FIDA

A Software Tool for Parameter Estimation from Flight Test Data

Shantha Kumar N

Flight Mechanics and Control Division

National Aerospace Laboratories

Bangalore 560 017

Abstract

A software package called FIDA is developed and implemented in PC MATLAB for estimating aircraft stability and control derivatives from flight test data using different system identification techniques. FIDA also contains data pre-processing tools to remove wild points and high frequency noise components from measured flight data. FIDA is a menu driven and user interactive software which is useful to scientists/flight test engineers/pilots who are engaged in experimental flights and analysis of flight test data. Also it has an educational value for students and practising engineers who are new to the field of aircraft parameter estimation.

1. Introduction

Parameters (**stability and control derivatives**) estimation from flight test data of atmospheric vehicles is an essential step in building their mathematical model, because it helps in

- verifying theoretical and wind tunnel model predictions

- upgrading of simulation model

- design of stability augmentation and flight control systems

- fault detection and diagnosis.

The flight test data for parameter estimation are generally acquired by carefully planned and conducted flight test maneuvers on the vehicles in order to derive maximum information on the characteristics of the vehicle. The flight data thus acquired generally contains data spikes and high frequency noise and requires pre-processing to eliminate them before using the data for parameter estimation.

In order to aid an analyst in the above task, a <u>fl</u>ight <u>d</u>ata <u>a</u>nalysis software package called FIDA is developed in PC MATLAB. The package consists of flight data pre-processing and different parameter estimation techniques:

1. Generalised Least Square - Differential Correction (GLSDC).

2. Output Error Minimization (OEM)

3. Extended Kalman Filtering (EKF).

4. V-Λ (Lambda) Factorization based Extended Kalman Filtering (VLEKF).

FIDA is menu driven and user interactive. The user can select any one or more techniques to perform parameter estimation.

In this paper, the details of flight data pre-processing and different parameter estimation techniques that are in FIDA are presented. FIDA is initially validated with simulated data and

then applied to flight test data of NAL's Light Canard Research Aircraft (LCRA). The results are presented and compared with the corresponding results from other identification techniques.

2. Pre-processing of Flight Test Data

Flight test data generally contains noise and spikes due to: the noise generated by the measuring instruments, the vibration picked up by the sensors, the man made interference in the measurement and recording and due to the interference from other systems. This raw data as such is not suitable to be used for subsequent analysis like parameter estimation. Hence there is a need for pre-processing of flight data to remove data spikes and noise before data is subjected to parameter estimation process. As part of FIDA two MATLAB program modules have been developed for data pre-processing and these programs can be executed by choosing appropriate menu in FIDA.

Spike Removal: A simple procedure of cut and replace is adopted for spike removal. In this procedure (called **Editing**), the spike (or bunch of spikes if they are closely located) is first identified by viewing the time history plot. From mouse pointer the small segment of data containing spike(s) is removed and replaced by interpolated data using few points on either side of removed portion. Also user can select a small time segment of data around wild point and expand into bigger plot (i.e. zooming), in case it is difficult to identify the wild point(s). Fig 1 illustrates the spike removal process. First plot shows a recorded flight variable which contains noise and spikes. The small segment of data is expanded (by pressing Z key and selecting points S1 and S2 from mouse pointer) into bigger plot as shown in second plot. Then user is required to select two points on either side of wild point (points A,B,C,D in second plot) using mouse pointer. Between points B and C, the data will be eliminated and

new data is inserted through interpolation by taking data between A to B and C to D. Last plot shows the edited data. Cubic spline function available in MATLAB is utilized for the interpolation. The program is self explanatory and user can perform spike removal through on line instruction.

Noise Filtering: The Fast Fourier Transform (FFT) method is used for noise filtering from the flight data since FFT function is directly available in MATLAB and it gives spectrum from which user can conveniently select a pass band. The program first generates spectrum plot of a flight data trajectory after FFT and user should select frequency limits to be retained (pass band) through mouse pointer on the spectrum plot. This makes the Fourier coefficients of unwanted frequencies set to zero and then the data is transformed back into time domain. While selecting the frequency limits, the user should take care to see that pass band contains all the natural modes of aircraft dynamics. Fig 2 illustrates the process of noise filtering.

3. Data Compatibility Check

The flight test data recorded from sensors are generally prone to bias and scale factor errors. Hence the data compatibility check is an essential step before the flight data is used for parameter estimation. And it also determines the degree of confidence in the recorded flight data. The data compatibility check which is popularly known as flight path reconstruction utilises the redundancy present in the recorded inertial and air data variables to obtain the best estimate of states together with scale factors and bias errors in the measured signals.

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 output variables are the flight path velocity, V, angle of

attack, α, and the sideslip angle, β. Filtering techniques like Extended Kalman filtering (EKF) and V-Λ(Lambda) factorization based EKF which are described in the following sections can be conveniently used for data compatibility check.

4. Aircraft Equations of Motion and Parameter Estimation

The aircraft equations of motion are based on classical Newtonian mechanics and consists of kinematics, gravitational and aerodynamic terms. The kinematic and gravitational terms are easy to quantify. But to quantify the aerodynamic force and moment, acting on aircraft, it is required to know/estimate stability and control derivatives.

The mathematical form of each aerodynamic force and moment acting on the aircraft is assumed to be

$$A_f = K \bar{q} C_f \qquad (1)$$

where A_f is the aerodynamic force or moment, K is a constant involving aircraft reference quantities like mass, inertia, wing area, mean aerodynamic chord etc., $\bar{q}$ is the dynamic pressure and C_f is a non-dimensional force or moment coefficient which is further assumed to be in the form:

$$C_f = C_{f_o} + C_{fa} a + C_{fb} b + C_{fc} c + - - - - - - \qquad (2)$$

where C_{f_o}, C_{fa}, C_{fb}, C_{fc}, etc. are the stability and control derivatives and a,b,c, etc. are simple functions of states or control inputs.

One of the most popular forms of representing aircraft equations of motion in the time domain is state space form[1]. The equations of motion for a rigid body aircraft are a set of nonlinear differential equations

$$\dot{X} = f \left[X(t), U_m(t), \overline{\Gamma} \right] + w(t) \qquad (3)$$

where, state $X(t)$ generally consists of linear velocity components u, v and w along aircraft fixed body axes, the angular velocity components p, q and r about same body axes, and the Euler angles ϕ, θ, φ which describe the orientation of body axis frame w.r.t earth fixed frame of reference. The control input $U_m(t)$ generally includes control surface deflections δ_e(elevator), δ_a(aileron), δ_r(rudder) and δ_f(flaps) and possibly thrust setting/thrust vectoring related parameters. Γ is the vector of stability and control derivatives. And the vector $w(t)$ represents process noise e.g. atmospheric turbulence.

An output equation representing the discrete time measurement can be expressed as

$$Y = h \left[X(t), U_m(t), \Gamma, \overline{\Omega} \right] + v(t) \qquad (4)$$

where, the output vector Y typically consists of flight path velocity V, the angle of attack, α, the angle of side slip, β, the angular velocities, p, q and r, the Euler angles ϕ, θ and φ and possibly linear acceleration a_x, a_y and a_z along the body axes. Ω is again the vector of unknown parameters in the observation equations. And the vector $v(t)$ represents the noise present in the measured signals.

Then the problem of aircraft parameter identification is to estimate the unknown parameter vector

$$\hat{\Theta} = \left[X_0^T \ \Gamma^T \ \Omega^T \right]^T \qquad (5)$$

where X_0 is a vector of initial conditions of states which are unknown.

The simple approach in the identification of the aircraft stability and control derivatives is to estimate the coefficients of the linearized state and observation equations[1,2]. This will result in the estimation of dimensional derivatives which contain the dynamic pressure $\bar{q}$. But

when airspeed significantly changes during the flight, one has to use non-dimensional form of derivatives in the model with dynamic pressure $\bar{q}$ as a variable.

5. System Identification Techniques

FIDA is a software tool which estimates the aircraft stability and control derivatives directly from the flight test data. FIDA contains four different parameter estimation techniques and user can select the required technique for a particular application. The following section gives the details of each of the technique.

5.1 Generalized Least Square Differential Correction[3] (GLSDC)

It is a batch -iterative procedure suitable for parameter estimation when there is no process noise. It can be regarded as complementary to popular nonlinear parameter estimation method namely maximum likelihood (ML) method. It can also be used as a startup procedure to estimate initial parameter values which are subsequently required for other parameter estimation methods.

The differential correction algorithm is applied to obtain $\hat{\Theta}$ from the measured noisy data as

$$\hat{\Theta}_{(i+1)} = \hat{\Theta}_{(i)} + \left[\left(F^T w F \right)^{-1} F^T w \Delta Y \right]_{(i)} \qquad (6)$$

where $\quad F = \left[\delta Y / \delta X_0 \mid \delta Y / \delta \Gamma \mid \delta Y / \delta \Omega \right]$

w is weighting matrix

ΔY is a vector of residuals

$\Delta Y = Z_m ( ) - Y ( ) \qquad$ k=1,2,...,N (N is the total no. of data points).

Z_m is a vector of measured signals.

The first submatrix of F is given as

$$\delta Y(t_i)/\delta X(t_0) \;=\; \left[\delta h(X(t_i))/\delta X(t_i)\right]\left[\delta X(t_i)/\delta X(t_0)\right] \qquad (7)$$

with $\quad \dfrac{d}{dt}\left[\delta X(t)/\delta X(t_0)\right] = \left[\delta f(t,X(t))/\delta X(t)\right]\left[\delta X(t)/\delta X(t_0)\right]$ (8)

The above transition matrix differential equation can be solved with identity matrix to start with as shown below.

$$\left[\Phi(t,\,t_0)\right] = \left[\delta X(t)/\delta X(t_0)\right] = I + \int_{t_0}^{t}\left[\delta f/\delta X(\tau)\right]\left[\delta X(\tau)/\delta X(t_0)\right]d\tau \qquad (9)$$

$$\dfrac{d}{dt}\left[\Phi(t,\,t_0)\right] = \left[I + F_1\,\Delta t + \dfrac{F_2\,\Delta t^2}{2!} + \;-\;-\;-\;-\;-\; \right](10)$$

The second submatrix in F is given by,

$$\delta Y/\delta\Gamma \;=\; \left(\delta h/\delta X\right)\left(\delta X/\delta\Gamma\right) + \delta h/\delta\Gamma \qquad (11)$$

where $\delta X(t)/\delta\Gamma$ is the solution of

$$\dfrac{d}{dt}\left(\delta X/\delta\Gamma\right) = \delta f/\delta\Gamma + \left(\delta f/\delta X\right)\left(\delta X/\delta\Gamma\right) \quad (12)$$

The equation (12) is solved in the following manner,

$$\left[\Psi(t,\,t_0)\right] = \left[\delta X(t)/\delta\Gamma\right] = I + \int_{t_0}^{t}\left[\delta f/\delta X(\tau)\;\;\delta X(\tau)/\delta\Gamma + \delta f/\delta\Gamma\right]d\tau$$

$$\qquad\qquad (13)$$

$$= \left[I + F_1\,\Delta t + \dfrac{F_2\,\Delta t}{2!} + \;-\;-\;-\;-\; \right]\left[\Psi(t,\,t_0)\right] + F_\Gamma\,\Delta t$$

where $\Psi(t,\,t_0)$ is zero matrix to start with.

The last submatrix in F is obtained as

$$\delta y/\delta\Omega \;=\; \delta h/\delta\Omega \qquad\qquad (14)$$

The partial differentiation operation are performed numerically by using finite difference method and the state integration is performed by numerical integration namely fourth order Range-Kutta method

5.2 Output Error Minimization Technique (OEM)

The OEM minimises the error between the measured and model responses produced by identical inputs. It is assumed that there is no process noise. The OEM is applicable to both linear and non-linear systems[4]. To estimate the parameters Θ, the cost function to be minimised is defined as

$$E(\Theta) \; = \; \frac{1}{2} \; \sum_{k=1}^{N} \left[Z_{m(k)} - Y_{(k)} \right]^{T} R^{-1} \left[Z_{m(k)} - Y_{(k)} \right] \; + \; \frac{N}{2} \; \ln|R| \qquad (15)$$

Minimisation of the above cost function w.r.t. Θ yields the estimates of Θ as:

$$\hat{\Theta}_{i+1} \; = \; \hat{\Theta}_{i} \; + \; \mu \, \Delta\Theta_{i} \qquad (16)$$

where subscript i represents iterations, the constant μ is called the damping factor which can be used to improve the convergence of the algorithm, and

$$\Delta\Theta_{i} = \left\{ \sum_{k} \left(\frac{\partial Y_{(k)}}{\partial\Theta} \right)^{T} R^{-1} \left(\frac{\partial Y_{(k)}}{\partial\Theta} \right) \right\}^{-1} \left\{ \sum_{k} \left(\frac{\partial Y_{(k)}}{\partial\Theta} \right)^{T} R^{-1} (Z_{m(k)} - Y_{(k)}) \right\} \qquad (17)$$

The first term in equation (17) is the Gauss Newton approximation to the second gradient of the cost function $E(\Theta)$ and is called the information matrix. Equation (16) in terms of the first and second gradients can be written as

$$\hat{\Theta}_{i+1} \; = \; \hat{\Theta}_{i} \; + \; \nabla^{2}E(\Theta) \; \nabla E(\Theta) \qquad (18)$$

Thus, to compute the first and second gradients, we need to compute the term $\dfrac{\partial Y_{(k)}}{\partial\Theta}$. This term is called the sensitivity matrix and is computed using transition matrix method (as in GLSDC).

5.3 Extended Kalman Filter

The extended Kalman filter (EKF) is a recursive algorithm that estimates system states and state error covariance by using the measured system output Z_m and the known input U_m in the specified nonlinear system and output models with assumed process and measurement noise statistics. With the linear model, the EKF is reduced to well known Kalman filter which gives optimal solution. The EKF gives approximate solution to nonlinear problem and the solution involves the linearization of the nonlinear process about each new estimate as soon as it becomes available, so that large initial estimation errors do not propagate. Simultaneous estimation of the states and unknown parameters (stability and control derivatives) of the system is possible by augmenting the state vector with the unknown parameters and applying filtering algorithm to the augmented nonlinear model. EKF has been successfully applied to flight path reconstruction problem[5].

In EKF, the system represented by equations 3 and 4 are first linearized about the prior best estimates of the states at each instant of time by finite difference method as follows

$$A_{(k)} = \left[\frac{\delta f}{\delta X}\right]_{X_{(k)}, U_{m(k)}} \quad ; \quad B_{(k)} = \left[\frac{\delta f}{\delta U_m}\right]_{X_{(k)}, U_{m(k)}} \quad ; \quad C_{(k)} = \left[\frac{\delta h}{\delta X}\right]_{X_{(k)}, U_{m(k)}} \quad (19)$$

The linearized system is then discretised in time by computing the system state transition matrix $\Phi_{(k+1,k)}$ from $A_{(k)}$ as follows

$$\Phi_{(k+1,k)} = e^{A \Delta t} \quad (20)$$

where Δt is the sampling time.

The EKF consists time update (prediction step) and measurement update (correction step). In time update, the time propagation of states and state error covariance matrix are obtained by the following equations,

$$\tilde{X}_{(k+1)} = \hat{X}_{(k)} + \int f\left[X(t), U_m(t)\right] dt \quad (21)$$

$$\tilde{P}_{(k+1)} = \Phi_{(k+1,k)} \hat{P}_{(k)} \Phi^T_{(k+1,k)} + G_{A(k)} Q G^T_{A(k)} \quad (22)$$

In correction step, the measurement update of states and state error covariance matrix are obtained by using output measurement as and when they are available as follows,

$$\hat{X}_{(k+1)} = \tilde{X}_{(k+1)} + K_{(k+1)}\left[Z_{m(k+1)} - h\{\tilde{X}_{(k+1)}\}\right] \quad (23)$$

$$\hat{P}_{(k+1)} = \left[I - K_{(k+1)} C_{(k+1)}\right] \tilde{P}_{(k+1)} \quad (24)$$

where $K_{(k+1)}$ is the Kalman gain at instant k+1 which is given by

$$K_{(k+1)} = \tilde{P}_{(k+1)} C^T_{(k+1)} \left[C_{(k+1)} \tilde{P}_{(k+1)} C^T_{(k+1)} + R\right]^{-1} \quad (25)$$

The state integration (equation 21) is done by fourth order Runge-Kutta method.

5.4 V-Λ Factorization Based Extended Kalman Filtering

Although Kalman filter has been applied to many practical systems with varying degree of success, the main problem has been with the P equations (equation 24), which is numerically ill conditioned[6]. This is mainly due to the round off in computation which makes covariance matrix non positive definite where as it must be atleast semi positive definite. This problem can be minimized to some extent by expressing P equation in longer form as shown below[5],

$$\hat{P}_{(k+1)} = \left[I - K_{(k+1)} C_{(k+1)}\right] \tilde{P}_{(k+1)} \left[I - K_{(k+1)} C_{(k+1)}\right]^T + K_{(k+1)} R K^T_{(k+1)} \quad (26)$$

However algorithm which involves square root or factorization of the covariance matrix has better numerical properties[6]. In FIDA a V-Λ factorization based Kalman filter is implemented in which the state error covariance matrix P is spectral decomposed as[7]

$$P = V \Lambda V^T \quad (27)$$

where V is the matrix whose columns are the eigen vectors of P and Λ is the diagonal eigen value matrix. The discrete time version of the V-Λ filter are based on the singular value decomposition (SVD) as a main tool, which renders them extremely numerically robust and accurate[7]. The new algorithm is of great importance in certain applications where continuous monitoring of the eigenfactors is necessary in order to reveal singularities as and when they occur. The SVD function available in MATLAB is computationally efficient and the same is used in filter algorithm.

In this filter algorithm, the time update of state and V-Λ factors are computed as follows

$$\tilde{X}_{(k+1)} = \hat{X}_{(k)} + \int f\left[X(t), U_m(t)\right] dt,.$$

$$\tilde{V}_{(k+1)} \left[\tilde{\Lambda}^{\frac{1}{2}}_{(K+1)} \quad \Big| \quad 0 \right] Z^T_{(k+1)} = SVD(A_{ug}) \quad (28)$$

where

$$A_{ug} = \left[\Phi_{(k+1,k)} \, \hat{V}_{(k)} \, \hat{\Lambda}^{\frac{1}{2}}_{(k)} \quad \Big| \quad G_{(k)} \, Q_{(k)} \right] \quad (29)$$

The $\Phi_{(k+1,k)}$ is computed as in equation (19) and $A_{(k)}$ as in equation (20).

The measurement update of V-Λ factors are computed as

$$\hat{V}_{(k+1)} \left[\hat{\Lambda}^{-\frac{1}{2}}_{(K+1)} \quad 0 \right] Z^T_{(k+1)} = SVD(A_{ug}) \quad (30)$$

where

$$A_{ug} = \left[\tilde{V}_{(k+1)} \, \tilde{\Lambda}^{-\frac{1}{2}}_{(k+1)} \quad C_{(k+1)} \, R^{-\frac{T}{2}}_{(k+1)} \right] \quad (31)$$

The $C_{(k+1)}$ is computed as in (20).

Also the measurement update of states are computed as

$$\hat{X}_{(k+1)} = \tilde{X}_{(k+1)} + Kg_{(k+1)} \left[Z_{m(k+1)} - h\{ \tilde{X}_{(k+1)} \} \right] \quad (32)$$

where the Kalman gain Kg is computed as

$$Kg_{(k+1)} = \hat{V}_{(k+1)} \hat{\Lambda}^{\frac{1}{2}}_{(k+1)} \left[\left(\hat{V}_{(k+1)} \hat{\Lambda}^{\frac{1}{2}}_{(k+1)} \right)^{T} C^{T}_{(k+1)} \right] R^{-1} \quad (33)$$

6. Case Study

This section illustrates how FIDA can be used to estimate aircraft stability and control derivatives from flight test data. For this purpose flight test data of NAL's Light Canard Research Aircraft (LCRA) which was generated in late 1991 were utilised. Two sets of LCRA flight test data generated at the same flight condition (Altitude = 8000 feet and Mach no. = 0.16) were analysed to examine the consistency in estimating stability and control derivatives. The time histories of important signals from these flight test data are shown in figure 3. Since forward acceleration (a_x) was not recorded during the flight test, the kinematic consistency check was not feasible. Hence directly parameter estimation was carried out using different system identification techniques from FIDA.

6.1 Mathematical Model

For parameter estimation, 5.4 seconds data from first set of flight data and 7.5 seconds data from second set of flight data shown in figure 3 were selected where longitudinal mode (short period) was found to be present due to elevator input. The following simple second order longitudinal model was used to fit the flight data and estimate L_α, L_q, L_{δ_e}, M_α, M_q and M_{δ_e}.

State Equations:

$$\dot{\alpha} = -\left[L_\alpha \alpha + L_q q + L_{\delta_e} \delta_e \right] + q$$

$$\dot{q} = M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e$$

$$\dot{q}_l = \frac{1}{\tau_1}\left[q - q_l \right]$$

$$\dot{a}_{n_l} = \frac{1}{\tau_2}\left[\{ \dot{\alpha} - q \} U_0 + X_{a_n} \dot{q} + Z_{a_n} \{ q^2 + p_m^2 \} - a_{n_l} \right]$$

Measurement Equations:

$$\alpha_m = \left[\alpha - \frac{X_\alpha}{U_0} \right] + \Delta\alpha$$

$$q_m = K_q q_l + \Delta q$$

$$a_{n_m} = a_{n_l} + \Delta a_n$$

Two extra state equations $\dot{q}_l$ and $\dot{a}_{n_l}$ were used here to correct the data for time shifts[1] that were observed in measured pitch rate and normal acceleration signals. Since there is substantial roll rate during maneuver (as seen from figure 3), the measured roll rate signal p_m is included as input to the model. Along with stability and control derivatives, bias ($\Delta\alpha$, Δq and Δa_n in the respective signals) and scale factor (K_q in q signal) which were found to be present in the measured signals were estimated. Following constants and position correction for sensors were included in the model:

$U_0 = 53.65$ m/s

$X_\alpha = 3.24$ m

$X_{a_n} = 0.64$ m

$Z_{a_n} = 0.313$ m

6.2 Parameter Estimation

Using different parameter estimation techniques each at a time from FIDA, the above mentioned stability and control derivatives were estimated and compared with corresponding estimates[8] from maximum likelihood estimation technique (MLE) which is being successfully using at NAL,FMCD for the last 10 years. MLE is not part of the present FIDA package, however, the OEM technique is nearly equivalent to MLE. The Table 1 shows the comparison. Since some convergence problem were encountered in the estimation of L_q and L_{δ_e}, their values were fixed at corresponding known reference values. This convergence problem may be due to insufficient information corresponding to those derivatives in the flight test data. The estimated derivatives are consistent from all the techniques (for both the sets of data). Also from the estimates it is clear that there is significant bias in all the output signals and scale factor in measured pitch rate. The negative scale factor in pitch rate implies that the measured pitch rate signal is in reverse polarity. Also estimated $\frac{1}{\tau_1}$ and $\frac{1}{\tau_2}$ indicates that there is time shift in measured pitch rate and normal acceleration signals equivalent to τ_1 and τ_2 seconds respectively. This time shift in measured signals may be due to delay in sensor response and/or due to human error during time tagging of different measured signals.

The short period natural frequency (ω_{sp}) and damping ratio (ζ_{sp}) are computed from the estimated derivatives as follows[9]:

$$\omega_{sp} = \sqrt{\left(M_\alpha - M_q L_\alpha\right)}$$

$$\zeta_{sp} = \frac{-\left(M_q - L_\alpha\right)}{2\,\omega_{sp}}$$

Typical time history plots from the different parameter estimation techniques are shown in figures 4 and 5.

7. Concluding Remarks

A software package called FIDA is developed and implemented in PC MATLAB for estimating aircraft stability and control derivatives from flight test data using different system identification techniques. FIDA also contains data pre-processing tools to remove wild points and high frequency noise components from measured flight data. FIDA is a menu driven and user interactive software which is useful to scientists/flight test engineers/pilots who are engaged in experimental flights and analysis of flight test data. Also it has an educational value for students and practising engineers who are new to the field of aircraft parameter estimation. FIDA is initially validated with simulated data and demonstrated its capability of estimating stability and control derivatives from actual flight test data. The package has a wider applicability to many general dynamical systems. Commercialisation of this package can be discussed based on the potential users enquiries and requirement.

8. Why MATLAB ?

Most of the routines in this package were originally developed in FORTRAN and to use these programs, some skill/experience was needed. In order to make the package commercially viable, it was required to make it menu driven and user friendly with interactive facility. Hence it was decided to implement the whole code in **MATLAB** which is very user friendly software tool. MATLAB is being used for flight data analysis and parameter estimation since five years and we at FMCD,NAL feel that it is very useful and easy to use tool for development purposes.

9. References

1. Maine R.E. and Iliff K.W., Application of Parameter Estimation to Aircraft Stability and Control-Output Error Approach., NASA RP 1168, 1986.

2. Iliff K.W., Parameter Estimation for Flight Vehicles., J. Guidance., Vol. 12, No. 5, Sept-Oct 1989.

3. Rekha A.N., Parameswaran V. and Raol J.R., PC based Parameter Estimation Algorithm using Generalised Least Squares Differential Correction Method (GLSDC). NAL PD FC 9314, Oct 1993.

4 Girija G, Parameswaran V, Raol J.R and Srinathkumar S., Estimation of Lift and Drag characteristics of an aircraft from flight data., Journal of the Aeronautical Society of India, Vol. 43, No. 4, Nov. 1991

5. Parameswaran V. and Plaetschke E., Flight Path Reconstruction using Extended Kalman Filtering Technique., DLR-IB-90-41, 1990.

6. Bierman G.J., Factorization Methods for Discrete Sequential Estimation. Academic Press, New York, 1977.

7. Oshman Y. and Bar-Itzhack I.Y., Square Root Filtering via Covariance and Information Eigenfactors., Automatica, Vol. 22, No. 5, pp 599-604, 1986.

8.   Shantha Kumar N and Raol J R, Flight Data Analysis of LCRA - Longitudinal Case., NAL interim report, August 1992.

9. McRuer D.T., Ashkenas J and Graham D., Aircraft Dynamics and Automatic Control., Princeton University Press, Princeton, N.J., U.S.A., 1978.

9. Acknowledgement

10. Tables

Table 1: Comparison of Estimated Parameters (I set of LCRA Flight Data)

Parameters	MLE[8]	GLSDC	OEM	EKF	VLEKF
L_α (/rad)	2.4000	2.4000	2.4008	2.4513	2.4513
L_q (/rad/sec)	0.0259*	0.0259*	0.0259*	0.0259*	0.0259*
L_δ (/rad)	0.1150*	0.1150*	0.1150*	0.1150*	0.1150*
M_α (/rad)	-14.4000	-14.5169	-14.5320	-14.5234	-14.5234
M_q (/rad/sec)	-0.6265	-0.6744	-0.6637	-0.5721	-0.5721
M_δ (/rad)	12.2560	12.4137	12.3646	11.6408	11.6408
$1/\tau_1$	25.0000	27.5812	27.7697	24.8467	24.8467
$1/\tau_2$	8.3700	8.5556	8.5653	8.0540	8.0540
K_q	-0.9268	-0.9254	-0.9259	-0.9788	-0.9788
$\Delta\alpha$ (rad)	0.0410	0.0410	0.0411	0.0406	0.0406
Δq (rad/sec)	0.0900	0.0900	0.0899	0.0900	0.0900
Δa_n (mtr/sec^2)	4.2660	4.2798	4.2855	4.3468	4.3468
ω_{sp} (rad/sec)	3.9883	4.0169	4.0138	3.9907	3.9907
ζ_{sp}	0.3811	0.3827	0.3817	0.3788	0.3788

Table 2: Comparison of Estimated Parameters (II set of LCRA Flight Data)

Parameters	MLE[8]	GLSDC	OEM	EKF	VLEKF
L_α (/rad)	2.1165	2.1157	2.1152	2.1435	2.1435
L_q (/rad/sec)	0.0259*	0.0259*	0.0259*	0.0259*	0.0259*
L_δ (/rad)	0.1150*	0.1150*	0.1150*	0.1150*	0.1150*
M_α (/rad)	-11.8486	-12.0423	-12.0771	-11.2217	-11.2217
M_q (/rad/sec)	-0.8414	-0.8527	-0.8530	-0.9142	-0.9142
M_δ (/rad)	11.5435	11.7069	11.7310	10.8037	10.8037
$1/\tau_1$	25.3636	27.1767	26.7084	24.9939	24.9939
$1/\tau_2$	9.4682	9.6171	9.5306	9.8848	9.8848
K_q	-0.9369	-0.9348	-0.9361	-1.0004	-1.0004
$\Delta\alpha$ (rad)	0.0400	0.0403	0.0404	0.0407	0.0407
Δq (rad/sec)	0.1014	0.1011	0.1010	0.1000	0.1000
Δa_n (mtr/sec^2)	3.7050	3.7139	3.7093	3.7679	3.7679
ω_{sp} (rad/sec)	3.6891	3.7212	3.7258	3.6306	3.6306
ζ_{sp}	0.4011	0.3990	0.3983	0.4211	0.4211

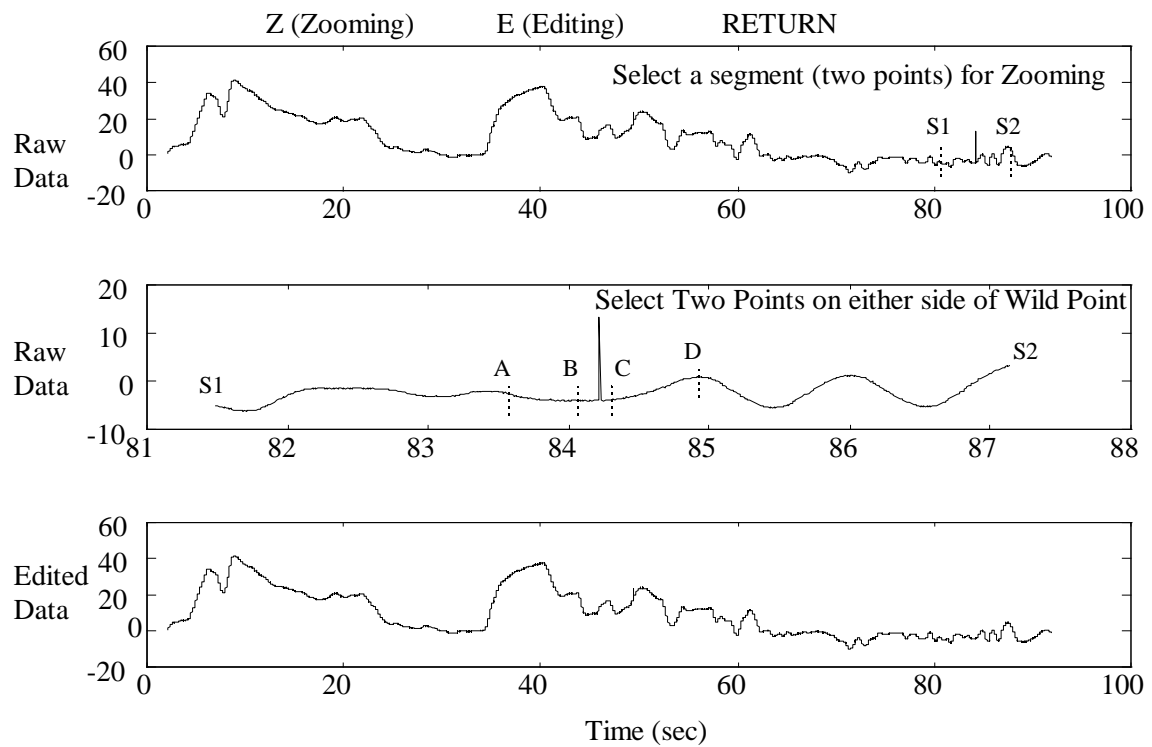* These derivatives were fixed at reference values while estimating others.
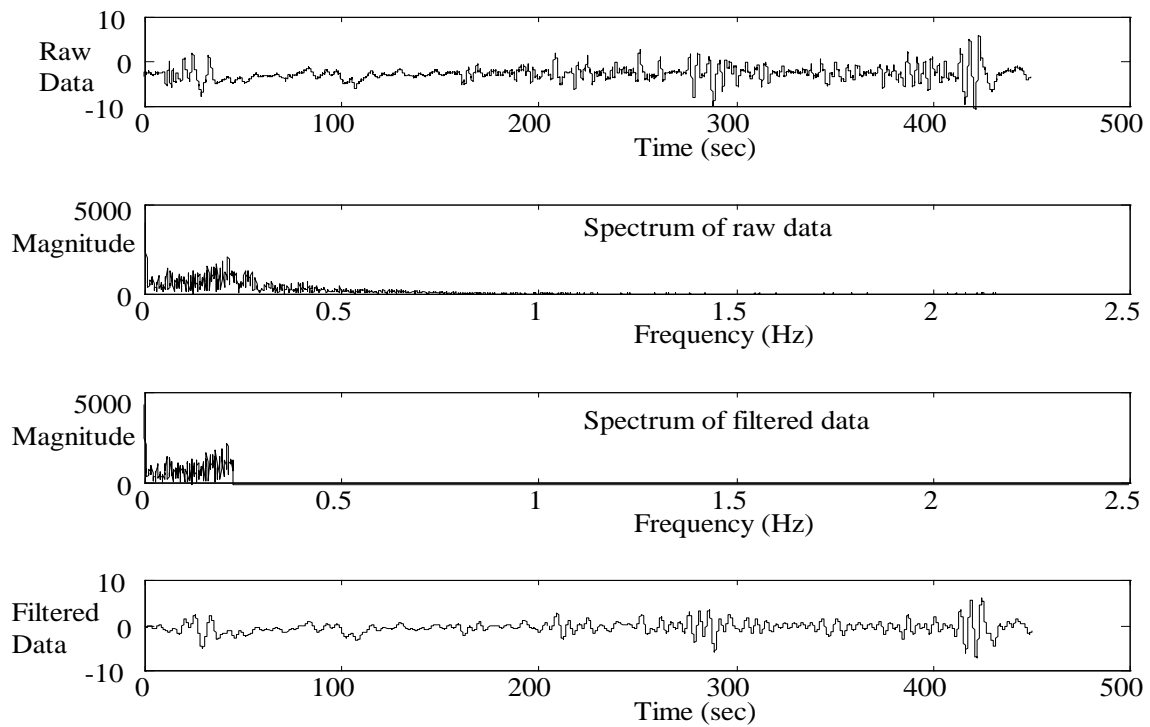
11. Figures

Figure 1: Editing Process

Figure 2: Noise Filtering Process

I set data · II set data

α_m (rad)

q_m (rad/sec)

a_{n_m} (mtr/sec^2)
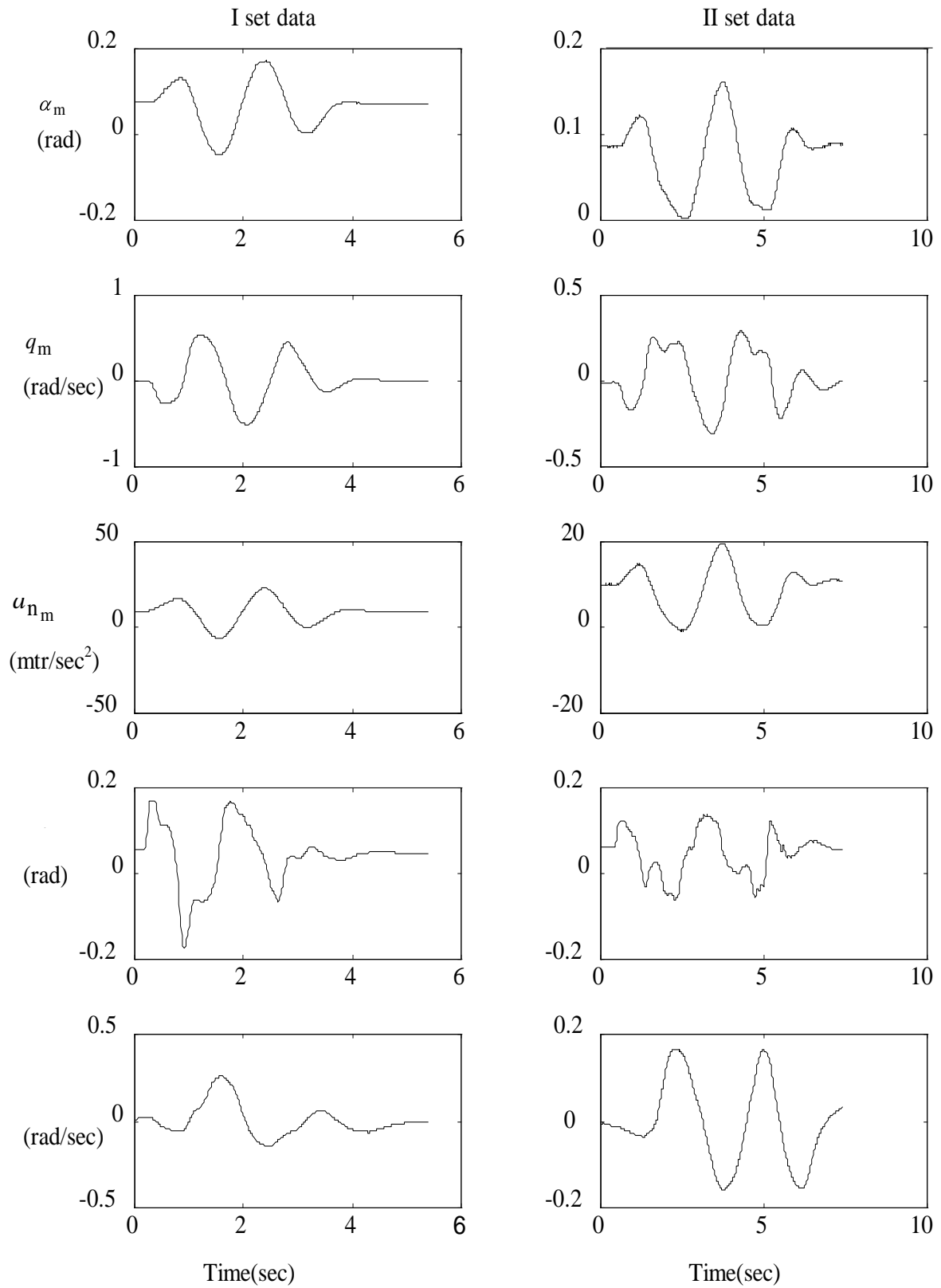
(rad)

(rad/sec)

Time(sec)
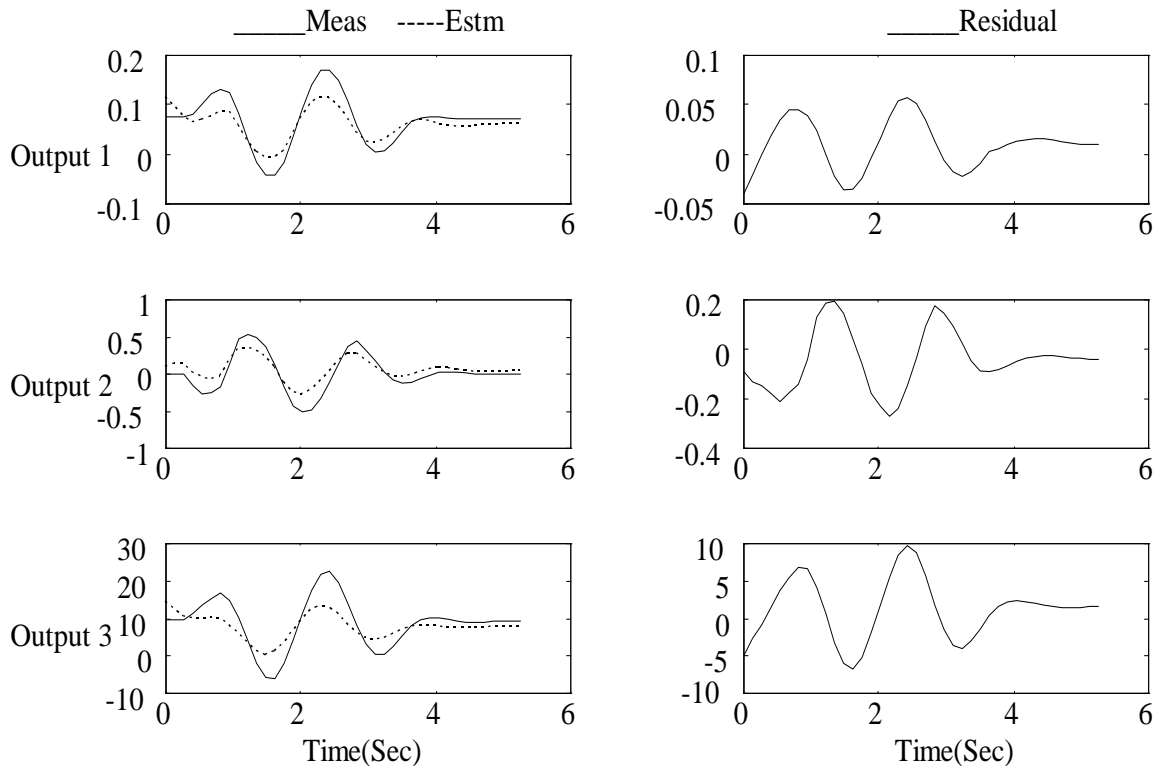
Time(sec)

Figure 3: LCRA Flight Test Data

Figure 4(a): Time history match at zeroth iteration from GLSDC and OEM (I set data)

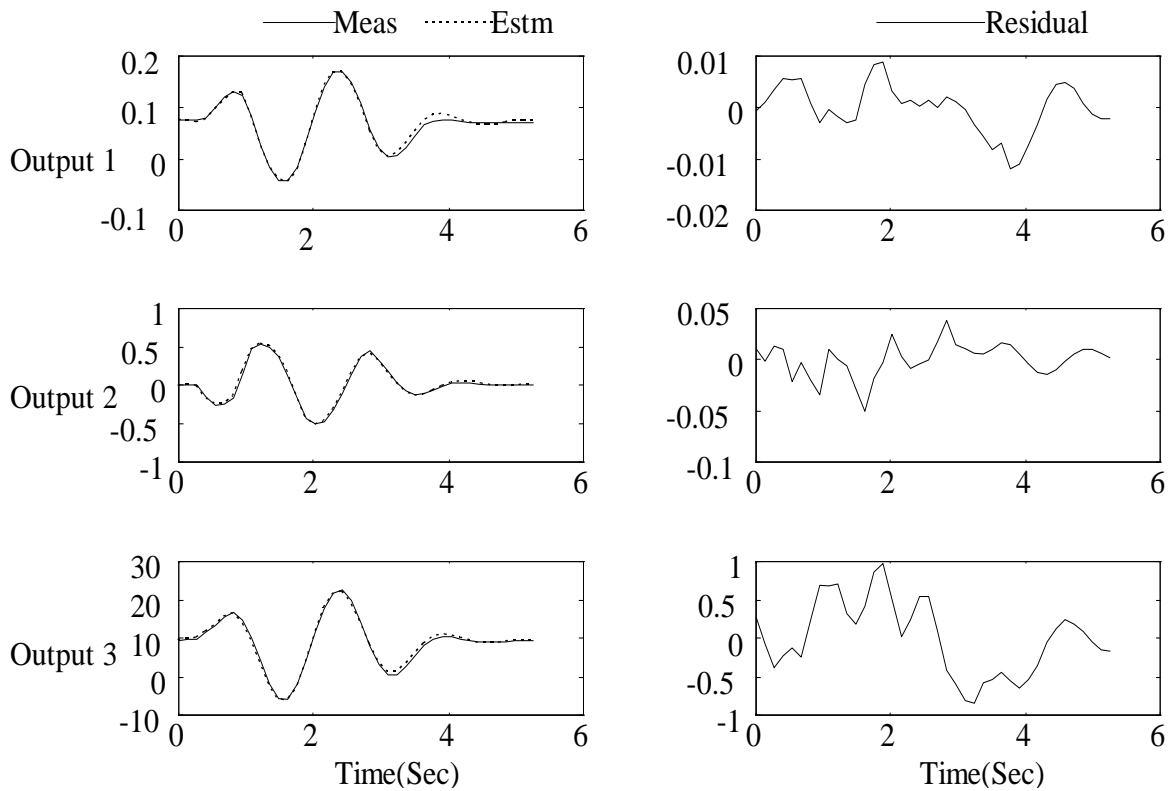Figure 4(b): Time history match after 5th iteration from GLSDC and OEM (I set data)

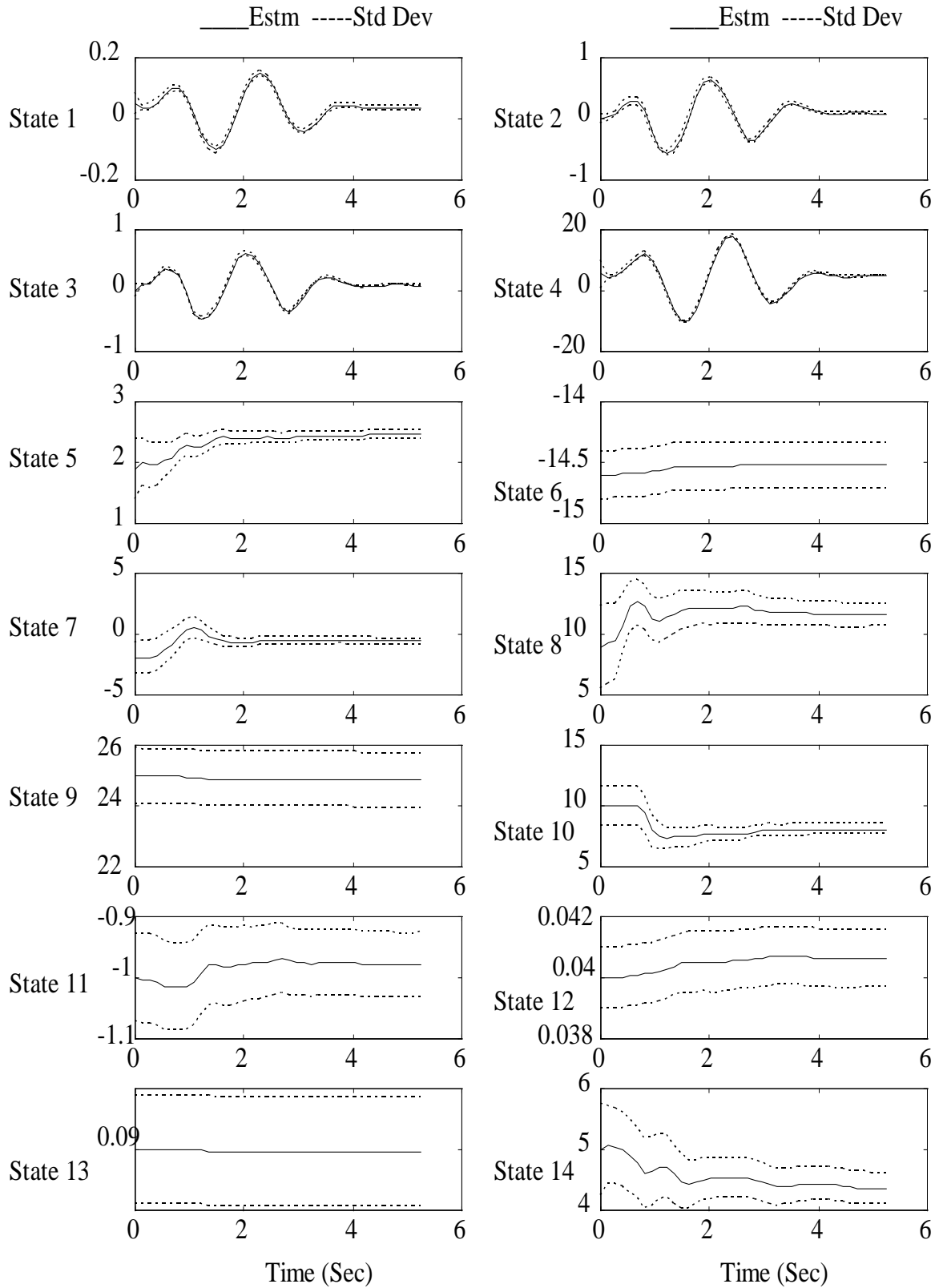MATLAB Users Conference 1996, Bangalore, INDIA.

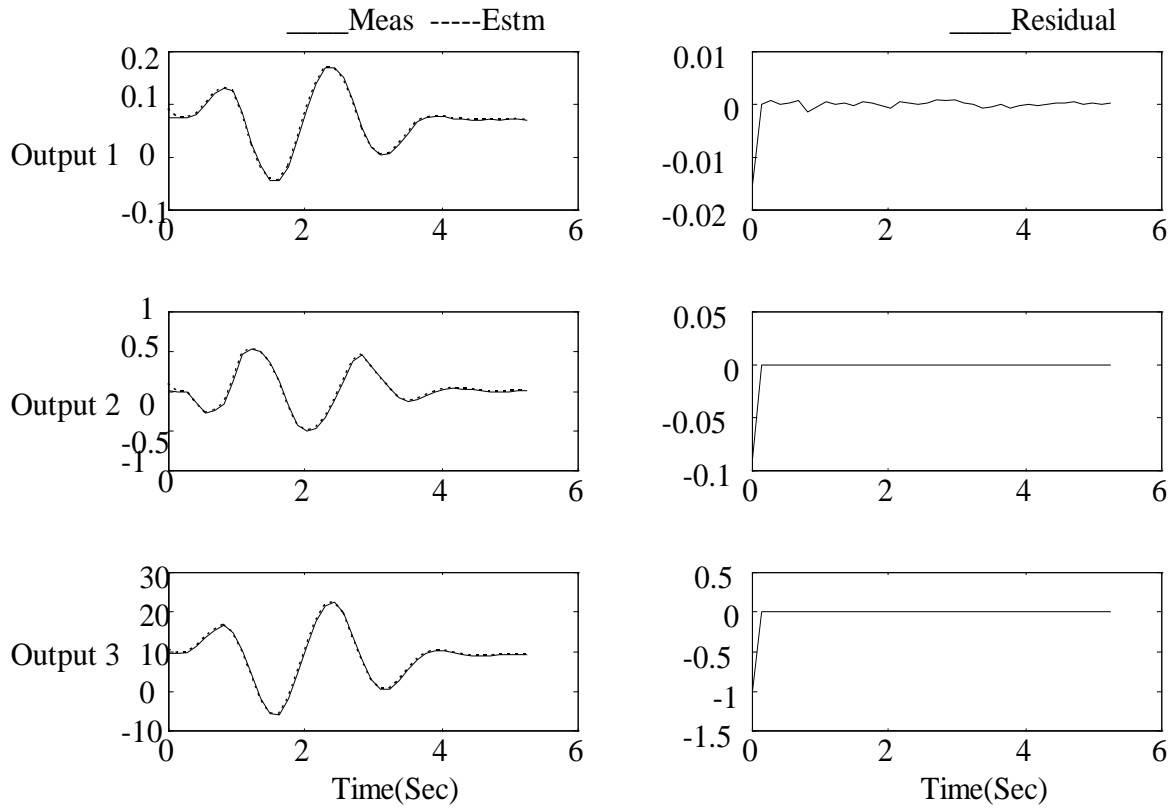Figure 5(a): Estimated States with standard deviation from EKF and VLEKF (I set data)

Figure 5(b): Output time history match from EKF and VLEKF (I set data)