the aerospace industry are tagged as NFF [1]–[3]
- Over 90% of aircraft electronics maintenance costs can be attributed to NFFs [4]

Problem: The absence of faults due to uncertainty during ill-designed maintenance tests is a main cause for NFFs [2], [5]

Global Optimization for Built-In Test Input Design

For safety-critical systems, the conservative approach for BIT design suffices

Goal: Develop a maintenance test (Built-In Test (BIT)) that produces unique system responses for a fault-free system and all of its fault scenarios even at its worst-case scenario of uncertainty

Method: Utilize global optimization techniques to solve a max-min program, reformulated as a semi-infinite program, involving the system inputs and uncertainty

The max-min approach is often considered to be sub-optimal due to its conservative nature of “raising the floor”, i.e. finding the best worst-case

However, for safety-critical systems with strict regulations such as in the aerospace industry, this approach is sufficient due to its guarantees
Mathematical Formulation

Model, output, max-min, and implicit function equations

Model equations: $f^{[f]}(\tilde{x}, u, \theta_p, \theta_u, \theta_f) = 0$, $\forall f \in \{0, 1, ..., N_f\}$ (1)

Output equations: $y^{[f]} = \tilde{x} + w$ (2)

Max-min program: $\max_{u \in U} \min_{\theta_u \in \Theta_u, \theta_f \in \Theta_f} G(\tilde{x}, u, \theta_p, \theta_u, \theta_f)$ (3)

s.t. $f(\tilde{x}, u, \theta_p, \theta_u, \theta_f) = (f^{[1]}, f^{[2]}, ..., f^{[N_f]}) = 0$

Implicit function: $\tilde{x} = x(u, \theta_p, \theta_u, \theta_f)$ exists such that $f(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) = 0$, $\forall (u, \theta_p, \theta_u, \theta_f) \in U \times \Theta_p \times \Theta_u \times \Theta_f$

is satisfied

$[f]$ : fault scenario
$f^{[f]}$ : governing equations
$\tilde{x} \in \tilde{X} \subset \mathbb{R}^{N_x}$ : system states
$u \in U \subset \mathbb{R}^{N_u}$ : inputs
$\theta_p \in \Theta_p \subset \mathbb{R}^{N_{\theta_p}}$ : design parameters
$\theta_u \in \Theta_u \subset \mathbb{R}^{N_{\theta_u}}$ : uncertain parameters
$\theta_f \in \Theta_f \subset \mathbb{R}^{N_{\theta_f}}$ : fault parameters
$y^{[f]} \in Y \subset \mathbb{R}^{N_y}$ : outputs
$\tilde{x} \in \tilde{X} \subset \mathbb{R}^{N_y}$ : measured states
$w \in W \subset \mathbb{R}^{N_y}$ : noise
$f$ : combined equations
$G$ : objective function
$x$ : implicit function
Mathematical Formulation

Extensive SIP, feasibility criterion, and WCD SIP

Extensive semi-infinite program (SIP):

\[
\min_{u \in U, \eta \in H} -\eta
\]

s.t. \( g(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f, \eta) = \eta - G(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) \leq 0 \) \hspace{1cm} (4)

\( \forall (\theta_u, \theta_f) \in \Theta_u \times \Theta_f \)

Feasibility criterion: \( g(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f, \eta) = \eta + \eta_{feas} - \sum_{i=1}^{N_y} \sum_{f=0}^{N_f-1} \sum_{g=f+1}^{N_f} (y_i^{[f]} - y_i^{[g]})^2 \) \hspace{1cm} (5)

Worst-case BIT design (WCD) SIP:

\[
\min_{u \in U, \eta \in H} -\eta
\]

s.t. \( \eta + \eta_{feas} - \sum_{i=1}^{N_y} \sum_{f=0}^{N_f-1} \sum_{g=f+1}^{N_f} (y_i^{[f]} - y_i^{[g]})^2 \leq 0 \) \hspace{1cm} (6)

\( \forall (\theta_u, \theta_f) \in \Theta_u \times \Theta_f \)

\( \eta \in H \subset \mathbb{R} \) : SIP auxiliary variable

\( g \) : Feasibility criterion

\( \eta_{feas} \) : BIT feasibility parameter

\( N_f \) : number of faults

\( N_y \) : number of outputs

\( y_i^{[f], [g]} \) : \( i^{th} \) output of fault scenarios \( f, g \)
Worst-case BIT design algorithm [6]–[8]

- Initialize uncertainty
- Set iteration count to 1
- Begin iteration
  - Solve outer program for BIT design, analyzing all previous uncertainty sets
  - Solve inner program at BIT design, for updated worst-case uncertainty set
  - Update iteration count
- Examine continuation criteria
  - If true, begin next iteration
  - If false, end algorithm, worst-case design found

Algorithm 1 SIP Max-Min Algorithm

Require: $\theta_u^{[1]} \in \Theta_u, \theta_f^{[1]} \in \Theta_f$

1. $K \leftarrow 1$
2. while $\hat{\eta}^{[K]} < \eta^{[K]} \land K \leq K_{max}$ do
3. \hspace{1cm} $(\eta^{[K]}, u^{[K]}) \leftarrow \min_{u \in U, \eta \in H} -\eta$
4. \hspace{1.5cm} s.t. $\eta - G(x(u, \theta_p, \theta_u^{[K]}, \theta_f^{[K]}), u, \theta_p, \theta_u^{[K]}, \theta_f^{[K]}) \leq 0,$
5. \hspace{1.5cm} $\forall k \in \{1, 2, ..., K\}$
6. \hspace{1.5cm} $(\hat{\eta}^{[K]}, \theta_u^{[K+1]}, \theta_f^{[K+1]}) \leftarrow \min_{\theta_u \in \Theta_u, \theta_f \in \Theta_f} G(x(u^{[K]}, \theta_p, \theta_u, \theta_f), u^{[K]}, \theta_p, \theta_u, \theta_f)$
7. $K \leftarrow K + 1$
8. end

$(u^{opt}, \theta_u^{opt}, \theta_f^{opt}) \leftarrow (u^{[K-1]}, \theta_u^{[K]}, \theta_f^{[K]})$

References:

Case Study: Three Tank System Description

Three tank system is a benchmark for FDI

Faults Studied: $\Theta_f \in \Theta_f$
- Pump 1 - Failure
- Tank 2 - Leak

Inputs:
- Pump 1 - Flow Rate $u_1$
- Pump 2 - Flow Rate $u_2$

Outputs:
- Tank 1 - Level $\tilde{x}_1$
- Tank 2 - Level $\tilde{x}_2$
- Tank 3 - Level $\tilde{x}_3$

Uncertainties Present: $\Theta_a \in \Theta_a$
- Valve 13 - Outflow Coefficient
- Valve 32 - Outflow Coefficient
- Valve 20 - Outflow Coefficient

Design/Model Parameters:
- Tank Cross-Sec Area $A_T$
- Pipe Cross-Sec Area $A_P$
- Particle Cross-Sec Area $A_P$
- Gravitational Acceleration Constant $g = 9.80665\, m/s^2$
- Pi $\pi = 3.14159265358979$
- Tank Height $x_{\text{max}} = 0.75\, m$

\[
f(\mathbf{u}, \mathbf{u}_0, \alpha, \beta) = \begin{bmatrix} f_1(u), f_2(u), f_3(u) \end{bmatrix} = 0 = \Delta x_{i_j} = \tilde{x}_{i_j} - \tilde{x}_{j_i}
\]

\[
\begin{align*}
- c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + u_1 \\
- c_2 \alpha & \cdot \text{tanh}(k_2 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + c_2 \alpha \cdot \text{tanh}(k_2 \tilde{x}_2) \sqrt{2g \Delta x_{13}^2} + e + u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
- c_2 \alpha & \cdot \text{tanh}(k_2 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + c_2 \alpha \cdot \text{tanh}(k_2 \tilde{x}_2) \sqrt{2g \Delta x_{13}^2} + e + u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
- c_2 \alpha & \cdot \text{tanh}(k_2 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + c_2 \alpha \cdot \text{tanh}(k_2 \tilde{x}_2) \sqrt{2g \Delta x_{13}^2} + e + u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
- c_2 \alpha & \cdot \text{tanh}(k_2 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + c_2 \alpha \cdot \text{tanh}(k_2 \tilde{x}_2) \sqrt{2g \Delta x_{13}^2} + e + u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
- c_2 \alpha & \cdot \text{tanh}(k_2 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + c_2 \alpha \cdot \text{tanh}(k_2 \tilde{x}_2) \sqrt{2g \Delta x_{13}^2} + e + u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
+ c_1 \alpha & \cdot \text{tanh}(k_1 \tilde{x}_1) \sqrt{2g \Delta x_{13}^2} + e + (\alpha - 1) u_1 \\
\end{align*}
\]
Case Study: Different BIT Designs

Four different operating conditions were analyzed for BIT effectiveness

BIT designs

- Nominal
- Mean
- Conservative
- Worst-case

<table>
<thead>
<tr>
<th>Inputs $(m^3s^{-1} \times 10^{-4})$</th>
<th>$u_1$</th>
<th>$u_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>Nominal</td>
<td>0.41</td>
<td>0.41</td>
</tr>
<tr>
<td>Mean</td>
<td>0.97</td>
<td>0.10</td>
</tr>
<tr>
<td>Conservative</td>
<td>0.70</td>
<td>0.10</td>
</tr>
<tr>
<td>Worst-case</td>
<td>0.80</td>
<td>0.10</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Case Study: Objective Function Surface

WCD lies on the intersection of the tank height constraint and the objective function $G$

\[
G(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) = -n_{feas} + \sum_{i=1}^{N_y} \sum_{f=0}^{N_f-1} \sum_{g=f+1}^{N_f} (y_i^f - y_i^g)^2
\]
Separation of anticipated outputs:

\[
G(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) = -n_{feas} + \sum_{i=1}^{N_i} \sum_{f=0}^{N_f-1} \sum_{g=f+1}^{N_f} (\bar{y}_i^{[f]} - \bar{y}_i^{[g]})^2
= -0.1 + 0.14 = 0.04
\]
Case Study: Mean BIT Design

Improved separation, but violates constraints for numerous cases of uncertainty

Three Tank System Mean BIT Design

Uncertain Scenarios
- Fault-Free
- Fault 1
- Fault 2
- Fault 3

Separation of anticipated outputs:

\[
G(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) = \]

\[
-\eta_{feas} + \sum_{i=1}^{N_f} \sum_{f=0}^{N_f-1} \sum_{g=f+1}^{N_f} (\bar{y}_i^{[f]} - \bar{y}_i^{[g]})^2
\]

\[
= -0.1 + 1.33 = 1.23
\]
Case Study: Mean BIT Design w/ Conservative Constraint

Manages tank height constraint violations but results in underperformance of separation

Three Tank System Conservative Mean BIT Design

Separation of anticipated outputs:

\[ G(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) = \]

\[ -\eta_{feas} + \sum_{i=1}^{N_f} \sum_{f=0}^{N_f-1} \sum_{g=f+1}^{N_f} (\bar{y}^{[f]}_i - \bar{y}^{[g]}_i)^2 \]

\[ = -0.1 + 0.38 = 0.28 \]
Case Study: Worst-Case BIT Design
Maximizes separation and maintains constraint feasibility for all uncertainty scenarios

Separation of anticipated outputs:

\[
G(x(u, \theta_p, \theta_u, \theta_f), u, \theta_p, \theta_u, \theta_f) =
-\eta_{feas} + \sum_{i=1}^{N_y} \sum_{f=0}^{N_f} \sum_{g=f+1}^{N_f} (\bar{y}_i^{[f]} - \bar{y}_i^{[g]})^2
= -0.1 + 0.63 = 0.53
\]
Conclusions

Method developed aims at improving fault detection and isolation at the worst-case scenario(s) of uncertainty

BIT design at the worst-case scenario of uncertainty shows improvement in output separation in comparison to the nominal, mean, and conservative mean BIT designs

Global feasibility provided, guaranteeing robustness of the BIT design which is important for safety-critical systems
This work was sponsored by the United Technologies Corporation Institute for Advanced Systems Engineering (UTC-IASE) of the University of Connecticut. Any opinions expressed herein are those of the authors and do not represent those of the sponsor.

The authors would like to express their gratitude to Matthew Wilhelm for his help with obtaining a global solution in the Julia programming language using the developed package Easy-Advanced Global Optimization (EAGO) [9].

In-collaboration with:

Process Systems and Operations Research Laboratory

Web: https://psoruconn.edu/