2-28-2020

Development of Real-Time System Identification to Detect Abnormal Operations in a Gas Turbine Cycle

Harry Bonilla-Alvarado
Iowa State University and Ames Laboratory, hbonilla@iastate.edu

Kenneth Bryden
Iowa State University and Ames Laboratory, kmbryden@iastate.edu

Lawrence Shadle
United States Department of Energy

David Tucker
United States Department of Energy

Paolo Pezzini
Ames Laboratory, ppezzini@ameslab.gov

Development of Real-Time System Identification to Detect Abnormal Operations in a Gas Turbine Cycle

Abstract

This paper presents a novel online system identification methodology for monitoring the performance of power systems. This methodology was demonstrated in a gas turbine recuperated power plant designed for a hybrid configuration. A 120-kW Garrett microturbine modified to test dynamic control strategies for hybrid power systems designed at the National Energy Technology Laboratory (NETL) was used to implement and validate this online system identification methodology. The main component of this methodology consists of an empirical transfer function model implemented in parallel to the turbine speed operation and the fuel control valve, which can monitor the process response of the gas turbine system while it is operating. During fully closed-loop operations or automated control, the output of the controller, fuel valve position, and the turbine speed measurements were fed for a given period of time to a recursive algorithm that determined the transfer function parameters during the nominal condition. After the new parameters were calculated, they were fed into the transfer function model for online prediction. The turbine speed measurement was compared against the transfer function prediction, and a control logic was implemented to capture when the system operated at nominal or abnormal conditions. To validate the ability to detect abnormal conditions during dynamic operations, drifting in the performance of the gas turbine system was evaluated. A leak in the turbomachinery working fluid was emulated by bleeding 10% of the airflow from the compressor discharge to the atmosphere, and electrical load steps were performed before and after the leak. This tool could detect the leak 7 s after it had occurred, which accounted for a fuel flow increase of approximately 15.8% to maintain the same load and constant turbine speed operations.

Keywords

gas turbine, fault detection, system identification

Disciplines

Energy Systems

Harry Bonilla-Alvarado
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: hbonilla@iastate.edu

Kenneth M. Bryden[1]
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: kmbryden@iastate.edu

Lawrence Shadle
U.S. Department of Energy National Technology
Energy Laboratory,
3610 Collins Ferry Road,
Morgantown, WV 26507
e-mail: Lawrence.Shadle@netl.doe.gov

David Tucker
U.S. Department of Energy National Technology
Energy Laboratory,
3610 Collins Ferry Road,
Morgantown, WV 26507
e-mail: david.tucker@netl.doe.gov

Paolo Pezzini
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: ppezzini@ameslab.gov

Development of Real-Time System Identification to Detect Abnormal Operations in a Gas Turbine Cycle

This paper presents a novel online system identification methodology for monitoring the performance of power systems. This methodology was demonstrated in a gas turbine recuperated power plant designed for a hybrid configuration. A 120-kW Garrett microturbine modified to test dynamic control strategies for hybrid power systems designed at the National Energy Technology Laboratory (NETL) was used to implement and validate this online system identification methodology. The main component of this methodology consists of an empirical transfer function model implemented in parallel to the turbine speed operation and the fuel control valve, which can monitor the process response of the gas turbine system while it is operating. During fully closed-loop operations or automated control, the output of the controller, fuel valve position, and the turbine speed measurements were fed for a given period of time to a recursive algorithm that determined the transfer function parameters during the nominal condition. After the new parameters were calculated, they were fed into the transfer function model for online prediction. The turbine speed measurement was compared against the transfer function prediction, and a control logic was implemented to capture when the system operated at nominal or abnormal conditions. To validate the ability to detect abnormal conditions during dynamic operations, drifting in the performance of the gas turbine system was evaluated. A leak in the turbomachinery working fluid was emulated by bleeding 10% of the airflow from the compressor discharge to the atmosphere, and electrical load steps were performed before and after the leak. This tool could detect the leak 7 s after it had occurred, which accounted for a fuel flow increase of approximately 15.8% to maintain the same load and constant turbine speed operations. [DOI: 10.1115/1.4046144]

Keywords: gas turbine, fault detection, system identification

1 Introduction

One of the most critical aspects in the operation of thermal power plants is the need for reliable and robust solutions that would enable the early detection and diagnosis of potentially harmful events that can occur during operations. Due to the substantial cost of maintenance during forced outages, early detection of abnormal operations or failures represents a critical task that may help to prevent damage to the system components and could resolve unscheduled outages. Faults and failures can affect the dynamic operation of the system in different forms, such as intermittent behaviors and incipient or abrupt failures that can be defined as a permanent interruption of system abilities. There are three main methods to enable the early detection of faults and failures; these include a knowledge-based method, a data-based approach, and a model-based approach.

In the first method, the knowledge-based approach can be referred to as fuzzy logic and rule-based inference, which are generally defined from an expert operator that can implement very simple control logics that may detect when the system operates at abnormal conditions [1,2]. The general idea is to reduce the need of the operator when the system operates around specific conditions by implementing a set of rules that can be added and removed easily. The main advantage of this approach is that there is no need for a model of the process. However, development and maintenance of the control logic can be as costly as having a reliable and knowledgeable expert [3].

In the second method, the data-based approach can be generally used to detect faults and failures only by exploiting available historical data and by employing techniques such as principal components analysis (PCA), spectrum analysis, and pattern recognition using neural networks or machine learning techniques [4–6]. For instance, Sun et al. discuss a dynamic PCA suitable for boiler leak detection that reduced the false alarms in the existing monitoring system [7].

In the third method, the model-based approach is used through a variety of techniques such as computational observers (i.e., Kalman filters) and parameter estimation techniques for fault detection [8]. Addel-Geliel et al. applied a model-based computational observer for fault detection in the model of an industrial boiler [9]. The model-based observer was implemented in parallel to the process to predict the dynamic behavior of the system. A fault condition was detected when the prediction mismatched the nominal behavior of the system from an acceptable condition.

The primary downside of a model-based technique is that it requires developing an accurate model that can predict the dynamic performance at each operating condition. Although alternative activities were focused on combining multi-model-based techniques to detect faults and failures at different operating conditions [10], the model-based approach is still a challenging technique because it requires the implementation of multiple models at several operating points. Recognizing this issue, Chen et al. proposed using recursive parameter estimation algorithms to identify system dynamics and model parameters at different operating conditions

[1]Corresponding author.

Contributed by the Advanced Energy Systems Division of ASME for publication in the JOURNAL OF ENERGY RESOURCES TECHNOLOGY. Manuscript received October 16, 2019; final manuscript received December 19, 2019; published online January 29, 2020. Assoc. Editor: Ashwani K. Gupta.

[11]. This approach could overcome the limitation of developing a model at each operating condition by updating or adapting the parameters of the model by running a recursive estimation algorithm when the system operates at different conditions [12–14].

To combine the fidelity of a model-based technique with the flexibility of a recursive algorithm, we developed a monitoring tool that updates the parameters of an empirical model in real time to detect faults and failures. An empirical transfer function model is generally used to design single-input single-output computational controllers, but in this approach, the monitoring tool was used to represent the main component of the model-based technique. As suggested in previous work, the model was implemented in parallel to the process to predict the dynamic behavior of the system [9]. In contrast to previous work, the transfer function parameters were estimated only during nominal operations with a recursive linear least square algorithm that was run for a given period of time. When the calculations of the parameters converged to new steady-state values, they were fed into the transfer function model for online prediction. The prediction from the transfer function model in combination with a control logic based on a moving average approach was used to detect when the system operated at nominal or abnormal conditions.

This monitoring tool was implemented and validated at the Hybrid Performance Project (Hyper) at the U.S. Department of Energy's National Energy Technology Laboratory (NETL) [15]. An experimental test was conducted using the bleed-air (BA) valve to emulate the effect of a leak or fouling in the combustor of the gas turbine system. An abrupt fault was emulated by leaking 10% of the working fluid in a single step from the compressor discharge to the atmosphere using the operation of the bleed-air valve. The response of the mismatch between the transfer function prediction and the process measurement was also evaluated by 10 kW electrical load steps that were performed before and after the leak. Before performing the leak test, a scoping test was conducted to evaluate the feasibility of using the monitoring tool with a single 10 kW electric load step.

2 Hardware

The Hyper project at NETL combines physical and virtual components to study the integration of diverse power technologies into a gas turbine cycle [15]. Because of the control issues of hybrid power systems, the Hyper facility has been used for the development and validation of advanced control strategies [16]. Figure 1 shows a schematic diagram of the physical hardware components of the Hyper facility used in this study. The hardware system is represented by a gas turbine recuperated cycle designed for hybrid configurations. Two heat exchangers preheat the compressed air discharged by the turbomachinery with the exhaust gases of the turbine system. A 2.0 m^3 space is located between the heat exchangers and the gas turbine combustor to host the balance of plant components that could be used to represent a hybrid system, such as a fuel cell stack or a thermal energy storage device. Downstream the gas turbine combustor, a mixing volume is located to blend the exhaust of the combustor and the bypass flow from the compressor discharge and from the outlet of the heat exchangers. Then, the mixed flow is expanded in the turbine.

2.1 Gas Turbine Generator. A 120 kW Garrett Series 85 auxiliary power unit was modified for this testing facility. The gas turbine compressor system is composed of a single shaft, directly coupled turbine, and a two-stage radial compressor. The compressor was designed to deliver approximately 2 kg/s of discharge flow and a pressure ratio of 4:1. The gas turbine is coupled to a gear-driven synchronous generator, and the load to the generator is transferred to a 120-kW resistor bank that dissipates the power output of the source. The load bank is used to replicate the real-life demand of the power system. The turbine operates at a nominal speed of 40,500 rpm. A Woodward proportional-integral-derivative (PID) controller acts on the swift fuel valve during start-up operations or to maintain constant turbine speed during electric load perturbations.

2.2 Swift Fuel Valve. The swift fuel valve is electrically actuated, and a high-speed stepper motor rotates a 2.54-cm sonic needle and nozzle, which controls the fuel flow going into the combustor at a 5-ms rate if needed.

2.3 Bleed-Air Bypass Valve. The BA bypass valve exhausts air from the compressor discharge directly into the atmosphere. This valve is also used to ensure enough surge margin to the compressor during start-up. A 15-cm valve with a 7.5-cm body is used as a bleed-air actuator in the Hyper facility. The range of operations is between 100% and 88% of the closing position when the electric load is engaged to the turbine. During start-up and nominal condition operations, the valve is set at 94% to provide a sufficient surge margin to the compressor. For this study, the valve was also used to emulate leaks from the compressor discharge to the atmosphere, and it was opened from a 94% closed position to 90% in a single step.

2.4 Turbine Speed Optical Sensor. Three optical sensors were installed to measure turbine rotational speed, which is used as feedback for the PID controller. Each sensor optically acquires the light reflected from the rotating target on the end of the generator shaft. This signal is then transmitted to the control system at a 5-ms rate and averaged.

2.5 Industrial Control Platform. The Woodward industrial control platform MicroNet Plus is used at the Hyper facility to control the entire gas turbine process. The MicroNet Plus is a digital control system that utilizes a 400-MHz Motorola MPC5200

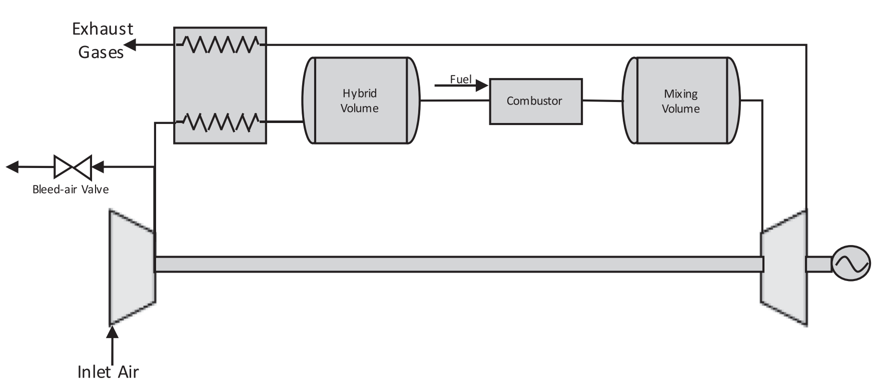

Fig. 1 Gas turbine cycle diagram of the hybrid performance facility at NETL, Morgantown, WV

microprocessor designed for control applications such as gas turbines, steam turbines, hydro turbine control, and diesel and gas engine control. The principle characteristic is the robustness to control or monitor all process variables simultaneously, including turbine rotational speed, load control, combustion control, anti-surge control, and filtering. The MicroNet computer can provide a time response and time stamping equal to 5–10–20–80–160 ms. Control algorithms can be implemented using the Graphical Application Programming (GAP) language, which is a block-oriented programming language. Algorithm iterations using the GAP are limited to only a 5-ms time response; otherwise, it has to be implemented in C functions or MATLAB functions that convert MATLAB code into C++.

3 Design of the Monitoring Tool

In previous work, an adaptive tracking technique based on a recursive parameter estimation algorithm was implemented by Lopez et al. to estimate changes in the system parameters due to degradation in the health condition [17]. The method presented was capable of running as a continuous monitoring tool or as an adaptive health recognition mechanism tracking slow changes due to natural and abrupt degradation effects on the system parameters caused by spontaneous damage events. In this work, a recursive parameter estimation algorithm was still used as an adaptive health recognition mechanism, but it differs from Lopez et al. in that it was combined with a model-based technique that could provide a better evaluation of failures or degradation effects during the dynamic operation of a gas turbine system. In addition, instead of running the recursive algorithm continuously and evaluating a change in the system parameters continuously, new parameters were only estimated when needed.

3.1 Model Implementation in Parallel to the Process. In this monitoring tool, the flexibility of a recursive algorithm was combined with the fidelity of a model-based technique, which was represented by an empirical transfer function model. Figure 2 shows the architecture that was implemented in the industrial control platform of the Hyper facility to predict the dynamic operation of the gas turbine system. An empirical transfer function model was implemented in parallel to the fuel valve/turbine speed response to provide a real-time prediction of the dynamic system. An empirical auto-regressive exogenous (ARX) transfer function is a single-input single-output model that relates input–output data using previous time steps. The general structure of the ARX transfer function establishes a relationship between the response of an actuator and the measurement of a process variable, as shown in Eq. (1) [18].

$$y(t) + a_1 \cdot y(t-1) + \cdots + a_n \cdot y(t-n_a) = b_1 \cdot u(t-1) + \cdots + b_n \cdot u(t-n_b) \tag{1}$$

In Eq. (1), y and u represent the process variable and the actuator response, respectively, at the previous time steps, and a_i and b_i represent the transfer function parameters that must be estimated to reproduce an accurate model of the process. Using z^{-1} as a backward shift operator to generalize previous time steps, Eq. (1) can be

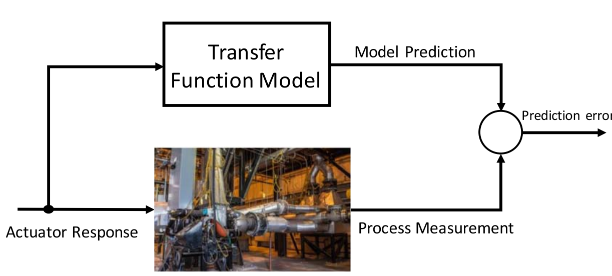

Fig. 2 Monitoring tool architecture that was designed to predict the dynamic operation of the gas turbine system

rearranged in the following format $y(t)+ a_1 \cdot y(t) \cdot z^{-1} + \cdots + a_n \cdot y(t) \cdot z^{-n_a} = b_1 \cdot u(t) \cdot z^{-1} + \cdots + b_n \cdot u_{(t)} \cdot z^{-n_b}$. Then, Eq. (1) can be rewritten as Eq. (2) by grouping $y(t)$ and $u(t)$ to each side of the equation and defining a discrete-time transfer function model as the relationship between the response of a process variable over the actuator response $y(t)/u(t)$.

$$G(z) = \frac{Y}{U} = \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \cdots + b_n \cdot z^{-n_b}}{1 - a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \cdots + a_n \cdot z^{-n_a}} \tag{2}$$

As previously mentioned, z^{-1} represents a backward shift operator that is used to generalize the plant measurement and the actuator command time steps using a z-transform variable, which relates the time domain with the frequency domain.

3.2 Recursive Parameter Estimation Algorithm. An accurate transfer function model is generally obtained with an accurate estimation of the transfer function parameters a_i and b_i. These parameters are generally determined through a mathematical cost function that minimizes the difference between the plant measurement and the model output estimation. Equation (3) shows the general cost function that is used to minimize the difference between the model output and the process measurement [19].

$$\min_{\theta} J(\theta) = \frac{1}{N} \cdot \sum_{t=1}^{N} \| y(t) - y_{est}(t, \theta) \|^2 \tag{3}$$

The cost function is an iterative mean square error calculation on the full data set that evaluates the difference between the model estimate, $y_{est}(t)$, and the process measurement, $y(t)$, at each iteration. The optimal parameters in Eq. (3) are determined by calculating the minimum value of the objective J in the function of the vector θ, which groups all the transfer function parameters that must be calculated $\theta = \left[a_1 \quad \cdots \quad a_{n_a} \vdots \; b_1 \quad \cdots \quad b_{n_b} \right]^T$. The minimum value of the cost function is generally determined by setting the first derivative operator to zero $(dJ(\theta)/d\theta) = 0$. Equation (3) can also be rewritten by grouping the process variables time steps in the vector $\varphi(t) = [-y(t-1) \cdots -y(t-n_a) \vdots u(t-1) \cdots u(t-n_b)]^T$, where the number of data points (n_a and n_b) is related to the order of the model. Hence, Eq. (4) represents the objective J in the function of θ and $\varphi(t)$.

$$\min_{\theta} J(\theta) = \frac{1}{N} \cdot \sum_{t=1}^{N} \| y(t) - \underbrace{\varphi^T(t) \cdot \theta}_{y_{est}(t)} \|^2 \tag{4}$$

From Eq. (4), $y(t)$ represents the process variable measured at the current time step, whereas $y_{est}(t)$ represents the estimation of the process variable at the current time step but using time data of the previous time step. Equations (3) and (4) represent the basis of the recursive least square algorithm, which is focused on iteratively minimizing the difference between the model output estimation and the process measurement $[y(t) - \underbrace{\varphi^T(t) \cdot \theta}_{y_{est}(t)}]$. Equation (5) represents the recursive least square estimation of the transfer function parameters that can be implemented online without matrix inversion.

$$\theta_{est}(t) = \theta_{est}(t-1) + K(t) \cdot [y(t) - \underbrace{\varphi(t-1) \cdot \theta_{est}(t-1)}_{y_{est}(t)}] \tag{5}$$

Optimal parameters are calculated at each iteration as new data is available. At each time step, the parameters estimated at the previous iteration $\theta_{est}(t-1)$ are modified based on the difference between the process measurement variable $y(t)$ at the current time step and the model output estimation $y_{est}(t) = \varphi(t-1) \cdot \theta_{est}(t-1)$ calculated at the current time step but with the parameters determined at the previous time step. At each time step, this difference is adjusted with a scaling factor $K(t)$ presented in Eq. (6), which provides an

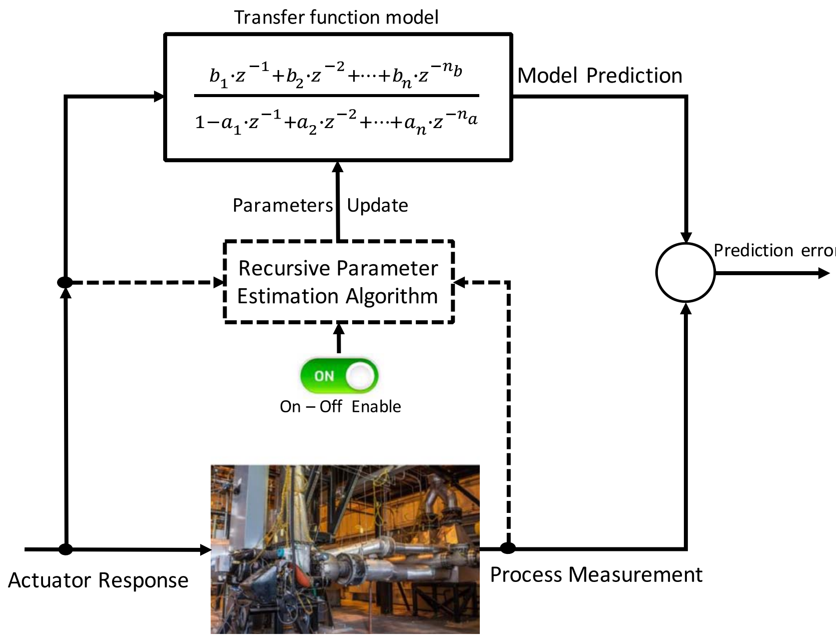

Fig. 3 Monitoring tool architecture including the switch that allows turning on and off the recursive parameter estimation algorithm

iterative weighting vector that weights the difference between the process measurement variable and the model output estimation.

$$K(t) = \frac{P(t-1) \cdot \varphi(t-1)}{1 + \varphi^T(t-1) \cdot \; P(t-1) \cdot \varphi(t-1)} \tag{6}$$

In Eq. (6), $\varphi(t-1)$ represents the vector with the input–output data points of the process, whereas $P(t-1)$ is a matrix updated at each iteration that is structured to avoid a real-time matrix inversion [19]. For example, in the first iteration if no prior knowledge of the process is known, $P(t)$ can be initiated to a large diagonal matrix; by doing this, the algorithm will calculate the best first guess by dividing the numerator, which is a vector, by a simple scalar value. Equation (7) shows the definition of $P(t)$.

$$P(t) = \{P(t-1) - [K(t-1) \cdot \varphi(t-1) \cdot P(t-1)]\} \tag{7}$$

In the recursive least square algorithm, $P(t)$ is updated as new data becomes available. Over time, a large number of data can make $P(t)$ conservative to old data; on the other hand, a small number of data can make $P(t)$ reluctant to learn from new data.

3.3 On-Off Implementation of the Recursive Algorithm. As previously mentioned, instead of running the recursive algorithm continuously and evaluating a change in the system parameters continuously (Lopez et al.), in this work, new parameters were only estimated when needed. If the recursive algorithm was continuously running, the parameters were also continuously optimized by tracking the continuous change in the state of the system based on dynamic operations. The downside to this approach is that evaluating a change in the parameters during optimization may not be used to determine a significant control logic that could detect abnormal conditions for multiple operating ranges.

Thus, in this work, the parameters were only identified during normal conditions or when the system operated at a new operating point. The recursive algorithm only ran for a limited period of time, and the new parameters were fed to the transfer function model when they converged to new steady-state values. The algorithm was stopped when there was not a significant difference in the calculations of the parameters between the two following iterations. This approach significantly simplified the development of the transfer function model parameters at each operating condition, and it made it possible to capture sudden failures or degradation effects when drifting in the transfer function prediction occurred. Using this approach, the operator has the option to calculate new parameters when the system operates at multiple operating ranges, for instance at a different electric load operating point. Figure 3 shows the architecture of this monitoring tool with the switch that allows turning on and off the recursive parameter estimation algorithm.

3.4 Threshold Logic. As previously mentioned, it was found to be more effective to combine the estimation of the multiple parameters into a transfer function model and obtain a single prediction output. Then, this prediction output was compared against the process measurement to establish a constant threshold logic that could confirm normal conditions or detect failure or abnormal operations. In general, the development of this threshold logic represents one of the main concerns in any monitoring tool because it can fail in detecting abnormal conditions or it can generate false alarms that are useless to the operator.

In this monitoring tool, as shown in Fig. 4, the threshold logic was developed downstream the calculation of the dynamic response of the prediction error. The prediction error between the process measurement and the transfer function model was fed as an input to the threshold logic. Due to noise and dynamic operations of the system, as shown in Eq. (8), the prediction error was filtered through the implementation of a recursive moving average calculation to prevent false alarms.

$$S[t] = S[t-1] + Pe\left[t + \left(\frac{W}{2}\right)\right] - Pe\left[t - \left(\frac{W}{2}\right)\right] \tag{8}$$

In Eq. (8), Pe represents the prediction error, S represents the iterative summation, and W represents the window size. By using the recursive formulation of the moving average, the number of calculations is reduced by adding the newest available data point in the moving window and subtracting the last data point from the window. Then, at each iteration, the sum is divided by the size of the moving window.

The output of the moving average was squared to magnify the original prediction error and to have only positive values that

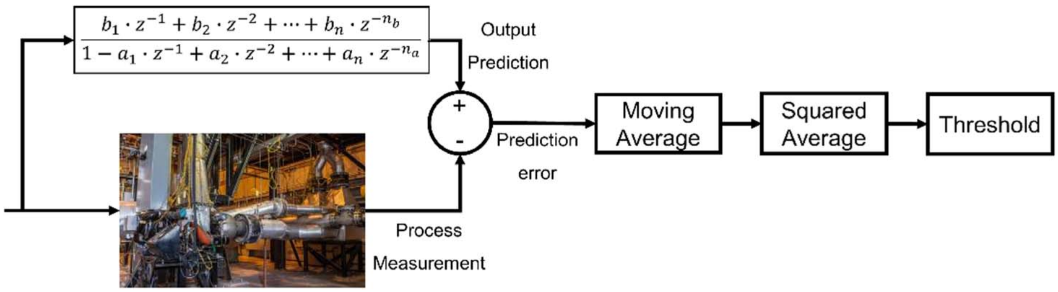

Fig. 4 Detection module used to evaluate the prediction error and detect abnormal operations

could be compared to a single threshold limit. If the square value of the prediction error was within the threshold, it was assumed that the prediction error matched the plant measurement, and the steady-state or normal operation of the process could be confirmed. On the other hand, if the square value of the prediction error was greater than the predefined threshold, it was assumed that a mismatch between the prediction error and the plant measurement occurred. This mismatch was used to detect abnormal conditions during the dynamic operations of the power plant.

4 Derivation of the Monitoring Tool in a Physical Gas Turbine System

Before simulating any leaks in the gas turbine system, a scoping test was conducted to evaluate the feasibility of the monitoring tool. As shown in Fig. 5, a single electric load step of 10 kW was used to evaluate the feasibility of the algorithm, validate the monitoring tool, and derive the mathematical approach into a practical case. Experimental tests at the Hyper facility are generally performed once the thermal steady-state operation is achieved. The thermal steady-state of the gas turbine system is achieved when the turbine rotational speed operates at the nominal setpoint of 40,500 rpm, and the skin temperature of the mixing volume varies less than 1.0 K for a 30-s period. During start-up and around nominal conditions, a single-input single-output PID controller controls the turbine speed by handling the fuel flow into the combustor. At the nominal turbine speed operation, an electric load bank is used to engage a 40-kW load to the generator shaft.

4.1 Model Implementation in Parallel to the Process. The challenge of implementing an identification algorithm is based on selecting the model structure and the order that best describes the process [20]. The model selected must be accurate enough to reproduce adequate characteristics of the physical process. In general, higher-order models provide better accuracy of the physical process, but the order selected cannot be too high; otherwise, it may cause uncertainties in the parameters that can affect the accuracy of the model. As shown in Eq. (9), in this work, the order of the model was initialized as a second-order because it was considered the best compromise between accuracy and causing uncertainties in the parameters.

$$G(z) = \frac{Y}{U} = \frac{b_1 \cdot z^{-1} + b_2}{1 - a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \tag{9}$$

4.2 Recursive Parameter Estimation Algorithm. For a second-order model, the size of the vector that contains the process variables was based on four data points, $\varphi(t) = [-y(t-1) \;\; -y(t-2) \; \vdots \; u(t-1) \;\; u(t-2)]^T$. Two previous time steps were used for the process measurement ($y(t-1)$ and $y(t-2)$) and the previous two-time steps for the inputs ($u(t-1)$ and $u(t-2)$). Similarly, the size of the vector that contains the transfer function parameters that were calculated with the recursive algorithm was based on four values $\theta = \begin{bmatrix} a_1 & a_2 & \vdots & b_1 & b_2 \end{bmatrix}^T$, two parameters for the denominator (a_1 and a_2) and two parameters for the numerator (b_1 and b_2).

As shown in Fig. 3, the actuator response, fuel valve position, the process measurement, and the turbine rotational speed were fed as inputs to the recursive algorithm to identify the transfer function parameters for that control loop. Then, as shown in the pseudo-code of Fig. 6, an initialization of the parameters was required before starting the algorithm. Because it was assumed that no prior knowledge of the process was known, the θ vector that contains transfer function parameters was initialized to zero initial condition, $\theta = \begin{bmatrix} 0 & 0 & \vdots & 0 & 0 \end{bmatrix}^T$. Thus, a_1, a_2 and b_1, b_2 were equal to zero before running the algorithm. Then, the $P(t)$ matrix was initialized as a large diagonal matrix. The pseudo-code in Fig. 6 was implemented in the Woodward industrial control platform of the Hyper facility in C language, and it was run at 80 ms time steps.

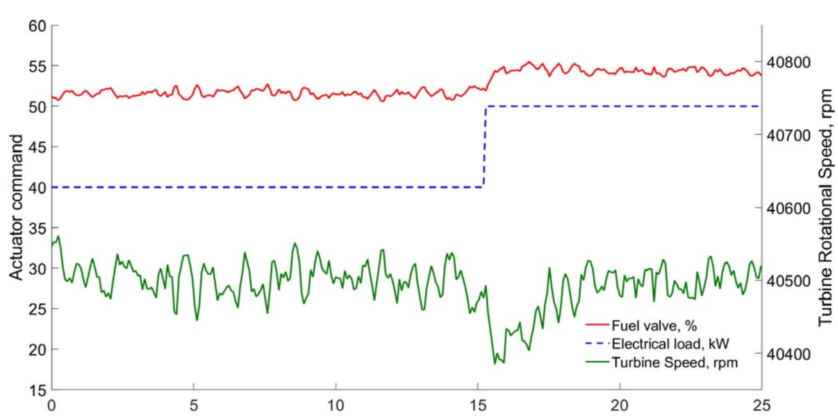

Fig. 5 10 kW Electric load step used to derive and validate the monitoring tool

Initialization of $\varphi(t)$ based on the order of the model
Initialization of model orders n_a and n_b
Initialization of the P matrix

Iterative loop until values converge:

$$K(t) = \frac{P(t-1)\cdot\varphi(t-1)}{1+\varphi^T(t-1)\cdot P(t-1)\cdot\varphi(t-1)}$$

$$\theta_{est}(t) = \theta_{est}(t-1) + K(t)\cdot\left[y(t) - \underbrace{\varphi(t-1)\cdot\theta_{est}(t-1)}_{y_{est}(t)}\right]$$

$$P(t) = \{P(t-1) - [K(t-1)\cdot\varphi(t-1)\cdot P(t-1)]\}$$

End

Fig. 6 Pseudo-code of the recursive parameter estimation algorithm

4.3 On-Off Implementation of the Recursive Algorithm. The recursive algorithm was started when the nominal state of the gas turbine system was confirmed. When the turbine rotational speed operated at the nominal setpoint of 40,500 rpm, the skin temperature of the mixing volume had a temperature gradient of less than 0.03 K per seconds, and the turbine electric load was set at 40 kW. As shown in Fig. 7, the recursive algorithm ran for only 10 s and was automatically stopped when the parameters converged to new constant values, or in other words, when there was not a significant difference in the parameter calculations between two following iterations. Figure 7 shows how the parameters changed while the recursive algorithm was running, and Table 1 presents the new parameters identified by the algorithm after 10 s of iteration.

Table 1 Transfer function parameters

Steady-state parameters using the recursive algorithm	$\dfrac{-0.09741\cdot z + 0.09741}{z^2 - 0.9425\cdot z - 0.002317}$

As shown in Fig. 8, when the recursive algorithm was running, the parameters were not fed to the transfer function model and the predicted turbine speed was set at the constant value of 40,500 rpm, which represents the nominal turbine speed operation. After 10 s, when new parameters were determined, they were fed to the transfer function model. Between 10 and 15 s, the turbine operates at the nominal condition and the prediction matched the actual measurement in the turbine speed.

4.4 Threshold Logic. In the threshold logic, the prediction error was evaluated to isolate abnormal conditions or failures affecting the system using the detection architecture shown in Fig. 4. The main goal of this architecture is to produce a fault signal or an alarm that indicates when a fault occurs. As previously discussed and shown in Fig. 9, before triggering an alarm, a filtering technique based on a moving average approach was used to reduce the noise and prevent false alarms in the prediction error. The moving average had a window size of 40 data points; it was determined as the best compromise between the lowest noise in the data and reduction in the sharpness of the signal. The moving average was implemented in the Woodward industrial control platform of the Hyper facility in C language, and it was run at 80 ms time steps.

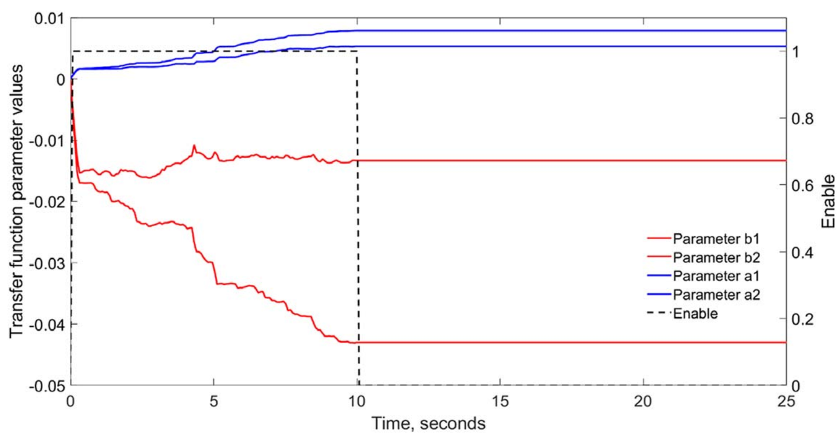

Fig. 7 Transfer function parameters calculation using the recursive linear least square algorithm

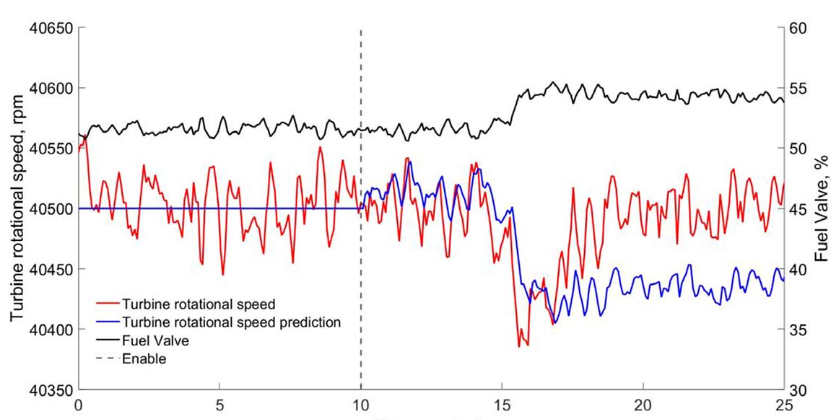

Fig. 8 Turbine speed measurement and prediction after parameters were estimated and fed to the transfer function model

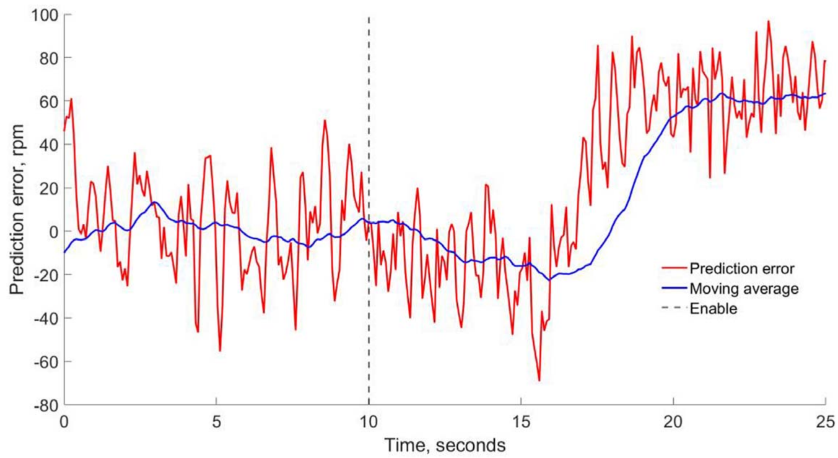

Fig. 9 Prediction error and output of the moving average filter when a 10-kW electric load step was reproduced in the gas turbine system

Figure 8 shows an electric load perturbation from 40 kW to 50 kW applied to the system for around 15 s. Initially, the turbine speed prediction followed the measurement, but a deviation occurred when the PID controller recovered the turbine speed by increasing fuel flow going into the combustor. The prediction does not completely recover 40,500 rpm because this monitoring tool only takes into consideration the amount of fuel used in the control system without the feedback of the turbine speed. In other words, in the control system, the PID controller makes changes in the fuel flow based on the actual turbine speed measurements, but in the monitoring tool, the transfer function model does not have the turbine speed as feedback to make the prediction. However, even though the complete dynamics of the closed-loop system was not reproduced, the change in the magnitude of the turbine speed prediction was still used as an advantage to detect the deviation in the prediction error. Figure 10 showed the averaged squared error that occurred when an electric load change was applied to the gas turbine system.

The advantage of this threshold logic was mainly focused on capturing the reaction of the controller during the dynamic operation of the power plant. During dynamic operations, the controller keeps the turbine speed constant by increasing or decreasing fuel flow, and a deviation can be captured in the prediction error when extra fuel is used to adjust the gas turbine operation.

5 Experimental Methodology

In this work, the fuel valve/turbine speed controller that regulates the dynamic operation of the gas turbine system was monitored using the monitoring tool described in the previous sections. Because the bleed-air valve discharges compressed air into the atmosphere, it was used to reproduce a potential failure that could simulate a leak in the working fluid of the gas turbine operation between the compressor and the gas turbine combustor. When an opening step in the bleed-air valve is performed, compressed air is discharged into the atmosphere and less thermodynamic power is provided to the turbine to generate the same amount of energy. During this operation, the gas turbine controller increases the amount of fuel flow going into the combustor to compensate for the loss of compressed air in the system.

As shown in Fig. 11 an abrupt fault was simulated by leaking 10% of the working fluid in a single step. The bleed-air valve was opened from a 94% closed position to 90% when the system operated at the nominal turbine speed operation and at thermal steady-state, and when it generated 40 kW of electric load. Before and after the bleed-air valve was changed, 10 kW electric load steps were also used to characterize the response in the prediction of the transfer function. The electric load changes created perturbations in the operation of the gas turbine system that was handled by the fuel valve/gas turbine speed controller. These electric load changes emulated unpredictable perturbations experienced during power plant operations and were used to study the threshold value needed before and after the leak during dynamic operations of the gas turbine system. As shown in Fig. 11, before the leak, three series of electrical load changes were repeated. A single series consisted of 10 kW steps up and down, and each change was held for 60 s.

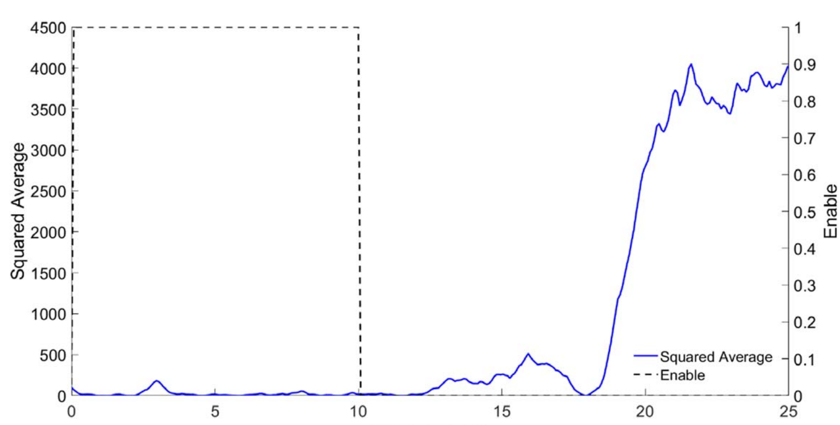

Fig. 10 Averaged squared error when an electric load change was applied to the gas turbine system

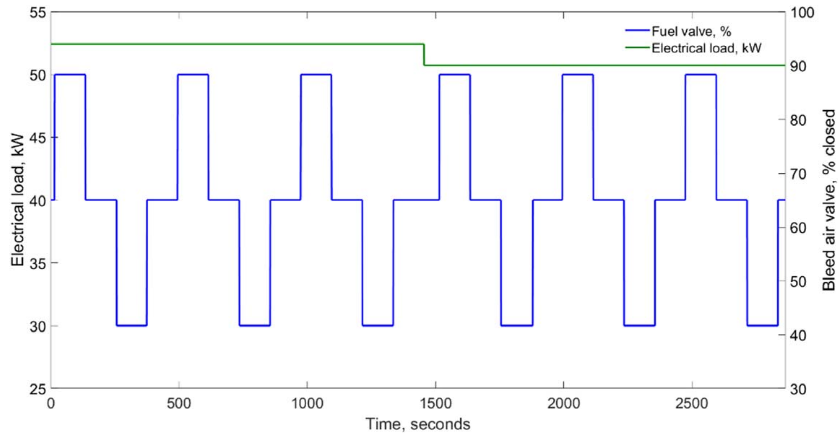

Fig. 11 Electric load steps and leak emulation during turbine operations

6 Results and Discussion

The monitoring tool developed in this work was implemented and validated in the control system platform of the Hyper facility at NETL to monitor the real-time performance of the recuperated gas turbine system. In the experimental tests, the turbine speed was maintained constant by a single-input single-output controller that changed the fuel flow in the gas turbine combustor to compensate for electric load perturbations or the simulated leak in the working fluid of the process. As shown in Fig. 11, multiple steps in the electric load operation around nominal conditions were reproduced to evaluate dynamic prediction from the monitoring tool. As shown in Fig. 12, a larger amount of fuel flow was used when the leak was reproduced at 1456 s. The bleed-air valve was opened by 4% in a single step and right after that change, the controller increased the amount of fuel flow going into the combustor to maintain turbine speed at the nominal setpoint and to compensate for the compressed air discharged into the atmosphere.

The amount of fuel flow used by the controller had a direct effect on the prediction error of the monitoring tool. Compared to the electric load steps performed before the leak, the controller was outputting more fuel than before the leak. As shown in Fig. 13, a spike in the average squared error indicated that a fault occurred in the system. After the leak and during the nominal electrical load operation (i.e., 40 kW), the fuel valve operation increased by 4.3%, which accounted for 15.8% of additional fuel flow used in the combustor. In addition, after the leak, the amplitude in the average squared error was higher than before the leak for the same amount of perturbations in the electrical load. This higher deviation can be used to confirm that the system operated with a potential failure. The threshold was setup at an average square value of 10,000, which allowed the monitoring tool to capture the leak 7 s after the valve was opened.

7 Conclusions

Due to the increasing penetration of renewables and the continued use of power plant cycling, the detection of potential failures, performance drifting, and degradation of critical components has become a critical issue to be resolved. In this work, a monitoring tool based on a combination of a model-based technique and a recursive parameter estimation algorithm was developed to detect performance deterioration due to a leak in the working fluid of a modified micro-gas turbine system. Experimental tests at the U.S. Department of Energy's National Energy Technology Laboratory (NETL) Hyper project were performed to implement and validate the sensitivity of the tool when 10% of a leak in the working fluid was emulated by opening the bleed-air valve in a single step. Electrical load steps were also performed before and after the leak to characterize the dynamic prediction of this monitoring tool.

The advantage of this monitoring tool was mainly focused on evaluating the response of the controller during dynamic operations by capturing a larger amount of fuel flow when a fault event occurred. In general, the controller regulates the turbine speed by increasing or decreasing the amount of fuel flow going into the gas turbine combustor. This monitoring tool captured the larger amount of fuel flow by calculating the prediction error between the process measurement and the prediction of a transfer function model that was implemented in parallel to the process. The monitoring technique proposed in this work was mainly

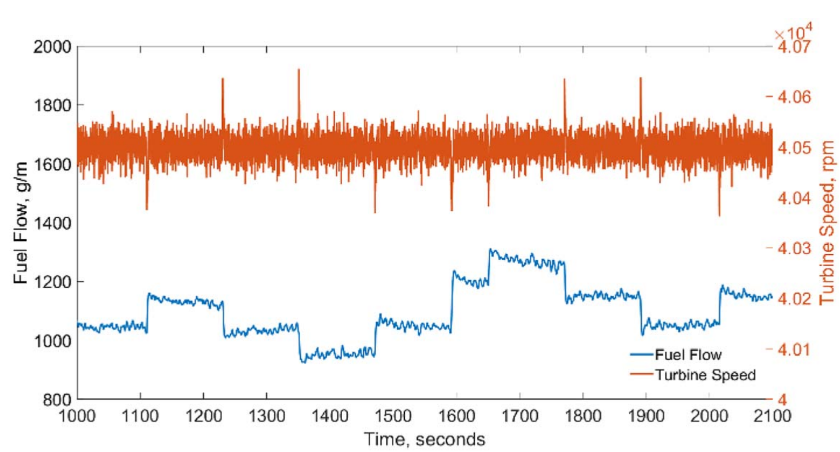

Fig. 12 Turbine speed and fuel flow response before and after a leak

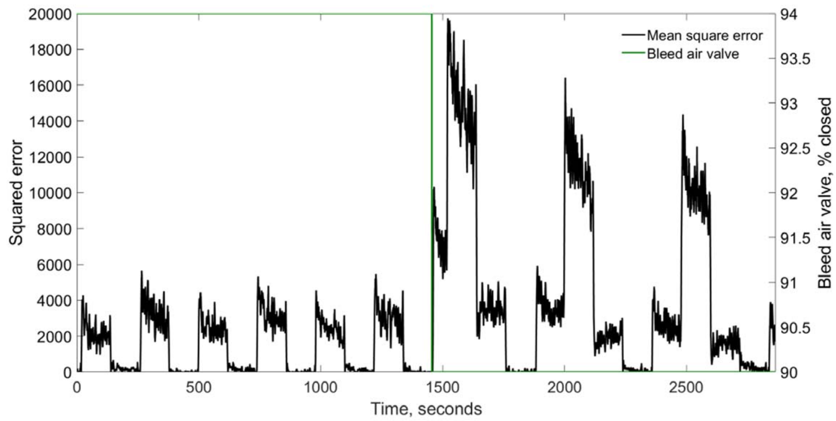

Fig. 13 Response of the averaged squared error for the prediction error before and after the leak was emulated

focused to run the parameter estimation algorithm only for a given period of time and pass the parameters to the transfer function model when they converged to new steady-state values. This approach significantly simplified the development of a transfer function model at each operating condition. The prediction error was calculated and then filtered into a moving average algorithm that determined the average square error based on 40 data points (i.e., the last 3 s of operation). Deviation from nominal operations was used as an indicator of abnormal conditions based on a threshold that was set up large enough to tolerate load-following operations during nominal conditions, but still capable of capturing failures.

Results showed that the monitoring tool was able to detect a 10% leak in the working fluid of the gas turbine system by detecting an increase in the amount of fuel flow of 15.8% to maintain the turbine speed at nominal operations. The fault was detected seven seconds right after the leak occurred, which indicated to the operator that the system had a failure within that control loop due to the mismatch observed in the prediction of the transfer function model. This monitoring tool provided a beneficial approach compared to other model-based techniques previously discussed because it provides the flexibility to develop the model parameters in real-time starting from zero initial condition without affecting the operation of the process.

8 Future Work

Future work will extend this monitoring tool to define an adaptive threshold logic that can detect failures during variable power plant operations. Generally, most of the monitoring tools found in power plants apply a fixed value threshold such as the one used in this paper. However, this approach can create false alarms, and adaptive threshold techniques would provide a more feasible option because it considers different operating conditions of the plant. For example, the adaptive threshold approach will consider that the threshold at full or nominal condition may have a different value compared to when the system operates at part-load conditions.

Acknowledgment

This research was supported by the US Department of Energy—Office of Fossil Energy under Contract No. DE-AC02-07CH11358 through the Ames Laboratory.

Nomenclature

a = transfer function parameters in the denominator
b = transfer function parameters in the numerator
n = number of parameters
t = time
u = actuator response
y = process measurement variable
G = transfer function
J = objective function
K = weight matrix
N = number of data points
P = covariance Matrix
S = recursive summation
W = window size
est = estimation
Pe = prediction error
θ = vector including transfer function parameters
φ = input–output data points

References

[1] Miljković, D., 2011, "Fault Detection Methods: A Literature Survey," Proceedings of the MIPRO 2011–34th International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, May 23–27, pp. 750–755.
[2] Toffolo, A., 2009, "Fuzzy Expert Systems for the Diagnosis of Component and Sensor Faults in Complex Energy Systems," ASME J. Energy Resour. Technol., **131**(4), p. 042002.
[3] C. Angeli, 2010, "Diagnostic Expert Systems: From Expert's Knowledge to Real-Time Systems," Advanced Knowledge Based Systems: Models, Applications & Research, Technomathematics Research Foundation, Kolhapur, India, Vol. 1, pp. 50–73.
[4] Garcia-Alvarez, D., Fuente, M. J., Vega, P., and Sainz, G., 2009, "Fault Detection and Diagnosis Using Multivariate Statistical Techniques in a Wastewater Treatment Plant," IFAC Proc. Volumes, **42**(11), pp. 952–957.
[5] Gong, X., and Qiao, W., 2011, "Bearing Fault Detection for Direct-Drive Wind Turbines via Stator Current Spectrum Analysis," Proceedings of the 2011 IEEE Energy Conversion Congress and Exposition, Phoenix, AZ, Sept. 17–22, pp. 313–318.
[6] Heo, S., and Lee, J. H., 2018, "Fault Detection and Classification Using Artificial Neural Networks," IFAC-PapersOnLine, **51**(18), pp. 470–475.
[7] Sun, X., Marquez, H. J., Chen, T., and Riaz, M., 2005, "An Improved PCA Method With Application to Boiler Leak Detection," ISA Trans., **44**(3), pp. 379–397.
[8] Wang, J., Zhang, Y., Li, J., Xiao, P., Zhai, Z., and Huang, S., 2017, "A New Approach for Model-Based Monitoring of Turbine Heat Rate," ASME J. Energy Resour. Technol., **139**(1), p. 012004.
[9] Addel-Geliel, M., Zakzouk, S., and El Sengaby, M., 2012, "Application of Model Based Fault Detection for an Industrial Boiler," Proceedings of the 20th Mediterranean Conference on Control and Automation, MED 2012, Barcelona, Spain, July 3–6, pp. 98–103.
[10] Farber, J. A., and Cole, D. G., 2018, "Using Multiple-Model Adaptive Estimation and System Identification for Fault Detection in Nuclear Power Plants," Proceedings of the ASME 2018 Internation Mechanical Engineering Congress and Exposition, Pittsburgh, PA, Nov. 9–15, pp. 1–9.
[11] Chen, J., and Patton, R. J., 1999, *Robust Model-Based Fault Diagnosis of Dynamic Systems*, Springer, Boston, MA.
[12] Sun, X., Chen, T., and Marquez, H. J., 2002, "Boiler Leak Detection Using a System Identification Technique," Ind. Eng. Chem. Res., **41**(22), pp. 5447–5454.
[13] Mohammadpour, J., Grigoriadis, K., Franchek, M., and Zwissler, B. J., 2010, "Real-Time Diagnosis of the Exhaust Recirculation in Diesel Engines Using Least-Squares Parameter Estimation," ASME J. Dyn. Syst. Meas. Contr., **132**(1), p. 011009.

[14] Kim, Y. M., 2015, “Threshold Selector for Fault Detection on Closed-Loop Predictor-Based Recursive System Identification,” Int. J. Control Autom. Syst., **13**(6), pp. 1375–1381.
[15] D. Tucker, P. Pezzini, and K. M. Bryden, 2018, “Cyber-physical Systems: A new Paradigm for Energy Technology Development,” Proceedings of the ASME 2018 Power Conference, Lake Buena Vista, FL, June 24–28, Vol. 1, pp. 1–10.
[16] Pezzini, P., Bryden, K. M., and Tucker, D., 2018, “Multicoordination Control Strategy Performance in Hybrid Power Systems,” ASME J. Electrochem. Energy Convers. Storage, **15**(3), p. 031007.
[17] Lopez, I., and Sarigul-Klijn, N., 2008, “System Identification and Damage Assessment of Deteriorating Hysteretic Structures,” Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Schaumburg, IL, Apr. 7–10, pp. 1–14.
[18] Restrepo, B., Bonilla, H., Pezzini, P., Bryden, K., and Tucker, D., 2018, “PID Control Design and Demonstration Using a Cyber-Physical Fuel Cell/gas Turbine Hybrid System,” Proceedings of the ASME 2018 Power Conference, Lake Buena Vista, FL, June 24–28, Vol. 1, pp. 1–11.
[19] Ljung, L., 1999, *System Identification: Theory for the User*, 2nd ed., Prentice Hall PTR, Saddle River, NJ.
[20] Makara, K., Jérémi, R., and Jean, F., 2009, “On-Line Parameter Estimation of PMSM in Open Loop and Closed Loop,” Proceedings of the IEEE International Conference on Industrial Technology, Churchill, Victoria, Australia, Feb. 10–13, pp. 1–6.