Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Symbolic regression of uncertainty-resilient inferential sensors for fault diagnostics ⋆

William T. Hale * **George M. Bollas** *

* *Department of Chemical & Biomolecular Engineering, UTC Institute for Advanced Systems Engineering, University of Connecticut, 159 Discovery Dr, Storrs, CT, 06269, USA (e-mail: {william.hale; george.bollas} @uconn.edu)*

Abstract: An algorithm is presented for the design of inferential sensors for fault diagnostics in thermal management systems. The algorithm uses input and output sensed system information to improve the detection and isolation of a fault by generating inferential sensors that augment the measured information to: (i) reduce the evidence of uncertainty in the inferred variables, and thus decrease false alarm and nondetection rates; and (ii) provide distinguishable responses to faults, and thus reduce reduce the rate of misdiagnoses. The novelty of the algorithm is its use of genetic programming to evolve explainable inferential sensors that maximize information criteria specific to fault diagnostics. The chosen criteria: (i) least squares regression; and (ii) D_s-optimality (calculated from the Fisher Information Matrix), leverage symbolic mathematics and automatic differentiation to obtain parametric sensitivities of the measured outputs and inferential sensors. The algorithm is included in a standard work for fault diagnostics, where its effectiveness is assessed through k-NN classification and illustrated in an application to an aircraft cross-flow plate-fin heat exchanger.

Keywords: fault detection and diagnosis, experiment design, AI methods for FDI

1. INTRODUCTION

Data analytics typically involves methods and apparatus that explore and improve the overall quality and efficiency of a process or system. Data analytics can be performed in a descriptive (e.g., characterization, association, correlation) or predictive (e.g., classification, regression) manner by leveraging developments in statistics (Gajjar et al., 2018; Zhou et al., 2014), machine learning (Domingos, 2012; Jordan and Mitchell, 2015), artificial intelligence (Liu et al., 2018), and deep learning (LeCun et al., 2015; Liu et al., 2017; Schmidhuber, 2015). The techniques developed today aim at tackling the issues of noise, uncertainty, and errors present in data that can result in erroneous conclusions. An approach to dealing with these issues is to implement inferential sensors (i.e., latent variables or soft sensors) (Fortuna et al., 2007). Inferential sensors are indirect estimates of complex process variables or parameters that cannot be measured or require additional infrastructure investment. These indirect system measurements are often in the form of analytical relationships or other functional combinations of available system information (measured system inputs and outputs). These analytical relationships can be based on physical laws, domain system knowledge, or empirical relationships such as: regression models, support vector machines, neural networks, or genetic programs. Inferential sensors can provide additional, more accurate insight into process variables impacted by noise and uncertainty, at a lower cost than more sophisticated physical sensing methods. The added insight can result in more robust information compared to traditional physical sensors by reducing the evidence of noise and uncertainty, which is critical to the field of fault diagnostics and prognostics (Hale et al., 2019).

In this paper, symbolic regression and test design optimization are used to create inferential sensors for the purposes of fault detection and isolation (FDI). This method combines genetic and mathematical programming to find optimal inferential sensors and system operating points that improve the information with respect to faults in the system. The inferential sensors use existing system measurements at an optimal operating point to reduce the impact of noise and uncertainty and provide better FDI capability. The method is applied to a system where increased uncertainty during operation negatively impacts system performance, reliability, and safety by masking the occurrence of faults (resulting in missed detections) or mimicking faulty performance (resulting in false alarms). The algorithm and workflow of this method extends upon the design of robust built-in tests for active FDI (Hale and Bollas, 2018; Hale et al., 2019; Palmer et al., 2019; Palmer and Bollas, 2019a,b). The product of this work is an algorithm that produces an optimal test design (i.e., admissible input values, the set of physical sensors used, and the inferential sensors that augment the sensor architecture) that improves fault diagnostics and prognostics for systems prone to noise and uncertainty.

⋆ 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.

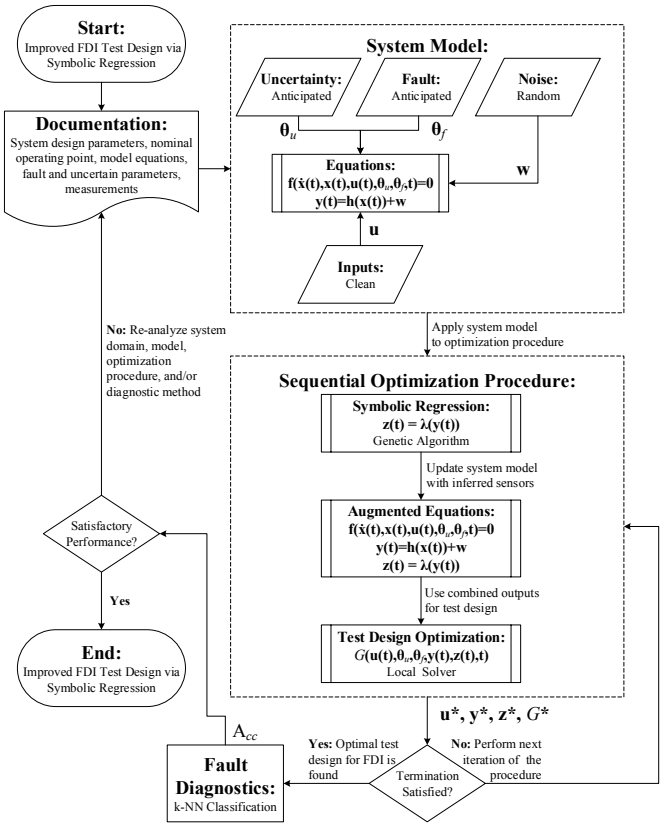

Fig. 1. Block-flow diagram of the sequential inferential sensor and input design method proposed.

2. METHODS

The proposed model-based active FDI workflow is shown in Fig. 1. The three main steps of this workflow involve system modeling, optimization for FDI, and execution of FDI. Of these steps, the main focus is on the algorithm for FDI optimization in systems with uncertainty.

2.1 Inferential sensor and input design using a sequential optimization procedure

The framework of Fig. 1 utilizes accurate steady-state or dynamic models of the system subject to uncertainty and anticipated faults, including its healthy state. The developed system model is used in a sequential optimization procedure for improving fault diagnostics. First, the system model is used to generate predictions of the system outputs at a given operating point and anticipated uncertainty (i.e., mean values for uncertain variables and parameters). These system outputs are then used to symbolically regress inferential sensors that maximize a desired objective. Next, the regressed inferential sensors augment the original system model and are used for FDI test design optimization, where the best test design (i.e., the selected physical sensors, inferential sensors and test conditions) are calculated. The augmented model is used to run Monte Carlo simulations at the optimal test conditions over the range of uncertainty (assumed normally distributed), to assess the effectiveness of the test design. The pseudo-algorithm of this procedure is presented in Alg. 1.

The set of differential algebraic equations that accurately models the system of interest and its anticipated faults and uncertainty is:

$$\mathbf{f}^{[f]}(\dot{\mathbf{x}}^{[f]}(t), \mathbf{x}^{[f]}(t), \mathbf{u}(t), \boldsymbol{\theta}_u, \boldsymbol{\theta}_f^{[f]}, t) = \mathbf{0}, \quad \forall [f] \in \{[0], ..., [N_f]\} \tag{1}$$

where $\mathbf{f}^{[f]} : D_{x^{[f]}} \times D_u \times D_{\boldsymbol{\theta}_u} \times D_{\boldsymbol{\theta}_f^{[f]}} \rightarrow \mathbb{R}^{N_x}$ is the system of equations that are assumed to be continuously differentiable over its open domain $D_{x^{[f]}} \subset \mathbb{R}^{N_x}$, $D_u \subset \mathbb{R}^{N_u}$, $D_{\boldsymbol{\theta}_u} \subset \mathbb{R}^{N_{\boldsymbol{\theta}_u}}$, $D_{\boldsymbol{\theta}_f^{[f]}} \subset \mathbb{R}^{N_{\boldsymbol{\theta}_f}}$. The superscript $[f]$ denotes the fault scenario of interest and N_f is the total number of faults studied (with $[f] = [0]$ representing the fault-free system). The variable $\mathbf{x}^{[f]} \in D_{x^{[f]}} \subset \mathbb{R}^{N_x}$ is the vector of system states, $\mathbf{u} \in U = \{\mathbf{u} \in \mathbb{R}^{N_u} : \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U\}$ is the vector of admissible system inputs, $\boldsymbol{\theta}_u \in \Theta_u = \{\boldsymbol{\theta}_u \in \mathbb{R}^{N_{\boldsymbol{\theta}_u}} : \boldsymbol{\theta}_u^L \leq \boldsymbol{\theta}_u \leq \boldsymbol{\theta}_u^U\}$ is the vector of uncertain parameters, $\boldsymbol{\theta}_f^{[f]} \in \Theta_f^{[f]} = \{\boldsymbol{\theta}_f^{[f]} \in \mathbb{R}^{N_{\boldsymbol{\theta}_f}} : {\boldsymbol{\theta}_f^{[f]}}^L \leq \boldsymbol{\theta}_f^{[f]} \leq {\boldsymbol{\theta}_f^{[f]}}^U\}$ is the vector of parameters corresponding to fault $[f]$, and t is time. The system outputs are expressed as:

$$\mathbf{y}^{[f]}(t) = \mathbf{h}(\mathbf{x}^{[f]}(t)) + \mathbf{w}, \quad \forall [f] \in \{[0], ..., [N_f]\} \tag{2}$$

where $\mathbf{y}^{[f]} \in Y^{[f]} \subset \mathbb{R}^{N_y}$ is the vector of system outputs corresponding to $[f]$, $\mathbf{h}$ is the system of equations mapping the system states to the measured outputs, and $\mathbf{w} \in W \subset \mathbb{R}^{N_y}$ is the vector of measurement noise. Using the measured system outputs and inputs, inferential sensors can be developed:

$$\mathbf{z}^{[f]}(t) = \boldsymbol{\lambda}(\mathbf{u}(t), \mathbf{y}^{[f]}(t)), \quad \forall [f] \in \{[0], ..., [N_f]\} \tag{3}$$

where $\mathbf{z}^{[f]} \in Z^{[f]} \subset \mathbb{R}^{N_z}$ is the vector of inferential sensors corresponding to $[f]$ and $\boldsymbol{\lambda}$ is the system of equations mapping the measured inputs and outputs to the inferential sensors. These inferential sensors are augmented to the original system model. The initial conditions at time t_0 for (1), (2), and (3) are expressed as:

$$\gamma_0 = \begin{cases} \mathbf{f}(\dot{\mathbf{x}}(t_0), \mathbf{x}(t_0), \mathbf{u}(t_0), \boldsymbol{\theta}_u, \boldsymbol{\theta}_f, t_0) = \mathbf{0}, \\ \mathbf{y}(t_0) = \mathbf{h}(\mathbf{x}(t_0)), \\ \mathbf{z}(t_0) = \boldsymbol{\lambda}(\mathbf{y}(t_0), \mathbf{u}(t_0)) \end{cases} \tag{4}$$

where γ_0 is the combined vector of initial conditions. The general formulation of the sequential optimization procedure for inferential sensor and input design is:

$$\begin{aligned} G^* = & \max_{\mathbf{u} \in U, \mathbf{y} \in Y, \mathbf{z} \in Z} \quad G(\mathbf{u}(t), \tilde{\boldsymbol{\theta}}_u, \tilde{\boldsymbol{\theta}}_f, \mathbf{y}(t), \mathbf{z}(t), t) \\ \text{s.t. } & \mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{u}(t), \tilde{\boldsymbol{\theta}}_u, \tilde{\boldsymbol{\theta}}_f, t) = (\mathbf{f}^{[1]}, ..., \mathbf{f}^{[N_f]}) = \mathbf{0}, \\ & \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{w} = (\mathbf{y}^{[1]}, ..., \mathbf{y}^{[N_f]}), \\ & \mathbf{z}(t) = \boldsymbol{\lambda}(\mathbf{u}(t), \mathbf{y}(t)) = (\mathbf{z}^{[1]}, ..., \mathbf{z}^{[N_f]}), \\ & \gamma_0 = \begin{cases} \mathbf{f}(\dot{\mathbf{x}}(t_0), \mathbf{x}(t_0), \mathbf{u}(t_0), \boldsymbol{\theta}_u, \boldsymbol{\theta}_f, t_0) = \mathbf{0}, \\ \mathbf{y}(t_0) = \mathbf{h}(\mathbf{x}(t_0)), \\ \mathbf{z}(t_0) = \boldsymbol{\lambda}(\mathbf{u}(t_0), \mathbf{y}(t_0)) \end{cases} \end{aligned} \tag{5}$$

where $G : D_u \times D_{\boldsymbol{\theta}_u} \times D_{\boldsymbol{\theta}_f} \times D_y \times D_z \times D_t \rightarrow \mathbb{R}$ is the objective function that defines the FDI capability, $\mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{u}(t), \tilde{\boldsymbol{\theta}}_u, \tilde{\boldsymbol{\theta}}_f, t) = (\mathbf{f}^{[0]}, ..., \mathbf{f}^{[N_f]}) = 0$ is the system of differential algebraic equations combined for all fault scenarios from (1), augmented with the state variables $\mathbf{x} = (\mathbf{x}^{[0]}, ..., \mathbf{x}^{[N_f]}) \in X \subset \mathbb{R}^{N_x(N_f+1)}$ and parameters corresponding to faults $\tilde{\boldsymbol{\theta}}_f = (\tilde{\boldsymbol{\theta}}_f^{[0]}, ..., \tilde{\boldsymbol{\theta}}_f^{[N_f]}) \in \Theta_f \subset \mathbb{R}^{N_{\boldsymbol{\theta}_f}(N_f+1)}$ at their anticipated values ($\sim$), $\tilde{\boldsymbol{\theta}}_u \in \Theta$ is the vector of uncertain parameters at their anticipated values, $\mathbf{y} \in Y \subset \mathbb{R}^{N_y(N_f+1)}$ is the combined vector of system outputs, and $\mathbf{z} \in Z \subset \mathbb{R}^{N_z(N_f+1)}$ is the combined

Algorithm 1 Symbolic Regression and Optimization for Inferential Sensor and Input Design.

initialization;
Model: Equations, States, Inputs, Uncertain Parameters, Fault Parameters, Outputs
Genetic Program: Basis Functions, Inputs, Population, Generations, Complexity, Parameter Ranges, Objective Function, Selection Type, Crossover and Mutation Probabilities
Optimization: Objective Function, Initial Guess, Constraints, Termination Criteria

algorithm;
1: **Initialize:** randomly generate initial population of N_{pop} inferential sensors and save in the population $\Lambda_0 = \{z_1, ..., z_{N_{pop}}\}$
2: **Genetic Program Procedure:**
while not at final generation, $k \neq K_{max}$ **do** population evolution
Evolve Population
Performance: calculate objective function G_i of each individual $z_i,\ i = 1, ..., N_{pop}$, in Λ_k and rank;
Elitism: select the best $N_{elite} = p_{elite}\ N_{pop}$ individuals from Λ_k and save them to the next generation Λ_{k+1};
Direct: select $N_{direct} = p_{direct}\ N_{pop}$ individuals from Λ_k and save them to the next generation Λ_{k+1};
Crossover: select $N_{cross} = p_{cross}\ N_{pop}$ pairs of individuals from Λ_k, perform crossover, and save them to the next generation Λ_{k+1};
Mutation: select $N_{mutate} = p_{mutate}\ N_{pop}$ individuals from Λ_k, perform mutation, and save them to the next generation Λ_{k+1};
Update: calculate objective function G_i of each individual $z_i,\ i = 1, ..., N_{pop}$, in the next generation Λ_{k+1} and rank;
Optimize: perform 3: **Test Design Optimization Procedure** using the best ranked individual(s) in Λ_{k+1} to find optimal operating point $\mathbf{u}$ and selected sensors $\mathbf{y}$ and $\mathbf{z}$;
repeat for next generation $k = k + 1$;
end while
Return $\mathbf{u}^*, \mathbf{y}^*, \mathbf{z}^*, G^*$
3: **Test Design Optimization Procedure:**
Select Sensors and Optimize Operating Point
Initial Guess: provide the optimizer with an initial guess for $\mathbf{u}^*$;
Simulation: simulate the system at the initial guess/future iterations of $\mathbf{u}$;
Objective Function: calculate the objective function G given the input $\mathbf{u}$ and selected sensors $\mathbf{y}$ and $\mathbf{z}$;
Convergence: iterate until the optimal objective function G^* is found at the optimal input $\mathbf{u}^*$ using the selected sensors $\mathbf{y}^*$ and $\mathbf{z}^*$;
Update: update the measured outputs used in 2: **Genetic Program Procedure** for the optimal operating point;

vector of inferential sensors. It is important to note that the objective function G should be appropriately chosen for the problem at hand. Using Alg. 1 and the equations above, a genetic programming procedure is implemented to evolve a population of N_{pop} inferential sensors $\boldsymbol{\Lambda} = \{\mathbf{z}_1, ..., \mathbf{z}_{N_{pop}}\}$ of varying complexity (built from a list of basis functions), whose independent variables are the measured inputs and outputs. The basis functions used to form and evolve the initial population of inferential sensors can originate from domain expert knowledge of the key physics pertaining to the system and faults of interest, to better capture the evidence of faults. The first generation of individuals in the genetic programming procedure is randomly generated from these basis functions. The population then undergoes evolution, where a percentage of the population is selected for direct reproduction (for the best performers), crossover, and mutation. After evolution, the most informative individual(s) for FDI, based on the selected metric/objective, are selected from the population and used in optimizing the system operating point $\mathbf{u}(t)$ to further improve FDI. Once the new optimal operating point is found, the measured inputs and outputs are updated and the next generation of evolution in the genetic program is performed. This process continues until the maximum number of generations is reached. At this point, the best test design (i.e., optimal/selected physical sensor(s), inferential sensor(s), and operating point(s)) is provided for diagnostics and prognostics.

2.2 Test design criteria for fault diagnostics

Two criteria were considered for designing the inferential sensors and operating points of FDI. The first was a least-squares objective function, commonly used in regression. This objective function minimizes the squared error between the predicted and actual values of the parameter(s) of interest (i.e., fault(s)), where the inferential sensors of (3) are designed to predict an anticipated fault from the noisy measured outputs in the presence of system uncertainty. The objective function was formulated as:

$$G(\mathbf{u}(t), \tilde{\boldsymbol{\theta}}_u, \tilde{\boldsymbol{\theta}}_f, \mathbf{y}(t), \mathbf{z}(t), t) = \sum_{i=1}^{N_s} \sum_{j=0}^{N_f} (\tilde{\theta}_f^{[j]} - z(i)^{[j]})^2, \quad (6)$$

where N_s is the number of sampling points (at steady-state, z is indexed by the Monte Carlo samples instead of time). A key requirement of this method is that data on the fault/class exists beforehand. When this is unavailable, other objectives need to be chosen, such as the D_s-optimality criterion (Atkinson and Bogacka, 1997). The D_s-optimality criterion utilizes the Fisher Information Matrix (FIM), a common metric used in the fields of statistics and design of experiments (Han et al., 2016a,b; Galvanin et al., 2006; Bock et al., 2007). In its application to fault diagnostics and prognostics, D_s-optimality minimizes the the covariance between the parameters representing faults and uncertainty (improving isolation) while neglecting the covariance between the parameters representing uncertainty (improving detection), thus reducing the joint confidence region between the uncertain and fault parameters. The general formulation of the D_s-optimality criterion is expressed as:

$$G(\mathbf{u}(t), \tilde{\boldsymbol{\theta}}_u, \tilde{\boldsymbol{\theta}}_f, \mathbf{y}(t), \mathbf{z}(t), t)) = \boldsymbol{\psi}(\mathbf{H}) = \frac{|\mathbf{H}|}{|H_{uu}|} \quad (7)$$

where $\boldsymbol{\psi}$ is the test design criterion (in this case D_s-optimality), $\mathbf{H}$ is the FIM, and H_{ff}, H_{fu}, H_{uf}, and H_{uu} are submatrix blocks in the FIM that provide information on the relationship between faults, faults and uncertainty, and uncertainty, respectively. The information gained from the FIM depends on the outputs chosen for FDI, which in this work includes both the measured outputs and inferential sensor outputs. The FIM can be calculated by taking the partial derivatives of the selected measured outputs and inferential sensors with respect to the uncertain and fault parameters, known as their parametric

sensitivities. The outputs are selected using the binary vector $\mathbf{a} = (a_1, ..., a_{N_y+N_z})$ and their sensitivities are calculated using forward sensitivity analysis and symbolic differentiation. Mathematically, the FIM is expressed as:

$$\mathbf{H} = \begin{bmatrix} H_{ff} H_{fu} \\ H_{uf} H_{uu} \end{bmatrix} = (\mathbf{1}^T \mathbf{a})^{-1} \sum_{i=1}^{N_y+N_z} \sum_{j=1}^{N_y+N_z} a_i a_j \sigma_{ij}^{-2} \mathbf{Q}_i^T \mathbf{Q}_j \quad (8)$$

where $(\mathbf{1}^T \mathbf{a})^{-1}$ is a normalization factor equal to the number of measured outputs and inferential sensors selected, the elements of $\mathbf{a}$ are the binary decision variables corresponding to the chosen measured or inferential outputs, $\sigma_{i,j}$ is the known variance between the i-th and j-th outputs, and $\mathbf{Q}_i$ is the sensitivity matrix of the i-th output containing the partial derivatives with respect to anticipated uncertain and fault parameters. For brevity, in the following we present the results of using the objective function (6) in (5) and discuss the value of the resulting inferential sensors.

2.3 Methods of classification for FDI

Common methods for fault diagnostics using the optimal operating point and selected sensors (physical and inferential) are neural networks, principal component analysis, and support vector machines (Yu, 2013; Moosavian et al., 2013; Palmer and Bollas, 2019a). For this work, the k-nearest neighbors (k-NN) algorithm was selected due to its simplicity and commercial success (Chen and Shah, 2018). Performing classification using k-NN involves classifying the system measured outputs $\mathbf{y}$ and the inferred outputs $\mathbf{z}$ through searching the neighborhood of nearby historical points to provide confusion matrices of the actual vs. predicted classes and the overall (mean) correct classification rate, A_{cc} (Hale et al., 2019). This historical dataset, known as the training data, is obtained here by running a Monte Carlo simulation for the given uncertainty domain. The accuracy of k-NN classification is gauged by creating a prediction data set through Monte Carlo simulation, independent from the training data, that is tested against the trained classifier to calculate the confusion matrices and correct classification rate.

3. APPLICATION TO AN AIRCRAFT SYSTEM

3.1 Classification of particulate fouling in a cross-flow plate-fin heat exchanger amid uncertainty

Alg. 1 is demonstrated on a cross-flow plate-fin heat exchanger (PFHE) system that is prone to fouling and has uncertain flow conditions. The diagram of the PFHE, whose system model was originally studied by Palmer et al. (2016), is shown in Fig. 2. The single admissible system input is the controlled mass flow rate of the hot stream, $u = \dot{m}_{h,i}$ (kg/s). System uncertainty exists in the cold air inlet stream moisture content, temperature, mass flow rate, and pressure $\boldsymbol{\theta}_u = (\omega_{\mathrm{H_2O},c,i}, T_{c,i}, \dot{m}_{c,i}, P_{c,i})$, with normal distributions as presented in Table 1. The fault of the system is particulate fouling in the cold stream side, expressed as a change in thermal fouling resistance $\boldsymbol{\theta}_f^{[f]} = R_f^{[f]}$. Three levels of fouling are studied: 20% blocked, 50% blocked, and 80% blocked, along with the fouling-free scenario. The measured outputs of the system

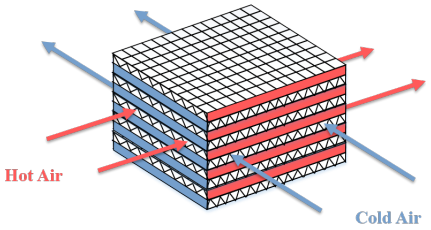

Fig. 2. Cross-flow plate-fin heat exchanger diagram from Palmer et al. (2016). Uncertainty exists in the flow conditions of the cold air stream (blue) and is presented in Table 1.

Table 1. Parameters of uncertainty $\boldsymbol{\theta}_u$ and faults $\boldsymbol{\theta}_f^{[f]}$ and their normally distributed $\mathcal{N}(\mu, \sigma^2)$ values with mean μ and variance σ^2.

Faults and Uncertainties	Distribution
Moisture Content (kg H_2O/kg dry air)	$\mathcal{N}(2.0, 0.0625)$
Temperature (°C)	$\mathcal{N}(30.0, 1.0)$
Mass Flow (kg/s)	$\mathcal{N}(1.0, 0.0025)$
Pressure (Pa)	$\mathcal{N}(10^5, 6.25 \cdot 10^6)$
Thermal Fouling Resistance (m^2K/W)	
Fault-Free: 0% Blocked	$\mathcal{N}(0.00, 0.0)$
Fault 1: 20% Blocked	$\mathcal{N}(1.60 \cdot 10^{-3}, 0.0)$
Fault 2: 50% Blocked	$\mathcal{N}(4.00 \cdot 10^{-3}, 0.0)$
Fault 3: 80% Blocked	$\mathcal{N}(6.40 \cdot 10^{-3}, 0.0)$

are $\mathbf{y} = (T_{c,o}, T_{h,o}, P_{c,o}, P_{h,o})$ where $T_{c,o}$ (°C) is the cold stream outlet temperature, $T_{h,o}$ (°C) is the hot stream outlet temperature, $P_{c,o}$ (Pa) is the cold stream outlet pressure, and $P_{h,o}$ (Pa) is the hot stream outlet pressure.

The impact of uncertainty on the PFHE outputs was explored through a Monte Carlo simulation of 150 samples for the fault and uncertainties reported in Table 1. This was done at two different operating points (i.e., inputs) in order to understand the impact system operating conditions have on fault diagnostics. The resulting measured outputs at their final steady-state for each of the 150 Monte Carlo samples are shown in Fig. 3. The "Nominal" operating point was taken to be $u = 0.25$ kg/s and the "Optimal" operating point was taken at $u = 1.0$ kg/s from previous work (Palmer et al., 2019). Looking at Fig. 3, it is obvious that there exists significant overlap in the measured outputs due to the impact of uncertainty for each of the 4 fault scenarios studied. This was evident at both operating points (with a slight improvement in the "Optimal" test), creating challenges in the detection and isolation of faults. To quantify the challenge of FDI given the overlap in measured outputs, k-NN classification was performed using a training data set of 10,000 Monte Carlo simulation samples for both operating points. The actual class (i.e., fault scenario) of each of the 150 Monte Carlo samples in Fig. 3 was predicted using the trained classifier. As expected from the overlapping outputs, the correct classification rates were found to be very poor, ranging from 25%–55% for the two operating points.

3.2 Inferential sensor design and implementation for steady-state FDI of different fouling levels

The goal of designing inferential sensors was to improve the overall rate of correct classification of fouling by optimizing

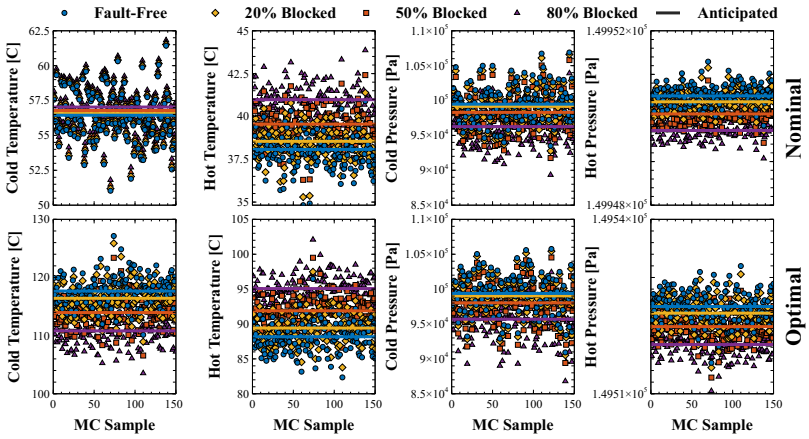

Fig. 3. PFHE measured outputs at the "Nominal" $u = 0.25$ kg/s and "Optimal" $u = 1.0$ kg/s operating point from Monte Carlo Simulation. The solid lines show the measured outputs at the anticipated value of uncertainty in Table 1 for each fault scenario.

Table 2. Arbitrary (z_1) and optimal ($z_2 - z_3$) inferential sensors.

Inferential Sensor	Equation
z_1	$\exp(y_2/y_1)$
z_2	$-31.94 + 28.40\sin(\sqrt{y_1}) + 28.40\sin(y_1^{0.25})$ $+14.43\sin(y_4) - 28.45\sqrt{y_1}\cos(y_4)$ $-0.00019y_2\exp(\sqrt{y_1}) + 3.50\cos(y_4)(y_1 + \sqrt{y_2})$
z_3	$3.26 + 0.024y_1y_2 - 0.01y_1^2 - 0.01y_2^2$

the system operating point and incorporating inferential sensors into FDI to reduce the impact of uncertainty. To illustrate the benefit that operating points and inferential sensors can provide in achieving this goal, an example inferential sensor z_1 was arbitrarily created (equation shown in Table 2) using the outputs of Fig. 3 as the independent variables. This inferential sensor ("Inferential Sensor # 1") is shown in Fig. 4 for each sample of the Monte Carlo simulation at the "Optimal" operating point. In Fig. 4, it can be seen that using the mathematical operators of z_1 on the measured outputs, significantly reduced the noise associated with the different realizations of uncertainty. This allowed more accurate conclusions/classifications to be made on the (now) fully separated fault scenarios, which was validated by the 100% correct classification rate obtained when performing k-NN classification with z_1 at the "Optimal" operating point. However, the arbitrary inferential sensor z_1 had a correct classification rate of only 62% at the "Nominal" operating point, highlighting the need and value of optimizing both the inferential sensor and operating point for diagnostics.

After validation of the benefit of inferential sensors in FDI, Alg. 1 was used to create inferential sensors that explicitly infer the value of thermal fouling resistance R_f from the measured outputs, at both operating points. The objective function (6) was selected and optimized through a tree-based genetic program, using the open source software GPTIPS 2 (Searson, 2015), to produce an optimal inferential sensor z_2 at the "Nominal" operating point and an optimal inferential sensor z_3 at the "Optimal" operating point (equations shown in Table 2). The uncertain fouling predictions of z_3 ("Inferential Sensor # 3") is shown in Fig. 4. The accuracy of the inferential sensor in the pre-

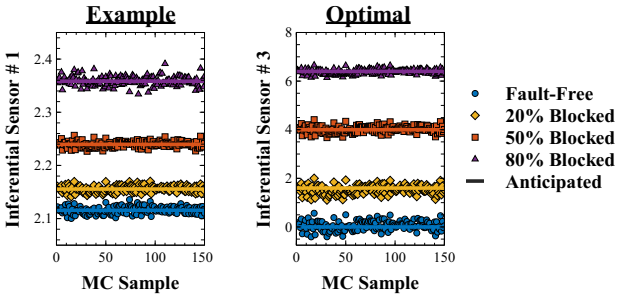

Fig. 4. PFHE Monte Carlo simulation of the example inferential sensors z_1 and the optimal inferential sensor z_3 in Table 2 at the "Optimal" operating point $u_1 = 1.00$ kg/s. The solid lines show the inferential sensor output at the anticipated value of uncertainty in Table 1 for each fault scenario.

diction of thermal fouling resistance for each fault scenario (anticipated values $R_f = \{0, 1.4, 4.0, 6.4\}$) was excellent. The separation between the four fault scenarios in Fig. 4 makes the classification of data transformed through the optimal inferential sensor z_3 very easy. Specifically, the correct classification rate was 100% in this case. It is interesting to note that inferential sensor z_3 formed an analytical relationship between the two temperature outputs (y_1 and y_2) while ignoring the pressure outputs (y_3 and y_4). A further discussion on the algorithm's ability to select sensors that provide more pertinent information for fault diagnostics will be had in future work.

3.3 Dynamic inferential sensor design

Finally, the trajectories of both the "Nominal" and the "Optimal" operating points were studied, along with the transition from the former to the latter. The same issue of overlap in the measured outputs of Fig. 3 was observed in the dynamic responses. The challenge of the symbolic regression algorithm was then to identify invariants (the constant fault value) in noisy transient data. This issue has been addressed in Schmidt and Lipson (2009) and a similar method was deployed here. The invariants in the data were searched using genetic programming to learn variables that remain constant in (6). The method presented here was

successful in finding one inferential sensor that improves the isolation of each fault scenario for each steady state (the initial and the final) of the dynamic test. Invariants in the transient data were approximated satisfactorily, but all the inferential sensors exhibited some dynamic response, which currently is under investigation. Overall, algorithm Alg. 1 with the objective of (6) were shown successful in developing latent variables that can detect constant faults at tests that vary over time (variable steady states and transient data).

CONCLUSIONS

We presented a novel algorithm for developing optimal inferential sensors and operating points that reduce the impact of uncertainty on inferred outputs during fault diagnostics. Techniques from optimization and symbolic regression were deployed to create inferential sensors with much richer information, via their mathematical representations of the measured inputs and outputs. It was shown that deploying inferential sensors in FDI at an optimal operating point improves the correct classification rates of in FDI, achieving rates of 100%.

REFERENCES

Atkinson, A.C. and Bogacka, B. (1997). D- and Ds-optimum compound designs for the order of a chemical determining reaction. *Technometrics*, 39(4), 347–356.

Bock, H.G., Kostina, E., and Kostyukova, O. (2007). Covariance Matrices for Parameter Estimates of Constrained Parameter Estimation Problems. *SIAM Journal on Matrix Analysis and Applications*, 29(2), 626–642.

Chen, G.H. and Shah, D. (2018). Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends in Machine Learning*, 10(5–6), 337–588.

Domingos, P. (2012). A few useful things to know about machine learning. *Commun. ACM*, 55(10), 78–87.

Fortuna, L., Graziani, S., Rizzo, A., and Xibilia, M.G. (2007). *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer, London.

Gajjar, S., Kulahci, M., and Palazoglu, A. (2018). Real-time fault detection and diagnosis using sparse principal component analysis. *Journal of Process Control*, 67, 112–128.

Galvanin, F., Barolo, M., Bezzo, F., and Macchietto, S. (2006). A framework for model-based design of parallel experiments in dynamic systems. *Computer Aided Chemical Engineering*, 21(C), 249–254.

Hale, W.T. and Bollas, G.M. (2018). Design of Built-In Tests for Active Fault Detection and Isolation of Discrete Faults. *IEEE Access*, 6, 50959–50973.

Hale, W.T., Wilhelm, M.E., Palmer, K.A., Stuber, M.D., and Bollas, G.M. (2019). Semi-infinite programming for global guarantees of robust fault detection and isolation in safety-critical systems. *Computers & Chemical Engineering*, 126, 218–230.

Han, L., Zhou, Z., and Bollas, G.M. (2016a). Model-based analysis of chemical-looping combustion experiments. Part I: Structural identifiability of kinetic models for NiO reduction. *AIChE Journal*, 62(7), 2419–2431.

Han, L., Zhou, Z., and Bollas, G.M. (2016b). Model-based analysis of chemical-looping combustion experiments. Part II: Optimal design of CH 4 -NiO reduction experiments. *AIChE Journal*, 62(7), 2432–2446.

Jordan, M.I. and Mitchell, T.M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444.

Liu, R., Yang, B., Zio, E., and Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108, 33–47.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F.E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11–26.

Moosavian, A., Ahmadi, H., Tabatabaeefar, A., and Khazaee, M. (2013). Comparison of two classifiers; K-nearest neighbor and artificial neural network, for fault diagnosis on a main engine journal-bearing. *Shock and Vibration*, 20(2), 263–272.

Palmer, K.A. and Bollas, G.M. (2019a). Active fault diagnosis for uncertain systems using optimal test designs and detection through classification. *ISA Transactions*, 93, 354–369.

Palmer, K.A. and Bollas, G.M. (2019b). Sensor Selection Embedded in Active Fault Diagnosis Algorithms. *IEEE Transactions on Control Systems Technology*, 1–14.

Palmer, K.A., Hale, W.T., and Bollas, G.M. (2019). Active Fault Identification by Optimization of Test Designs. *IEEE Transactions on Control Systems Technology*, 27(6), 2484–2498.

Palmer, K.A., Hale, W.T., Such, K.D., Shea, B.R., and Bollas, G.M. (2016). Optimal design of tests for heat exchanger fouling identification. *Applied Thermal Engineering*, 95, 382–393.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.

Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, 324(5923), 81–85.

Searson, D.P. (2015). *GPTIPS 2: An Open-Source Software Platform for Symbolic Data Mining*, 551–573. Springer International Publishing, Cham.

Yu, J. (2013). A support vector clustering-based probabilistic method for unsupervised fault detection and classification of complex chemical processes using unlabeled data. *AIChE Journal*, 59(2), 407–419.

Zhou, L., Chen, J., Song, Z., Ge, Z., and Miao, A. (2014). Probabilistic latent variable regression model for process-quality monitoring. *Chemical Engineering Science*, 116, 296–305.