Defence Science Journal, Vol 43, No 4, October 1993, pp 429-447

PC-Based Flight Patb Reconstruction Using UD Factorisation Filtering Algorithm

Girija Gopalaratnam and J.R. Raol
Flight Mechanics *and Control* Division
National Aerospace Laboratories, Bangalore - 560 017

ABSTRACT

The results of flight path reconstruction using UD factorisation-based Kalman filtering algorithm are presented. The algorithm was implemented using PC-MATLAB functions and validated for simulated as well as real flight data. It is of considerable relevance to analysis of aircraft accident data and general flight data for aerospace vehicles.

1. INTRODUCTION

Flight path reconstruction is the process of determining the time histories of an aircraft's position and velocity from measurements made in flight. The results of flight path reconstruction can be used to identify the aerodynamic model of the aircraft through regression analysis and hence to obtain aircraft performance data.

The dynamic flight data recorded from sensors are prone to bias and scale factor errors. Flight path reconstruction utilizes the redundancy present in the recorded inertial and air data variables to obtain the best estimate of states together with scale factor and bias errors.

Flight path reconstruction involves estimation of the state of the aircraft, and errors of inertial, air data and Euler angle measurements. State estimation is done using the aircraft kinematic equations relating acceleration, velocity and displacement and including biases and scale factors as unknown parameters. Thus, flight path reconstruction becomes a joint state and parameter estimation problem. UD factorisation-based Kalman filtering algorithm is used for state and parameter estimation. UD factorisation has certain advantages: triangular structure of matrices, which avoids time-consuming square rooting operations, numerical reliability, stability and accuracy[1,2].

The UD filtering algorithm was initially implemented on NAL's UNIVAC computer for linear state estimation problem[6]. It has been used for analytic sensor failure detection and for correction studies for a fighter aircraft[3]. The application of the algorithm was then extended to handling nonlinear kinematic consistency checking for aircraft data[4].

In this paper, the results of flightpath reconstruction implemented using PC-MATLAB functions are presented. It is programmed in interactive manner for PC AT 3861387 microcomputer. For linearisation of nonlinear functions, a finite difference method is used. The aircraft data is simulated using a nonlinear model, including bias, scale factors and random noise. The filtering algorithm is validated for linear/nonlinear models using simulated data and some real flight data.

2. FLIGHT PATH RECONSTRUCTION PROBLEM

Using the accelerometer and rate gyro measurements and the kinematic equations of motion of the vehicle, its states are reconstructed and the scale factor and bias errors are estimated. The reconstructed states are used to compute the flow angles and velocity and compared with the measured angles and velocity.

Received 22 April 1992

The mathematical model used for flight path reconstruction, in general, is described by kinematic equations with state variables consisting of three linear velocities, u, v and w, and three Euler angles ϕ, θ and φ. The input variables are the linear accelerations, a_x, a_y and a_z, and the angular rates, p, q and r. The observations are the flight path velocity, V, angle of attack, α, and sideslip angle, β. The following are the state and observation equations used:

State equations:

$$\dot{u} = -qw + rv + a_x - g\sin\theta$$
$$\dot{v} = -ru + pw + a_y + g\cos\theta\sin\phi$$
$$\dot{w} = -pv + qu + a_z + g\cos\theta\cos\phi$$
$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$
$$\dot{\theta} = q\cos\phi - r\sin\phi$$
$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta \qquad (1)$$

Measurement equations:

$$V = \sqrt{u^2 + v^2 + w^2}$$
$$\beta = \tan^{-1}[v/u]$$
$$\alpha = \tan^{-1}[w/u] \qquad (2)$$

The input and output measurements could be erroneous due to the random noise present, unknown measurement biases and scale factor errors in instrumentation. For our problem, control variables p, q, r, a_x, a_y and a_z are assumed to be biased. Measurements of observables V, β and α are assumed to be biased and improperly scaled. The process and measurement noises are assumed to be Gaussian with zero mean. The above kinematic equations can be written in state space form as:

$$\dot{x}(t) = f(x, u, \Theta) + G\,w(t)\,;\; x(0) = x_0$$
$$y(t) = h(x, u, \Theta) \qquad (3)$$
$$y_m(k) = y(k) + v(k) \qquad k = 1, 2, \ldots, N$$

where

$x^T = [u, v, w, \phi, \theta, \psi]$ is the state vector

$u^T = [p, q, r, a_x, a_y, a_z]$ is the input vector

$y^T = [V, \beta, \alpha]$ is the output vector

$y_m^T = [V_m, \beta_m, \alpha_m]$ is the output measurement vector

$w^T = [w_u, w_v, w_w, w_\phi, w_\theta, w_\psi]$ is the process noise vector

$v^T = [n_V, n_\beta, n_\alpha]$ is the measurement noise vector

The vector Θ contains the unknown parameter: scale factors and bias errors in the measurements. Given the above nonlinear model and a set of noisy input and output measurements, the flight path reconstruction problem involves estimation of the system state x and the unknown parameters Θ. This is done by UD factorisation-based extended Kalman filtering algorithm for the present case.

3. EXTENDED KALMAN FILTERING AND UD FACTORISATION

Extended Kalman filter is a sub-optimal solution to a nonlinear filtering problem. The nonlinear functions are linearised about each new estimated/filtered state trajectory as soon as it becomes available. Simultaneous estimation of states and parameters is achieved by augmenting the state vector with the unknown parameters and applying the filtering algorithm to the augmented nonlinear model[9, 10]

The new augmented state vector is

$$x_a^T = [x^T / \theta^T]$$
$$\dot{x}_a^T = \begin{bmatrix} f(x_a, u, t) \\ 0 \end{bmatrix} + \begin{bmatrix} G \\ 0 \end{bmatrix} w(t) \qquad (4)$$
$$= f_A(x_a, u, t,) + G_A\,w(t)$$

$$y(t) = h_a(x_a, u, t) \qquad (5)$$

$$y_m(k) = y(k) + v(k)\,,\; k = 1, \ldots, n$$

where

$$f_A(t) = [f^T \mid 0^T]\,;\; G_A^T = [G^T \mid 0^T]$$

The estimation algorithm is obtained by linearising Eqns (4) and (5) around the prior/current best estimate of the state at each time and then applying filtering algorithm to the linearised model. The linearised system matrices are defined as:

$$A(k) = \left.\frac{\delta f_A}{\delta x_a}\right|_{x_a = \hat{x}_a(k),\; u = u(k)}$$

and the state transition matrix

$\Phi(k) = \exp[-A(A(k) \cdot \Delta T]$,

where

$\Delta T = t_{k+1} - t$

For the sake of clarity and completeness, the filtering algorithm is given in two parts; (i) time propagation, and (ii) measurement update.

3.1 Time Propagation

The current estimate is used to predict the next state, so that the states are propagated from the present state to the next time instant.

The predicted state is given by

$$\hat{x}_a(k+1/k) = \hat{x}_a(k) + \int_{t_k}^{t_{k+1}} f_A[\hat{x}_a(t), u(k)]\, dt \qquad (6)$$

In the absence of knowledge of process noise, Eqn (8) gives the predicted estimate of the state based on the initial/current estimate. The covariance matix propagates from instant k to $k+1$ as

$$P(k+1/k) = \Phi(k)\, \hat{P}(k)\, \Phi^T(k) + G_A(k) Q_A G^T(k) \qquad (7)$$

where P $(k+1/k)$ is the predicted covariance matrix for instant value of $k+1$, G_A is the process noise related coefficient matrix, and Q is the process noise covariance matrix.

3.2 Measurement update

The Kalman filter updates the predicted estimates by incorporating the measurements as and when they become available as follows:

$$\hat{x}_a(k+1) = \hat{x}_a(k+1/k) + K(k+1)\{y_m(k+1) - h_a[\hat{x}_a(k+1/k), u(k+1), t]\} \qquad (8)$$

where K is the Kalman gain matrix.

The covariance matrix is updated using the Kalman gain and the linearized measurement matrix from the predicted covariance matrix $\hat{P}(k+1/k)$ as

$$\hat{P}(k+1) = [I - K(k+1) H(k+1)]\, \hat{P}(k+1/k)\, [I - K(k+1) H(k+1)]^T + K(k+1) R(k) K^T(k+1) \qquad (9)$$

The Kalman gain is given by

$$K(k+1) = \hat{P}(k+1/k)\, H^T(k+1)\, [H(k+1)\, \hat{P}(k+1/k)\, H^T(k+1) + R]^{-1} \qquad (10)$$

Substitution of Kalman gain matrix $K(k+1)$ into the process covariance matrix gives

$$\hat{P}(k+1) = [I - K(k+1) H(k+1)]\, \hat{P}(k+1/k) \qquad (11)$$

Since Eqn (11) is numerically ill-conditioned, UD factorisation-based implementation of Eqns (8)-(10) is used for flight path reconstruction.

4. UD FACTORISATION FILTERING

The UD factorisation filter has the following merits[2].

(a) It is numerically reliable, accurate and stable,
(b) It is a square root type algorithm, but does not involve square rooting operations,
(c) The algorithm is most efficiently and simply mechanised by processing vector measurements (observables), one component at a time, and
(d) For linear systems, UD filter is algebraically equivalent to the Kalman filter.

In the UD filter, the covariance update formulae and the estimation recursion are reformulated, so that the Covariance matrix does not appear explicitly. Specifically, we use recursions for U and D factors of covariance matrix $P = UDU^T$, where U is a unit upper triangular matrix and D is a diagonal matrix. Computing and updating with triangular matrices involve fewer arithmetic operations and thus greatly reduce the problem of round off errors which might cause[10] ill-conditioning and subsequent divergence of the algorithm. The filter algorithm is given in two parts:

4.1 Time Update

We have for the covariance update,

$$P(k+1/k) = \Phi \hat{P}(k) \Phi^T + G_A Q G_A^T \qquad (12)$$

Given $\hat{P} = \hat{U}\hat{D}\hat{U}^T$ and Q as the process noise covariance matrix, the time update factors $\bar{U}$ and $\bar{D}$ are obtained through modified Gram-Schmidt orthogonalisation process.

We may define $W = [\Phi \hat{U} \mid G_A]$ $\bar{D} = \text{diag}[\hat{D}, Q]$

with $W^T = [w_1, w_2, \ldots w_n]$

P is reformulated as $P = \bar{W}\bar{D}\bar{W}^T$. The U, D factors of $\bar{W}\bar{D}\bar{W}^T$ may be computed as described below.

For $j = n, \ldots, 1$ the following equations are recursively evaluated.

$$D = \langle w_1, \ldots, w_j \rangle_D$$

$$\bar{U} = (1/\bar{D}_j) \langle w_1, \ldots, w_i \rangle_D \qquad i = 1, 2, \ldots, j-1 \qquad (13)$$

$$w_i = w_i - \bar{U}_{ij} w_j$$

where $\langle w_1, \ldots, w_j \rangle_D = w_i^T D w_j$ is the weighted inner product between w_i and w_j.

4.2 Measurement Update

The measurement update in Kalman filtering combines *apriori* estimate $\bar{x}$ and error covariance $\bar{P}$ with a scalar observation $z = a^T x + v$ to construct an updated estimate and covariance given as:

$$
\begin{aligned}
&K = \bar{P}\, a / \alpha, \\
&\hat{x} = \bar{x} + K\,(z - a'f), \\
&\alpha = a\; P\, a + r \\
&\hat{P} = \bar{P} - K\, a\, \bar{P}
\end{aligned}
\qquad (14)
$$

where $\bar{P} = \bar{U}\, \bar{D}\, \bar{U}^T$, a = measurement matrix, r is the measurement noise covariance, and z = noisy measurements.

Kalman gain K and updated covariance factors $\hat{U}$ and $\hat{D}$ can be obtained from the following equations:

$$
\begin{aligned}
&f = \bar{U}^T a \;;\; f^T = (f_1, \ldots, f_n) \\
&v = \bar{D} f \;;\; v_i \bar{d}_i \;; i = 1, 2, \ldots, n \\
&\hat{d}_1 = d_1 r/\alpha_1, \alpha_1 = r + v_1 f_1 \;;
\end{aligned}
\qquad (15)
$$

For $j = 2, \ldots, n$ recursively the following equations are evaluated :

$$
\begin{aligned}
&\alpha_j = \alpha_{j-1} + v_j f_j \\
&\hat{d}_j = \bar{d}_j \alpha_{j-1}/\alpha_j \\
&\bar{d}_j = \bar{d}_j \alpha_{j-1}/\alpha_j \\
&\hat{u}_j = \bar{u}_j + \lambda_j k_j, \lambda_j = -f_j/\alpha_{j-1} \\
&K_{j+1} = K_j + v_j \bar{u}_j \\
&\text{where } \hat{U} = [\hat{u}_1, \ldots, \hat{u}_n]
\end{aligned}
\qquad (16)
$$

and Kalman gain is given by $K = K_{n+1}/\alpha_n$

where $\bar{d}$ = is the predicted diagonal element, and $\hat{d}_j$ is the updated diagonal element of the D matrix.

As already mentioned, calculation of the matrices $A(k)$ and $H(k)$ for nonlinear systems is accomplished by finite difference method. In the present mechanisation, A and Hare computed by finite difference method as against the analytical method, so that there is no need to make any programming changes when alternative nonlinear models are to be used.

$$
A_{ij} = \frac{\delta f_i}{\delta x_j} = \frac{f_i(x_j + \Delta x_j) - f_i(x_j)}{\Delta x_j} \qquad (17)
$$

for $i = 1, 2, \ldots, m$; and $j = 1, 2, \ldots. n$;

Δx_j = perturbations step sue = ϵx_j

For a small perturbation Δx in each of the unknown variables, the perturbed values $f(x_j + Ax;)$ of each of the unperturbed values $f = x_j$ are computed. The corresponding elements of A_{ij} are given by the finite difference in functions to changes in that parameter. A step size of $\epsilon = 10^{-5}$ is considered to be adequate.

This factorisation algorithm has been implemented in MATLAB (PC AT 386/387) using the existing functions of MATLAB as well as newly developed functions.

5. VALIDATION WITH SIMULATED DATA

To validate the code developed and to assess the performance oi the filter, three cases are considered.

Case I: Short period dynamics of an aircraft is simulated using the following state and observation equations.

State *equations:*

$$
\begin{aligned}
&\dot{\alpha} = \frac{Z_\alpha}{U_0} + q + b_1 + w_1 \\
&q = M_\alpha \alpha + M_q q + M_{\delta e}\, \delta e + b_2 + w_2
\end{aligned}
\qquad (18)
$$

Observation equations:

$$
\begin{aligned}
&\alpha_m = \alpha + v_1 \\
&q_m = q + v_2 \\
&n_z = \frac{-Z_\alpha}{g} \alpha + \text{bias} + v_3
\end{aligned}
\qquad (19)
$$

The system equations are written in state spaceform as:

$$
\begin{aligned}
&\dot{x} = Ax + Bu + \text{noise} \\
&z = Hx + \text{noise}
\end{aligned}
\qquad (20)
$$

where $x = [\alpha q]$ is the state vector

$z = [a,,,q_m, n_z]$ is the measurement vector

Matrices A, B and Hare given in Appendix 1 (Data file lisimdat.m). A doublet is used as input to generate the responses. Simulated data is generated by using function **DLSIM** with a sampling time of 0.03125 s. Ramdom noise with variances equal to 20 per cent of the true signal variances is generated using RAND, and added to the states and measurements. The appropriate noise statistics are fed into the filtering algorithm along with the simulated data trajectories and the states are estimated. Figures. 1 and 2 show simulated true and

noisy states and measurements. The results of UD filtering are presented in Figs. 3 to 6. In Fig. 3, the estimated measurements are plotted against the noisy measurements. In Fig. 4, the estimatedstates are plotted along with the standard deviations. Figure 5 shows residuals with their bounds computed from the estimator results. Only 2 per cent of the samples exceed the hounds. In Fig. 6, the autocorrelations of the residuals are plotted with the bounds. Less than 5 per cent of the autocorrelation values are out of the bounds, confirming that the filter residuals are white.

For the estimated states, the bounds shown in Fig. 4 are the standard deviations computed from the estimated covariance P.

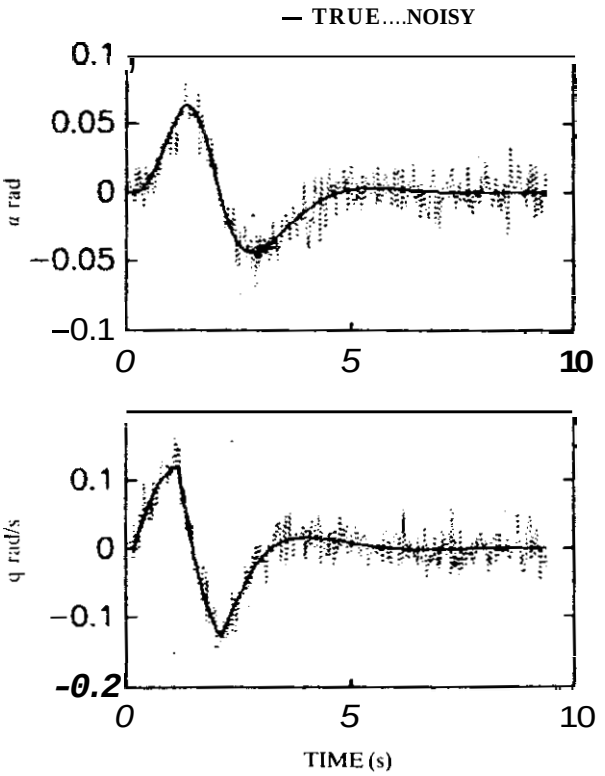


Figure 1. True and noisy states (Case I).

$$\text{Standard deviation} = \sqrt{(UDU^T)_{ii}} \qquad (21)$$

Covariance of the residuals is computed using

$$R_e(k+1) = H(k+1) * \bar{P}(k+1/k) * H^T(k+1) + R(k+1) \qquad (22)$$

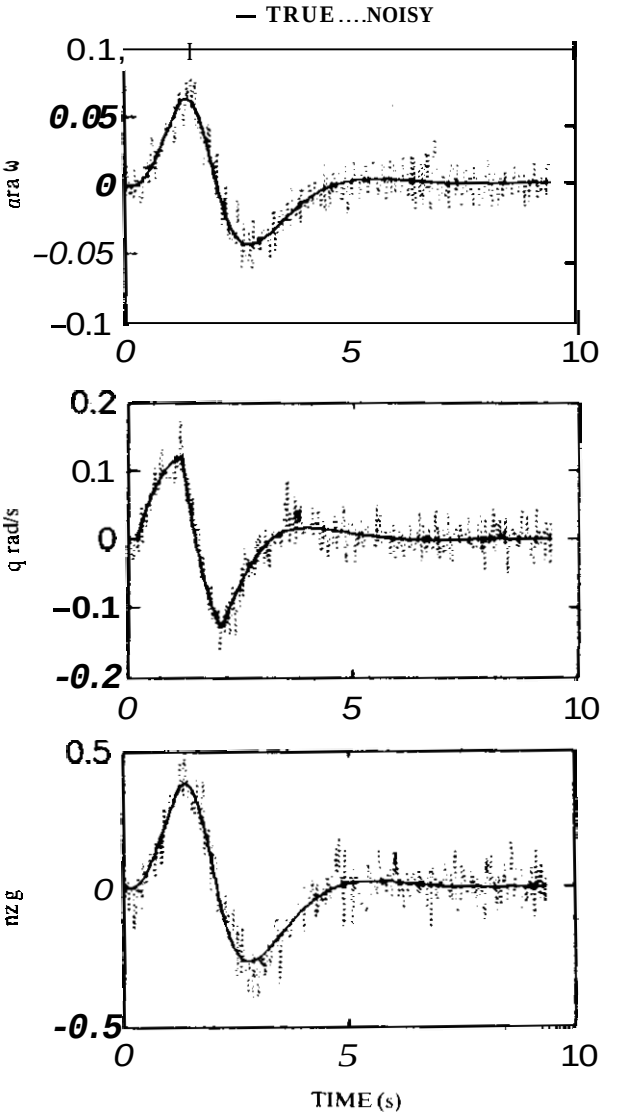


Figure 2. True and noisy measurements (Case I).

The Values $\pm \sqrt{2 * (Re(k+1)_{ii}}$ are plotted in Fig. 5 as bounds for residuals. If the covariance estimates are reasonable, 95 per cent of the samples should lie within the interval''.

The bounds for the autocorrelation function, which are used as a check for whiteness of residuals, are computed using $\pm 1.96/\sqrt{N}$, where N is the number of samples.

Case II: The following fourth order non-linear model was used to simulate the aircraft data by giving a doublet to the elevator. Fourth order RK integration method is used for simulation and the sampling time is chosen as 0.03 s.

176

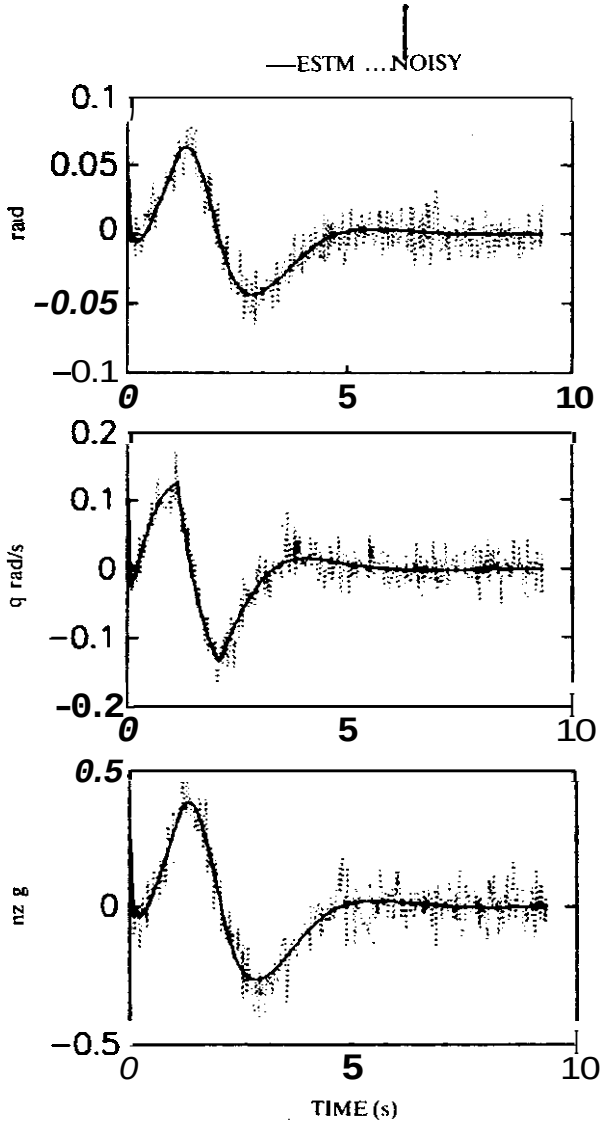


Figure 3. Estimated and noisy measurements (Case I).

The values of the derivatives used in simulation are given in Table 1.

Table 1. Derivatives used in simulation of non-linear longitudinal a/c data (Case I)

Derivative	Value
C_{x0}	0.0413
C_{xu}	−0.0581
C_{xw}	0.1570
C_{z0}	−0.1929
C_{zu}	0.2862
C_{zw}	−4.1040
C_{m0}	−0.0891
C_{mu}	0.1048
C_{mq}	−8.1959
C_{mw}	−0.4894
$C_{m\dot{w}}$	−1.4014

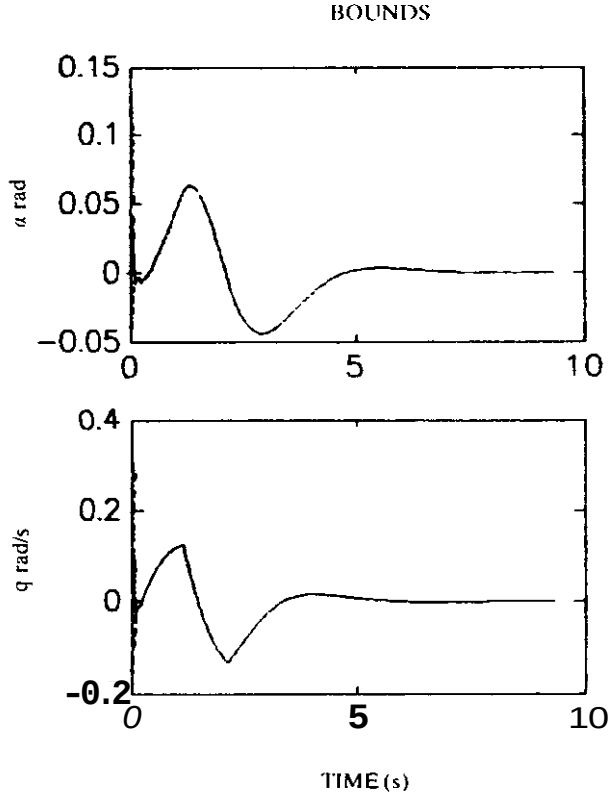


Figure 4. Estimated states with bounds (Case I).

State equations:

$$\dot{u} = -qw - g\sin\theta + \frac{\bar{q}s}{m}\left(C_{x_0} + C_{xu}\frac{u}{U_0} + C_{xw}\frac{w}{U_0}\right) + Fe/m$$

$$\dot{w} = -qu + g\cos\theta\cos\phi + \frac{\bar{q}s}{m}\left(C_{z_0} + C_{zw}\frac{w}{U_0} + C_{zu}\frac{u}{U_0}\right) \quad (23)$$

$$a_{xm} = \frac{\bar{q}s}{m}\left(C_{x_0} + C_{xu}\frac{u}{U_0} + C_{xw}\frac{w}{U_0}\right) + fe/m$$

$$a_{zm} = \frac{\bar{q}s}{m}\left(C_{z_0} + C_{zu}\frac{u}{U_0} + C_{zw}\frac{w}{U_0}\right) \quad (24)$$

$$q_m = q$$

For longitudinal flight path reconstruction, the following third order model was used.

$$\dot{u} = -qw + a_x - g\sin\theta$$

$$\dot{w} = qu + a_z + g\cos\theta$$

$$\dot{\theta} = q$$

177

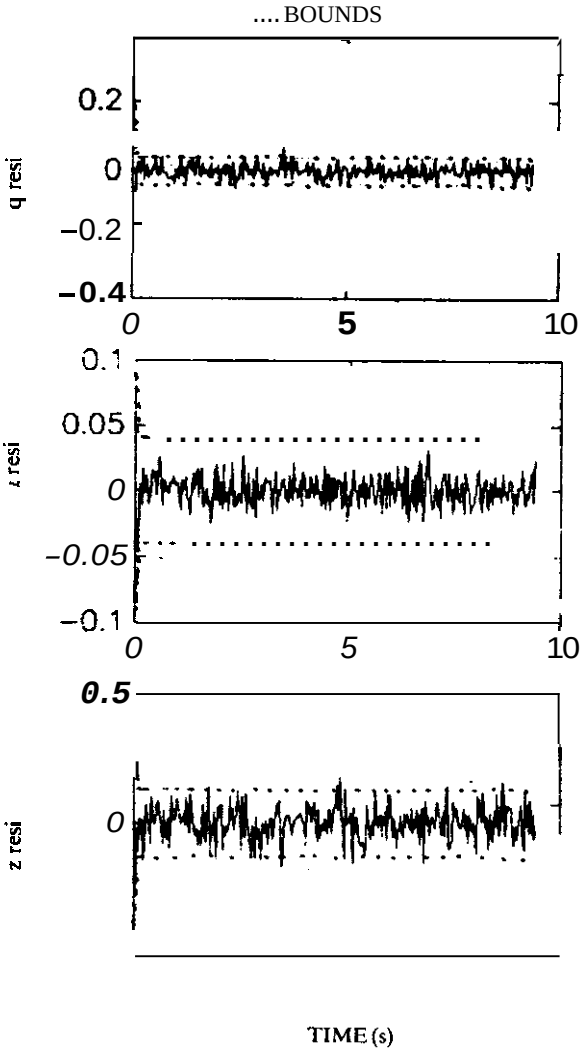


Figure 5. Residuals with bounds (Case I).

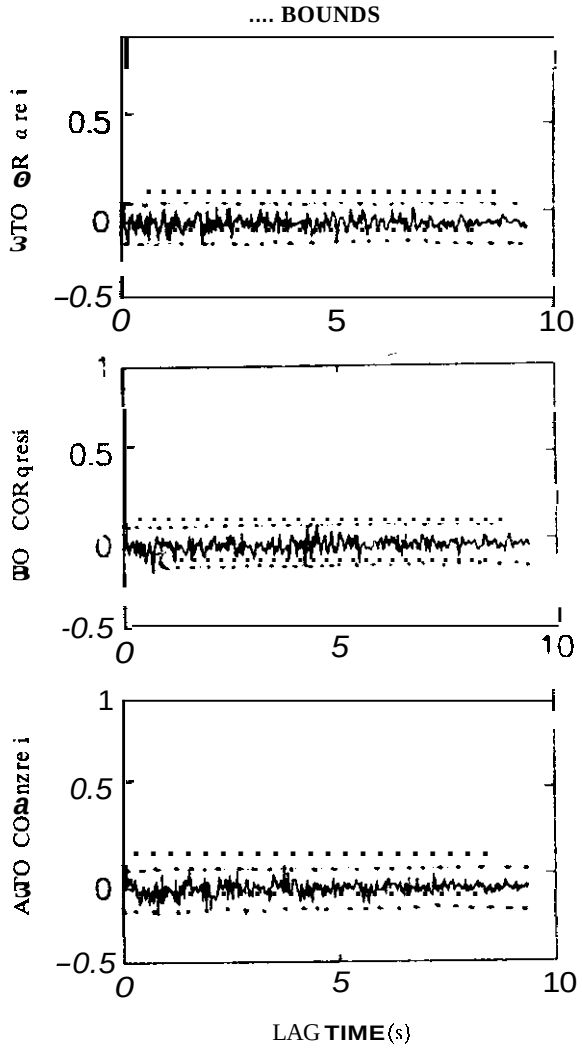


Figure 6. Autocorrelation of residuals with bounds (Case I).

Measurement equations:

$$V = \sqrt{u^2 + v^2 + w^2}$$

$$a = Tan^{-1}[(w - qx_a + py_a)/u] \qquad (25)$$

$$\theta_m = \theta$$

Biases were added to the accelerations a_x, a_z, q and V and scale factors applied to a measurement. Random noise with 20 per cent of signal variance was added to each of the states and measurements. Figures 7 and 8 show the true simulated states and measurements plotted against the noisy states and measurements.

The following model was used for the filtering:

State equations:

$$\dot{u} = -(q_m - Aq)\, w + (A, - AA,) - g\sin\theta$$

$$w = (q_m - \Delta q)\, u + (A_{zm} - \Delta A_z) + g\cos\theta \qquad (27)$$

$$\dot{\theta} = (q_m - \Delta q)$$

Observation equations:

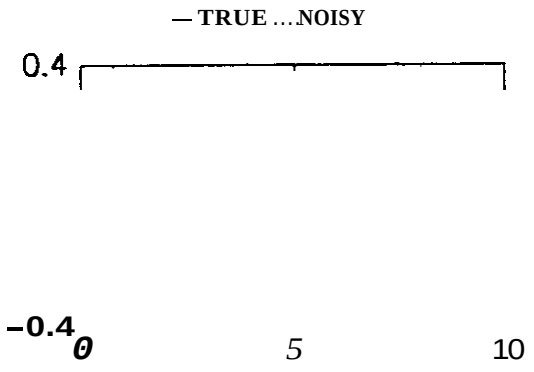
$$V_m = \sqrt{(u^2 + w^2) + AV}$$

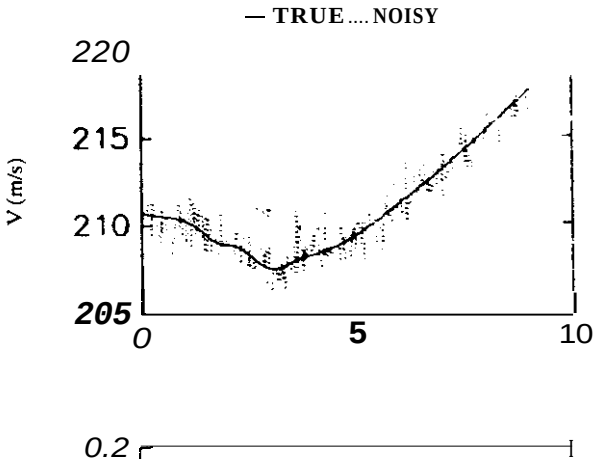
$$a_{,,} = K_a \tan^{-1}\left(\frac{w}{u}\right) + AV \qquad (28)$$

$$\theta_m = \theta$$

178

For the above model, the augmented state and input vectors are given by:





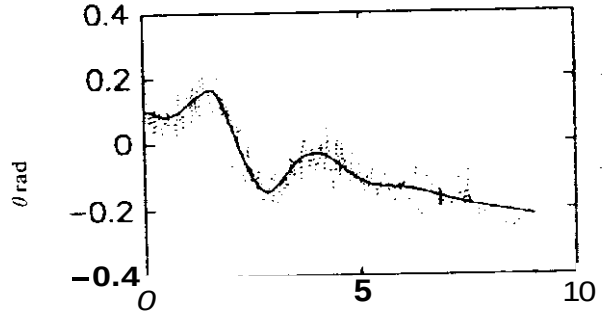
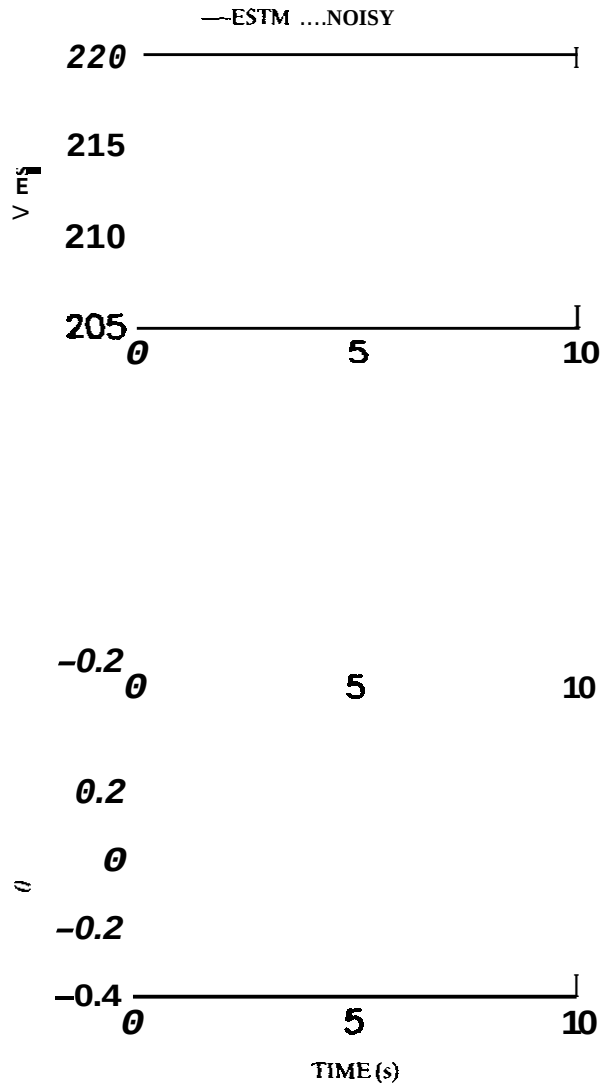


Figure 9 is a cross-plot of estimated and noisy measurements. Figure 10 shows the three states and augmented states plotted along with their standard deviations. In Fig. 11 the residuals are plotted with their bounds. About 15 points of the residuals are out of bounds, giving a confidence of about 5 per Cent in the estimates. From the autocorrelation function of residuals plotted in Fig. 12, it is evident that the residuals form a white process. The velocity rms error and error in theta are plotted in Fig. 13. The velocity rms error is computed using the relation

$$= \sqrt{(u - \hat{u})^2 + (v - \hat{v})^2 + (w - \hat{w})^2} \qquad (30)$$

and error in theta is $(0 - \hat{\theta})$.



The values of the scale factors and biases used in simulation, the starting values for filtering and those estimated through the algorithm along with their standard deviations are listed in Table 2. The closeness of the values and the low standard deviation bounds on the estimated states validate the algorithm developed for both linear and nonlinear cases.

111: Flight path reconstruction for real flight data of a transport aircraft.

Flight path reconstruction of longitudinal dynamics of a transport aircraft is done using the following seventh order state equations and seven measurements.

State equations:

$$\dot{u} = -(q - \Delta q)w + r_m v + (g_x - \Delta a_x) - g \sin\theta$$

$$\dot{v} = -r_m u + p_m w + a_y + g \cos\theta \sin\phi + (a_{zm} - \Delta a_z)$$

$$= p_m + (q_m - \Delta q)\sin\phi \tan\theta + r_m \cos\phi \tan\theta \quad (31)$$

$$\dot{\psi} = ((q_m - \Delta q)\sin\phi + r\cos\phi)\sec\theta$$

Measurement equations:

$$\beta_m = K_\beta \tan^{-1}[v/u] + \Delta\beta$$

$$a_z = K_\alpha \tan^{-1}[w/u] + \Delta\alpha$$

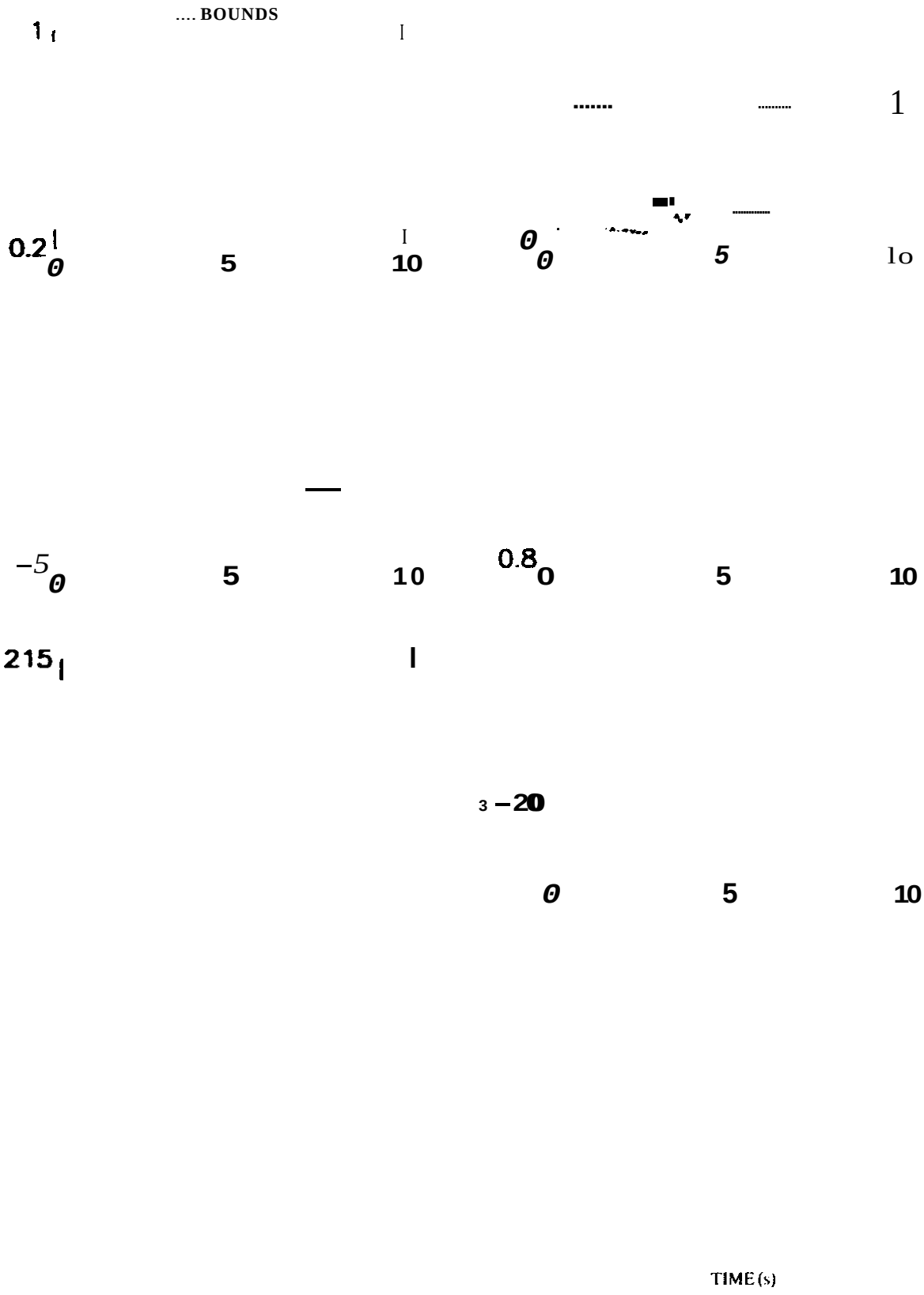
$$= \theta$$

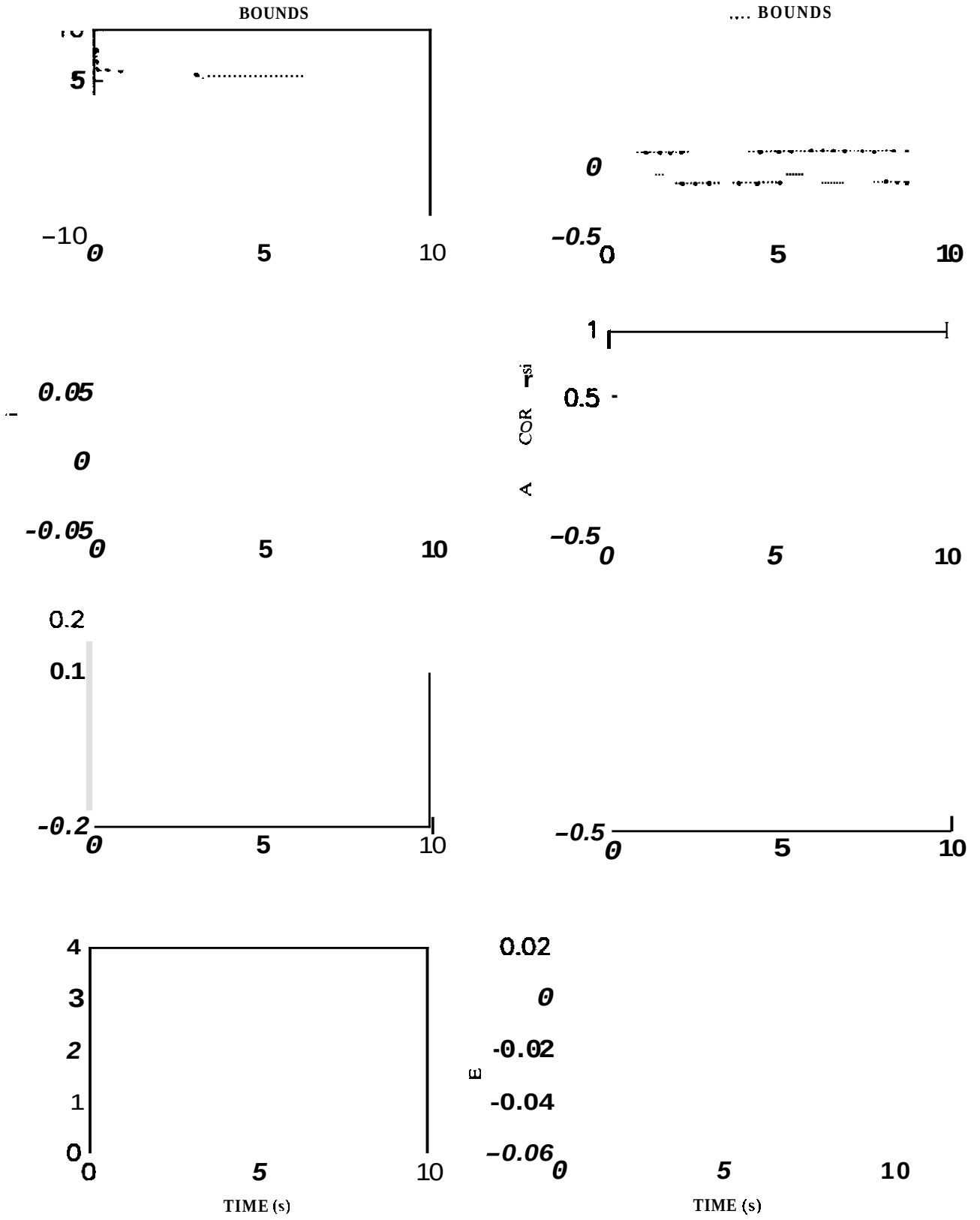
$$\phi_m = \phi$$

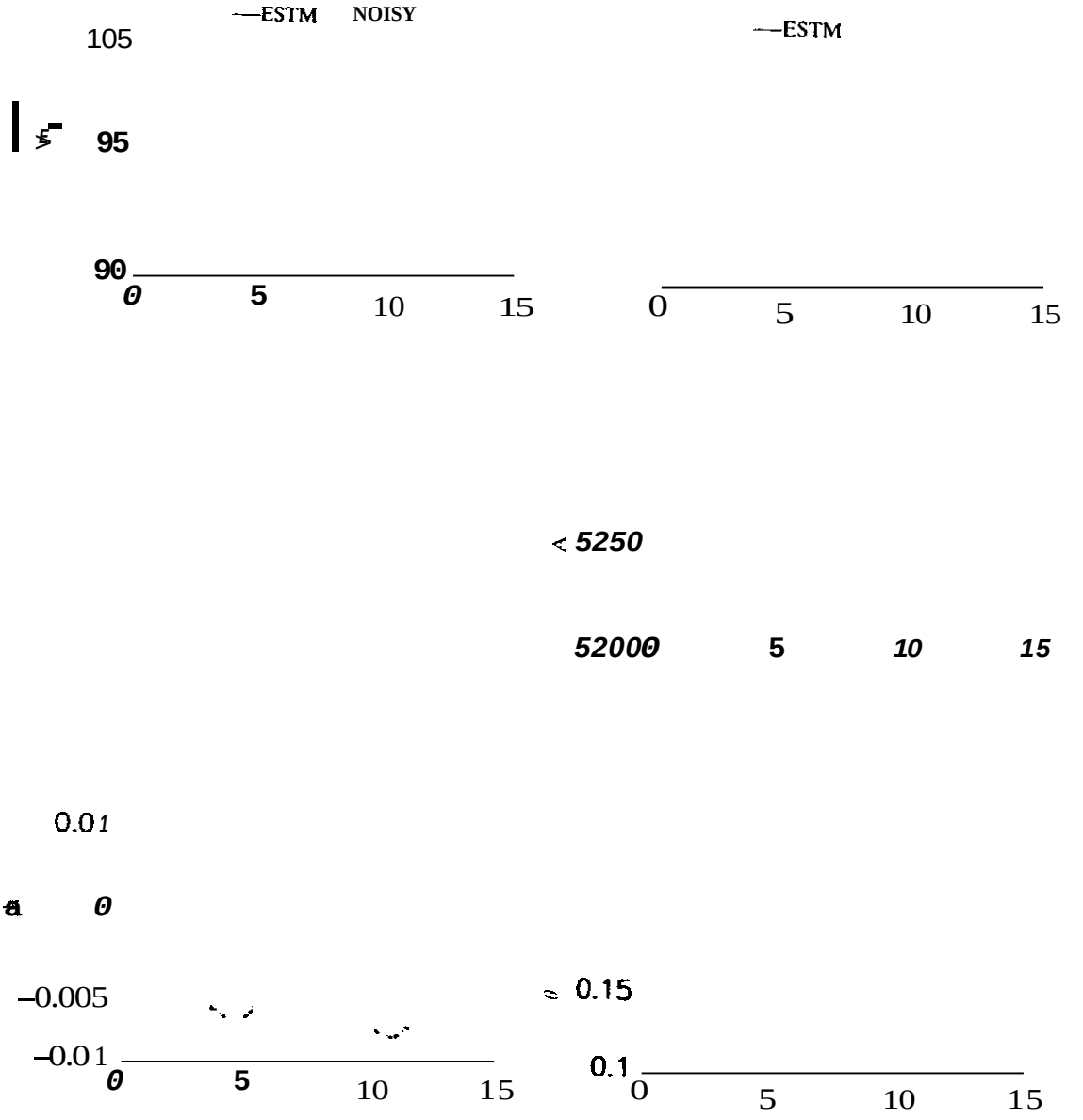
$$h_m = h$$

A typical time segment of 12.0s with a sampling time of 0.05 s is used in the analysis. Figure 14 is a plot of the

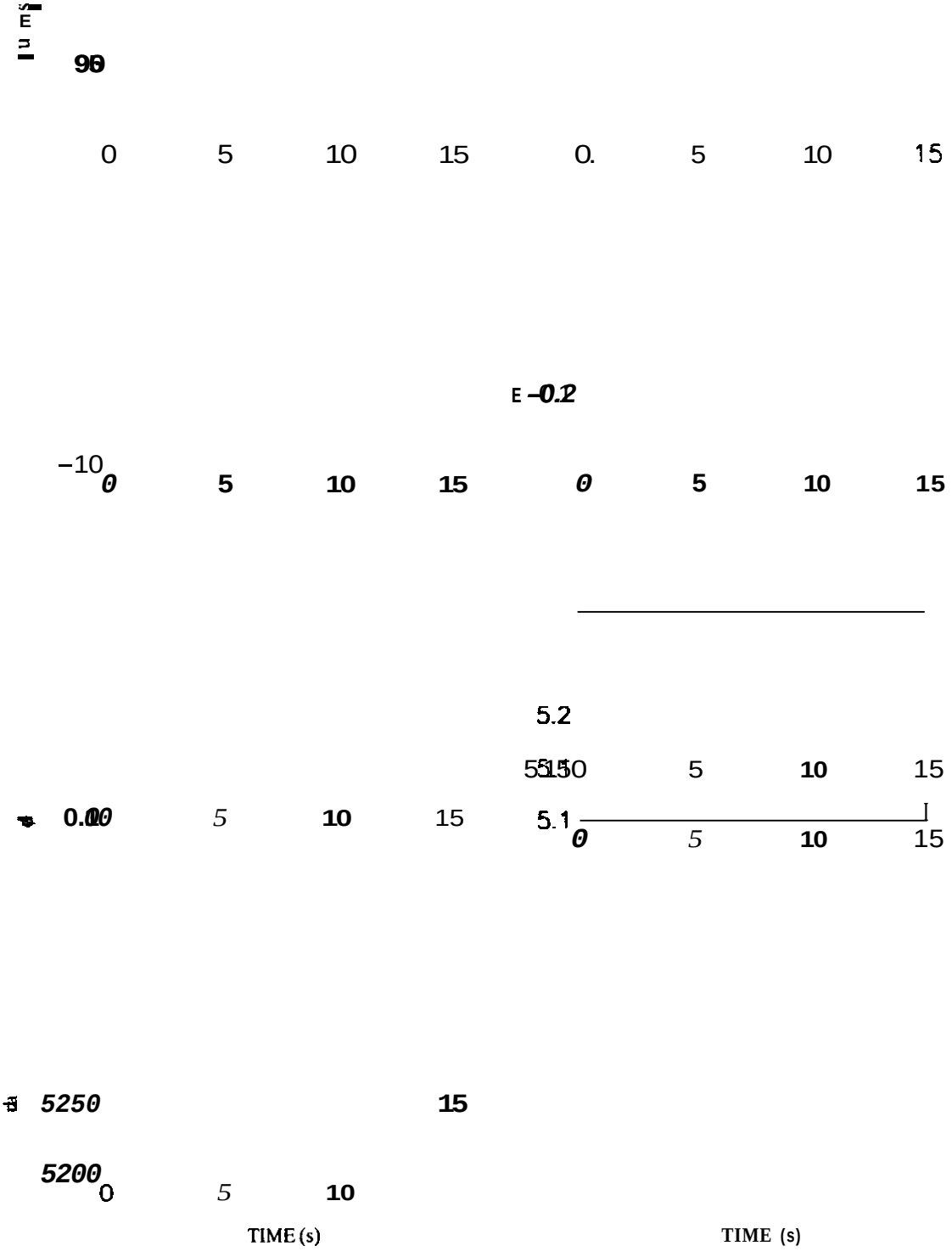
Δa_x	0.1128	0.313	
	0.5078	0.707	
Δq	0.00039	0.0005	0.0006 –5)
Aa	0.002	0.0005	
AV	5.0	7.0	5.374 (3.64)
K_α	1.2	1.0	1.2010 (4.86e–5)


.... BOUNDS
0.2
0
5
10
0
0
5
10
−5
0
5
10
0.8
0
5
10
215
3 −20
0
5
10
TIME (s)


BOUNDS
BOUNDS
5
0
-10
0
5
10
0.05
0
-0.05
0
5
10
1
0.5
-0.5
0
5
10
0.2
0.1
-0.2
0
5
10
-0.5
0
5
10
4
3
2
1
0
0
5
10
0.02
0
-0.02
-0.04
-0.06
0
5
10
TIME (s)
TIME (s)


—ESTM NOISY
105
95
90
0
5
10
15
—ESTM
0
5
10
15
5250
52000
5
10
15
0.01
0
–0.005
–0.01
0
5
10
15
0.15
0.1
0
5
10
15

BOUNDS



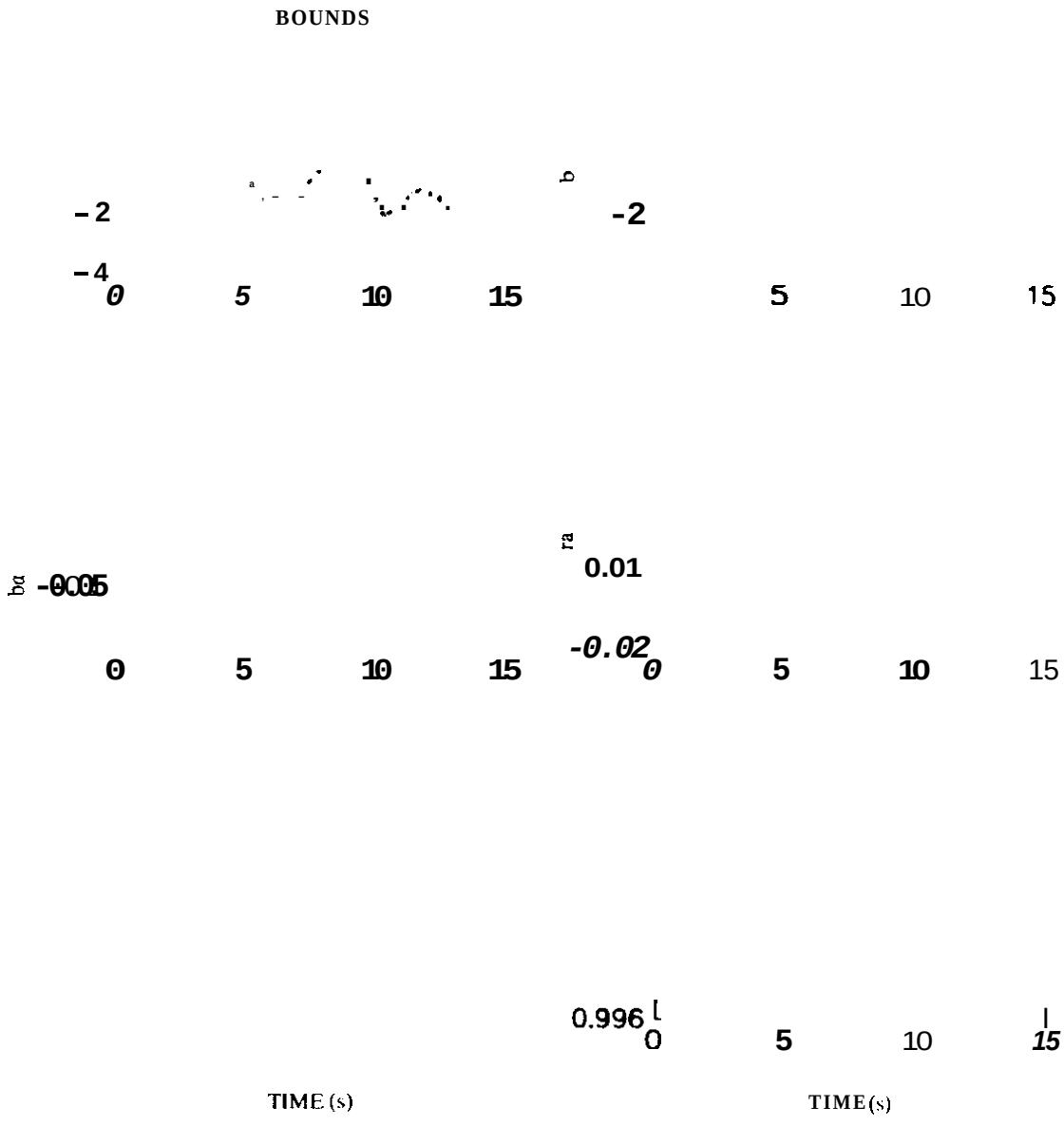
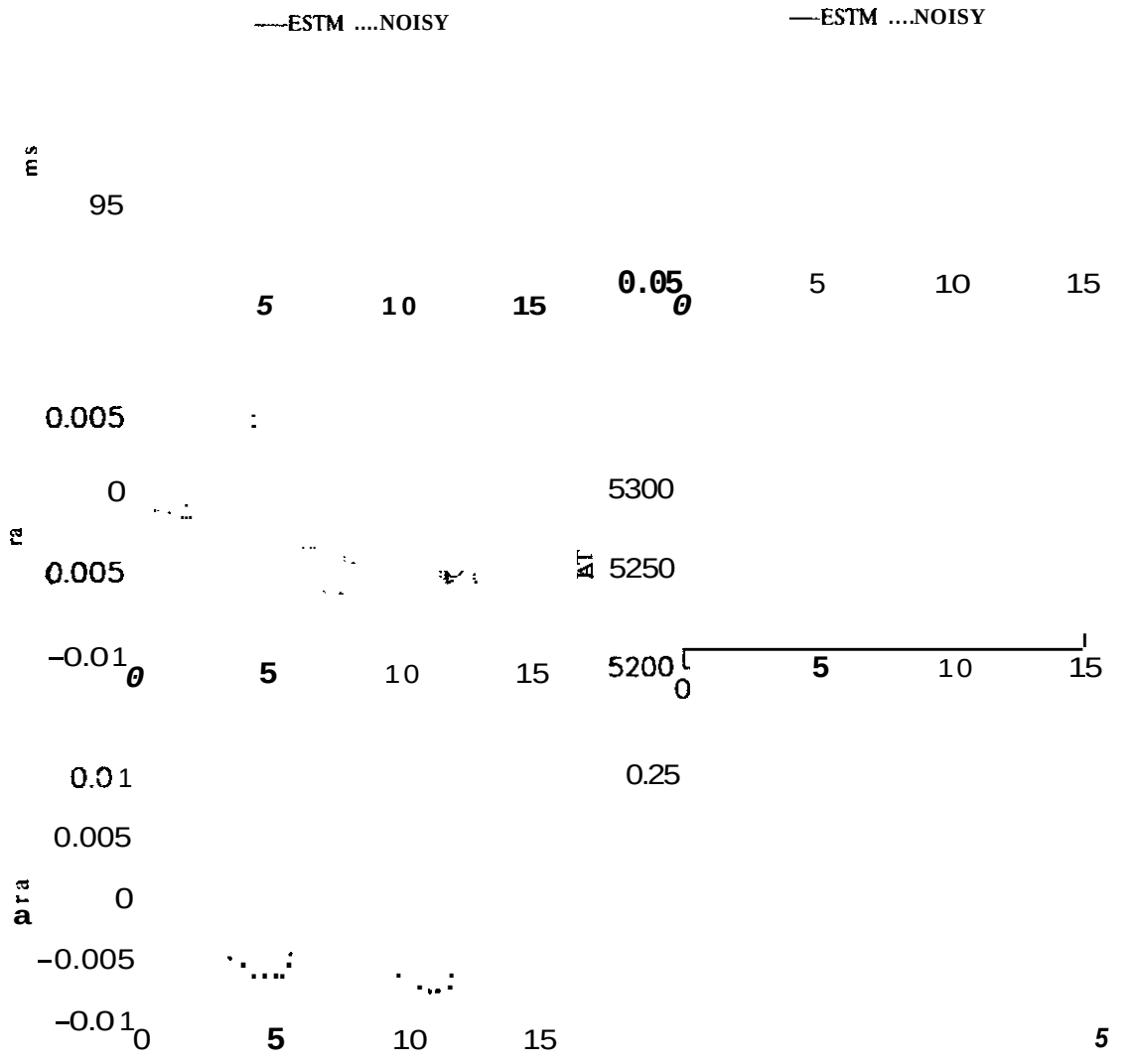
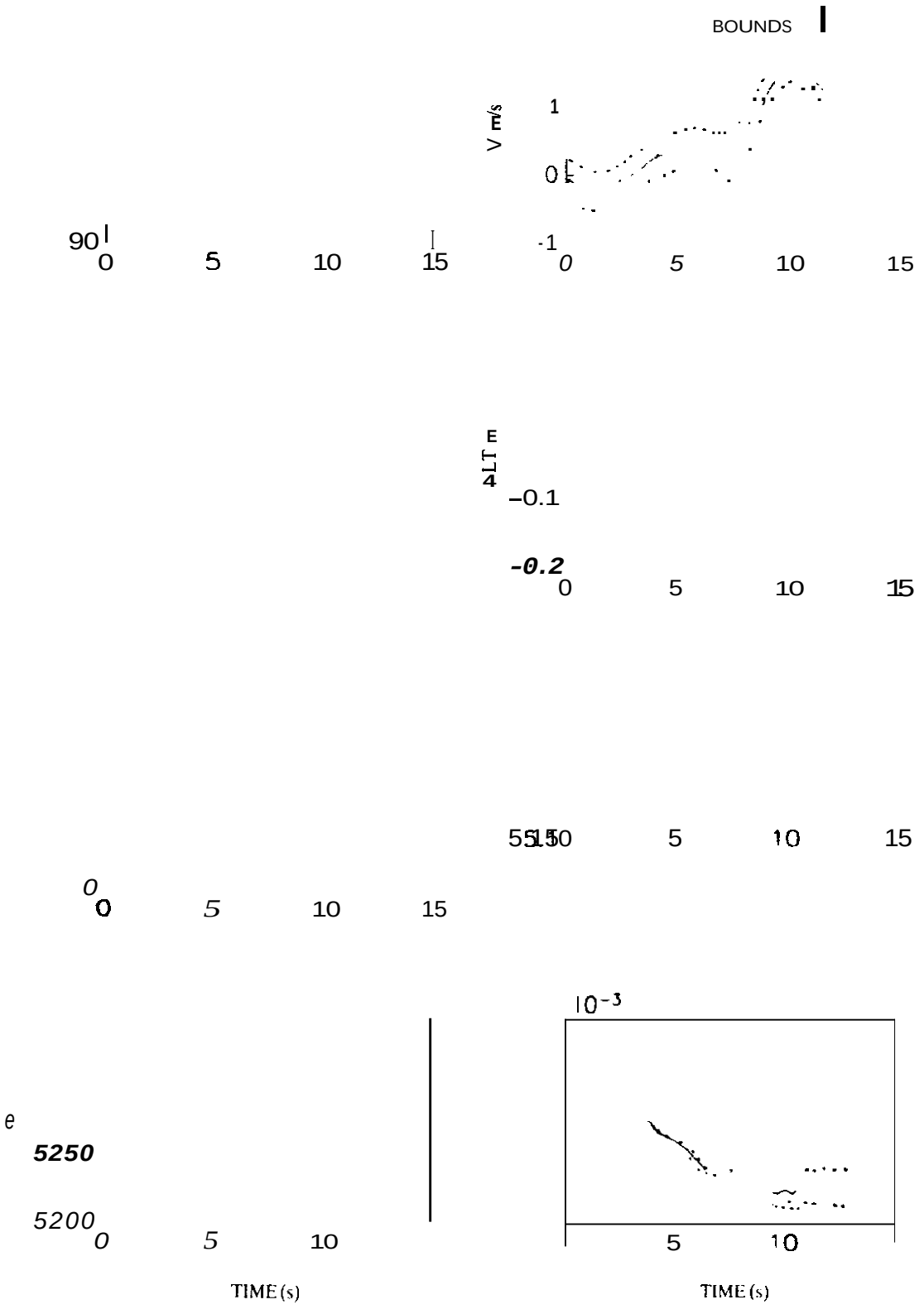


Figure 15. (continued)

curve fit between the measured and estimated measurements. Figure 15 shows the estimated states and parameters with their standard deviations. From Fig. 14 it is seen that the match between the estimated and measured trajectory is not good. The measured responses reveal that there is time error in considering the *a* measurement. A closer look at the measurement model showed that the corrections in position errors of sensors to measure V, and β with respect to the centre of gravity were not made. These corrections were incorporated in the measurement model and Fig. 16 shows the improved response matches. Figure 17 is a plot of the convergence of states and parameters. The convergence of some of the parameters is poor; the response match of some of the lateral measurements is also poor. This is mainly because the maneuvre being analysed is a longitudinal one and hence the lateral mode is not excited and as such these can be removed



16. real data (Case III).


BOUNDS
V m/s
1
0
-1
90
0
5
10
15
-0.1
-0.2
0
5
10
15
5250
5200
0
5
10
10^{-3}
TIME (s)
TIME (s)

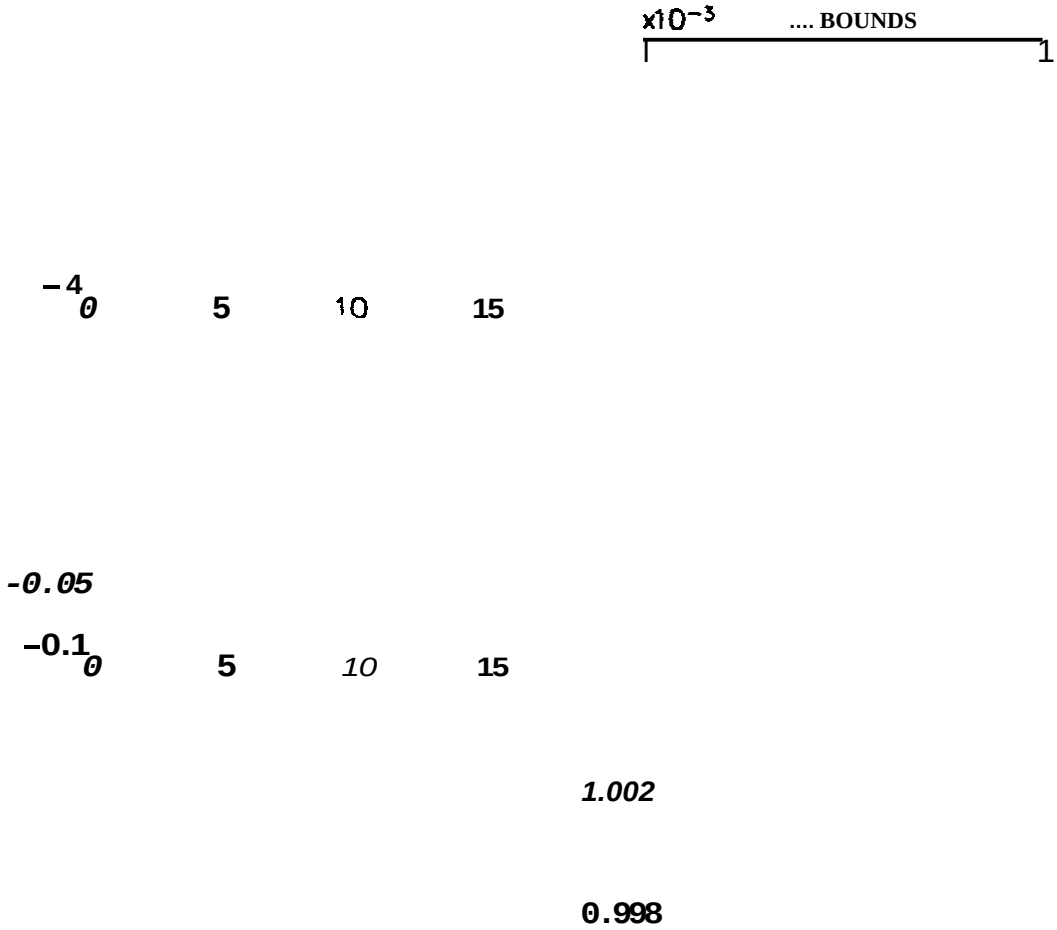


Figure 17. (continued)

	0.000	−.0034	9.527e−04
	O.W	4.592e−04	0.002
	0.000	2.553e−05	5.229e−06
$\Delta\alpha$	0.000	0.013	2.415e−04
	O.W	−0.0186	6.25e−05
K_a	1.000	0.995	6.06e−07
	1.W	1.0037	3.12e−08

from the observations. The parameters estimated for the transport aircraft with their standard deviations are listed in Table 3.

6. FURTHER SCOPE OF APPLICATION OF FPR ALGORITHM

The UD filter-based algorithm can be modified to include the measurements from different sets of sensors,

e.g., radar measurements, telemetered signals from the vehicle, etc. This requires formulation of measurement models appropriate to the type of measurements being used.

Suppose there are several different types of sensors which provide measurements on m physical responses. Some of the sensors may measure the same quantity providing redundant measurements. There may be situations where the different measurements required are obtained from different sets of measurement sources. The Kalman filter algorithm can be used to handle all these various situation so as to get the best estimates of the states of the vehicle.

The output measurement equations can be written as

$$y_1 = \quad u,$$

$$y_{m1}(k) = y_1(k) + \upsilon_1(k) \text{ (say for onboard measurements)}$$

$$Y2 = H_2 \quad u, \tag{33}$$

$$y_{m2}(k) = y_2(k) + \upsilon_2(k) \text{ (say for measurements from radar)}$$

υ_1 and are the measurement errors for the two sets of sensor measurements. Optimal use of this combined information model would generally give more accurate FPR results. Appropriate changes in the covariance matrices for the measurement of noise processes could be incorporated to handle measurements from various sensors. This aspect has applications in FPR of missiles and other aerospace vehicles and accident data analysis requiring accurate estimation of position of the vehicle. Since the algorithm is validated for linear/nonlinear systems with simulated and data, presently efforts are on toexpandthescopeofthealgorithmonthe above lines. It is also being updated to perform aerodynamic parameter estimation, thereby providing a unified methodology for flight data analysis using PC-based MATLAB. This alternative methodology handles process noise as well as measurement noise, unlike the output error and equation error methods which handle respectively only measurement noise and process noise.

7. CONCLUSION

In this paper, the implementation and evaluation of UD filtering algorithm for FPR are described. The algorithm is mechanised using existing as well as newly developed MATLAB functions implementable on PC AT 386/387 microcomputer. The results show that the code is validated for both simulated and real data. It can be extended to handle FPR for missiles and other aerospace vehicles. The algorithm developed can be used for aerodynamic parameter estimation as well. In this context. it is contemplated to obtain information on the covariances of noise processes (control input and measurement noise) by using time series modelling approach.

REFERENCES

1. Gelb. A. (Ed.) Applied optimal estimation. MIT Press, Massachussetts, 1974.
2. Bierman, G.J. Factorization methods for discrete Sequential Estimation. Academic Press. New York, 1977.
3. Raol J.R. & Sinha. N.K. Orbit determination via UD filters. *In* Proc. IEEE Conf. on Decision Control, 12-14 December 1984. Las Vegas, USA.
4. Raol, J.R. & Sinha. N.K. Estimation of orbital states of a satellite. *In* Proc. IFAC Symposium on Identification and Slate Parameter Estimation. 3-7 July 1985, London, UK.
5. Raol, J.R. & Sinha, N.K. On the orbit determination prohlem. *IEEE* Acrospuce *and* Electronics *Systems,* 1985, AES-21.274-91.
6. Azra Jabeen, Prabhulatha & Rose Joseph. Implementation of UD factorisation algorithm. Dept. of Electronics Engg., MVJ College of Engg., Bangalore, 1988. BE Project Report.
7. Raol, J.R. ; Girija, G. Parameswaran, V. A sensor failure detection scheme using analytical redundancy and U-D filtering algorithm for LCA. NAL, Bangalore, 1990. NAL PD FC
8. Ravishanker, P. Kinematic consistency checking for aircraft data using factorisation based extended Kalman filtering techniques. Regional Engineering College, Kakatiya University, Warangal, 1989. MTech Thesis.
9. Parameswaran, V. & Ermin, Plaetschke. Flight path reconstruction using extended Kalman filtering techniques. 1990, DLR- FB 90-41, also as NAL PD FC 9103.
10. Evans, R.J.; Goodwin, G.C.; Feik, R.A.; Martin, C. & Lozano-Leal, R. Aircraft flight data compatibility checking using maximum likelihood

and extended Kalman filter estimation. *In* Proc. 7th IFAC Symposium, Identification and System Parameter Estimation, 1985, York, UK. pp. 487-92.

11. Candy, J.V. Signal processing, model based approach. McGraw-Hill Book Co., New York, 1986.

Appendix 1

```
% function [a, b. h, d, x0, uc] = lisimdat (N)
% data file for simulation of linear data-carel
% a, b. c, d are system matrices
% x0 - initial condition an states. uc - doublet input used for % simulation
function [a, b, c, d, x0, uc] = lisimdat (N)
a = [- 0.753088 1.0 ;
- 1.37662 - 1.11833] ;
b = [0.0 ;
    - 2.4903] ;
c = [1 0 ;
   0 1 ;
   6.044 0] ;
d = [0; 0; 0] ; ku = .1;
x0 = [0.0 0.0] ;
uz =          ;
up = ku * ones (1:30) ;
un = - ku * ones (1:30) ;
NN = N-65 ;
uzl = zeros (1:NN) ;
uc = [uz un up uzl] ;
% END of file

% data file for UD filtering of linear data-case1
% function [u, nx, ns, nw, ny, xte] = lifildat
% u is the upper triangular matrix for filtering
% nx = total No. of augmented states
% ns = No. of states. nw = order of the measurement noise matrix
   ny = No. of observables, xte = initial values for filtering
function [u, nx, ns, nw, ny, xte] = fildatal

nx = 2 ;
ny = 3 ; ns = 2 ; nw = 2 ;

% END of file

% function [ud, q, cr, nx, ny, nw, np,      = gkfildat
% data file for simulation of non linear data-case II
% u is the upper triangular matrix for filtering
% nx = total No. of augmented states
% nw = order of the measurement noise matrix
% ny = No. of observables, xte = initial values for filtering
% np = No. of parameters
   q = process noise covariance matrix ;
% cr = measurement noise covariance matrix ;
                                  np,     =

   = [208.0 18.0 0.15                  7.0      1.0]


      for i = 1 : nx
            ud (i, i) = uu (i) ;
      end
gmat = eye (nw) ;
zs =          nw) ;
gmat = [gmat ;     ;
ud = [ud gmat] ;
qdiag = [0.4392 5.5078 .0008]; q = eye (ns) ;
      for i = 1 : ns
            q(i,i) = qdiag (i);
      end
cr = [0.5870 0 0;0 0.0002 0;0 0
% END of FILE


% Initial data for filtering realdata
% u is the upper triangular matrix for filtering
% nx = total No. of augmented states
% nw order of the measurement noise matrix
% ny No. of observables, xte = initial values for filtering
% np No. of parameten
% q = process noise covariance matrix;
% cr = measurement noise. covariance matrix;
% fuction [ud, q, r, nx, ny, xte] = realdat
function [ud, q, cr, nx, ny, nw, np,      = realdat
nx = 15 ; nw = 4 ; ns = 4 ; np
ny = 6 ;
   =          .0017 103.6        - .66 .42 - 5.50 .041 .0712    1.02
- 21.78 - 1.23 .204    .21      ;
ud =     (nx) ;
uu = [0.01 0.0001 1.0 0.01 .0001          0.0841
     0.0196 .0001                ;
      for i =    : nx
            ud (i, i) = uu (i) ;
      end
gmat = eye (nw) ;
zs = zeros (np, nw) ;
gmat = [gmat ;


         for i =
               (i, i) = qdiag (i);
         end
rdiag = [4.0d-6 1.6d-9 9.0d-2 1.6d-4 1.0d-9 1.0d-9    ;
cr = eye (ny) ;
         for i = 1 : ny
            cr (i, i) = rdiag     ;
         end
% END OF FILE
```