Model Predictive Estimation of Evolving Faults

Sikandar Samar
Information Systems Laboratory
Stanford University
sikandar@stanford.edu

Dimitry Gorinevsky
Information Systems Laboratory
Stanford University
gorin@stanford.edu

***Abstract*— In this work we present an optimization based statistical estimation approach for diagnostics in large scale systems. The fault estimation scheme relies on prediction residuals generated by detailed prediction models of the system under consideration. The system dynamics are generally nonlinear. We linearize the system around its nominal operation and estimate deviations (faults) from the nominal behavior. The statistical estimation approach is based on numerical optimization of a log-likelihood function. It allows us to estimate time varying fault parameters in an online setting, and can accommodate the loss of some sensor measurements during system operation. The proposed estimation approach is explained through examples from aerospace applications.**

I. Introduction

This work focuses on systems for which we have detailed prediction models. These models may contain a nonlinear dependence of the system dynamics on the unknown fault parameters. We assume that the magnitude of the evolving faults is small and they don't significantly change the dynamics of the system. This allows us to linearize the system around its nominal (no fault) trajectory. The system will ideally follow the nominal path during its operation. Any deviations from this nominal behavior may be indicative of a faulty operation. Such faults increase the probability of system failure, and have serious economical and safety implications. It is therefore critical to have a reliable method for an accurate online estimate of the developing faults. The proposed fault estimation scheme is applicable to a wide class of problems where detailed system models are available. Application areas include several safety critical systems such as spacecraft and aircraft,process plants, startup/shutdown sequences in nuclear reactors and engines etc.

There has been a lot of emphasis in system monitoring, fault detection and diagnosis. Different data-driven and model-based schemes have been proposed for fault identification [1], [2], [3]. In a wide variety of applications, fault detection is carried out by implementing a simple threshold logic. In the case where only output signals are available, signal-based methods for fault detection are used. These include spectral analysis, maximum entropy based estimation, and bandpass filters etc. More recently, parameter estimation and observer-based methods have been used for system diagnostics [4], [5].

Most model-based fault detection methods deal with the *residuals*, also referred to as the *parity variables*. The residuals reflect the deviation between the actual plant measurement and the model-based prediction. A non-zero residual serves as a fault indicator. There are several techniques for generating residuals depending upon the type of models under consideration. Output observer-based approach is used for fixed parametric models, parity equations are used for fixed parametric or nonparametric models, identification and parameter estimation is used for adaptive nonparametric or parametric models. Once the residuals are obtained, the next step is to detect the faults by residual analysis. Different residual evaluation methods, like neural networks, fuzzy logic and Bayes classification, are used in diagnostics. We use convex optimization based methods to find statistically optimal estimates of the incipient faults from the given residuals. The optimization based approach is easily scalable to diagnostics of large scale systems. It provides statistically optimal estimates in an efficient manner to enable real time online detection.

In this paper, we present two main implementations of the optimization based approach for determining the unknown fault parameters. The *batch estimation* approach uses the entire data available at any instant of system operation. As a result, the estimates tend to improve as more measurements become available with time. A major drawback of using a batch estimation approach relates to memory limitations and computational time. The *moving horizon estimation* approach has a Kalman filter type recursive formulation and solves a problem of fixed dimension at every step. For more details of the moving horizon estimation scheme, see [6], [7], [8].

As a base-line case, consider the fault parameters entering the system dynamics linearly. If we allow only gaussian noise, the resulting optimal solution is given by batch least-squares. We consider a problem that is more general than the unconstrained least-squares formulation. We deal with linear time varying models and account for any a priori information about the unknown faults in the form of constraints. More general noise distributions, like uniform, exponential and laplacian noise can be handled in the developed approach. Estimates of the time varying fault parameters are obtained by solving a constrained convex programming problem for each batch/moving horizon. This approach is robust to loss of some sensors during system operation.

This paper is composed as follows. First, a brief description of the type of problems being considered is presented in Section II. The models of different types of faults that can be estimated by the proposed approach are given in

Section III. A convex formulation of the statistical estimation scheme is presented in Section IV. Section V describes the computation of the matrix of fault sensitivities that is critical to the residual-based estimation approach. Batch optimization algorithms are described and validated by an example of rocket diagnostics in VI. The proposed approach is then used to solve a moving horizon estimation problem in Section VII. Some concluding remarks are presented in Section VIII.

II. Technical Problem Statement

Model-based approach is frequently used for system analysis in many application areas. In this work, we deal with detailed predictive models of the system under consideration. The prediction model may depend nonlinearly on some faults of interest. The zero fault case determines the nominal trajectory of the system. We begin by linearizing the system around its nominal trajectory. Using this linear relationship, we estimate any deviations (faults) from the nominal behavior.

The *prediction model* divides the system operation in an input/output form. The model predicts the outputs given the inputs. The predicted output is determined by the underlying system dynamics. The model describes the dynamics of the system through a set of ordinary differential equations (ODE) or static maps. This ODE model is either derived from the basic principles of physics and/or obtained empirically. Once developed, the model may be numerically integrated to simulate the system operation. Fig. 1 shows schematics of a prediction model: $x(t) \in \mathbf{R}^n$ is the state vector (measured variables), $f(t) \in \mathbf{R}^p$ is the vector of seeded faults and $t \in \mathbf{Z}^+$ denotes the sampling time. Note that we allow fault inputs to the prediction model. The output of the prediction

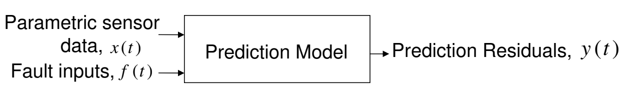

Fig. 1. Schematic diagram of a prediction model

model is the vector of prediction residuals, $y(|f) \in \mathbf{R}^m$. These model-based prediction residuals are the difference between the predicted output and the actually observed output, assuming nominal (no fault) system behavior. If the prediction model accurately describes the system operation, the residuals are zero in the absence of faults, i.e., $y = 0$ when $f = 0$. A non-zero residual value on the other hand results from off-nominal behavior and indicates some deviation of the parameters from their nominal values. This may correspond to performance deterioration in the system.

The parameters of interest may enter the system dynamics in a non-linear manner. As a consequence, the prediction model is in general non-linear. In many applications, the magnitude of the underlying faults is usually small, i.e., the perturbations from the nominal parameter values are not too large. We can thus make the fundamental assumption that the change in underlying system dynamics caused by the (small) change in nominal parameters values is relatively small under most operating conditions. This allows us to linearize the model around the nominal where $f = 0$. Using this linear approximation for the prediction model, we get a linear fault residual relationship that can be conveniently expressed as

$$y(|f) = Sf + e, \tag{1}$$

where $S \in \mathbf{R}^{m \times p}$ is called the fault sensitivity matrix or the matrix of fault signatures. It can be obtained by Jacobian linearization of the prediction model. If the model is not available in an analytical form, the sensitivity matrix is obtained by a secant method. The computations may be performed on-line or off-line depending upon the application under consideration. Details about the calculation of the fault sensitivity matrix are discussed in Section V. The noise term e accounts for modeling and sensor measurement errors. Note that if $e = 0$ in (1) (no noise, no modeling errors), then the residual in the absence of fault $y(|0) = 0$. This will indicate nominal system operation. We assume e to be uncorrelated normally distributed noise sequence with zero mean and covariance Q

$$e_\tau \backsim N(0, Q). \tag{2}$$

The fault estimation problem is to find the unknown fault parameters f, given the prediction residuals y and the matrix of fault signatures S.

III. Fault Evolution Models

To obtain a statistically optimal estimate of the unknown faults, we complement the linear residual model (1) with the statistical model of the unknown fault sequence. The proposed fault estimation scheme is applicable to a variety of fault models that arise in different application areas. In our framework, the most general model for the evolution of the unknown faults has the form

$$f(t+1) = \Phi(f(t)) + \gamma(t), \tag{3}$$

where $\Phi(\cdot)$ is some linear or nonlinear function of the fault at the previous step. $\gamma(\tau)$ is the process noise driving the evolution of the fault vector at the update cycle τ. The noise is assumed to be independent, identically distributed (IID) with probability density function $p(\cdot)$.

A. Linear Model with Gaussian Noise

As a baseline case, we consider $\Phi(\cdot)$ to be linear and the process noise γ to be gaussian. Then the fault evolution model (3) reduces to

$$f(t+1) = af(t) + \gamma(t), \tag{4}$$

The noise γ is gaussian with zero mean and variance σ^2. Its probability density is given as

$$p(z) = (2\pi\sigma^2)^{-1/2} e^{-z^2/2\sigma^2}. \tag{5}$$

This linear gaussian model covers the familiar random walk approach of probabilistic modeling of an unknown data sequence.

B. Linear Model with Uniform Noise

In some applications, a linear model with a uniform noise distribution instead of the gaussian noise better describes the evolution of faults. If we can consider $\gamma(t)$ to be uniformly distributed on $[-a, a]$, then the probability density on this interval is given as

$$p(z) = \frac{1}{2a}, \qquad z \in [-a, a]. \tag{6}$$

C. Linear Model with Laplacian Noise

We may also extend our model to fit the situations where fault evolution is better described by a Laplacian noise as opposed to other symmetric distributions. If the let $\gamma(t)$ to be Laplacian, then the probability density is given as

$$p(z) = \frac{1}{2a} e^{-|z|/a}, \qquad a > 0. \tag{7}$$

D. Linear Model with Exponential Noise

We introduce another fault evolution model in which $\gamma(t)$ is an uncorrelated exponentially distributed noise sequence. The exponential distribution is used to model fault trends that accumulate with time, e.g., mechanical damage. It is used to model faults such as fatigue damage which are explained by the well known Palmgren-Miner cumulative damage theory. A symmetric distribution implies that the probability of fault increasing or decreasing with time is the same. A one sided exponential distribution is therefore used to model monotonic fault trends. The probability density of the exponentially distributed noise is given as

$$p(z) = \frac{1}{\lambda} e^{-z/\lambda}, \qquad z \geq 0. \tag{8}$$

We now mathematically formulate the fault estimation problem as the solution of a convex log-likelihood function.

IV. Solution Approach

To obtain a statistically optimal estimate of the unknown fault parameters, such as the *maximum likelihood* (ML) or the *maximum a posteriori probability* (MAP) estimate, we make use of the concept of conditional probability. For any two random variables f and y, the conditional probability is denoted $P(f|y)$ with the corresponding conditional probability density represented by $p_{f|y}$. It is natural to think in terms of conditional probabilities when we have dynamical models of the form (1). For such discrete time Markov processes, the known prediction residual completely determines the unknown future fault evolution up to the random disturbances given by e.

The ML estimate of the random variable f given y is

$$\text{ML estimate} := \arg\max_x p_{y|f}. \tag{9}$$

The MAP estimate of the random variable f given y is

$$\text{MAP estimate} := \arg\max_x p_{f|y}. \tag{10}$$

The ML estimate considers f as a parameters whereas the MAP estimate takes f to be a random variable with some prior density. This density represents our prior information about what the values of the unknown fault vector might be, before we observe the prediction residuals. Using the Bayes' rule, we define the performance index J as

$$J := -\log p_{f|y} = -\log p_{y|f} - \log p_{f_k}(.) + c, \tag{11}$$

where subscript k is used to index different types of faults. $f_k(.)$ denotes the time series $f^k(1) \ldots f^k(t)$ for each fault as given by the appropriate fault evolution model of Section III, and $c = \log p_y$ is a constant that plays no role in determining the estimate . We obtain the MAP estimate of the unknown fault sequence by minimizing the loss index (11)

$$\hat{f}_{MAP} := \arg\min\left[-\log p_{f|y}\right], \qquad p_{f|y} \neq 0. \tag{12}$$

The first term in the performance index, $-\log p_{y|f}$ is determined by the linear model (1). From our earlier assumption of a gaussian distribution for the noise term e in (2), we can rewrite the performance index (11) as

$$J = \frac{1}{2}(y - Sf)^T Q^{-1}(y - Sf) \; + \; \sum_k J_k, \tag{13}$$

where $J_k = -\log p_{f_k(.)}$ depends on the nature of each fault in the fault vector f. The type of fault is determined by the choice of the corresponding fault model. This term can be interpreted as taking our prior knowledge of the faults into account. It essentially penalizes estimates of faults that are unlikely according to the prior density. If we assume that $\gamma(t)$ in the fault evolution model (3) are mutually independent for different k, and have independent probability distributions with the density functions $p_k(x)$, then

$$J_k = \sum_{t=2}^{N} -\log p_k\left(f_k(t) - \Phi(f_k(t-1))\right), \tag{14}$$

where N denotes the length of the estimation horizon.

V. Fault sensitivity computations

As described in Section II, a non-zero residual vector in indicates an off-nominal behavior and implies presence of faults. The diagnostics algorithms use the prediction residuals along with the fault signatures as inputs to determine the fault estimates. The fault sensitivity matrix in (1) provides a linear mapping between the unknown faults and the residuals. This linearized dependence of the residuals on the faults allows us to efficiently solve the estimation problem by reducing it to a quadratic programming problem. The idea of linearization of the non linear map about the nominal is motivated by the analogous approach in the control problems. In a control problem formulation, the use of linear time varying (LTV) and linear time invariant (LTI) maps is widely accepted. Estimation is the dual of control, and it is natural to follow a similar approach in this setup.

The fundamental assumption in our discussion up to this point is that the fault sensitivity matrix in (1) is a known quantity. We now give an explanation of its computation. The dependence of the residual data vector on the faults given by the prediction model in Fig. 1, is in general not available in an analytical form for computing the sensitivity

(Jacobian) matrix S. It can however be computable point-wise by running a simulation of the prediction model. The sensitivity matrix can then be numerically estimated by a finite difference method. This is done by simply incrementing each component of the fault vector and then running the prediction model with the corresponding fault inputs to compute point-wise values of the residual data vector $y(|f)$. The finite difference estimate of the columns of S is obtained by normalizing increments of the observed residual vector change. Mathematically, the secant estimate for a column $S(t)^{(j)}$ of the sensitivity matrix S at time instant t is given by

$$S(t)^{(j)} = \left[\frac{y(t|se^{(j)}) - y(t|0)}{s} \right], \tag{15}$$

where $e^{(j)}$ is the unit fault vector with all entries zero except the unit component j, and s is the secant step size. The choice of s determines the tradeoff between the nonlinearity error and the numerical accuracy. A small s might limit the numerical accuracy while reducing the nonlinearity error. The sensitivity matrix computations may be performed on-line, or off-line prior to system operation, depending upon the type of application.

A. On-line and Off-line Sensitivity Computations

In some cases, the sensitivity matrix computation may be performed off-line prior to the system operation. The pre-computed sensitivity matrix may then be stored and later used by the estimation algorithms during on-line computations. Such a computation of sensitivity matrix assumes that the system will closely follow its nominal trajectory during its operation. If the actual state of the system doesn't match the desired trajectory, then there will be an inaccuracy in the computed sensitivity matrix. In many space applications, where the pre-planned trajectory of the vehicle is available, off-line computations may be more suitable. The results for the satellite launch vehicle example of the next section are obtained using off-line computations.

An alternate method is to compute the sensitivity matrix on-line using the actual system state obtained from the sensors. In this approach the nominal prediction model is run alongside the prediction models with each of the fault inputs in an online setting using the actual system state at that time of the flight. The prediction model is a static input output map in this case and requires the measured states in real time to compute the sensitivities. In aircraft applications, where there are disturbances in the form of wind gusts, on-line processing of the fault signatures greatly increases the accuracy of the estimates. The downside is the increases computational burden on the on-board processors.

VI. Batch estimation approach

The batch estimation approach makes use of all the available data at any instant in time. The estimates improve as more data becomes available due to better statistical averaging. We denote the residual data vector accumulated from start to the current estimation update cycle number τ as

$$Y_\tau = \left[\begin{array}{ccc} y(1) & \ldots & y(\tau \cdot M) \end{array} \right]^T, \tag{16}$$

where M represents the length of each update cycle. The sampled-data estimation logic assumes that fault parameters $f(t)$ are constant through each estimate cycle, i.e., $f(t) = F(\tau) \in \mathbf{R}^p$, for $M(\tau - 1) < t \leq M\tau$. This assumption reduces the number of the fault values that are estimated and improves statistical averaging properties of the estimation scheme. At the same time, there is little loss of estimation performance in addition to the already accepted sampling time delay of the estimation update. We let the fault parameter vector accumulated from the start to the estimation update cycle number τ be denoted as

$$F_\tau = \left[\begin{array}{ccc} f(1) & \ldots & f(\tau) \end{array} \right]^T. \tag{17}$$

The algorithm objective is finding the unknown fault parameter vector F_τ. This requires relating F_τ to the available residual data vector Y_τ. As mentioned earlier, if the faults don't change the underlying system dynamics substantially, we can linearize the dependence between the residuals and the unknown faults. This linear relationship given in (1) is expressed in the batch mode up to the current update cycle as

$$Y_\tau(|F_\tau) = S_\tau \cdot F_\tau + e_\tau, \tag{18}$$

where e_τ is the noise vector, and S_τ is the fault sensitivity matrix for data up to the current estimation cycle τ. The sensitivity matrix S_τ corresponding to the estimation update cycle τ does not need to be computed from scratch at each update cycle. Instead it can be computed once for the terminal update cycle, say $\tau = T_f$. For any $\tau < T_f$, the matrix S_τ is a truncation of the matrix S_T (a $M\tau \times \tau$ submatrix of the $MT \times T$ matrix S_T).

A. Rocket Ascent Example

In this section we briefly discuss the application of the batch algorithms to diagnostics of rocket flight control. The example is discussed in [9], where details of the vehicle telemetry data $x(t)$, prediction residuals $y(t)$, unknown faults $f(t)$, and the relevant subsystem prediction models are given. The developed algorithms are applied to the full nonlinear models in [9] and are shown to give good estimates even in the presence of nonlinearities.

Even the simplified models of a rocket control system are highly nonlinear and protected by proprietary information. We work with a given linear model of the system in the form (18) to validate the batch estimation approach. The diagnostics problem is to estimate the following four parametric faults during the first phase of rocket ascent

$$f := \left[\begin{array}{c} \text{Thrust Loss, percent} \\ \text{Drag Increase, percent} \\ \text{Gimbal Sluggishness, percent} \\ \text{Pitch Sensor Offset, percent} \end{array} \right], \tag{19}$$

where thrust loss is modeled as a monotonic fault, drag increase is considered a step, gimbal sluggishness is assumed to be constant, and pitch sensor offset is non-monotonic.

The nature of the faults determines the choice of the probability distribution in (14). For the two monotonic faults, thrust loss and drag increase, we use the linear fault evolution model with exponential noise as given in Section III-D. Substituting this exponential distribution for the process noise γ_k in (14) yields

$$J_k = \sum_{\tau=2}^{L} \frac{1}{\lambda} \left[F_k(\tau) - F_k(\tau-1) \right]. \qquad (20)$$

Since $p_k(x) = 0$ for $x < 0$, the condition in (12) implies that to obtain the MAP estimate for the two monotonic faults we need to minimize the loss index (13) subject to the constraints

$$F_k(\tau+1) \geq F_k(\tau). \qquad (21)$$

For the non-monotonic fault of pitch sensor offset, we use the fault evolution model of Section III-A. The symmetric distribution is used as it implies that the probability of the fault increasing or decreasing is the same. Substituting this gaussian distribution for the process noise γ_k in (14) yields

$$J_k = \frac{1}{2\sigma} \sum_{\tau=2}^{L} \left[F_k(\tau) - F_k(\tau-1) \right]^T \left[F_k(\tau) - F_k(\tau-1) \right]. \qquad (22)$$

The assumed gaussian distribution of the noise γ_k for a non-monotonic fault results in a quadratic penalty term in the loss index (13) and the problem becomes an unconstrained generalized least-squares estimation.
Gimbal sluggishness appears as a constant fault. A constant fault describes a fault condition that does not change during the planned system operation. In this case, $J_k = 0$ in (14) and we minimize the loss index (13) subject to an equality constraint

$$F_k(\tau+1) = F_k(\tau). \qquad (23)$$

The prediction residual vector $y(t)$ is a combination of the guidance, navigation, and control subsystem residuals and the gimbal subsystem residual

$$y(t) := \begin{bmatrix} \text{Vertical acceleration residual, ft/s}^2 \\ \text{Flight angle rate residual, rad/s} \\ \text{Pitch acceleration residual, rad/s}^2 \\ \text{Servo rate residual, rad/s} \end{bmatrix}. \qquad (24)$$

A uniformly distributed noise was added to the residual data to make the estimation problem more realistic. The fault sensitivity matrix is provided for the first phase of rocket launch that lasts approximately 150 seconds. Fig. 2 shows the columns of the given linear operator S_τ for the update cycle $\tau = 6$, i.e., $75 \leq t \leq 90$. Here the operator is causal in the sense that $Y_\tau(F_\tau)$ does not depend on F_ρ, $\rho > t/M$. It is also clear from the figure that the map is sparse in time because of the negligible influence of fault after-effect on the residual vector for the next cycle. The sampling is at a high rate of 100 ms and we decide to run the estimation update every 15 seconds.

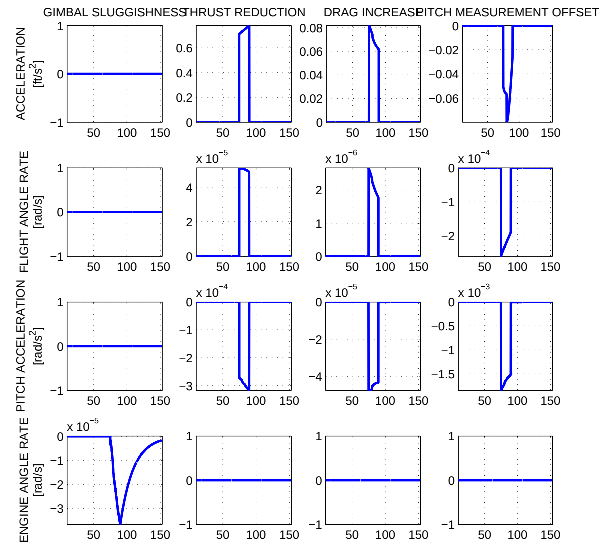

Fig. 2. Fault signatures for an interval during first stage ascent

Fault estimates are obtained using the batch estimation scheme by minimizing the loss index in (13), with the appropriate J_k term chosen from (20) and (22) and the constraints (21) and (23). For validation, the obtained estimates are compared against the faults that were actually seeded in the given linear model. Fig. 3 shows the estimates computed at $t = 70$, 115, and 150. As seen in the plots the estimates improve with time as more data is accumulated, and match the unknown (seeded) faults reasonably well despite the added noise.

VII. Moving Horizon Estimation Approach

As more data becomes available for estimation, the size of the batch optimization problem increases, and the problem may become computationally intractable beyond a certain dimension. To overcome this problem, a moving horizon estimation approach is used. It has a Kalman filter type recursive formulation and enables embedded implementation for real time online diagnostics. Moving horizon formulation for constrained estimation is presented in detail in [7].

In a moving horizon formulation, we always solve a problem of fixed size (horizon) at each instant. Let N denote the length of the moving horizon. We can then rewrite the loss index (13) for the moving horizon estimation (MHE) problem as

$$J = \frac{1}{2}(\bar{y} - \bar{S}\bar{f})^T Q^{-1}(\bar{y} - \bar{S}\bar{f}) \;\; - \log p_{f_k}, \qquad (25)$$

where

$$\begin{aligned} \bar{y} &= \left[\; y(t-N) \quad \ldots \quad y(t) \; \right]^T, \\ \bar{f} &= \left[\; f(t-N) \quad \ldots \quad f(t) \; \right]^T, \\ \bar{S} &= \mathbf{diag}\left(\; S(t-N) \quad \ldots \quad S(t) \; \right). \end{aligned}$$

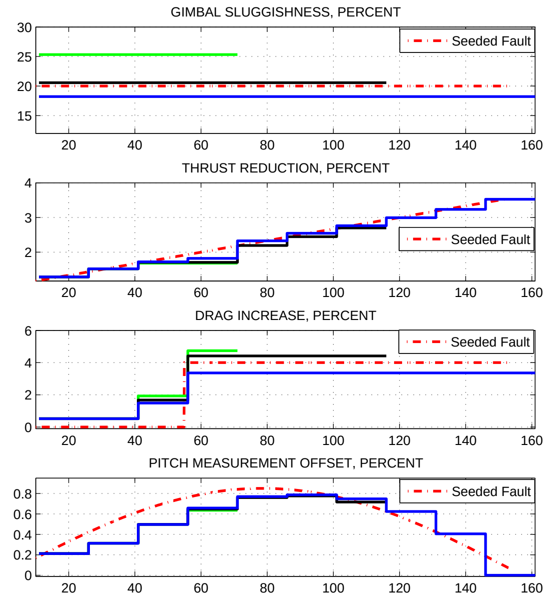

Fig. 3. Seeded faults and fault estimates for Rocket ascent

There should be an additional term in the MHE problem that defines the hand over of initial condition for each moving window. In general, a quadratic penalty term to penalize deviations from the initial estimate is used. This additional penalty can be expressed as

$$\frac{1}{2}[f(t-N)-f_0]^T Q_0^{-1}[f(t-N)-f_0], \tag{26}$$

where Q_0 is the initial condition covariance and $f_0 = f(t-N|t-1)$, i.e., the estimate of $f(t-N)$ at the instant $(t-1)$. If the initial condition covariance is very large then Q_0^{-1} is very small and we can ignore the quadratic penalty term during the MHE solution. On the other hand, if there is reasonable confidence about the initial point estimate at the previous step, then Q_0^{-1} is large and the initial penalty term has a significant contribution to the loss index (25). For very small covariance of the initial condition, we can also specify an explicit equality constraint of the form $f(t-N|t) = f(t-N|t-1)$ during the MHE solution.

A. Unmanned aerial vehicle example

The moving horizon estimation scheme is applied to a linearized model of longitudinal dynamics of the Aerosonde UAV. The Aerosonde was the first UAV to fly across the Atlantic. We present a univariate example to explain the concept.

A single fault, step loss in lift coefficient C_L, was seeded in the model after 50 seconds. The vertical speed residual after the addition of uniform noise was used as an input to the estimation scheme. The residual plot is shown in Fig. 4. The fault signature of the lift loss was computed by linearizing the model about a specified trim state with and without the seeded fault. The results of the lift loss estimate are shown in Fig. 4, where a moving horizon window of $N = 30$ was used in the computations.

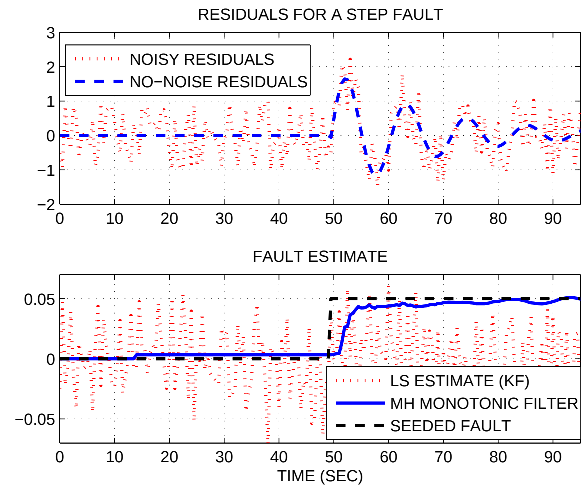

Fig. 4. Residuals and fault estimate for Aerosonde UAV

VIII. Conclusion

In this paper we have developed algorithms for estimating time varying fault parameters for systems with detailed prediction models. The algorithms are based on numerical optimization techniques and perform efficient online computations for incipient fault detection. The proposed approach is validated by an application to rocket flight control and UAV diagnostics. This approach of estimating parametric faults using convex optimization techniques provides a unique diagnostics ability for online fault detection while accommodating a priori fault constraints, multi-rate system operation, and possible sensor loss.

References

[1] E. Russell, L. H. Chiang, and R. D. Braatz. *Data-Driven Methods for Fault Detection and Diagnosis in Chemical Processes*. Springer-Verlag, 2000.

[2] L. H. Chiang, E. Russell, and R. D. Braatz. *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag, 2001.

[3] J. Gertler. *Fault detection and diagnosis in engineering systems*. Marcel Dekker, New York, 1998.

[4] R. H. Chen and J. L. Speyer. Robust multiple-fault detection filter. *International Journal of Robust and Nonlinear Control*, 12(8), 2002.

[5] R. K. Douglas and J. L. Speyer. H-infinity bounded fault detection filter. *AIAA Journal of Guidance, Control, and Dynamics*, 22(1):129–138, 1999.

[6] C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, February 2003.

[7] S. Samar, D. Gorinevsky, and S. Boyd. Moving horizon filter for monotonic trends. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[8] G. Ferrari-Trecate, D. Mignone, and M. Morari. Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control*, 47(10):1663–1676, 2002.

[9] D. Gorinevsky, S. Samar, J. Bain, and G. Aaseng. Integrated diagnostics of rocket flight control. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2005.