41st IEEE CDC · Las Vegas, NV, December 10 - 13, 2002

Model-based Diagnostics for Small-scale Turbomachines

Dimitry Gorinevsky*, Emmanuel Nwadiogbu†, Dinkar Mylaraswamy‡

Honeywell International

Abstract

This paper describes a case study of model-based diagnostics system development for an aircraft Auxiliary Power Unit (APU) turbine system. The off-line diagnostics algorithms described in the paper work with historical data of a flight cycle. The diagnostics algorithms use detailed turbine engine systems models and fault model knowledge available to an engine manufacturer. The developed algorithms provide fault condition estimates and allow for consistent detection of incipient performance faults and abnormal conditions.

1 Introduction

This paper provides a case study in development of model-based diagnostics algorithms for a small turbomachine used in aerospace applications. Being a safety-critical system, it is an industry standard that an electronic control unit of a turbomachine includes build-in-test and fault diagnostics functionality. This paper is focused on a different application – diagnostics of incipient faults. Such faults often exhibit themselves as a deterioration trend in the turbomachine performance and eventually lead to a need to perform expensive repair and overhaul activities. Timely detection of incipient faults enables preventive maintenance and has significant economic importance.

Accurate and reliable detection and parameter estimation of incipient faults requires detailed and thorough understanding and knowledge of the equipment. Such understanding is available to Honeywell as a turbine engine manufacturer in the form of detailed design and control analysis models. This paper describes diagnostics algorithms based on such detailed models of the engine performance and dynamics. Model-based approaches to diagnostics of controlled systems have been discussed in many papers, e.g., see the surveys in [10, 7, 3, 4]. The choice of a particular technical approach for diagnostics in a particular application is defined by practical engineering considerations such as reliability, availability of models, availability of data, available computational resources, development and support cost. Some of the applications of model-based diagnostics to larger turbomachines are discussed in [2, 8].

Though the approach described is sufficiently general, the focus of this paper is the Honeywell Auxiliary Power Unit (APU) 331-400 that is used in Airbus 320 aircraft. The APU is a small turbomachine, auxiliary to the main propulsion engines. It is connected to an electrical generator and provides power when aircraft is on the ground. It also provides compressed bleed air that is being used by several aircraft subsystems; the most important use is for starting the main propulsion engines of the aircraft.

APUs and small propulsion engines developed, manufactured, and supported by Honeywell are as complex as large propulsion engines but much more cost sensitive. This presents several unique challenges in terms of developing practical diagnostics system for such engines. First, there are few on-line sensors. Thus, diagnostics of many faults have to be performed through indirect measurement data. Second, there is a much larger fleet of operating engines per service personnel compared to larger propulsion engines. This makes saving human expert labor on inspection of the engine physical state a much more important business issue. There is therefore, a business need for an automated computer- based diagnostics system. Since the main goal of the diagnostic system is to automate on-ground inspection of the engine, using engine data logged through a flight cycle does not impose any system performance limitations. The data can be used in a batch mode, there is no need to process the data on-line as it arrives. This is also consistent with the goal of detecting incipient faults. Even if such fault is detected, the engine service can be performed only at the end of the flight cycle, on the ground.

An important feature of the APU diagnostics is that the turbine shaft speed is mostly maintained constant by the control system and changes only during the APU start. Further, the full load on the machine is usually exercised only once or twice in every flight cycle, i.e., when the APU performs the main engine start and has to operate under maximal load. The performance deterioration could become most visible and can be estimated most reliably at the time of the speed or load transient. The data for the fault estimation is available as a batch after the flight cycle. The fault estimation is performed off-line using this data. In the application example of this paper, the particular segment of data used in the identification covers the time of the APU start.

2 Modeling

This paper distinguishes between two types of models that are closely interrelated but functionally different. The first one is a *simulation model* that is used for control and diagnostic system development, verification and validation. The second one is a *prediction model* that uses the system input data to predict the outputs.

* Honeywell Global Control Laboratory, 1 Results Way, Cupertino, CA 95014; email: gorinevsky@ieee.org; dimitry.gorinevsky@honeywell.com

† Honeywell Engine Systems and Services, Phoenix, AZ

‡ Honeywell Global Control Laboratory, Minneapolis, MN

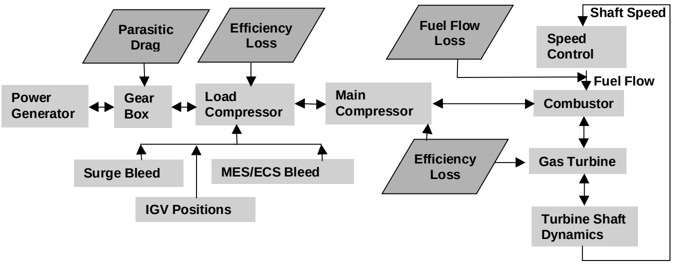

Figure 1: Component-level model of the engine showing the faults discussed in the paper.

The engine is modeled as a sequence of interconnected components as shown in Figure 1. The displayed connection of the main blocks roughly corresponds to continuity of gas flow and power transmission through the main shaft of the APU.

Ambient air entering the engine is split at the plenum, a part enters the main compressor, and the remainder enters the load compressor. The flow of air through the load compressor is regulated by inlet guide vanes. High-pressure air from the main compressor enters the combustor, where fuel is injected through annular nozzles. Hot gases from the combustor expand in the gas turbine, which drives the engine shaft. Part of this useful work is expended by the main compressor. The remaining part of the useful shaft work is expended by the load compressor to provide bleed air and is partially available at the generator to meet electrical needs of an aircraft. Exhaust air from turbine is expelled outside the aircraft.

The controller shown as a block with a double-line boundary regulates the fuel flow through the engine combustor. The skewed parallelograms indicate some of the faults considered in the model. The simulation model, unlike the prediction model, does not include controller and uses the measured fuel flow as one of the inputs.

2.1 Fault Modeling

Many of the incipient faults that result in the deterioration of the engine performance can be modeled as adverse changes in the parameters that affect compressor efficiency, turbine efficiency and parasitic torque on the shaft. The current model uses lumped parameters like efficiencies to capture performance deterioration. Consequently, incipient fault detection can be performed by identifying deviation of the performance parameter from its nominal values. One of the longer-term goals of our work is to incorporate additional models of the material deterioration, surface erosion, and aerodynamical effects to gain deeper insight into the root cause of performance deterioration. Such detailed modeling enables to reason the cause of the performance deterioration without removing and disassembling the engine.

In this work, the faults were modeled as parameter changes that are constant over the data collection batch. The skewed parallelograms in Figure 1 indicate the faults included in the model in this project. These fault parameters are

- p_1 – performance loss for the power section.
- p_2 – performance loss for the load compressor (LC).
- p_3 – constant parasitic torque drag on the main shaft.
- p_4 – fuel system nozzle degradation flow reduction

These fault parameters make a fault parameter vector $p = [p_1 \;\; p_2 \;\; p_3 \;\; p_4]^T$. The same fault descriptions were used in the simulation model and in the prediction model, though the values of the fault parameters seeded in the simulation model were not available to the diagnostics algorithms employing the prediction model. The algorithms had to estimate the fault parameters from the available output data.

2.2 Two-time scale dynamics

The component-level model in Figure 1 describes heat and mass transfer in the gas flow through the engine (gas path dynamics) as well as the load balance on and acceleration of the main shaft (shaft dynamics). The time constants of the gas path dynamics are on the order of milliseconds. The shaft dynamics are much slower and have sub-second scale. The system with such different time scales is a "stiff" system from the point of view of the computer simulation software. Mathematically, the system can be described as a singularly perturbed system of the form

$$\begin{aligned} \dot{x} &= f(x, z, w, t; p), && (1)\\ \epsilon \dot{z} &= g(x, z, w, t; p), && (2)\\ y &= h(x, z; p), && (3) \end{aligned}$$

where x collects the state vector components corresponding to the slow dynamics, z is a part of the state vector corresponding to the fast dynamics, vector y describes the observations (readings of the available sensors), and vector w collects the external system inputs, including the control input (fuel flow). The small parameter ϵ describes the ratio of the time scales between the slow and fast dynamics. In our case, $\epsilon \approx 10^{-3}$. The vector p describes fault model parameters. Changes in these parameters are indicative of incipient faults.

A typical component based model of a turbine engine, has a single slow state $x = N$, the r.p.m. speed of the engine main shaft. The fast states z includes gas pressures, temperatures and mass flows in the gas path. The inputs w of the model (1)–(3) include measured parameters such as fuel flow in the combustor, ambient pressure and the temperature, electrical and pneumatic load data and some additional sensor measurements. The outputs y include engine speed, exhaust gas temperature (EGT), and readings of load compressor discharge temperature and pressure sensors. Note that the slow state $x = N$ is measured directly. The observation equation (3) can be presented in a more specific form.

$$y = \begin{bmatrix} x \\ Cz \end{bmatrix}, \qquad (4)$$

where C is an observation matrix on an appropriate size selecting the measured components of the vector z

The initial diagnostics algorithm development to follow is based on the deterministic model of the form (1)–(4). The algorithms developed for such deterministic model have certain robustness – resilience – to the sensor noise and measurement error as demonstrated below.

To achieve a statistically optimal estimation of the fault parameters from the noisy data, the noise has to be introduced in the model and taken into account when developing the diagnostics algorithms. To this end, the observation equations (3), (4) can be modified to include a colored noise sequence vector $\xi = \xi(t)$.

$$y \quad = \quad h(x, z; p) + \xi = \left[\begin{array}{c} x \\ Cz \end{array} \right] + \xi \tag{5}$$

The assumed statistical characteristics of the noise ξ are discussed further on in the paper.

2.3 Prediction model, residual computation

Since the gas path dynamics are two-three orders of magnitude faster than the shaft dynamics, the system (1)–(3) can be analyzed as a singularly perturbed system. It is well known (e.g., see [11]) that if the fast dynamics in the singularly perturbed system are stable, then (except for a possible initial transient) its solution is always close to the slow motion manifold obtained by setting $\epsilon = 0$ in the perturbed system. The stability of the gas path dynamics is always maintained in the turbomachine by keeping the system away from the instabilities in the gas path dynamics like the regimes leading to a stall or surge.

For the system (1)–(3) this means a little precision is lost by replacing the differential equation (2) with the following algebraic equation

$$0 \quad = \quad g(x, z, w, t; p), \tag{6}$$

The fast variable z in (1) and (3) at each point in time should be resolved from (6) as an explicit function of x and w. Fast and accurate solution of the algebraic equation (6) is an important part of fault estimation algorithm.

The differential algebraic equation system (1), (3), (6) has only dynamical variables x corresponding to the slow dynamics. It also includes static maps corresponding to the explicit function (6). For the 331-400 APU as well as for most other turbomachines with a single shaft, the slow variable x is a scalar – the engine shaft speed. A possibility of handling first-order dynamics only greatly simplifies development of the diagnostics algorithms as described in the next subsection.

As mentioned above, a small engine has an extremely limited number of sensors that can be used for detection and estimation of faults. The accuracy of the estimation can be however improved by using the dynamical information in the data. Since the slow variable $x = N$ is measured directly, the fault estimation is performed by complementing the observation vector y in (3), (4) with the derivative $\dot{x}$. The next subsection discusses how the engine acceleration $\dot{N} \equiv \dot{x}$ can be estimated directly from the engine speed data. An extended observation vector includes the acceleration. By using (1) and (3) it can be computed as

$$\left[\begin{array}{c} \dot{x} \\ y \end{array} \right] = \left[\begin{array}{c} f(x, z, w, t; p) \\ h(x, z, w; p) \end{array} \right], \tag{7}$$

where in accordance with (4) $x = x(t)$ is directly measured. Further, the input vector $w = w(t)$ in r.h.s. (7) is a part of the available data, and $z = z(t; p)$ can be computed as an explicit function from (6) given $x(t)$ and $w(t)$. Thus, the r.h.s. of (7) can be computed from the available data provided the fault parameters p are known. For $p = 0$, the r.h.s. of (7) gives the prediction model. The prediction model output can be compared against the extended observation vector in the l.h.s. (7) to detect presence of the faults and estimate their parameters.

The described concept of computing a residual of model-based prediction is illustrated in Figure 2. In this figure, the "Static Map" block corresponds to solving the algebraic equation (6) for the fast gas path variables z; the "Shaft Dynamics" block corresponds to the slow dynamics given by (1).

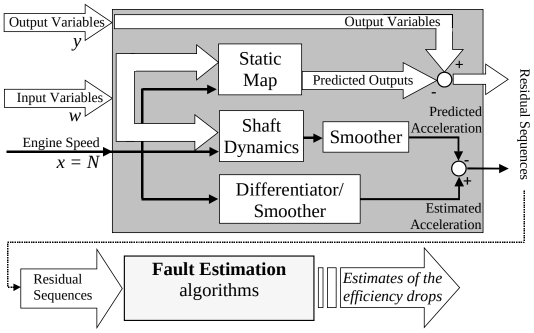

Figure 2: Fault estimation algorithm concept

The differentiator and smoother blocks shown in Figure 2 are described in the next section. The design of the differentiator/smoother algorithms has to take into account noise in the data sequence being differentiated. Since numerical differentiation tends to enhance the noise, a great care is required when designing the differentiator. This is described in some detail in the below.

The fault estimation algorithm mentioned in Figure 2 is described in the next subsection.

2.4 Fault parameter estimation - no noise

The algorithm for identification of the fault parameters work with the prediction residual data as illustrated in Figure 2 is described in this subsection and does not assume a detailed model of the measurement noise. Since only three fault parameters are estimated from a long series of data by a least square error method, the estimation is insensitive to noise. A discussion of the issues associated with the presence and modeling of the measurement noise is discussed in the next two sections of the report.

In particular, the next section describes the design of the FIR differentiator/smoother operator D corresponding to the respective bock in Figure 2, as well as the smoother operator L shown in the same figure. The smoothing operator is defined as

$$L(z) = \frac{1}{1 - z^{-1}} D(z), \tag{8}$$

where z is the discrete time Laplace variable corresponding to one sample shift in discrete time. No matter how the differentiator operator $D(z)$ is computed, it can be represented as a product of a finite difference differentiation $1 - z^{-1}$ and the smoothing $L(z)$.

In accordance with Figure 2 and (17), the following residual sequence is used for estimating the fault parameter vector p.

$$r(t;p) = \left[\begin{array}{c} (Dx)(t) - (Lf)(x,z,w),t;p) \\ y(t) - h(x,z,w;p), \end{array} \right] \tag{9}$$

where time t is assumed to be scaled such that the sampling interval is unity, D and L are related by (17). Obviously in the absence of the measurement noise the residual sequence $r(t; p = 0) = 0$ provided that the data has been generated by the system (1), (3), (6) where $p = 0$. In practice, because of the measurement noise and other model imperfections, the residual is always nonzero.

For estimating fault parameter vector p consider a sequence of the sampled data (9) collected over a time interval $t \in [0,\ N]$ and introduce the following notation

$$R(p) = \left[\begin{array}{c} r(0;p) \\ r(1;p) \\ \vdots \\ r(N;p) \end{array} \right] \tag{10}$$

An estimate of the parameter vector p minimizing the quadratic loss index $\|R(p)\|^2 \rightarrow \min$ can be found numerically. Though, generally speaking, $R(p)$ is a nonlinear function, the incipient faults usually result in a relatively small change of the fault parameters, such as compressor efficiency. Thus, the dependence $R(p)$ can be linearized in the vicinity of the nominal system (where $p = 0$) and presented as

$$R(p) \approx R_0 + \frac{\partial R}{\partial p} p, \tag{11}$$

where $R_0 = R(0)$ is the model prediction residual obtained assuming the faults are absent, $p = 0$. The matrix $\frac{\partial R}{\partial p}$ in (11) can be conveniently evaluated using a finite increment method, by repeatedly computing residual (11) for different values of p.

By using (11), the least-square optimal estimate of the fault parameter vector p can be obtained in the form

$$\hat{p} \quad = \quad - \left(\frac{\partial R}{\partial p}^T \frac{\partial R}{\partial p} \right)^{-1} \frac{\partial R}{\partial p}^T R_0 \tag{12}$$

One of the main issues that have to be resolved to achieve practical applicability of the described algorithm is computational performance. Each computation of the residual (9) requires to run a comprehensive simulation model for the duration of simulation time equal or exceeding the data batch time. Another important practical issue is robustness of the estimate to the noise in the data. Yet another issue is related to individual variations of the performance between different engine units. This makes necessary to augment the estimation of the fault parameters with a trending of these estimates from one flight cycle to another. These issues are discussed in the paper sections to follow.

3 FIR differentiator design

The problem of estimating a derivative of a measured signal has to be approached carefully because in practice such measurements are always corrupted by noise - random or quantization. Since all data in consideration are sampled at the controller rate, a sampled- time formulation of the problem will be considered. The problem of designing a statistically optimal differentiator can be posed as a *deconvolution* problem. The deconvolution problem is to estimate an unknown input u of a linear system from a measurement x corrupted by a noise. The statistical properties of the noise and the signal x are assumed to be known.

The numerical differentiation can be stated as the following deconvolution problem

$$x = \frac{1}{1 - z^{-1}} v + E(z)e, \quad v = U(z)\eta, \tag{13}$$

where z^{-1} is a unit delay operator, $e = e(t)$ and $\eta = \eta(t)$ are uncorrelated Gaussian white noise sequences. The transfer function $E(z)$ defines the noise model and $U(z)$, the signal model.

The noise model $E(z)$ is can be directly identified from the spectral estimate of the steady-state data, where the useful signal $v(t)$ is known to be zero. The signal model $U(z)$ essentially defines differentiator bandwidth specs and is selected by the designer. It is assumed that $U(z)$ has a zero at $z = 1$. This assumption is needed to guarantee that the signal $x(t)$ has a finite covariance.

A solution of the deconvolution problem is sought in the form

$$\hat{v} \equiv \hat{\dot{x}} = D(z)x, \tag{14}$$

where $D(z)$ is a linear operator.

Deconvolution problems are discussed in [1, 6, 12]. We use an optimal non-causal Wiener filter solution of the deconvolution problem that can be obtained in the form (see [5] for more detail)

$$D(z) = \frac{UU^{\tilde{}}}{1 - z^{-1}} \left[EE^{\tilde{}} + \frac{UU^{\tilde{}}}{(1 - z^{-1})(1 - z)} \right] \tag{15}$$

where $U^{\tilde{}} = U(z^{-1})$. Note, that in the literature Wiener filter often means a *causal* Wiener filter. Such a causal filter obtained is a result of factorization (projection) of a noncausal least-square optimization equations. The Wiener filter in this paper is a *noncausal* least-square optimal filter.

The transfer function $D(z)$ in (15) describes a noncausal operator and has poles both inside and outside of the unit circle. This transfer function $D(z)$ is a 2-sided z-transform of a corresponding non-causal pulse response kernel $d(t)$ (see [9] for a background on 2-sided z-transform). The operator D can be implemented as a convolution with the kernel $d(t)$.

$$(Dx)(t) = \sum_{\tau=-\infty}^{\infty} d(\tau)x(t - \tau) \tag{16}$$

Since $D(z)$ in (15) has no poles on the unit circle, the kernel $d(t)$ in (16) decays exponentially for $t \rightarrow \pm\infty$. For a practical purpose the kernel can be implemented as a delayed noncausal FIR operator.

The convolution kernel $d(t)$ can be computed in straightforward way as an inverse of the Fourier transform $D(e^{i\omega})$, where $\omega \in [0,\ 2\pi]$ is the frequency. This requires substituting $z = e^{i\omega}$ into (15) and noticing that $EE^{\tilde{}}$ computed on the unit circle has a meaning of the spectral power for the noise corrupting the differentiated signal x. As mentioned above, the spectral power of the signal $UU^{\sim}(z = e^{i\omega}) = |U(z = e^{i\omega})|^2$ is used as a bandwidth tuning knob in the design.

Figure 3 (left) illustrates the truncated FIR convolution kernel of the designed differentiator. This kernel $d(t)$ can be presented as a composition of a pure differentiator and FIR kernel $l(t)$ of a smoothing operator $L(z)$. The smoothing operator is defined as

$$L(z) = \frac{1}{1 - z^{-1}} D(z), \tag{17}$$

The FIR kernel $l(t)$ is illustrated in Figure 3 (right). In the algorithms of the next subsection, the numerically differentiated signal is compared to a smoothed prediction sequence. This is necessary to ensure consistency and the zero error in the ideal case.

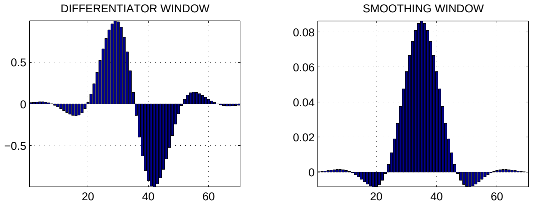

Figure 3: The designed optimal differentiator window (left) and the corresponding smoothing window (right)

Figure 4 illustrates the performance of the designed differentiator. The upper plot shows a sequence of steady-state engine speed data with added first order step response signal. The spectral power of the steady state noise has been used in the differentiator design. The lower plot shows the derivative estimate obtained from the noisy data. As one can see, it matches the smoothed version of the applied first order step response (the second curve) quite well.

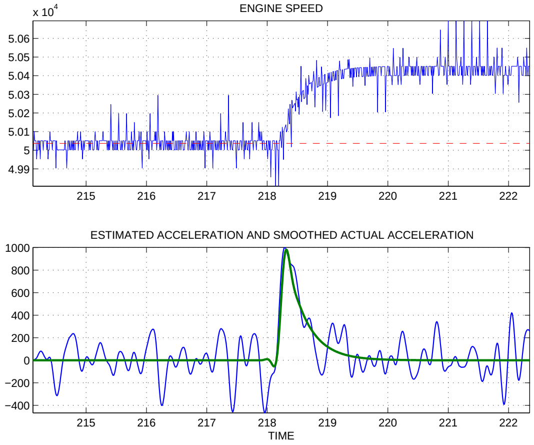

Figure 4: Estimation of engine acceleration from noisy engine speed data

4 Results for 331-400 APU

The evaluation of the described fault estimation approach was performed using a high-fidelity detailed simulation model of the Honeywell 331-400 APU starting. The closed-loop controller was simulated in detail including the digital control logic and a hydromechanical fuel control unit. The engine simulation included pneumatic and electrical loads, oil-lube system, starter, accessory gearbox, and inlet model simulations. A detailed simulation model of the sensors included realistic effects of noise, analog pre-filtering, and sampling quantization. The detailed models of the considered faults were included in the simulation.

The following data were available for the diagnostics and logged during the simulation at 0.2 sec rate.

> **Input Data:** Engine shaft speed N, inlet pressure P2, inlet temperature T2, mass fuel flow W$_f$, generator load, oil temperature, mass air flow through load compressor W$_c$, and IGV (inlet guide vane) position.
>
> **Output Data:** Load compressor temperature increase LCDT, and exhaust gas temperature EGT

This data was logged in the simulation and stored in a file. The diagnostics algorithms used this data only and did not have information about the faults seeded. This closely resembled data availability for 331-400 APU operation.

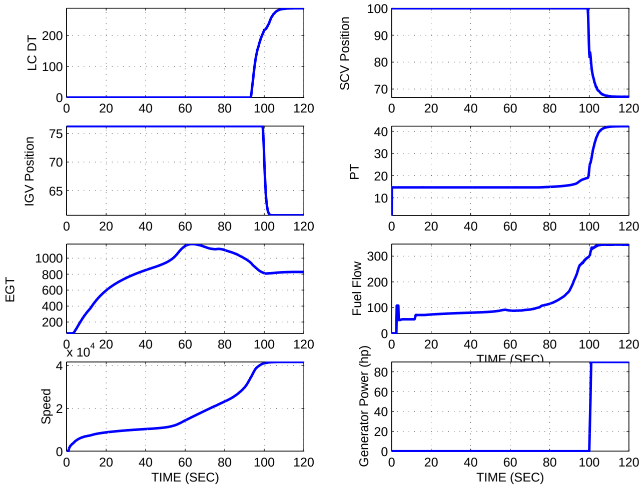

Figure 5: Input data from a data file used for the diagnostics.

The diagnostics algorithms described in this paper were implemented in Matlab. The prediction model of the 331-400 APU used by the diagnostics algorithms was simplified compared to the detailed simulation model. It included main gas-path dynamics but ignored some of the detail in the simulation model, such as sensor data pre-fitering. Thus, there was a mismatch between the prediction model and the high-fidelity simulation, which contributed to the realism of the diagnostics algorithm use. The diagnostics algorithms use the detailed simulation results from a file. The data used for the diagnostics is illustrated in Figure 5. The above mentioned input data was used to compute the predictions for the outputs, given the faults. In addition to that, a prediction of the acceleration was computed as illustrated in Figure 2 and explained in Section 3.

To obtain the data in Figure 5 three faults were simultaneously seeded as follows:

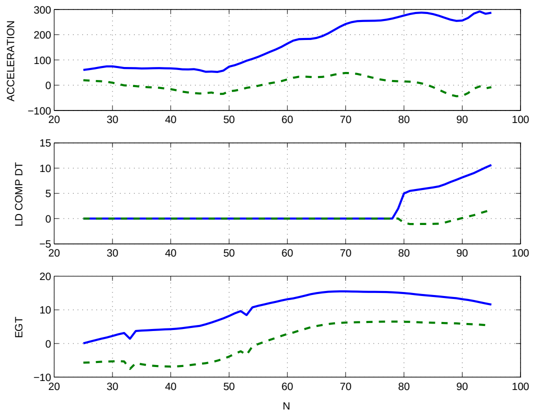

Figure 6: Prediction residuals in the vector $R_0 = R(0)$ used for the diagnostics (solid lines) and prediction residuals for the obtained fault estimate $R(\hat{p})$.

No performance loss for the power section.

A 1% performance loss for the load compressor (LC).

A parasitic torque drag on the main shaft of 0.4 (about 2% of the maximal torque spec).

A 2% fuel flow reduction caused by the nozzle degradation

When used on the field data, the fault estimated will be trended in time to compensate for engine-to-engine variation. To emulate this, the estimate (9) of the seeded fault was corrected by the estimate obtained for zero fault seeded. The estimates of the fault parameters obtained using the described diagnostics algorithms from the data in Figure 5 were as follows:

A -0.49% performance loss for the power section (performance gain)

A 0.99% performance loss for the load compressor

A parasitic torque drag on the main shaft of 0.42

A 2% fuel flow reduction

These estimates have errors of about 20% which allows for adequate discrimination of various engine problems. The estimation error sources include: sensor noise and offsets that were simulated, modeling errors (the simplified prediction model was used), and using a linear approximation (11) of the nonlinear map (10) in the estimation.

The model prediction errors are illustrated in Figure 6. The solid lines in Figure 6 show the prediction residuals corresponding to Figure 5. The dashed lines are the prediction residuals that remain unexplained by the estimated faults. These dashed lines correspond to $R(\hat{p})$ in (9), (10), where $\hat{p}$ is the fault estimate. For an ideal prediction model the dashed lines should be identical zero, $R(\hat{p}) = 0$. Their deviation from zero illustrates the modeling errors. Despite these errors, reasonable fault estimates have been obtained using the described methodology.

5 Conclusions

This paper presented a case study for diagnostics of a small turbine engine – Honeywell 331-400 APU. The incipient fault estimation is performed from batch data by identifying deficiencies of the performance parameters in a detailed model of the engine. The algorithms include numerical estimation of the engine acceleration from the noisy engine speed data , which is performed with a rigorously designed finite impulse response differentiator. The estimation approach is based on assuming the performance parameters change is small and respective system output change can be obtained by linearizing the model. The paper demonstrates that location and magnitude of the performance faults can be estimated accurately despite the noise and nonlinearity, by using only exhaust gas temperature, compressor exit temperature, and engine speed data as prediction variables.

References

[1] Ahlen, A. and M. Sternrad “Wiener filter design using polynomial equations,” *IEEE Trans. on Signal Processing*, Vol 39, No 11, 1991, pp. 2387–2399.

[2] Craig, R., A.M. Davison and A. Birk “Steady state performance simulation of auxillary power unit with faults for component diagnosis,” *ASME Int. Gas Turb. Inst.* , Munich, Germany, May 2000

[3] Frank, P.M. “Fault diagnosis in dynamic systems using analytical knowledge-based redundancy—a survey and some new results,” *Automatica*, Vol. 26, No 3,1990, pp. 459–474.

[4] Gertler, J. “Survey of model-based fault isolation and detection in complex plants,” *IEEE Control Systems Magazine*, Vol. 8, No. 6, 1988, pp. 3-11.

[5] Gorinevsky, D. and D. Mylaraswamy, “Designing model-based fault estimator for a separation column,” *ACC*, 2001, Arlington, VA, pp. 1765–1770.

[6] Grimble, M.J. “H_2 inferential filtering, prediction, and smoothing with application to rolling mill gauge estimation,” *IEEE Tr. on Signal Processing*, Vol. 42, No. 8, 1994, pp. 2078–2093.

[7] Isermann, R. and P. Balle “Trends in the application of model based fault detection and diagnosis of technical processes,” *13th IFAC World Congress*, San Francisco, CA,1996, Vol. N, pp. 1-2

[8] Merrington, G.L. “Fault diagnosis in gas turbines using a model-based technique,” *Trans. ASME, J. Eng. Gas Turb. and Power*, Vol. 116, 1994, pp. 374–380.

[9] Oppenheim, A.V., R.W. Schafer, and J.R. Buck *Discrete-Time Signal Processing*. Prentice Hall, 1999

[10] Patton, R.J., P.M. Frank and R.N. Clark *Fault diagnosis in dynamic systems theory and applications*. Prentice-Hall. Englewood Cliffs, NJ,1989

[11] Roos, H., M. Stynes and L. Tobiska *Numerical methods for singuraly perturbed differential equations: Convection-diffusion and flow problems*, Springer Series in Computational Mathematics,1996

[12] Zhang, H.-S., X.-J. Liu and T.-Y. Chai “A new method for optimal deconvolution,” *IEEE Trans. on Signal Processing*, Vol 45, No. 10, 1997, pp. 2596–2599.