

Implementation of real-time moving horizon estimation for robust air data sensor fault diagnosis in the RECONFIGURE benchmark[★]

Yiming Wan^{*} Tamas Keviczky^{*}

^{*} Delft University of Technology, 2628CD, Delft, The Netherlands
(e-mail: y.wan@tudelft.nl; t.keviczky@tudelft.nl).

Abstract: This paper presents robust fault diagnosis and estimation for the calibrated airspeed and angle-of-attack sensor faults in the RECONFIGURE benchmark. We adopt a low-order longitudinal model augmented with wind dynamics. In order to enhance sensitivity to faults in the presence of winds, we propose a constrained residual generator by formulating a constrained moving horizon estimation problem and exploiting the bounds of winds. The moving horizon estimation problem requires solving a nonlinear program in real time, which is challenging for flight control computers. This challenge is addressed by adopting an efficient structure-exploiting algorithm within a real-time iteration scheme. Specific approximations and simplifications are performed to enable the implementation of the algorithm using the Airbus graphical symbol library for industrial validation and verification. The simulation tests on the RECONFIGURE benchmark over different flight points and maneuvers show the efficacy of the proposed approach.

© 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Fault detection and isolation, moving horizon estimation, real-time computation.

1. INTRODUCTION

In aircraft applications, the industrial state-of-the-art for sensor fault detection and isolation (FDI) relies on triplex hardware redundancy, and performs a majority voting scheme to compute a consolidated value by discarding any failed sources (Goupil et al., 2015). This scheme works well if only one sensor source becomes faulty, but it would be inadequate to address simultaneous multiple sensor faults within the triplex redundancy. As currently investigated in the RECONFIGURE project (Goupil et al., 2015), one possibility to extend guidance and control functionalities for future aircraft could be the incorporation of analytical redundancy to detect, isolate, and estimate sensor faults without adding new redundant sensors.

A crucial issue with the analytical redundancy based FDI technique in aircraft applications is how to simultaneously maintain its robustness to wind disturbances and optimize its fault sensitivity (Marzat et al., 2012). In Eykeren and Chu (2014); Hansen and Blanke (2014), an extended Kalman filter based FDI method was proposed based on the assumption of constant winds. Without any limiting assumption about wind dynamics, the disturbance decoupling method based on differential geometry was used in Castaldi et al. (2014) to perfectly decouple the wind effect in the generated residual signal. A similar geometric FDI method was also discussed in Vanek et al. (2011) to deal with multiplicative model uncertainties in an aircraft example. In Marcos et al. (2005), a robust H_∞ FDI filter

was proposed to estimate the fault signal while attenuating the disturbance effect for a transport aircraft. A sliding mode linear parameter varying fault reconstruction scheme was applied in Alwi and Edwards (2014) to yaw rate sensor faults with extensive industrial validation and verification (V&V). A combination of the nullspace based residual generator and the signal-based method were validated in Ossmann and Joos (2016) for the angle-of-attack (AOA) sensor faults in the RECONFIGURE project.

In our previous work (Wan et al., 2016), we have proposed a constrained residual generator to enhance fault sensitivity for simultaneous AOA and calibrated airspeed (VCAS) sensor fault diagnosis in the presence of winds. The constrained residual generator is formulated as a constrained moving horizon estimation (MHE) problem which requires solving a nonlinear program in real time. Compared to conventional unconstrained residual generators, the incorporation of constraints in residual generation further improves fault sensitivity without sacrificing disturbance robustness. Other relevant aircraft FDI methods related to the above mentioned robustness issue include, but not limited to,

This paper presents our continued efforts in implementing our MHE based FDI method proposed in the above reference for real-time computation on flight control computers (FCCs). This is a challenging task because computationally intensive nonlinear programming has to be solved within a short sampling interval. Furthermore, for industrial V&V, the proposed method needs to be implemented using the Airbus graphical symbol library SAO (Briere and Traverse, 1993; Fernández et al., 2015). This SAO library includes a significantly limited set of

[★] This work has received funding from the European Union's Seventh Framework Programme (FP7-RECONFIGURE/2007-2013) under grant agreement no. 314544.

mathematical operation blocks, e.g., no single block for matrix manipulations, thus making the implementation of advanced computationally intensive algorithms even more difficult. Our algorithm implementation adopts a real-time iteration scheme with interior-point (IP) sequential quadratic programming (SQP) strategies. It ensures fixed computational cost per sample by limiting the number of iterations and admitting suboptimal solutions. Specific approximations and simplifications are taken to speed up computation. This implementation has been successfully tested in a number of scenarios using the high-fidelity non-linear RECONFIGURE benchmark. Its real-time applicability is preliminarily confirmed by Airbus, although future work is needed to further reduce its computational cost.

2. BENCHMARK PROBLEM

One of the objectives in the RECONFIGURE project is to exploit analytical redundancy to address simultaneous multiple AOA and VCAS sensor faults, which cannot be easily handled by the conventional triplex monitoring technique. The purpose is to detect and isolate any faulty AOA and VCAS sensors, and at the same time, to provide reliable estimation of AOA and VCAS. Table 1 lists eight benchmark scenarios considered in this paper. They include 4 different maneuvers, and cover 8 flight points at different combinations of low or high altitudes and speeds. Horizontal and vertical winds are present in all the considered scenarios. The fault types include oscillation, jamming, runaway, bias, and non-return to zero (NRZ). Examples of these fault signal profiles can be found in Goupil et al. (2015). In all the listed fault cases, faults occur simultaneously in two of the three redundant sensors of AOA or VCAS.

2.1 Modeling of longitudinal motions and wind dynamics

Since only longitudinal dynamics is investigated in this paper, the following model is adopted:

$$\begin{cases} \dot{\alpha}(t) = f(\alpha(t), \Theta(t)) + u_{\alpha}(t) \\ \dot{\mathbf{w}}(t) = \mathbf{u}_w(t) \\ \mathbf{y}(t) = h(\alpha(t), \mathbf{w}(t), \Theta(t)) \\ \mathbf{y}_m(t) = \mathbf{y}(t) + \mathbf{n}(t) \end{cases} \quad (1)$$

with the definitions $\Theta = [V_g \ \theta \ q \ n_x \ n_z \ z]^T$, $\mathbf{w} = [W_x \ W_z]^T$, $\mathbf{u}_w = [u_{w,x} \ u_{w,z}]^T$, $\mathbf{y} = [\alpha \ V_z \ V_c]^T$, and $\mathbf{n} = [n_{\alpha} \ n_{V_z} \ n_{V_c}]^T$. The system outputs $\mathbf{y}(t)$ include AOA α , vertical speed V_z , and VCAS V_c . The output measurements $\mathbf{y}_m(t)$ are corrupted with white Gaussian measurement noises $\mathbf{n}(t)$. W_x and W_z represent horizontal and vertical wind speeds, respectively. The model parameter Θ consists of ground speed V_g , pitch angle θ , pitch rate q , horizontal load factor n_x , vertical load factor n_z , and altitude z , which are all measurable. The input noise u_{α} accounts for the model mismatch. The output equations in (1) for V_z and V_c are $V_z = h_{V_z}(\alpha, \mathbf{w}, \Theta)$ and $V_c = h_{V_c}(\alpha, \mathbf{w}, \Theta)$, respectively. For each redundant AOA sensor measurement $\alpha_m^{(i)}$ or VCAS sensor measurement $V_{c,m}^{(i)}$, $i = 1, 2, 3$, the latent sensor faults $f_{\alpha}^{(i)}$ and $f_{V_c}^{(i)}$ are additive, i.e.,

$$\alpha_m^{(i)} = \alpha + f_{\alpha}^{(i)} + n_{\alpha}^{(i)}, \quad V_{c,m}^{(i)} = V_c + f_{V_c}^{(i)} + n_{V_c}^{(i)}.$$

The first-order integrating model in (1) is a simple yet powerful approximation of the wind dynamics. $u_{w,x}$ and $u_{w,z}$ represent horizontal and vertical wind accelerations.

The system model (1) is adopted due to several considerations: a) it avoids using other air data measurements which are considered as unreliable in the presence of AOA or VCAS sensor faults, and involves only inertial sensors; b) it includes no aerodynamic parameters, avoiding the issue of robustness to uncertain aerodynamic parameters; c) its low state dimensions are attractive for real-time computation. The readers are referred to Wan et al. (2016) for more details about the model (1).

3. FAULT-TOLERANT MOVING HORIZON ESTIMATION SCHEME

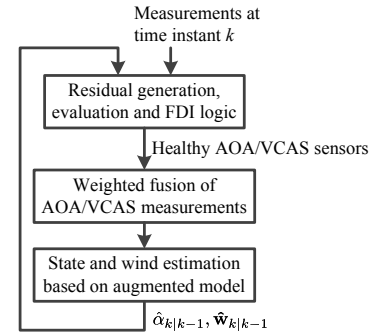


Fig. 1. Fault detection and isolation scheme

As depicted in Figure 1, our FDI scheme consists of three consecutive steps: a) isolating faulty AOA/VCAS sensors based on generated residual signals; b) fusing the healthy AOA and VCAS sensors into two weighted mean values; c) estimating states and winds. The residual signals for FDI are generated as the difference between the AOA/VCAS measurements $\{\alpha_{m,k}^{(i)}, V_{c,m,k}^{(i)}\}$ and their one-step-ahead predictions $\{\hat{\alpha}_{k|k-1}, \hat{V}_{c,k|k-1}\}$, i.e.,

$$r_{\alpha,k}^{(i)} = \alpha_{m,k}^{(i)} - \hat{\alpha}_{k|k-1}, \quad r_{V_c,k}^{(i)} = V_{c,m,k}^{(i)} - \hat{V}_{c,k|k-1}, \quad i = 1, 2, 3.$$

Here, the index k denotes the samples at time instant t_k . The residual signals are evaluated by their root mean square (RMS) values over a sliding window:

$$J_{\star,k}^{(i)} = \sqrt{\frac{1}{N_{\text{eval}}} \sum_{j=k-N_{\text{eval}}+1}^k (r_{\star,j}^{(i)})^2} \quad (2)$$

where \star represents “ α ” and “ V_c ”, N_{eval} is the length of residual evaluation window. With suitable threshold $J_{\star,\text{th}}$, the i th AOA or VCAS sensor is concluded to be faulty if we have $J_{\star,k}^{(i)} > J_{\star,\text{th}}$ for n_d times during the past time window $[k - N_{\text{eval}} + 1, k]$, which allows a confirmation time for the detected fault.

The redundant AOA sensors identified as fault-free are fused into a weighted mean measurement α_m :

$$\alpha_{m,k} = \sum_{i \in \{J_{\alpha,k}^{(i)} \leq J_{\alpha,\text{th}}\}} \beta_{\alpha,k}^{(i)} \alpha_{m,k}^{(i)}, \quad (3)$$

$$\beta_{\alpha,k}^{(i)} = \frac{1}{\sum_{j \in \{J_{\alpha,k}^{(j)} \leq J_{\alpha,\text{th}}\}} \frac{1}{(J_{\alpha,k}^{(j)})^2}} \frac{1}{(J_{\alpha,k}^{(i)})^2}.$$

Table 1. Test scenarios. (“Yes” and “No” represents the existence and non-existence of the horizontal/vertical winds. “F” and “H” stand for “Faulty” and “Healthy”, respectively.)

#	Maneuver	Altitude [ft]	Speed [kts]	Horizontal wind	Vertical wind	Fault type	AOA sensors			VCAS sensors		
							$\alpha_m^{(1)}$	$\alpha_m^{(2)}$	$\alpha_m^{(3)}$	$V_{c,m}^{(1)}$	$V_{c,m}^{(2)}$	$V_{c,m}^{(3)}$
1	Load factor control	7475	207	Yes	No	NRZ	F	F	H	F	F	H
2		1565	203	Yes	No	Oscillation	F	F	H	F	F	F
3		15896	277	Yes	Yes	Oscillation	F	F	H	F	F	H
4	Flight path angle control	31331	322	Yes	Yes	Bias	H	H	H	F	F	H
5		30666	332	Yes	Yes	Jamming	H	H	H	F	F	H
6		15658	351	Yes	Yes	Runaway	H	H	H	F	F	H
7	AOA protection	943	162	Yes	Yes	Jamming	F	F	H	F	F	H
8		968	173	Yes	Yes	NRZ	H	H	H	F	F	H

Similar to Eykeren and Chu (2014), the above adaptive weights $\beta_{\alpha,k}^{(i)}$ are computed from the residual RMS values, so that the sensors with larger residual RMS values are assigned with lower weights. We follow the same procedure to compute the weights $\beta_{vc,k}^{(i)}$ and the weighted mean value $V_{c,m,k}$ from the VCAS sensors identified as fault-free. Before a faulty sensor is detected, the undetected faulty sensor is given a lower weight in (3) due to its larger residual RMS value, therefore the weighted fusion reduces the estimation errors caused by fault detection delays.

The state and wind estimation algorithm computes the one-step ahead predictions $\hat{\alpha}_{k|k-1}$ and $\hat{V}_{c,k|k-1}$ from the filtered estimates $\hat{\alpha}_{k-1|k-1}$ and $\hat{\mathbf{w}}_{k-1|k-1}$ based on the model (1). A nonlinear constrained MHE problem is formulated to exploit the bounds of α , \mathbf{w} , u_α , and \mathbf{u}_w in residual generation, which will be explained in Section 4. It is well known that the incorporation of state constraints in a state estimation problem can improve the estimation accuracy (Simon, 2010). Moreover, from the FDI point of view, compared to conventional unconstrained residual generation, the incorporation of constraints in residual generation can improve fault sensitivity without reducing disturbance robustness Wan et al. (2016).

4. MOVING HORIZON ESTIMATION AND ITS REAL-TIME IMPLEMENTATION

The nonlinear constrained MHE method is the core of our proposed FDI scheme given in Section 3. In this section, we will first formulate the MHE problem, and then show how it is implemented with SAO for real-time computation.

The MHE framework builds on the discrete-time approximation of the continuous-time model (1):

$$\alpha_{k+1} = \alpha_k + t_s f(\alpha_k, \Theta_k) + t_s u_{\alpha,k}, \quad (4a)$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + t_s \mathbf{u}_{w,k}, \quad (4b)$$

$$\bar{\mathbf{y}}_{m,k} = h(\alpha_k, \mathbf{w}_k, \Theta_k) + \bar{\mathbf{n}}_k, \quad (4c)$$

where t_s is the sampling interval, (4a) and (4b) are obtained via approximated numerical integration applied to (1). In (4c), the output vector $\bar{\mathbf{y}}_{m,k}$ consists of the vertical speed measurement $V_{z,m}$ and the two fused measurements α_m and $V_{c,m}$ defined in (3), i.e.,

$$\bar{\mathbf{y}}_{m,k} = [\alpha_{m,k} \ V_{z,m,k} \ V_{c,m,k}]^T. \quad (5)$$

Then $\bar{\mathbf{n}}_k$ is defined similarly to (3).

Given a moving horizon including N samples of output measurements $\{\bar{\mathbf{y}}_{m,l}, \bar{\mathbf{y}}_{m,l+1}, \dots, \bar{\mathbf{y}}_{m,k}\}$ ($l = k - N + 1$) at time instant k , the MHE problem is formulated as

$$\min_{\mathbf{x}_i, \mathbf{u}_i} \frac{1}{2} \|\mathbf{x}_l - \hat{\mathbf{x}}_{l|k-1}\|_{\mathbf{P}^{-1}}^2 + \frac{1}{2} \sum_{i=l}^{k-1} \|\mathbf{u}_i\|_{\mathbf{Q}^{-1}}^2 \quad (6a)$$

$$+ \frac{1}{2} \sum_{i=l}^k \|\bar{\mathbf{y}}_{m,i} - h(\mathbf{x}_i, \Theta_i)\|_{\mathbf{R}_i^{-1}}^2$$

$$\text{s.t. } \mathbf{x}_{i+1} = F(\mathbf{x}_i, \mathbf{u}_i, \Theta_i), \quad (6b)$$

$$\mathbf{u}^{\text{LB}} \leq \mathbf{u}_i \leq \mathbf{u}^{\text{UB}}, \quad i = l, \dots, k-1,$$

$$\mathbf{x}^{\text{LB}} \leq \mathbf{x}_i \leq \mathbf{x}^{\text{UB}}, \quad i = l, \dots, k,$$

where

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \alpha \\ \mathbf{w} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} u_\alpha \\ \mathbf{u}_w \end{bmatrix}, \mathbf{P} = \begin{bmatrix} p_\alpha & \mathbf{0} \\ \mathbf{0} & p_w \mathbf{I}_2 \end{bmatrix}, \\ \mathbf{Q} &= \begin{bmatrix} q_\alpha & \mathbf{0} \\ \mathbf{0} & q_w \mathbf{I}_2 \end{bmatrix}, \mathbf{R}_i = \text{diag}(R_{\alpha,i}, R_{vz}, R_{vc,i}), \\ R_{\star,i} &= \sum_{j \in \{J_{\star,k}^{(j)} \leq J_{\star,\text{th}}\}} \left(\beta_{\star,k}^{(j)} \right)^2 R_{\star}, \end{aligned} \quad (7)$$

with \star representing “ α ” and “ vc ”. The function $F(\cdot)$ in (6b) represents the right-hand side of (4a) and (4b). Detailed explanation about the above MHE formulation is referred to Rawlings and Bakshi (2006). The positive scalars p_α , p_w , R_α , R_{vz} , and R_{vc} are tuning parameters in the weighting matrices, and $R_{\star,i}$ takes its form in (7) according to the definition of the fused estimates in (3). These parameters can be tuned for the trade-offs between sensitivity to faults and robustness to disturbances, which has been analyzed in Wan et al. (2016).

To solve the nonlinear programming problem (6), we adopt a generalized Gauss-Newton (GGN) SQP strategy and use the IP method to solve each quadratic programming (QP) subproblem. For real-time computation within a short sampling interval, we perform only one SQP iteration per sample and fix the number of iterations in solving each QP subproblem. This real-time iteration strategy, which has been reported in the literature, e.g., in Kuhl et al. (2011), admits sub-optimality of the solution to enable fixed computational cost per sample. In the following subsections, we will present the details of its SAO implementation.

4.1 Generalized Gauss-Newton SQP

At each time instant k , the original problem (6) is first linearized by applying the GGN SQP strategy, around the solution $\hat{\mathbf{x}}_{i|k-1}$ ($i = l, \dots, k$) and $\hat{\mathbf{u}}_{i|k-1}$ ($i = l, \dots, k-1$) of the previous time instant $k-1$. Note that $\hat{\mathbf{x}}_{k|k-1} = F(\hat{\mathbf{x}}_{k-1|k-1}, \hat{\mathbf{u}}_{k-1|k-1}, \Theta_{k-1})$ is the predicted estimate at

Algorithm 1 Primal barrier interior-point algorithm

Initialization: $\kappa = \kappa_{\text{init}}$, $\{\Delta \mathbf{x}_i^- = 0, \Delta \mathbf{u}_i^- = 0\}_{i=l, \dots, k}$.

for $j = 0 \rightarrow n_\kappa n_{QP} - 1$ **do**

Linearization: compute all the quantities in (13).

Compute the search direction $\{\Delta^2 \mathbf{x}_i, \Delta^2 \mathbf{u}_i\}_{i=l, \dots, k}$ with Algorithm 2.

Line search:

$$s_j = \max \left\{ 2^{-n_s} \left| \begin{array}{l} \Delta \star_i^{\text{LB}} \leq \eta_{\star,i}(2^{-n_s}) \leq \Delta \star_i^{\text{UB}}, \\ i = l, \dots, k, \text{ and } n_s \leq n_s^{\text{max}} \\ \text{is a nonnegative integer} \end{array} \right. \right\}$$

where $\eta_{\star,i}(\mu) = \Delta \star_i^- + \mu \Delta^2 \star_i$ with \star representing \mathbf{u} and \mathbf{x} . s_j is rounded to zero if $\eta_{\star,i}(2^{-n_s^{\text{max}}})$ cannot satisfy all the inequality constraints.

Update:

$$(\Delta \mathbf{x}_i^-, \Delta \mathbf{u}_i^-) \leftarrow (\Delta \mathbf{x}_i^-, \Delta \mathbf{u}_i^-) + s_j(\Delta^2 \mathbf{x}_i, \Delta^2 \mathbf{u}_i)$$

$$\kappa \leftarrow 0.1\kappa \text{ if } \text{mod}(j, n_\kappa) = 0$$

end for

Solution to the QP (8): $(\Delta \mathbf{x}_{i|k}, \Delta \mathbf{u}_{i|k}) \leftarrow (\Delta \mathbf{x}_i^-, \Delta \mathbf{u}_i^-)$

time instant $k - 1$. This leads to the following QP subproblem:

$$\min_{\Delta \mathbf{x}_i, \Delta \mathbf{u}_i} \frac{1}{2} \|\Delta \mathbf{x}_i\|_{\mathbf{P}^{-1}}^2 + \frac{1}{2} \sum_{i=l}^{k-1} \|\mathbf{r}_{u,i} - \Delta \mathbf{u}_i\|_{\mathbf{Q}^{-1}}^2 \quad (8a)$$

$$+ \frac{1}{2} \sum_{i=l}^k \|\mathbf{r}_{y,i} - \mathbf{C}_i \Delta \mathbf{x}_i\|_{\mathbf{R}_i^{-1}}^2$$

$$\text{s.t. } \Delta \mathbf{x}_{i+1} = \mathbf{f}_i + \mathbf{A}_i \Delta \mathbf{x}_i + \mathbf{B}_i \Delta \mathbf{u}_i, \quad (8b)$$

$$\Delta \mathbf{u}_i^{\text{LB}} \leq \Delta \mathbf{u}_i \leq \Delta \mathbf{u}_i^{\text{UB}}, \quad i = l, \dots, k-1,$$

$$\Delta \mathbf{x}_i^{\text{LB}} \leq \Delta \mathbf{x}_i \leq \Delta \mathbf{x}_i^{\text{UB}}, \quad i = l, \dots, k,$$

where

$$\Delta \mathbf{x}_i = \mathbf{x}_i - \hat{\mathbf{x}}_{i|k-1}, \quad \Delta \mathbf{u}_i = \mathbf{u}_i - \hat{\mathbf{u}}_{i|k-1},$$

$$\mathbf{r}_{u,i} = -\hat{\mathbf{u}}_{i|k-1}, \quad \mathbf{r}_{y,i} = \bar{\mathbf{y}}_{m,i} - h(\hat{\mathbf{x}}_{i|k-1}, \Theta_i),$$

$$\mathbf{A}_i = \text{diag}(1 + t_s \nabla_{\mathbf{x}_i} f, 1, 1), \quad \mathbf{B}_i = t_s \mathbf{I}_{n_u}, \quad \mathbf{C}_i = \nabla_{\mathbf{x}_i} h, \quad (9)$$

$$\mathbf{f}_i = \hat{\mathbf{x}}_{i|k-1} - \hat{\mathbf{x}}_{i+1|k-1} + t_s F(\hat{\mathbf{x}}_{i|k-1}, \hat{\mathbf{u}}_{i|k-1}),$$

$$\Delta \mathbf{u}_i^{\text{LB}} = \mathbf{u}^{\text{LB}} - \hat{\mathbf{u}}_{i|k-1}, \quad \Delta \mathbf{u}_i^{\text{UB}} = \mathbf{u}^{\text{UB}} - \hat{\mathbf{u}}_{i|k-1},$$

$$\Delta \mathbf{x}_i^{\text{LB}} = \mathbf{x}^{\text{LB}} - \hat{\mathbf{x}}_{i|k-1}, \quad \Delta \mathbf{x}_i^{\text{UB}} = \mathbf{x}^{\text{UB}} - \hat{\mathbf{x}}_{i|k-1}.$$

Its solution $\{\Delta \mathbf{x}_{i|k}, \Delta \mathbf{u}_{i|k}\}$ is computed by using the algorithm given in Section 4.2. Finally, the solution to the original problem (6) is updated as $\hat{\mathbf{x}}_{i|k} = \hat{\mathbf{x}}_{i|k-1} + \Delta \mathbf{x}_{i|k}$ and $\hat{\mathbf{u}}_{i|k} = \hat{\mathbf{u}}_{i|k-1} + \Delta \mathbf{u}_{i|k}$, and used to initialize the SQP iteration at the next time instant.

4.2 Solving the QP subproblem

An infeasible start primal barrier IP method is adopted to solve the QP subproblem (8). We first replace the inequality constraints in the QP (8) with barrier terms in its objective function, to get the approximate problem

$$\min_{\Delta \mathbf{x}_i, \Delta \mathbf{u}_i} \frac{1}{2} \|\Delta \mathbf{x}_i\|_{\mathbf{P}^{-1}}^2 + \frac{1}{2} \sum_{i=l}^{k-1} \|\mathbf{r}_{u,i} - \Delta \mathbf{u}_i\|_{\mathbf{Q}^{-1}}^2 \quad (10)$$

$$+ \frac{1}{2} \sum_{i=l}^k \|\mathbf{r}_{y,i} - \mathbf{C}_i \Delta \mathbf{x}_i\|_{\mathbf{R}_i^{-1}}^2 + \kappa \phi(\Delta \mathbf{u}, \Delta \mathbf{x})$$

$$\text{s.t. } \Delta \mathbf{x}_{i+1} = \mathbf{f}_i + \mathbf{A}_i \Delta \mathbf{x}_i + \mathbf{B}_i \Delta \mathbf{u}_i, \quad i = l, \dots, k-1$$

where $\kappa > 0$ is a barrier parameter, and the function $\phi(\cdot)$ is the log barrier defined as

$$\phi(\Delta \mathbf{u}, \Delta \mathbf{x}) = \sum_{i=l}^{k-1} \sum_{j=1}^{n_u} \phi_u(\Delta \mathbf{u}_i(j)) + \sum_{i=l}^k \sum_{j=1}^{n_x} \phi_x(\Delta \mathbf{x}_i(j))$$

$$\phi_\star(\Delta \star_i(j)) = -\log(\Delta \star_i^{\text{UB}}(j) - \Delta \star_i(j))$$

$$- \log(\Delta \star_i(j) - \Delta \star_i^{\text{LB}}(j)), \quad (11)$$

with \star representing \mathbf{u} and \mathbf{x} , and j referencing to the j th entry of the vector \star . A sequence of the approximate problems (10) are solved iteratively for a decreasing sequence of values of κ , as described in Algorithm 1. For real-time computation, the number of the κ values in the sequence is fixed to n_κ , and we perform n_{QP} iterations for each approximate problem (10) with a particular value of κ . A simple backtracking line search is used to ensure that the inequality constraints are satisfied at all iterations.

At each iteration in Algorithm 1, the Karush-Kuhn-Tucker (KKT) system of the approximate problem (10) is linearized and solved to compute the search direction represented by $\Delta^2 \mathbf{x}_i$ and $\Delta^2 \mathbf{u}_i$, $i = l, \dots, k$. This is equivalent to solving the following linear MHE problem with only equality constraints, omitting detailed explanations for the sake of brevity:

$$\min_{\Delta^2 \mathbf{x}_i, \Delta^2 \mathbf{u}_i} \frac{1}{2} \|\bar{\mathbf{r}}_x - \Delta^2 \mathbf{x}_l\|_{\mathbf{P}^{-1}}^2 + \frac{1}{2} \sum_{i=l}^{k-1} \|\bar{\mathbf{r}}_{u,i} - \Delta^2 \mathbf{u}_i\|_{\bar{\mathbf{Q}}_i^{-1}}^2$$

$$+ \frac{1}{2} \sum_{i=l}^k \|\bar{\mathbf{r}}_{y,i} - \bar{\mathbf{C}}_i \Delta^2 \mathbf{x}_i\|_{\bar{\mathbf{R}}_i^{-1}}^2$$

$$\text{s.t. } \Delta^2 \mathbf{x}_{i+1} = -\mathbf{r}_{p,i} + \mathbf{A}_i \Delta^2 \mathbf{x}_i + \mathbf{B}_i \Delta^2 \mathbf{u}_i,$$

$$i = l, \dots, k-1. \quad (12)$$

where we define

$$g_{\star,i} = \text{vect} \left(\left\{ \nabla \phi_\star(\Delta \star_i^-(j)) \right\}_{j=1,2,3} \right), \quad (13a)$$

$$\mathbf{L}_{\star,i} = \text{diag} \left(\left\{ \sqrt{\nabla^2 \phi_\star(\Delta \star_i^-(j))} \right\}_{j=1,2,3} \right), \quad (13b)$$

with \star being \mathbf{u} and \mathbf{x} ,

$$\mathbf{r}_{p,i} = -\mathbf{f}_i - \mathbf{A}_i \Delta \mathbf{x}_i^- - \mathbf{B}_i \Delta \mathbf{u}_i^- + \Delta \mathbf{x}_{i+1}^-, \quad (13c)$$

$$\bar{\mathbf{r}}_x = -\Delta \mathbf{x}_l^-, \quad \bar{\mathbf{r}}_{u,i} = \mathbf{Q}^{-1} (\mathbf{r}_{u,i} - \Delta \mathbf{u}_i^-) - \kappa g_{u,i}, \quad (13d)$$

$$\bar{\mathbf{r}}_{y,i} = \begin{bmatrix} \mathbf{r}_{y,i} - \mathbf{C}_i \Delta \mathbf{x}_i^- \\ -\sqrt{\kappa} \mathbf{L}_{x,i}^{-1} g_{x,i} \end{bmatrix}, \quad \bar{\mathbf{C}}_i = \begin{bmatrix} \mathbf{C}_i \\ \sqrt{\kappa} \mathbf{L}_{x,i} \end{bmatrix}, \quad (13e)$$

$$\bar{\mathbf{R}}_i = \text{diag}(\mathbf{R}_i, \mathbf{I}_{n_x}), \quad \bar{\mathbf{Q}}_i = (\mathbf{Q}^{-1} + \kappa \mathbf{L}_{u,i}^T \mathbf{L}_{u,i})^{-1}. \quad (13f)$$

Note that with the scalar entries $\{x(i)\}$, $\text{vect}(\{x(i)\})$ and $\text{diag}(\{x(i)\})$ in (13a) and (13b) represent a column vector and a diagonal matrix, respectively. These quantities in (13) are the result of linearizing the KKT system of the QP (10). Inspired by Haverbeke (2011), a structure-exploiting Riccati based approach is adopted to solve the linearized KKT system, which will be detailed in Algorithm 2.

4.3 Implementation aspects

The implementation using SAO is a critical step for achieving real-time computation of the above algorithm. The following aspects have been considered to either speed up computation or simplify the implementation while maintaining good estimation performance.

The overall computational cost is kept small by setting the horizon length N of the MHE problem (6) to be 3, the length N_{eval} of the residual evaluation window in (2) to be 10, the number of iterations n_{κ} and n_{QP} in Algorithm 1 to be both two. Extensive numerical simulations show good results even with these small number of iterations.

The AOA or VCAS measurements should not be involved in the proposed MHE algorithm when all redundant AOA or VCAS sensors are identified as faulty. However, this cannot be done by directly removing AOA or VCAS from the output equation (4c), because the time-varying dimension of the output equation, which implies vectors and matrices of time-varying sizes in the MHE algorithm, cannot be easily handled within the SAO library. To simplify the SAO implementation for the above issue, we let the output equation (4c) remain the same, but set only the first or third row of the matrix \mathbf{C}_i in (8a) to be zero after losing all AOA or VCAS sensors, respectively. In this way, the feedback information from AOA or VCAS becomes ineffective when necessary, and the SAO implementation still works with vectors and matrices of fixed sizes.

Moreover, some approximations in computing the matrices \mathbf{C}_i and $\bar{\mathbf{C}}_i$ have been made as explained next. Let $\mathbf{C}_i(k_1, k_2)$ represent the entry at the k_1 th row and the k_2 th column of the matrix \mathbf{C}_i . Over the flight envelop, the two entries $\mathbf{C}_i(2, 2)$ and $\mathbf{C}_i(3, 3)$ are negligible compared to other entries of \mathbf{C}_i . Therefore we approximate $\mathbf{C}_i(2, 2)$ and $\mathbf{C}_i(3, 3)$ to be zero. $g_{*,i}$ and $\mathbf{L}_{*,i}$ in (13) and the entries of the third row of \mathbf{C}_i involve logarithm and power computations which are approximated by lookup tables.

The observability of the linearized system in (8) is mainly determined by \mathbf{C}_i , because $t_s \nabla_{\mathbf{x}_i} f$ in (9) is approximately zero and \mathbf{A}_i is very close to an identity matrix for all time instants i over the flight envelope. The states become unobservable when all AOA or VCAS sensors are detected to be faulty and removed. This observability issue is due to the fact that all air data sensors are considered to be unreliable and then only inertial sensors can be included in the model described in Section 2.1. Therefore, the following two scenarios cannot be fully addressed by our method:

- When all AOA sensors are removed, neither AOA nor VCAS can be reliably estimated, because all the states α , W_x , and W_z become unobservable.
- When all VCAS sensors are removed, AOA can still be reliably estimated, while VCAS cannot, because α and W_z are still observable, while W_x becomes unobservable.

It should be noted that in the second scenario above, we approximate $\mathbf{C}_i(2, 2)$ to be zero as mentioned in the last paragraph, in order to make the MHE algorithm work with the observable subsystem associated with α and W_z and discard the unobservable W_x .

Computing the search direction by solving the linearized KKT system of (10) dominates the computational cost of Algorithm 1. This step follows Algorithm 2 in the SAO implementation by taking the following strategies:

Algorithm 2 Solve the search direction

```

1: Riccati recursion based factorization:
2:  $\hat{\mathbf{P}}_l = \mathbf{P}$ 
3: for  $i = l \rightarrow k$  do
4:    $\Pi_i \leftarrow \bar{\mathbf{C}}_i \hat{\mathbf{P}}_i$ ,  $\Xi_i \leftarrow (\bar{\mathbf{R}}_i + \Pi_i \bar{\mathbf{C}}_i^T)^{-1}$ ,  $\Omega_i \leftarrow \bar{\mathbf{C}}_i^T \Xi_i$ 
5:    $\mathbf{K}_i \leftarrow \hat{\mathbf{P}}_i \Omega_i$ ,  $\mathbf{P}_i^f \leftarrow \hat{\mathbf{P}}_i - \mathbf{K}_i \Pi_i$ 
6:   if  $i < k$  then
7:      $\hat{\mathbf{P}}_{i+1} \leftarrow \mathbf{A}_i \mathbf{P}_i^f \mathbf{A}_i^T + \mathbf{B}_i \bar{\mathbf{Q}}_i \mathbf{B}_i^T$ 
8:   end if
9: end for
10: Forward recursion:
11:  $\Delta^2 \hat{\mathbf{x}}_l = \bar{\mathbf{r}}_x$ 
12: for  $i = l \rightarrow k$  do
13:    $\check{\mathbf{r}}_i \leftarrow \bar{\mathbf{r}}_{y,i} - \bar{\mathbf{C}}_i \Delta^2 \hat{\mathbf{x}}_i$ ,  $\Delta^2 \mathbf{x}'_i \leftarrow \Delta^2 \hat{\mathbf{x}}_i + \mathbf{K}_i \check{\mathbf{r}}_i$ 
14:   if  $i < k$  then
15:      $\Delta^2 \mathbf{u}'_i \leftarrow \bar{\mathbf{r}}_{u,i}$ 
16:      $\Delta^2 \hat{\mathbf{x}}_{i+1} \leftarrow -\mathbf{r}_{p,i} + \mathbf{A}_i \Delta^2 \mathbf{x}'_i + \mathbf{B}_i \Delta^2 \mathbf{u}'_i$ 
17:   end if
18: end for
19: Backward recursion:
20:  $\Delta^2 \mathbf{x}_k \leftarrow \Delta^2 \mathbf{x}'_k$ ,  $\lambda_{k-1} \leftarrow -\Omega_k \check{\mathbf{r}}_k$ 
21: for  $i = k-1 \rightarrow 0$  do
22:    $\xi_i \leftarrow \mathbf{A}_i^T \lambda_i$ ,  $\Delta^2 \mathbf{u}_i \leftarrow \Delta^2 \mathbf{u}'_i - \bar{\mathbf{Q}}_i \mathbf{B}_i^T \lambda_i$ 
23:    $\Delta^2 \mathbf{x}_i \leftarrow \Delta^2 \mathbf{x}'_i - \mathbf{P}_i^f \xi_i$ 
24:   if  $i > 0$  then
25:      $\lambda_{i-1} \leftarrow \xi_i - \Omega_i (\check{\mathbf{r}}_i + \Pi_i \xi_i)$ 
26:   end if
27: end for

```

- The intermediate results, e.g., Π_i , Ξ_i , Ω_i , \mathbf{K}_i in lines 4-5, $\check{\mathbf{r}}_i$ in line 13, and ξ_i in line 22 of Algorithm 2, are reused in subsequent computations.
- The symmetric or diagonal matrix structures are exploited in all the matrix manipulations.
- To compute Ξ_i in line 4 of Algorithm 2, the block matrix inversion formula is applied so that the inversion of the 6×6 matrix $\bar{\mathbf{R}}_i + \Pi_i \bar{\mathbf{C}}_i^T$ can be reduced to the inversion of several 3×3 matrices and some matrix multiplications. Inverting a 3×3 matrix is computed via the analytical adjugate formula.

5. SIMULATION RESULTS

The test scenarios in Table 1 are simulated under both fault-free and faulty conditions using the RECONFIGURE Functional Engineering Simulator (Goupil et al., 2015; Fernández et al., 2015). To mitigate the effect of wind disturbances and model mismatches, the bounds of wind speed and accelerations are set to be 120 kts and 30 kts/s, and the detection thresholds of AOA and VCAS are $J_{\alpha, \text{th}} = 2.3$ deg and $J_{\alpha, \text{th}} = 12$ kts, respectively. The estimation performance is evaluated by the absolute estimation errors (AEEs) in their maximum and averaged values, and the detection performance is evaluated by the detection delay for each faulty sensor. In all the fault-free scenarios, the maximum AEEs of AOA and VCAS are less than 0.5 deg and 1.9 kts, respectively, while their mean AEEs are even smaller. There is no false alarm in the fault-free runs.

As shown in Table 2, all the sensor faults have been successfully detected and isolated. Most of the detection delays are less than 0.45s, except for 0.92s in Scenario

6-F and 23.278s in Scenario 7-F. The relatively long detection delay for the VCAS sensor faults in Scenario 7-F is due to the trade-off between detection delays and false alarms: the detection threshold is determined to avoid false alarms at the cost of increased detection delays. Despite these detection delays, the maximum AEEs of AOA and VCAS are still satisfying, i.e., 0.82 deg and 1.85 kts, respectively, in all the faulty cases. The only exception is the unacceptable VCAS estimation error in Scenario 2-F. The reason is that all three VCAS sensors are faulty and removed from the MHE algorithm, which makes the horizontal wind unobservable, as explained in Section 4.3. The readers are referred to Wan and Keviczky (2016) for more details of simulation results.

The real-time computational cost of our SAO implementation is 5.8 ms per sample under a preliminary assessment by Airbus. This highlights the feasibility of applying online optimization based MHE methods on FCCs. Future work will focus on further reduction of the algorithm complexity and simplification of the SAO implementation.

Table 2. Simulation results. (In the first column, the indices “i-F” correspond to the i th maneuver with faults, as listed in Table 1)

#	Absolute estimation error		Detection delay	
	AOA max/mean [deg]	VCAS max/mean [kts]	AOA [s]	VCAS [s]
1-F	0.1628/0.0420	1.7748/0.1024	0.378	0.378
2-F	0.4938/0.1134	74.3612/42.3752	0.368	0.368
3-F	0.4661/0.0452	0.8239/0.0349	0.433	0.393
4-F	0.3647/0.0483	0.6010/0.0300	-	0.363
5-F	0.3742/0.0490	0.6111/0.0288	-	0.363
6-F	0.6044/0.0637	1.0238/0.0461	-	0.920
7-F	0.4339/0.2764	0.2978/0.1225	0.398	23.278
8-F	0.3629/0.2266	0.3136/0.1167	-	0.373

6. CONCLUSION

This paper presents the implementation of our proposed constrained MHE based air data sensor FDI method for industrial V&V. The involved nonlinear programming at each time instant is solved by the generalized Gauss-Newton sequential quadratic programming strategy and the interior-point method. For real-time computation, we fix the number of iterations per sample. The structure-exploiting Riccati based method is used to solve the linearized KKT system. Specific algorithmic approximations and simplifications are performed to enable its implementation using the Airbus graphical symbol library. A number of simulation tests on the RECONFIGURE benchmark over different flight points and maneuvers show promising results.

REFERENCES

- Alwi, H. and Edwards, C. (2014). Development and application of sliding mode LPV fault reconstruction schemes for the ADDSAFE benchmark. *Control Engineering Practice*, 31, 148–170.
- Briere, D. and Traverse, P. (1993). Airbus A320/A330/A340 electrical flight controls - a family of fault-tolerant systems. In *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, 616–623. Toulouse, France.
- Castaldi, P., Mimmo, N., and Simani, S. (2014). Differential geometry based active fault tolerant control for aircraft. *Control Engineering Practice*, 32, 227–235.
- Eykeren, L.v. and Chu, Q.P. (2014). Sensor fault detection and isolation for aircraft control systems by kinematic relations. *Control Engineering Practice*, 31, 200–210.
- Fernández, V., Montañó, J., Recupero, C., Kerr, M., Rosa, P., Boada-Bauxell, J., and Dalbies, L. (2015). A tool for industrial verification and benchmarking of FDD/FTC designs. In *Proceedings of the 9th IFAC Symposium on SafeProcess*, 1006–1011. Paris, France.
- Goupil, P., Boada-Bauxell, J., Marcos, A., Rosa, P., Kerr, M., and Dalbies, L. (2015). An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives. In *Proceedings of 9th IFAC Symposium on SafeProcess*, 976–981. Paris, France.
- Hansen, S. and Blanke, M. (2014). Diagnosis of air-speed measurement faults for unmanned aerial vehicles. *IEEE Transactions on Aerospace and Electronic Systems*, 50(1), 224–239.
- Haverbeke, N. (2011). *Efficient Numerical Methods for Moving Horizon Estimation*. Phd thesis, Katholieke Universiteit Leuven.
- Kuhl, P., Diehl, M., Kraus, T., Schloder, J., and Bock, H. (2011). A real-time algorithm for moving horizon state and parameter estimation. *Computers and Chemical Engineering*, 35(1), 71–83.
- Marcos, A., Ganguli, S., and Balas, G. (2005). An application of H_∞ fault detection and isolation to a transport aircraft. *Control Engineering Practice*, 13, 105–119.
- Marzat, J., Piet-Lahanier, H., Damongeot, F., and Walter, E. (2012). Model-based fault diagnosis for aerospace systems: a survey. *Proceedings of the Institution of Mechanical Engineering, Part G: Journal of Aerospace Engineering*, 226(10), 1329–1360.
- Ossmann, D. and Joos, H. (2016). Enhanced detection and isolation of angle of attack sensor faults. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. San Diego, California. Doi:10.2514/6.2016-1135.
- Rawlings, J.B. and Bakshi, B.R. (2006). Particle filtering and moving horizon estimation. *Computers and Chemical Engineering*, 30(10-12), 1529–1541.
- Simon, D. (2010). Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory and Applications*, 4(8), 1303–1318.
- Vanek, B., Szabo, Z., Edelmayer, A., and Bokor, J. (2011). Geometric LPV fault detection filter design for commercial aircrafts. In *Proceedings of the 2011 AIAA Guidance, Navigation, and Control Conference*. Portland. Doi:10.2514/6.2011-6681.
- Wan, Y. and Keviczky, T. (2016). Implementation of real-time moving horizon estimation for robust air data sensor fault diagnosis in the reconfigure benchmark. Submitted to ArXiv.
- Wan, Y., Keviczky, T., and Verhaegen, M. (2016). Robust air data sensor fault diagnosis with enhanced fault sensitivity using moving horizon estimation. Accepted by the 2016 American Control Conference.