Embedded Estimation of Fault Parameters in an Unmanned Aerial Vehicle

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

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

Stephen P. Boyd
Information Systems Laboratory
Stanford University
boyd@stanford.edu

***Abstract*— In this paper, we present a model-based approach for estimating fault conditions in an aircraft. We formulate fault estimation as a convex optimization problem, where estimates are obtained by solving a constrained quadratic program (QP). A moving horizon framework is used to enable recursive implementation of the constrained QP of fixed size. The estimation scheme takes into account a priori known *m*onotonicity constraints on the faults. Monotonicity implies that the fault conditions can only deteriorate with time. We validate the proposed estimation scheme on a detailed nonlinear simulation model of the Aerosonde unmanned aerial vehicle (UAV) in the presence of winds and turbulence. An excellent performance of the developed approach is demonstrated.**

I. Introduction

Model-based fault estimation algorithms have been studied in the controls community for the past two decades; see survey papers [1], [2], [3]. These diagnostic methods involve analysis of *residuals*, also referred to as the *parity variables*. The residuals reflect a discrepancy between measured variables and their model-based predictions. A non-zero residual serves as a fault indicator. The techniques for generating residuals have been developed for a variety of quantitative models, such as parametric models and state space models, as well as qualitative models like expert systems. Recently, there has also been considerable research on fault diagnosis for nonlinear systems [4], [5], [6].

This work focuses on model-based fault estimation in an aircraft. Reliable methods for accurate online estimation of developing fault conditions are of great significance in aerospace and other safety critical systems. Several papers have looked at aircraft failure detection through residual analysis. A basic method is to check the residuals against a threshold. More sophisticated approaches to aircraft parameter estimation based on recursive least squares and Kalman filtering are described in [7], [8], [9]. This paper extends the earlier results by taking into account a knowledge of constraints on the fault parameters. In particular, we incorporate monotonicity constraints which express the deterioration as irreversible. To find statistically optimal constrained estimates of the incipient faults from the given residuals, we use convex optimization methods. A recursive formulation for the constrained convex problems is implemented by using a *moving horizon estimation* (MHE) approach. For more details of the theory of moving horizon estimation, see [10], [11].

To the best of the authors knowledge, constrained recursive convex programs have not previously been used for fault estimation in an aircraft. One possible reason is that, previously, it was not possible to solve such problems reliably in a fraction of a second on an avionics hardware. The earlier related work on optimization-based MHE was in chemical engineering applications (process plants), where the sample time of many seconds or even minutes is acceptable. In this paper we demonstrate that an off-the-shelf solver allows achieving $200-500$ millisecond update on a PC in a realistic aircraft application. The computation time can be further reduced by $1-2$ orders of magnitude by developing specialized solvers.

The aircraft considered in this paper is the Aerosonde unmanned aerial vehicle (UAV). A detailed nonlinear simulation model used for control design of the UAV can be readily re-used to implement the proposed estimation scheme. Various faults related to structural damage and propulsion system degradation are modelled as changes in the aircraft lift and drag coefficients. These faults are seeded in the flight simulation. The diagnostic algorithms are then verified by comparing the on-line fault estimates to the seeded faults.

II. Technical problem statement

We begin by formulating the fault estimation problem for an aircraft. A nonlinear six-degree-of-freedom (6-DOF) aircraft dynamics model used for design and validation of GN&C (Guidance, Navigation, and Control) system of an aircraft generally has the form

$$\begin{aligned} \dot{p} &= \phi(v), \\ \dot{v} &= \psi(p, v, \zeta, u(t), w(t), f), \end{aligned} \tag{1}$$

where p and v denote the vectors of (linear and angular) coordinates and velocities respectively, *i.e.*, $p =$ (latitude, longitude, altitude, roll angle, pitch angle, yaw angle), and $v =$ (velocity north, velocity east, velocity down, roll rate, pitch rate, yaw rate). The vector $u(t)$ consists of control inputs; ζ is the vector of auxiliary parameters; $w(t)$ denotes the input disturbance, caused usually by wind gusts; f is the fault parameter vector (in a nominal condition $f = 0$). The functions ϕ and ψ are static nonlinear maps. Detailed nonlinear dynamical models of the form (1) are commonly developed for control design and analysis of the aircraft. In the estimation framework we assume that the state variables, control inputs, and the auxiliary parameters are available either directly from sensors on the aircraft, or they can be calculated from such sensor data. Our goal is to estimate the faults from these observations. This is explained in the sequel.

A. Prediction Residuals

Model-based fault estimation schemes involve residual generation followed by some type of residual analysis. The 6-DOF aircraft model allows us to use six (linear and angular) acceleration residuals. The residuals are computed as the difference between the actually observed accelerations of the aircraft and the accelerations predicted by the dynamic model assuming that that there are no faults

$$y(|f = 0) \quad = \quad a - \hat{a}, \tag{2}$$

where $y \in \mathbf{R}^m$ is the vector of acceleration residuals. The notation $y(|f)$, where $f \in \mathbf{R}^p$ is the vector of faults, is used to emphasize the fault dependence of the residuals. The vectors a and $\hat{a}$ denote the observed and predicted accelerations respectively. If the prediction model accurately describes the aircraft operation and there are no faults, the residuals should be zero. A non-zero residual indicates that the aircraft dynamics deviate from the nominal model. This may correspond to performance deterioration in the system.

The residuals can also be caused by disturbances. In particular, turbulence effects caused by wind gusts are commonly encountered in an actual aircraft flight. Full knowledge of the wind speed measurements is not a realistic assumption. In our estimation framework, we assume that a low pass filtered knowledge of the winds $\hat{w}$ is available, where

$$\hat{w} = F(s)w, \tag{3}$$

and

$$F(s) = \frac{1}{\tau s + 1} \tag{4}$$

is a low pass filter. The time constant of the filter is used as a tuning parameter during estimation. The estimation scheme has less demands on wind speed measurement system and as a consequence, is more practical, if it yields reliable estimates for a higher value of the time constant.

In many cases (including the application example in Section IV) there are no accurate accelerometers on board the aircraft. In that case the observed accelerations in (2) can be obtained by differentiating the velocities of the aircraft, *i.e.*,

$$a = F_a(s)sv, \tag{5}$$

where s is the Laplace operator and $F_a(s)$ is an appropriate smoothing (low-pass) filter. The filter $F_a(s)$ might have the same form as the filter (4) but with a different time constant. The filter $F_a(s)$ should prevent amplification of the noise signal caused by differentiation and wind gusts.

The accelerations $\hat{a}$ in (2) are obtained from the prediction model, which uses essentially the same blocks as the aircraft GN&C simulation model (1). The prediction model is however an instantaneous input-output map, which is schematically shown in Figure 1. The model takes parametric sensor data (aircraft states, aircraft controls, auxiliary parameters, low pass filtered wind speed measurements $\hat{w}$), and the fault vector as its inputs. The output of the prediction model is the vector of predicted accelerations of the aircraft. To match the smoothed differentiation of the observed acceleration (5), the output of the prediction model is also low-pass filtered to obtain $\hat{a}$ as

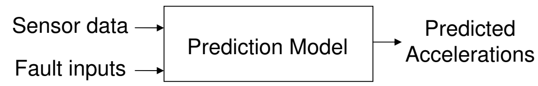

Fig. 1. Schematic diagram of a prediction model

$$\hat{a} = F_a(s)\varphi(p, v, \zeta, u, \hat{w}; f = 0), \tag{6}$$

where φ denotes the aircraft prediction model map.

B. Fault Sensitivity of Prediction Residuals

The sensitivity of prediction residuals to various faults plays a key role in our estimation scheme. The faults usually enter the aircraft dynamics in a non-linear manner. As a consequence, the prediction model map φ is in general non-linear. In most cases however, the magnitude of the underlying faults is small, *i.e.*, the change in underlying aircraft dynamics caused by the faults is relatively small under most operating conditions. We can thus make the fundamental assumption that the prediction model can be linearized around the nominal map φ with $f = 0$. Using this linear approximation, we get a linear relationship between the residual and the fault that can be conveniently expressed as

$$y(|f) = S\,f + e, \tag{7}$$

where $S \in \mathbf{R}^{m \times p}$ is called the fault sensitivity matrix or the matrix of fault signatures. The noise term e in (7) accounts for modeling and sensor measurement errors. Note that if $e = 0$, *i.e.*, no noise and no modeling errors, then the residual in the absence of fault $y(|0) = 0$. This will indicate nominal system operation.

Notice that commonly there is no analytical map available for the full nonlinear aircraft model. The simulation model ψ in (1) and the prediction model φ in (6) are only available as computational blocks. We therefore use a secant estimate of the sensitivity matrix S. The fault sensitivities are computed by first calculating the difference between the predicted accelerations with a given fault input and the predicted accelerations with zero fault input. The difference is then normalized by the input fault magnitude to obtain the fault sensitivity for a given fault, *i.e.*, sensitivity of the i^{th} fault is given as

$$S_{f_i} = \frac{\varphi(\cdot|f_i) - \varphi(\cdot|0)}{|f_i|}, \tag{8}$$

where $\varphi(\cdot|f_i)$ is the predicted acceleration in the presence of fault i and $\varphi(\cdot|0)$ is the predicted acceleration with zero fault input. To compute the fault signatures for p faults, we need to run $(p + 1)$ copies of the prediction model in parallel; one for each of the p faults and an additional to obtain

$\varphi(\cdot|0)$. The computation of the sensitivity matrix S may be performed on-line or off-line depending upon the application in hand. The described approach is related to (but different from) the practice of using a bank of Kalman filters for fault identification [13].

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 estimation via Convex Optimization

We present an optimization-based statistical estimation approach for diagnostics in aircraft. The fault estimation scheme relies on residuals generated by detailed model of the aircraft under consideration, and the fault signatures of the computed residuals. The statistical estimation approach is based on numerical optimization of a log-likelihood function.

To derive the likelihood function, we assume that e in (7) is an uncorrelated normally distributed noise sequence with zero mean and covariance Q, *i.e.*,

$$e \backsim N(0, Q). \tag{9}$$

This noise statistics might not be realistic as e could include the modeling error part. Yet, this is a convenient assumption that is commonly taken in such problems because it leads to a least-squares type estimation. The covariance Q in (9) is used as a tuning parameter for the estimation algorithm.

A. Fault Evolution Model

To obtain a statistically optimal estimate of the unknown faults, we complement the linear residual model (7) with a 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. For details of various fault models that can be accommodated in the proposed scheme, see [14].

In this paper, we are concerned with faults that are *monotonic*, *i.e.*, they only increase (or decrease) with time. Such faults arise in a variety of applications. In the Aerosonde UAV example that is described in this paper, we model parameters related to the structural damage of the aircraft. Monotonicity in this context implies that the damage can only get worse during the course of the flight. We introduce the following fault evolution model in which $\gamma(t)$ is an uncorrelated exponentially distributed noise sequence that models the monotonic fault parameters

$$f(t+1) = f(t) + \gamma(t). \tag{10}$$

The probability density of the exponentially distributed noise is given as

$$p(z) = \begin{cases} \frac{1}{\lambda} e^{-z/\lambda} & z \geq 0 \\ 0 & z < 0, \end{cases} \tag{11}$$

where λ is the parameter of the exponential distribution.

B. Maximum A Posteriori Probability (MAP) Estimation

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 (7). 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. This dependence is accurately captured by the conditional probability density function.

Maximum a posteriori probability (MAP) estimation can be considered as a Bayesian version of maximum likelihood estimation, with a prior probability density on the underlying parameter f. We assume f (the fault vector to be estimated) and y (the observed residuals) to be random variables. In the MAP estimation method, our estimate of f, given the observation y, is given by

$$\hat{f}_{\text{map}} = \text{argmax}_f \, p_{f|y} \tag{12}$$

$$= \text{argmax}_f (p_{y|f} . p_f), \tag{13}$$

where (13) is obtained by direct application of the Bayes rule to (12). The second term in the above equation, p_f, can be interpreted as taking our prior knowledge of the fault parameters into account. Taking negative logarithms, we can express the MAP estimate as

$$\hat{f}_{\text{map}} = \text{argmin}_f (-\log p_{y|f} - \log p_f). \tag{14}$$

The second term in the objective penalizes estimates of f that are unlikely, according to the prior density (*i.e.*, f with p_f small). Substituting the assumed gaussian distribution of the noise e, and the exponential distribution of the noise γ, we obtain the objective function

$$J(t_i, t_f) = \frac{1}{2}(y - Sf)^T Q^{-1}(y - Sf) + \sum_{t=t_i}^{t_f} \frac{1}{\lambda}(f(t+1) - f(t)) \tag{15}$$

which is to be minimized to obtain $\hat{f}_{\text{map}}$. Here t_i and t_f denote the beginning and end of the batch for the minimization of the objective. Since the assumed exponential distribution of the underlying fault vector according to (11) satisfies $p_f(z) = 0$ for $z < 0$, the minimization is subject to the constraints

$$f(t+1) \geq f(t). \tag{16}$$

The fault estimates are, thus, obtained by minimizing the quadratic negative log-likelihood function (15) subject to the linear constraints (16). This is a constrained quadratic programming (QP) problem. Such problems fall in the broader category of convex optimization problems. Efficient algorithms are available to compute reliable solutions of convex problems. We make use of interior point methods to solve the constrained QP problem in an embedded setting, and obtain on-line estimates of the unknown faults.

C. MAP Moving Horizon Estimation

The MAP estimate of the unknown fault parameters can be computed by solving a constrained quadratic programming problem. The size of the QP problem determines the computational efficiency of the estimation scheme. In most aerospace applications, reliable estimates are needed in real time to enable any prognostic measures. The data availability rates in aircrafts might be on an order of a few hundred milliseconds. If the estimation scheme takes into account all the available data at any instant, the size of the QP will grow making the problem computationally intractable within a few minutes. In most instances, we require an embedded filter that can only have limited on-board memory allocation. As a result, storing all the available data for estimation is infeasible.

To overcome memory limitations and ensure efficient estimation, we use a moving horizon (MH) estimation scheme that allows us to solve a fixed size QP at every step by enabling forgetting of the past data. This results in a Kalman filter type recursive formulation that is suitable for embedded implementation. The fixed step size is determined by the length of the moving horizon. The horizon length is chosen large enough to allow sufficient statistical averaging of the noise in the data but small enough to enable fast solution of the QP problem.

In MHE, we need to formalize a way of forgetting the past data. Let N denote the choice of the length of the moving horizon. At each moving step in the embedded filter, we leave out the oldest current measurement while accepting an incoming new measurement. Let $f(k)$ be the estimate of the point k in the current horizon, and $f(k|N^-)$ be the estimate of the same point computed in the previous horizon. We impose a quadratic penalty on the deviation of the initial point estimate of each window from the estimate of that initial point computed in the previous window. Mathematically, we can express the MH MAP estimation problem with initial condition handover as

$$\begin{aligned}\hat{f}_{\mathrm{map}} = \mathrm{argmin}_f\, & \tfrac{1}{2}(f(t_i) - f(t_i|N^-))^T R^{-1}(f(t_i) - f(t_i|N^-)) \\ & + J(t_i, t_f)] \,. \end{aligned} \qquad (17)$$

If t denotes the current sampling time then $t_i = t - N + 1$ and $t_f = t$. The choice of the covariance R depends on our confidence in the past estimate of the initial point of the current window and is used as a tuning parameter in the estimation scheme.

IV. Application to Aerosonde UAV

We demonstrate the proposed MH MAP estimation scheme by detecting some representative faults during the flight of the Aerosonde UAV. This small low cost autonomous plane can be used for a variety of remote sensing applications, particulary weather data acquisition. For details of the Aerosonde development history, design and operational specifications, see [12]. The simulation model used

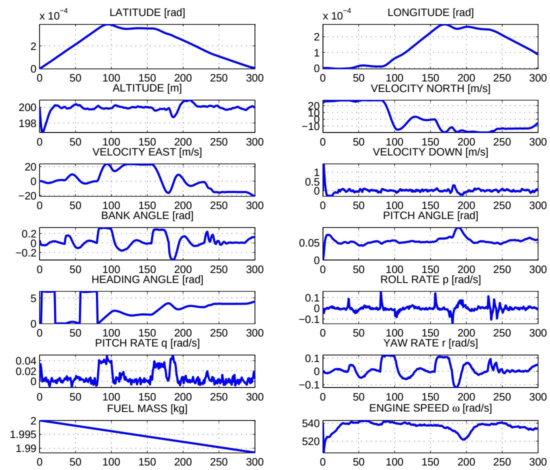

Fig. 2. Aerosonde UAV states

for Aerosonde was developed using the AeroSim Blockset, which is a Matlab/Simulink block library [15]. This nonlinear 6-DOF Aerosonde dynamics model is commercially available at [15]. The detail of the Aerosonde simulation model and a thorough description of the constituent blocks used in its development is available in these references. Here we only give a brief overview for completeness.

The nonlinear 6-DOF Aerosonde model has the form (1). The Aerosonde control vector is $u(t)$ = (flap, elevator, aileron, rudder, throttle, mixture, ignition). The first four terms represent aerodynamic controls and the last three are the propulsion controls. The simulation model allows as input a vector of user specified background wind velocities $w(t)$ in the navigation (North-East-Down) frame. All positions are measured in meters, linear velocities in meters/second, angles in radians, angular rates in radians/second. The overall state vector x for the Aerosonde is

$$x = \begin{bmatrix} p & v & \zeta \end{bmatrix}^T,$$

where $\zeta = (m_f, \omega_e)$ is the vector of auxiliary parameters for the Aerosonde. m_f is the fuel mass (in kilograms) and ω_e is the engine speed. The aircraft states during the 300 second flight are shown in Figure 2.

We simulate the aircraft dynamics, kinematics, guidance and control, and propulsion in the presence of winds and turbulence. The aerodynamic force and moment are computed using a linear combination of the aerodynamic derivatives. The propulsion system of the Aerosonde UAV includes a piston engine and a fixed pitch propeller. The aircraft inertia parameters (mass, CG position, moment of inertia) are computed taking into account the fuel usage. The simulation also considers changes in gravity and earth magnetic field with aircraft position.

A state-feedback model predictive control (MPC) guidance scheme is used as part of the aircraft control (autopilot). The MPC guidance scheme solves a constrained quadratic programming problem at a sample time of two seconds

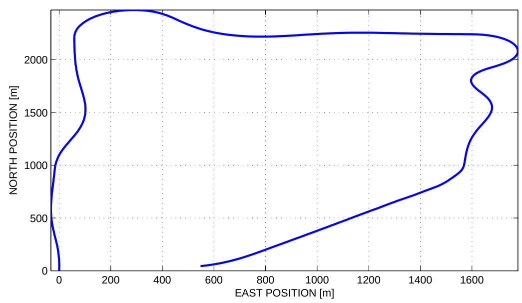

Fig. 3. UAV Trajectory in North-East Plane

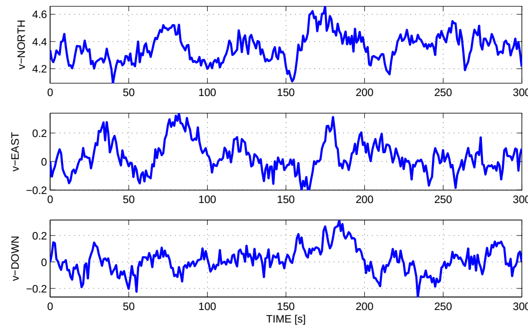

Fig. 4. Wind Velocity in Inertial Frame

to compute the yaw rate command. A simple PI altitude hold controller is a part of the autopilot. The details of the autopilot are not relevant here. The only important thing to note is that the aircraft can track a given trajectory in the presence of winds. This allows us to seed various faults during different flight regimes to validate our estimation algorithms. We use the autopilot to guide the aircraft to fly in an almost closed path. The aircraft trajectory in Figure 3 shows that the controller performs fairly well even in the presence of winds to follow the given north-east waypoints.

The aircraft simulation applies von Karman turbulence shaping filters for longitudinal, lateral and vertical components to three white noise sources. The von Karman filter parameters depend upon the background wind magnitude and the current aircraft altitude. Wind shear effects are considered on the angular velocities and accelerations for pitch and yaw.

The aircraft dynamical model has a set of ordinary differential equations (ODEs) that are numerically integrated. The ODE system is stiff due to the different time scales, and large differences (several orders of magnitude) in the magnitude of the variables in the simulation. The PC based simulation runs for 300 seconds with a fixed step Runge-Kutta integration of ODEs. The fixed step size is 50 milliseconds (ms). The sensor and control data is sampled at a 500 ms sampling interval before being used for estimation. The diagnostic algorithms are embedded in the simulation for online fault estimation.

A. *Modeled Faults*

The MH MAP estimation algorithm offers great flexibility in the estimation of parametric faults. We can estimate constant, step, monotonic, and non-monotonic faults using this algorithm. The only limiting factor is the observability of the unknown faults through available sensor data. The faults chosen in this example pertain to the changes in the lift and drag coefficients of each wing. The ability to simultaneously estimate these changes in the wing aerodynamic coefficients can help us detect evolving structural damage to the aircraft. These chosen parameters are thus representative of a variety of real fault scenarios. The fault vector consists of the following four faults that need to be estimated by the proposed scheme:

$$f = \begin{bmatrix} \text{Right Wing Lift Loss} \\ \text{Right Wing Drag Increase} \\ \text{Left Wing Lift Loss} \\ \text{Left Wing Drag Increase} \end{bmatrix},$$

where the lift loss and drag increase are introduced as a percentage change in the lift and drag coefficients of the UAV.

The standard Aerosonde model developed using the Aerosim block set assumes aerodynamic symmetry, *i.e.*, the total computed aerodynamic force is applied at the longitudinal axis of the aircraft. To model the faults in the simulation, the calculation of the aerodynamic force was modified. The modified version computes the aerodynamic force for each wing separately. The point of application of the total aerodynamic force of each wing is assumed at the center of each wing. The calculations for the total aerodynamic moment were also modified to take into account the aerodynamic forces at each wing.

In addition to these four faults, some other fault parameters that may be easily modeled may include propulsion loss during the UAV flight. Introducing a change in the pitch moment of the aircraft can correspond to tail/actuator damage, and it can be modeled readily in the current simulation.

B. *Simulation results*

The nonlinear simulation is carried out in the presence of strong winds. The wind speeds are of the order of $20-25$ percent of the aircraft speed. The wind gusts combined with the turbulence effects are shown in Figure 4. A 10 second low pass filtered knowledge of the wind speeds is provided to the estimation scheme, *i.e.*, $\tau = 10$ for the filter in (3). The estimation algorithms assume no knowledge of wind angular rates which are very difficult to measure.

Figure 5 shows the linear and angular acceleration residuals for the Aerosonde computed using (2). The linear acceleration residuals (a_x, a_y, a_z) are measured in m/s^2, and the angular acceleration residuals $(\alpha_x, \alpha_y, \alpha_z)$ are measured in rad/s^2. The time constant for the low pass filter $F_a(s)$ in (5) and (6) used to smooth the prediction residuals is 4 seconds. The time constant of the low pass smoother affects the speed of the estimation response and its noise rejection

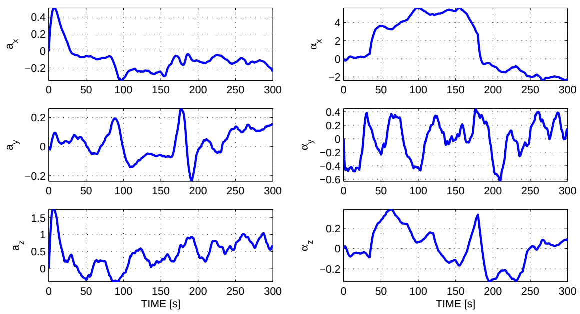

Fig. 5. Prediction Residuals

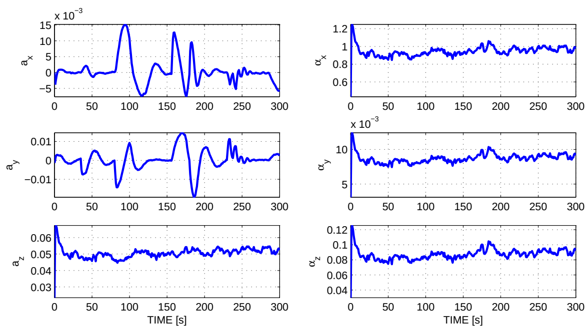

Fig. 6. Sensitivity to Right Wing Lift Loss

properties. A higher value of time constant implies slow detection of fault parameters but yields better noise rejection.

The fault signatures for each of the four faults are computed online using (8). The fault signatures don't need to be computed at the data sampling rate. We can substantially reduce the burden on the embedded filter by computing the fault signatures only once for each flight regime, and reusing them until the aircraft enters a different regime. Due to the presence of substantial winds in the simulation, we compute the fault signatures every 500 ms to obtain accurate fault estimates during different flight maneuvers. The fault sensitivity for percentage change in right wing lift coefficient is shown in Figure 6.

The residuals and fault signatures serve as input to the moving horizon maximum a posteriori probability (MH MAP) estimation scheme. The estimation algorithm solves the quadratic programming problem (17) with linear constraints (16) at each step for the four monotonic faults. A moving horizon window size of $N = 50$ is used in this example. The estimation update is every 500 ms, The fault estimates obtained using the proposed MH MAP estimation scheme are shown in Figure 7. The estimation algorithms clearly recover the seeded faults quickly and accurately even in the presence of strong winds.

V. Conclusion

In this paper we present a statistical estimation approach for online estimation of the unknown time varying fault parameters. The fault parameters are a priori known to

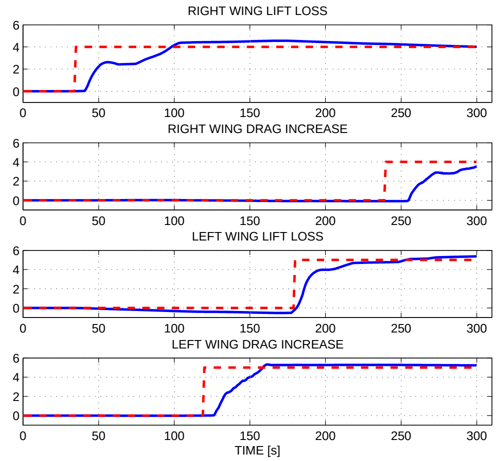

Fig. 7. Moving Horizon MAP Fault Estimates

be monotonic. The estimates are obtained by solving a linearly constrained quadratic programming problem at each step. The approach is validated through application to the Aerosonde unmanned aerial vehicle.

References

[1] R. Isermann. Process fault detection based on modeling and estimation methods—a survey. *Automatica*, 20(4):387–404, 1984.

[2] J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, 8(6):3–11, 1988.

[3] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy-a survey and some new results. *Automatica*, 26(3):459–474, 1990.

[4] J. Chen and R. J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[5] C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46(06):853–865, June 2001.

[6] X. Zhang, M. Polycarpou, and T. Parisini. A robust detection and isolation scheme for abrupt and incipientfaults in nonlinear systems. *IEEE Transactions on Automatic Control*, 47(04):576–593, April 2002.

[7] L. Smith, P. R. Chandler, and M. Pachter. Regularization techniques for real-time identification of aircraft parameters. In *AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, August 1997.

[8] S. Glavaski and M. Elgersma. Active aircraft fault detection and isolation. In *Proceedings of IEEE Systems Readiness Technology Conference AUTOTESTCON*, pages 692–705, August 2001.

[9] S. Brunke and M. Campbell. Estimation architecture for future autonomous vehicles. In *Proceedings of American Control Conference*, pages 1108–1114, June 2002.

[10] 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.

[11] 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.

[12] Aerosonde, aircraft overview. Aerosonde unmanned aerial vehicle. Web site: `http://www.aerosonde.com/aircraft`.

[13] T. Kobayashi and D. L. Simon. Evaluation of an enhanced bank of kalman filters for in-flight aircraft engine sensor fault diagnostics. *Journal of Engineering for Gas Turbines and Power*, 127(03):497–504, July 2005.

[14] S. Samar, D. Gorinevsky, and S. Boyd. Model predictive estimation of evolving faults. In *Proceedings of the American Control Conference*, Minneapolis, Minnesota, June 2006.

[15] AeroSim, Matlab/Simulink block library . Aerosim blockset. Web site: `http://www.u-dynamics.com/aerosim/default.htm`.