Defence Science Journal, Vol. 60, No. 1, January, 2010, pp. 61-75

Novel Redundant Sensor Fault Detection and Accommodation Algorithm for an Air-breathing Combustion System and its Real-time Implementation

Rahee Walambe[*] , Nitin K Gupta[**], Niteen Bhange[*], N Ananthkrishnan[*,**], Ik Soo Park[***], Jong Ho Choi[***], and Hyun Gull Yoon[***]

[*]*Coral Digital Technologies (P) Ltd, Bangalore-560 043*
[**]*IDeA Research & Development (P) Ltd, Pune-411 001*
[***]*Industry University Collaboration Foundation, Chungnam National University, Daejeon, Korea*
[*]*E-mail: walambe.rahee@gmail.com*

ABSTRACT

Failure of sensors used to provide a feedback signal in control system can cause serious deterioration in performance of system, and even instability may be observed. Based on knowledge of aircraft engine systems, the main cause of fault in such air-breathing combustion systems (ACS) with no rotating parts is due to the pressure sensors. Fast online detection of faults before the error grows very large and accommodation is critical to the success of the mission. However, at the same time, it is necessary to avoid false alarms. Hence, early detection of small magnitude faults with acceptable reliability is very challenging, especially in the presence of sensor noise, unknown engine-to-engine variation and deterioration and modeling uncertainty. This paper discusses the novel fault detection and accommodation (FDA) algorithm based on analytical redundancy based technique for ACS.

Keywords: Extended K alman filter, analytical redundancy, air-breathing combustion systems, ACS, fault detection and accommodation, FDA

NOMENCLATURE

A^{th}	Nozzle throat area
α	Angle of attack
B	Backpressure factor
R	Universal gas constant
τ_{ij}	Time constant between station i and j
$\dot{m}_f$	Fuel flow rate
P_1	Free stream pressure
P_4	Static pressure at station 4
P_{4_margin}	Backpressure margin
P_5	Static pressure at station 5
T_{05}	Total temperature at station 5
P_7	Static pressure at station 7
T_{07}	Total temperature at station 7
P_{07}	Total pressure at station 7
$\dot{m}_4$	Fuel flow rate at station 4
$\dot{m}_5$	Fuel flow rate at station 5
$\dot{m}_7$	Fuel flow rate at station 7
$\dot{m}_8$	Fuel flow rate at station 8
$\hat{P}_4^1, \hat{P}_4^2, \hat{P}_4^3$	P_4 measured by three pressure sensors
P_{port}	Pressure at FADS orifice port
$Q_{\text{k}}, _{\text{k}}$	Covariance matrices
FAR	Fuel to air ratio
W_{i}	Weightage factor for i^{th} EKF channel
$\dot{m}_{8\,modeled}$	Modelled value of $\dot{m}_8$ (kg/s)
q	Dynamic pressure
v_k	Unknown measurement noise
w_k	Unknown process noise
x_k	State at time k
ss	Steady state value of the variable
$\hat{x}$	Estimated value of the variable x

1. INTRODUCTION

In recent work[1], a controller for an air-breathing combustion systems (ACS) has designed. The controller's main objective is to regulate the thrust so that the desired acceleration is obtained at all flight conditions while maintaining supercritical intake operation. The performance of the controller is tested by simulating a nominal flight trajectory involving an accelerated climb from 2.1 Mach at 1.4 km altitude to 3.0 Mach at 14.5 km altitude, followed by cruise at that condition. A unique feature of this controller is that it only requires measurement of a single variable, internal to the engine, i.e., the intake

Received 22 September 2009

backpressure. Accurate, reliable and fail-safe measurement of backpressure is a key to this controller design[2]. A fault detection and accommodation (FDA) algorithm is required for this purpose.

The FDA algorithm reported here employs an intelligent analytical redundancy-based FDA algorithm working over triplex redundant backpressure sensor hardware.

2. CONTROLLER DESIGN

2.1 Model and Controller Description

A schematic of the ACS with station numbers marked is shown in Fig.1.

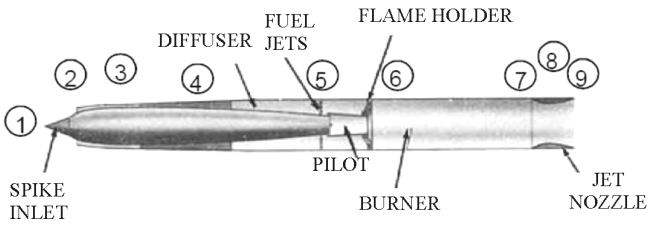


Figure 1. Schematic of air-breathing combustion system.

For simulation, the combustion system is modelled as three sub-systems: Intake, combustor, and nozzle (a separate fuel supply system is also modelled). These sub-systems are linked to produce a global model that correctly represents the physics of the combustion system. A detailed description of the model development and implementation is provided by O'Brian[3], *et al*. and Gupta[4], *et al*.

Only intake backpressure, P_4, is assumed to be measurable. The controller design of the ACS is shown in Fig. 2. Separate PID controllers for fuel flow rate and throat area, with P_4 margin as the commanded variable, are designed at several operating points on the acceleration and cruise segments of the flight. A separate PID controller is designed for the fuel supply system. Various components of the controller are individually tested extensively. Finally, a composite closed-loop simulation is successfully carried out taking the system through the acceleration phase to the desired cruise condition with a smooth switching between the acceleration and cruise segments.

The next stage of the research consisted of implementation of an appropriate FDA algorithm.

2.2 Analytical Redundancy: Survey

Numerous approaches to FDA in dynamical systems have been reported in the literature. Generally, FDA techniques are classified into three categories based on:

(i) hardware redundancy,
(ii) analytical redundancy, and
(iii) knowledge-based redundancy.

The present work, concentrates on analytical redundancy management. However, analytical redundancy has not been used as a substitute for hardware redundancy; instead, it has been used to provide a measure of intelligence to an FDA algorithm that uses triplex hardware redundancy for the P_4 sensors.

The field of model-based FDA[5-11] for linear systems is well-studied. Traditionally, analytical redundancy based methods[12] have been used to provide an indirect measurement of the variable of interest. However, when variables measured by different sensors are related by physical equations, the principle of analytical redundancy can also be used as a diagnostic tool to test whether the sensor outputs satisfy these known relationships. A set of sensors is healthy if an

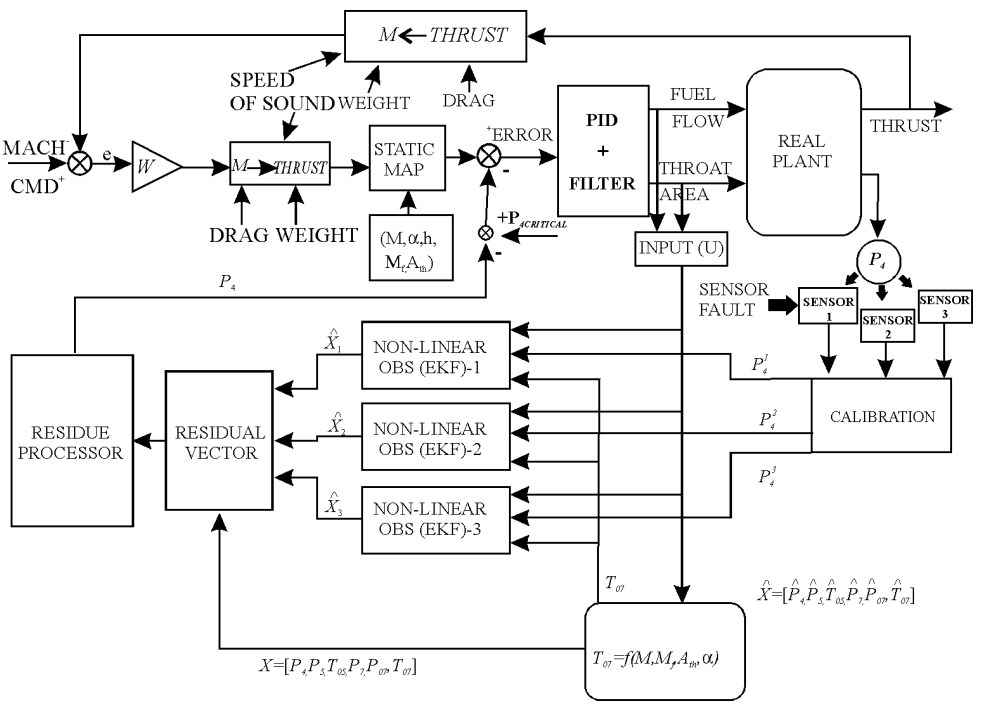


Figure 2. Model-based FDA algorithm for P_4 sensor.

equation which relates them is verified, and at least one sensor of the set has failed if the relationship is violated[13-15].

Analytical redundancy is an attractive option for high-performance aerospace vehicles that need to operate safely, reliably and with a high-level of mission success, even in the presence of faults. Weight and volume considerations often make it difficult to justify redundant sensors and actuators in aerospace applications. Analytical redundancy management holds the promise of reduction in price, weight and power consumption when implemented onboard. For air-breathing engines, multiple redundancies are harder to achieve due to lack of operating space, cost, engineering complexity and added maintenance requirements. This motivated to propose the use of an analytical redundancy algorithm as a higher-level functionality, on top of the hardware redundancy, to provide intelligence to the FDA algorithm and also improve its reliability.

2.3 Overview of FDA for Backpressure Sensor

First step in development of an observer-based FDA algorithm is a realistic representation of the physical system, which includes system dynamics, faults and all kinds of possible unknown inputs. Residual generation using observer-based method is presented here.

Figure 2 (lower part) shows model-based (analytical-redundancy-based) FDA for P_4 sensors. Triplex redundant static pressure sensor using Kulite pressure sensor (located at axial stations between 4 and 5 in intake duct/combustor entry, well before fuel injection station 5) can be seen. Calibration is necessary as pressure falls in duct due to boundary layer growth. Modelling of various possible sensor faults and noise is included, and calibration for the sensor location is done based on CFD data.

The intake backpressure, P_4 value is measured by a triplex system. These 3 estimates are termed as $\hat{P}_4^1, \hat{P}_4^2, \hat{P}_4^3$. These measured P_4 values may not be accurate owing to various sensor faults. In order to detect, isolate, and accommodate the faults (if any), analytical redundancy-based FDA algorithm is implemented. The P_4 measurement from each sensor is fed to separate Extended Kalman Filter (EKF) (forming an EKF bank). The EKF removes the disturbances from the measurements and output of EKF consists of the clean P_4 estimates along with P_5, T_{05}, P_7, T_{07}, P_{07}, $\dot{m}_5$ and $\dot{m}_7$. The EKF bank then generates the residual vector which is used by the residual processor for determination of accurate value of P_4 to be fed back to the system. Review of the EKF theory and design as implemented for ACS is presented in Section 4. To accommodate the faults, adaptive weightage-based fault accommodation algorithm is developed. The underlining assumption to this development is that no more than one sensor fails at any given time.

3. DESIGN AND IMPLEMENTATION OF EXTENDED KALMAN FILTER

For estimating correct system states and filtering the noise and disturbances present in the P_4 measurements, the EKF[16] is applied. To estimate the states, EKF needs two independent measurements. In this case, one measurement is obtained by the actual P_4 sensor and the second independent measurement is obtained from the modeled T_{07} (discussed in Section 4). These two measurements are fed to the Kalman Filter (KF). Thus, there are three KFs in all, one for each of the three backpressure sensors. The continuous time version of KF is used. Since the system is nonlinear, the EKF, which uses the linearized Jacobian matrices at specific operating conditions along the trajectory, is implemented. The Jacobian matrices are obtained by the small perturbation theory.

3.1 Discrete-time Kalman Filter

The KF model assumes the true state[17] at time k is evolved from the state at (k–1):

$$X_k = F_k X_{k-1} + B_k U_k + w_k \quad (1)$$

where, F_k is the state transition model which is applied to the previous state x_{k-1}; B_k is the control-input model which is applied to the control vector u_k; w_k is the unknown process noise which is assumed to be drawn from a zero mean multivariate normal distribution with process noise covariance Q_k.

$$w_k \approx N(0, Q_k) \quad (2)$$

w_k acts as a disturbance to the plant. It represents effects of unmodelled high-frequency plant dynamics that is modelled as zero-mean, white Gaussian noise. At time k, an observation (or measurement) z_k of the true state x_k is made according to

$$Z_k = H_k X_k + V_k \quad (3)$$

where, H_k is the nonlinear observation model which maps the true state space into the observed space and v_k is the observation noise which is assumed to be zero-mean Gaussian white noise with covariance R_k.

$$v_k \approx N(0, R_k) \quad (4)$$

The initial state, and the noise vectors at each step $\{x_0, w_1, \ldots, w_k, v_1 \ldots v_k\}$ are all assumed to be mutually independent.

In KF, only the estimated state from the previous time-step and the current measurement are needed to compute the estimate for the current state. No history of observations and/or estimates is required. The state of the filter is represented by the following two variables:

(i) $\hat{X}_{k|k}$, the estimate of the state at time k given observations up to and including time k;

(ii) $P_{k|k}$, the error covariance matrix (a measure of the estimated accuracy of the state estimate).

The discrete KF has two distinct phases: Predict and Update. The predict phase uses the state estimate from the previous time-step to produce an estimate of the state at the current time-step. In the update phase, measurement information at the current time-step is used to refine this

prediction to arrive at a new, (hopefully) more accurate state estimate, again for the current time-step.

3.2 State Estimation Equations for Discrete Kalman Filter

Optimal state estimate $\hat{x}$ and the state covariance matrix P are propagated from measurement time (k-1) to measurement time (k), based on the previous value, the system dynamics and the previous control input and error of actual system. This is done by numerical integration of the following equations:

Predicted state and estimate covariance:

$$\hat{X}_{k|k-1} = F_k \hat{X}_{k-1\,|k-1} + B_{k-1} u_{k-1}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1} \tag{5}$$

where, F_k is the state transition model which is applied to the previous state X_{k-1}

Update Step:

Innovation or measurement residual

$$\tilde{Y}_k = Z_k - H_k(\hat{X}_{k|k-1})$$

Innovation (or residual) covariance

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

Optimal Kalman gain

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

Updated state estimate

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \tilde{Y}_k$$

Updated estimate covariance

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{6}$$

Values of *F, H, Q* and *R* matrices are application dependent.

The Section 3.3 discusses the KF applied to the continuous time systems. The equations are obtained from the original KF equations for discrete KF.

3.3 Continuous Time Kalman Filter

The Kalman-Bucy filter[18, 19] is a continuous time version of the KF. It is based on the state space model:

$$\frac{d}{dt}x(t) = F(t)x(t) + w(t)$$
$$Z(t) = H(t)x(t) + v(t) \tag{7}$$

where, the covariances of the noise terms *w(t)* and *v(t)* are given by *Q(t)* and *R(t),* respectively. The filter consists of two differential equations, one for the state estimate and one for the covariance:

$$\frac{d}{dt}\hat{x}(t) = F(t)\hat{x}(t) + K(t)(z(t) - H(t)(\hat{x}(t)$$

$$\frac{d}{dt}P(t) = F(t)P(t) + P(t)F^T(t) + Q(t) - K(t)R(t)K^T(t) \tag{8}$$

where, the Kalman gain is given by

$$K(t) = P(t)H(t)R^{-1}(t) \tag{9}$$

Note that in this expression for $K(t)$, the covariance of the observation noise $R(t)$ represents at the same time the covariance of the prediction error (or innovation) $\tilde{y}(t) = z(t) - H(t)(\hat{x}(t)$; these covariances are equal only in the case of continuous time. The distinction between the prediction and update steps of discrete-time Kalman filtering does not exist in continuous time. The second differential equation, for the covariance, is an example of a Riccati equation.

3.4 Non-Linear System Problems: Extended Kalman Filter

The basic KF is limited to a linear assumption. However, most non-trivial systems are nonlinear. The nonlinearity can be associated either with the process model or with the observation model or with both. A KF that linearises about the current mean and covariance is referred to as an EKF. EKF linearises all nonlinear models so that the traditional linear Kalman filter can be applied. An EKF design has to be carried out to estimate states, while it should filter sensor noise and modelled measurement parameters.

3.4.1 *Extended Kalman Filter Formulation*

In the extended KF, the state transition and observation models need not be linear functions of the state but may instead be (differentiable) functions.

For discrete systems:

$$X_k = f(X_{k-1,} u_k) + w_k$$
$$Z_k = h(X_k) + v_k \tag{10}$$

For continuous systems:

$$\frac{d}{dt}x(t) = f(x(t), u(t)) + w(t)$$
$$z(t) = h(x(t)) + v(t) \tag{11}$$

The function, *f*, can be used to compute the predicted state from the previous estimate and similarly the function, *h*, can be used to compute the predicted measurement from the predicted state. However, *f* and *h* cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed. At each time-step the Jacobian is evaluated with current predicted states. These matrices can be used in the KF equations. This process essentially linearises the nonlinear function around the current estimate.

The ACS system under consideration is a nonlinear system, as can be seen from the dynamics equations presented in the following section. Hence, the EKF is implemented to estimate the states of the system.

3.4.2 Dynamics of the Air-breathing Combustion Systems

The dynamics of the system is represented by the relationships shown:

$$\dot{\hat{P}}_4 = \frac{1}{\tau_{54}}\left(\hat{P}_{4_{ss}} - \hat{P}_4\right)$$

$$\dot{\hat{m}}_5 = \frac{1}{\tau_{45}}\left(\hat{m}_{5_{ss}} - \hat{m}_5\right)$$

$$\dot{\hat{P}}_5 = \frac{1}{\tau_{75}}\left(\hat{P}_{5_{ss}} - \hat{P}_5\right)$$

$$\dot{\hat{T}}_{05} = \frac{1}{\tau_{75}}\left(\hat{T}_{5_{ss}} - \hat{T}_5\right)$$

$$\dot{\hat{m}}_7 = \frac{1}{\tau_{57}}\left(\hat{m}_{7_{ss}} - \hat{m}_7\right)$$

$$\hat{P}_7 = \frac{1}{B}\left(\hat{m}_7 - A_{th}\frac{\hat{P}_{07}\,\beta}{\sqrt{\hat{T}_{07}}}\right)$$

$$\dot{\hat{P}}_{07} = \frac{1}{\tau_{75}}\left(\hat{P}_{07_{ss}} - \hat{P}_{07}\right)$$

$$\dot{\hat{T}}_{07} = \frac{1}{\tau_{57}}\left(\hat{T}_{07_{ss}} - \hat{T}_{07}\right) \qquad (12)$$

where,

$$\beta = \frac{\sqrt{\gamma}}{\sqrt{R}}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \qquad (13)$$

The above equations for the ACS dynamics are nonlinear; hence the EKF is employed. The EKF design as applied to ACS is discussed in details.

3.4.3 Extended Kalman Filter as Applied to Air-Breathing Combustion Systems

The ACS system is modelled as a continuous time system, and hence, the continuous version of KF (Kalman-Bucy filter) discussed earlier is applied to the ACS model. The corresponding Eqns (11) are used with the following:

$X(t)$ = state vector = $[P_4\ P_5\ T_{05}\ P_7\ P_{07}\ T_{07}\ \dot{m}_5\ \dot{m}_7]$

$U(t)$ = control input = $[\dot{m}_f\ ,\ A_{th}]$

$Z(t)$ = measured output = $[P_4]$

Modelled output = $[P_{07},\ T_{07}]$

$v(t)$ = observation noise

Similar to the KF theory explained earlier, signal w(t) is an unknown process noise that acts as a disturbance to the plant. It represents effects of unmodelled high-frequency plant dynamics that is modelled as zero-mean, Gaussian white noise. Process noise covariance matrix Q(t) (referred to as Q) describing the random process is given as:

$$w(t) \sim N(0, Q) \qquad (14)$$

$v(t)$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $R(t)$. $v(t)$ should be selected such that all failure modes are accounted in it. Measurement noise covariance matrix $R(t)$ (referred to as R) is given by

$$v(t) \sim N(0, R) \qquad (15)$$

By applying the small disturbance theory, the dynamic relationships for ACS in Eqn (12) can be linearised. Here each variable is assumed to be composed of two parts: a constant component associated with the linear part and the perturbation associated with nonlinear model. The state transition and observation matrices are defined to be the following Jacobians:

$$F = \left.\frac{\partial f}{\partial x}\right|_{\hat{X}_{t-1|t-1}, u_t} = \frac{\partial f(x(t), u(t))}{\partial x}$$

$$H = \left.\frac{\partial h}{\partial x}\right|_{\hat{X}_{t|t-1}} \qquad (16)$$

For simplicity in the notation the time-step subscript with the Jacobians F, H are not used even though these are in fact different in every cycle.

$X_c(t)$ of $\Delta X(t)$ can be linearised about the central estimate as:

$$X(t+1) \cong f_0(\hat{X}_c(t), u_0(t)) + \left.\frac{\partial f_0(x, u_0)}{\partial x}\right|_{x=\hat{x}(t)} (x(t) - \hat{x}_c(t)) \qquad (17)$$

where $X_c(t)$ is central estimated value about which perturbation $\Delta X(t)$ is added. As in EKF, replacing and rewriting above Eqn (17), one has

$$X(t+1) \cong f_0(\hat{X}_c(t), u_0(t)) + F(\hat{x}_c(t))(x(t) - \hat{x}_c(t)) \qquad (18)$$

3.4.4 Linearising Air-Breathing Combustion Systems dynamics for Extended Kalman Filter Implementation

The nonlinear plant dynamics is complex and it consumes significant resources during simulations. Since, the FDA algorithm is to be implemented in real-time, it should be based on a simpler linear model that can be processed much faster. Hence, a family of linearised model of the nonlinear model at various operating conditions was developed, which is suitably fast and provides close results to the nonlinear plant when properly scheduled with a suitable parameter (Mach number in climb, and angle of attack in cruise). This linear model of the plant estimates the states using the state space model as:

$$dx/dt = Ax + Bu$$

$$y = Cx + Du \qquad (19)$$

Since the first-order partial derivatives (Jacobians) are used instead of the first-order differential equations, the above model can be rewritten as

$$dx/dt = Amat * x(t) + Bmat * u(t) \qquad (20)$$

$$y = Cmat * x(t) + Dmat * u(t)$$

where, *Amat, Bmat, Cmat, Dmat* are the Jacobian matrices corresponding to the given operating conditions.

The exact Jacobians at a number of specific operating conditions were obtained and for intermediate operating conditions the corresponding Jacobians by linear interpolation

were obtained. Let us assume that our process has a state vector $x = [P_4, P_5, T_{05}, P_7, P_{07}, T_{07}, \dot{m}_5, \dot{m}_7]$, but that the process is governed by the nonlinear stochastic Eqn:

$$\frac{d}{dt}x(t) = f(x(t),u(t)) + w(t)$$
$$y(t) = h(x(t)) + v(t) \quad (21)$$

where, the random variables $w(t)$ and $v(t)$ again represent the process and measurement noise. Please note that $w(t)$ and $v(t)$ are inherently present in the system. Using the linear model of plant for state estimation, the above Eqns can be modified as:

$$\frac{d}{dt}x(t) = f(x(t),u(t)) + w(t) = [Amat.x(t) + Bmat.u(t)] + w(t)$$
$$y(t) = h(x(t)) + v(t) = Cmat.x(t) + Dmat.u(t) + v(t) \quad (22)$$

To estimate the process with nonlinear difference and measurement relationships, new governing equations that linearize the estimate are written:

The state transition matrix F is given by

$$f = \frac{\partial F}{\partial x}\bigg|_{\hat{X}_{t-1|t-1},u_t} = \frac{\partial F(x(t),u(t))}{\partial x} \quad (23)$$

$$f = \frac{\partial F}{\partial x}\bigg|_{\hat{X}_{t-1|t-1},u_t} = \frac{\partial F(x(t),u(t))}{\partial x} = \frac{\partial (Amat.x(t) + Bmat.u(t) + w(t))}{\partial x} \quad (24)$$

The observation matrix H is given by:

$$H = Hmat = \frac{\partial H}{\partial x}\bigg|_{\hat{X}_{t|t-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (25)$$

The determination or selection of the Q and R matrices is dependent on the application requirements, operating conditions, and trustworthiness of measurements.

4. RESIDUAL GENERATION AND FAULT ACCOMMODATION

4.1 T_{07} Modelling

The T_{07} (total temperature at the end of combustor exit) is used as a second independent measurement for the EKF implementation. While P_4 is obtained from actual sensor measurement, T_{07} is read from a look-up table using the fuel-to-air (FAR), ratio as a parameter for different flight conditions.

The nonlinear model of ACS is simulated for different flight conditions with varying FAR, and the T_{07} and FAR values for each operating condition are tabulated. Figure 3 shows the 2-D table in graphical form.

Two different sets of T_{07} look-up tables were generated. The first was looked up during the acceleration phase, and makes use of the Mach number and FAR as the two look-up parameters. During cruise phase the second set of tables, which makes use of angle of attack and FAR as the two lookup-parameters was used. During the switching from acceleration to cruise, a switching function was used which blends the acceleration and cruise values linearly.

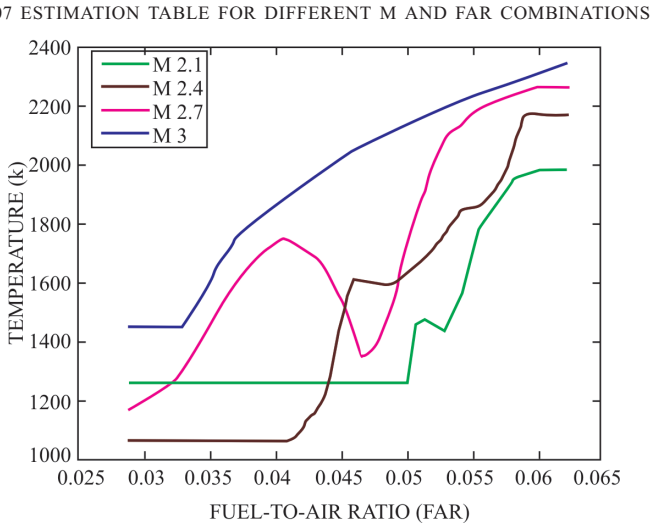


Figure 3. T_{07} plots for different M and fuel-to-air ratio combinations.

4.2 Residual Processor

Using the estimated values of the modelled T_{07} obtained from the FAR-to-T_{07} static map and P_{07} from each EKF, along with the known bias setting of the nozzle throat area, it is possible to get an estimate of the choking mass flow rate at the nozzle throat. Thus, three estimated values of nozzle mass flow rate are available from the bank of three EKFs. It is also possible to independently estimate the nozzle mass flow rate from measurements of the free stream quantities, such as static pressure, total pressure, and static temperature, by the air data system (ADS). Finding the difference between each EKF-estimated nozzle mass flow rate and the ADS-estimated mass flow rate gives three values of error, also called residue or residual. In the absence of fault in a sensor, the residue generated by it is expected to be very small, ideally zero, whereas in the presence of a fault, the residue is expected to deviate significantly from zero. Thus, each residue will provide a

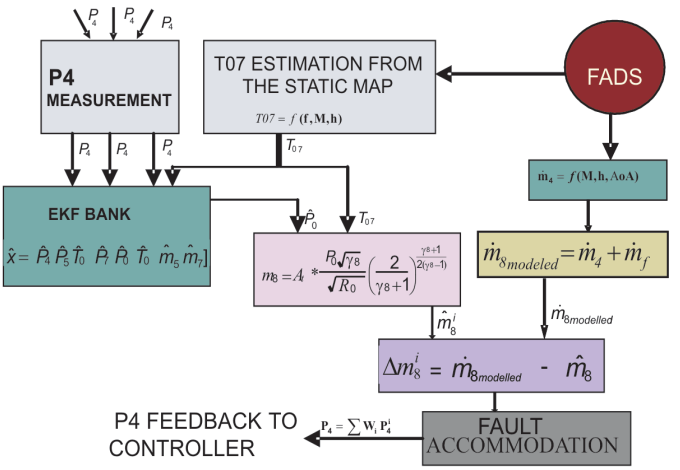


Figure 4. Residual processor for ACS.

good indication whether the sensor in that particular EKF channel is faulty or not, thus providing fault detection capability. Figure 4 shows the residual processor as part of the FDA algorithm.

The present choice of residue is effective because changes in backpressure (such as due to a fault) correlate well with changes in total pressure at combustor exit, and hence, with the estimated nozzle mass flow rate.

The inverse of the residue is also a measure of confidence in that particular measurement; hence, can be used for fault accommodation. The strategy explored here is to use the inverse of each residual to compute a normalised weighting factor that is dynamic, i.e., changes with time. This can be thought of as an adaptive weighting algorithm. The final P_4 value is then computed by applying the weighting factor to each of the three values, and is used as the feedback signal to the plant.

4.3 Fault Accommodation

Present FDA algorithm does not require the faulty sensor to be identified. It also does not demand the faulty sensor to be eliminated. The fault accommodation part of the FDA employs an adaptive weightage assignment algorithm without declaring and eliminating a faulty sensor, in each time-step, an adaptive weightage is assigned to each P_4 sensor and the weighted output is obtained. An important advantage of using this approach is that weighted value of P_4 (to be fed back), computed using the three P_4 sensors, is as accurate as possible for each iteration. For any iteration, if the sensor has not entirely failed but measures a wrong P_4 value, the sensor is not discarded forever; instead, the faulty measurement for that iteration is suppressed by allocating less weightage to it. A variable weighting factor is assigned to each measurement. In case all the three sensors are healthy, the mean value of the P_4 measurements is fed back to the plant.

The weighting factor W_i is computed as follows:

(i) The deviation of each estimated $\dot{m}_8$ (from the EKF channel) from the $\dot{m}_{8modelled}$ is computed. for i = 1 to 3 (for 3 EKF channels):
$\Delta\dot{m}_8^i$ = abs ($\dot{m}_{8modelled_i} - \hat{\dot{m}}_8^i$).

(ii) The ratio of each $\Delta\dot{m}_8^i$ to the $\dot{m}_{8modelled}$ is computed. This ratio decides how far or close estimated $\dot{m}_8$ is from the $\dot{m}_{8modelled}$. for i = 1 to 3 (for 3 EKF channels):
ratio (i) = $\Delta\dot{m}_8^i / \dot{m}_{8modelled}$

(iii) Minimum of the three ratios (*ratio_min*) is determined.

(iv) The minimum ratio value is then subtracted from the individual ratio for each channel. For i = 1 to 3 (for 3 EKF channels): ratio_new(i) = ratio(i)-ratio_min;

(v) Weightage for each channel is computed as: For i = 1 to 3 (for 3 EKF channels): $W(i) = 10^{-10(\text{ratio\_new(i)})}$]

This makes sure that the channel with lowest ratio (zero) is given the highest weightage (one), and the others are assigned exponentially reducing weights based on their ratios. The exponential curve demonstrating the above relationship is shown in Fig. 5.

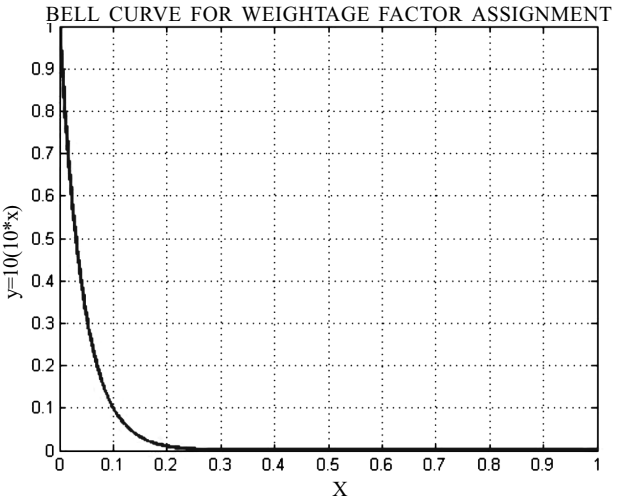


Figure 5. Exponential function for weightage assignment.

The exponential weightage formula $W(i) = 10^{-10(\text{ratio\_new(i)})}$ is obtained by trial and error method by carrying out a number of simulations.

These weights are then normalised to find the final weights for each $\dot{m}_8^i$ (corresponding to each EKF channel):

$$W_{i\_norm} = W_i / \sum_{i=1}^{3} W_i \tag{26}$$

This weighting reflects the faults in any sensor measurement and also accommodates it. Once the weighting factor is calculated, the P_4 mean value is calculated as:

$$P_4 = \sum W_{i\_norm} P_4^i \tag{27}$$

This P_4 value is then fed back to the plant. The weightage factor computation for 3 hypothetical cases is shown in Table 1.

The W_i is the percentage of weightage of each measurement towards the P_4 value to be fed back. Above three cases demonstrate that the conceptual development for computation of W_i is suitable for range of $\dot{m}_8$ values and for extreme cases of deviations.

There was a need to develop the FDA in the case of only one sensor failure, but it can be observed that the designed FDA algorithm allows for failure and accommodation

Table 1. Fault accommodation scheme demonstration

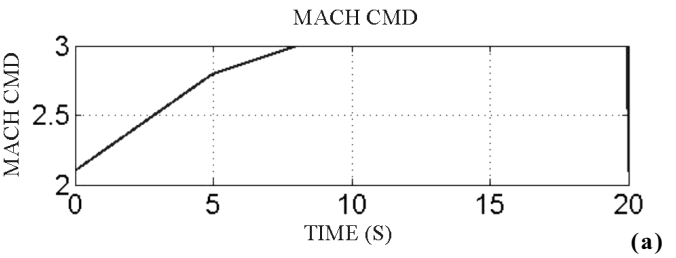
Mass flow rate at station 8 (kg/s)	Case 1	*W*	Case 2	*W*	Case 3	*W*
$\dot{m}_{8modelled}$	16.5	-	10.15	-	10.15	-
$\dot{m}_8^1$	6.2	4.77e^{-6}	16.8	2.087e^{-7}	18.5	6.647e^{-9}
$\dot{m}_8^2$	14.98	~1	9.80	0.3361	5	9.449e^{-6}
$\dot{m}_8^3$	7.2	1.927e^{-5}	10.1	0.6639	10.1	~1

of two sensors as shown in Table 1. If all three sensors fail and measure incorrect value of P_4, then the closest (although wrong) measurement (and corresponding EKF channel) will be assigned highest weightage and that will contribute the highest to the feedback P_4 value. But it is important to note that this value will not be a required P_4 value and the system may not work satisfactorily.

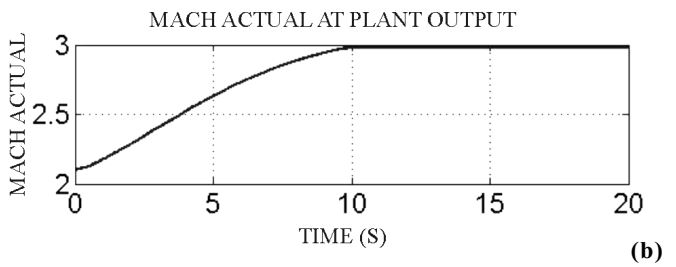
5. RESULTS AND EVALUATION OF FDA

To evaluate and validate the FDA algorithm, a number of simulations are carried out. This section presents the results from the FDA algorithm in the presence of different faults. It is observed that the system operates inefficiently with fault. In fact, some faults may lead to complete failure of the system. For testing and simulation purposes, the fault is introduced once the system reaches steady state (at around 0.1 s). The fault introduction details are as explained below. Figures 6-9 present the results obtained by running the closed-loop simulations in the presence of various fault conditions.

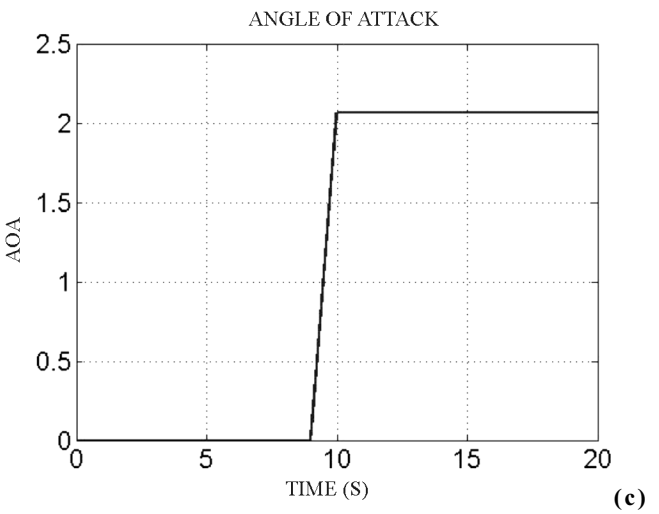
(a) Fault-free Case



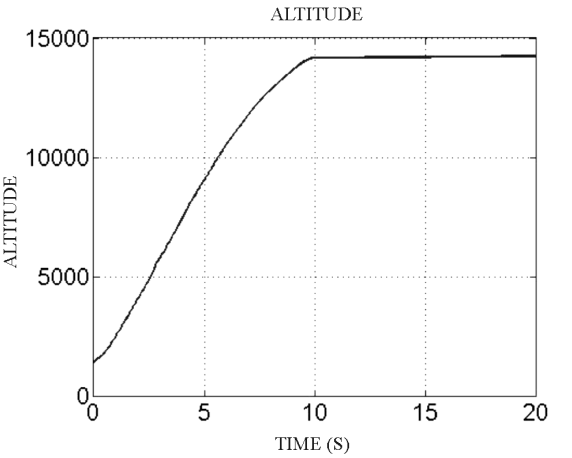
(a)



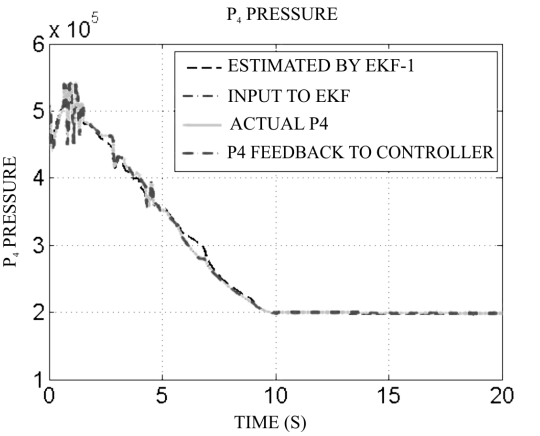
(b)



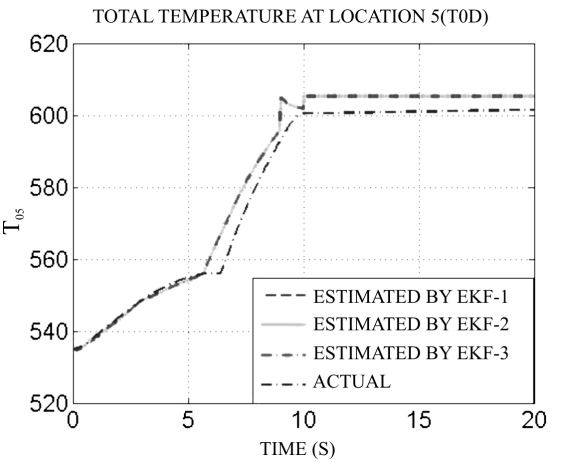
(c)



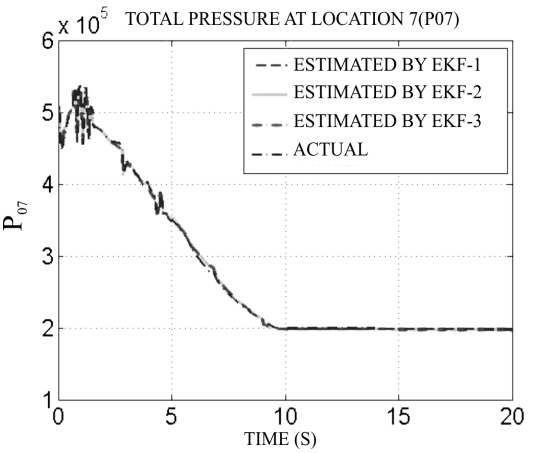
(d)



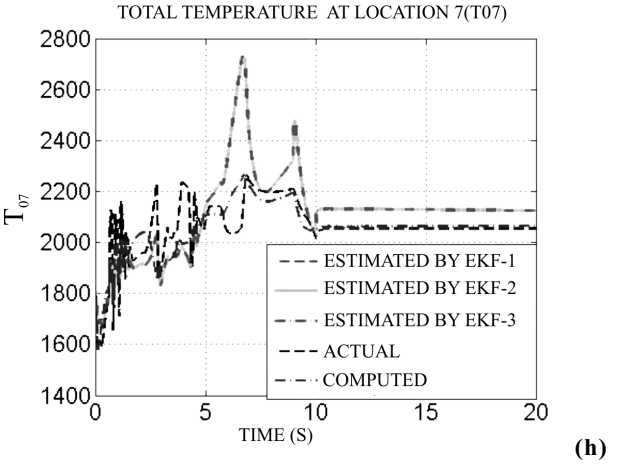
(e)



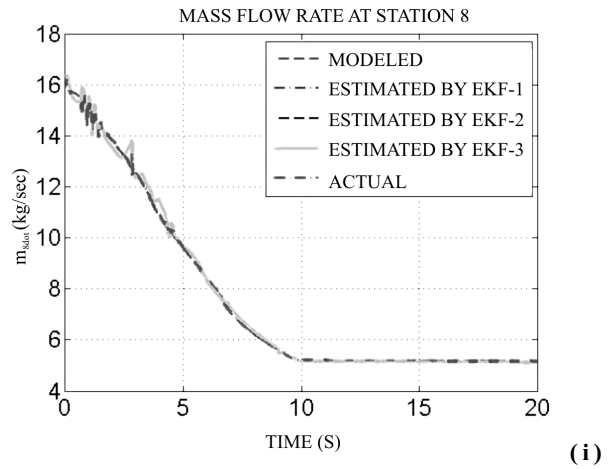
(f)



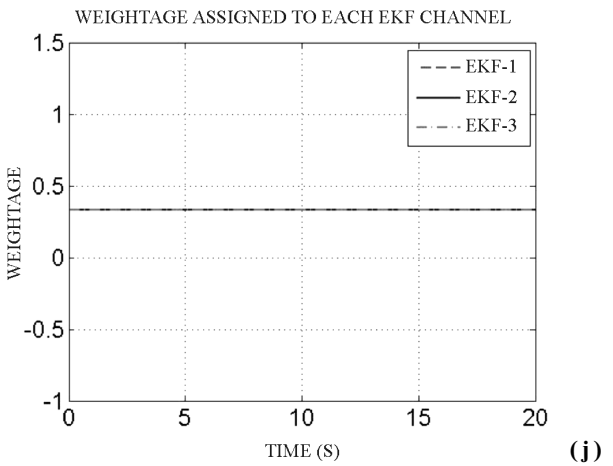
(g)



(h)



(i)



(j)

Figure 6. (a-j) Closed loop simulation for fast response mach profile in the absence of fault. (All EKF have exactly same P_4 plots in no fault conditions)

(b) Ramp Bias:

In this simulation, the output is made to slowly deviate from its nominal value. The simulation is fault-free for the first 0.1 sec. At t=0.1 sec, bias value is added to the sensed value of voltage by a ramp change. This erroneous sensed value is fed back to the controller, which slowly decreases fuel flow rate until the lower limit of $P_{4margin}$ is attained.

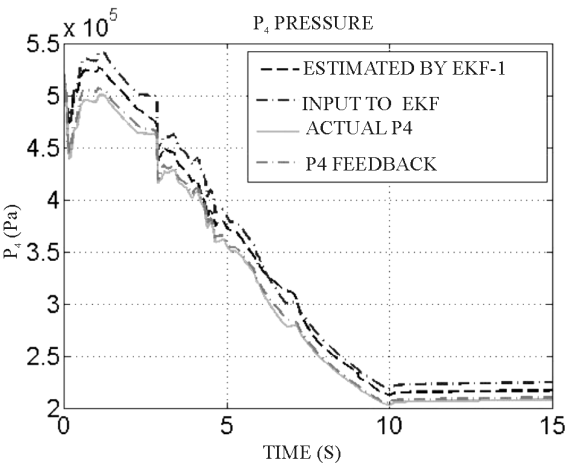
***(c) Pressure (P_4) Stuck at a Non-zero Constant Value*:**

The simulation is fault-free for the first 0.1 s. At t=0.1s, the sensed value at the sensor port is held constant at 1.5 bar by a step change. This erroneous sensed value is fed back to the controller creating a $P_{4margin}$ error. The controller therefore injects less fuel in trying to raise $P_{4margin}$ to the commanded value but ends up raising the actual $P_{4margin}$ to a high value. But, since sensor output is stuck, the $P_{4margin}$ error continues to persist.

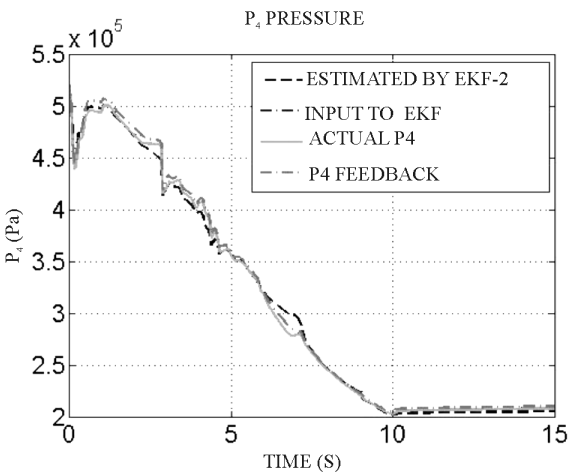
***(d) Mis-alignment*:**

Due to misalignment of P_4 sensor, static pressure measured by pressure sensor will be larger and hence during simulation at time = 0.1 s, the sensed value at the sensor port is deliberately increased by 5 per cent higher than the true pressure value by impulse change. This erroneous higher pressure is further fed back into the controller. The controller tries to maintain that erroneous P_4 value as its commanded value by decreasing fuel flow rate, and this in turn lowers the thrust and Mach number.

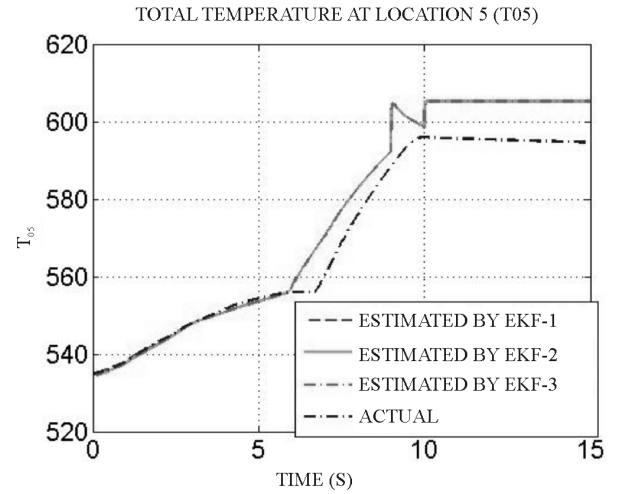
Fault 1: Misalignment Fault



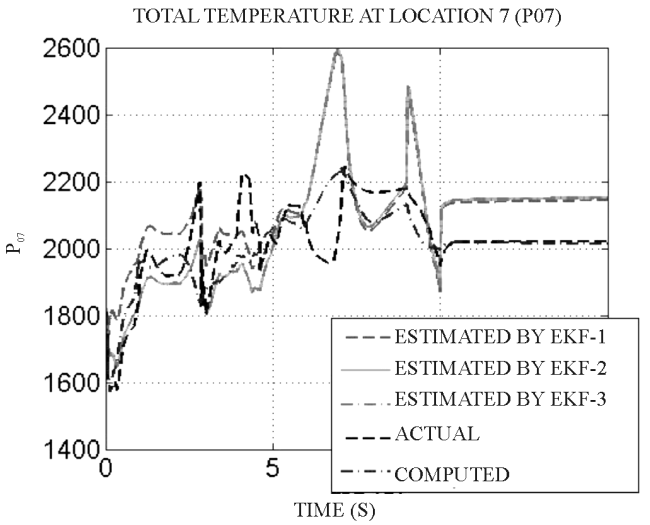
(a)



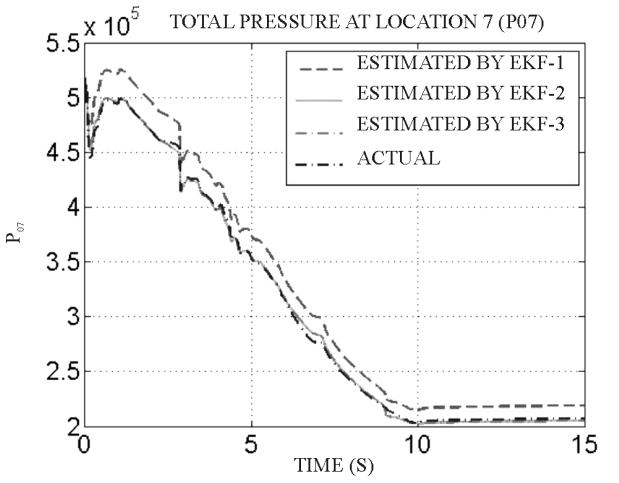
(b)



(c)



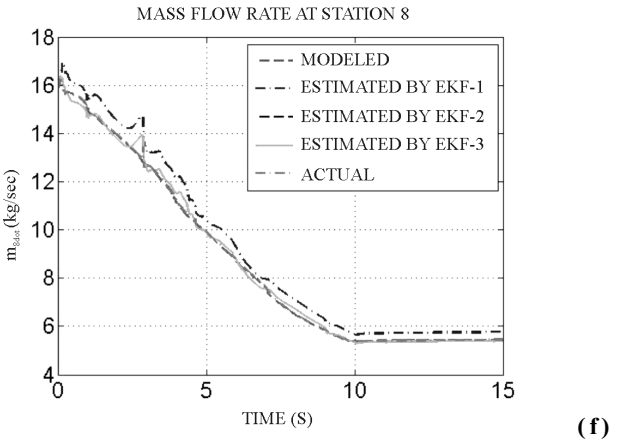
(d)



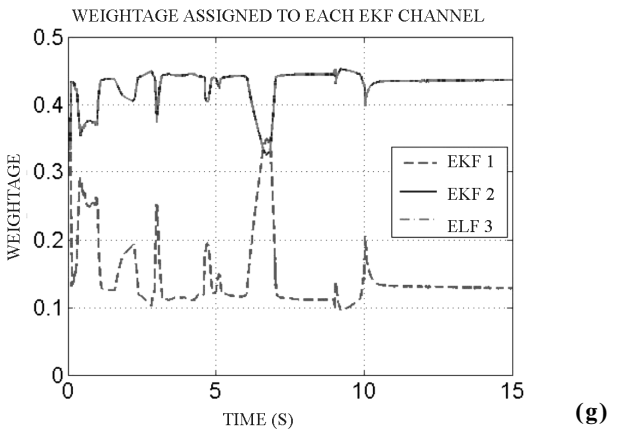
(e)

(e) Temperature Compensating System Failure:

In this fault, response to the error introduced due to temperature bias shift and temperature sensitivity shift in case of failing of the temperature compensating system is considered. Error of about twice/thrice the magnitude of a function of T_4 is introduced in P_4 measurement. Fault is introduced at 0.1s. As T_4 temperature is higher than the temperature at which sensors are calibrated, output voltage



(f)

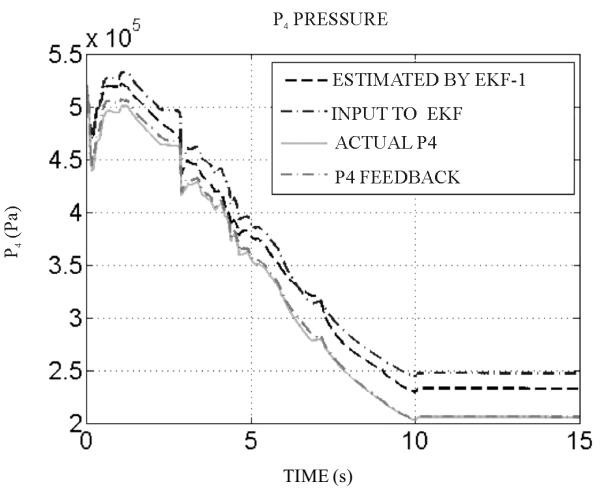


(g)

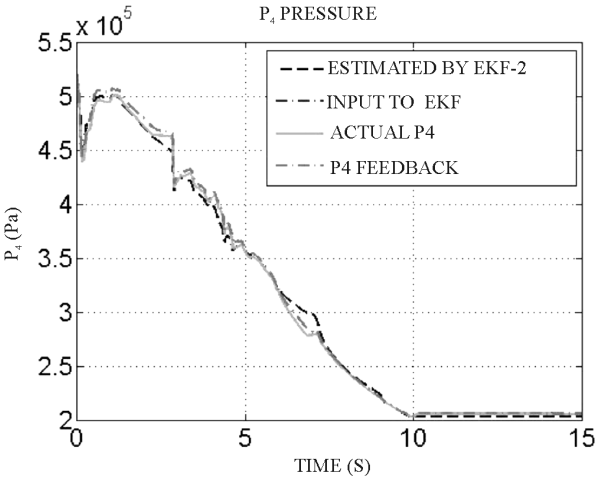
Figure 7. (a-g) Closed-loop simulation for fast response mach profile in the presence of misalignment fault.

is sensed at a higher value than the true value. This implies that the pressure is also sensed to be of a higher value. This erroneous higher pressure is further fed back into the controller. The controller tries to maintain that erroneous P_4 value as its commanded value by decreasing fuel flow rate, and this in turn lowers the thrust and Mach number. Because of the decrease in Thrust, $P_{4margin}$ command gets decreased up to its lowest limit, and commanded thrust goes on increasing.

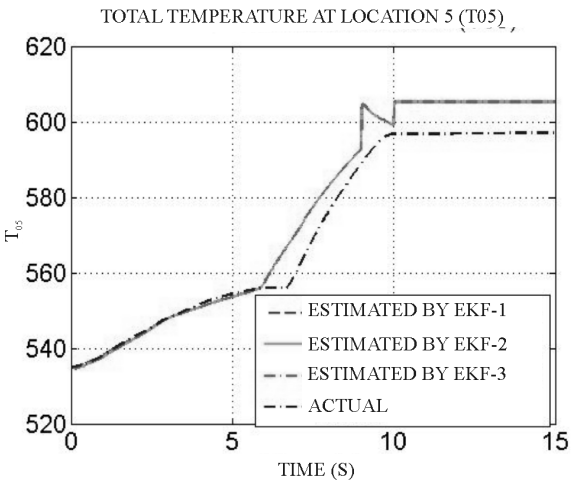
Fault 2: Temperature Compensating System Failure



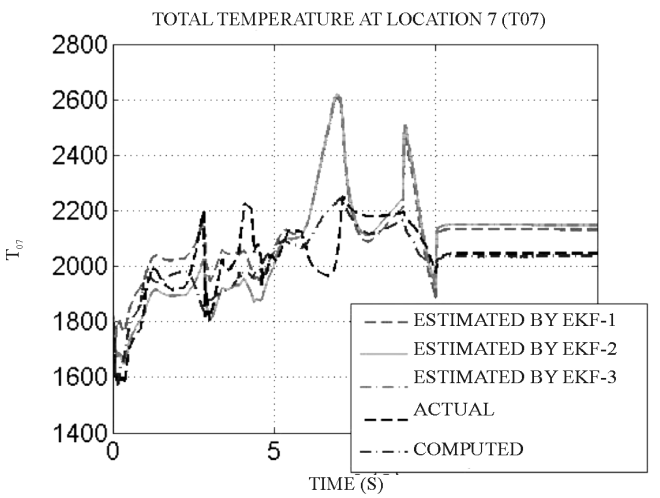
(a)



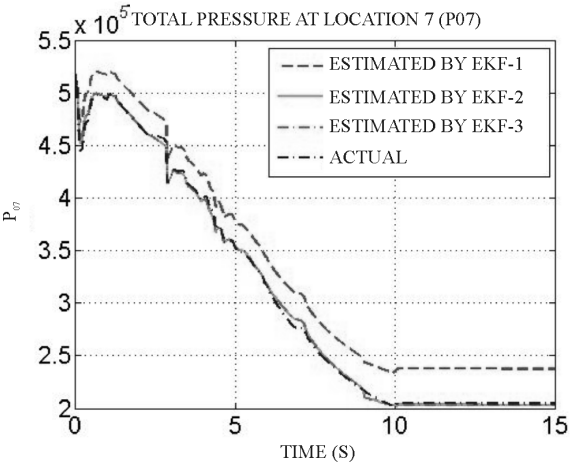
(b)



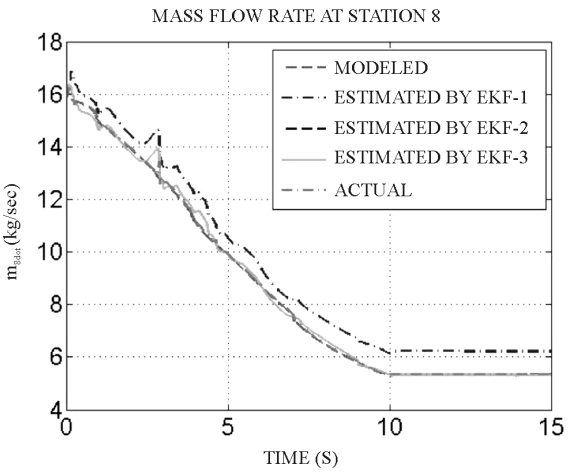
(c)



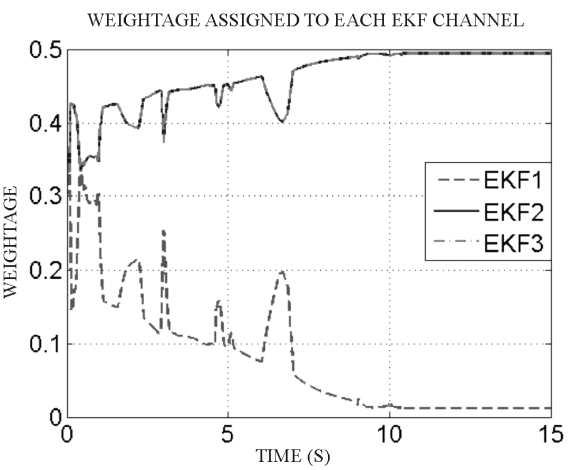
(d)



(e)



(f)



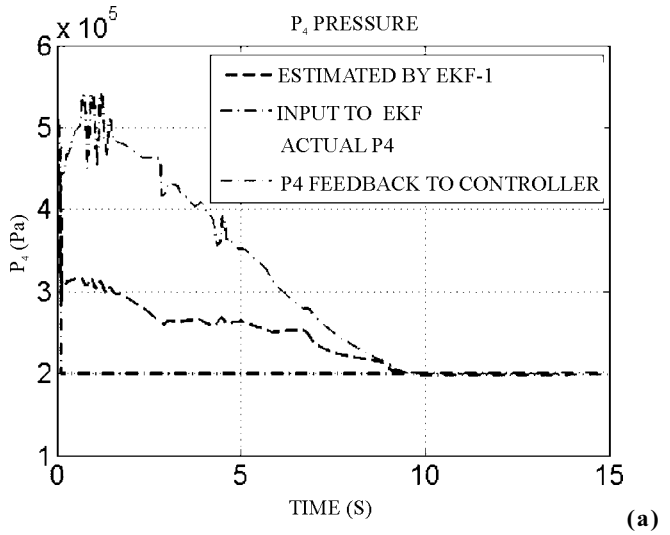
(g)

Figure 8. (a-g) Closed-loop simulation for fast response Mach profile in the presence of temperature compensating system failure.

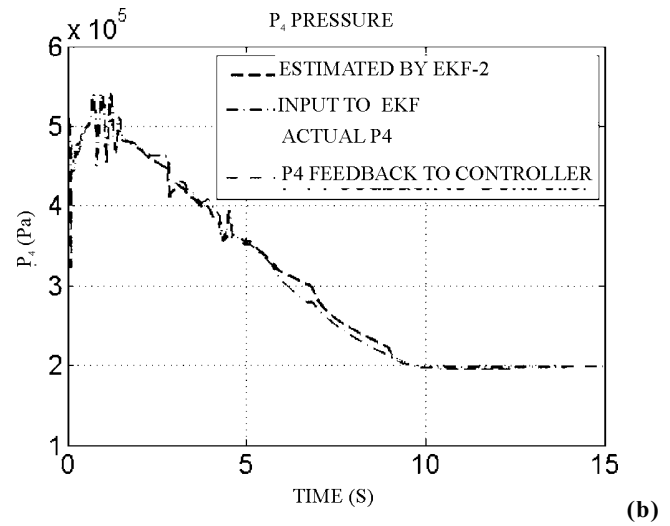
(f) Noise:

The simulation is fault-free for the first 0.1 sec. At t=0.1 s, white noise of power spectral density (PSD) 1×10^{-6} value is added in output voltage and the latter is seen to fluctuate about its true value. It is observed that sometimes $P_{4\text{margin}}$ hits its lower limit.

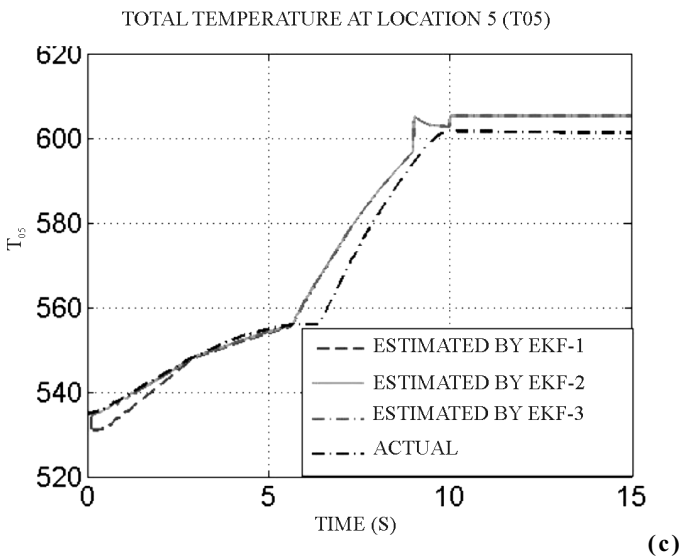
Fault 3: Pressure sensor stuck at a constant value



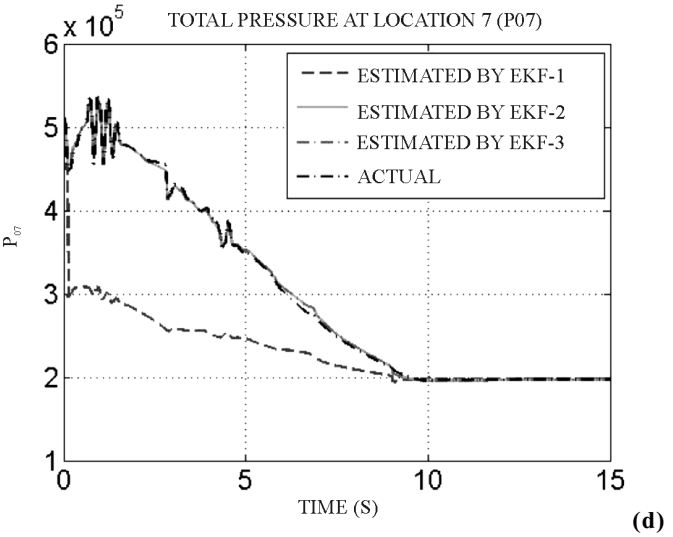
(a)



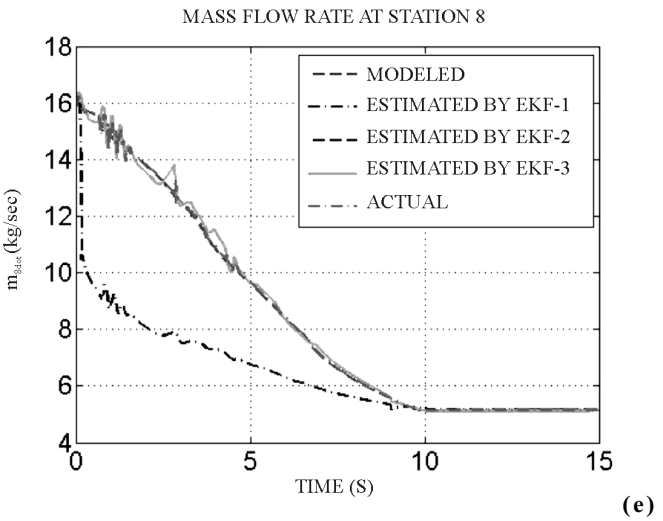
(b)



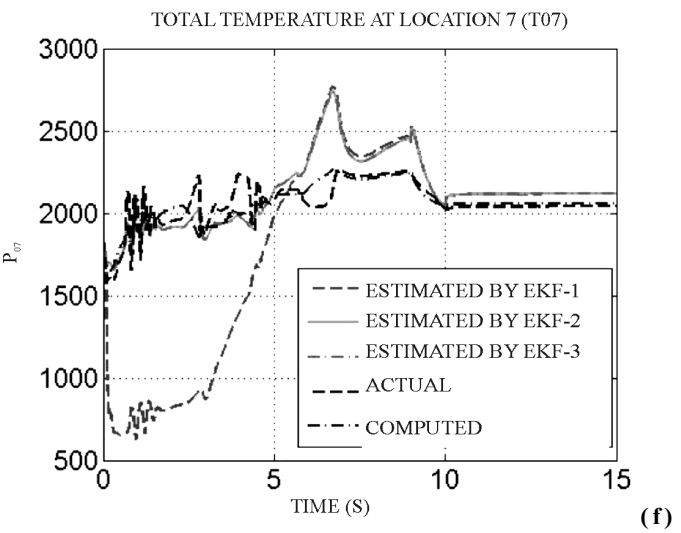
(c)



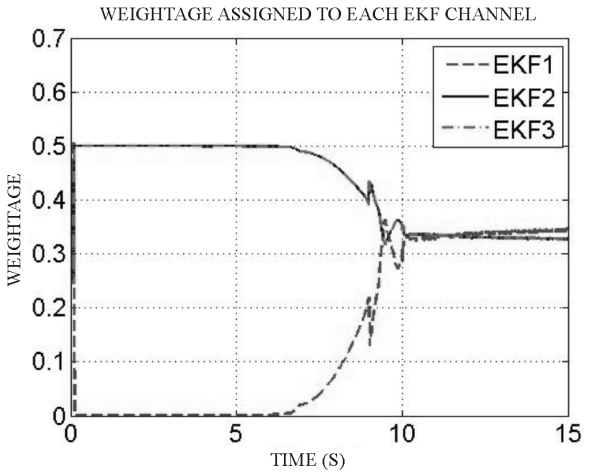
(d)



(e)



(f)



(g)

Figure 9. (a-g) Closed-loop simulation for fast response mach profile in the presence of pressure sensor stuck at a constant value.

5.1 Evaluation of FDA Algorithm from Results

A novel concept of using the mass flow rate at station 8 has been developed, as two independent estimates of a variable are not directly available for use in analytical redundancy-based FDA algorithm. Use of $\dot{m}_8$ proves to be a promising approach as shown in the closed-loop response plots for different conditions (Figs 6-9). Following conclusions can be drawn from the plots for different operating conditions:

1. Bank of EKF is implemented as part of FDA scheme. EKF design for all operating conditions works well. It facilitates the separation of disturbances and also generates the estimated state vector. Capability of EKF to estimate the faulty values is observed through number of simulations for known sensor faults. Results show that the estimation of states by EKF represents the faults faithfully. From plots for state parameters P_4, P_{07}, T_{05}, T_{07} and $\dot{m}_8$ it can be seen that the estimated parameters from EKF deviate from the actual values after introduction of fault.
2. In the plots for P_4, it can be seen that for fault-free condition, all three P_4 values are the same and these are also very close to the actual P_4 value as well as the P_4 value fed back to the controller after the application of FDA.
3. From plots for $\dot{m}_8$ at all operating conditions, it can be seen that the estimated $\dot{m}_8$ responds well to the fault. Thus, a clear signature of the fault can be noted for each fault type and operating condition. The $\dot{m}_{8modelled}$ ($\dot{m}_4 + \dot{m}_f$) follows the $\dot{m}_{8_{actual}}$. After the fault is introduced at 0.1s, the estimated $\dot{m}_8$ ($\hat{\dot{m}}_8$) slowly starts deviating from the $\dot{m}_{8modelled}$/ $\dot{m}_{8_{actual}}$. Hence choice of $\dot{m}_8$ as a key parameter for fault detection proves to be a promising concept.
4. In plots for T_{07}, three values of T_{07} viz., actual T_{07}, T_{07} fed to EKF (modelled T_{07} value), T_{07} estimated are plotted. As can be seen the modelled T_{07} is very close to the actual T_{07} value. Thus the method used for determining T_{07} (Section 4.1) is fairly accurate and acceptable.
5. It can be seen that the P_{07} follows the P_4 closely.
6. Result of the adaptive weightage assignment scheme is also shown. The plots demonstrate that the scheme works well. Very small weightage is assigned to thefaulty P_4 value and hence the P_4 fed back is very close to the actual P_4 value.

It is important to note that the present algorithm does not require the type of fault to be detected to accommodate it. It is also not necessary to set threshold and declare a sensor faulty. These are significant advantages over previous purely hardware-redundant FDA algorithms.

The simulations show that the FDA algorithm is able to successfully provide a good value of backpressure to the controller under a variety of fault cases in the sensors. The complete closed-loop simulation, with the air-breathing combustion system and the controller, using the backpressure from this FDA algorithm has demonstrated good results for both fault-free and different fault cases (Figs 6-9).

5.2 Real-time Implementation of FDA

The FDA algorithm developed as part of this is required to run in real-time once it is deployed on the actual system, i.e., it should complete the calculations for a time-step in time smaller than that time-step, and wait for the inputs (sensor data) to be made available to begin processing for the next time-step. It needs to be demonstrated that the algorithm is sufficiently simple and low in complexity so that it can be processed in real-time on processors available today.

To maintain numerical stability, the time-step should be smaller than the time constant used to model the system. From combustor look-up tables, it is found that the lowest time constant used is around 0.002 s, putting a lower limit on the update frequency at 500 Hz. For real-time execution at 500 Hz, the processor is required to complete one time-step of the entire FDA algorithm within 0.002 s.

To test real-time performance, a Simulink model with a linear plant, EKF bank and fault detection/ accommodation block is prepared, as shown in Fig. 10.

One needs to check the real-time performance of the FDA algorithm and the calibration block. So the linearised plant model is used to provide the P_4 input to the FDA. The Simulink is configured to run using fixed time-stepping of 0.002 s and ODE1 (first-order Euler method) solver, on a computer with 1.7 GHz processor, 512 MB RAM, Windows XP, Simulink version 6.4. It was observed that the simulation of 20 s of flight takes around 1-2 s. This clearly indicates that the processor power available on this system is adequate to execute this algorithm in real-time. To further demonstrate this, the real-time blockset[20] was used to slow down the simulation to real-time and to evaluate the amount of time CPU waited before it could begin the next time-step. It is observed that the processing for each time-step was completed on an average well within 2 μs and the CPU has to wait

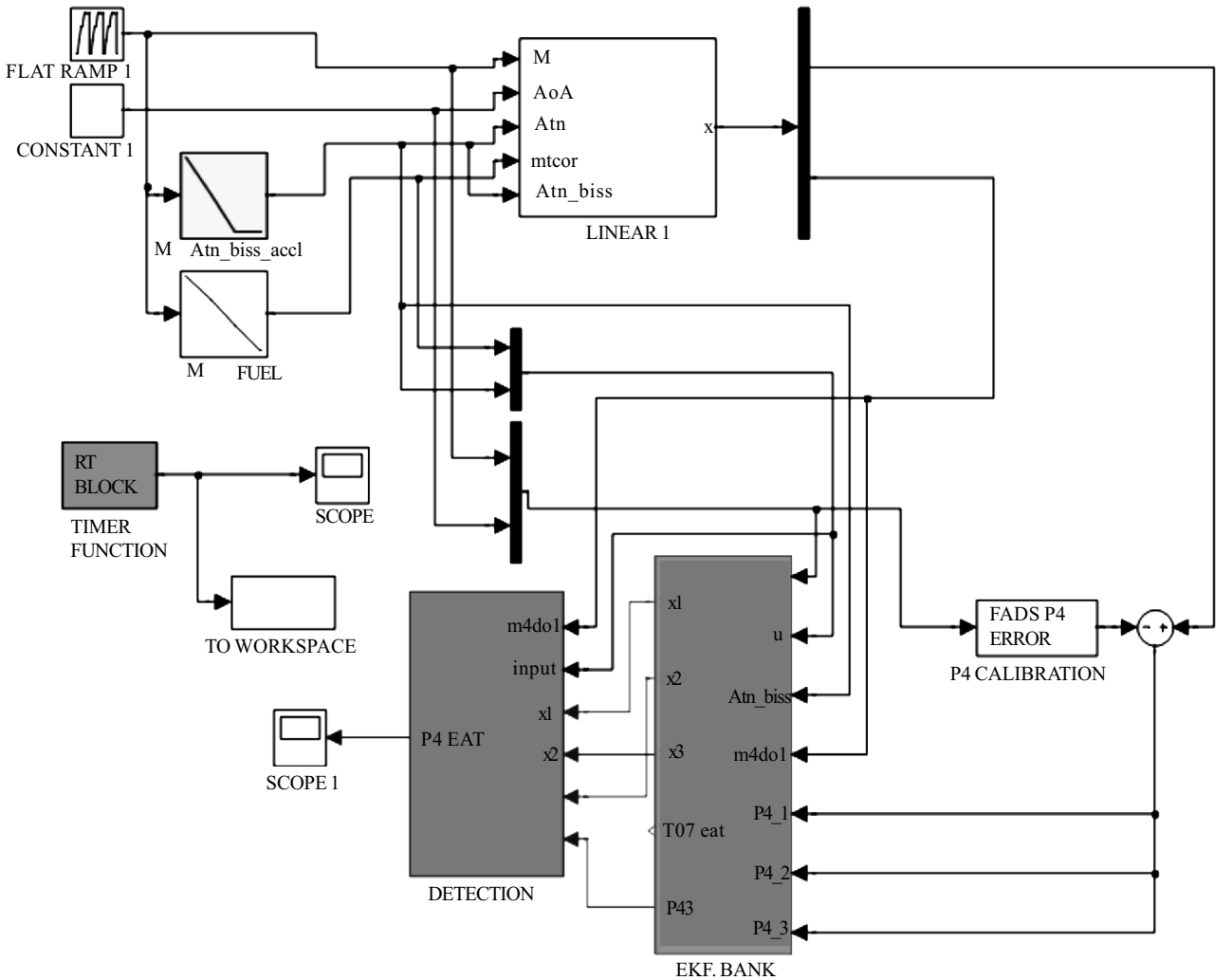


Figure 10. Simulink model used for evaluating real-time performance of FDA.

for significant time before it could start processing for the next time-step.

The key step in this process has been the use of a linear plant in place of the nonlinear plant for the EKF which significantly reduced the computation complexity. The system used for this demonstration uses a non-real-time operating system (Windows) and has several tasks running in parallel with the simulation. The performance of the system would significantly improve when a real-time operating system is deployed and the processor is used for execution of the FDA algorithm alone. Since the model could run in real-time on a non-optimised system, this is adequate to prove that the code is easily implementable in real-time on an optimised system.

6. CONCLUSIONS

This work has focused on the development and testing of a novel FDA algorithm for an air-breathing combustion system which uses an innovative analytical redundancy-based algorithm to provide intelligence to a triplex redundant P_4 sensor measurement hardware. In the study the following issues were addressed, and were successfully resolved and evaluated:

1. EKF bank was designed for state estimation and disturbance removal from the P_4 sensor measurement. The EKF bank also generated the residual vector, which is fed to the residual processor for fault detection and accommodation.
2. The residual processing algorithm consisting of the following has been thoroughly tested:
   - T_{07} computation from the known parameters using a look-up table
   - P_{07} estimation from the measured P_4
   - Estimation of $\dot{m}_8^i$ from three $\hat{P}_{07}$ and T_{07} values Comparison of $\dot{m}_8^i$ with $\dot{m}_{8_{modeled}}^i$ for detection of fault
   - Weightage factor calculation for individual $\dot{m}_8^i$ value based on the deviations from the $\dot{m}_{8_{modeled}}^i$.
   - Demonstration of error plots for fault detection and sensor removal

Main contribution of this work is the use of analytical

redundancy based FDA algorithm for P_4 sensor. A novel concept of using the mass flow rate at station 8 is developed, as two independent estimates of a variable are not directly available for use in analytical redundancy based FDA algorithm. Use of $\dot{m}_8$ proves to be a promising approach as shown in the closed-loop response plots for different conditions in Figs. 6-9. The real-time implementation results show that the present design of the FDA algorithm is suitable to run on an embedded hardware.

REFERENCES

1. Bharani Chandra, P.; Gupta, N.K.; Ananthkrishnan, N.; Renganathan, V.S.; Park, I.S. & Yoon, H.G. Modeling, dynamic simulation, and controller design for an air- breathing combustion system**.** *In* 47th AIAA Aerospace Sciences Meeting, Orlando, Florida, Jan 2009. AIAA Paper 2009-708,
2. Rausch, R.; Viassolo, D.E.; Kumar, A.; Goebel, K.; Eklund, N.; Brunell, B. & Bonanni, P. Towards in-flight detection and accommodation of faults in aircraft engines. AIAA 1st Intelligent Systems Technical Conference, Chicago, Illinois, Sep. 2004. AIAA-2004-6463.
3. O'Brian, T.F, Starkey R.P. & Lewis, M.J. Quasi-one-dimensional high-speed engine model with finite-rate chemistry. *J. Propulsion Power*, 2001, **17**(6), 1366-374.
4. Gupta, N.K.; Gupta, B.K.; Ananthkrishnan, N.; Shevare, G.R.; Park, I.S. & Yoon, H.G. Integrated modeling and simulation of an airbreathing combustion system dynamics. AIAA Modeling and Simulation Technologies, Conference and Exhibit, Hilton Head, South Carolina, August 2007. AIAA Paper 2007-6374.
5. Chen, J. & Patton, R.J. Robust model-based fault diagnosis for dynamic systems. Kluwer Academic Publishers, 1999.
6. Gertler, J.J. Fault detection and diagnosis in engineering systems, Ed.1. Marcel Dekker, 1998.
7. Isermann, R. & Ballé, P. Trends in the application of model-based fault detection and diagnosis of technical processes. *Contr. Eng. Practice*, 1997, **5**, 709-19.
8. Patton, R.J. Fault tolerant control: The 1997 situation. *In* Proceedings of IFAC Safe Process. Hull, UK, 1997, 1033-055.
9. .Frank, P.M. Analytical and qualitative modelbased fault diagnosis - A survey and some new results. *Euro. J. Contr.*, 1996, **2**(1), 6-28.
10. Massoumnia, M.A.; Verghese, G.C. & Willsky, A.S. Failure detection and identification. *IEEE Trans. Autom. Contr.*, 1989, **34**(3), 316-21.
11. Willsky, A.S. A survey of design methods for failure detection in dynamic systems. *Automatica*,1976, **12**(6).
12. Frank, P. Fault diagnosis in dynamic system using analytical and knowledge based redundancy – A survey and some new results. *Automatica*, 1990, **26**(3), 459-74.
13. Patton, R. Fault detection and diagnosis in aerospace system using analytical redundance. *IEE Compu. Contr. Eng. J.*, 1990, **2**, 127-36.
14. Patton, R.; Frank, P. & Clark, R. (Eds.). Fault diagnosis in dynamic systems, theory and application. Control Engineering Series. Prentice Hall, 1989.
15. Welch, G. & Bishop, G. An introduction to the Kalman filter. University of North Carolina, 2004. TR 95-041.
16. Kalman, R.E. A new approach to linear filtering and prediction problems. *Trans. ASME–J. Basic Eng.*, 1960, **82**(1), 35-45.
17. Brown, R.G. Introduction to random signal analysis and Kalman filtering, John Wiley and Sons, 1983
18. Bucy, R.S. & Joseph, P.D. Filtering for stochastic processes with applications to guidance. John Wiley & Sons, 1968, Ed. 2. AMS Chelsea Publ., 2005.
19. Jazwinski, Andrew H. Stochastic processes and filtering theory. Academic Press, New York, 1970.
20. Daga, L. 2008. http://leonardodaga.insyde.it/Simulink/RTBlock set.htm

Contributors

Dr Rahee Walambe received BE (Electrical Engineering) from Pune University and PhD (Robotics and AI) from Lancaster University (Intelligent Control Group), UK. She is currently working as an independent consultant and is associated with Pune University where she provides consultancy and teaching services. Her research interests include controls and applications, mobile robotics (Kinematics and Design), emotions modeling in artificial systems, safety integration within mobile robot architectures.

Mr Niteen P. Bhange received BE (Mechanical Engineering) from Pune University and MTech (Aerospace Engineering) from IIT Kanpur. He is currently working with Coral Digital Technologies (P) Ltd. His areas of specialisation are flight mechanics, control, and guidance.

Mr Nitin Kumar Gupta obtained post graduate degrees from IIT Bombay and University of Maryland, USA. He is currently serving as the Director and CEO at IDeA Research, a commercial R&D lab based in Pune. His research interests include dynamics, modeling, simulation, controls, and embedded systems.

Dr N. Ananthkrishnan has received his Bachelors, Masters and Doctoral degrees in Aerospace Engineering from IIT Bombay and Kanpur. He is presently working as Director and Co-Founder, IDeA Research & Development (P) Ltd, Pune. His areas of specialisation are modeling and simulation, dynamics and control, design and optimisation, especially for aerospace systems.