IMPLEMENTATION OF REAL-TIME PARAMETER ESTIMATION IN LABVIEW AND DSP

C. Kamali*, J.R. Raol* and A.A. Pashilkar*

Abstract

This paper presents the results of real time aircraft parameter estimation technology in an aircraft real time simulator. The equation error Recursive Least Squares (RLS) method is used for performing the parameter estimation. The RLS technique is successfully adapted for on-line parameter estimation by a novel methodology that will work when measurements of state derivatives are not available. Subsequently, the algorithm is implemented in Lab VIEW and in digital signal processor (DSP) hardware to demonstrate its suitability for further implementation at flight test centers and for on-board parameter estimation for aircraft.

***Keywords**:Real-time parameter estimation, Equation error, Recursive least squares, Digital filtering*

Nomenclature

A = system matrix
B = control matrix
E{} = expectation operator
H = Filter transfer function
J = cost function
N = mtotal number of discrete observations
P = covariance matrix
t = time
p = number of unknown parameters
q = pitch rate in deg/sec
α = angle of attack in degs
β = unknown parameter vector
ε = equation error
λ = forgetting factor
δ_e = elevator control surface input
x(t) = state vector
y(t) = observation vector
Δt = sampling time
σ^2 = equation error variance

Introduction

Real-time estimation of aircraft model parameters has significant advantages for efficient flight-testing, flight envelope expansion, and real-time monitoring for flight safety. In addition, reconfigurable control techniques based on parameter estimation require real-time estimation of stability and control derivatives. The aerodynamic stability and control derivatives are estimated using flight measured input/output data in a recursive information-processing scheme [1-4]. The attractive option of using parameter estimation for gain/phase margins is covered in references [4, 5]. Flight safety monitoring through neutral point and maneuver point estimation can be accomplished through parameter estimation [6]. The real-time estimation of neutral and maneuver point can save flight test time.

There are algorithms reported in open literature for real time parameter estimation. The Kalman Filter (KF), Upper Diagonalization Extended KF (UDEKF) and the Extended Forgetting factor RLS (EFRLS) can be used for real time parameter estimation [1,7]. These techniques perform joint state and parameter estimation leading to increased computational load. The EKF/UDEKF requires suitable starting values and knowledge on noise covariances. The least squares estimation using the recursive Discrete Fourier Transform (DFT) can also be potentially used for real time parameter estimation. This technique exhibits comparatively slower convergence and requires more number of Floating point Operations (FLOPS) per cycle compared to Recursive Least Squares (RLS) [3]. The equation error RLS described in this paper is an existing algorithm in signal processing domain. In the present work, this algorithm is adapted for aircraft on-line

---

* Scientist, Flight Mechanics and Control Division, National Aerospace Laboratories, Post Box No. 1779, Bangalore-560 017, India; Email : ckamali@css.nal.res.in

Manuscript received on 10 Jan 2007; Paper reviewed, revised and accepted on 07 May 2007

parameter estimation. The proposed algorithm does not require any starting values and noise covariances. It converges comparatively faster and requires less number of FLOPS than the DFT technique. It is sufficiently robust for process and measurement noise [3]. This technique is also found to be suitable for unstable and highly augmented aircraft [3]. The RLS requires the accurate measurements of states and state derivatives. The recent advances in modern instrumentation technology make the measurement of all aircraft state variables possible. The state derivatives are computed on-line with the help of digital filter that combines the role of a differentiator and a low pass filter. Hence, this is a novel implementation of well-established RLS in time domain. This approach can also be extended for performing real time parameter estimation in the absence of aircraft calibrated air data [4]. The problem of covariance wind-up might hamper the performance of this algorithm when it is used for adaptive control. A stabilization approach can be provided when it is used for such applications [4, 8].

It is important that the on-line estimation capability of the proposed algorithm be tested/demonstrated using real time hardware to see if any real time implementation issues arise, because the algorithm is to be subsequently used for flight test related activities. Therefore, the RLS is implemented in a Distributed PC based Engineer in the Loop Simulator (DELS). The DELS is a real-time simulator built by National Aerospace Laboratories (NAL), Bangalore, for an unstable aircraft [9-11]. To achieve the above goal, LabVIEW© (Laboratory Virtual Instrument Engineering Workbench) software is also used. The motivation of LabVIEW© work is to propose a software tool that can be used at the flight test centers for on-line parameter estimation. Embedded software solution is an attractive option to perform real time parameter estimation on-board. Hence, the mechanization of RLS is carried out in DSP hardware, using the simulated data. The aircraft short period estimation is demonstrated on these platforms as it plays a very important requirement for the applications mentioned earlier.

Equation Error Recursive Least Squares (RLS)

The aircraft longitudinal and lateral dynamics can be approximated by the following continuous-time state variable model:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ x(0) &= x_0 \\ y(t) &= x(t) \end{aligned} \tag{1}$$

The discrete time observation can be represented as

$$\begin{aligned} Z_i &= y_i + v_i \\ i &= 1, 2, \ldots, N \end{aligned}$$

where N is the total number of observations.

At any discrete time $t = n$, assemble

$$\begin{aligned} X &= \left[x_1(n)\ x_2(n) \ldots x_k(n)\ u_1(n)\ u_2(n) \ldots u_k(n)\right] \\ Y &= \left[\dot{x}_1(n)\ \dot{x}_2(n) \ldots \dot{x}_k(n)\right] \end{aligned} \tag{3}$$

The state equation at any discrete time can be represented as

$$Y = X\beta + \varepsilon \tag{4}$$

where β is the vector of unknown parameters to be estimated, which contains the elements of system and control matrices A and B and ε is the equation error.

X is obtained after filtering using the following IIR filter

$$H = \frac{1}{s^2 + \sqrt{2}s + 1} \tag{5}$$

Y is obtained after filtering using the following IR filter

$$H = \frac{s}{s^2 + \sqrt{2}s + 1} \tag{6}$$

Thus, the problem of parameter estimation in time domain can be formulated as a standard LS regression problem with the following cost function.

$$J = \sum_{i=1}^{N} \lambda^{N-i} |\varepsilon(i)|^2 \quad i = 1, 2, \ldots, N \tag{7}$$

where λ is a positive constant close to 1, when λ equals 1 the method reduces to ordinary least squares. The use of the weighing factor λ is intended to ensure that data in the distant past are ‘forgotten’ in order to afford the possibility of following the statistical variations of the observable data when the filter operates in a non-stationary environment. Hence, this scheme is also known as exponentially weighted recursive least squares.

The unknown parameter vector β can be estimated recursively using the following steps: Initialize the algorithm by setting $P_0 = \delta^{-1} I$, where δ is a small positive constant and

$$\hat{\beta}_0 = 0$$

For each instant of time, $n = 1,2,..., N$ compute

$$\begin{aligned}
\pi_n &= X_n \; P_{n-1} \\
\kappa_n &= \lambda_n + \pi_n \; X_n^T \\
k_n &= P_{n-1} \; X_n^T / \kappa_n \\
\alpha_n &= Y_n - \left(\hat{\beta}_{n-1} \; X_n^T\right)^T \\
\hat{\beta}_n &= \hat{\beta}_{n-1} + (k_n \, \alpha_n)^T \\
P'_{n-1} &= k_n \, \pi_n \\
P_n &= \frac{1}{\lambda} (P_{n-1} - P'_{n-1})
\end{aligned} \tag{8}$$

here P is the correlation matrix. In addition, the covariance matrix of $\hat{\beta}$ is computed as

$$\text{cov}\,(\hat{\beta}) = E\left\{ (\hat{\beta} - \beta)\,(\hat{\beta} - \beta)^T \right\} = \sigma^2 P \tag{9}$$

where σ is the equation error variance and it can be estimated on-line using

$$\sigma^2 = \left[(Y - X\hat{\beta})^T * (Y - X\hat{\beta}) \right] \tag{10}$$

Furthermore, the standard deviation of the estimation error for the kth unknown of the 'p' parameters in β can be evaluated as the square root of the (k, k) main-diagonal coefficient of the covariance matrix. The standard deviation allows an on-line assessment of the accuracy of the estimated parameter. Hence, this technique does not require any prior knowledge on noise covariances.

Distributed PC based Engineer in Loop Simulator (DELS)

The DELS includes the cockpit and the control stick, pedals and throttle of that of a real aircraft. Visual cues are provided by computer graphics. The ingredients of the DELS are the following:

- A pilot's station comprising the cockpit, the throttle, the control stick, rudder and associated discrete switches
- A data acquisition system to convert the control inputs to digital domain
- Software module to solve the flight dynamics using pilot inputs
- Visualization software and single window projection system
- Executive software to enable linking all the ingredients and to enable simulated flight.

Figure 1 shows the block diagram of DELS in a distributed PC environment highlighting the real time parameter estimation block within the flight model. The flight model used in DELS is realistic as it incorporates aerodynamic data, mass, center of gravity, fuel data, and engine data of an unstable aircraft operating with highly augmented control law.

Results and Discussion from DELS

The RLS is implemented in DELS using FORTRAN programming language and its functionality is verified for short period aircraft parameter estimation. In DELS, the aircraft data is available at every 12.5 millisecond. IIR filters are used to obtain X and Y signals described in eq. (3) for solving the linear regression problem. The 's' transfer function of IIR filters are converted to 'z' transfer function using 'Tustin' transformation. From the 'z' transfer function, difference equation is constructed for implementing the filters. The desired flight condition (Mach, altitude, fuel configuration, landing gear position and atmosphere) is chosen for flying. The aircraft is trimmed for straight and level flight and then the maneuver for parameter estimation is performed by the pilot. The results are presented for two flight conditions. The pilot input along with convergence of parameters is shown in Fig.2. The pilot input for flight condition-2 is shown in Fig.3 along with the convergence plots of parameters. It can be noticed that the estimated parameters generally converge to their true values as soon as the maneuver is over. The results are tabulated in Table-1. The performance is evaluated off-line using the following measure:

Parameter Estimation Error Norm (PEEN)

$$PEEN = \frac{norm\ (\beta_t - \hat{\beta})}{norm\ (\beta_t)} * 100 \tag{11}$$

β_t is the vector of true parameters and $\hat{\beta}$ is the vector of Estimated parameters. The PEEN for both flight conditions are acceptable.

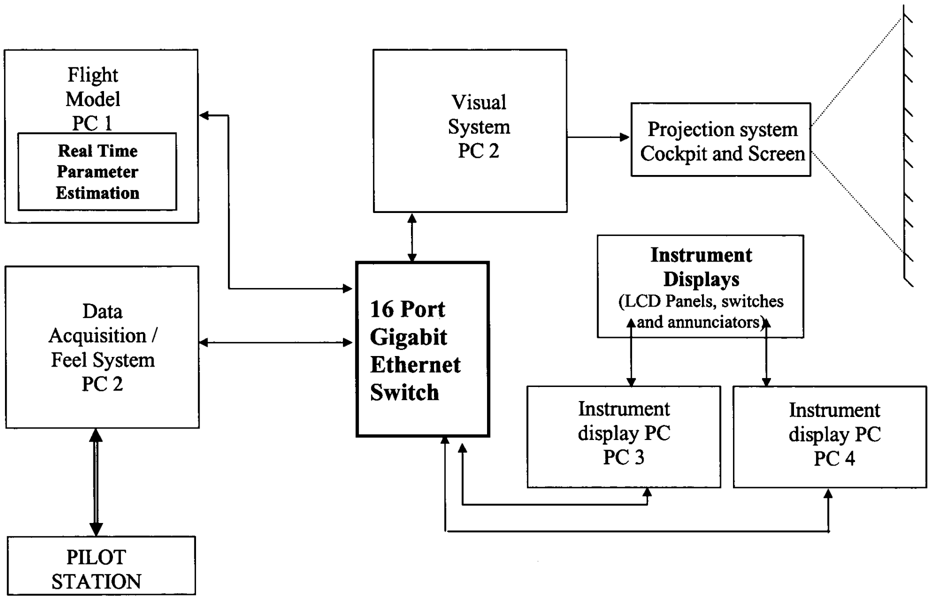


Fig.1 Block diagram of DELS

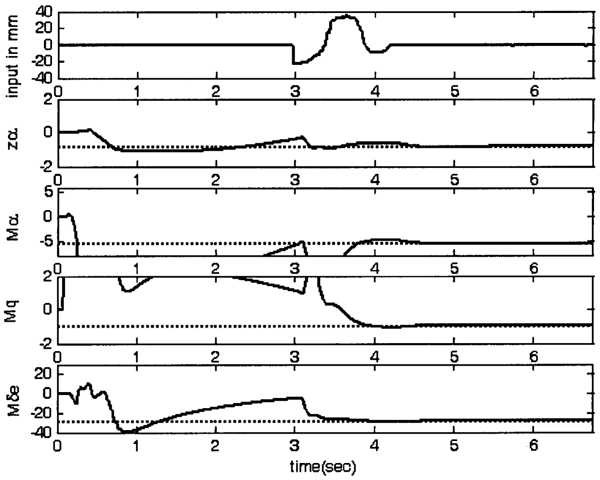


Fig.2 Convergence of estimated parameters (DELS, Mach = 0.8, altitude = 7000 meters)

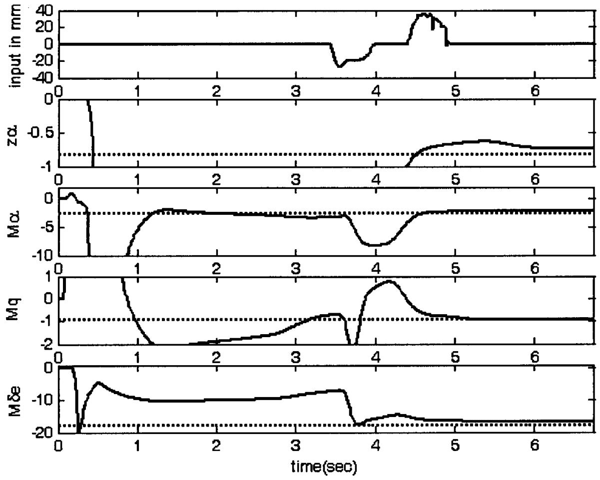


Fig.3 Convergence of estimated parameters (DELS, Mach = 0.5, altitude = 3000 meters)

On-line Parameter Estimation Using LabVIEW

The LabVIEW© is a graphical programming language that incorporates virtual instruments that imitate physical instruments such as oscilloscope, meters etc. LabVIEW© contains three main components: the front panel, the block diagram, and the icon/connector pane. LabVIEW© supports several low-level protocols that can be used for communication between computers. It provides facility to connect hardware for data acquisition. Hence, it is decided to implement RLS in LabVIEW© that could certainly yield a handy software to be used at the flight-test centers. To validate the software, simulated data is used. Aircraft data simulation is also accomplished in LabVIEW©. The implementation and on-line estimation demonstration is carried out in the following manner.

The data simulation program is made to reside in one PC (Personal Computer) called data generation PC. The VI (Virtual Instrumentation) block diagram for data generation is shown in Fig. 4. In the present work, the state equations are actually implemented instead of using a

Table-1 : Results of estimation implemented in DELS

Aircraft Short Period Parameters	True Values for Flight Condition-1 Mach=0.8, Altitude = 7000 meters	Estimated Values for Flight Condition-1	True Values for Flight Condition-2 Mach=0.5, Altitude = 3000 meters	Estimated Values for Flight Condition-2
Z_α	-0.8646	-0.778	-0.8281	-0.7326
Z_q	0.9837	0.9852	0.9746	0.9626
$Z_{\delta e}$	-0.3465	-0.4051	-0.3427	-0.4300
M_α	-5.5344	-5.5190	-2.6840	-2.2671
M_q	-0.9984	-0.9543	-0.9562	-0.9292
$M_{\delta e}$	-29.7336	-28.4191	-17.7674	-16.8797
PEEN	**4.3559**		**5.4851**	

readymade state space block available in LabVIEW©. The demonstration assumes the availability of state measurements. Hence, the output equations are not required to be implemented. The front panel for this block diagram is shown in Fig. 5. The front panel is designed to provide a GUI (Graphical User Interface) for entering the aircraft model parameter values for which, the data simulation is intended. Since this is aircraft short period (longitudinal) data simulation, pilot pitch stick input is to be required to excite the short period mode. To accomplish this requirement, a dial instrument is placed in the front panel, which can be adjusted to apply a pitch stick doublet input. The minimum and maximum value of dial instrument can be fixed at any desired value. The front panel is also designed to have graphical plots for displaying the simulated aircraft data.

Parameter Estimation PC

As previously mentioned the RLS algorithm is implemented in LabVIEW and is made to reside in another PC called parameter estimation PC. The block diagram for the implementation of RLS using VI is shown in Fig. 6. The implementation procedure of RLS in LabVIEW is shortly given as follows:

- The control toolbox is used to implement transfer functions, the mathematics toolbox is used to implement all basic mathematical operations and the linear algebra toolbox is used to implement matrix related operations.
- The transfer function blocks placed in the VI block diagram are analog IIR filters represented by Eq. (5) and (6).
- In the VI block diagram, the equation implementation terminating with block denoted by 'x_0' is the parameter update equation; the equation implementation terminating with block denoted by 'p' is the covariance update equation.
- In Lab VIEW, there are differences between arrays and matrices. Hence, after filtering X and Y signals from the IIR filters, a block used for converting 'array to matrix' is used to convert all augmented signals into matrices and then all subsequent computations are performed. The same emphasis is insisted, whenever there is a multiplication, transpose and inversion operations.
- The front panel for this VI is shown in Fig. 7. The front panel is designed to have indicators to observe the estimated values of parameters. The indicators are named as Estimated parameters (Z_alpha to M_deltae). The front panel also incorporates on-line plotting facility for the estimated parameters. In this work, software development is validated using simulated data and the true values of parameters are plotted along with the estimated values for checking the convergence of estimation to true values in the front panel.

Communication Between PCs

In the present work UDP (User Datagram Protocol) is used to establish the communication between the data generation PC and parameter estimation PC. UDP provides simple, low-level communication among processes on computers and is not a connection-based protocol such as TCP (Transmission Control Protocol), where one needs

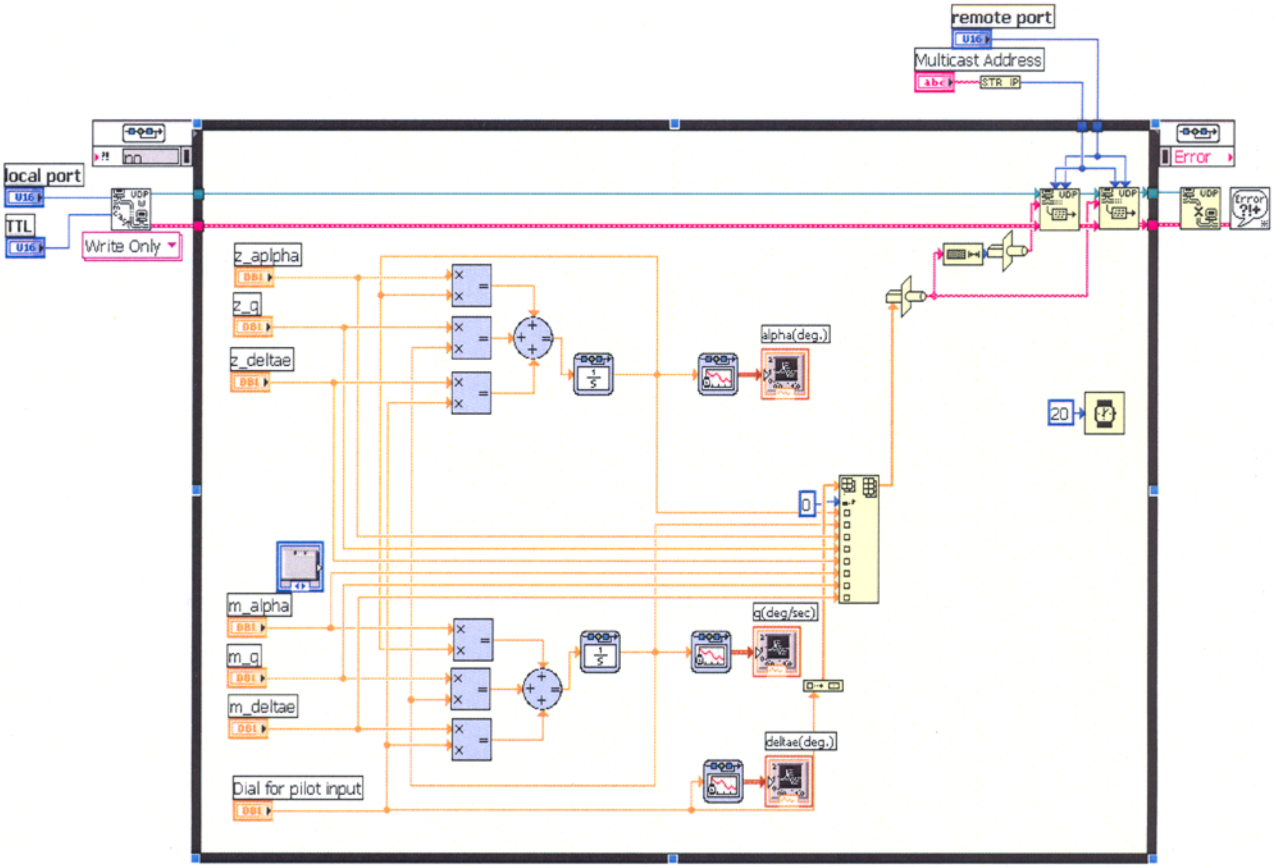
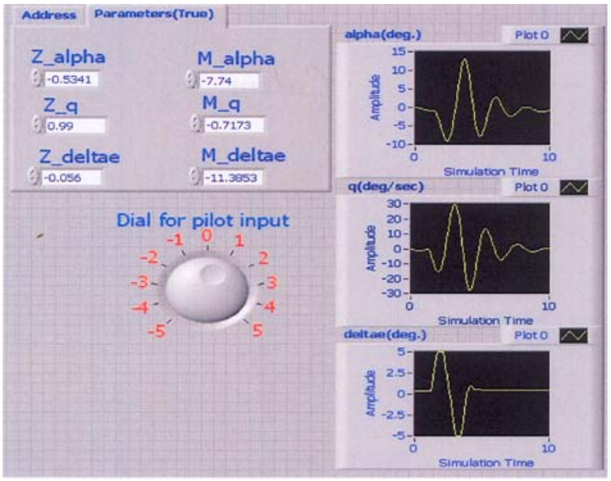


Fig.4 VI Block diagram for data generation



Z_alpha -0.5341
M_alpha -7.74
Z_q 0.99
M_q -0.7173
Z_deltae -0.056
M_deltae -11.3853

Fig.5 VI Front panel for data simulation

to establish a connection with a destination before the data is sent or received. In UDP, processes communicate by sending datagrams to a destination computer or port. A port is the location where the data is sent. IP (Internet Protocol) handles the computer-to-computer delivery. After the datagram reaches the destination computer, UDP moves the datagram to its destination port. If the destination port is not open, UDP discards the datagram. In the present work, the data generation PC is sending the aircraft data and parameter estimation PC is the destination, where data is received.

Results and Discussion of Estimation Using LabVIEW

The VI that performs the data generation and the VI that performs parameter estimation are made to run simultaneously. At the data generation front panel the dial is adjusted to give a pilot's doublet pitch stick input. The data generation PC acts like the aircraft plant. The short period mathematical model has the following description:

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix} = \begin{bmatrix} Z_{\alpha} & Z_{q} \\ M_{\alpha} & M_{q}\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix} Z_{\delta e}\\ M_{\delta e}\end{bmatrix}\delta_{e} \qquad (12)$$

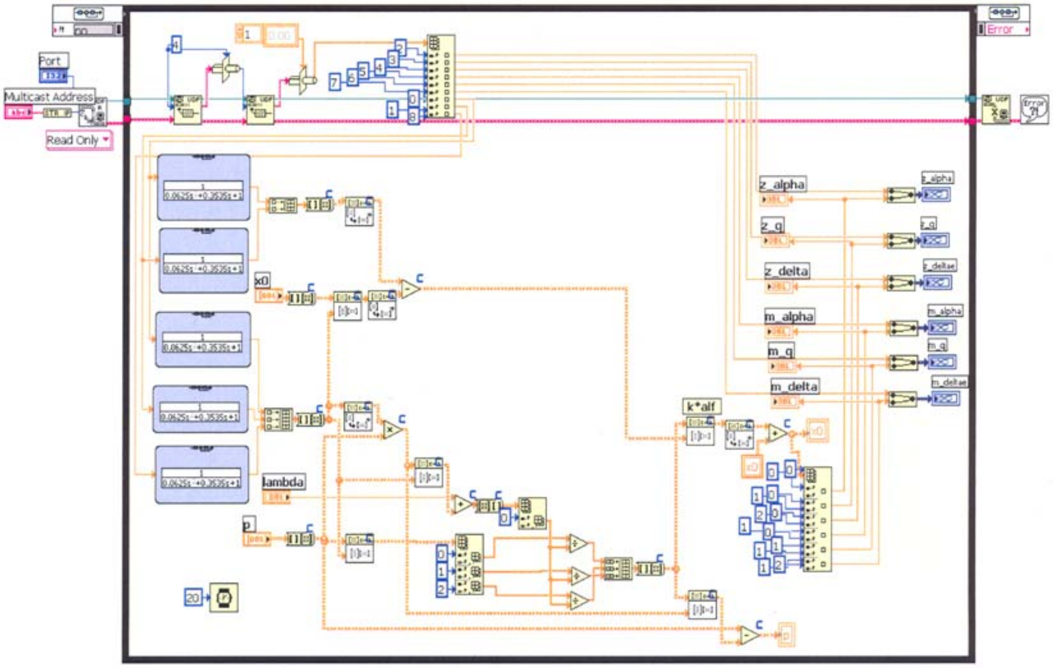
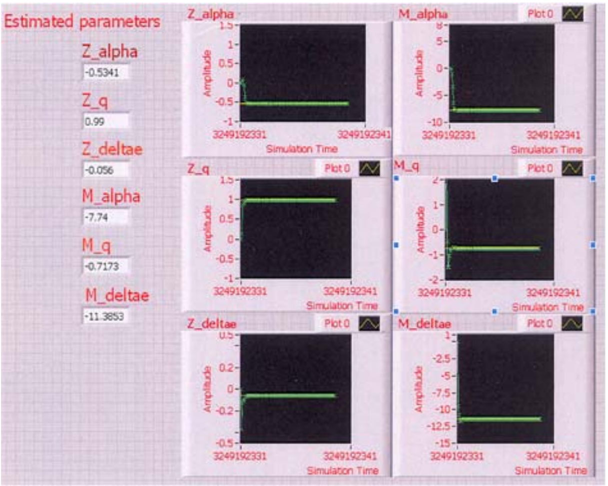


Fig.6 VI Block diagram for RLS implementation


Estimated parameters

Z_alpha: -0.5341

Z_q: 0.99

Z_deltae: -0.056

M_alpha: -7.74

M_q: -0.7173

M_deltae: -11.3853

Fig.7 VI Front panel to visualize the on-line estimated results

The front panel indicators show the aircraft model for which the data is simulated as can be seen from Fig. 5. This data is acquired by the parameter estimation PC through UDP protocol and is fed to the algorithm. The algorithm estimates the parameter as the data come in-line. The variation of estimated parameters and their convergence is observed in the front panel. The indicators indicate the numerical values of estimated parameters. The results shown by on-line plots and indicators are given in Fig. 7. It can be noted that the estimated parameters converge to their true values. The estimated results are tabulated in Table-2. In the present work, the data is assumed as real-time and analog signal. Hence, the IIR filters are also implemented as analog filters. The analog signal assumption does not require any information about sampling time of incoming data. In reality, the aircraft telemetry signals are sampled at definite sampling frequency. However, this can easily be tackled in LabVIEW© by discretizing the IIR filters. Instead of continuous transfer function, discrete transfer function is to be used to implement the filters. The discrete transfer

Table-2 : Results of estimation implemented in LabVIEW

Aircraft Short Period Parameters	True Values	Estimated Values
Z_α	-0.5341	-0.5341
Z_q	0.99	0.99
$Z_{\delta e}$	0.056	0.56
M_α	-7.74	-7.74
M_q	0.7173	0.7173
$M_{\delta e}$	-11.3853	-11.3853
PEEN		**0**

function can be found out from corresponding continuous transfer function using the Tustin transformation. It can be noted that the performance measure PEEN is 0. This may be because of the use of direct analog signals without any noise and of sufficiently high precision.

Mechanization of RLS in DSP Processor

In an aircraft to perform on-line parameter estimation on-board, embedded environment is an attractive option, as it requires less memory and occupies less space. Hence in this work, RLS is mechanized in DSP hardware and its functionality is validated using' simulated data that will show its suitability on embedded environment. The TMS320C6711 DSP processor is used to accomplish the above task.

Results and Discussion (RLS-DSP Implementation)

The Texas Instruments Code Composer Studio (CCS) was used to generate the assembly code generation for the DSP processor. The following are the steps followed in creating a project in CCS:

- In code Composer Studio^{TM}, create a folder for the project.
- Copy the contents of CCStudio_v3.10\tutorial\target\PROJECT_NAME folder to this new folder.
- From the Project menu, choose New.
- In the Project Name field, type PROJECT_NAME
- In the Location field, browse to the working folder created in step 1.
- In the Project Type field, select Executable (.out).
- In the Target field, select target configuration and click Finish.
- The CCS program creates a project file called PROJECT_NAME.pjt. This file stores project settings and references and the various files used by the project.

The following types of file extensions are to be added to the project:

.lib This library provides runtime support for the target DSP chip.
.c This file contains source code that provides the main functionality of this project.
.h This file declares the buffer C-structure as well as define any required constants.
.pjt This file contains all of your project build and configuration options.
.asm This file contains assembly instruction.
.cmd This file maps sections to memory.

The proposed RLS algorithm described under section (Equation Error RLS) was programmed in C language using CCS for real time parameter estimation. The aircraft short period data simulation for the parameter estimation was accomplished using the fourth order Runge Kutta algorithm (RK4). The RK4 algorithm was also programmed in C language using CCS and it is a part the main source code. The pilot pitch stick input was provided as input to RK4 algorithm that in turn yields aircraft states. The aircraft states together with control input goes into the parameter estimation algorithm and the parameters are estimated. There is also a facility to modify the application in assembly language. In the present work there was no necessity to do this as RLS along with RK4 met the timing requirements according to the sampling frequency of the pilot input. The pilot input was sampled at 10 milliseconds and RLS along with RK4 was found to take 1.4 millisecond. Also the CCS has inbuilt code optimization options which rapidly increase the execution speed of an application without requiring the developer to create or modify assembly code. This optimization option was chosen so that suitable level of performance is achieved without needing to code in assembly.

The project for parameter estimation resides in the host PC. The host PC was made to communicate with the processor through the parallel port. A snapshot depicting this is given in Fig. 8. The assembly code generated by the CCS was downloaded to the processor and was made to run in the processor. The results were collected at the host PC. The parameters of the short period model described below were estimated using the processor.

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_w & Z_q \\ M_w & M_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta e} \\ M_{\delta e} \end{bmatrix} \delta_e \tag{13}$$

Graphs of 10-second time histories for the estimated parameters are shown in Fig. 9. The 'X' axis unit in the plot is to be multiplied with sampling interval to interpret the time. On the left side of Fig. 9 the numerical values of estimated parameters are given under the variable name BETA. It can be noted that the estimated parameters converged to their true values and the algorithm shows its feasibility in embedded environment. The results are tabulated in Table-3, which shows acceptable PEEN.

Conclusion

The paper proposes a well-validated and efficient real-time aircraft parameter estimation technology through its integration in a real time aircraft simulator. The software development of the algorithm in LabVIEW© has shown its feasibility to be used at the flight test centers for real time parameter estimation. The LabVIEW© provides support for the hardware integration to acquire the analog/digital signals for parameter estimation. These real signals can be fed to the developed software for parameter

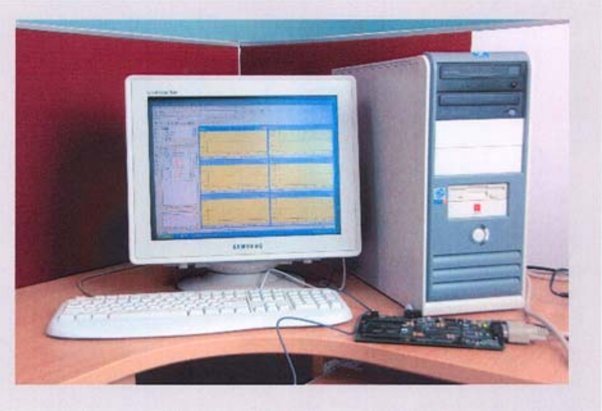
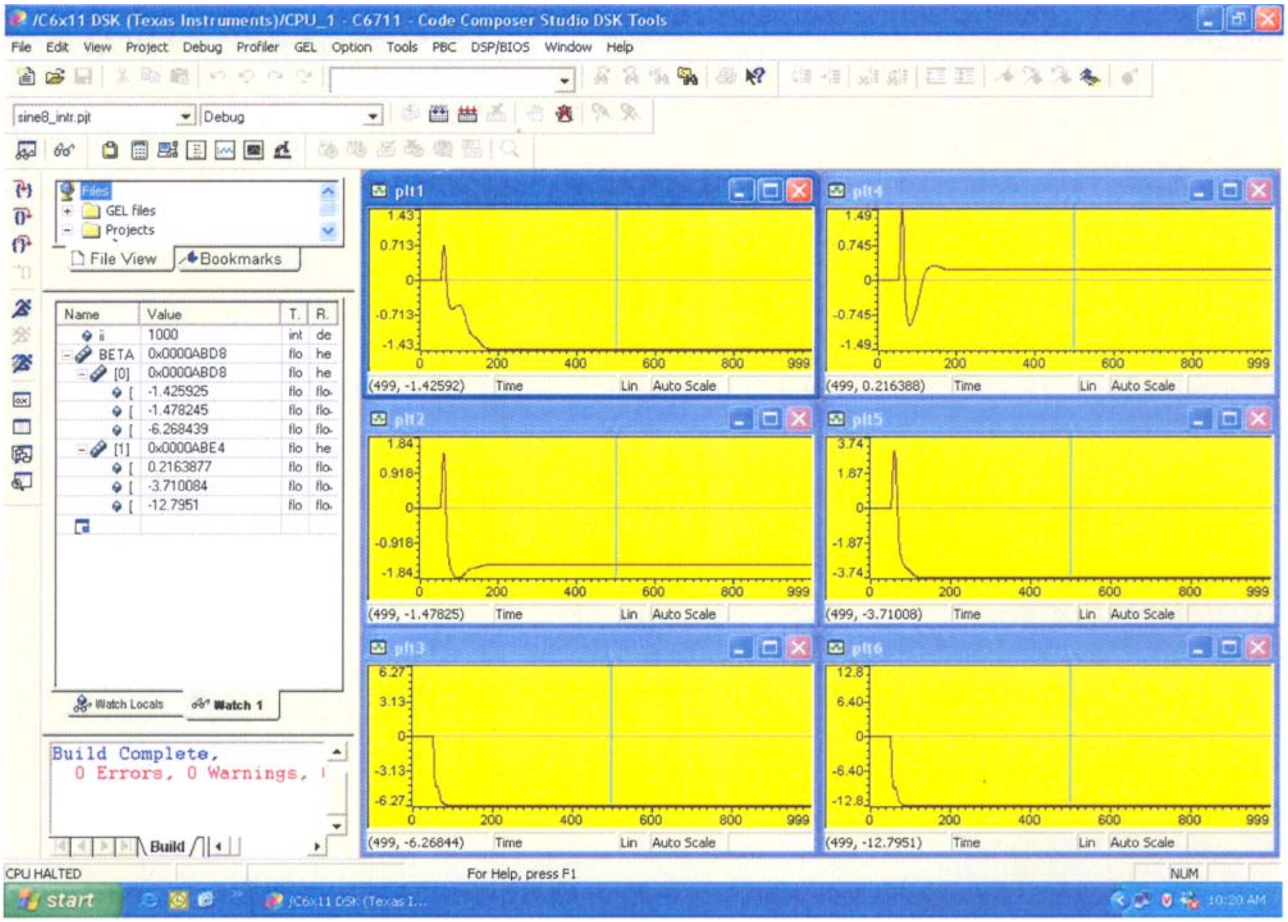


Fig.8 Snapshot of TMS320C6711 connected to the host PC

Table-3 : Results of estimation implemented in DSP Hardware

Aircraft Short Period Parameters	True Values	Estimated Values
Z_α	-1.4249	-1.4249
Z_q	-1.4768	-1.4768
$Z_{\delta e}$	-6.2632	-6.2632
M_α	0.2163	0.2163
M_q	-3.7067	-3.7067
$M_{\delta e}$	-12.7840	-12.7840
PEEN		**0.0862**



Name	Value	T.	R.
ii	1000	int	de
BETA	0x0000ABD8	flo	he
[0]	0x0000ABD8	flo	he
[	-1.425925	flo	flo
[	-1.478245	flo	flo
[	-6.268439	flo	flo
[1]	0x0000ABE4	flo	he
[	0.2163877	flo	flo
[	-3.710084	flo	flo
[	-12.7951	flo	flo

```
Build Complete,
  0 Errors, 0 Warnings,
```

Fig.9 Results obtained from TMS320C711 estimation in place of the data simulation part described in this paper. The mechanization of the proposed algorithm in TMS320C6711 processor shows its capability to be implemented for on- board applications in an aircraft.

Acknowledgements

The authors would like to thank Mrs. Padma Madhuranath, Group Head, Moncy T Thomas and K P Srikanth, Scientists, Flight Simulation, FMCD, NAL for extending the aircraft simulator facility to carry out this work. Thanks are also due to Dr. S.V. Narasimhan and Mrs. S. Veena, Scientists of ALD, NAL for providing the DSP processor and valuable inputs. The authors wish to acknowledge Mrs. Seema, Research intern, FMCD, NAL for her coordination in the LabVIEW work.

References

1. Raol, J.R., Girija, G. and Singh, J., "Modeling and Parameter Estimation of Dynamic Systems", IEE Control Engineering Book Series, Vol. 65, lEE, United Kingdom, August, 2004.
2. Marcello R Napolitano, Yongkyu Song and Brad Seanor., "On-line Parameter Estimation for Restructurable Flight Control Systems", Aircraft Design, 2001, pp.19- 50.
3. Kamali, C. and Pashilkar, A.A., "Real Time Parameter Estimation for Reconfigurable Control of Unstable Aircraft", NAL Project Document No. FC0603.
4. Kamali, C., Pashilkar, A.A. and Raol, J.R., "Study of RLS for sTability Margin Evaluation and Post Failure Data Analysis", NAL Project Document No. FC 0613, July, 2006.
5. Vijay V.Patel., Girish Deodhare and Shyam Chetty., "Near Real Time Stability Margin Estimation from Piloted 3-2-1-1 Inputs", AIAA Aircraft Technology Integration and Operations (ATIO2002), Technical Forum, October, 2002.
6. Srinathkumar., Padma Madhuranath and Girija Gopalaratnam., "Estimation of Neutral and Maneuver Points of Aircraft by Dynamic Maneuvers", AIAA Paper No.93-3620-CP, August, 1993.
7. Basappa and Jategaonkar, R V., "Evaluation of Recursive Methods for Aircraft Parameter Estimation", DLR IB 111-2003/32.
8. Shore, D. and Bodson, M., "Flight Testing of a Reconfigurable Control System on an Unmanned Aircraft", Journal of Guidance, Control and Dynamics, Vol.28, No. 4, July-Aug, 2005, pp.698-707.
9. Srikanth, K P, Moncy T. Thomas and Padma Madhuranath., "Design and Development of DELS - Part 1: Out of the Window Visuals", NAL Project Document No. FC 0608, March, 2006.
10. Moncy T. Thomas., Srikanth, K.P. and Padma Madhuranath., "Design and Development of DELS - Part 2: System Integration", NAL Project Document No.FC 0609, March, 2006.
11. Padma Madhuranath, "Introduction to Flight Simulation", NAL - UNI Lecture Series 10, August, 1997.
12. Robert C. Nelson., "Flight Stability and Automatic Control", McGraw-Hill International Editions, Aerospace Series, 1990.