MIXED-INTEGER NON-LINEAR PROGRAMS FOR OPTIMAL DETECTION AND ISOLATION OF FAULTS

William T. Hale[1], Kyle A. Palmer[1] and George M. Bollas*[1]

[1]Department of Chemical & Biomolecular Engineering, University of Connecticut, 191 Auditorium Road, Unit 3222, Storrs, CT, 06269-3222, USA

Abstract

A model-based method for detecting and isolating discrete actuator and sensor faults is proposed. The application of the presented method is on the design of software-based maintenance procedures referred to as built-in tests (BITs). These BITs often occur outside of normal system operation and allow wider system input variability. This provides an opportunity to design optimal BITs that maximize the capability to detect and isolate faults. First, a general mathematical framework for fault detection and isolation through BIT design is presented. Next, the proposed formulation is applied to the design of optimal BITs for fault detection and isolation of faults common to gas compression systems. We conclude by presenting the benefits of the proposed method in detecting and isolating faults, as a solution to a constrained optimization problem with the system admissible inputs as manipulated variables.

Keywords

fault detection, fault isolation, process optimization, industrial applications

Introduction

The objective of fault diagnostics is to improve system accessibility, reliability and safety while alleviating system maintenance costs. Over the past few decades, the increase in system complexity has created a need for improvements in fault diagnostics to mitigate the issues caused by system non-linearity, heterogeneity and sensor availability limitations. Fault detection and isolation (FDI) methods are critical in the design of modern systems and the selection of their operating conditions. Therefore, significant effort has been devoted to improving the fault diagnostic and prognostic capabilities in the aerospace, automotive and energy industries (Venkatasubramanian et al., 2003; Isermann, 2005; Hwang et al., 2010; Willsky, 1976).

The ability to detect faults, the rate of false alarms and the frequency of events called “No Fault Found” (NFF) are important in FDI. NFF events occur when a fault is detected during operation, but not precisely identified at subsequent levels of testing (Soderholm, 2007). Khan et al. (2014b,a) attributes NFFs to incorrect system diagnostics and inaccurate replication of field conditions during maintenance. It is therefore obvious that the detection and isolation of faults often depends on system conditions. If a maintenance test is poorly designed then fault symptoms are likely to be absent or false.

The fault classes common to industrial systems are grouped as process, actuator and sensor faults. The most difficult faults to diagnose occur slowly over time or intermittently, leading to ineffective FDI. Khan et al. (2014b) identified a need for improvement in FDI, especially with regards to built-in tests (BITs). BITs are techniques that integrate and automate methods of FDI in system operations. The selection of a BIT design can significantly impact the isolation of faults from system uncertainty, improving maintenance costs, system reliability and safety (Venkatasubramanian et al., 2003).

FDI methods can be passively or actively integrated into system fault diagnostics. Passive FDI methods detect irregularities from a predicted or anticipated behavior that occurs during standard operations. Many ap-

*To whom all correspondence should be addressed *george.bollas@uconn.edu*

proaches have been proposed over the years to improve passive FDI methods (Isermann, 1984, 2005; Hwang et al., 2010; Willsky, 1976; Chow and Willsky, 1984). However, the robustness of these approaches depends significantly on system conditions and the severity and complexity of the fault(s) (Esna Ashari et al., 2012). Active methods employ system reconfiguration to reach an "optimal" state or trajectory that can better detect and isolate faults. Active FDI methods have gained attention in the last decade (Esna Ashari et al., 2012; Niemann, 2006; Punčochář et al., 2015). A typical approach to implementing active FDI methods is through the use of a BIT. Active and passive FDI can be used in a BIT through the use of analytical (model-based) redundancy. Analytical redundancy uses prior system knowledge to develop model equations that represent the anticipated system behavior over a specified range of operating conditions. These models can be used when selecting the BIT design to determine what the best operating conditions are for FDI. When selecting a feasible BIT design, the assessment must consider the effect of system noise, unknown inputs and model error on the robustness of the BIT.

The optimization of FDI design methods has recently become a focus of interest in research (Zolghadri, 2012; Kim et al., 2013; Yin et al., 2014; Rosich et al., 2007). Typically, these methods detect system faults by designing optimal and robust filters, thresholds and sensor placement. In this work, an active FDI method is proposed that optimally isolates faults. The focus is on developing an approach that improves BIT robustness, by optimizing the set of admissible inputs to improve detection and isolation. The BIT is formulated as a constrained optimization problem that handles the controllable system inputs as manipulated variables in order to generate the best feasible set of conditions that correspond to a uniquely measurable system response, in response to a fault or set of faults. This method is applied to a gas compression system with non-directly measurable faults. Finally, the conclusions of the work and considerations made for future work are discussed.

Mathematical Formulation

A mathematical description of the proposed model-based FDI method is expressed as a set of steady state algebraic equations that describe the system:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}_p, \boldsymbol{\theta}_f) = 0, \tag{1}$$

where $\mathbf{f}$ is the set of governing equations, $\mathbf{x} \in \mathbb{R}^n$ is the $n \times 1$ state vector, $\mathbf{u} \in \mathbb{R}^r$ is the $r \times 1$ vector of manipulated inputs, $\boldsymbol{\theta}_p \in \mathbb{R}^p$ is the $p \times 1$ vector of design parameters and $\boldsymbol{\theta}_f \in \mathbb{R}^q$ is the $q \times 1$ vector of fault parameters. The system faults F^* are described mathematically by their respective fault parameter(s), $\boldsymbol{\theta}_f$. The system measurements are expressed as:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}_p, \boldsymbol{\theta}_f), \tag{2}$$

where $\mathbf{h}$ is the set of equations that map the $m \times 1$ output vector, $\mathbf{y} \in \mathbb{R}^m$, to system states, inputs and parameters. The model equations of Eq. (1) and (2) represent a clean nominal system and/or a faulty "virtual system", which have identical structures, but different values for their fault parameters, $\boldsymbol{\theta}_f$. The fault parameters of the nominal system model, $\boldsymbol{\theta}_f^0$, have values that represent clean system conditions (fault-free) and the fault parameters of the "virtual system" model, $\boldsymbol{\theta}_f^*$, have values that represent faulty system conditions of F^*. Normally distributed white measurement noise, $\mathbf{w}(n_s) \in \mathbb{R}^m$, is also applied to the "virtual system" model measurements at each n_s sampling point, as shown below:

$$\mathbf{y}^*(n_s) = \mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}_p, \boldsymbol{\theta}_f^*) + \mathbf{w}(n_s), \tag{3}$$

where $\mathbf{y}^*(n_s) \in \mathbb{R}^m$ is the vector of virtual system measurements. These virtual system measurements can be compared to anticipated trajectories of a clean system, $\mathbf{y}$, using a statistical function over all sampling points to generate FDI features, which can be used to identify faulty symptoms:

$$\mathbf{s}^* = \phi(\mathbf{y}, \mathbf{y}^*(n_s)), \tag{4}$$

where $\mathbf{s}^* \in \mathbb{R}^m$ is the $m \times 1$ vector of features and $\boldsymbol{\phi}$ is the statistical function used to identify deviations from nominal behavior. In this work, the maximum absolute value of the residuals over the BIT duration was assumed a sufficient metric for $\boldsymbol{\phi}$ in Eq. (4) to develop the features, $\mathbf{s}^* = \max\limits_{1 \leq n_s \leq N_s} |\mathbf{y}^*(n_s) - \mathbf{y}(n_s)|$. The anticipated clean system measurements $\mathbf{y}$ are duplicated to match the size of the "virtual system" noisy measurements when developing features, shown above as $\mathbf{y}(n_s)$. More robust statistical analyses as described by Joe Qin (2003) can be applied when defining $\boldsymbol{\phi}$ in the presence of model uncertainty, measurement noise and input disturbance without affecting the functionality of the method presented here. The individual elements of the features vector $\mathbf{s}^*$ are compared to their respective elements of the $m \times 1$ design threshold vector $\boldsymbol{\lambda} \in \mathbb{R}^m$ to create a

$m \times 1$ vector of symptoms $\mathbf{S} \in \mathbb{R}^m$:

$$\mathbf{S} = \begin{cases} 0, & s_i^* \leq \lambda_i \quad \text{(no symptom)}, \\ 1, & s_i^* > \lambda_i \quad \text{(symptom)}, \end{cases} \forall i = 1, ..., m \tag{5}$$

where s_i^* is the feature and λ_i is the threshold corresponding to sensor i. The comparison of the "virtual system" responses to the clean system responses forms a symptoms vector that describes the health status of a system and can be used to detect and isolate the faults present:

$$F^* = \begin{cases} 0, & \text{if } \mathbf{S} = 0 \quad \text{(no fault detected)}, \\ 1, & \text{if } \mathbf{S} - \mathbf{S}_{ref}^* = 0 \quad \text{(fault isolated)}, \\ !, & \text{if } \mathbf{S} - \mathbf{S}_{ref}^* \neq 0 \wedge \mathbf{S} \neq 0 \quad \text{(unknown)}, \end{cases} \tag{6}$$

where $\mathbf{S}_{ref}^*$ is the reference symptoms vector that corresponds to fault F^* and "!" expresses the possibility of a fault being detected from a non-zero symptoms vector (i.e., $\mathbf{S} = [1, 0, 0, 0, 0]^T$), without matching the reference symptoms vector of any fault considered in the preliminary analysis, and is therefore not isolated. The challenge after a fault is detected is to isolate it. The method of isolation considered here, is to use the nominal and "virtual system" models to create an $m \times N_f$-dimensional (number of sensors by number of faults) reference symptoms map that describes the anticipated behavior/symptom(s) of each fault or set of faults for a given set of inputs. Fault isolation then translates to the activity of symptom matching, where the symptoms vector of the unknown system $\mathbf{S}$ is compared to the reference symptoms vector $\mathbf{S}_{ref}^*$ of F^* to determine if the fault is present. This activity is executed for every fault in the reference symptoms map.

The design of BIT is the determination of system maintenance test procedures that examine the health of the system, and detect and isolate the present fault(s). When designing BITs, there are numerous options available to tune the test such as: the number of tests (sets of inputs, N_{test}); the duration of these tests (τ); their dynamic responses; the frequency and type of sensors used; the statistical functions and thresholds used to analyze the system features; and the admissible inputs. To optimally design a BIT for a system, given the flexibility in tuning options, a mathematical formulation that maximizes the number of unique symptoms vectors in the reference symptoms map can be constructed using any or all of these tuning options as the manipulated variable(s). The simplest formulation of BIT as an optimization problem is shown below, using only the admissible inputs, $\mathbf{u}$, as the manipulated variables. The number of tests N_{test} was also included in the formulation, as it allows complete fault isolation, as discussed later:

$$\begin{aligned}
&\max_{N_{test}, \mathbf{U}=[\mathbf{u}_1, \ldots, \mathbf{u}_{N_{test}}]} Q = \text{rowsize}(\text{unique}(\mathbf{S}_{ref})) \\
&\text{s.t.} \\
&\forall j = 1, ..., N_f, \quad \forall k = 1, ..., N_{test}: \\
&\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \theta_p, \theta_f^0) = 0, \\
&\mathbf{y}_k^0 = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \theta_p, \theta_f^0), \\
&\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \theta_p, \theta_f^j) = 0, \\
&\mathbf{y}_k^j(n_s) = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \theta_p, \theta_f^j) + \mathbf{w}(n_s), \\
&\mathbf{s}_k^j = \max_{1 \leq n_s \leq N_s} |\mathbf{y}_k^j(n_s) - \mathbf{y}_k^0(n_s)|, \\
&\mathbf{S}_{ref,k}^j = \begin{cases} 0, & s_{i,k}^j \leq \lambda_i \\ 1, & s_{i,k}^j > \lambda_i \end{cases} \quad \forall i = 1, ..., m, \\
&\mathbf{S}_{ref,k} = [\mathbf{S}_{ref,k}^1, ..., \mathbf{S}_{ref,k}^{N_f}] = \begin{bmatrix} S_{ref,1,k}^1 & \cdots & S_{ref,1,k}^{N_f} \\ \vdots & \ddots & \vdots \\ S_{ref,m,k}^1 & \cdots & S_{ref,m,k}^{N_f} \end{bmatrix}, \\
&\mathbf{S}_{ref} = [\mathbf{S}_{ref,1}; \ldots; \mathbf{S}_{ref,N_{test}}], \\
&\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \in \mathbb{R}^n, \\
&\mathbf{u}^L \leq \mathbf{u}_k \leq \mathbf{u}^U \in \mathbb{R}^r, \\
&N_{test} > 0 \in \mathbb{Z},
\end{aligned} \tag{7}$$

where Q is the number of unique symptoms vectors in the designed reference symptoms map and $\mathbf{U}$ is the matrix of admissible inputs. The number of unique symptoms vectors is calculated using the function "unique", that forms a new matrix of all the unique columns in $\mathbf{S}_{ref}$, of which the row size is maximized. The aim of Eq. (7) is to develop the largest feasible reference symptoms map that contains unique columns. This is done by solving for the set of admissible inputs that creates the largest number of unique symptoms vectors for each fault in the reference symptoms map. The BIT formulation of Eq. (7) is subject to the set of governing equations and nominal fault parameters, θ_f^0, that represent the clean system and its respective anticipated system measurements, $\mathbf{y}_k^0$, the set of governing equations and fault parameters, θ_f^j, that represent the faulty virtual system and its respective noisy system measurements, $\mathbf{y}_k^j(n_s)$, the FDI features, $\mathbf{s}_k^j$, the design thresholds, λ_i, and the state and input bounds. These constraints are applied for all sampling points, N_s, measurements, m, faults, N_f and input sets, N_{test}. Although the number of input sets is not explicitly used as an optimization variable, it is capable of being one and defines the complexity of the BIT design.

Example System Application

The system modeled and studied in this work is a variable speed/geometry compressor with a recycle valve to protect the compressor from surge (Agarwal, 1984; McAuliffe and Beers, 2006; Batson and Narayanan, 1995). The admissible inputs to the system are the valve position, compressor rotational speed, variable diffuser (VD) position and system outlet pressure. The measured variables consist of the compressor inlet pressure, compressor outlet pressure, compressor outlet temperature and system mass flow rate. There is also a compressor surge margin calculation, used to gauge safe operation, that is a function of the compressor inlet pressure, compressor outlet pressure and system mass flow rate. A physics-based model of the system was developed and is shown in Fig. 1, described in greater detail in Hale and Bollas (2016). The system was modeled using the object-oriented equation-based language Modelica® (Modelica Association, 2010) within the dynamic modeling and simulation software environment of Dymola® (Dassault Systèmes, 2016).

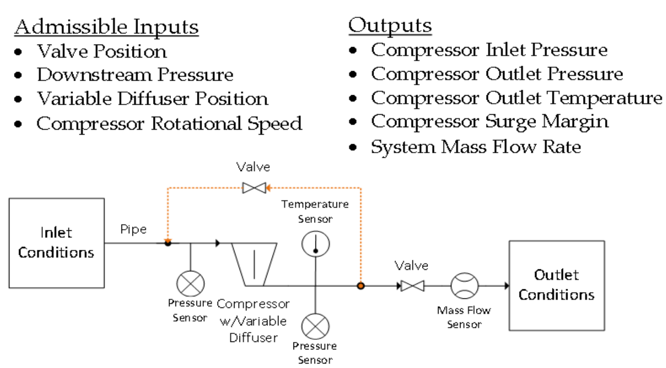

Figure 1. Diagram of the gas compression system studied in this work with the system admissible inputs and outputs identified.

The inlet conditions of the system of Fig. 1 were held constant at standard atmospheric conditions (288.5 K and 101325 Pa) for the BIT design. Normally distributed white noise, determined from the uncertainty and noise levels of the system sensors, was added to the "virtual system" model with standard deviations of $2.5K$, $50Pa$ and $0.01kg/s$ for the temperature, pressure and mass flow sensors, respectively. Eight faults frequent to this particular system and their respective fault parameters are shown in Table 1. Each of these faults was injected into the "virtual system" model and assumed to be constant for the duration of the BIT. The first 7 faults, $F_1 - F_7$, were analyzed individually. The final fault, F_8, was analyzed as a combination of the individual faults, F_3, F_4, F_7. In this first-level analysis only one combination of faults was studied; however, other combinations can be included without fundamental restrictions.

Table 1. Description of the faults injected into the gas compression system model of Fig 1.

Faults	Altered Parameter Values
F_1: Valve Stuck Closed	Valve Position, $\theta_{valve} = 0.0$
F_2: Valve Stuck Mid	Valve Position, $\theta_{valve} = 0.5$
F_3: Valve Stuck Open	Valve Position, $\theta_{valve} = 1.0$
F_4: VD Stuck Closed	VD Position, $\theta_{VD} = 0.0$
F_5: VD Stuck Mid	VD Position, $\theta_{VD} = 0.5$
F_6: VD Stuck Open	VD Position, $\theta_{VD} = 1.0$
F_7: Mass Flow Sensor Bias	Percent Bias, $\theta_{bias} = 1.15$
F_8: Faults F_3, F_4 & F_7	See Above

The optimization problem was refined by selecting appropriate system input constraints to Eq. (7) to solve for the optimal BIT design. The selected input constraints reflect the physical and design limitations of the studied system. The Modelica model equations were exported to MATLAB® (The Mathworks Inc., 2013) as a Functional Mockup Unit (FMU) using the Functional Mockup Interface (FMI)d. The Modelon® FMI-Toolbox (Modelon AB, 2014) was used to import the FMU into MATLAB® where the optimal BIT design was calculated with MATLAB® OPTI Toolbox and Mesh Adaptive Direct Search algorithm (NOMAD). NOMAD is used to solve for the global, non-differentiable, mixed integer nonlinear programming problem presented in this work (Le Digabel, 2011).

Design of BIT for a Gas Compression System

In this section, the equations presented earlier are applied to the gas compression system described the previous section to improve the detection and isolation of discrete faults. The BIT design of Eq. (7) is assessed in its capability to improve system FDI, described in this paper as the uniqueness of the reference symptoms map. The optimal solution to the BIT design Eq. (7) is shown in Table 2, with respective input constraints.

In this work, we analyzed BIT design optimizations for several cases of N_{test} ($N_{test} = 1, 2, 3$). Given the

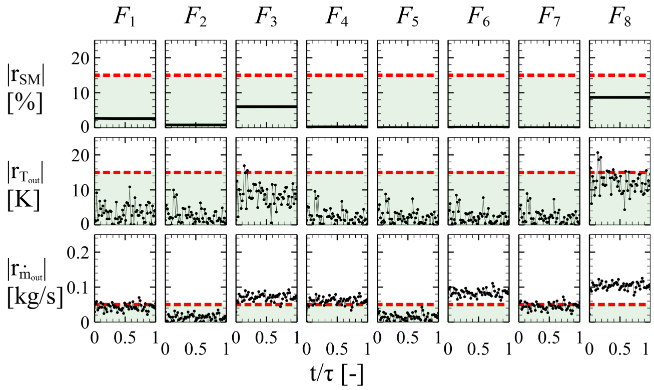

Figure 2. Steady state plots of the absolute value residuals between the "virtual system" and clean model measurements for each fault at the first optimal set of inputs from Table 2, $\mathbf{u}_1 = [0.4, 1.55e4, 0.4, 1.21e5]^T$.

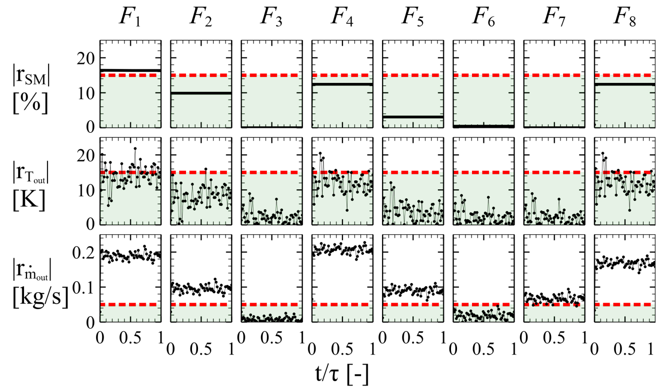

Figure 3. Steady state plots of the absolute value residuals between the "virtual system" and clean model measurements for each fault at the second optimal set of inputs from Table 2, $\mathbf{u}_2 = [1.0, 2.05e4, 0.9, 1.31e5]^T$.

Table 2. BIT design constraints and optimal solution.

Inputs $\mathbf{U}$	Min	Opt	Max
$u_{1,1}$: Valve Position $[-]$	0.00	0.40	1.00
$u_{2,1}$: Compressor Speed $[rpm]$	$1.15e4$	$1.55e4$	$2.15e4$
$u_{3,1}$: Diffuser Position $[-]$	0.00	0.40	1.00
$u_{4,1}$: Exit Pressure $[Pa]$	$1.00e5$	$1.21e5$	$1.35e5$
$u_{1,2}$: Valve Position $[-]$	0.00	1.00	1.00
$u_{2,2}$: Compressor Speed $[rpm]$	$1.15e4$	$2.05e4$	$2.15e4$
$u_{3,2}$: Diffuser Position $[-]$	0.00	0.90	1.00
$u_{4,2}$: Exit Pressure $[Pa]$	$1.00e5$	$1.315e5$	$1.35e5$

Table 3. Reference symptoms map for the two optimal input sets of Table 2, $\mathbf{S}_{ref} = [\mathbf{S}_{ref,1}; \mathbf{S}_{ref,2}]$.

$\mathbf{S}_{ref}$	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
$S^j_{ref,\mathrm{SM},1}$	0	0	0	0	0	0	0	0
$S^j_{ref,T_{out},1}$	0	0	1	0	0	0	0	1
$S^j_{ref,\dot{m},1}$	1	0	1	1	0	1	1	1
$S^j_{ref,\mathrm{SM},2}$	1	0	0	0	0	0	0	0
$S^j_{ref,T_{out},2}$	1	1	0	1	0	0	0	1
$S^j_{ref,\dot{m},2}$	1	1	0	1	1	0	1	1

system characteristics, the number of sensors and the number of faults, two sets of inputs ($N_{test} = 2$) were required to completely isolate all 8 faults from one another. Figures 2 and 3 present the system responses of each faulty "virtual system" state(s), corresponding to the respective input set. The responses are plotted as the absolute value of the residuals between the "virtual system" measurements and the anticipated clean system measurements. The dotted red lines in the figures are design thresholds that represent the regions of allowable response deviations, as determined by analysis of the system uncertainty.

The responses from the two sets of inputs were analyzed to form an optimal reference symptoms map of size $N_{test} \times m \times N_f$ $(2 \times 3 \times 8)$ using Eq. (5). To clearly display the uniqueness of the reference symptoms vectors over both input sets, the two individual reference symptoms maps were concatenated to form Table 3. Table 3 shows the combined symptoms vectors of faults F_1 through F_8. It is clear that each symptoms vector (column) is different, meaning that the studied faults can be isolated from one another on the basis of their unique system responses. Using the optimal input sets, a BIT can be constructed and executed during maintenance at the ideal operating conditions allowing more precise conclusions in regards to the health status of the system.

CONCLUSIONS

A model-based active-FDI method was proposed for the design of built-in tests and demonstrated on a gas compression system. The faulty system responses were optimized with the objective of maximizing their uniqueness, by manipulating the constrained admissible inputs. The uniqueness of the faulty system responses was expressed through a reference symptoms map of vectors describing the anticipated fault symptoms. It was concluded that all faults displayed unique system responses (every fault has a unique symptoms vector in the reference symptoms map) allowing for perfect detection and isolation. Future work will focus on handling system uncertainty and model error in the formulation, to improve the method robustness, while also implementing the direction of the residuals to improve detection and isolation capabilities at a reduced number of tests.

Acknowledgments

This project was sponsored by the UTC Institute for Advanced Systems Engineering (UTC-IASE) of the University of Connecticut and the United Technologies Corporation. Any opinions expressed herein are those of the authors and do not represent those of the sponsor. The authors would like to thank Benson Christalin, Yilin Mo and Richard Murray for initial discussions on problem formulation.

References

Agarwal, S. C. (1984). Centrifugal Compressor Surge Control System.

Batson, B. W. and Narayanan, K. (1995). Surge Control Method and Apparatus.

Chow, E. Y. and Willsky, a. S. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transaction on Automatic Control*, 29(7):603–614.

Dassault Systèmes (2016). Dymola - Dynamic Modeling Library.

Esna Ashari, A., Nikoukhah, R., and Campbell, S. L. (2012). Effects of feedback on active fault detection. *Automatica*, 48(5):866–872.

Hale, W. T. and Bollas, G. M. (2016). Design of Built-In Tests for Active Fault Detection and Isolation of Discrete Faults. *Automatica*, (In Review).

Hwang, I., Kim, S., Kim, Y., and Seah, C. E. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653.

Isermann, R. (1984). Process fault detection based on modeling and estimation methodsA survey. *Automatica*, 20(4):387–404.

Isermann, R. (2005). Model-based fault-detection and diagnosis - Status and applications. *Annual Reviews in Control*, 29:71–85.

Joe Qin, S. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502.

Khan, S., Phillips, P., Hockley, C., and Jennions, I. (2014a). No Fault Found events in maintenance engineering Part 2: Root causes, technical developments and future research. *Reliability Engineering & System Safety*, 123:196–208.

Khan, S., Phillips, P., Jennions, I., and Hockley, C. (2014b). No Fault Found events in maintenance engineering Part 1: Current trends, implications and organizational practices. *Reliability Engineering & System Safety*, 123:183–195.

Kim, K.-K. K., Raimondo, D. M., and Braatz, R. D. (2013). Optimum Input Design for Fault Detection and Diagnosis: Model-based Prediction and Statistical Distance Measures. *Proceedings of the 2013 European Control Conference*, 0:1940–1945.

Le Digabel, S. (2011). Algorithm 909: {NOMAD}: Nonlinear optimization with the {MADS} algorithm. *ACM Transactions On Mathematical Software*, 37(4):44:1—-44:15.

McAuliffe, C. and Beers, C. M. (2006). Integral Add Heat and Surge Control Valve For Compressor.

Modelica Association (2010). Modelica - A Unified Object-Oriented Language for Physical Systems Modeling.

Modelon AB (2014). FMI Toolbox for MATLAB/Simulink.

Niemann, H. (2006). A Setup for Active Fault Diagnosis. *Automatic Control, IEEE Transactions on*, 51(9):1572–1578.

Punčochář, I., Široký, J., and Šimandl, M. (2015). Constrained Active Fault Detection and Control. *IEEE Transactions on Automatic Control*, 60(1):253–258.

Rosich, A., Sarrate, R., Puig, V., and Escobet, T. (2007). Efficient optimal sensor placement for model-based FDI using an incremental algorithm. *Proceedings of the IEEE Conference on Decision and Control*, pages 2590–2595.

Soderholm, P. (2007). A system view of the No Fault Found (NFF) phenomenon. *Reliability Engineering and System Safety*, 92(1):1–14.

The Mathworks Inc. (2013). MATLAB.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):293–311.

Willsky, A. S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611.

Yin, S., Luo, H., and Ding, S. X. (2014). Real-time implementation of fault-tolerant control systems with performance optimization. *IEEE Transactions on Industrial Electronics*, 61(5):2402–2411.

Zolghadri, A. (2012). Advanced model-based FDIR techniques for aerospace systems: Today challenges and opportunities. *Progress in Aerospace Sciences*, 53:18–29.