5Q1

THE UNIVERSITY OF KANSAS CENTER FOR RESEARCH, INC.
2291 Irving Hill Drive—Campus West
Lawrence, Kansas 66045

(NASA-CR-168438) DEVELOPMENT OF A SIMPLE, SELF-CONTAINED FLIGHT TEST DATA ACQUISITION SYSTEM Progress Report (Kansas Univ. Center for Research, Inc.) 208 p HC A10/MF A01 CSCL 14B G3/35

N82-17478

Unclas 11611

Progress Report on

PHASE II:
DEVELOPMENT OF A SIMPLE, SELF-CONTAINED FLIGHT TEST DATA ACQUISITION SYSTEM

KU-FRL-407-6

NASA Grant NSG-4019

Prepared by: Ronald R. L. Renz

Principal Investigators: Jan Roskam
Dale I. Rummer

Work Performed by: Mark A. Mosser
Alex Kotsabasis
Robert Clarke
Douglas Shane

University of Kansas Flight Research Laboratory
Lawrence, Kansas 66045
February 1981

ABSTRACT

DEVELOPMENT OF A SIMPLE, SELF-CONTAINED FLIGHT TEST DATA ACQUISITION SYSTEM

This report describes work done under a continuing program to develop a simple, self-contained flight test data acquisition system. In the past, instrumenting an airplane for flight testing has taken a great deal of time and money. With recent advances in sensor and microprocessor technology, a simple, low-cost system could be developed which would be applicable to general aviation airplanes.

This system was conceived to obtain performance and stability characteristics of airplanes. The design criteria for the system were that it be easy to install, self-contained, and simple; that it require no special/difficult flight techniques; and that it be applicable to general aviation airplanes and low in cost.

The system developed meets these criteria for doing lontitudinal and lateral stability analysis. The package consists of three modules. These are 1) microprocessor controller and data acquisition module, 2) transducer module, and 3) power supply module. The system is easy to install and occupies space in the cabin or baggage compartment of the airplane. All transducers are contained in these modules except the total pressure tube, static pressure air temperature transducer, and control position transducers.

The data reduction technique used was the NASA-developed MMLE program. This has been placed on a microcomputer, and all data reduction is done on the microcomputer. This greatly reduces the cost of the data reduction. Also, when compared with the analogue recording techniques, still being used, there has been a large improvement in the accuracy of results.

The flight testing program undertaken has proven both the flight testing hardware and the data reduction method to be applicable to the current field of general aviation airplanes.

This report describes the instrumentation system developed, the data reduction method used, and important results of the flight test program.

ACKNOWLEDGEMENTS

We would like to thank NASA Dryden Flight Research Center via Mr. Ken Szalai, Technical Monitor, and Dr. E. Kordes for the financial support of this program. Special thanks also go to Cessna Aircraft Company for the assistance they have given in this phase of the program. Sincerest thanks go to our secretary and grammarian, Mrs. Nancy Hanson, for the many hours spent typing and debugging this manuscript.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

APPENDICES:

LIST OF SYMBOLS

All parameters in this report are referenced to a system of body axes as shown in Figure 1.1.

Symbol	Definition	Dimension
$A = -X$	Force in A direction	lb
A_x, A_X	Longitudinal acceleration	g
A_y, A_Y	Lateral acceleration	g
A_z, A_Z	Verticle acceleration	g
$A_N = -A_Z$	Normal acceleration	g
$[A]$	Stability matrix	
$[B]$	Control matrix	
b	Wing span	ft
$\{c\}$	Vector of unknowns for MMLE	
$\bar{c}$	Mean aerodynamic chord	ft
$C_A = \frac{A}{\bar{q}S} = -C_X$	Coefficient of force in A direction (A = -X)	
$C_{A_\alpha} = \frac{\partial C_A}{\partial \alpha}$	Variation of body A coefficient with angle of attack	rad^{-1}
$C_{A_u} = \frac{\partial C_A}{\partial (\frac{u}{U})}$	Variation of body A coefficient with speed	
$C_{A_{\delta_{E,c}}} = \frac{\partial C_A}{\partial \delta_{E,c}}$	Variation of body A coefficient with elevator or canard angle	rad^{-1}
C_{A_o}	Nondimensional longitudinal force equation bias	
$C_D = \frac{D}{\bar{q}S}$	Drag force coefficient	

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$C_{D_\alpha} = \frac{\partial C_D}{\partial \alpha}$	Variation of drag coefficient with angle of attack	rad^{-1}
$C_{D_u} = \frac{\partial C_D}{\partial (\frac{u}{U_1})}$	Variation of drag coefficient with speed	
$C_{D_{\delta_E}} = \frac{\partial C_D}{\partial \delta_E}$	Variation of drag coefficient with elevator angle	rad^{-1}
$C_L = \frac{L}{\bar{q}S}$	Lift force coefficient	
$C_{L_\alpha} = \frac{\partial C_L}{\partial \alpha}$	Variation of lift coefficient with angle of attack	rad^{-1}
$C_{L_{\dot{\alpha}}} = \frac{\partial C_L}{\partial (\frac{\dot{\alpha}\bar{c}}{2U_1})}$	Variation of lift coefficient with rate of change of angle of attack	
$C_{L_q} = \frac{\partial C_L}{\partial (\frac{q\bar{c}}{2U_1})}$	Variation of lift coefficient with pitch rate	
$C_{L_u} = \frac{\partial C_L}{\partial (\frac{u}{U_1})}$	Variation of lift coefficient with speed	
$C_{L_{\delta_E}} = \frac{\partial C_L}{\partial \delta_E}$	Variation of lift coefficient with elevator angle	rad^{-1}
$C_\ell = \frac{L}{\bar{q}\ S\ b}$	Rolling moment coefficient	
$C_{\ell_\beta} = \frac{\partial C_\ell}{\partial \beta}$	Variation of rolling moment coefficient with sideslip angle	rad^{-1}

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$C_{\ell_p} = \frac{\partial C_\ell}{\partial p}$	Variation of rolling moment coefficient with roll rate	rad^{-1}
$C_{\ell_r} = \frac{\partial C_\ell}{\partial r}$	Variation of rolling moment coefficient with yaw rate	rad^{-1}
$C_{\ell_{\delta_{A,R}}}$	Variation of rolling moment coefficient with aileron or rudder angle	rad^{-1}
$C_m = \frac{M}{\bar{q}S\bar{c}}$	Pitching moment coefficient	
$C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha}$	Variation of pitching moment coefficient with angle of attack	rad^{-1}
$C_{m_{\dot{\alpha}}} = \frac{\partial C_m}{\partial (\frac{\dot{\alpha}\bar{c}}{2U_1})}$	Variation of pitching moment coefficient with rate of change of angle of attack	
$C_{m_q} = \frac{\partial C_m}{\partial (\frac{\bar{q}c}{2U_1})}$	Variation of pitching moment coefficient with pitch rate	
$C_{m_u} = \frac{\partial C_m}{\partial (\frac{u}{U_1})}$	Variation of pitching moment coefficient with speed	
C_{m_T}	Pitching moment coefficient due to thrust	
$C_{m_{T_\alpha}} = \frac{\partial C_{m_T}}{\partial \alpha}$	Variation of thrust pitching moment coefficient with angle of attack	rad^{-1}
$C_{m_{T_u}} = \frac{\partial C_{m_T}}{\partial (\frac{u}{U_1})}$	Variation of thrust pitching moment coefficient with speed	
$C_{m_{\delta_{E,c}}} = \frac{\partial C_m}{\partial \delta_{E,c}}$	Variation of pitching moment coefficient with elevator or canard angle	rad^{-1}

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
C_{m_o}	Nondimensional pitching moment equation bias	
$C_N = \frac{N}{\bar{q}S} = -C_Z$	Normal force coefficient. ($N = -Z$)	
$C_{N_\alpha} = \frac{\partial C_N}{\partial \alpha}$	Variation of normal force coefficient with angle of attack	rad^{-1}
$C_{N_u} = \frac{\partial C_N}{\partial (\frac{u}{U_1})}$	Variation of normal force coefficient with speed	
$C_{N_{\delta_{E,c}}} = \frac{\partial C_N}{\partial \delta_{E,c}}$	Variation of normal force coefficient with elevator or canard angle	rad^{-1}
C_{N_o}	Nondimensional normal force equation bias	
$C_n = \frac{N}{\bar{q}\ Sb}$	Yawing moment coefficient	
$C_{n_\beta} = \frac{\partial C_n}{\partial \beta}$	Variation of yawing moment coefficient with sideslip angle	rad^{-1}
C_{n_T}	Yawing moment coefficient due to thrust	
$C_{n_{T_\beta}} = \frac{\partial C_{n_T}}{\partial \beta}$	Variation of thrust yawing moment coefficient with sideslip angle	rad^{-1}
$C_{n_p} = \frac{\partial C_n}{\partial p}$	Variation of yawing moment coefficient with roll rate	rad^{-1}
$C_{n_r} = \frac{\partial C_n}{\partial r}$	Variation of yawing moment coefficient with yaw rate	rad^{-1}
$C_{n_{\delta_{A,R}}} = \frac{\partial C_n}{\partial \delta_{A,R}}$	Variation of yawing moment coefficient with aileron or rudder angle	rad^{-1}

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$C_{T_x} = \frac{T_x}{\bar{q}S}$	Thrust force coefficient in X direction	
$C_{T_{x_u}} = \frac{\partial C_{T_x}}{\partial(\frac{u}{U_1})}$	Variation of thrust force coefficient with speed	
$C_X = \frac{X}{\bar{q}S}$	Force coefficient in X direction	
$C_{X_\alpha} = \frac{\partial C_X}{\partial \alpha}$	Variation of longitudinal force coefficient with angle of attack	rad^{-1}
$C_{X_u} = \frac{\partial C_X}{\partial(\frac{u}{U_1})}$	Variation of longitudinal force ~~coefficient with speed~~	
$C_{X_{\delta_{E,c}}} = \frac{\partial C_X}{\partial \delta_{E,c}}$	Variation of longitudinal force coefficient with elevator or canard angle	rad^{-1}
C_{X_o}	Nondimensional longitudinal force equation bias	
$C_y = \frac{Y}{\bar{q}S} = C_Y$	Force coefficient in Y direction	
$C_{y_\beta} = \frac{\partial C_y}{\partial \beta}$	Variation of side force coefficient with sideslip angle	rad^{-1}
$C_{y_p} = \frac{\partial C_y}{\partial p}$	Variation of side force coefficient with roll rate	rad^{-1}
$C_{y_r} = \frac{\partial C_y}{\partial r}$	Variation of side force coefficient with yaw rate	rad^{-1}
$C_{y_{\delta_{A,R}}} = \frac{\partial C_y}{\partial \delta_{A,R}}$	Variation of side force coefficient with aileron or rudder angle	rad^{-1}

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$C_Z = \frac{Z}{\bar{q}S}$	Force coefficient in Z direction	
$C_{Z_\alpha} = \frac{\partial C_Z}{\partial \alpha}$	Variation of vertical force coefficient with angle of attack	rad^{-1}
$C_{Z_u} = \frac{\partial C_Z}{\partial (\frac{u}{U_1})}$	Variation of vertical force coefficient with speed	
$C_{Z_{\delta_{E,c}}} = \frac{\partial C_Z}{\partial \delta_{E,c}}$	Variation of vertical force coefficient with elevator or canard angle	rad^{-1}
C_{Z_o}	Nondimensional vertical force equation bias	
C_c	Y axis force coefficient in wind tunnel axes	
D	Drag force	lb
[D]	MMLE weighting matrix	
g,G	Force of gravity	ft sec^{-2}
[G]	MMLE observation matrix	
[H]	MMLE observation matrix	
H_p	Pressure altitude	ft
[I]	Identity matrix	
I_{xx}, I_{yy}, I_{zz}	Moment of inertia about the X, Y, and Z axes respectively	slug ft^2
I_{xz}	Product of inertia	slug ft^2
J	MMLE cost function	
KTAS	True airspeed	knots
ℓ,L	Rolling moment (perturbed, total)	ft lb
L	Lift force	lb

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
L	Iteration number	
L_β	Dimensional variation of rolling moment with sideslip angle	sec^{-2}
L_p	Dimensional variation of rolling moment with roll rate	sec^{-1}
L_r	Dimensional variation of rolling moment with yaw rate	sec^{-1}
$L_{\delta_{A,R}}$	Dimensional variation of rolling moment with aileron or rudder angle	sec^{-2}
L_o	Rolling moment equation bias	sec^{-2}
m, M	Pitching moment (perturbed,total)	ft lb
m	Mass	slug
MP	Engine manifold pressure	
M_α	Dimensional variation of pitching moment with angle of attack	sec^{-2}
$M_{\dot{\alpha}}$	Dimensional variation of pitching moment with rate of change of angle of attack	sec^{-1}
M_q	Dimensional variation of pitching moment with pitch rate	sec^{-1}
M_u	Dimensional variation of pitching moment with speed	$ft^{-1}\ sec^{-1}$
M_{T_α}	Dimensional variation of pitching moment due to thrust with angle of attack	sec^{-2}
M_{T_u}	Dimensional variation of pitching moment due to thrust with speed	$ft^{-1}\ sec^{-1}$

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$M_{\delta_{E,c}}$	Dimensional variation of pitching moment due to elevator or canard angle	sec^{-2}
M_o	Pitching moment equation bias	sec^{-2}
M_θ	Dimensional variation of pitching moment with pitch angle	sec^{-2}
n, N	Yawing moment (perturbed, total)	ft lb
$N = -Z$	Normal force	lb
N_β	Dimension variation yawing moment with sideslip angle	sec^{-2}
N_{T_β}	Dimensional variation of yawing moment due to thrust with sideslip angle	sec^{-2}
N_p	Dimensional variation of yawing moment with roll rate	sec^{-1}
N_r	Dimensional variation of yawing moment with yaw rate	sec^{-1}
$N_{\delta_{A,R}}$	Dimensional variation of yawing moment with aileron or rudder angle	sec^{-2}
N_o	Yawing moment equation bias	sec^{-2}
p, P	Roll rate	rad sec^{-1}, deg sec^{-1}
P_D	Dynamic pressure	knots (speed) lb ft^{-2}
P_S	Static pressure	ft (altitude) lb ft^{-2}
P_T	Total pressure	lb ft^{-2}
q, Q	Pitch rate	rad sec^{-1}

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$\bar{q}$	Dynamic pressure	lb ft^{-2}
r,R	Yaw rate	rad sec^{-1} deg sec^{-1}
RPM	Engine rotational speed	
[R]	Acceleration transformation matrix	
S	Wing area	ft^{2}
t,T	Time point	sec
T	Temperature	°F
T_X	Thrust force in X direction	lb
u,U	Speed (perturbed, total)	ft sec^{-1} mph
{u(t)}	Control vector	
v	Perturbed sideward velocity	ft sec^{-1}
{v}	MMLE variable bias vector	
V_X, V_x	Longitudinal velocity	ft sec^{-1}
V_Y, V_y	Lateral velocity	ft sec^{-1}
V_Z, V_z	Normal velocity	ft sec^{-1}
w	Perturbed downward velocity	ft sec^{-1}
{x(t)}	State vector	
X	Force in X direction	lb
$\bar{X}$	Distance in the X direction from the center of gravity	ft
X_α	Dimensional variation of X-force with angle of attack	ft sec^{-2}
X_u	Dimensional variation of X-force with speed	sec^{-1}
X_{T_u}	Dimensional variation of X-force due to thrust with speed	sec^{-1}

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$X_{\delta_{E,c}}$	Dimensional variation of X-force with elevator or canard angle	ft $\sec^{-2}$
X_o	Longitudinal force equation bias	ft $\sec^{-2}$
$\{y(t)\}$	Computed observation vector	
$y_i = \{y(i)\}$	Computed observation vector at time i	
Y	Force in Y direction	lb
$\bar{Y}$	Distance in Y direction from the center of gravity	ft
Y_β	Dimensional variation of Y-force with sideslip angle	$\sec^{-1}$, ft $\sec^{-2}$
Y_p	Dimensional variation of Y-force with roll rate	ft $\sec^{-1}$
Y_r	Dimensional variation of Y-force with yaw rate	ft $\sec^{-1}$
$Y_{\delta_{A,R}}$	Dimensional variation of Y-force with aileron or rudder angle	ft $\sec^{-2}$ $\sec^{-1}$
Y_o	Lateral acceleration equation bias	$\sec^{-1}$
$\{z(t)\}$	Measured observation vector	
$z_i = \{z(i)\}$	Measured observation vector at time i	
$Z = -N$	Force in the Z direction	lb
$\bar{Z}$	Distance in Z direction from the center of gravity	ft
Z_α	Dimensional variation of Z-force with angle of attack	ft $\sec^{-2}$

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
$Z_{\dot{\alpha}}$	Dimensional variation of Z-force with rate of change of angle of attack	ft sec^{-1}
Z_q	Dimensional variation of Z-force with pitch rate	ft sec^{-1}
Z_u	Dimensional variation of Z-force with speed	sec^{-1}
$Z_{\delta_{E,c}}$	Dimensional variation of Z-force with elevator or canard angle	ft sec^{-2}
Z_o	Vertical force equation bias	ft sec^{-1}

Greek Symbol

α	Angle of attack	rad
β	Angle of sideslip	rad
ψ	Euler heading angle	rad
θ	Euler pitch angle	deg, rad
ϕ	Euler roll angle	deg, rad
$\dot{\theta}_o$	Bias in Euler pitch rate equation	rad sec^{-1}
δ_E , δ_e	Elevator angle	deg, rad
δ_A , δ_a	Aileron angle	deg, rad
δ_R , δ_r	Rudder angle	deg, rad
δ_c	Canard angle	deg, rad
ρ	Air density	slugs ft^{-3}
$\dot{\varphi}_o$	Bias in Euler roll rate equation	rad sec^{-1}
$\omega_{n_{SP}}$	Undamped natural frequency of the short period mode	Hz

LIST OF SYMBOLS (continued)

Symbol	Definition	Dimension
ω_{n_P}	Undamped natural frequency of the phugoid mode	Hz
ω_{n_D}	Undamped natural frequency of the dutch roll mode	Hz
$\{\eta\ (t)\}$	Noise vector	
∇_c	First gradient with respect to c	
$\nabla_c{}^2$	Second gradient with respect to c	

Subscript	Definition
1	Initial
B	At body axis at center of gravity
M	As measured by transducer
I	As installed wrt body axis at center of gravity
L	Left hand
R	Right hand
,s	Flight stability axes
,w	Wind axes
,wt	Wind tunnel stability axes

Superscript	
†	Transpose
'	State vector derivatives

A dot over a quantity denotes the time derivative of that quantity.

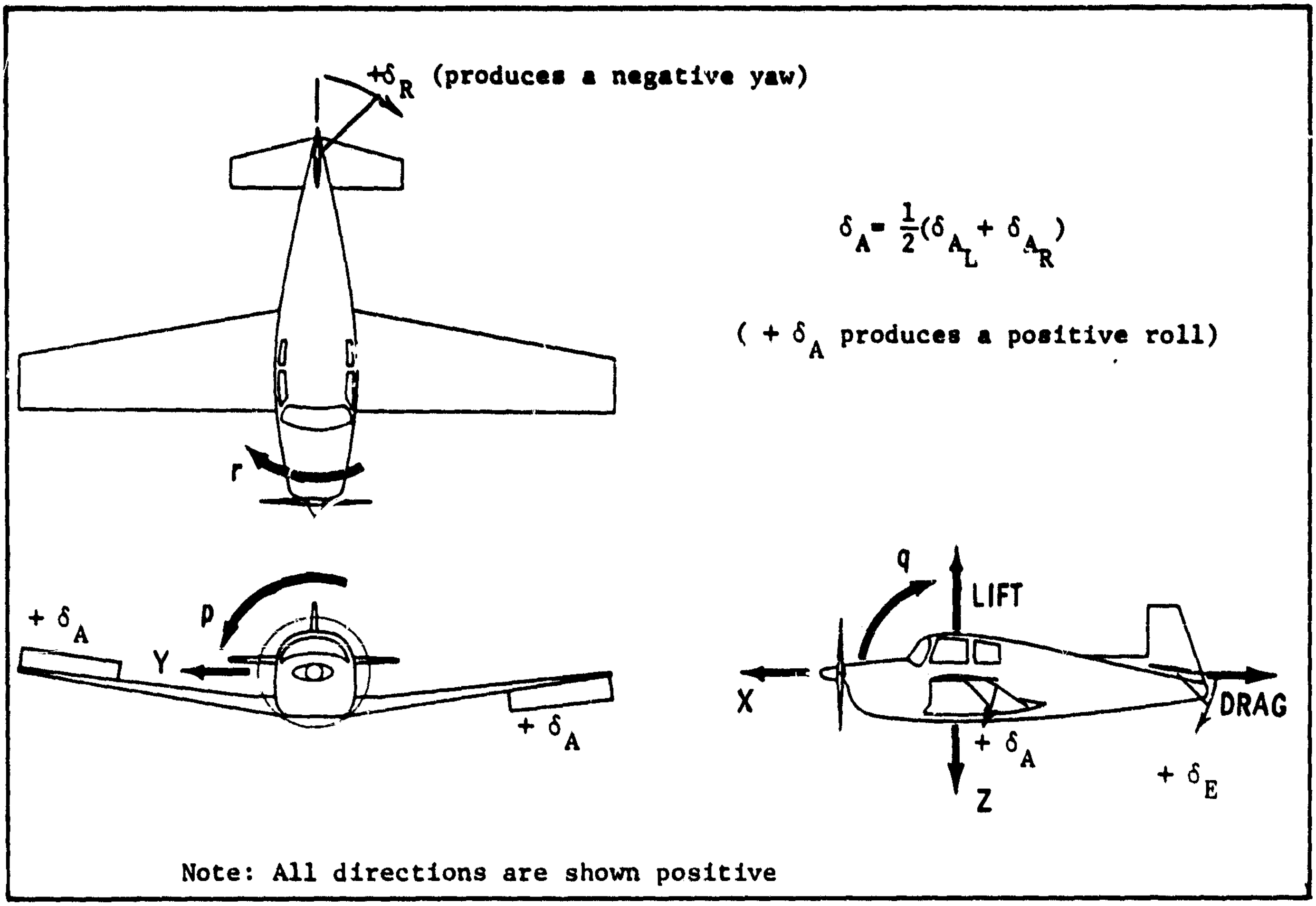


Figure i.1 Body axes system used in this report

PRECEDING PAGE BLANK NOT FILMED

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

PRECEDING PAGE BLANK NOT FILMED

LIST OF TABLES

1. INTRODUCTION

This report describes work completed during the second phase of a continuing program sponsored by the NASA Dryden Flight Research Center.* This program was accomplished during the period January 21, 1979, through February 15, 1981. The program encompasses the development of a simple, self-contained flight test data acquisition system. To date the program has consisted of two phases:

PHASE I

- A literature survey of flight testing methods (presented in Reference 1).
- The development and testing of a proof-of-concept system capable of longitudinal stability analysis (presented in Reference 2).

PHASE II

- Development and testing of a system capable of longitudinal and lateral stability analysis.

This report describes in detail the system concepts selected, as well as results of the flight test program used to show the validity of these concepts, as of the completion of Phase II.

The purpose of this project, and the design criteria developed are contained in Chapter 2. The literature survey (Reference 1) has been used as a primary data base for establishment of these

---

*Funding provided under NASA Grant NSG 4019 (FRL/CRINC 4070).

criteria. Other inputs have come from talks with personnel in the general aviation industry and of NASA Dryden Flight Research Center.

Chapter 3 describes the hardware selected and manufactured to meet the design requirements.

The instrumentation package employs transducers to allow both longitudinal and lateral stability analysis of general aviation type airplanes, although it can easily encompass most other types of airplanes. Due to the nature of the data reduction method utilized, a minimum number of high-accuracy transducers are required. Data from the transducers are recorded using an on-board microprocessor and digital cassette recorder. This has proven a simple, reliable method to obtain accurate flight data.

The system has been designed to allow it to be placed in the aircraft with a minimum amount of aircraft modification. A rechargeable battery pack was selected for airborne power to reduce the number of airplane modifications required. This has allowed total isolation from the aircraft electrical systems, which simplifies installation, enhances safety, and eliminates many electrical noise problems in the transducer signals. The transducers are all contained in one module, except for the following:

- total pressure probe,
- temperature probe,
- static pressure probe, and
- control position transducers.

A minimum of installation is also required for these devices, as they are literally "sticky-taped" to the airframe.

Presented in Chapter 4 is an evaluation procedure used to select a ground-based data reduction computer. Due to the extensive mathematical procedure used for data analysis, a powerful, high-level language microcomputer is required. The evaluation method used is described here, as well as the computer selected and used for this program.

The total flight testing process is included as Chapter 5. Discussed are the various computer programs and operating techniques developed. The heart of this system is the Modified Maximum Likelihood Estimation (MMLE) method which has been used for data reduction. The mathematics of this technique are included as Section 5.6.

The flight test program used for system development is included in Chapter 6. Tests have been performed using the KU-FRL* Cessna 172 airplane (shown in Figure 1.1). The type of flight test maneuver required is discussed, and results of the actual flight testing are presented.

A flight test program was conducted at Cessna Aircraft to evaluate the spin properties of their model 172 airplane. The data management portion of the KU-FRL system was used in conjunction with Cessna-supplied transducers for data acquisition and analysis. This program is described in Chapter 7.

Conclusions to be drawn as a result of the work carried out under this program, and recommendations for further work are included in Chapters 8 and 9.

---

*KU-FRL = University of Kansas Flight Research Laboratory.

Figure 1.1 Experimental configuration of the test airplane.

References, and reports describing this project are presented in Chapter 10.

Appendix A includes descriptions of all programs required for system operation.

There appears to be some confusion over the many reference axes systems used in airplane analysis. Included in the list of symbols is Figure 1.1, which explicitly defines the axes system used in this report. Appendix B is included to allow conversion of results in this report to several other standard axes definitions.

- - - - - ********** - - - - -

The system constructed under this program has proven under flight test the validity of the concept selected for longitudinal and lateral stability analysis. Throughout this flight test program, using the Cessna 172 airplane, the acquisition package performed reliably.

PRECEDING PAGE BLANK NOT FILMED

2. PURPOSE OF PROJECT

Flight testing has always required a high degree of complex instrumentation to get accurate results. This, in the past, and still evident today, has taken a great deal of time and money to equip each individual flight test article. Traditional systems are placed on aircraft on an individual basis, utilizing what is available at that time, coupled with the specific requirements of a particular test program. This has never really led to ideal or totally thought-out systems, and normally results in high costs or in too much time being required for instrumenting the airplane.

With the accurate instrumentation available today, and with the recent advances in microcomputer technology, it was seen that an accurate, multipurpose data acquisition system could be developed. The system described here has been developed to do just that.

The basis for design of this system is as laid out here.

EASE OF INSTALLATION - This has been a major design consideration. If possible, NO permanent modification should be done to the airplane. The system must be universally easy to install and should require a minimum of installation time and no special procedures. This factor includes calibration of the system installed on the airplane.

SELF-CONTAINED - The system should be totally self contained. This should include all data sources, data recording methods, power requirements and data reduction techniques.

SIMPLE - The system must be simple in concept and easy to use. The need for complex instrumentation, difficult calibration, and specialized operator knowledge must be kept to a mimimum.

FLIGHT TESTING - The system should not require any specialized piloting techniques to obtain accurate results.

CLASS OF AIRCRAFT - The system to be developed is primarily applicable to the general aviation type airplane. This criterion does not restrict the methods and theories, but it does define the requirements for the transducer ranges and accuracies.

RESULTS - The system is aimed at stability and performance parameter identification, but it must permit adaptation to other test requirements.

COSTS - The system should meet all of the above requirements, yet reduce the expenditure required for the instrumentation system as compared with current methods.

- - - - - ********** - - - - -

The system described in this report has been developed to prove that the concepts selected meet the above design requirements.

3. INSTRUMENTATION SYSTEM

The system described and constructed under this phase meets the objectives stated in Chapter 2. The instrumentation system can be broken up into four parts:

1) Data Management;
2) Transducers;
3) Power Supply;
4) Pilot Control.

The package is shown in the block diagram of Figure 3.1. The system is used in two forms: airborne for recording of flight data (Figures 3.2 and 3.3), and the ground-based portion for data transfer to the data reduction computer (Figures 3.4 and 3.5).

Installation of this system is straightforward and requires no permanent modifications to the airplane. The major modules are shown installed in the KU-FRL's Cessna 172 in Figures 3.6 and 3.7. The other components are shown installed on the airplane as they are described in Section 3.2. It is seen that the major modules are essentially strapped into the cabin compartment. The transducer module does require a more rigid attachment and is, therefore, held firmly in place by clamping it to the seat tracks.

Following is a detailed description of the instrumentation system, as well as the trade-offs considered in its design.

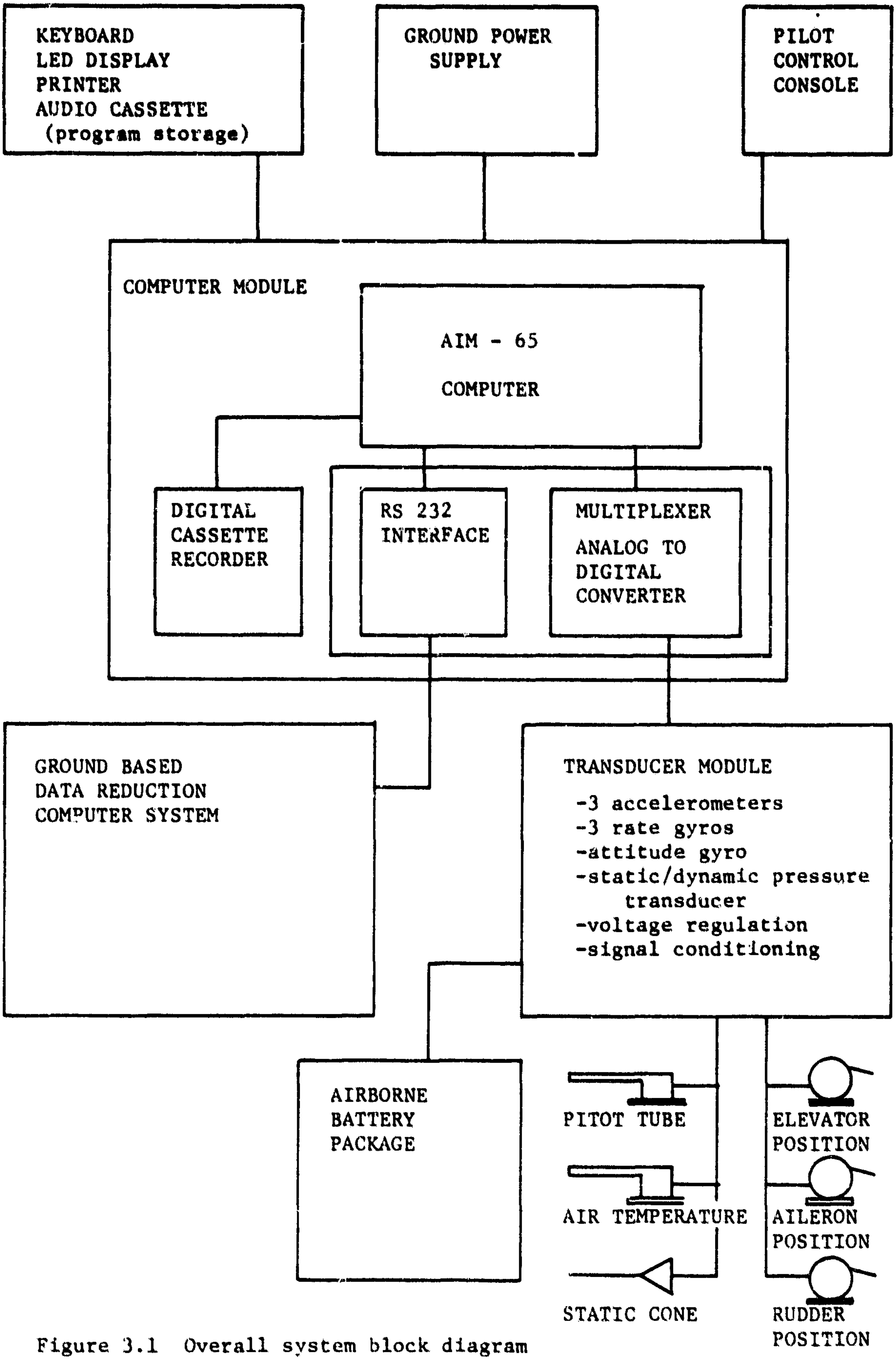
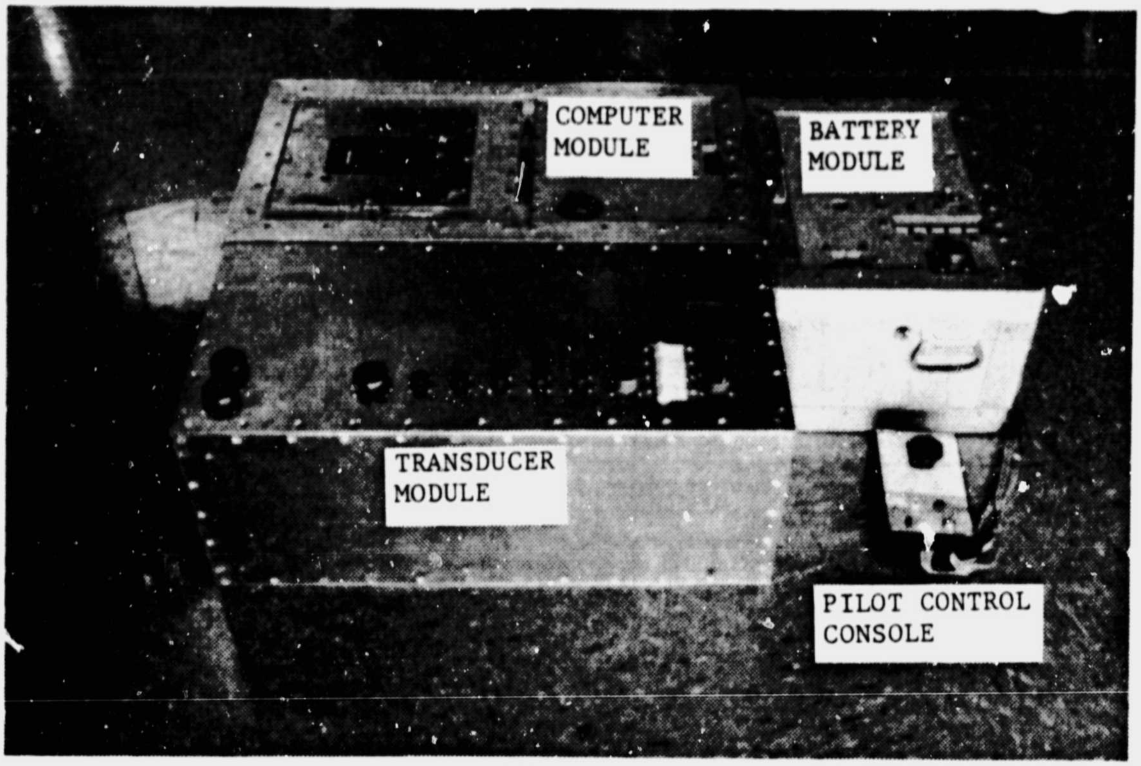


Figure 3.1 Overall system block diagram

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



WEIGHTS

BATTERY MODULE	60.5 (lb.)
COMPUTER MODULE	34.5
TRANSDUCER MODULE	32.5
CONTROL POSITION TRANSDUCERS *	1.5
PITOT PROBE *	0.2
TEMPERATURE PROBE *	0.2
PILOT CONTROL CONSOLE	1.0
MISCELLANEOUS (cables,clamps,etc.)*	2.0
TOTAL:	132.4

*not shown

Figure 3.2 Major components of the airborne system

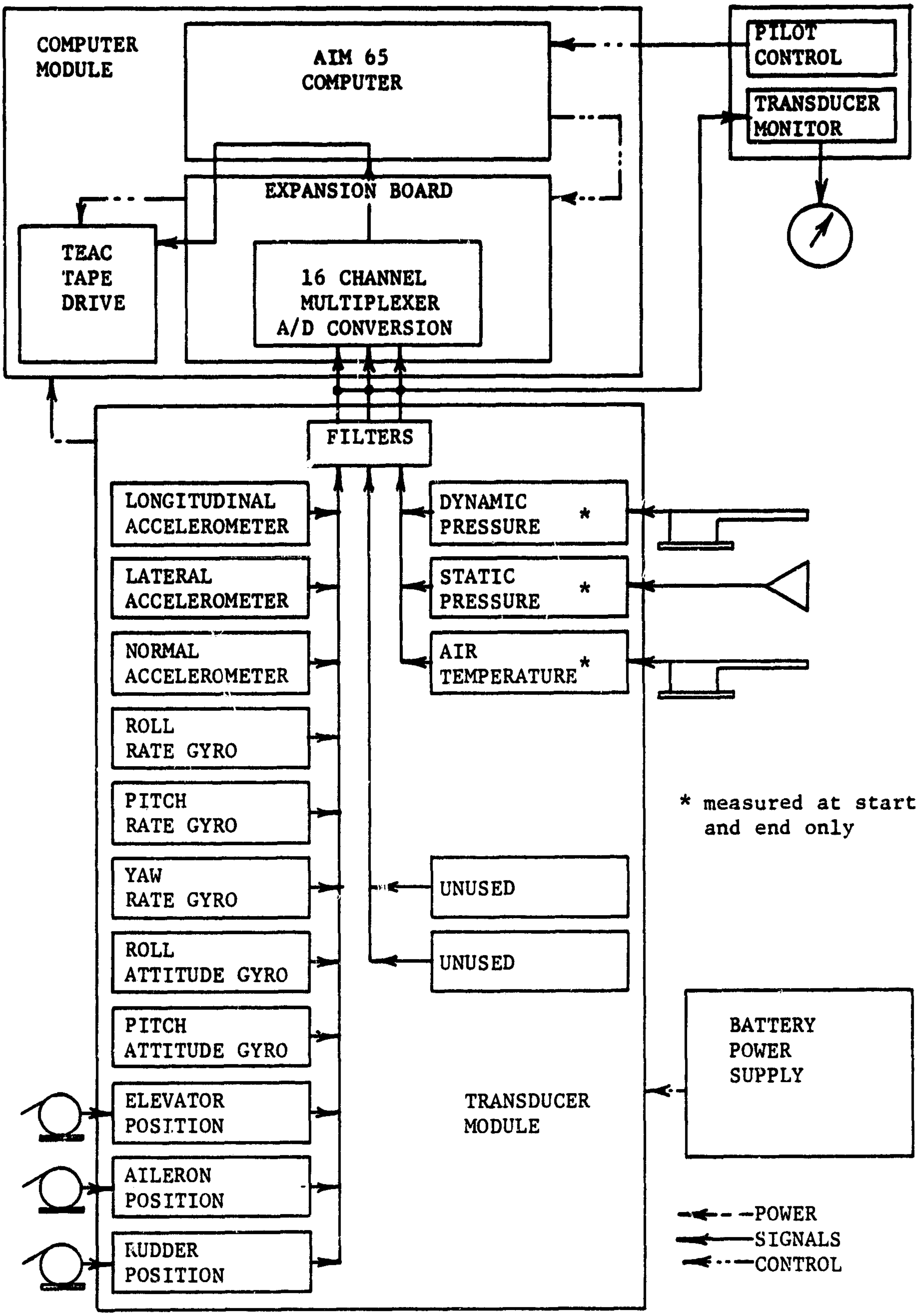


Figure 3.3 Block diagram of airborne system

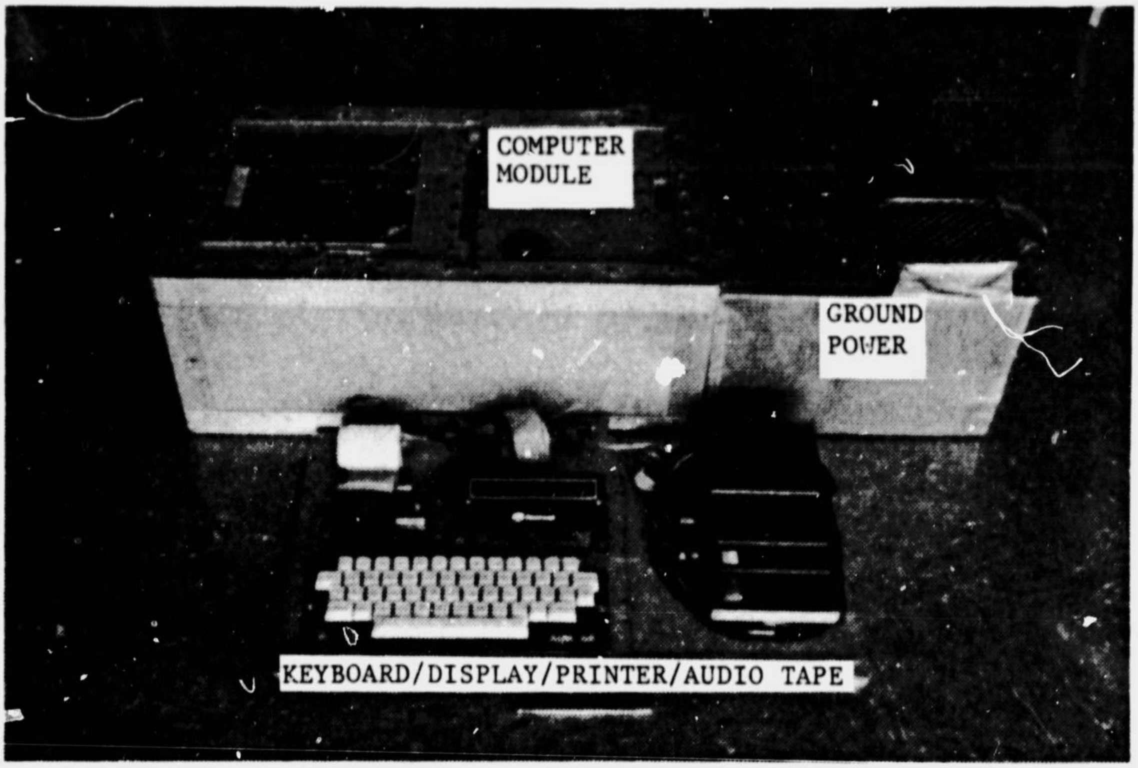


Figure 3.4 Data transfer system

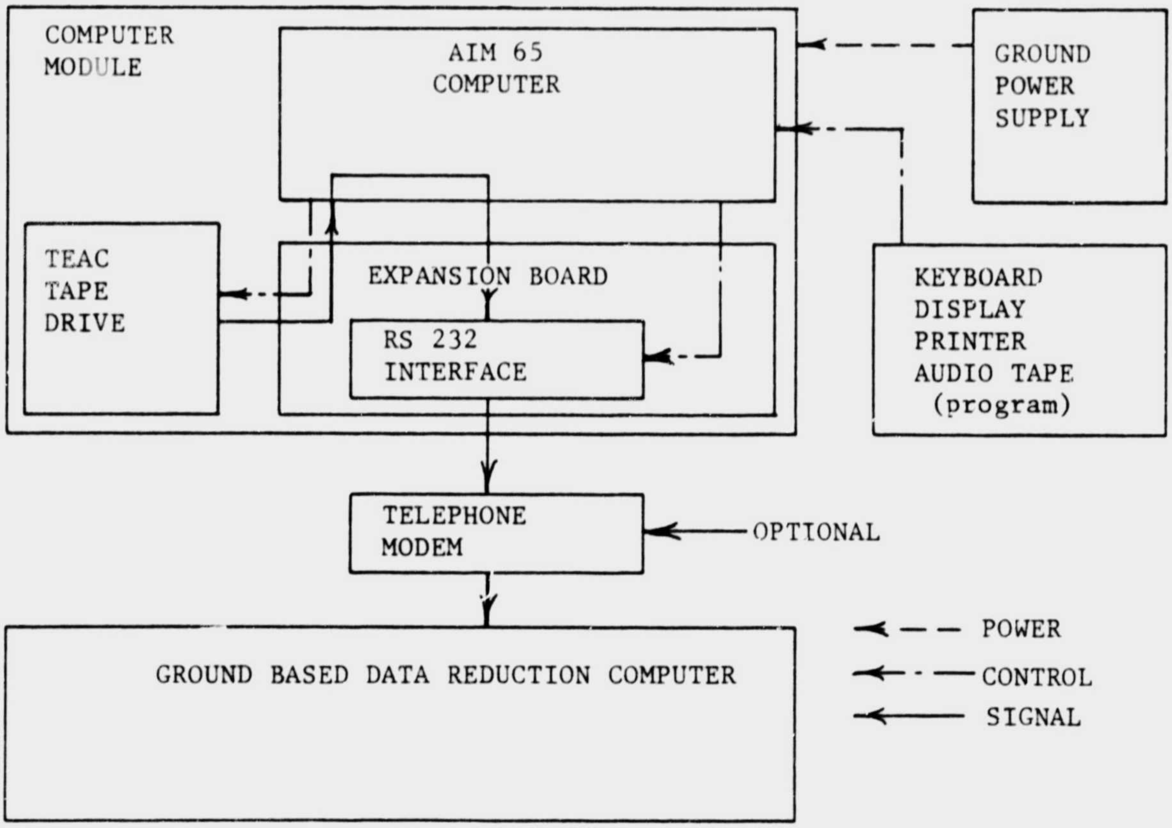


Figure 3.5 Block diagram of data transfer system

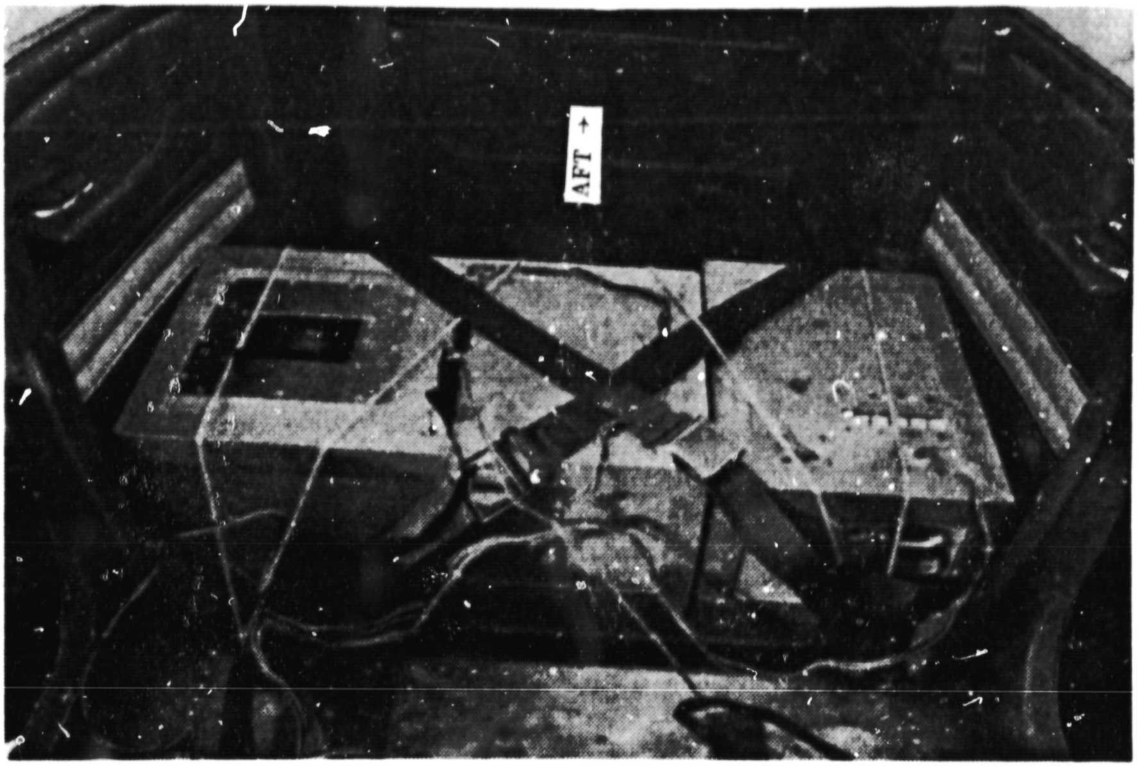


Figure 3.6 Battery and computer module installation

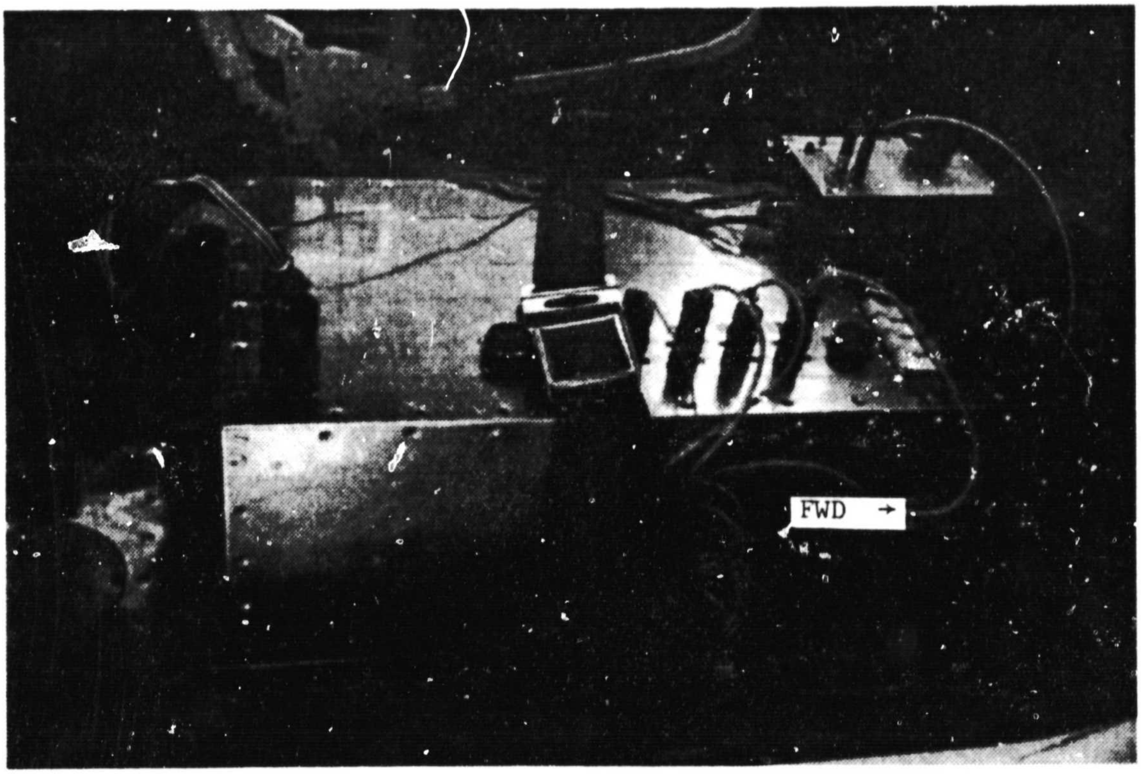


Figure 3.7 Transducer module installation

3.1 Data Management

It was decided to use a microprocessor controlled data management system. Using a commercially available computer simplified the design task, as well as reducing overall system complexity and cost. Also with this type of controller, versatility is easily achieved, especially if programs are stored on cassette tape rather than in computer hardware. The trade-offs considered prior to selecting this data management system were those of analog vs digital data storage, and airborne recording vs telemetry. Following is discussion of these trade-offs and a detailed description of the system constructed.

In the past, most on-board systems made use of analog recording, due primarily to the high cost and complexity of digital systems. In recent years, however, progress has been made in the digital field, resulting in small, inexpensive, and reliable digital devices, most available as solid-state integrated circuits. The recent advances in digital electronics technology have reduced both the complexity and cost. Coupling this with the lower likelihood of error in digital systems, it was decided to use a totally digital system for this package.

In the past, telemetry has been a much-used means of transmitting data to be recorded on the ground. Telemetry has an important place in aircraft flight testing, specifically in high-risk operations (such as flutter testing, spin testing, etc.). Its major disadvantages are the requirement of a ground station, and the associated high cost and complexity. Telemetry, however, has been primarily used in the past, due to the large size, complexity, and inaccuracies of the older recording media. Many improvements have been made in this regard with the introduction of small, reliable cartridge and cassette recording systems. This improvement is largely attributed to the recent advances in solid state electronics technology. For this system, on-board recording, making use of a digital cassette recorder, has been chosen.

The heart of the unit constructed is a Rockwell AIM 65 microcomputer. This is coupled through a Rockwell expansion interface to the other components. The other two major components of the airborne package's recording system are the Datel MDAS-16 multiplexer and analog-to-digital converter, the TEAC MT2-02 digital cassette tape transport, and RS232* interfacing port. These are shown in the block diagram of Figure 3.8.

The AIM-65 is an interactive single board computer using an 8-bit 6502 microprocessor. Contained on the computer board is 4K bytes of memory, as well as a monitor and symbolic assembler. (An 8K BASIC programming ROM** is also available for this computer.) A 20 character display, 20 column thermal printer and alphanumeric keyboard allow the user to interact with the computer. Two application connectors increase the computer's versatility. One allows interfacing to audio cassette recorder and other computer terminals. The second allows adding an expansion interface which facilitates

---

*RS232 = serial interfacing standard.

**ROM = read only memory.

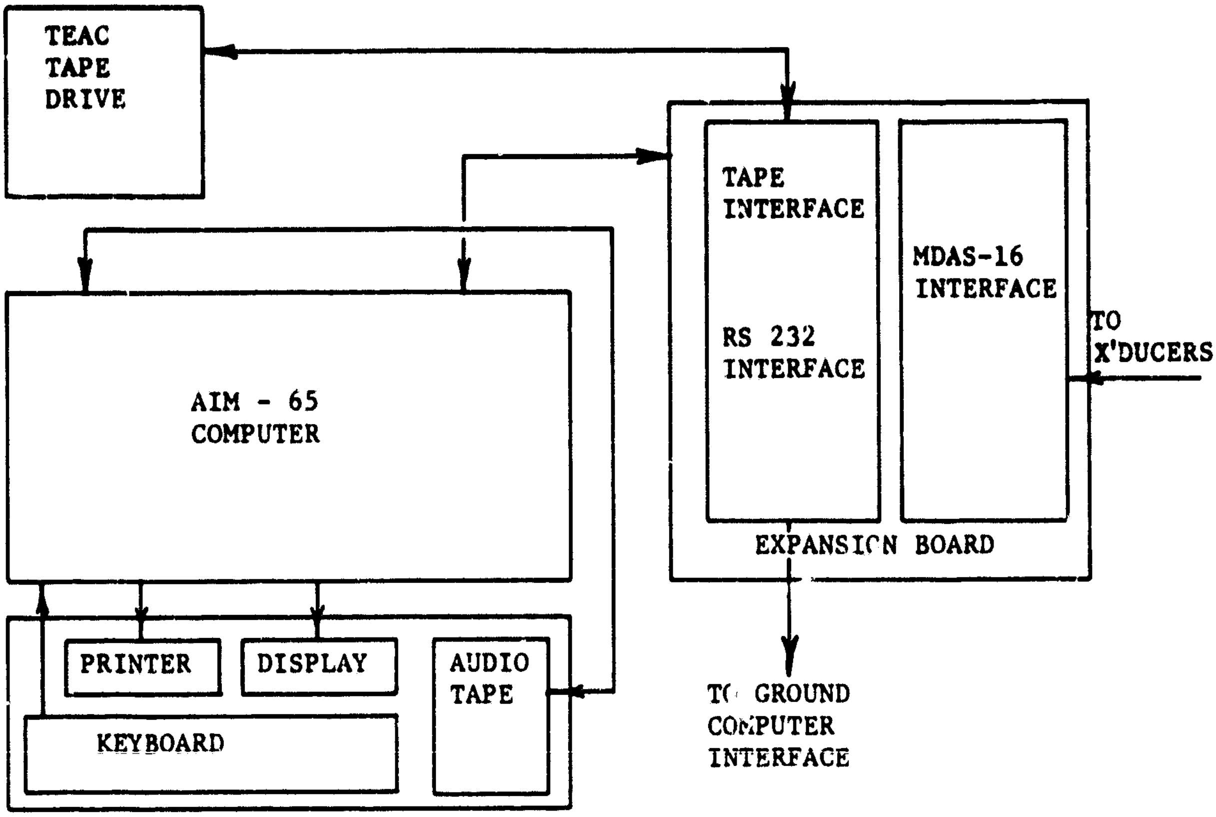


Figure 3.8 Computer module block diagram

additional features to be adapted to the standard computer. These features provide an easy-to-use Data Management Controller.

The user is able to easily program the computer using the symbolic assembler and monitor functions provided. Programs presently are stored on the audio cassette recorder. Using the additional ROM slots on the computer, or the addition of a ROM board on the expansion interface would allow regularly used programs to be permanently placed in the system.

The AIM-65 is connected, through the expansion board, by use of the MDAS-16, to the transducer package. The MDAS-16 is a 16-channel multiplexer coupled with a 12-bit analog-to-digital converter. This unit has the capability of addressing channels as desired

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

(either randomly or sequentially), using a microprocessor controller. Voltage input ranges can be selected (-5 volt to +5 volt was chosen for this syrtem). The unit has a 50 KHz through-put rate with 20 μ sec access time per channel. The MDAS-16 is shown in Figures 3.9 and 3.10. The MDAS-16 does require calibration. This procedure is described in detail in Reference 3.

The other major component of the data acquisition system is the TEAC tape transport (see Figure 3.11). This unit is a low-cost magnetic tape unit designed specifically for digital applications. It makes use of standard audio type cassette tapes for data storage. All interfacing required is included in the

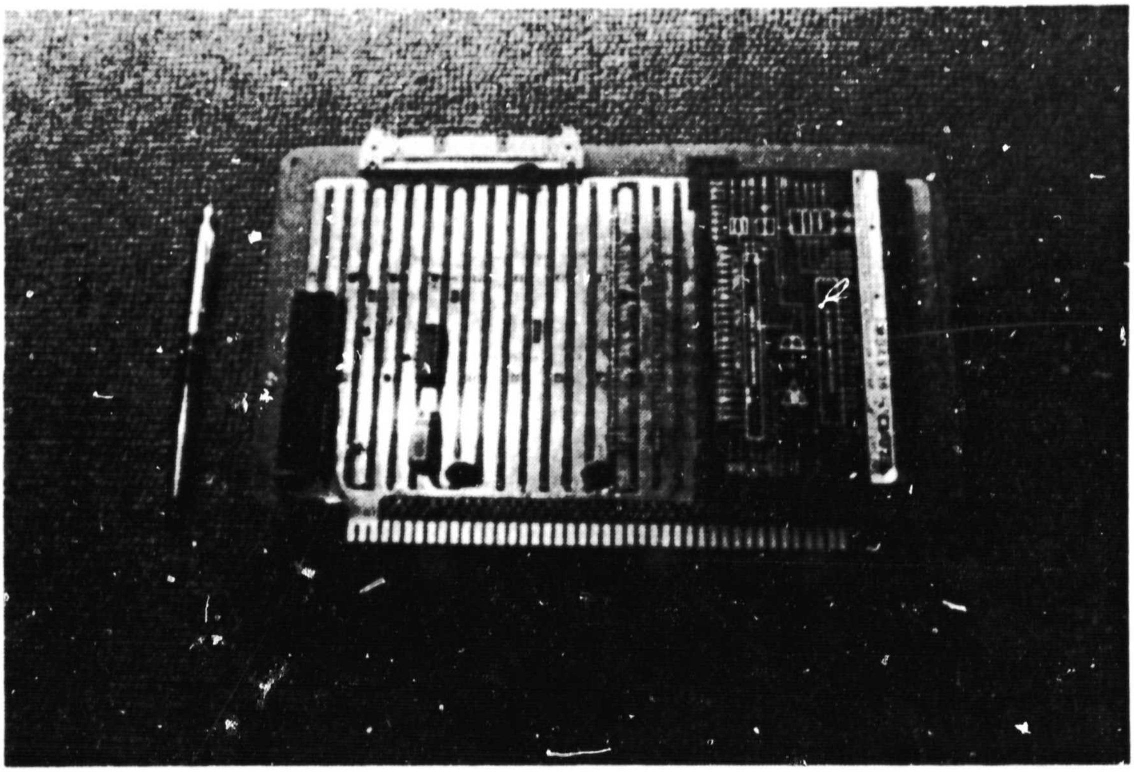


Figure 3.9 MDAS-16 data acquisition card

ORIGINAL PAGE IS
OF POOR QUALITY

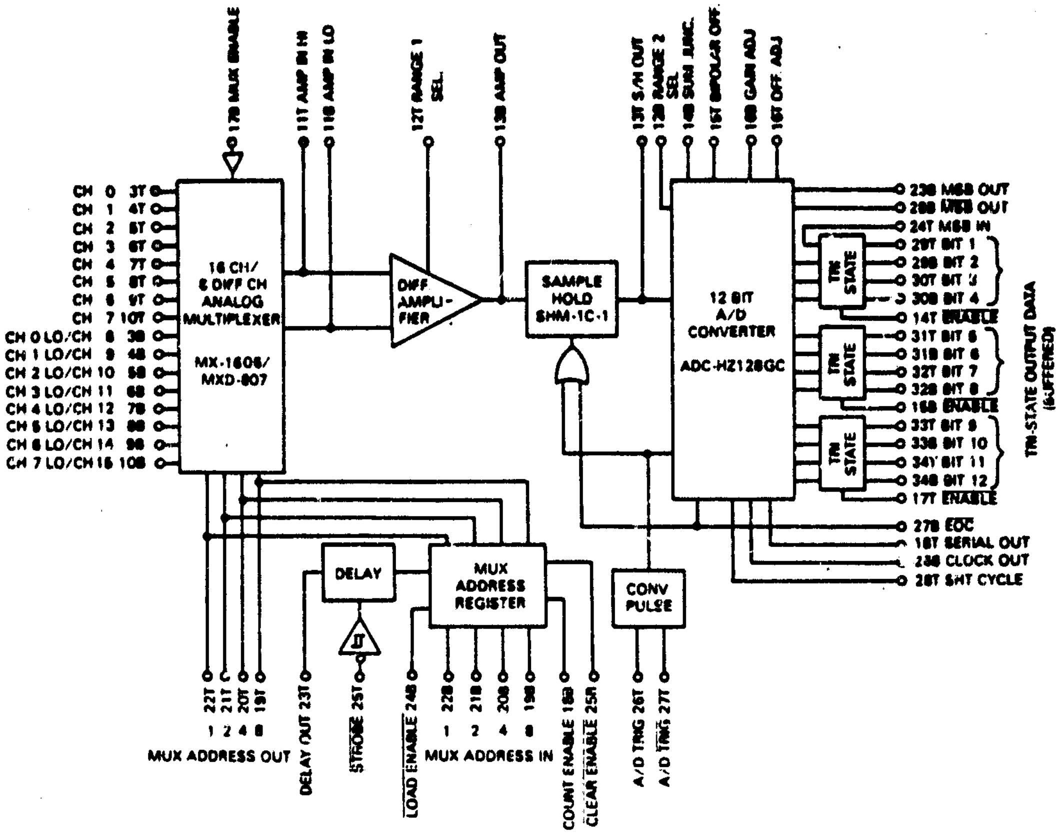


Figure 3.10 Block diagram of MDAS-16 data acquisition module

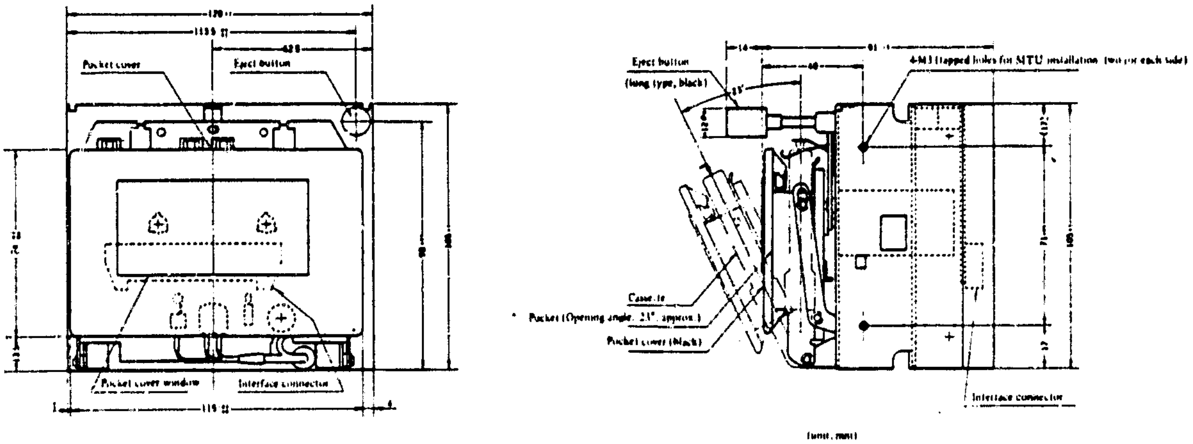


Figure 3.11 TEAC MT2-02 digital cassette tape transport

package. Input requirements are TTL[*]-compatible; and the tape unit requires only control signals, provided by the AIM-65 microcomputer, and parallel data input. All detailed control functions required by the tape unit are handled on board by the unit for both recording and playback. Only simple control signals are required to initiate the various functions.

The data management system is also used for data playback and transfer to the ground-based data reduction computer. It was decided to use the same recorder and computer system for playback of data and in-flight recording. This avoids possible problems due to mismatch of tape drives and also reduces overall system costs. An interface system compatible with standard computer RS232 ports was designed and constructed. A hard wire connection, or use of a modem through the telephone can thus be utilized. This type of interface allows data transfer to virtually any computer. A program on the AIM-65 controls the TEAC tape transport and sends the data over the line to the other computer. Once all the data are on the other computer, the Rockwell system is no longer required in the data reduction process. (See Chapter 5 for a complete description of the data reduction process.)

3.2 Transducers

At the outset of this program, it was decided to keep the number of containers in the total system to a minimum. Thus, most transducers, as well as their required signal conditioning, are

---

[*] TTL = Transistor-Transistor-Logic: Electrical standard.

contained in the transducer module. This module contains all filtering, all voltage regulation, and the transducer pallet. It was not possible to place all transducers in this module, as measurements such as control positions and outside air conditions were required to be measured. Following is a description of the methods used for selecting transducers required, as well as descriptions of the actual equipment selected.

The primary input to aid in the selection of the parameters to be measured was the literature describing the data reduction methods to be used (References 4-20). The transducers discussed in the references above are summarized in Table 3.1. Discussion with personnel at NASA, Dryden Flight Research Center, was the secondary input for transducer selection. The transducers selected allow optimal use of the data reduction technique considered (basically a maximum likelihood parameter estimation method; see Chapter 5 for a detailed description of this method).

The literature (References 4-17 and 20) was also used as the primary reference for selection of transducer accuracies required. The results are summarized in Table 3.2. The transducer ranges were selected after discussion with the general aviation manufacturers (the secondary reference), and consideration of the performance characteristics of this class of airplane.

The ranges and accuracies required for the various transducers selected are summarized in Table 3.3.

Table 3.1 Transducers used in various flight test programs

	SMETANA ref.18	DELFT ref. 4-11	SORENSON ref. 19	BONES PROGRAM ref 15	KLEIN ref. 17	SELECTED	
A_x	*	*	*	*	*	*	
A_y		*	*	*	*	*	
A_z		*	*	*	*	*	
V_x							
V_y							can be derived
V_z							
Alt.							
Temp.		*				*	
θ	*		*	*	*	*	
ϕ		*	*	*	*	*	
ψ							not normally needed
p			*	*	*	*	
q	*	*	*	*	*	*	
r		*	*	*	*	*	
$\dot{p}$			*	*			
$\dot{q}$			N.R.	*			can be derived
$\dot{r}$			*	*			
δ_E		*		*	*	*	
δ_A				*	*	*	
δ_R				*	*	*	
RPM		*					may be req'd or desireable for performance data
M.P.		*					
P_S		*			*	*	
P_T							can be derived
P_D	*	*	*	*	*	*	
α	*		*	*	*		
β			N.R.	*	*		can be derived
ρ	*						

TABLE 3.2 Transducer accuracies used in various flight test programs

	DELFT ref. 4-11	ECKHOLD & WELLS ref 20	KLEIN ref 17	ILIFF & MAINE ref 12-16	SELECTED
A_x	.001 g	.002 g	.005 g	0.1 % of full scale	.002 g
A_y		.02 g	or		.002 g
A_z		.02 g	2 %		.002 g
θ		1/2 °	.2° or		.5°
ϕ		1/2 °	2 %		.5°
p		.15°/sec	.2°/sec		.5° /sec
q	.02°/sec	.15°/sec	or		.5° /sec
r		.15°/sec	2 %		.5° /sec
δ_E		.4°	.2°		.5°
δ_A		.4°	or		.5°
δ_R		.4°	2 %		.5°
T		2° F			2° F
P_S	.1 m 160 ft	10 ft			10 ft
P_D		5 knots	2 knots		2 knots

Table 3.3 Transducer Accuracy and Range Used

Symbol	Sensor		Accuracy	Range
A_X	longitudinal acceleration		.002 g	±1 g
A_Y	lateral acceleration		.002 g	±0.5 g
A_N	normal acceleration		.002 g	-1.5 g to 4 g
θ	pitch angle		0.5°	±30°
ϕ	roll angle		0.5°	±30°
p	pitch rate		0.5°/sec	±50°/sec
q	roll rate		0.5°/sec	±50°/sec
r	yaw rate		0.5°/sec	±50°/sec
δ_E	elevator position		0.5°	
δ_A	aileron position		0.5°	
δ_R	rudder position		0.5°	
T	temperature	*	2°F	-65 to +120°F
P_S	static pressure	*	10 feet	0 to 25K feet
P_D	dynamic pressure	*	2 knots	40 to 150 knots

*Indicates transducers used to define initial and final conditions.

During a specific maneuver, T, P_S and P_D need only be measured at the start and finish to define the initial and final conditions. The other 11 channels require measurement throughout the maneuver to determine the dynamic characteristics and analyze stability and performance properties of the airplane.

To select the data acquisition rate required, the following factors must be considered:

- Minimum rate must be higher than the undamped natural frequency of the airplane to be tested.
- Minimum rate must be high enough to avoid time skewing of the data points.
- Minimum rate must be as low as possible to allow economy in the recording media and data reduction process.

In data analysis, to obtain reasonable representations of the frequency response, an acquisition rate of at least five times the undamped natural frequency should be used (Reference 21, Volume 1, Chapter 6). In the class of aircraft considered for this instrumentation system, the natural frequencies are of the following order (from Reference 22):

$\omega_{n_{SP}}$	0.5 - 1.0 Hz
ω_{n_P}	0.01 - 0.03 Hz
ω_{n_D}	0.25 - 0.60 Hz.

Therefore, the maximum frequency ($\omega_{n_{SP}}$) requires an acquisition rate of

$$1.0 \times 5 = 5 \text{ samples/sec.}^{*}$$

This is the minimum data requirement.

From References 12 and 14 and discussion with the authors it was determined that an acquisition rate of 100/sec is required to avoid time skewing problems. From the practical applications of the maximum likelihood estimation method, this rate (100/sec) also results in an excess of data that unnecessarily increases the computation time and costs.

Using a computer-controlled acquisition system allows scanning of the transducers as rapidly as possible (20 μ sec/channel, 220 μ sec total**), and then waiting until the next data point is required

---

[*] 10 samples/sec was chosen for the KU-FRL system, as this then definitely meets the minimum data requirement. This rate also seems to be somewhat of an acceptable industry standard.

[**] Values for the KU-FRL system.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

(0.1 sec later[*]). These data are temporarily stored in memory and then output to the TEAC tape. This technique allows a high scanning rate to avoid time skewing between channels (equivalent to 4545/sec[*]) and a low overall acquisition rate (10/sec[*]) to provide economy and still satisfy the minimum data requirement.

The transducers were primarily mounted on one pallet. This is shown in Figure 3.12. It was possible to include most transducers on one pallet with the exception of the

- pitot tube,
- temperature probe,
- static cone, and
- control position transducers.

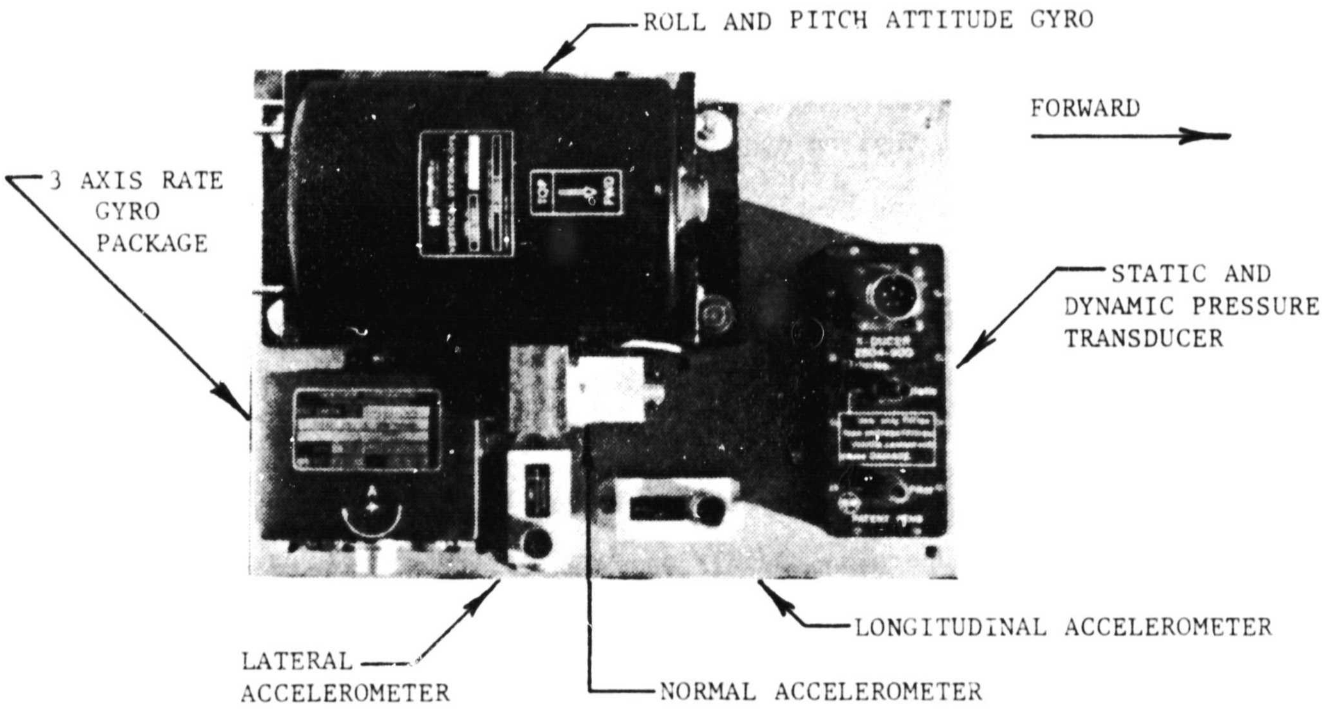


Figure 3.12 Transducer pallet

---

[*] Values for the KU-FRL system.

The pallet, contained within the transducer module, was mounted as close to the center of gravity of the airplane as possible. In this flight test program the transducer module has been clamped to the seat tracks of the Cessna 172, in the copilots's position.

- - - - - ********** - - - - -

Following are descriptions of the individual transducers used in this program.

3.2.1 Accelerometers

The accelerometers used in this package are of the force feedback (or closed loop) type. This type of accelerometer derives its measurement from determining the force required to maintain a mass at a zero location. This technique reduces the errors caused by mass displacement and also does not rely on springs (and their associated inaccuracies) as do the displacement (or open loop) type accelerometers. The disadvantage to the force feedback accelerometer is its relatively high cost.

It is essential to note that linear (as opposed to vibration) accelerometers be used for this type of package.

The accelerometers chosen are manufactured by Schaevitz Engineering. Their specifications are shown in Figure 3.13. These accelerometers are intended for the measurement of linear accelerations such as required for guidance control systems, or vehicle ride analysis. Both a precision sensor and electronics are integrated into the ac-

ORIGINAL PAGE IS
OF POOR QUALITY

Specifications at 20°C

LSB Linear

Range g	Nominal Natural Frequency Hz	Nominal Output Impedance kilohms
± 0.25	50	20
± 0.5	70	10
± 1.0	100	5
± 2.0	110	2.5
± 5.0	125	5
±10.0	140	2.5
±20.0	160	5
±50.0	200	5

Input Voltage	±15V DC nominal
Input Current	10 mA DC maximum (6mA DC average)
Full-Range Open-Circuit Output Voltage	±5.0V DC
Damping Ratio	0.6 typical (0.3 to 1.0 on request)
Linearity (Notes 1 & 2)	±0.05% of full scale output
Hysteresis (Note 2)	0.02% of full scale
Resolution (Note 2)	0.0005% of full scale
Cross-Axis Sensitivity (Note 3)	±0.002 g per g up to ±10 g range, inclusive ±0.005 g per g over ±10 g range
Bias	Less than 0.1% of full scale
Sensitive Axis to Case Alignment	±1°
Noise Output	5mV rms maximum
Operating Temperature	−40°C to +95°C
Storage Temperature	−55°C to +105°C
Thermal Coefficient of Sensitivity	0.02% per °C
Thermal Coefficient of Bias	0.002% per °C
Shock Survival	100 g − 11 ms
Weight	3 oz.

SB Series

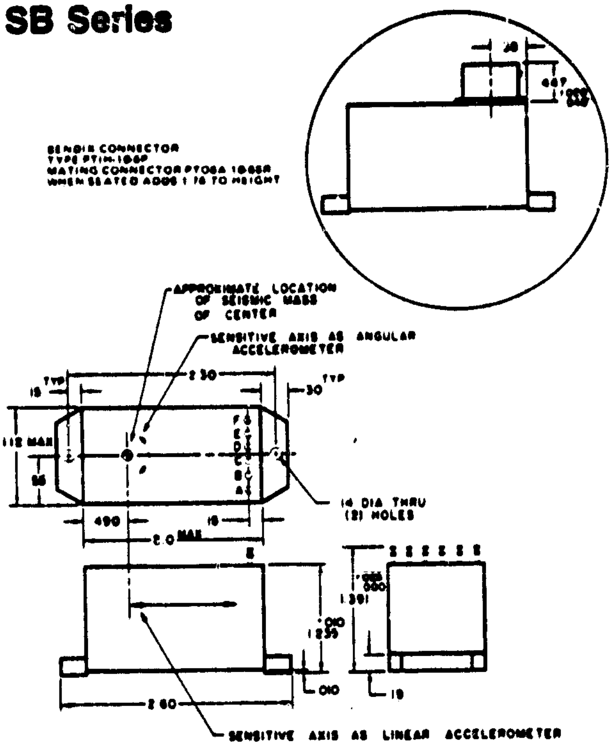


Figure 3.13 Schaevitz Engineering LSB series accelerometers

celerometer case. Interfacing is relatively simple, requiring only a DC input voltage, and then a measurement of the DC voltage output, corresponding to the acceleration sensed.

3.2.2 Filtering

The response characteristics of the accelerometers were such that they picked up the aircraft vibration caused by the engine.

The graph of Figure 3.14 shows the airframe vibration characteristics (measured using the accelerometers as transducers, and observing the output on an oscilloscope) as a function of engine speed. It is obvious from these curves that the vibration is caused by the engine and is a function of the engine speed. Also of note is the fact that all the vibration is at a frequency above 40 Hz.

A low pass filter with a cutoff frequency at 10 Hz would eliminate this vibration from the measurement signal. Using a two-pole, active filter with a response as shown in Figure 3.15 virtually eliminated this unwanted vibrational noise, yet leaves the desired measurement (occurring in the order of 1 Hz) essentially unchanged. (The measurements of the A_N accelerometer are presented as filtered and unfiltered measurements in Figure 3.16 to show this.)

In general, as was the case with this instrumentation package and the Cessna installation (see Chapter 7), only the accelerometers required any filtering.

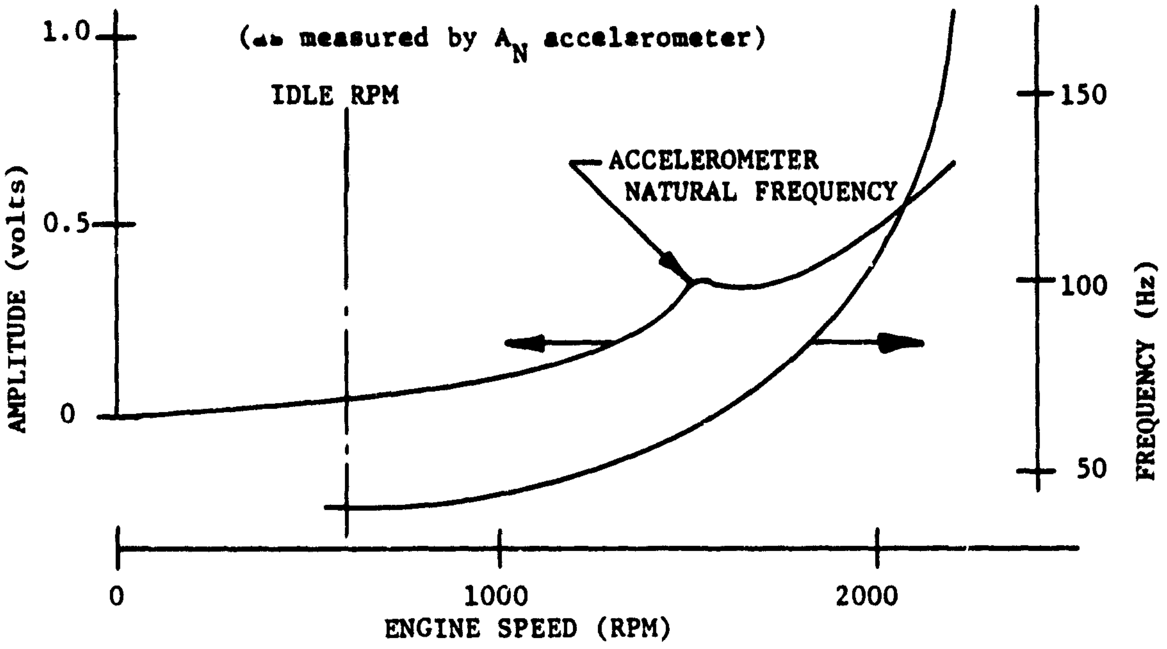


Figure 3.14 Measured airframe vibration

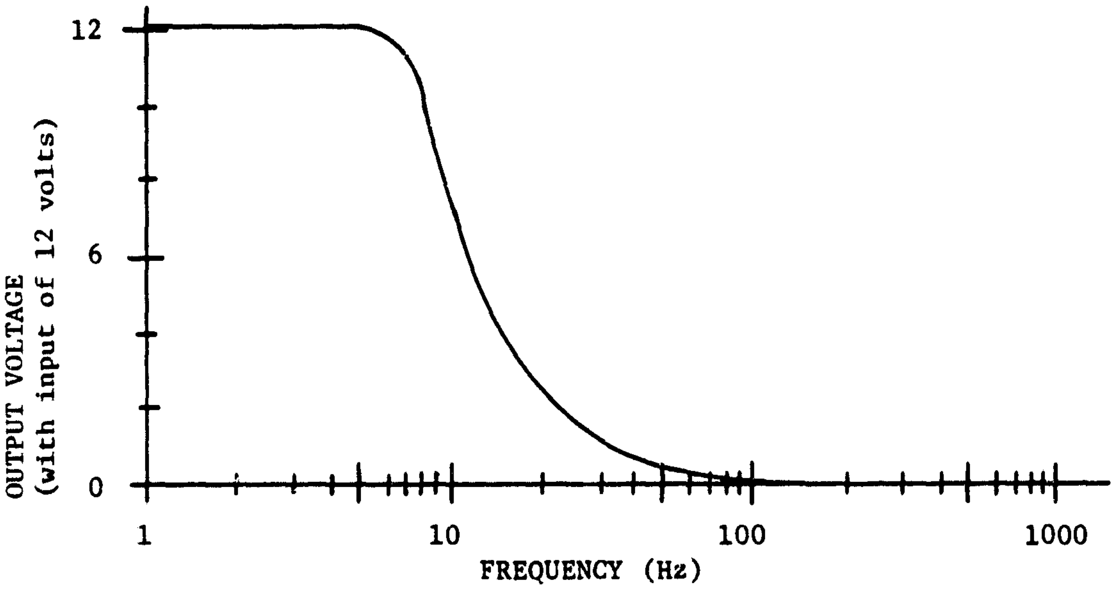


Figure 3.15 Measured filter frequency response

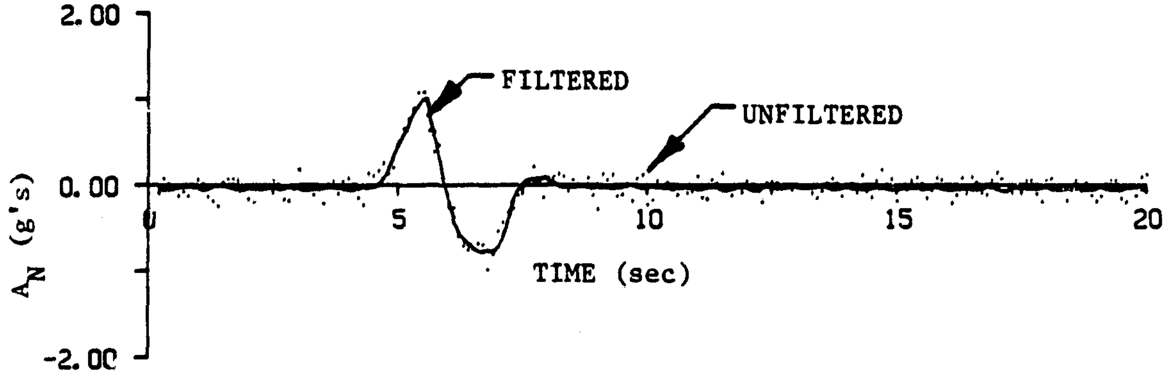


Figure 3.16 Filtered and unfiltered in flight measurements

One drawback of filtering signals is the introduction of a phase shift due to the filter. To counter this problem, all signals should be filtered the same amount, thus eliminating the problems of the phase shift.

3.2.3 Attitude Gyro

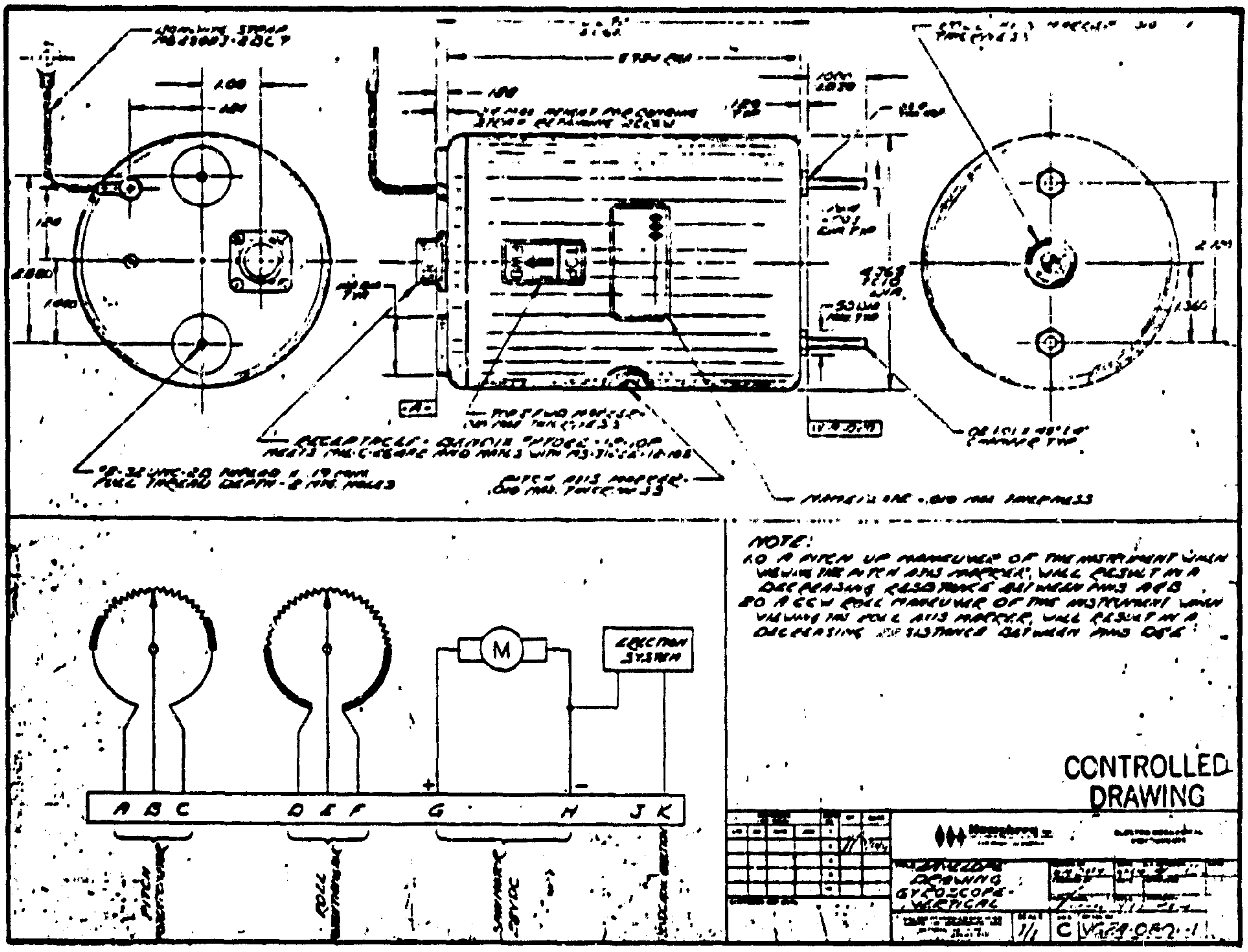
Both roll attitude and pitch attitude are obtained from a Humphrey VG-24 vertical gyroscope. Full specifications for this gyro are shown in Figure 3.17. This is a DC gyro, with potentiometers for determining the measurement (28 volt DC used for the motor, ±5 volt DC used for potentiometer excitation). This gyro has operated reliably during both phases of this program.

3.2.4 Rate Gyros

A three-axis DC/DC rate gyro package was used for roll-, pitch-, and yaw-rate measurement. The advantage of using a three-axis package rather than three separate gyros is that alignment for orthogonality upon installation is eliminated. Of course, failure of a single gyro will require the entire package to be removed for repair.

The gyros selected are of the displacement type (or open circuit). Closed circuit (or integrating gyros) will provide better accuracy; however, cost of these is approximately 10 times higher. The accuracy of a good quality displacement type gyro will meet the requirement (see Tables 3.2 and 3.3), especially considering the type of power input used (free of oscillations or any high frequencies).

**ORIGINAL PAGE IS
OF POOR QUALITY**



SPECIFICATIONS

RANGE - MECHANICAL	PITCH:±60° minimum ROLL: 360° continuous
- ELECTRICAL	PITCH:±60°, ±2.5° ROLL: ±90°, ±3.0°
OUTPUT	Potentiometer output
STATIC ERROR BAND	PITCH:±1.25% of full scale at 0° expanding linearly to ±2.08% of full scale at 60° ROLL:±0.83% of full scale at 0° expanding linearly to ±1.67% of full scale at 90°
RESISTANCE	1500 ±100 ohms
CONTACT RESISTANCE	2 ohms maximum at 20 mA
RESOLUTION	0.2% of full scale maximum
POWER DISSIPATION	1 watt at +165 °F
WIPER CURRENT	20 mA maximum
ELECTRICAL REQUIREMENTS	
SPIN MOTOR	
VOLTAGE	26 to 32 volts DC
CURRENT - STARTING	4.5 A maximum at 30 volts DC for 2.5 seconds
- RUNNING	1 A maximum at 30 volts DC
ERECTION	
VOLTAGE	26 to 32 volts DC
CURRENT	100 mA maximum intermittent
PERFORMANCE	
SPIN MOTOR TIME TO SPEED	5 minutes maximum
TIME TO ERRECT FROM MOTOR OFF	within 0.5° in 9 minutes
NORMAL OPERATING ERECTION RATE	2 to 9 °/minute after 3 min.
VERTICAL ACCURACY	within 0.5° of true vertical
FREE DRIFT RATE	0.5°/min. nominal; tested on ±3 1/2° Scorsby 6 min. run alternating

ENVIRONMENTAL CONDITIONS	
VIBRATION	vertical accuracy of ±2.0° shall be maintained during vibration of 0.01 inch D.A., 5 to 65 Hz; 2g, 65 to 500 Hz.
SHOCK	15g ; 11 msec; all axes
ACCELERATION - NON OPERATING	30 g; 1 min; vertical axis
- OPERATING	10 g; 1 min;applied in pitch or roll axis shall not produce a drift of greater than 10 °/min.
TEMPERATURE - OPERATING	-65 to +165 °F
- STORAGE	-80 to +185 °F
ALTITUDE	sea level to 40000 ft
SEA WATER IMMERSION	3 ft for 3 hr.
HUMIDITY	to 95% including condensation for 240 hrs.
SALT SPRAY	as encountered on shipboard or at coastal regions
SAND AND DUST	as encountered in desert regions
FUNGUS	external surfaces non-nutritive
EXPLOSION PROOF	shall not produce an explosion when operated in a fuel vapor rich area
RADIO NOISE INTERFERENCE	MIL-I-6181; paragraph 4.3.1 & 4.3.2
SERVICE LIFE	100 hrs minimum
SHELF LIFE	3 yrs minimum
INSULATED RESISTANCE	20 megohms minimum at 100 volts DC motor circuit exempt
WEIGHT	3.0 lb. maximum
SEALING	shall not leak under vacume equivalent to 40000 ft.

Figure 3.17 Humphrey VG-24 vertical gyroscope

The gyros selected are manufactured by Northrop. Voltage input required is 28 volts DC, and output voltage is from -5 to +5 volts DC. The gyros are Northrop G5 subminiature rate sensors. The gyro package specifications are included in Figure 3.18.

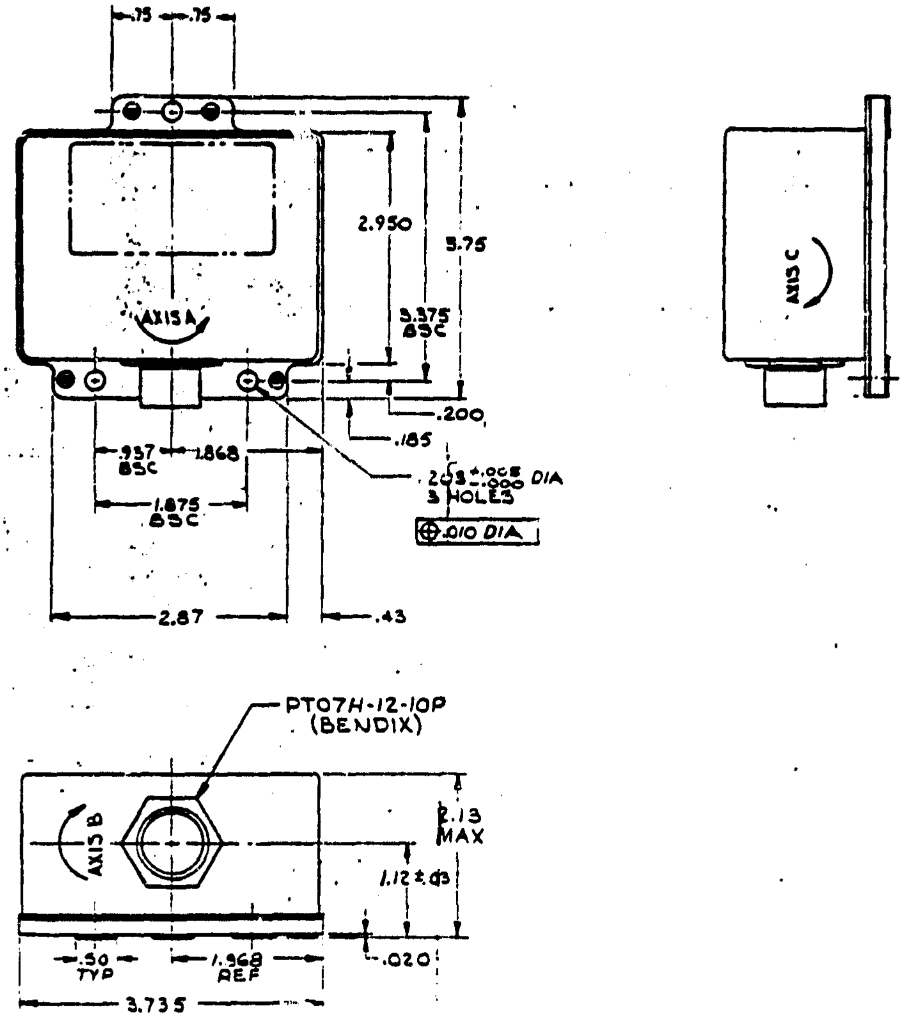
3.2.5 Control Position Transducers

Linear displacement transducers manufactured by Space-Age Control, Inc., were used to measure elevator position. This transducer is depicted in Figure 3.19. Due to the small size of this unit, it was decided to place it externally on the airframe. These transducers are installed as shown in Figures 3.20.

A novel technique for attaching the control position transducer (as well as the total pressure tube and temperature probe) has been used. Double-sided foam tape attaches the external devices onto the airframe. The mounting technique is depicted in Figure 3.21. The mounting method was first tested in the KU-FRL subsonic wind tunnel for wind speeds up to 119 mph. The tests in the tunnel were run for periods of up to 4 hours, with no degradation in rigidity of the mount (see Reference 23). The method has proven to give excellent results in the flight test program. The tape used is 3M number 4265 neoprene foam, the properties of which are included in the table on Figure 3.21.

It was anticipated that the mounting locations for the control position transducers would result in non-linear calibration curves. However, the calibration curves appeared to have a linear character

ORIGINAL PAGE IS
OF POOR QUALITY



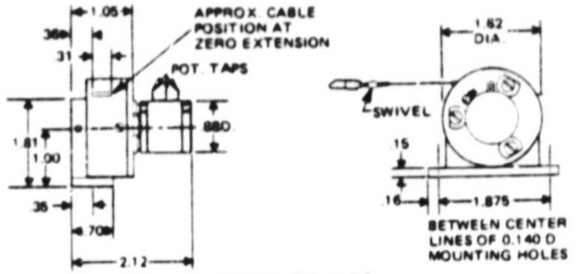
Weight	2.0 lb. (max)
Outline dimensions	3.75 x 3.75 x 2.13 in
Power input	15 w.(max)(31 vdc)
Input voltage limits	28 ±3vdc
Full-scale output	±5 vdc
Output impedance	5000 ohms (max)
Output load resistance	500K ohms (nominal)
Ripple	25 mv.peak-peak(max)
Zero rate setting	±1/2 % FS
Input range (roll/pitch/yaw)	50°/sec
Maximum input rate	600°/sec
Output voltage (at overrange limits)	±7 vdc
Output stability (input voltage variations)	1/2 % FS.
Repeatability	1 % FS.
Threshold	0.01 °/sec
Resolution	0.01 °/sec
Hysterisis	0.1 °/sec
Operating temperature	0 - 160 °F
Temperature sensitivity	Zero output 1% FS/100°F Scale Factor 3% FS/100°F

Warm up time	10 min
Motor acceleration time	30 sec (max)
Gimble deflection angle	±2° typical
Acceleration sensitivity	
Linear	0.05 °/sec/g
Angular	0.08 °/sec/rad/sec^2
Linearity	1/2 % FS, 0-1/2 scale 2 % FS, 1/2 - FS
Service life	100 hr(typical 14000 hr)
Insulation resistance	10 megohms (min),50 vdc
Damping ratio	0.5 to 0.9
Natural frequency	35 Hz (min)
Environments	
Shock	250g peak sawtooth, 5 msec
Vibration	0.1 g /Hz, 20-2000 Hz
Storage temperature	-65 - 200 °F
Radio interference	MIL-I-8161D

Figure 3.18 Northrop 3 - axis rate sensor

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

TECHNICAL INFORMATION



SERIES 160

MODEL NO.	DASH NO.	RANGE 0 TO (INCHES)	RESOLUTION INCHES
160 –	161	2	0.0033

Potentiometer life, 1-turn units, 1,000,000 cycles
3-turn units, 600,000 cycles
or 900 hours at rated power

Cable static tension at zero extension 10-16-oz

Op. temp., -85°F to +255°F

Resistance, 1000Ω - Other resistances available on special order are 5, 10, 20, 50, 100, 200, 500, 2K, 5K, 10K, & 20K. 45K (3 turn only), 50K and 100K (1 turn only)

Standard pots are used unless otherwise specified. Specials are available on special order only.

	Standard	Special
Resistance	±3%	±1%
Linearity (3 turn)	±0.25%	±0.20%
Linearity (1 turn)	±0.5%	±0.35%

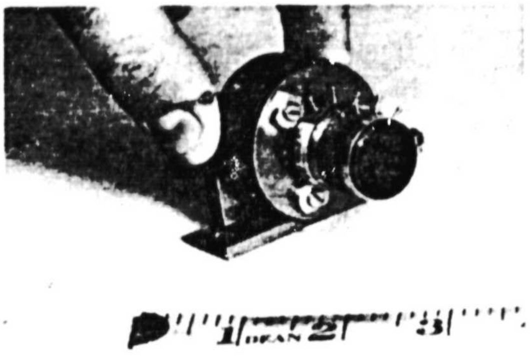
Max. current at 155°F (ambient) is 31.6 milliamps

Max. voltage across coil is 31.6 volts

Power rating, 1.0 watts at 155°F derated to 0.0 watts at 255°F

Insulation resistance, 1000 megohms min. at 500 VDC

Dielectric strength, 1000 volts RMS min. at 60 CPS



SAC Linear Displacement Transducers (LDT) consist of an extension cable, spirally wound on a spring-loaded rewind drum, which is coupled to a precision, wire-wound, rotary potentiometer. The cable end is attached to the object whose movements are to be monitored. As the cable is extended or retracted, the cable drum rotates the potentiometer wiper, varying the voltage at the wiper tap (No. 2) of the potentiometer. The voltage may be measured to reflect the position, direction, or rate of motion of the object attached to the cable.

Figure 3.19 Space Age Controls linear displacement transducer

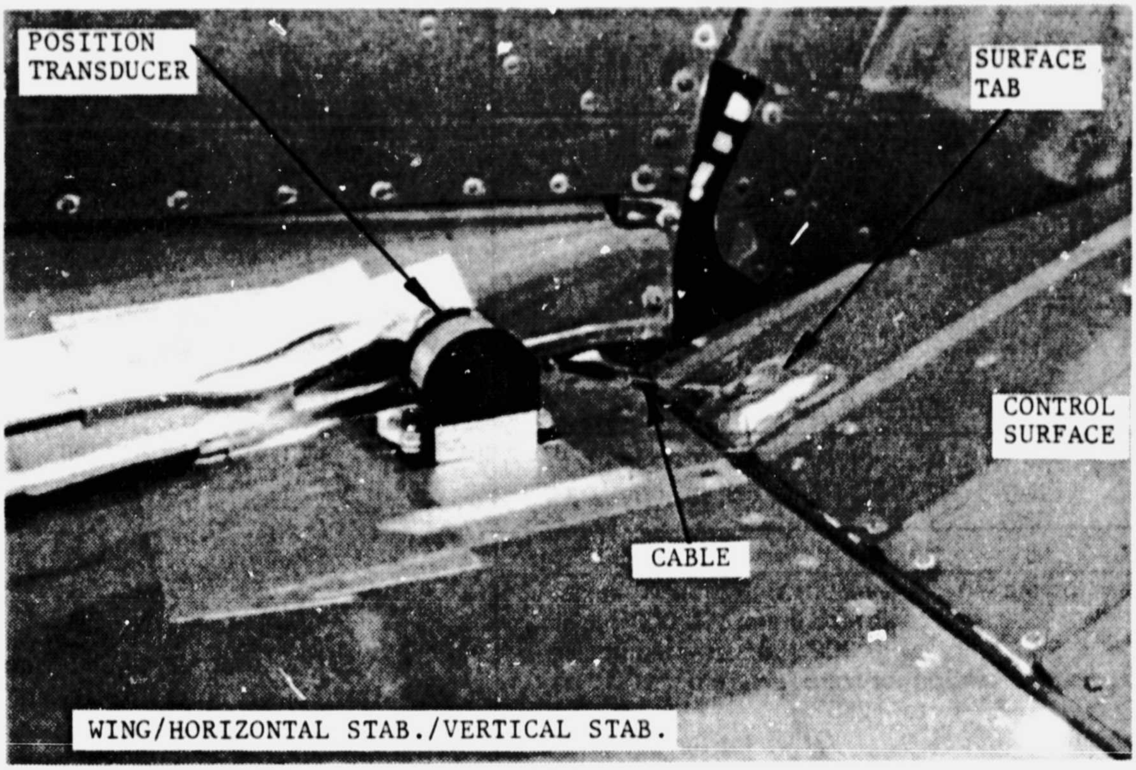


Figure 3.20(a) Control position transducer mounting detail

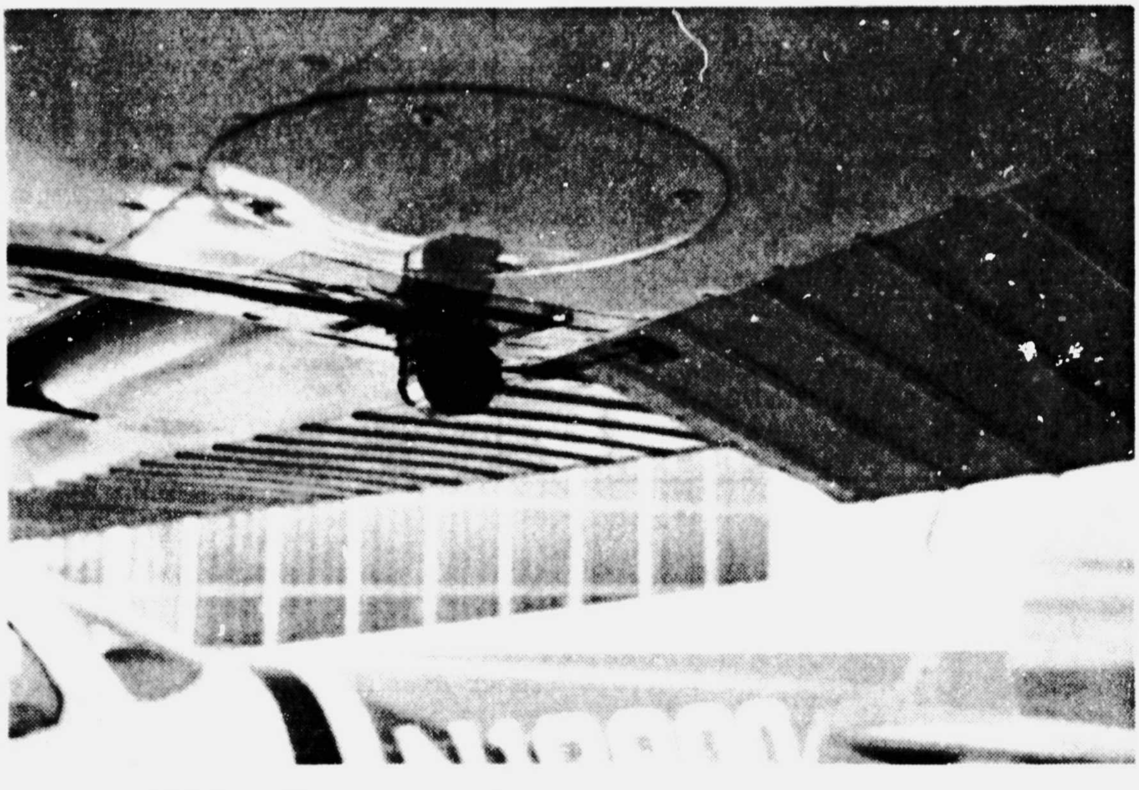


Figure 3.20(b) Aileron control position transducer

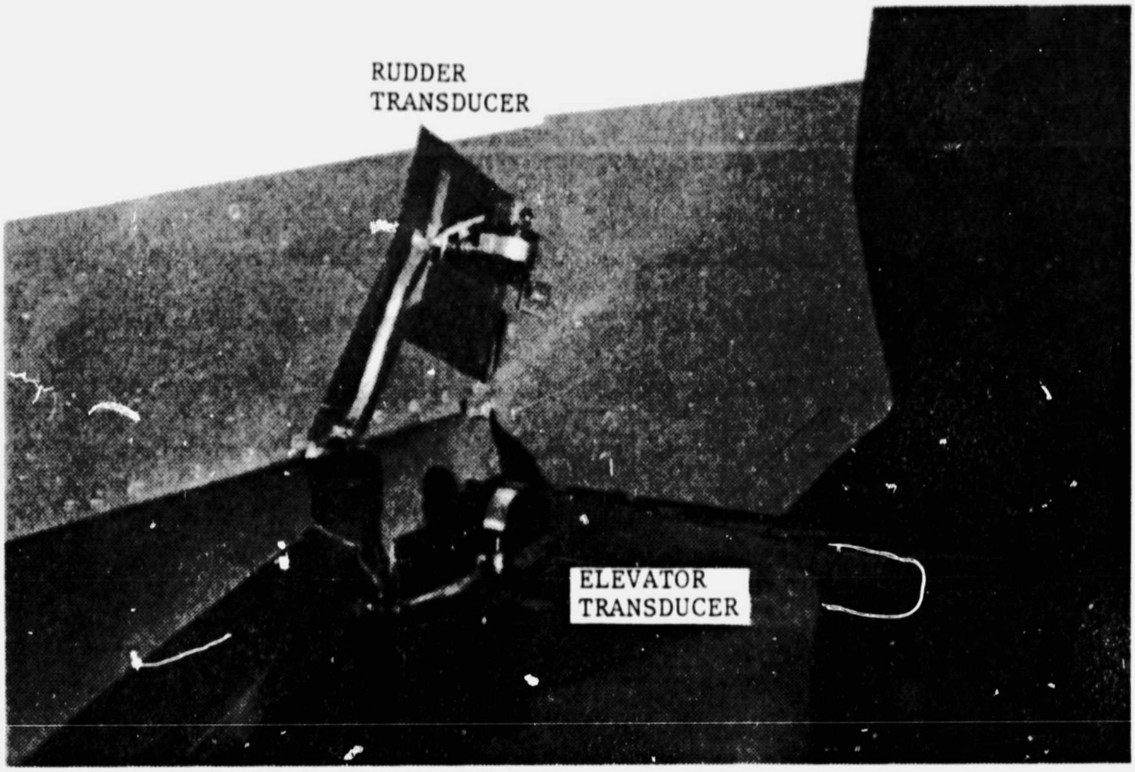
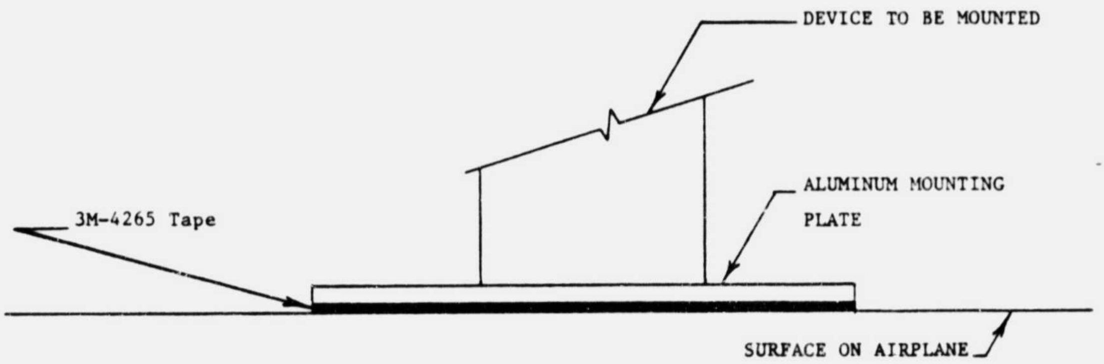


Figure 3.20(c) Rudder and elevator control position transducers



NOTE:-

- -lightly sand surface of airplane
- -clean with isopropyl alcohol
- -surface must be room temperature during attatchment
- -fair with duct tape

3M #4265 -DOUBLE COATED NEOPRENE FOAM TAPE

Adhesive	A-20 Firm Acrylic
Thickness	3/64 in.
Tensile	60 psi
Static Shear	66 psi
Temp max.	225 °F
Temp min	-20 °F

Figure 3.21 Mounting technique for external devices

(linear regression correlation coefficient of between 0.9976 and 0.9998) for the mounting locations used.

3.2.6 Static and Dynamic Pressure Transducer

A B&D Instruments Company 2504 series transducer (see Figure 3.22) was used for the static and dynamic pressure measurement. This device includes its own signal conditioning and converts the pressures to electrical signals utilizing semiconductor pressure transducers. Semiconductor transducers are largely affected by the ambient temperature; the B&D unit allows for this by heating the case and maintaining a constant temperature.

The pitot tube was designed and constructed according to Reference 24. (See Figure 3.23.) The pitot tube is attached to the underside of the wing (see Figures 3.24) using the foam tape method shown in Figure 3.21. The pitot tube allows a high angularity of the flow and still provides true readings. The distance from the wing is such that the tube is out of the boundary layer and thus provides a true total pressure reading as long as the pitot tube axis is close to the direction of airflow (±15°). The tube is mounted along the wing, halfway between the propeller arc and the wing tip (see Figure 3.25). This location minimizes flow effects due to the propeller slip stream and the wing tip vortices.

For the accurate measurement of static pressure, a trailing static cone is recommended (see Reference 25). Initial flights showed difficulty in deployment of the static cone after takeoff.

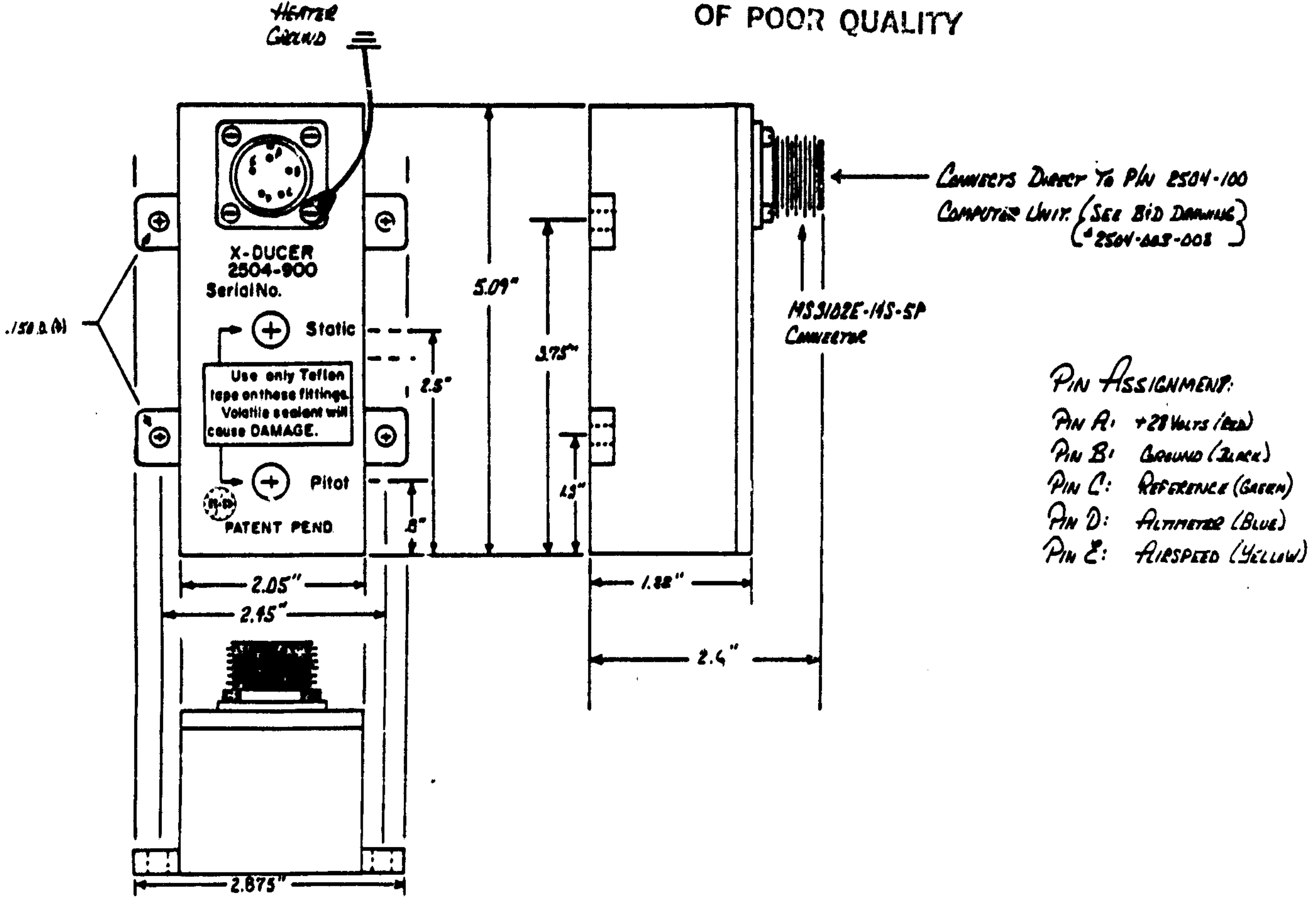


Figure 3.22 B&D Instruments 2054 pressure transducer

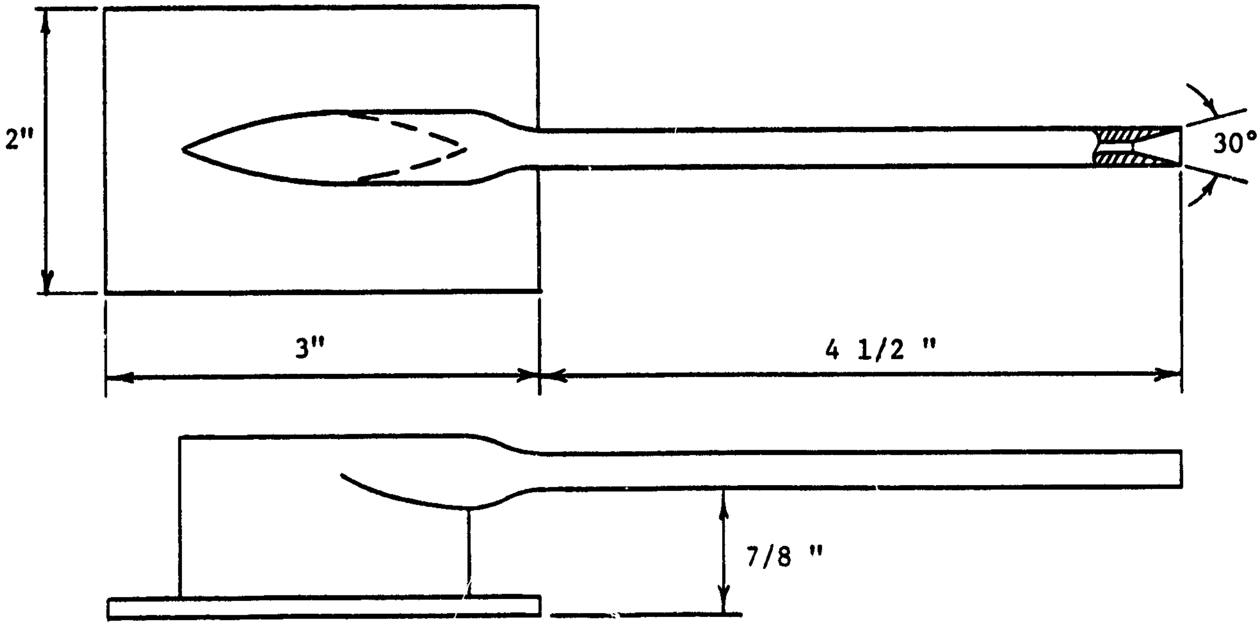


Figure 3.23 Pitot tube

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

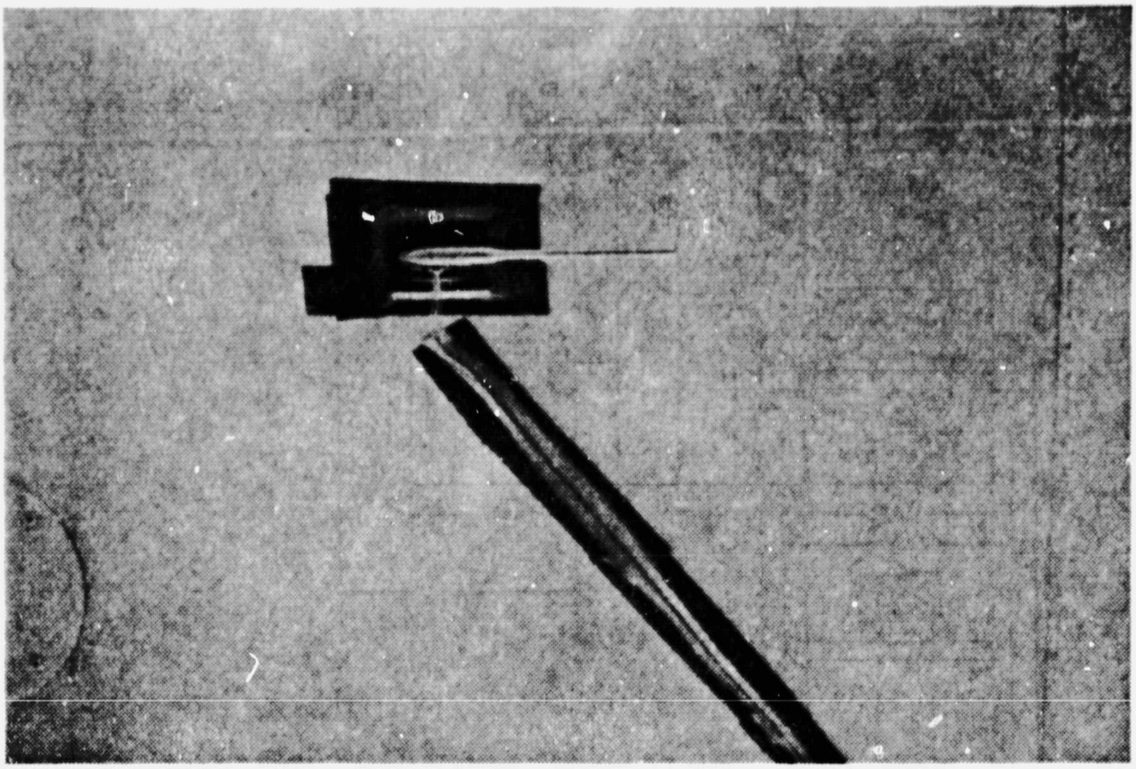


Figure 3.24(a) Pitot tube mounted on airplane

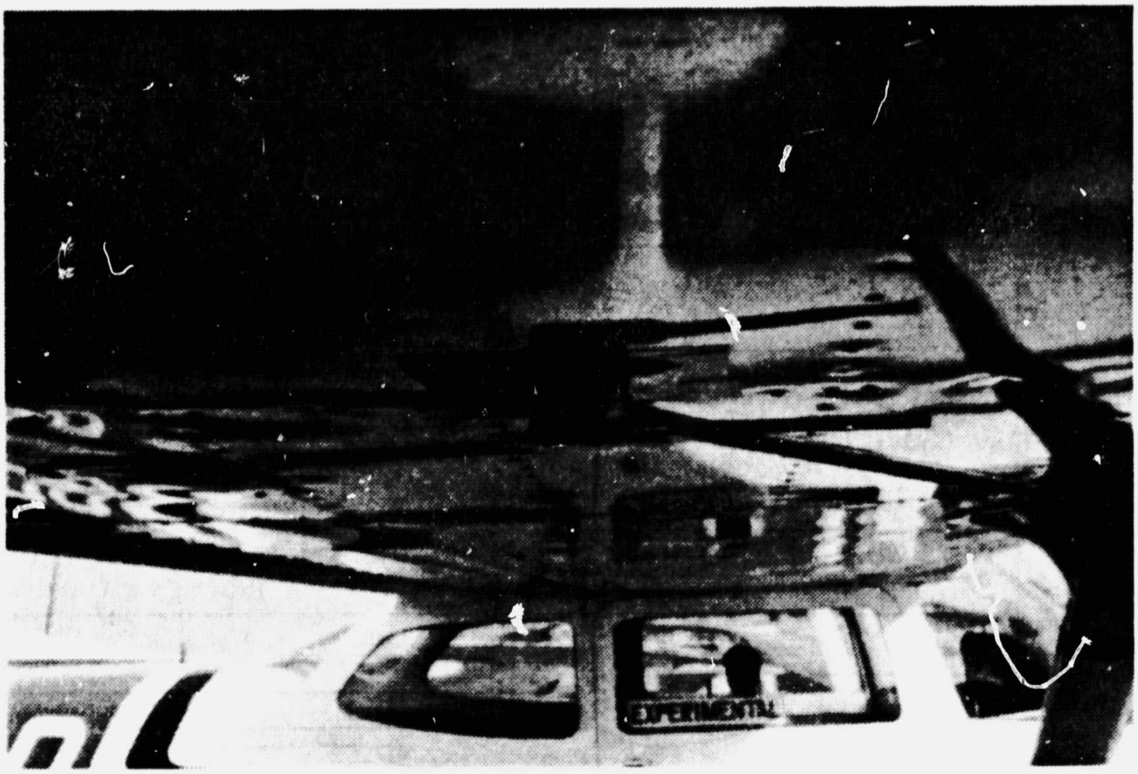


Figure 3.24(b) Pitot tube mounted on airplane

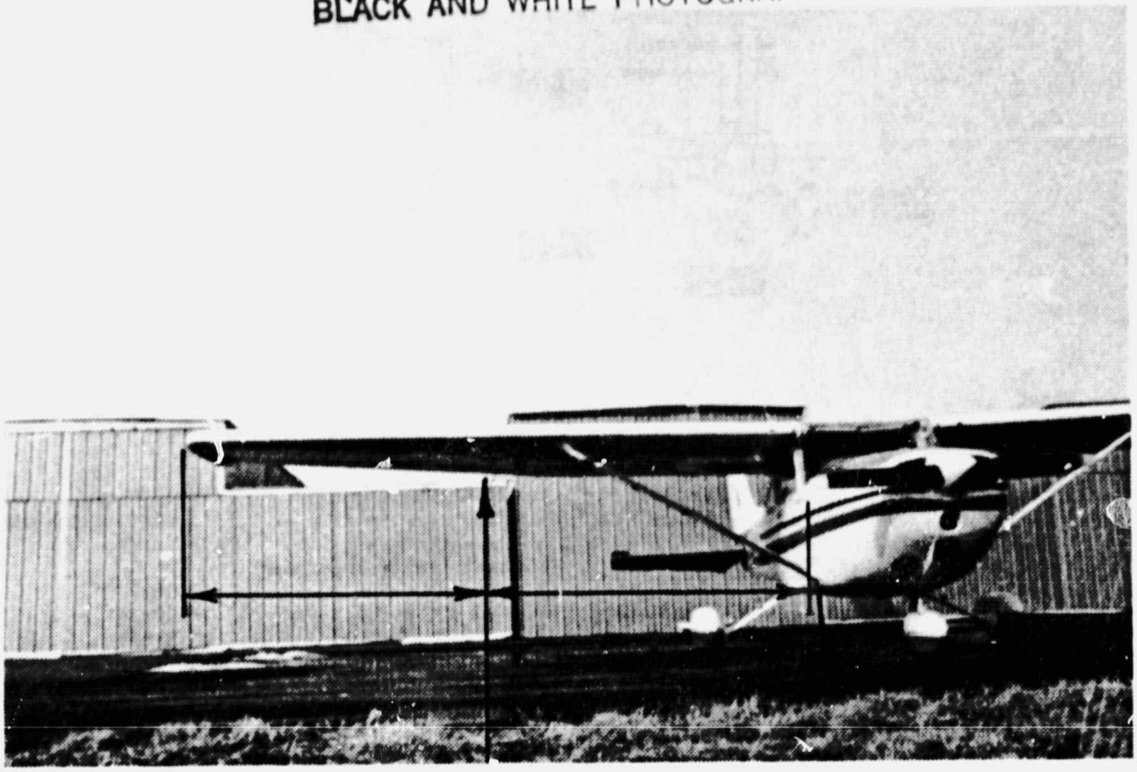
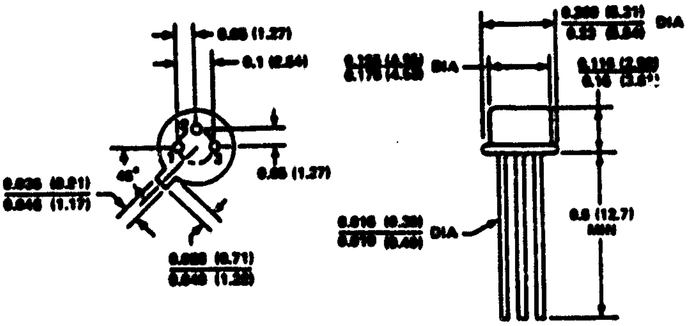


Figure 3.25 Pitot tube mounting location

The cone has not been used and is not essential if only stability analysis is performed. The airplane static system is sufficient for stability analysis; however, a more accurate method would be required for any performance testing.

3.2.7 Temperature Transducer

An Analog Devices Company Semiconductor temperature transducer was used for measurement of air temperature. Specifications are shown in Figure 3.26. The transducer is mounted in a probe, as shown in Figure 3.27. The temperature probe is mounted the same way as the pitot tube, using the double-sided tape method. The temperature probe is shown mounted on the airplane in Figures 3.28. The location of the probe is identical to that of the pitot tube, but on the opposite wing.



Model	AD590M
Absolute Maximum Ratings	
Forward voltage	+44 v
Reverse voltage	-20 v
Breakdown voltage(to case)	±200 v
Rated temperature range	-55°C to +150°C
Storage temperature	-65°C to +155°C
Lead temperature(soldering)	+300°C
PowerSupply	
Operating voltage range	+4v to +30v
Output	
Nominal current (+25°C)	298.2 μA
Nominal temp. coefficient	1μA/°C
Calibration error (+25°C)	±0.5°C max

Absolute error (rated range)	
No external adjustment	±1.7°C max
+25°C calib error = 0	±1.0°C max
Nonlinearity	±3.0°C max
Repeatability	±0.1°C max
Long term drift	±0.1°C/month max
Currant noise	40 pA$\sqrt{Hz}$
Power supply rejection	
+4v <Vs <+5v	0.5μA/v
+5v <Vs <+15v	0.2μA/v
+15v <Vs <+30v	0.1μA/v
Case isolation	10^{10} ohms
Effective shunt capacitance	100pF
Turn on time	20 μs
Reverse bias leakage (reverse voltage =10 v)	10 pA

Figure 3.26 Temperature transducer specifications

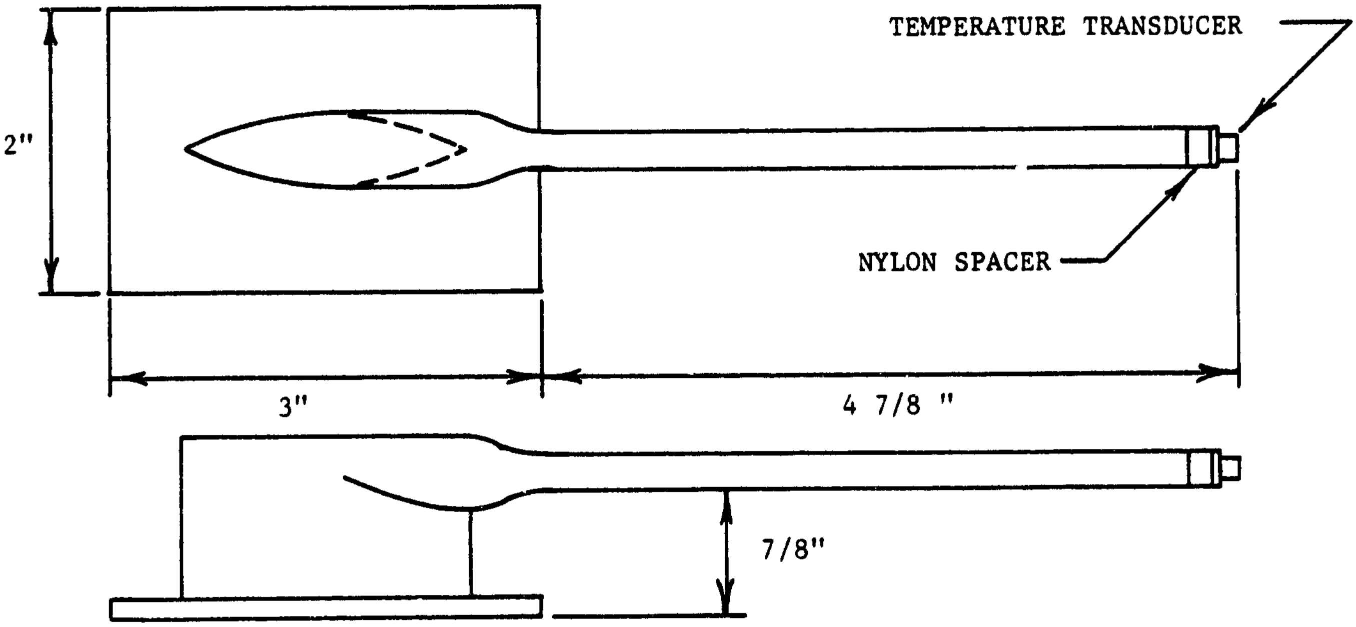


Figure 3.27 Temperature probe

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

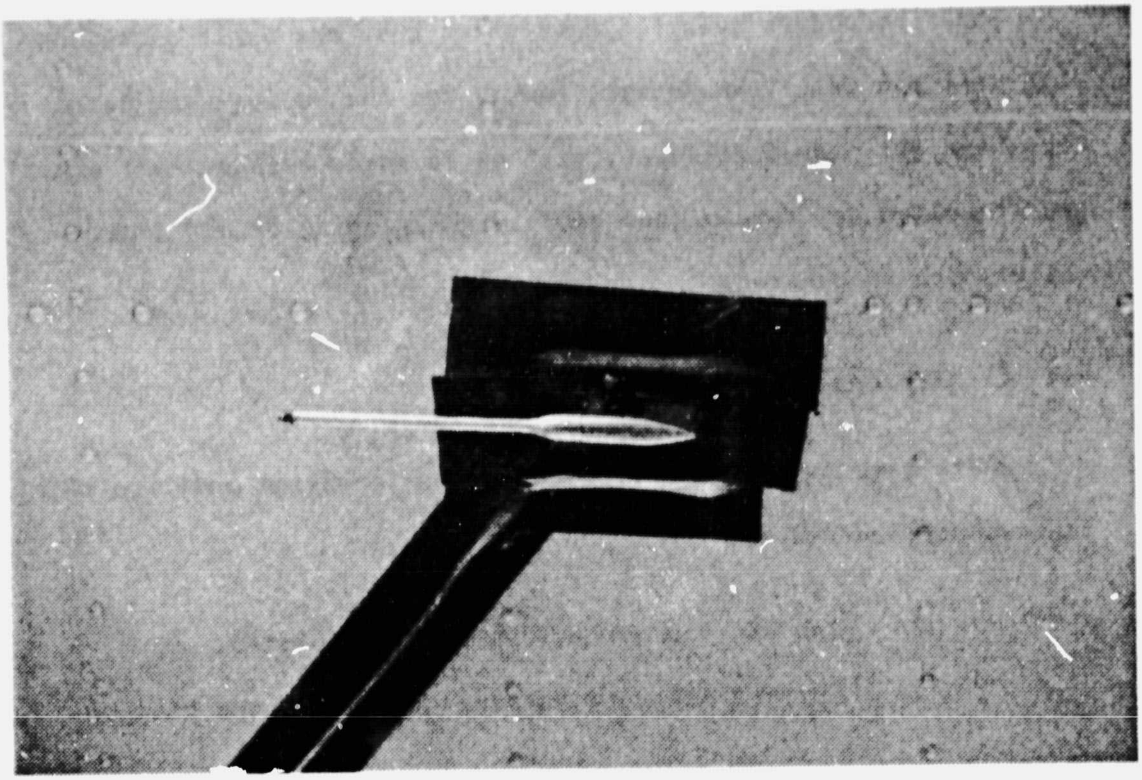


Figure 3.28(a) Temperature probe mounted on airplane

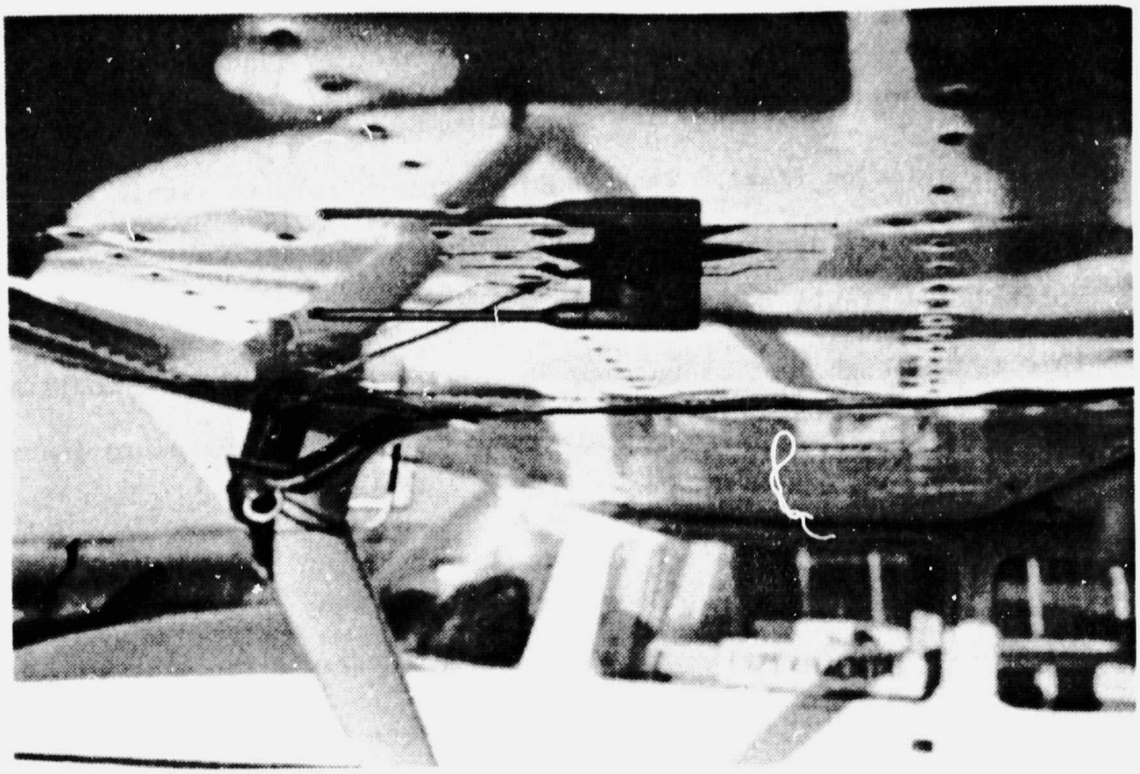


Figure 3.28(b) Temperature probe mounted on airplane

The trandsucers selected have shown that the basic decisions regarding specific transducers, ranges and accuracies were correct. They have all proved reliable, with no failures encountered; and none required any specialized signal conditioning or difficult calibration procedures.

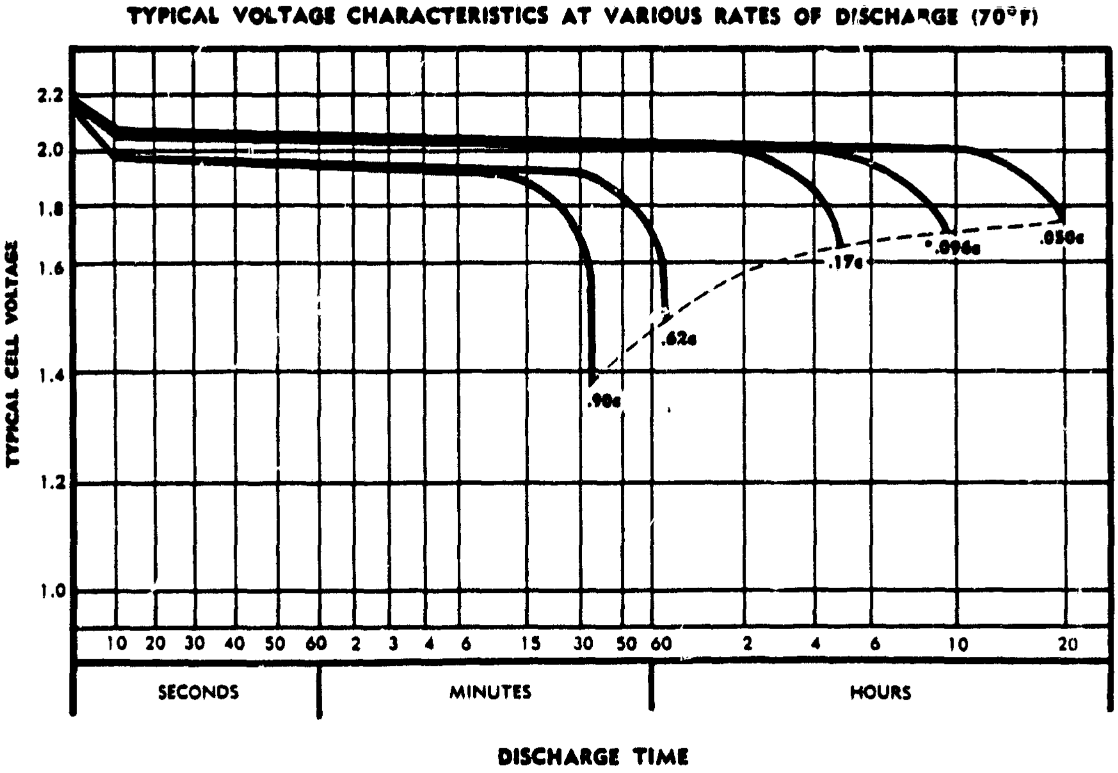
3.3 Power Supply

There were two options considered for supplying power to this instrumentation system:

1) Tap off the aircraft electrical system, or
2) Carry a separate battery package on the flight.

Considering option one, using the aircraft power system, offered several advantages. These were reduction in size of the instrumentation system, and no limited usage time due to battery rundown. It was realized, however, that there are several voltage standards on the current general aviation fleet. This would therefore require either a complex voltage control system or several systems to account for the various voltages available in the airplanes to be considered. Coupled with this is the high cost of voltage conversion systems. Also, modification would then be required to the airplane's electrical system to install the instrumentation package.

It was decided to explore the second option. A suitable rechargeable battery was found, manufactured by Eagle-Picher. These lead acid batteries are sealed, rechargeable, and maintenance free. A typical discharge curve is shown in Figure 3.29. These batteries when used in a deep cyclic regime (i.e., removing 50-100% of the battery rated capacity prior to recharge) have a recharge time of 12 to 20 hours. They have an expected lifetime of 100 to 150 complete charge/discharge cycles, with longer life expectancy when less than 100% depth of discharge is used. These batteries can also be used in any position. The cost of these batteries is such that several battery packs could be purchased for less than the price of one regulated voltage divider required if the airplane electrical system were used.



*To Determine Discharge Rate of Various Batteries Multiply Rated Capacity (C) by factor shown: for example — The rate at which an eight ampere hour battery must be discharged to yield a useful ten hours equals .096C or .096 x 8 A.H. = .77 amperes.

Figure 3.29 Battery discharge curve

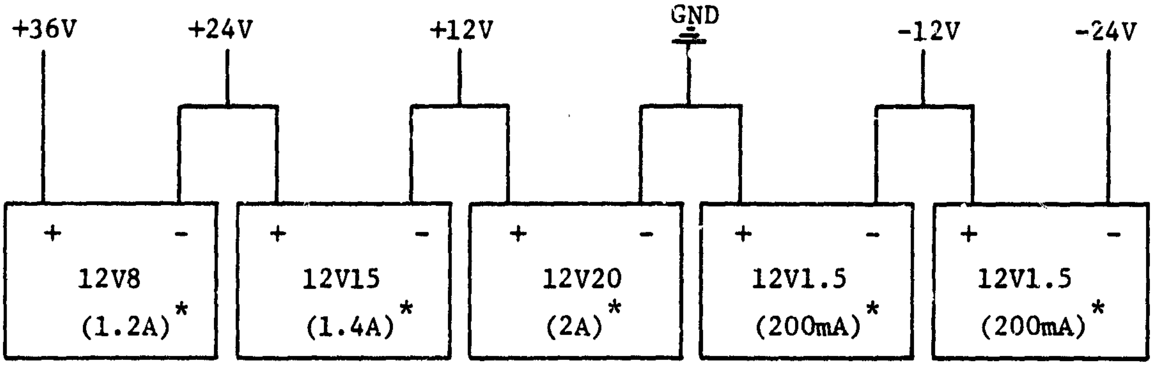
Another advantage of a battery system is stability of the voltage supplied to the system. This advantage stems from two conditions. One is the fact that no external loads are on the power supply; thus, the power being used is steady and unchanging. Second is the fact that no ripple or noise will be in the power supplied. With a ship-supplied system, ripple will be present in the voltage system due to the means of supplying power (from the generator or alternator systems). This steady voltage supply, and the lack of ripple when the batteries are used, results in transducers being able to normally exceed their advertised specifications.

The voltages required for the complete on-board data acquisition system are shown in Table 3.4. Batteries were selected to match the power requirements at the various voltages. The wiring schematic, as well as the specifications of the batteries selected, are shown in Figure 3.30. The batteries allow a minimum of 3 hours running time between recharge. (The 12-volt battery supplying the TEAC tape drive is discharged first.)

The biggest disadvantage when batteries are used is that of weight. The battery module, complete, weighs 60.5 lbs. This is the heaviest component in the entire system (see the table of Figure 3.2). Total weight of the entire instrumentation system including all cables is 132.4 lbs. This system weight is not a problem for the majority of general aviation airplanes.

Table 3.4 Power Requirements fo Data Acquisition Package

BATTERY VOLTAGE	REGULATED VOLTAGE	REQUIREMENT
+36	+28	Heater(P_S,P_D) Gyro motors(θ,ϕ,p,q,r)
+24	+15	Accelerometers(A_X,A_Y,A_N) Filters, MDAS-16
	+12	TEAC tape drive
+12	+5.5	P_S and P_D reference voltage
	+5	Potentiometers (θ,ϕ,δ_E,δ_A,δ_R) AIM 65 computer Temperature transducer
-12	-5	Potentiometers (θ,ϕ,δ_E,δ_A,δ_R)
-24	-15	Accelerometers(A_X,A_Y,A_N) Filters, MDAS-16



* maximum current requirement

		NOMINAL CAPACITY				DIMENSIONS (INCHES)				
BATTERY NUMBER	NOMINAL VOLTAGE	20 HR	10 HR	5 HR	1 HR	LENGTH	WIDTH	HEIGHT	TO TERMINAL	WEIGHT (LB)
CF12V20	12	20.0	19.0	17.5	12.5	6.51	4.91	6.53	6.75	16.2
CF12V15	12	15.0	14.5	13.0	9.0	7.22	3.34	6.50	6.75	12.8
CF12V8	12	8.0	7.7	7.0	5.0	6.00	4.00	3.75	3.97	7.[illegible]
CF12V1.5	12	1.5	1.4	1.3	0.9	7.02	1.33	2.40	2.69	1.9

Figure 3.30 Battery module schematic and specifications

3.4 Pilot Control

The pilot controls the instrumentation system using a box which can be placed on the seat beside him (see Figure 3.31). The control box performs essentially the same function as the ground keyboard, the switches on the box replacing the keys (which are really just momentary contact switches). The controls are described below.

3.4.1 System Control

This consists of three switches.

First is the "INITIALIZE" tape switch. This is a momentary contact switch which is used only after insertion of a fresh data tape. This function prepares the data cassette to accept data.

Second, the "RUN/STBY" toggle switch is used to control when data is being recorded. In the STBY position the system is non-active. In the RUN position, data is recorded. There are two of these switches, one of which is located on the pilot control wheel and the other, on the pilot control box.

Third is the "REWIND" switch. This is used at the end of a cassette or flight. Activation of this switch places an "end" mark on the data tape and rewinds the tape.

3.4.2 Transducer Readout Control

A high-impedance analog voltmeter is provided to the pilot so that he can observe a particular transducer as he requires. The meter's installation is shown in Figure 3.32. A rotary switch

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

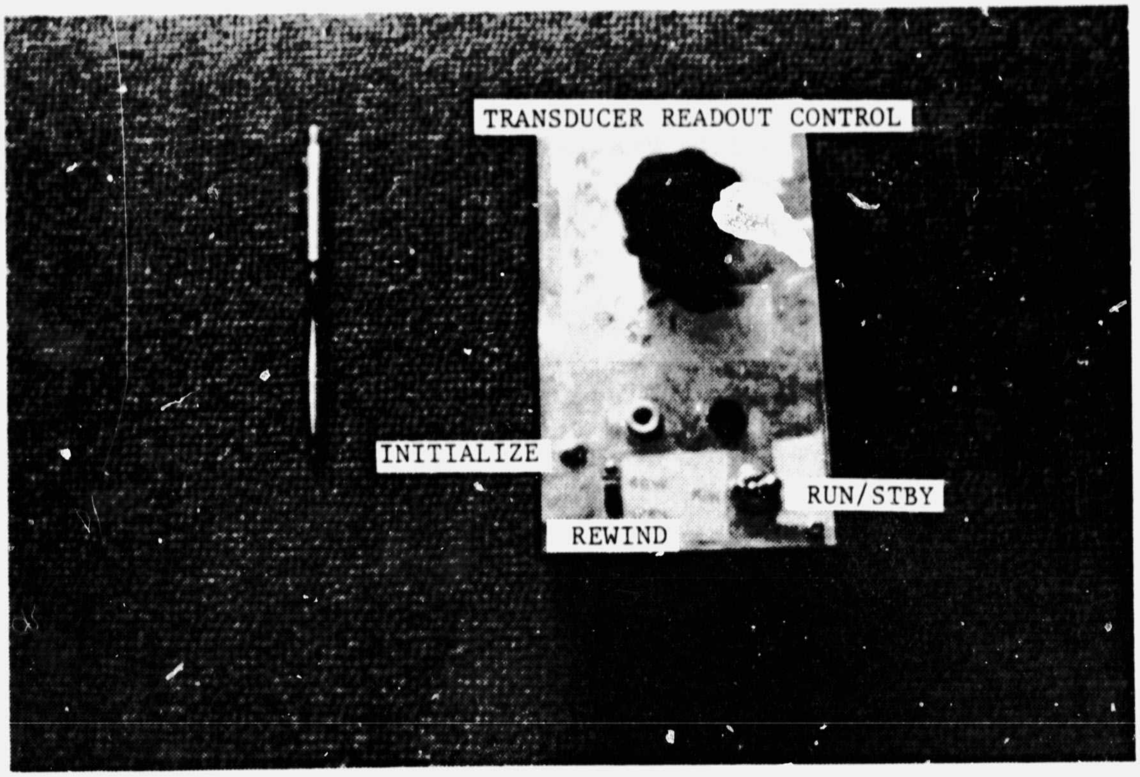


Figure 3.31 Pilot control console

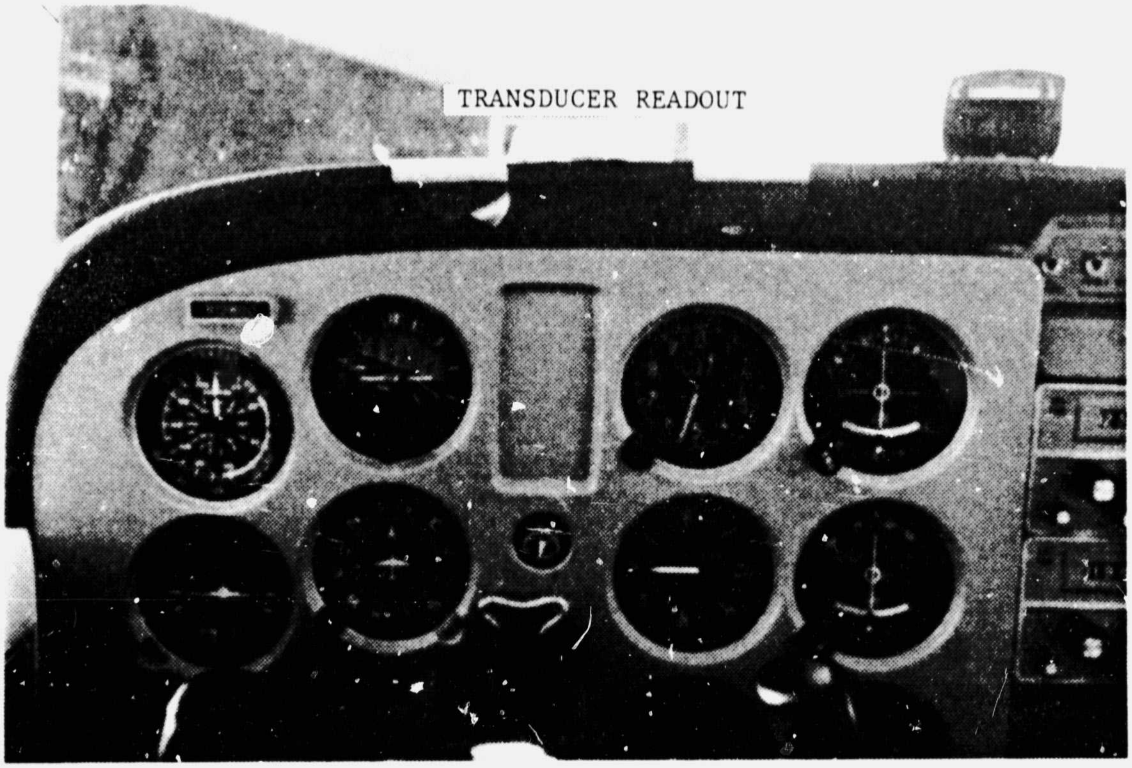


Figure 3.32 Transducer monitor

(on the pilot control console) controls the signal which is observed.

This feature is also used to verify that all transducers are operating correctly prior to a test flight.

4. GROUND COMPUTER SYSTEM

The MMLE* data reduction process described in this report requires a powerful computer capable of being programmed in a high level language. Phase I (Reference 2) pointed out the requirement for a computer system operating under a compiled language. This requirement is due to the lengthy execution times associated with interpretive languages. (A Hewlett Packard 9825 was used in Phase I, programmed in interpretive Basic.) This chapter presents a benchmark process which has been used to evaluate the capability of the computer systems to perform the data reduction tasks. Also, a description of the selected computer is presented.

A two-step evaluation process was used. The first program in this process, the INTEGER SPEED ROUTINE, is short and easy to implement and gives a ball-park speed estimate. Secondly, the FLOATING POINT SPEED ROUTINE is a lengthier program, more closely resembling the operations performed in the MMLE process. These programs are described below.

4.1 Integer Speed Routine

This is a short, easy-to-implement program giving a rough benchmark of the operating speed of computers. The idea for this routine was originally conceived in Reference 26. A listing is presented in Table 4.1. The program does not realistically reflect the MMLE data reduction process, but it can be easily implemented in virtually

---

*MMLE = Modified Maximum Likelihood Estimation (see Chapter 5).

Table 4.1 Integer Speed Routine
(Fortran Listing)

```
        DO 100 M=5,10000,2
        I=M/2
        DO 200 K=3,I,2
        J=(M/K)*K
        IF(J.EQ.M) GO TO 100
200     CONTINUE
        PRINT,M
100     CONTINUE
        STOP
        END
```

any language on most computers in little time. This increases the ease with which a benchmark can be run and gives a ball park estimate of a computer's speed performance. The results of this speed comparison are presented in Table 4.2.

For evaluation of this data, it was assumed that once through this program was equal to two iterations of the MMLE routine. Therefore, to obtain the desired data reduction time through MMLE of 5-20 minutes, the Integer Speed Routine needs to run at 2-8 minutes on an acceptable computer. From Table 4.2 it is seen that all acceptable computers had both a compiler, which compiled down to machine code, and a floating point hardware package. Also, all acceptable computers were either using 16-bit microprocessors or could be considered main frame machines. It was obvious that current 8-bit microcomputers would not be capable of performing the data reduction task in any reasonable time frame. This is evident by the fact that the AIM-65 (using a 6502, 8-bit microprocessor) could not meet the speed requirements even in assembly language.

This study narrowed the number of acceptable machines considerably.

Table 4.2 Integer Speed Comparison

PROCESSOR	MACHINE	LANGUAGE	INTERPRETER	COMPILER	FLOATING POINT HARDWARE	HRS:MIN:SEC	ACCEPTABLE
8-BIT MICRO	AIM 65	BASIC (PRINTER OUTPUT)	*			4: 14: 44	
		ASSEMBLY (LED OUTPUT)				0: 23: 36	
		ASSEMBLY (PRINTER OUTPUT)				0: 33: 40	
	TRS 80	LEVEL I BASIC	*			7: 12: 27	
		LEVEL II BASIC	*			6: 31: 10	
		ASSEMBLY				0: 21: 55	
		FORTRAN		*		0: 54: 18	
		MODEL II BASIC	*			3: 15: 00	
	APPLE II	INTEGER BASIC	*			2: 24: 31	
		FLOATING POINT BASIC	*			3: 56: 23	
16-BIT MICRO	TERAC 8510	PASCAL (COMPILE TO P CODE)	*	*		0: 30: 35	
	TEKTRONIX (4052)	BASIC	*			1: 23: 00	
	HP 9825	BASIC	*			1: 41: 17	
	HP 1000	FORTRAN RTE IV B (CRT OUTPUT)		*	*	0: 01: 23	✓
		" (NO OUTPUT)		*	*	0: 00: 48	✓
		FORTRAN RTEM (CRT OUTPUT)		*	*	0: 00: 57	✓
		" (NO OUTPUT)		*	*	0: 00: 44	✓
	IBM SERIES I	FORTRAN (NO OUTPUT)		*	*	0: 01: 30	✓
		" (PRINTER OUTPUT)		*	*	0: 04: 30	✓
	PDP 11/34*	FORTRAN (RSX 11 M) (CRT OUTPUT)		*	*	0: 07: 10	✓
		" (PRINTER OUTPUT)		*	*	0: 11: 20	
	MINC 11/23	FORTRAN RT11-IV (CRT OUTPUT)		*	*	0: 03: 36	✓
		" (DISC OUTPUT)		*	*	0: 03: 29	✓
		FORTRAN RT11-IV PLUS (CRT OUTPUT)		*	*	0: 03: 10	✓
		" (DISC OUTPUT)		*	*	0: 03: 00	✓
MAIN FRAME	HONEYWELL 60/66	FORTRAN		*	*	0: 00: 44	✓
		PL/1		*	*	0: 02: 13	✓
	CDC CYBER 70	FORTRAN (NON OPTIMIZED)		*	*	0: 00: 39	✓
		FORTRAN (OPTIMIZED)		*	*	0: 00: 37	✓
	IBM 370-148	PL/1 (OPTIMIZED)		*	*	0: 01: 19	✓

*The PDP 11/34 was operating in a multi-user mode. Its performance is estimated to be approximately 2-3 times faster than the 11/23 series computer in single-user mode.

4.2 Floating Point Speed Routine

To more closely resemble the MMLE data reduction process, yet still use a simple-to-implement program, the routine shown in Table 4.3 was developed. The program is made up of floating point matrix mathematics, which is what MMLE primarily contains.

Table 4.3 Floating Point Speed Routine (Fortran Listing)

```
10         REAL A(20,20),B(20,20),C(20,20),E(20,20),T(20,20),D(20,20),F
20         INTEGER I,J,K,M
30         PRINT,"START"
40         F=.098625
50         DO 400 M=1,40
60                 DO 200 I=1,20
70                 DO 200 J=1,20
80                         E(I,J)=0
90                         A(I,J)=F*I*J
100                        B(I,J)=F*I
110                        C(I,J)=F*J
120                        D(I,J)=F
130                        T(I,J)=0
140  200           CONTINUE
150                DO 300 I=1,20
160                DO 300 J=1,20
170                DO 300 K=1,20
180                        T(I,J)=T(I,J)+(A(I,K)*B(K,J))
190                        E(I,J)=E(I,J)+(E(I,K)*D(K,J))
200  300           CONTINUE
210                DO 400 I=1,20
220                DO 400 J=1,20
230  400                   E(I,J)=E(I,J)+T(I,J)
240                PRINT,"E="
250                DO 100 I=1,20
260                DO 100 J=1,20
270                        PRINT,E(I,J)
280  100   CONTINUE
290        PRINT,M
300        STOP
310        END.
```

The program approximates one iteration of the MMLE method. This is indicated by the 48 minute run time on the Hewlett Packard 9825, which requires approximately 50 minutes to perform one iteration of the MMLE program. In order to deem a computer acceptable for the MMLE process, it must be able to complete the floating point speed routine in the order of 1-5 minutes. It was decided that an MMLE execution time of 5-20 minutes would be acceptable (assuming 5 iterations).

The results of this test are presented in Table 4.4. It is seen that the 16-bit machines tested, operating in compiled Fortran, meet the speed requirement.

Table 4.4 Floating Point Speed Comparison

MACHINE		MIN:SECS
HP9825	(BASIC)	48:15
HONEYWELL 60/66		0:20.6
HP1000	(NO OUTPUT)	1:08.7
	(DISC OUTPUT)	2:07
IBM SERIES 1	(DISC OUTPUT)	0:58
MINC 11/03	(DISC OUTPUT)	5:35
MINC 11/23	(DISC OUTPUT)	4:00

4.3 Description of System

The results of the benchmark evaluation left several computers that were deemed acceptable. To select the best machine for the KU-FRL requirements, the following factors were also considered:

- Memory expansion capability
- Floating Point Hardware available
- RS232 ports/IEEE 488 ports* installed
- CRT Graphics capability
- Hard/Flexible disc storage
- Programming languages available
- Users group existing
- Delivery
- Cost

---

*Industry interfacing standards

Evaluating the acceptable computers, the DEC* MINC 11/03 computer was selected as best meeting the requirements. A description follows.

The MINC 11/03 is shown in Figure 4.1. The block diagram of Figure 4.2 shows the basic features and some of the options available.

The computer uses a 16-bit DEC LSI 11/03 processor, capable of addressing 64K bytes of memory, and contains a floating point hardware package, 4 RS232 ports, and an IEEE 488 port.

Data and program storage is handled using the dual RX02 flexible disc drives. These use 8" flexible discs, capable of holding 500 K bytes of information each.

Computer and program interaction is handled using the DEC-VT 105 graphics terminal. This permits inputting and outputting of data, as well as allowing graphical representation of the flight test results.

The RS232 ports are used for input and ouptut of the data. Four are provided. One is used for the VT 105 terminal, two are configured to allow data transfer from the Rockwell AIM-65, and one is used to control a hard copy printer.

The IEEE 488 port allows ease of interfacing to many industry standard components. Planned future use of this port is for a hard copy plotter for analysis and report quality plots of flight test data.

---

*DEC = Digital Equipment Corporation.

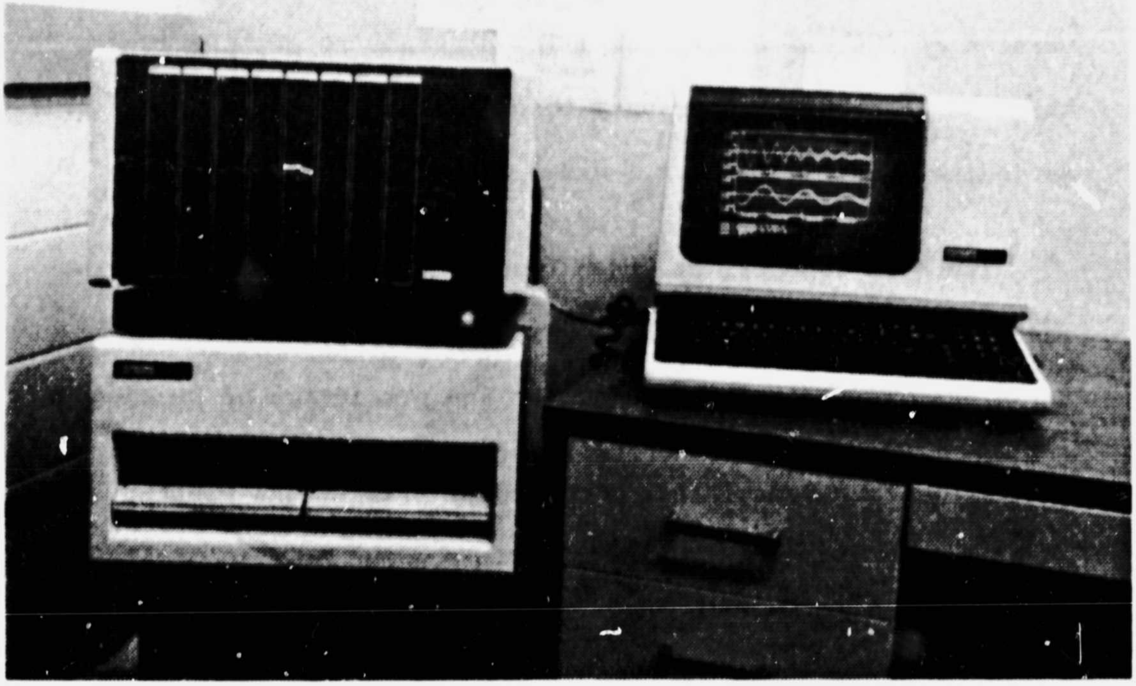
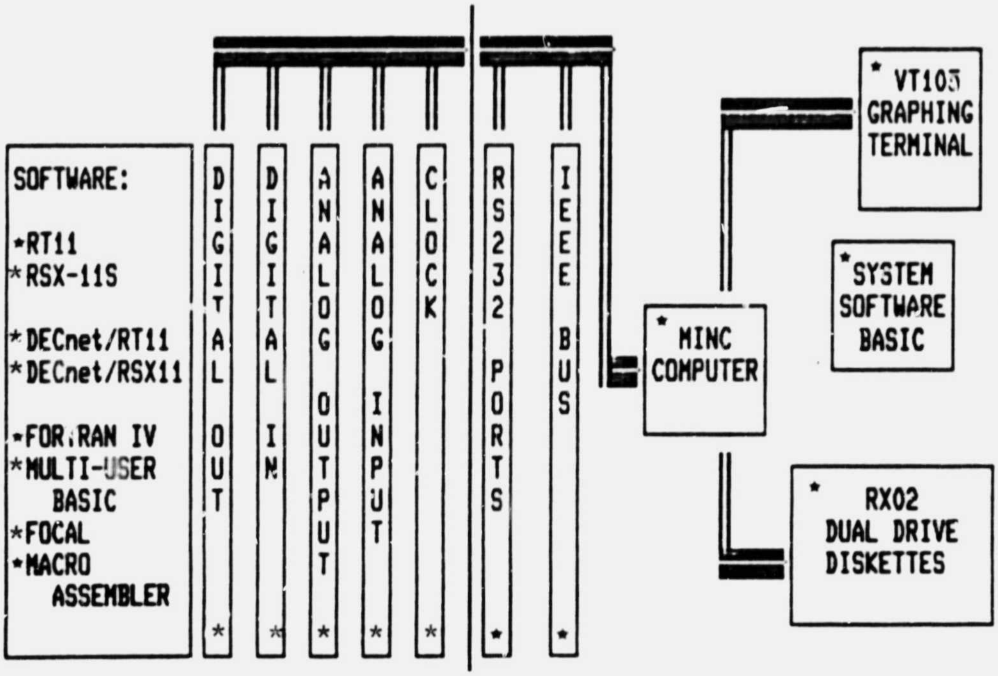


Figure 4.1 Digital Equipment Corp. MINC 11/03 computer



*Options available (not on KU-FRL system)

•Installed on KU-FRL system

Figure 4.2 Block diagram of MINC 11/03 computer

The standard MINC comes with BASIC language software. The KU-FRL package has the RT11-FORTRAN IV software option. This version of FORTRAN allows compiling programs to machine level, which was determined necessary to perform the data reduction task as indicated in Sections 4.1 and 4.2.

The MINC computer has been found capable of performing the function intended. The MMLE process takes approximately 20 minutes for 5 iterations, which is close to the prediction of Section 4.2.

It is recommended that a hard copy printer and plotter be added to the standard MINC to make it a complete data analysis system.

5. DATA REDUCTION METHOD

This chapter describes the data analysis procedures used for longitudinal and lateral stability analysis. The overall method is best depicted via the flow chart shown in Figure 5.1

For this phase the system described in Chapter 3 was used for airborne data acquisition. The KU-FRL's DEC-MINC 11/03 microcomputer was used for all further data processing. This computer is described in detail in Chapter 4. Segmenting the various data reduction programs into the blocks as shown in Figure 5.1 allowed effective data analysis.

This section describes theoretical aspects of the computer programs used. Flow charts and program listings are included as Appendix A.

5.1 Data Acquisition

This program is used as part of the airborne data recording system. It is written on the AIM-65 in machine level language to allow rapid execution. The program controls the MDAS-16 module, as well as the TEAC cassette recorder. (See Appendix A.1 for flowchart and listing.)

The program has three control inputs, which are located on the pilot control console. The first is the "INITIALIZE" tape button. This is used for getting the data cassette ready to record the signals. It is used only once per data cassette. This command

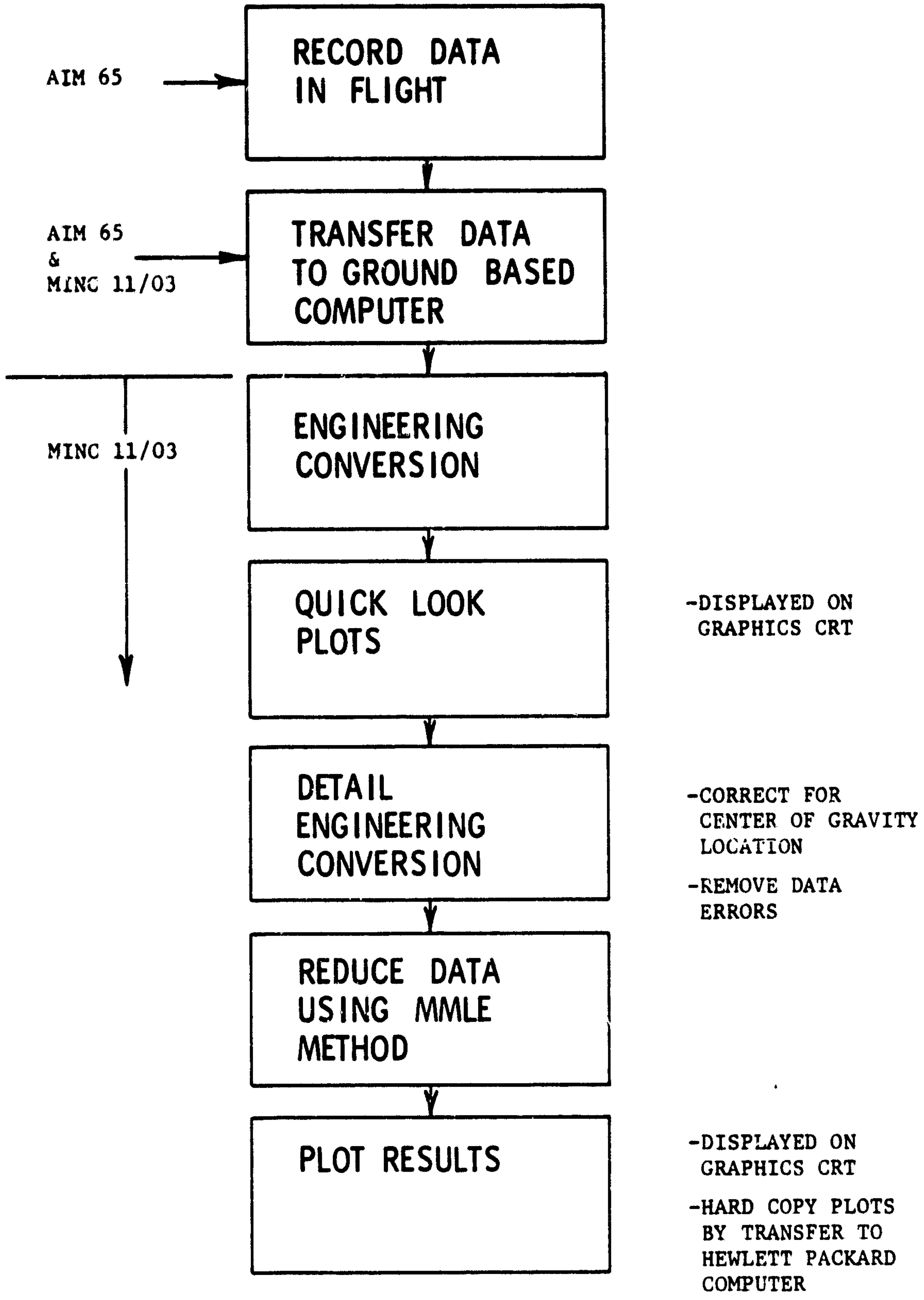


Figure 5.1 Data processing flow chart

rewinds the tape (if required), advances the tape past the beginning of the tape hole, and then writes a beginning-of-tape file mark.

Second, the "RUN/STBY" toggle switch is used to control the recording of data. Placing this switch in the RUN position begins the data recording process. The computer then sends control to the MDAS to sample the P_D, P_S and T channels. These are sampled 10 times each and then output to the TEAC cassette drive in one block. The program then runs through the other channels (A_X, A_Y, A_N, θ, ϕ, p, q, r, δ_E, δ_A, δ_R). These data are temporarily stored in computer memory. After a total of 0.1 seconds has elapsed, the computer then samples these channels again and also temporarily stores them in memory. After 10 such time points are in the computer memory, the AIM-65 outputs this to the TEAC in one block; and the process continues until the "RUN/STBY" switch is placed in STBY. Then the computer samples the P_D, P_S and T channels again and outputs these to the tape. After this the system idles, waiting for the next command.

To reduce the possibility of error, the highest order bits on the measurement channels are recorded twice. This is easily done, as the analog-to-digital conversion comes out as a 12-bit word, available as a tri-state output. The AIM-65 operates on the basis of 8-bit words; therefore, the 4 highest order bits of the data are recorded twice, resulting in two 8-bit words. These higher order bits are compared on readback as a means of error checking.

Third, the "REWIND" switch causes an end-of-tape mark to be written on the cassette and then rewinds the tape back to the start.

This program also keeps track of the run number, which is output at the beginning of each run to the cassette.

5.2 Transfer Data to Ground Based Computer

This operation requires both the AIM-65 and MINC 11/03 and a program for each to allow the two to be coupled. A standard RS232 serial interface is available on both computers. The data can be transferred across telephone lines if desired.

The data, before being transferred from the AIM-65, is checked for errors. This is done by comparing the high order bits, which have been recorded twice. A running total of any errors is kept and printed out by the AIM-65 on its display printer. Errors have not been significant in number, and therefore no correction is made. All errors to date have been caused by poor quality data cassettes. Using the qualified cassettes (see Table 5.1 and Reference 27) no data errors have been found in the flight data.

Table 5.1 Qualified Data Cassettes

Manufacturer	Type	Part No
3M	Scotch	834A/1-300
TDK	Data Cassette	HR-850 90C
MAXELL	Data Cassette	M-90
BASF	Digital Power Typing Cassette	52346

(Qualified as per Reference 27)

The AIM-65 program is shown in Appendix A.2; and the MINC 11/03 program, in Appendix A.3. These programs are used to transfer the flight data from the TEAC cassette tape to the MINC 11/03 disc. In this mode the AIM-65 keyboard is used for controlling the data transfer off of itself. The MINC 11/03 program loads the transferred data into its memory and then transfers this data to the data disc.

5.3 Engineering Conversion

The first step in the actual data analysis procedure is converting the raw data bits into their corresponding engineering units. The process involved first converts the bit pattern of each measurement to the voltage representation. (See Reference 3 for detailed explanation of this process.) Then, utilizing the particular transducer calibration curve, this voltage representation is converted to the units of the actual motion measured. Resulting from this, then, is the transducer measurement in the correct engineering unit. (See Appendix A.4 for program listing.)

This two-step process is presently required due to the calibration process utilized in this phase. Currently, transducers are excited using known inputs; and the transducer response is measured using a voltmeter. A suggested improvement in this process is to bypass the voltmeter, using the digital recording system in the calibration process. This improvement is planned to be implemented upon construction of the calibration rig suggested in Chapter 9.

5.4 Quick-Look Plots

The next step in the data analsys procedure is making the quick-look plots. The program of Appendix A.5 is used to do this. Basically this program plots the transducer outputs (uncorrected for C.G. location, etc.) on the graphics CRT. This is a rapid means of determining the portion of the recorded data that has the proper aircraft modes excited and is thus suitable for further analysis. Operator interaction has been minimized to reduce the overall time required for this step.

5.5 Detailed Engineering Conversion

This program is used to do a rigorous conversion of the data into the form required for the MMLE technique. Accounted for in this procedure are accurate transducer calibrations and instrument position corrections.

The first step in the instrument position correction process is to account for the misalignment between the transducers and the aircraft body axis. Secondly a correction must be applied to correct for the distance from the transducer center of gravity to the airplane center of gravity. The following equations are used. (See Reference 28 for a more rigorous presentation.)

$$\begin{aligned} \theta_B &= \theta_M - \theta_I \\ p_B &= p_M \cos(\theta_I) + r_M \sin(\theta_I) \qquad [5.1(a)] \\ r_B &= -p_M \sin(\theta_I) + r_M \cos(\theta_I) \end{aligned}$$

$$A_{X_B} = A_{X_M} \cos(\theta_I) - A_{N_M} \sin(\theta_I) + (r_B{}^2 + q_B{}^2) \frac{\bar{X}}{g} - (pq - \dot{r}) \frac{\bar{Y}}{g} - (pr + \dot{q}) \frac{\bar{Z}}{g}$$

$$A_{Y_B} = A_{Y_M} - (pq + \dot{r}) \frac{\bar{X}}{g} + (p^2 + r^2) \frac{\bar{Y}}{g} - (qr - \dot{p}) \frac{\bar{Z}}{g} \qquad [5.1(b)]^*$$

$$A_{N_B} = A_{X_M} \sin(\theta_I) + A_{N_M} \cos(\theta_I) + (pr - \dot{q}) \frac{\bar{X}}{g} + (qr + \dot{p}) \frac{\bar{Y}}{g} - (p^2 + q^2) \frac{\bar{Z}}{g}$$

where

B indicates Body axis at airplane center of gravity

M indicates as Measured by transducer

I indicates as Installed wrt Body axis at airplane center of gravity

This step also involved checking for and correcting any obvious data errors. If any filtering of unwanted noise is required, it would also be done at this stage; however, none has been needed to date. The quick-look plots are used as the major aid in this process.

A program listing is contained in Appendix A.6.

5.6 Modified Maximum Likelihood Estimator

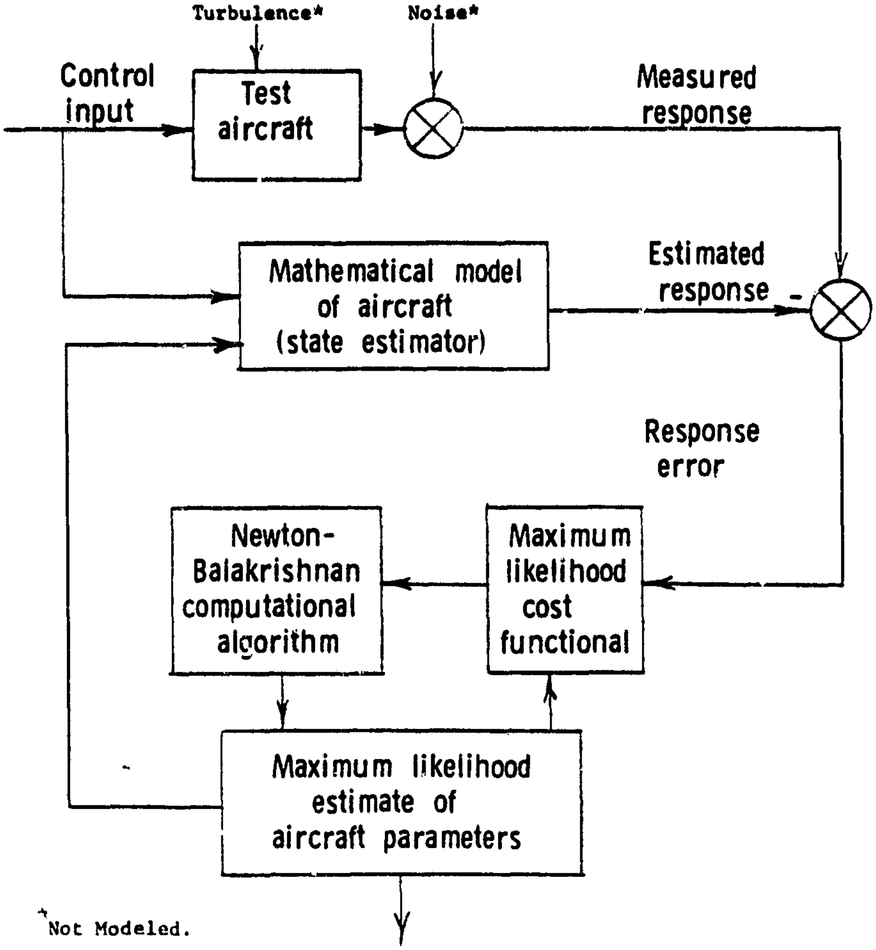
The flight data were processed through the Modified Maximum Likelihood Estimator (MMLE) developed by NASA (see References 12-16). This technique has been used by NASA for over 12 years. A simplified program (NASA Dryden "BONES" version of MMLE) has been placed on the MINC 11/03 computer. The actual program listings are included in Appendix A.7. Described here is the theory used in this technique, and some of the assumptions made for the KU-FRL version.

---

*Where $\dot{p}$, $\dot{q}$, and $\dot{r}$ are required, these are determined by digitally differentiating the p, q, and r measurements.

5.6.1 Parameter Estimation

The MMLE estimator is an iterative process that determines the coefficients of a given set of linear equations describing the motion of the aircraft. It does this by comparing the difference between actual in-flight measured responses of various states, and the predicted responses of these states using an estimate of the coefficients. The actual measured control input is used as the input for the estimating procedure. The estimated coefficients are updated each iteration, using the differences as determined above. The flow chart below shows the MMLE concept.



*Not Modeled.

Figure 5.2 Maximum likelihood estimation concept (from Reference 13)

5.6.2 Mathematical Model

The mathematical model used to describe the airplane is derived from the small perturbation equations of motion (see Reference 22).* These are shown here explicitly, in the non-dimensional form.

- for longitudinal (from Reference 22, Equation 6.1):

$$\begin{aligned}
m\dot{u} &= -mg\Theta\cos\Theta_1 + \bar{q}_1 S(-(C_{D_u} + 2C_{D_1})\frac{u}{U_1} + (C_{T_{x_u}} + 2C_{T_{x_1}})\frac{u}{U_1} - (C_{D_\alpha} - C_{L_1})\alpha - C_{D_{\delta_E}}\delta_E) \\
m(\dot{w} - U_1 q) &= -mg\Theta\sin\Theta_1 + \bar{q}_1 S(-(C_{L_u} + 2C_{L_1})\frac{u}{U_1} - (C_{L_\alpha} + C_{D_1})\alpha - C_{L_{\dot{\alpha}}}\frac{\dot{\alpha}\bar{c}}{2U_1} - C_{L_q}\frac{q\bar{c}}{2U_1} - C_{L_{\delta_E}}\delta_E) \\
I_{yy}\dot{q} &= \bar{q}_1 S\bar{c}((C_{m_u} + 2C_{m_1})\frac{u}{U_1} + (C_{m_{T_u}} + 2C_{m_{T_1}})\frac{u}{U_1} + C_{m_\alpha}\alpha + C_{m_{T_\alpha}}\alpha + C_{m_{\dot{\alpha}}}\frac{\dot{\alpha}\bar{c}}{2U_1} + C_{m_q}\frac{q\bar{c}}{2U_1} + C_{m_{\delta_E}}\delta_E)
\end{aligned} \qquad [5.2(a)]$$

- for lateral (from Reference 22, Equation 6.2):

$$\begin{aligned}
m(\dot{v} + U_1 r) &= mg\phi\cos\Theta_1 + \bar{q}_1 S(C_{y_\beta}\beta + C_{y_p}\frac{pb}{2U_1} + C_{y_r}\frac{rb}{2U_1} + C_{y_{\delta_A}}\delta_A + C_{y_{\delta_R}}\delta_R) \\
I_{xx}\dot{p} - I_{xz}\dot{r} &= \bar{q}_1 Sb(C_{\ell_\beta}\beta + C_{\ell_p}\frac{pb}{2U_1} + C_{\ell_r}\frac{rb}{2U_1} + C_{\ell_{\delta_A}}\delta_A + C_{\ell_{\delta_R}}\delta_R) \\
I_{zz}\dot{r} - I_{xz}\dot{p} &= \bar{q}_1 Sb\,(C_{n_\beta}\beta + C_{n_{T_\beta}}\beta + C_{n_p}\frac{pb}{2U_1} + C_{n_r}\frac{rb}{2U_1} + C_{n_{\delta_A}}\delta_A + C_{n_{\delta_R}}\delta_R)
\end{aligned} \qquad [5.2(b)]$$

Using the definitions shown in Table 5.2, Equations [5.2] can be converted to the dimensional form shown below.

- for longitudinal (from Reference 22, Equation 6.72):

$$\begin{aligned}
\dot{u} &= -g\theta\cos\theta_1 + X_u u + X_{T_u} u + X_\alpha\alpha + X_{\delta_E}\delta_E \\
\dot{w} - U_1 q &= -g\theta\sin\theta_1 + Z_u u + Z_\alpha\alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_{\delta_E}\delta_E \\
\dot{q} &= M_u u + M_{T_u} u + M_\alpha\alpha + M_{T_\alpha}\alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_{\delta_E}\delta_E
\end{aligned} \qquad [5.3(a)]$$

---

*The derivatives in Reference 22 are for the stability axes system. See Appendix B for conversion to the Body axes used in this report.

Table 5.2(a) Longitudinal Dimensional Stability Derivatives *

$$X_u = \frac{-\bar{q}_1 S(C_{D_u} + 2C_{D_1})}{mU_1} \quad (\text{sec}^{-1})$$

$$X_{T_u} = \frac{\bar{q}_1 S(C_{T_{x_u}} + 2C_{T_{x_1}})}{mU_1} \quad (\text{sec}^{-1})$$

$$X_\alpha = \frac{-\bar{q}_1 S(C_{D_\alpha} - C_{L_1})}{m} \quad (\text{ft sec}^{-2})$$

$$X_{\delta_E} = \frac{-\bar{q}_1 S C_{D_{\delta_E}}}{m} \quad (\text{ft sec}^{-2})$$

$$Z_u = -\frac{\bar{q}_1 S(C_{L_u} + 2C_{L_1})}{mU_1} \quad (\text{sec}^{-1})$$

$$Z_\alpha = -\frac{\bar{q}_1 S(C_{L_\alpha} + C_{D_1})}{m} \quad (\text{ft sec}^{-2})$$

$$Z_{\dot{\alpha}} = -\frac{\bar{q}_1 S C_{L_{\dot{\alpha}}} \bar{c}}{2mU_1} \quad (\text{ft sec}^{-1})$$

$$Z_q = -\frac{\bar{q}_1 S C_{L_q} \bar{c}}{2mU_1} \quad (\text{ft sec}^{-1})$$

$$Z_{\delta_E} = -\frac{\bar{q}_1 S C_{L_{\delta_E}}}{m} \quad (\text{ft sec}^{-2})$$

$$M_u = \frac{\bar{q}_1 S\bar{c}(C_{m_u} + 2C_{m_1})}{I_{yy}U_1} \quad (\text{ft}^{-1}\ \text{sec}^{-1})$$

$$M_{T_u} = \frac{\bar{q}_1 S\bar{c}(C_{m_{T_u}} + 2C_{m_{T_1}})}{I_{yy}U_1} \quad (\text{ft}^{-1}\ \text{sec}^{-1})$$

$$M_\alpha = \frac{\bar{q}_1 S\bar{c} C_{m_\alpha}}{I_{yy}} \quad (\text{sec}^{-2})$$

$$M_{T_\alpha} = \frac{\bar{q}_1 S\bar{c} C_{m_{T_\alpha}}}{I_{yy}} \quad (\text{sec}^{-2})$$

$$M_{\dot{\alpha}} = \frac{\bar{q}_1 S\bar{c}^2 C_{m_{\dot{\alpha}}}}{2I_{yy}U_1} \quad (\text{sec}^{-1})$$

$$M_q = \frac{\bar{q}_1 S\bar{c}^2 C_{m_q}}{2I_{yy}U_1} \quad (\text{sec}^{-1})$$

$$M_{\delta_E} = \frac{\bar{q}_1 S\bar{c} C_{m_{\delta_E}}}{I_{yy}} \quad (\text{sec}^{-2})$$

* from Reference 22, Table 6.3, page 413

Table 5.2(b) Lateral-Directional Dimensional Stability Derivatives *

$$Y_{\beta} = \frac{\bar{q}_1 S C_{y_{\beta}}}{m} \quad (\text{ft sec}^{-2})$$

$$Y_{p} = \frac{\bar{q}_1 S b C_{y_{p}}}{2mU_1} \quad (\text{ft sec}^{-1})$$

$$Y_{r} = \frac{\bar{q}_1 S b C_{y_{r}}}{2mU_1} \quad (\text{ft sec}^{-1})$$

$$Y_{\delta_A} = \frac{\bar{q}_1 S C_{y_{\delta_A}}}{m} \quad (\text{ft sec}^{-2})$$

$$Y_{\delta_R} = \frac{\bar{q}_1 S C_{y_{\delta_R}}}{m} \quad (\text{ft sec}^{-2})$$

$$L_{\beta} = \frac{\bar{q}_1 S b C_{\ell_{\beta}}}{I_{xx}} \quad (\text{sec}^{-2})$$

$$L_{p} = \frac{\bar{q}_1 S b^2 C_{\ell_{p}}}{2I_{xx}U_1} \quad (\text{sec}^{-1})$$

$$L_{r} = \frac{\bar{q}_1 S b^2 C_{\ell_{r}}}{2I_{xx}U_1} \quad (\text{sec}^{-1})$$

$$L_{\delta_A} = \frac{\bar{q}_1 S b C_{\ell_{\delta_A}}}{I_{xx}} \quad (\text{sec}^{-2})$$

$$L_{\delta_R} = \frac{\bar{q}_1 S b C_{\ell_{\delta_R}}}{I_{xx}} \quad (\text{sec}^{-2})$$

$$N_{\beta} = \frac{\bar{q}_1 S b C_{n_{\beta}}}{I_{zz}} \quad (\text{sec}^{-2})$$

$$N_{T_{\beta}} = \frac{\bar{q}_1 S b C_{n_{T_{\beta}}}}{I_{zz}} \quad (\text{sec}^{-2})$$

$$N_{p} = \frac{\bar{q}_1 S b^2 C_{n_{p}}}{2I_{zz}U_1} \quad (\text{sec}^{-1})$$

$$N_{r} = \frac{\bar{q}_1 S b^2 C_{n_{r}}}{2I_{zz}U_1} \quad (\text{sec}^{-1})$$

$$N_{\delta_A} = \frac{\bar{q}_1 S b C_{n_{\delta_A}}}{I_{zz}} \quad (\text{sec}^{-2})$$

$$N_{\delta_R} = \frac{\bar{q}_1 S b C_{n_{\delta_R}}}{I_{zz}} \quad (\text{sec}^{-2})$$

* from Reference 22, Table 6.8, page 445

- for lateral (from Reference 22, Equation 6.141):

$$\dot{v} + U_1 r = g\phi\cos\theta_1 + Y_\beta\beta + Y_p p + Y_r r + Y_{\delta_A}\delta_A + Y_{\delta_R}\delta_R$$

$$\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} = L_\beta\beta + L_p p + L_r r + L_{\delta_A}\delta_A + L_{\delta_R}\delta_R \qquad [5.3(b)]$$

$$\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} = N_\beta\beta + N_{T_\beta}\beta + N_p p + N_r r + N_{\delta_A}\delta_A + N_{\delta_R}\delta_R$$

Using the concept of state variable theory (see Reference 22), Equation [5.3] can be written in the following form:

$$[R]\,\{\dot{x}(t)\} = [A]\,\{x(t)\} + [B]\,\{u(t)\} \qquad [5.4]$$

where

$\{x(t)\}$ = state vector

$[R]$ = acceleration transformation matrix

$[A]$ = stability matrix

$[B]$ = control matrix

$\{u(t)\}$ = control vector.

Equation 5.4 can be written more explicitly in the form which follows:

- for longitudinal (where [R] = identity matrix):

$$\frac{d}{dt}\begin{bmatrix} q \\ U \\ \alpha \\ \theta \end{bmatrix} = \begin{bmatrix} M'_q & M'_u & M'_\alpha & M'_\theta \\ 0 & X'_u & X'_\alpha & -g\cos(\theta_1) \\ \dfrac{Z_q + U_1}{U_1 - Z_{\dot{\alpha}}} & Z'_u & Z'_\alpha & \dfrac{-g}{U_1 - Z_{\dot{\alpha}}}\sin(\theta_1)\cos(\phi_1) \\ \cos(\phi_1) & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} q \\ U \\ \alpha \\ \theta \end{bmatrix} + \begin{bmatrix} M_{\delta_E}' & M_{\delta_c}' & M'_o \\ X_{\delta_E}' & X_{\delta_c}' & X'_o \\ Z_{\delta_E}' & Z_{\delta_c}' & Z'_o \\ 0 & 0 & \dot{\theta}'_o \end{bmatrix}\begin{bmatrix} \delta_E \\ \delta_c \\ 1 \end{bmatrix} \qquad [5.5(a)]$$

(See Table 5.3 for explicit definition of these terms.)

- for lateral:

$$[R]\,\frac{d}{dt}\begin{bmatrix} p \\ r \\ \beta \\ \theta \end{bmatrix} = \begin{bmatrix} L'_p & L'_r & L'_\beta & 0.0 \\ N'_p & N'_r & N'_\beta & 0.0 \\ \sin(\alpha_1) & -\cos(\alpha_1) & Y'_\beta & \frac{g}{U_1}\cos(\theta_1)\cos(\phi_1) \\ 1.0 & \cos(\phi_1)\tan(\theta_1) & 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} L_{\delta_A}' & L_{\delta_r}' & L'_o \\ N_{\delta_A}' & N_{\delta_r}' & N'_o \\ Y_{\delta_A}' & Y_{\delta_r}' & Y'_o \\ 0.0 & 0.0 & \dot{\phi}'_o \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_r \\ 1 \end{bmatrix} \qquad [5.5(b)]$$

(See Table 5.3 for explicit definition of these terms.)

To allow determination of states other than the ones contained in $\{x(t)\}$, the following expression can be derived:

$$\{y(t)\} = \left[\frac{I}{G}\right]\{x(t)\} + \left[\frac{O}{H}\right]\{u(t)\} + \{\frac{O}{v}\}^{*} \qquad [5.6]$$

where

$\{y(t)\}$ = computed observation vector

$[G]$ = observation matrix

$[H]$ = observation matrix

$\{v\}$ = variable bias vector.

(See Table 5.4 for explicit definition of these terms.)

The computed observation vector, $\{y(t)\}$, corresponds to the measured observation vector, shown here:

$$\{z(t)\} = \{y(t)\} + \{\eta(t)\}^{*} \qquad [5.7]$$

where

$\{z(t)\}$ = measured observation vector = $\{\theta, \phi, p, q, r, A_X, A_Y, A_N, \delta_E, \delta_A, \delta_R, P_S, P_D, T\}$

$\{\eta(t)\}$ = measured noise vector.

From the terms of Equations [5.4], [5.6], [5.7], the vector

---

*From Reference 16

Table 5.3(a) Longitudinal, Dimensional State Vector Stability Derivatives

$$M'_q = M_q + M_{\dot{\alpha}} \frac{Z_q + U_1}{U_1 - Z_{\dot{\alpha}}} \approx M_q + M_{\dot{\alpha}} \quad (\mathrm{sec}^{-1})$$

$$M'_u = M_u + M_{T_u} + \frac{M_{\dot{\alpha}} Z_u}{U_1 - Z_{\dot{\alpha}}} \quad (\mathrm{ft}^{-1}\ \mathrm{sec}^{-1})$$

$$M'_\alpha = M_\alpha + M_{T_\alpha} + \frac{M_{\dot{\alpha}} Z_\alpha}{U_1 - Z_{\dot{\alpha}}} \quad (\mathrm{sec}^{-2})$$

$$M'_\theta = \frac{-M_{\dot{\alpha}} g \sin(\theta_1) \cos(\phi_1)}{U_1 - Z_{\dot{\alpha}}} \approx 0 \quad (\mathrm{sec}^{-2})^{\dagger}$$

$$M'_{\delta_{E,c}} = M_{\delta_{E,c}} + \frac{M_{\dot{\alpha}} Z_{\delta_{E,c}}}{U_1 - Z_{\dot{\alpha}}} \quad (\mathrm{sec}^{-2})$$

M'_o = pitching moment equation bias* (sec^{-2})

$$X'_u = X_u + X_{T_u} \quad (\mathrm{sec}^{-1})$$

$$X'_\alpha = X_\alpha \quad (\mathrm{ft\ sec}^{-2})$$

$$X'_{\delta_{E,c}} = X_{\delta_{E,c}} \quad (\mathrm{ft\ sec}^{-2})$$

X'_o = longitudinal acceleration equation bias $(\mathrm{ft\ sec}^{-2})^*$

$$\frac{Z_q + U_1}{U_1 - Z_{\dot{\alpha}}} \approx 1.0^{\dagger}$$

$$Z'_u = \frac{Z_u}{U_1 - Z_{\dot{\alpha}}} \approx \frac{Z_u}{U_1} \quad (\mathrm{ft}^{-1})$$

$$Z'_\alpha = \frac{Z_\alpha}{U_1 - Z_{\dot{\alpha}}} \approx \frac{Z_\alpha}{U_1} \quad (\mathrm{sec}^{-1})$$

$$\frac{-g \sin(\theta_1)\cos(\phi_1)}{U_1 - Z_{\dot{\alpha}}} \approx \frac{-g \sin(\theta_1)\cos(\phi_1)}{U_1} \quad (\mathrm{sec}^{-1})^{\dagger}$$

$$Z'_{\delta_{E,c}} = \frac{Z_{\delta_{E,c}}}{U_1 - Z_{\dot{\alpha}}} \approx \frac{Z_{\delta_{E,c}}}{U_1} \quad (\mathrm{sec}^{-1})$$

Z'_o = normal acceleration equation bias $(\mathrm{sec}^{-1})^*$

$\dot{\theta}'_o$ = pitch rate equation bias $(\mathrm{sec}^{-1})^*$

* Note: The equation bias terms are used to allow prediction of the complete state which is made up of the steady state and the perturbed state.

† Note: With the approximations above, Equation [5.5(a)] is rewritten as;

$$\frac{d}{dt}\begin{bmatrix} q \\ U \\ \alpha \\ \theta \end{bmatrix} = \begin{bmatrix} M'_q & M'_u & M'_\alpha & 0 \\ 0 & X'_u & X'_\alpha & -\cos(\theta_1) g \\ 1 & Z'_u & Z'_\alpha & -\sin(\theta_1)\cos(\phi_1)\frac{g}{U_1} \\ \cos(\phi_1) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ U \\ \alpha \\ \theta \end{bmatrix} + \begin{bmatrix} M'_{\delta_E} & M'_{\delta_c} & M'_o \\ X'_{\delta_E} & X'_{\delta_c} & X'_o \\ Z'_{\delta_E} & Z'_{\delta_c} & Z'_o \\ 0 & 0 & \dot{\theta}_o \end{bmatrix} \begin{bmatrix} \delta_E \\ \delta_c \\ 1 \end{bmatrix}$$

Table 5.3(b) Lateral, Dimensional State Vector Stability Derivatives

$L'_p = L_p$ (sec^{-1})

$L'_r = L_r$ (sec^{-1})

$L'_\beta = L_\beta$ (sec^{-1})

$L_{\delta_A}' = L_{\delta_A}$ (sec^{-2})

$L_{\delta_r}' = L_{\delta_r}$ (sec^{-2})

$N'_p = N_p$ (sec^{-1})

$N'_r = N_r$ (sec^{-1})

$N_{\delta_A}' = N_{\delta_A}$ (sec^{-2})

$N_{\delta_r}' = N_{\delta_r}$ (sec^{-2})

$N'_\beta = N_\beta + N_{T_\beta}$ (sec^{-1})

$Y'_\beta = \frac{Y_\beta}{U_1}$ (sec^{-1})

$Y_{\delta_A}' = \frac{Y_{\delta_A}}{U_1}$ (sec^{-1})

$Y_{\delta_r}' = \frac{Y_{\delta_r}}{U_1}$ (sec^{-1})

Y'_o = lateral acceleration equation bias (sec^{-1}) *

$\dot{\phi}'_o$ = roll rate equation bias (sec^{-1}) *

L'_o = rolling moment equation bias (sec^{-2}) *

N'_o = yawing moment equation bias (sec^{-2}) *

*NOTE: The equation bias terms are used to allow prediction of the complete state which is made up of the steady state and the perturbed state.

$$[R] = \begin{bmatrix} 1.0 & -\frac{I_{xz}}{I_{xx}} & 0 & 0 \\ -\frac{I_{xz}}{I_{zz}} & 1.0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ for $I_{xz} \simeq 0$; [R] = identity matrix

Table 5.4 Matrices Used in Observation equation

LONGITUDINAL

$$\{y(t)\}^{\dagger} = \{q,\ U,\ \alpha,\ \theta,\ \dot{q},\ A_X,\ A_N\}$$

$$\left[\frac{0}{v}\right]^{\dagger} = \{0,\ 0,\ 0,\ 0,\ \dot{q}_{bias},\ A_{X_{bias}},\ A_{N_{bias}}\}$$

$$\left[\frac{0}{H}\right] = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline M_{\delta_E}' & M_{\delta_c}' & M_o' \\ \frac{X_{\delta_E}'}{g} & \frac{X_{\delta_c}'}{g} & \frac{X_o'}{g} \\ \frac{-U_1}{g} Z_{\delta_E}' & \frac{-U_1}{g} Z_{\delta_c}' & \frac{-U_1}{g} Z_o' \end{array}\right] \qquad \left[\frac{I}{G}\right] = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline M_q' & M_u' & M_\alpha' & 0 \\ 0 & \frac{X_u'}{g} & 0 & 0 \\ 0 & 0 & \frac{-U_1 Z_u'}{g} & 0 \end{array}\right]$$

LATERAL

$$\{y(t)\}^{\dagger} = \{p,\ r,\ \beta,\ \phi,\ \dot{p},\ \dot{r},\ A_Y\}$$

$$\left[\frac{0}{v}\right]^{\dagger} = \{0,\ 0,\ 0,\ 0,\ \dot{p}_{bias},\ \dot{r}_{bias},\ A_{Y_{bias}}\}$$

$$\left[\frac{0}{H}\right] = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline L_{\delta_A}' & L_{\delta_R}' & L_o' \\ N_{\delta_A}' & N_{\delta_R}' & N_o' \\ \frac{U_1}{g} Y_{\delta_A}' & \frac{U_1}{g} Y_{\delta_R}' & \frac{U_1}{g} Y_o' \end{array}\right] \qquad \left[\frac{I}{G}\right] = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \text{[illegible]} & 0 & 1 \\ \hline L_p & L_r' & L_\beta' & 0 \\ N_p' & N_r' & N_\beta' & 0 \\ 0 & 0 & \frac{Y_\beta' U_1}{g} & 0 \end{array}\right]$$

$$\{c\} = f([A], [B], [G], [H], \{v\}) \tag{5.8}$$

(where f indicates "a function of") is defined as the vector of unknowns. It is this vector that the MMLE method estimates. MMLE determines the unknowns ($\{c\}$) by minimizing the cost function given by:

$$J = \frac{1}{T}\int_0^T \{z(t) - y(t)\}^{\dagger}[D]\,\{z(t) - y(t)\}\,dt \quad * \tag{5.9}$$

(T, t: indicates time)

or approximately in the discrete case:

$$J = \frac{1}{(N-1)}\sum_{i=1}^{N} \{z_i - y_i\}^{\dagger}[D]\,\{z_i - y_i\}\,\Delta t \quad * \tag{5.10}$$

(where i is the time index, and N the number of time points).

The weighting matrix, [D], is used to provide emphasis on the various measured states; in other words, to allow greater emphasis on the more accurate transducers, or the transducers that are more important to describe the maneuver performed.

The value of the cost functional, J, is minimized using the Newton-Raphson* method. This technique is an iterative procedure, utilizing an estimated value of the vector of unknowns, $\{c\}$, and the first and second gradients of the cost functional, J, with respect to the vector of unknowns, $\{c\}$. The equation

$$\{c\}_L = \{c\}_{L-1} - \{\nabla_c^2\, J\}_L^{-1}\,\{\nabla_c\, J\}_L^{\dagger} \quad * \tag{5.11}$$

(where L is the iteration number) is used to revise estimates for the vector of unknowns, $\{c\}$. The first and second gradients are given by:

---

*From Reference 16

C-2

$$\{\nabla_c J\} = \frac{2}{N-1}\sum_{i=1}^{N}\{z_i - y_i\}^{\dagger}[D]\ \nabla_c\{z_i - y_i\} \quad ^{*} \qquad [5.12]$$

$$\{\nabla_c^2 J\} = \frac{2}{N-1}\sum_{i=1}^{N}\nabla_c\{z_i - y_i\}^{\dagger}[D]\nabla_c\{z_i - y_i\} + \frac{2}{N-1}\sum_{i=1}^{N}\{z_i - y_i\}^{\dagger}[D]\nabla_c^2\{z_i - y_i\} \quad ^{*} \qquad [5.13]$$

The Balakrishnan* modification makes use of the fact that the term $\nabla_c^2\{z_i - y_i\}$ approaches zero with convergence and is thus neglected. The expression for the second gradient becomes:

$$\{\nabla_c^2 J\} = \frac{2}{N-1}\sum_{i=1}^{N}\nabla_c\{z_i - y_i\}^{\dagger}[D]\nabla_c\{z_i - y_i\} \qquad [5.14]$$

After several iterations the cost function converges near some small value. At this point the parameters of Equation [5.5] have been modified to obtain their most likely value which results in the best fit of the measured states.

5.6.3 Assumptions Used in Data Reduction

The following inputs and modifications were made to the MMLE method, allowing effective use of the technique on the MINC 11/03 computer.

Initial estimates of the derivatives in Equation [5.5] were obtained using the analytical methods of Reference 22. Although the MMLE technique does not require accurate knowledge of these derivatives, this procedure does speed convergence.

The MMLE program usually uses a modified least squares method for the first iteration to estimate the derivatives, as an aid to

---

*From Reference 16

speed convergence. This, however, requires measurement of most of the states indicated in [5.5]. The instrumentation package uses only a minimum of transducers, and all the states required for this least squares estimate are not measured. Using the least squares procedure would result in divergence of the first iteration. Therefore, the least squares estimate was not used, which did slow convergence of the derivatives.

A diagonal multiplying factor allows control over how large a change is made to the derivatives after each iteration. Too large a value of this factor causes sluggishness in the convergence, and too small a value will cause divergence. Further analysis into this factor will indicate its optimum value for best convergence.

The weighting matrix, [D], of Equation [5.9], was chosen after analysis of the instrumentation error magnitudes. The first run through the MMLE program, with measurements from this instrumentation package, provided a weighted error for each measurement state. As suggested in Reference 16, the values for the weighting matrix were chosen to attempt to equalize the weighted errors. After the values for the weighting matrix were chosen for the instrument package, they were then left at this for further maneuver analysis.

5.7 Time History Plotting

The MMLE reduction method not only produces the estimates for the derivatives, but also calculates the estimated time history for the various states. This is stored on the data disc by the MMLE program. The programs presented in Appendix A.8 retrieve both the predicted time histories and the measured time histories and plot them together on the graphics CRT. These graphs of the flight test maneuver are the visual indication of the goodness of the predicted airplane derivatives.

----- ********** -----

As is evident from the many programs provided, the final results of a flight test maneuver are obtained only after a multi-step procedure. This is primarily due to the nature of the methods being used in aircraft flight testing, as well as the limitations of computer technology being used.

6. KU-FRL FLIGHT TEST PROGRAM

Two series of flight tests have been conducted using the KU-FRL Cessna 172. The first series, conducted under Phase I, is presented in Reference 2. Presented here are the basic concepts of the type of flight maneuvers required, and results of the Phase II test program.

6.1 Flight Test Maneuver

Traditional flight testing methods have utilized primarily steady-state flight paths for data collection. This was due mostly to the data acquisition systems available. Unfortunately, this required a highly trained and competent test pilot to obtain realistic and valuable results.

With the current transducer and acquisition system technology available, flight testing need no longer rely on steady-state maneuvers to allow accurate state measurement. This development has resulted in the newer flight testing methods utilizing dynamic maneuvers.

When techniques such as the MMLE are used, the literature (Reference 29) indicates that the nature of the maneuver is not critical to determine the aircraft characteristics. What is important when using these techniques is to ensure that the proper aircraft modes have been excited. For example, a longitudinal maneuver should excite both the short-period and phugoid modes of the airplane. This realization (i.e., non-critical flight path) leads to the possibility of using lesser qualified pilots and still obtaining accurate results. All testing done on this program has been done by a pilot who had no previous flight test experience.

The control inputs presented in the traces of Chapter 6.2 are typical of the type of maneuver required. Several frequencies are excited, which tends to increase the validity of the results obtained. Also the total energy input is approximately symmetrical. In other words, the motion produced in one direction is offset by the motion produced in the opposite direction a short time later.

6.2 Results of Flight Test Program

Presented here are results of the Phase II flight test program. All flights were done at the conditions of Table 6.1.*

Table 6.1 Cessna 172 Flight Test Conditions

Wing area (S)	174 ft^2
Wing span (b)	35.8 ft
Inertias*	
I_{xx}	1029 slug ft^2
I_{zz}	1891 slug ft^2
I_{yy}	1092 slug ft^2
Mass (m)	59.46 slug
Weight	1913 lb
Center of Gravity (Body Station)	41.3 inch
Mean chord ($\bar{c}$)	4.9 ft
Speed (U_1)	176 ft/sec
Dynamic Pressure ($\bar{q}_1$)	33.69 lb/ft^2
Altitude	3000 ft

The plots following show the absence of noise in the measurement, as well as the typical maneuver required.

---

*Estimated by Reference 30

The fit of the estimated states compared to the actual states in the longitudinal maneuvers is good (Figures 6.1-6.3). The only state that is off consistently is the A_X term, which appears to be affected by a phase shift. The cause of this phase shift has not been determined.

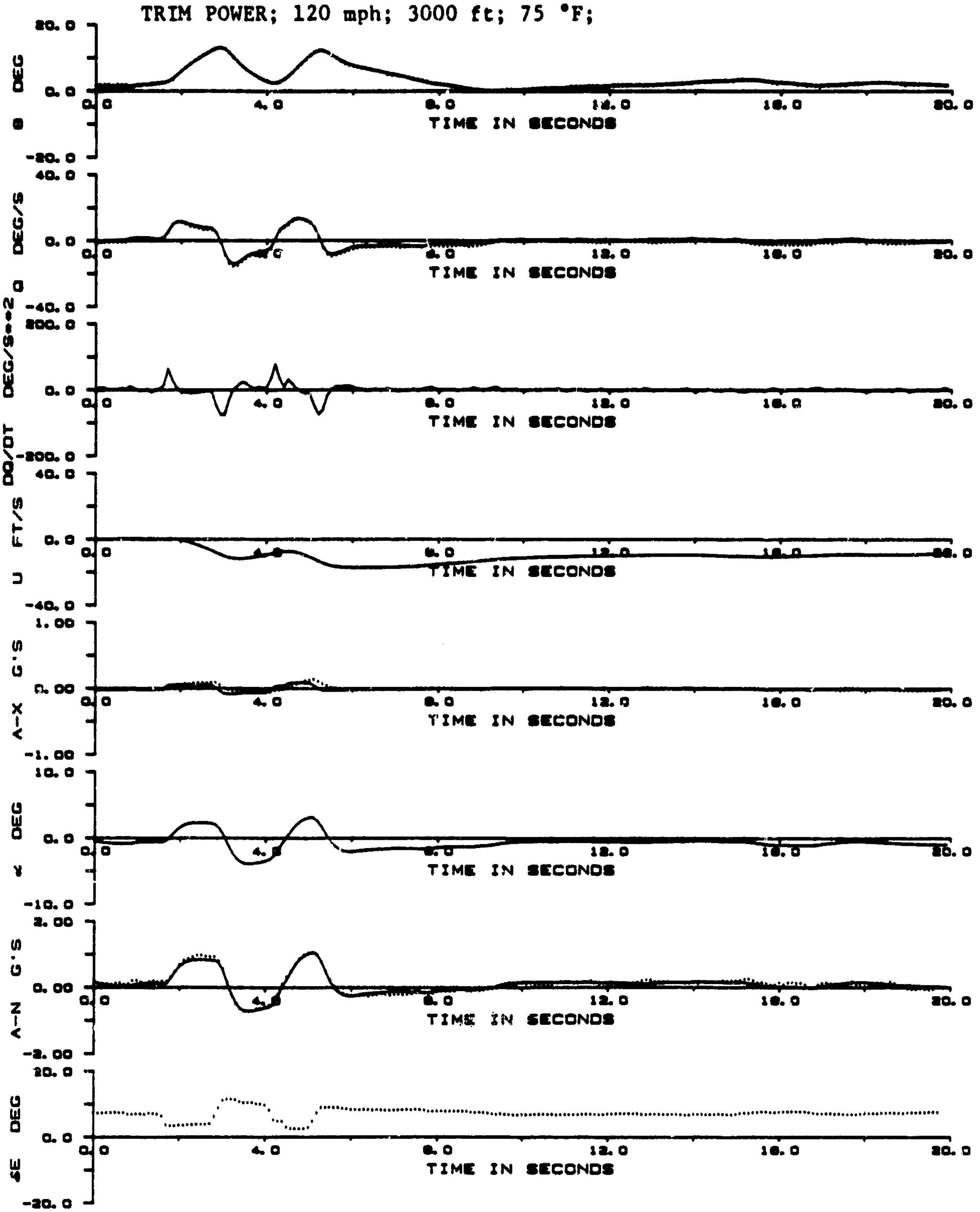
The estimated parameters have been compared with the analytical methods of Reference 22 and with flight test results obtained by NASA Langley on a Cessna 172 (Reference 30). This correlation is shown in Table 6.2 for the longitudinal maneuvers. It is seen that there is good correlation between some derivatives, but not between others. The best correlation appears to be with the one of run 23B, in which the speed derivatives have been held constant for the MMLE analysis.* This would tend to be the predicted result due to the mismatch in the A_X term, which is a major contributor to the speed prediction.

The lateral maneuvers are presented in Figures 6.4-6.6; and the correlation of derivatives, in Table 6.3. The fit of the measured and predicted states is again reasonable. Run 11 has the best fit as well as the overall best fit to the parameters. Again, however, the predicted coefficients are not within acceptable limits. The cause of this is not known.

Observing the rudder trace on Figure 6.6, what appears to be rudder float is evident (especially between 10 sec and 12 sec).

---

*This is the same procedure performed by NASA Langley, which makes no attempt to predict any speed derivatives.



TRIM POWER; 120 mph; 3000 ft; 75 °F;

Figure 6.1 Flight time history; Flight 19/10/80 Run 23A; Longitudinal

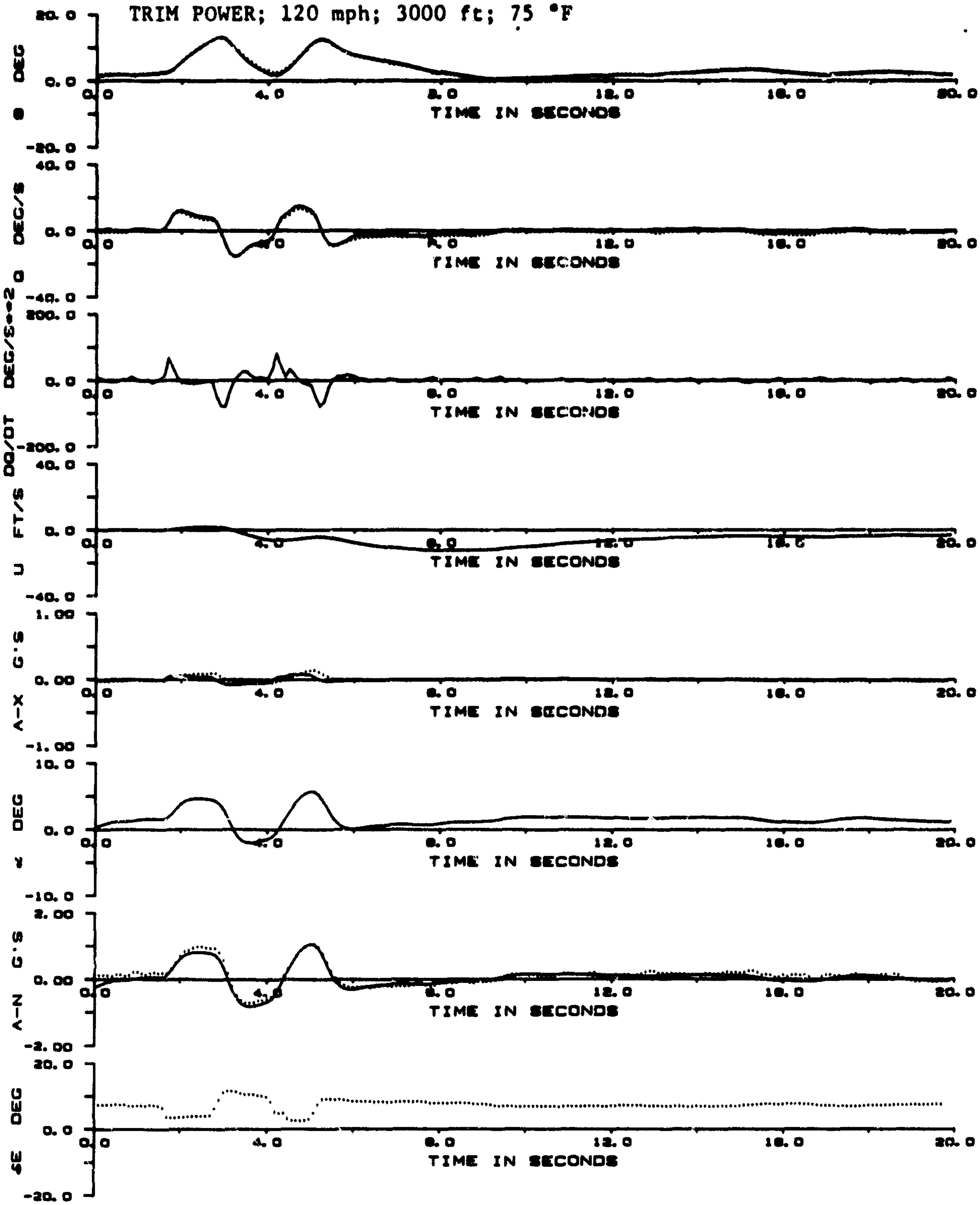

TRIM POWER; 120 mph; 3000 ft; 75 °F

Figure 6.2 Flight time history; Flight 19/10/80 Run 23B; Longitudinal

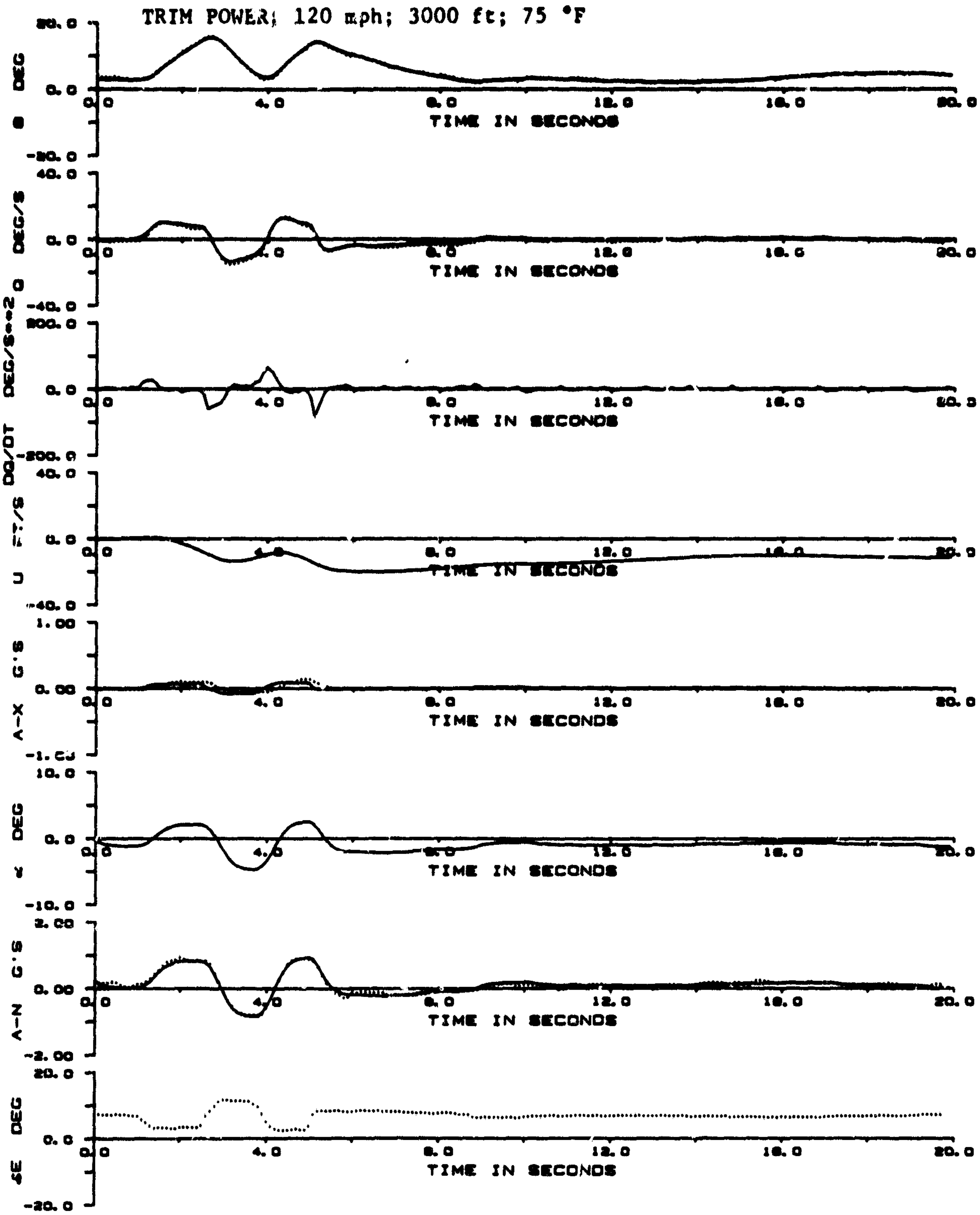


Figure 6.3 Flight time history; Flight 19/10/80 Run 51A; Longitudinal

Table 6.2 Comparison of Results, Longitudinal

STICK FIXED: AIRSPEED 176 ft/sec (120 MPH)					
Estimation Method	KU-FRL MMLE			NASA LANGLEY[1,2]	ANALYTICAL[2,3]
Gross Weight (lb)	1913			1848	2160
Center of Gravity (Body station, inches)	41.3			42.5	40.3
Flight No.	19/10/80				
Run No.	23A (Fig. 6.1)	23B (Fig. 6.2)	51A (Fig. 6.3)		
C_{m_q}'	(13%) -21.84	(3%) -18.72	(28%) -24.77	-19.34	(9%) -17.60
C_{m_u}'	0.093	CONSTANT 0	0.097	NOT PREDICTED	0
C_{m_α}'	(20%) -0.543	(17%) -0.563	(28%) -0.490	-0.678	(31%) -0.890
C_{X_u}'	-0.095	CONSTANT -0.100	-0.046	NOT PREDICTED	-0.100
C_{X_α}'	* -1.403	(27%) 0.688	* -1.450	0.54	(71%) 0.196
C_{Z_u}'	-0.003	CONSTANT -0.004	-0.003	NOT PREDICTED	0
C_{Z_α}'	(1%) -5.198	(13%) -4.534	(9%) -4.775	-5.22	(12%) -4.600
$C_{m_{\delta_E}}'$	(17%) -0.933	(17%) -0.927	(17%) -0.932	-1.118	(14%) -1.280
$C_{X_{\delta_E}}'$	-0.364	-0.303	-0.350	NOT PREDICTED	-0.060
$C_{Z_{\delta_E}}'$	(28%) -0.514	* 0.021	(60%) -0.161	-0.402	(7%) -0.430

( ) As compared with NASA Langley results.

* Wrong sign.

[1] Reference 30, Table IV, page 28, Maximum Likelihood Method, average of the two runs at Full Trim.

[2] See Appendix B for conversion to the Body axes system used in this report.

[3] Reference 22, Airplane A, page 590.

[4] C_{m_q}' has a large $C_{m_{\dot\alpha}}$ component which cannot be predicted.

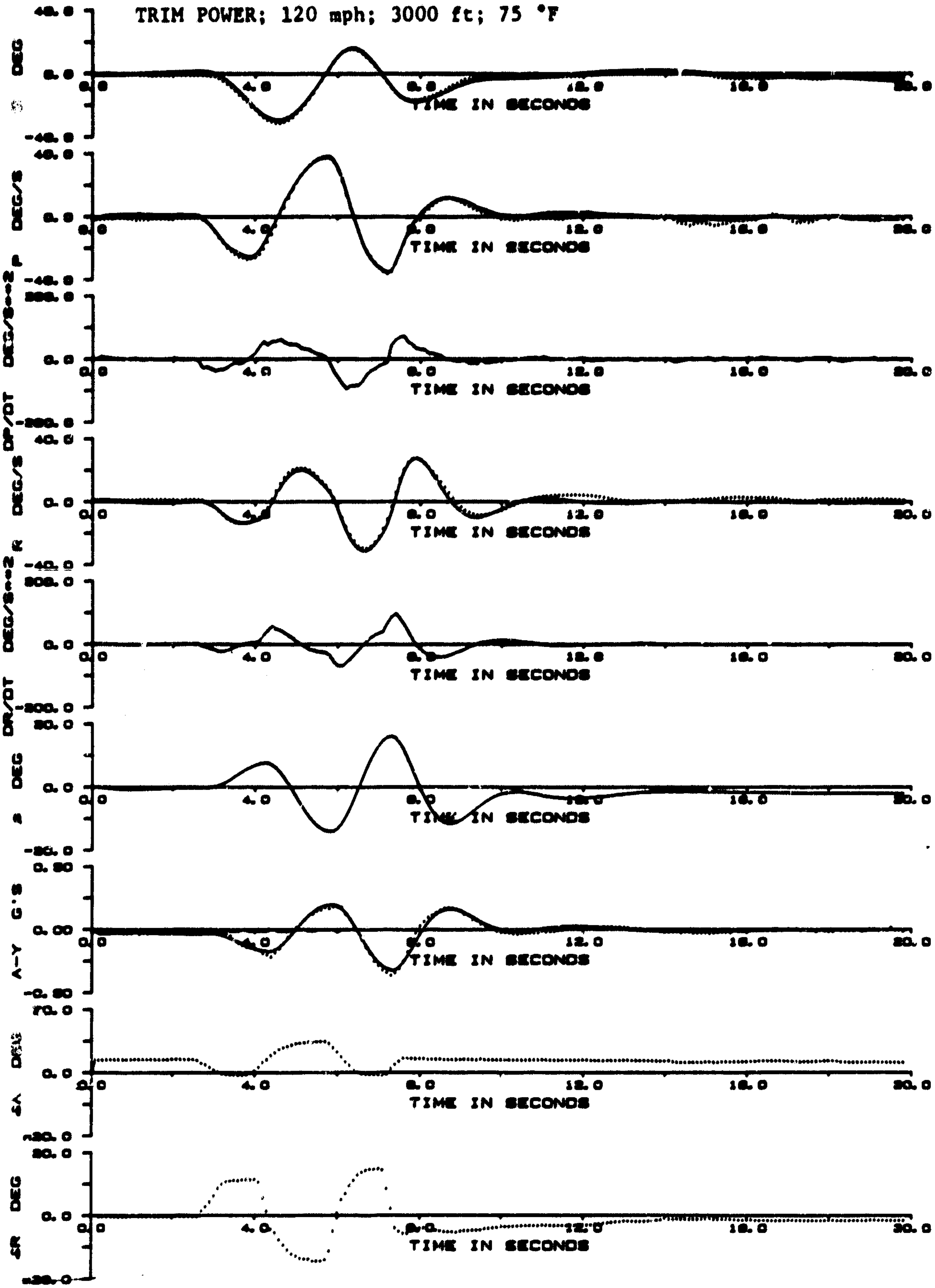


Figure 6.4 Flight time history; Flight 23/10/80 Run 11; Lateral

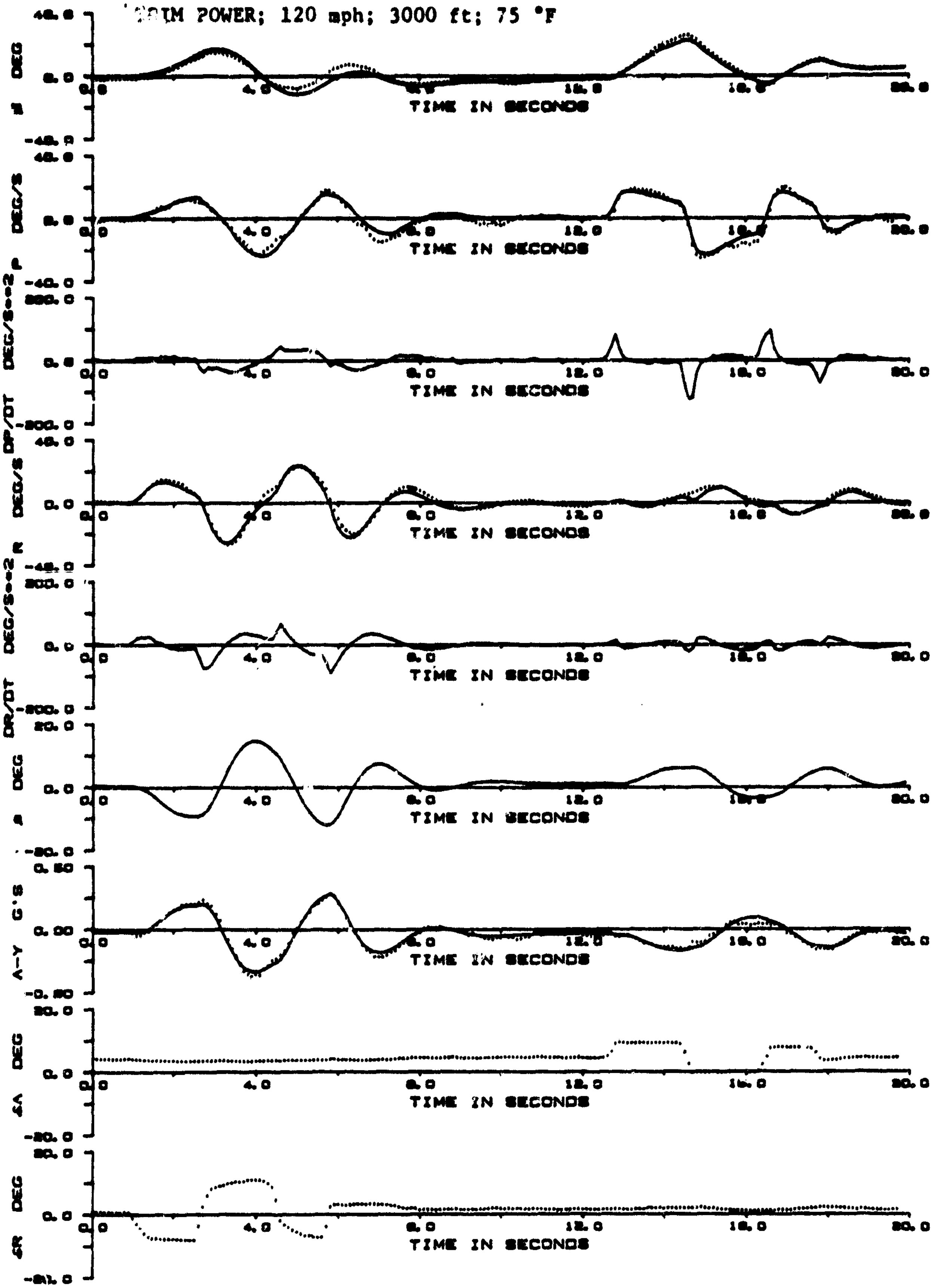


Figure 6.5 Flight time history; Flight 19/10/80 Run 26C; Lateral

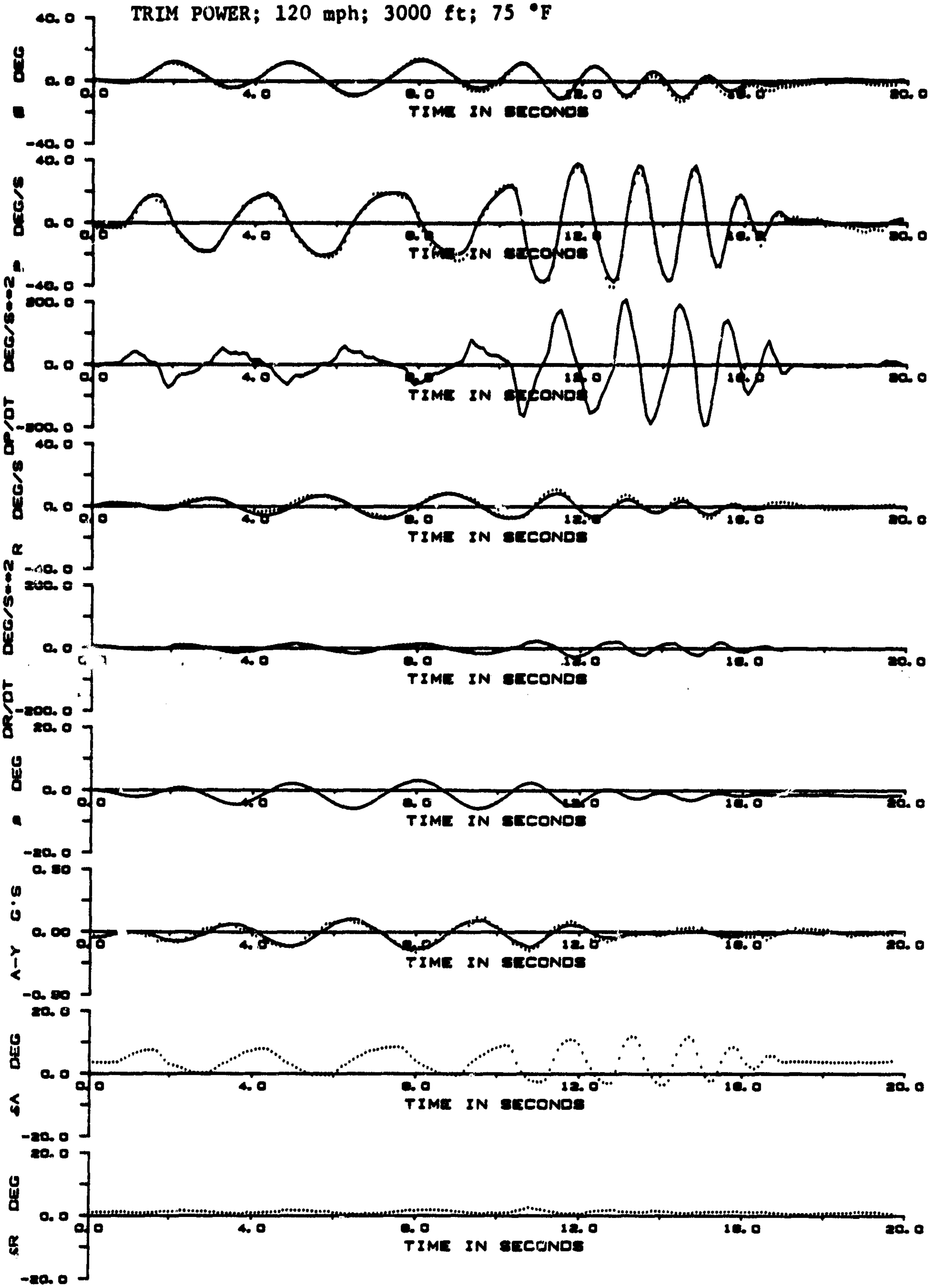

TRIM POWER; 120 mph; 3000 ft; 75 °F

Figure 6.6 Flight time history; Flight 19/10/80 Run 33B; Lateral

Table 6.3 Comparison of Results, Lateral

STICK FIXED: AIRSPEED 176 ft/sec (120 MPH)					
Estimation Method	KU-FRL MMLE			NASA LANGLEY[1,2]	ANALYTICAL[2,3]
Gross Weight (lb)	1913			1848	2160
Center of Gravity (Body station, inches)	41.3			42.5	40.3
Flight No.	23/10/80	19/10/80			
Run No.	11 (Fig. 6.4)	26C (Fig. 6.5)	33B Fig. 6.6)		
C_{ℓ_p}'	(13%) -0.402	(8%) -0.498	(24%) -0.351	-0.461	(2%) -0.470
C_{ℓ_r}'	(18%) 0.062	* -0.108	(83%) 0.139	0.076	(26%) 0.096
C_{ℓ_β}'	(34%) -0.049	(9%) -0.067	(54%) -0.034	-0.074	(20%) -0.089
C_{n_p}'	(2%) -0.062	(500%) -0.419	(21%) -0.076	-0.063	(52%) -0.030
C_{n_r}'	(14%) 0.109	(126%) -0.261	(26%) -0.071	-0.096	(3%) -0.099
C_{n_β}'	(16%) 0.037	(82%) 0.008	(16%) 0.037	0.044	(48%) 0.065
C_{y_β}'	(42%) -0.335	(20%) -0.464	(25%) -0.439	-0.582	(47%) -0.310
$C_{\ell_{\delta_A}}'$	(1%) 0.208	(1%) 0.207	(1%) 0.208	0.206	(1%) 0.178
$C_{n_{\delta_A}}'$	(10%) 0.009	(1000%) 0.128	(130%) 0.023	0.010	* -0.053
$C_{y_{\delta_A}}'$	-0.046	-0.050	-0.063	NOT PREDICTED	0
$C_{\ell_{\delta_R}}'$	(150%) 0.010	* -0.039	* -0.164	0.004	(275%) 0.015
$C_{n_{\delta_R}}'$	(23%) -0.040	(70%) -0.088	(33%) -0.035	-0.052	(27%) -0.066
$C_{y_{\delta_R}}'$	(97%) 0.003	(27%) 0.066	* -0.481	0.091	(105%) 0.187

( ) As compared with NASA Langley results.

* Wrong sign.

[1] Reference 30, Table VII, p. 32, case 34.

[2] See Appendix B for conversion to the Body axes system used in this report.

[3] Reference 22, Airplane A, page 590.

This maneuver was performed by holding the rudder pedals fixed, yet a float of 2°-3° is seen in the rudder. This magnitude of input could affect the parameters determined. This effect is due to a second order control surface term introduced by this float but not predicted by the MMLE mathematical model.

It is suggested that further work be done to evaluate whether this is the case, and perhaps to include control surface float into the mathematical representation. The effect of control surface float can be determined by varying the tension of the cable which moves the surface.

- - - - - ********** - - - - -

Possible problems that could be responsible for the differences in parameter prediction are listed here:

- Calibration of transducers. It is suggested that part of the error in parameters is due to inaccuracies in transducer calibration.

- Uniqueness. It has not yet been determined if the methods such as MMLE have a unique solution. The possibility does exist of more than one solution to any given maneuver.

- Control surface float. No attempt was made in this flight test program to ensure a minimum of float of the control surfaces. It is suggested that cable tensions be tightened to allowable maximums prior to flight testing.

7. CESSNA FLIGHT TEST PROGRAM

The versatility of this flight test package was demonstrated in a spin test program conducted by Cessna Aircraft. In this program the KU-FRL provided the data management portion of the instrumentation system described in this report. Cessna supplied the instrumentation and the airplane. A block diagram of this installation is shown in Figure 7.1

7.1 Instrumentation

The purpose of this program was to investigate the spin characteristics of Cessna's latest model 172 airplane. To do this, Cessna approached the KU-FRL as to the applicability of the instrumentation system for this type of test. After initial evaluation it was decided that the measurements described in Table 7.1 would be required. It was apparent that the KU-FRL transducer package was unable to meet these needs; however, the data management portion of the package would be able to.

The airplane used in this program is shown in Figure 7.2. The external modifications to the airplane include a spin chute as well as right-hand and left-hand wing tip booms.

The spin chute was added for safety reasons. A device for deploying the chute is provided to the pilot, allowing him to retrieve the airplane from an unrecoverable spin. Also a release mechanism is provided to release the chute after deployment and spin recovery. The pilot also wears a parachute in the event the

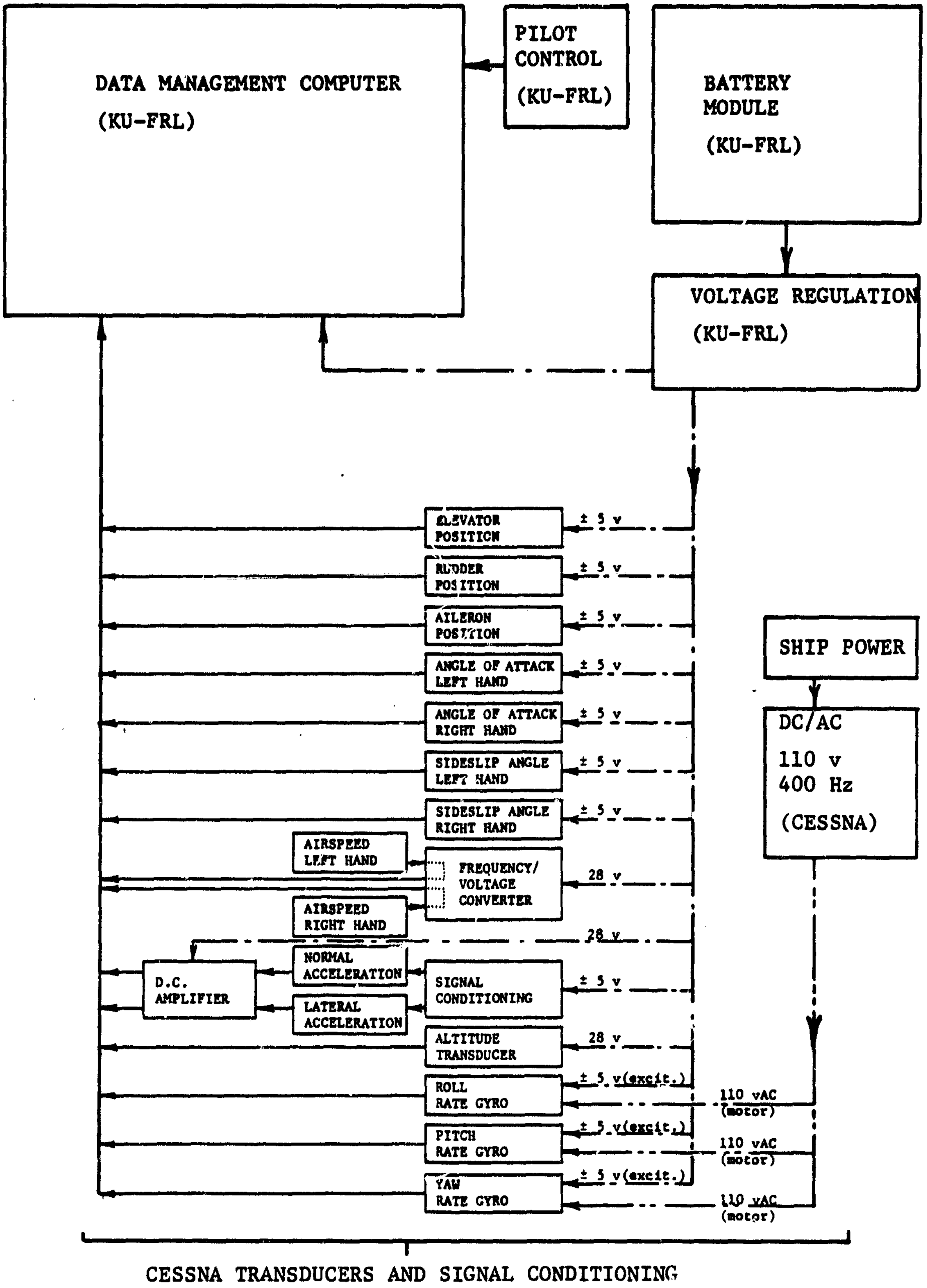


Figure 7.1 Block diagram of Cessna spin test installation

Table 7.1 Cessna Spin Test Measurement Requirements

SYMBOL	TRANSDUCER	RANGE
δ_e	ELEVATOR POSITION	FULL TRAVEL
δ_a	AILERON POSITION	FULL TRAVEL
δ_r	RUDDER POSITION	FULL TRAVEL
p	ROLL RATE	±360 °/sec
q	PITCH RATE	±360 °/sec
r	YAW RATE	±360 °/sec
A_z	NORMAL ACCELERATION	-3 to +5 g
A_y	LATERAL ACCELERATION	±3 g
α_L	ANGLE OF ATTACK LEFT HAND	-20 to +80 °
α_R	ANGLE OF ATTACK RIGHT HAND	-20 to +80 °
β_L	SIDESLIP ANGLE LEFT HAND	±45 °
β_R	SIDESLIP ANGLE RIGHT HAND	±45 °
$KTAS_L$	TRUE AIRSPEED LEFT HAND	20 to 180 knots
$KTAS_R$	TRUE AIRSPEED RIGHT HAND	20 to 180 knots
H_p	PRESSURE ALTITUDE	0 to 15000 ft

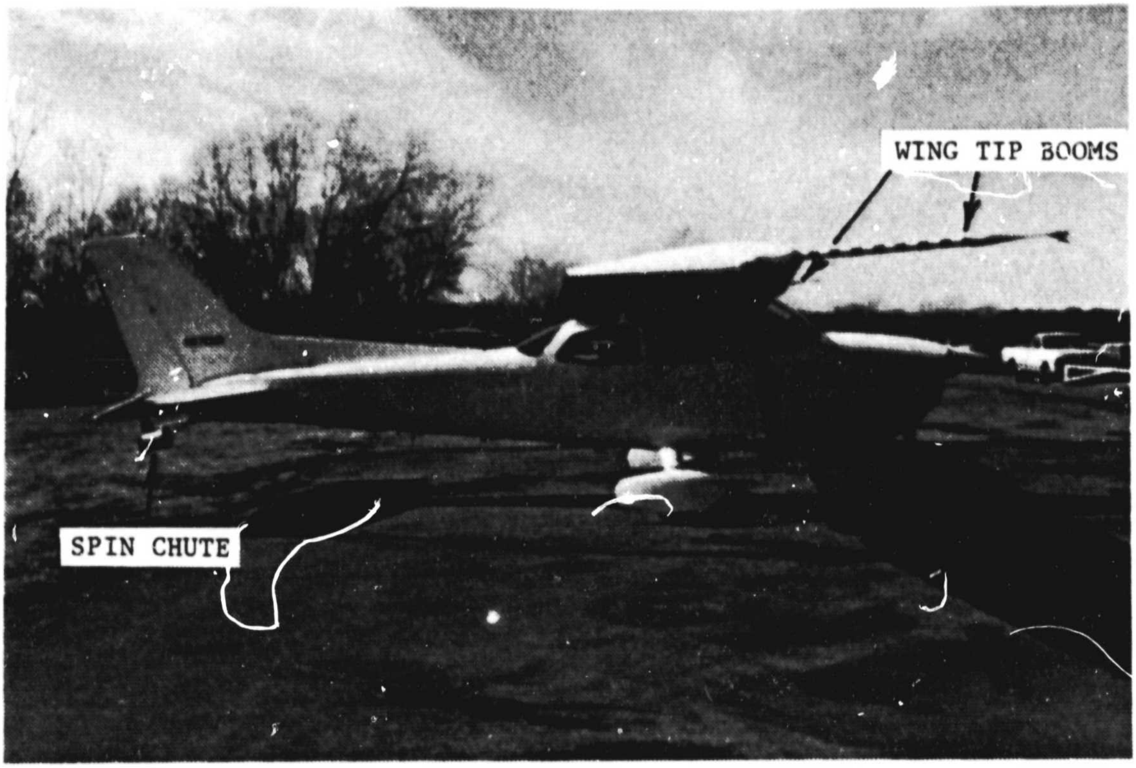


Figure 7.2 Cessna spin test airplane

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

spin chute does not deploy or will not release, permitting him to leave the airplane in safety.

The right-hand and left-hand wing tip booms are shown in more detail in Figures 7.3. The booms utilize a flow direction and airspeed sensor (described in detail in Reference 31). This sensor allows determination of the airspeed, angle of attack, and angle of sideslip (as shown in Figures 7.3). A probe is included on each wing tip to allow determining the true properties of the spin. The axis of a spin is generally not at the center of gravity of the airplane. (In the C172 it appears to be ahead of the center of gravity.) Providing both left-hand and right-hand measurements allows determining where this spin axis is by using the differences between the measurements from each side.

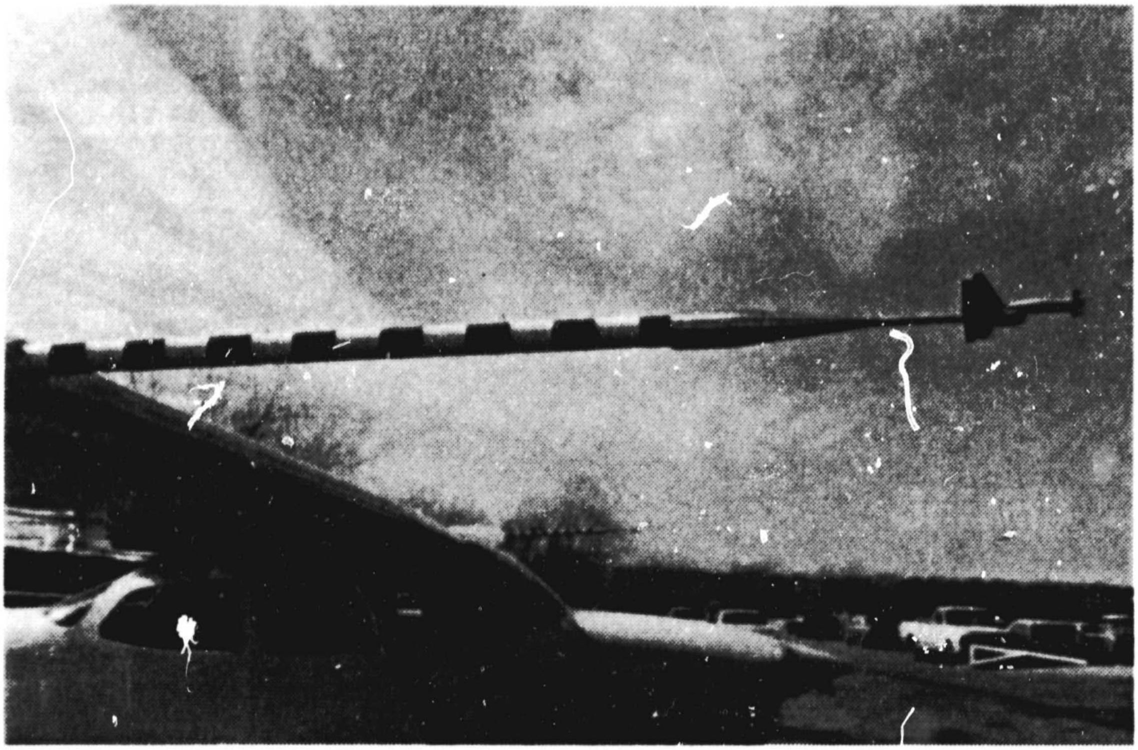


Figure 7.3(a) Cessna wing tip booms (supplied by NASA Langley)

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

Figure 7.3(b) Cessna wing tip booms (supplied by NASA Langley)

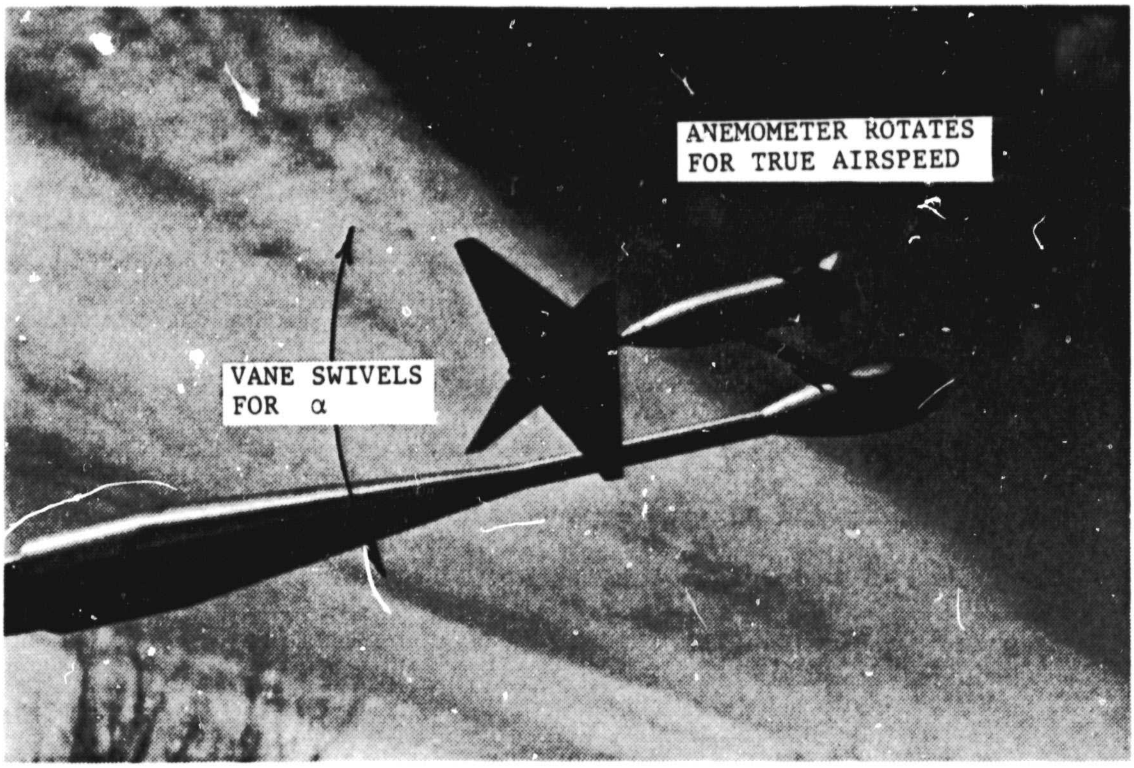


Figure 7.3(c) Cessna wing tip booms (supplied by NASA Langley)

Inside the airplane cockpit the Cessna inertial reference transducers (p, q, r, A_z, A_y) were mounted on the sensor pallet as shown in Figure 7.4. As can be seen, the KU-FRL power supply system was used in this installation. This was necessary to provide power for the computer and was also utilized to provide power for some of the transducers. Figure 7.1 shows the power sources used for the specific devices.

The KU-FRL data management computer is shown installed in the Cessna airplane in Figure 7.5.

A chase airplane was used in this flight test program. This was for safety purposes to provide an outside observer who could warn the pilot of the spin test airplane (over the communications radio) of any unexpected problems. Also, a video camera was carried onboard the chase airplane to record the spin visually.

7.2 Data Reduction

Data analysis for this spin program was done by Cessna on their Hewlett Packard 9825 microcomputer. Data was transferred from the KU-FRL system to the Cessna computer, using the standard RS232 ports on each machine (see Appendix A.9 for Hewlett Packard 9825 programs). After transfer, the data was plotted on Cessna's computer using the program in Appendix A.9. Figures 7.6 present the traces of several of the spins. It can be seen that the data recorded produce results capable of analysis.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

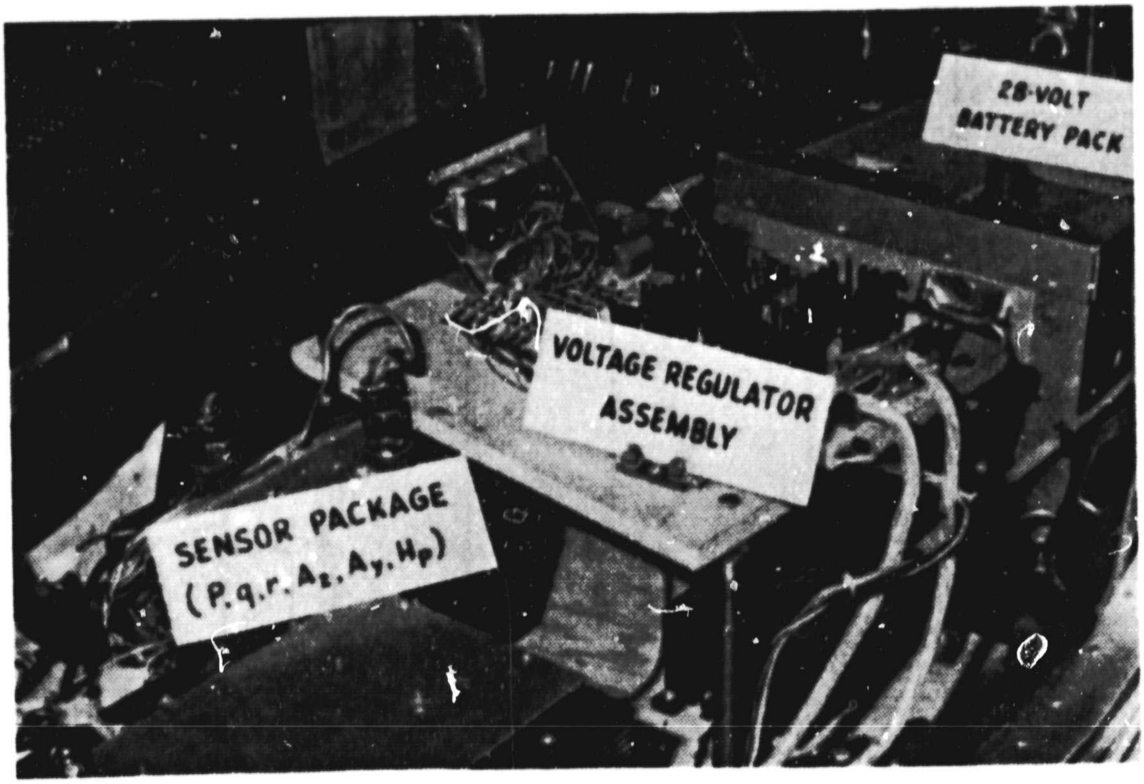


Figure 7.4 Cessna spin test instrumentation installation

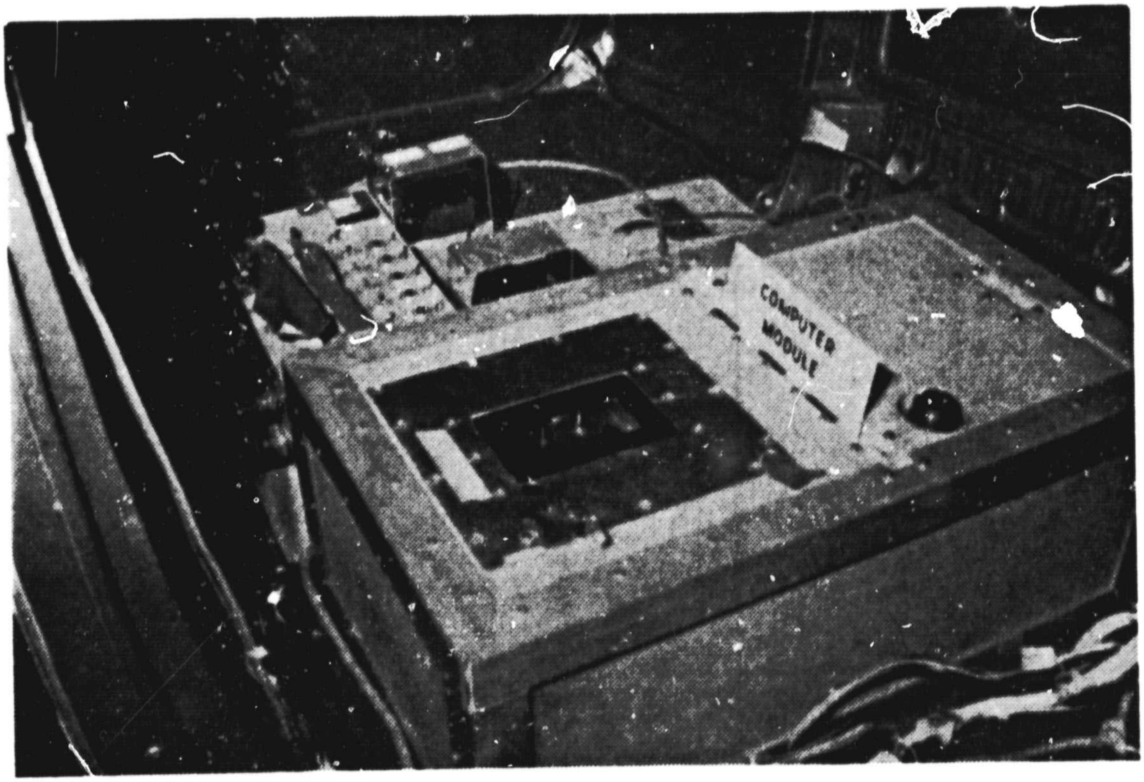


Figure 7.5 Cessna spin test computer installation

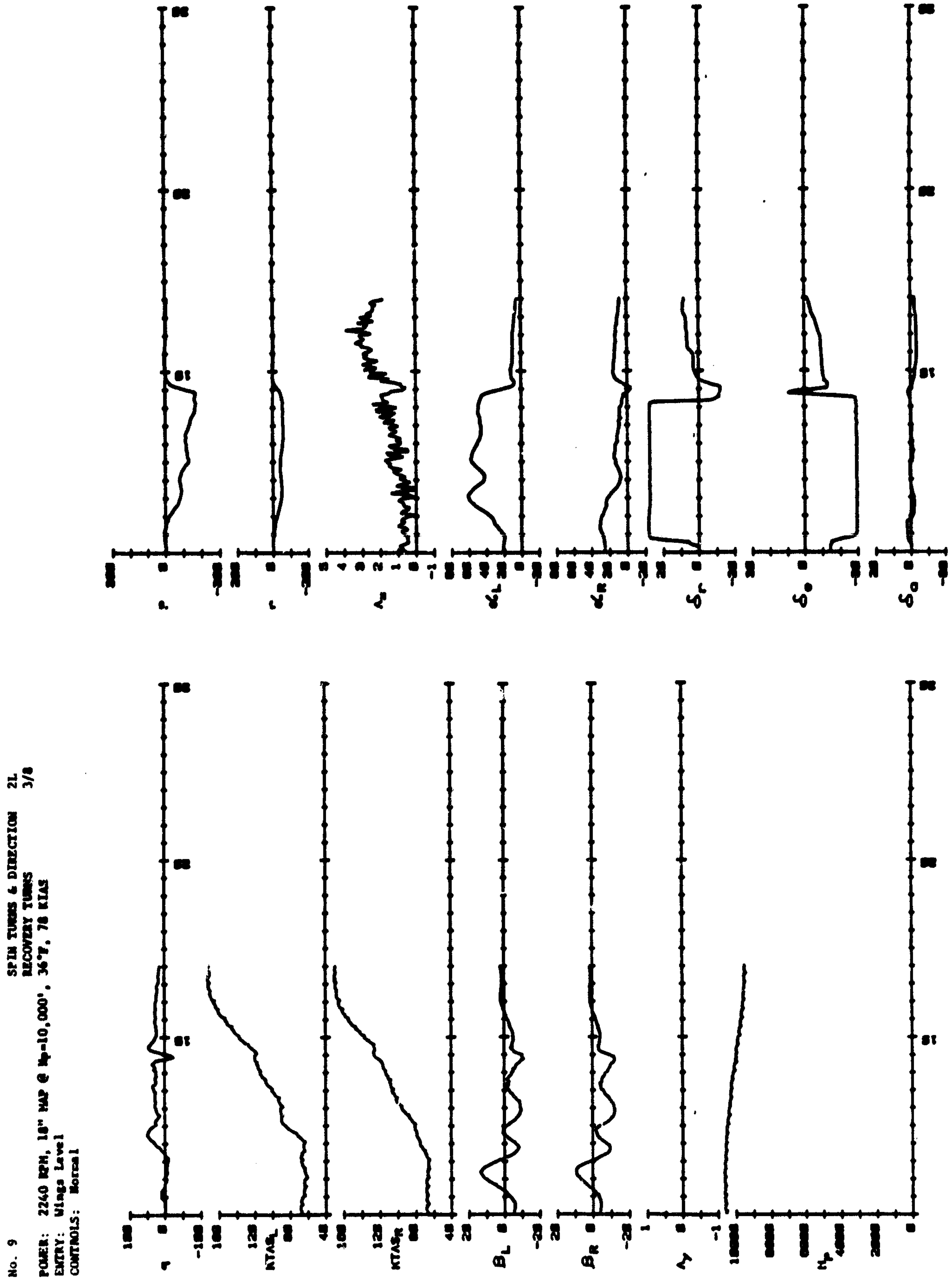


Figure 7.6(a) Cessna spin test, spin traces, spin No.9

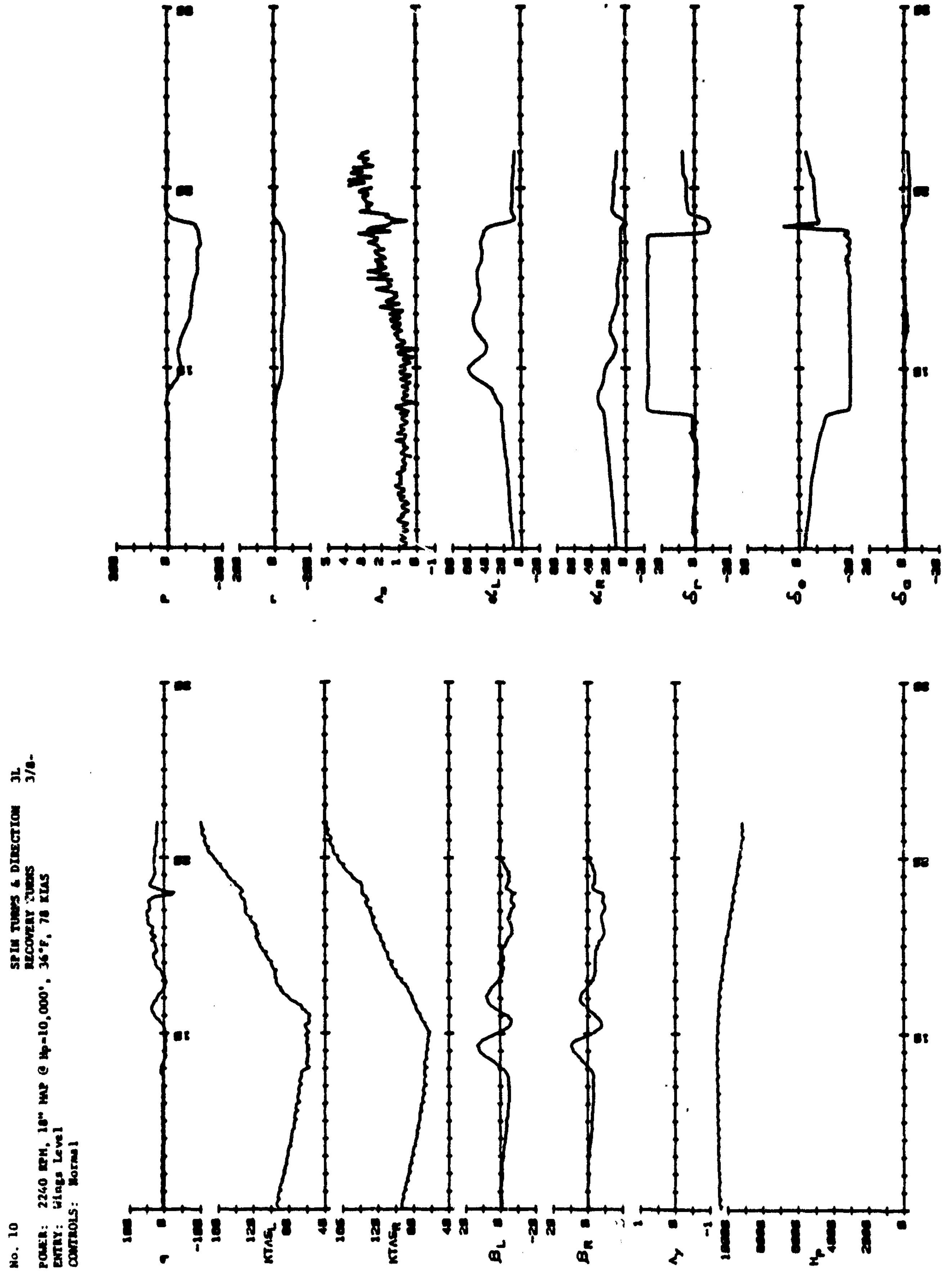


Figure 7.6(b) Cessna spin test, spin traces, spin No.10

7.3 Results of Spin Program

The results of the spin program, from the aspect of this report, show the adaptability of the KU-FRL-designed data management system. The versatility specifically designed into this portion of the system allows virtually any 16-channel instrumentation combination to provide the measurements. Also shown is the feasibility of using different data reduction computers by the use of the standard RS232 port for data transfer.

No real problems were encountered by Cessna personnel in using the data management system, even though none of them had any extensive microcomputer experience.

8. CONCLUSIONS

The flight test system designed and evaluated under this program has met the objectives outlined in Chapter 2. The system

- is easy to install,
- is virtually self contained,
- is simple in operation,
- requires no complex flight maneuvers,
- is applicable to general aviation airplanes,
- is capable of longitudinal and lateral stability analysis, and
- is low in cost.

This system has shown that the technology used is capable of the tasks to be performed.

In the data reduction method all the derivatives contained in Equation [5.5] can be determined. The method also allows determining any combination of these derivatives. It must be noted that these are the state vector dimensional derivatives which can be converted to the normally accepted stability derivatives (as per Reference 22) using Tables 5.2 and 5.3.

Areas have been discovered where further work is required. A comprehensive list is included in Chapter 9.

9. RECOMMENDATIONS FOR PHASE III

Four areas have been suggested throughout this report for improvement of the KU-FRL instrumentation system. These are summarized here.

9.1 Equipment

- Equipment is required for accurate transducer calibration. A pendulum arrangement as per Reference 32 is suggested as an excellent means of calibrating the transducers.

- Size reduction of equipment is suggested. To allow easier placement in aircraft, the size of the system could be reduced significantly, especially if the number of packages is increased to form more efficient space utilization.

9.2 Calibration

- All transducers should be calibrated as a system. Using the actual data acquisition package for transducer calibration is suggested as a means to reduce calibration errors. This should be done in conjunction with the calibration pendulum of 9.1 above.

9.3 Data Reduction

- Refinements are required to the current MMLE "BONES" program to simplify its use and add to the versatility.

- Further study is suggested to allow Performance analysis (i.e., Drag Polar) of the test airplane. Methods similar to those of References 4-11, 13, and 18 seem to provide promising solutions.

- Some of the features of the latest version of MMLE (Reference 33) should be added. Specifically, the Cramer-Rao bounds addition, and the correction for center of gravity offsets can be added directly into the MMLE program.

- The addition of the acceleration transformation matrix ([R] of Equation [5.4]) to the MMLE program should be explored.

- Determine the validity of the prediction of α and β by comparing with measured values.

9.4 Effect of Control Surface Float

- The effect of control surface cable tension should be evaluated to determine the influence this has on the parameters predicted.

9.5 Proof-of-Test Capability

- Tests are recommended in other general aviation airplanes to demonstrate the system's adaptability. Recommended are tests on a high performance, single-engine retractable, and on a light-to-medium, twin-engine airplane.

- The tests suggested above would also aid in providing further insight into the possible "Uniqueness" problem. This should be a definite area of research, to validate the MMLE (or similar) concepts.

10. REFERENCES

1. Renz, R. R. L.; Clarke, R.; Roskam, J.; "A Literature Survey of Performance and Stability Flight Testing"; KU-FRL-407-1; University of Kansas Flight Research Laboratory; Lawrence, Kansas; 1979.

2. Renz, R. R. L.; Clarke, R.; Hill, J. R.; Mosser, M.; Roskam, J.; Rummer, D. I.; "Progress Report on Phase I: Development of a Simple, Self-Contained Flight Test Data Acquisition System"; KU-FRL-407-3; University of Kansas Flight Research Laboratory; Lawrence, Kansas; 1980.

3. Renz, R. R. L.; "Calibration of MDAS-16 Analog-to-Digital Converter"; KU-FRL-407-5; University of Kansas Flight Research Laboratory; Lawrence, Kansas; 1981.

4. Gerlach, O. H.; "High Accuracy Instrumentation Techniques for Non-Steady Flight Measurements" (in Perry, M. A., ed.), Flight Test Instrumentation, Vol. 3, Pergamon Press; 1964.

5. Gerlach, O. H.; "Determination of Performance, Stability and Control Characteristics from Measurements in Non-Steady Maneuvers"; Stability and Control, AGARD Conf. Proc. No. 17, pgs 497-523; 1966.

6. Gerlach, O. H.; "Measurements of Performance, Stability and Control Characteristics in Non-Steady Flight with a High Accuracy Instrumentation System" (in Perry, M. A., ed.); Aerospace Instrumentation, Vol. 4, Pergamon Press; 1966.

7. Gerlach, O. H.; "The Application of Regression Analysis to the Evaluation of Instrument Calibration"; Flight Test Instrumentation, AGARD Conf. Proc. No. 32; 1967.

8. Gerlach, O. H.; "Determination of Performance and Stability Parameters from Non-Steady Flight Test Maneuvers"; SAE Paper #700236; 1970.

9. Gerlach, O. H.; "The Determination of Stability Derivatives and Performance Characteristics from Dynamic Maneuvers"; Report VTH-163; Delft University of Technology; Delft, The Netherlands; 1971.

10. Hosman, R. J. A. W.; "A Method to Derive Angle of Pitch, Flight Path Angle and Angle of Attack from Measurements in Non-Steady Flight"; Report VTH-156, Delft University of Technology; Delft, The Netherlands; 1971.

11. Hosman, R. J. A. W.; "Advanced Flight Test Instrumentation: Design and Calibration"; Memorandum M-222; Delft University of Technology; Delft, The Netherlands; October 1974.

12. Iliff, K. W.; Maine, R. E.; "Practical Aspects of Using a Maximum Likelihood Estimation Method to Extract Stability and Control Derivatives from Flight Data"; NASA TN D-8209; 1976.

13. Iliff, K. W.; "Estimation of Aerodynamic Characteristics from Dynamic Flight Test Data"; Dynamic Stability Parameters, AGARD Conf. Proc. #235; 1978.

14. Iliff, K. W.; Maine, R. E.; Montgomery, T. D.; "Important Factors in the Maximum Likelihood Analysis of Flight Test Maneuvers"; NASA T. P. 1459; 1979.

15. Iliff, K. W.; Maine, R. E.; "NASA BONES"; NASA Dryden; unpublished computer program.

16. Maine, R. E.; Iliff, K. W.; "A Fortran Program for Determining Stability and Control Derivatives from Flight Data"; NASA TN D-7831; 1975.

17. Klein, V.; Gregory, R.; "A Proposal for Self-Contained Instrumentation System for Flight Research on Stability and Control"; Report CIT-FI-72-013; Cranfield Institute of Technology; March 1972.

18. Smetana, F. O.; Fox, S. R.; "Flight Test Evaluation of Predicted Light Aircraft Drag, Performance, and Stability"; NASA CR-158076; 1979.

19. Sorensen, J. A.; Tyler, J. S.; Powell, D. J.; "Evaluation of Flight Instrumentation for the Identification of Stability and Control Derivatives"; AIAA Paper 72-963; September 1972.

20. Eckholdt, D. C.; Wells, W. R.; "A Survey of AFFDL Parameter Estimation Efforts and Future Plans"; Parameter Estimation Techniques and Applications in Aircraft Flight Testing"; NASA TN D-7647; 1973.

21. Mace, W. D.; Pool, A. (editors); "Flight Test Instrumentation Series"; AGARD No. 160, Vols. 1-8; 1973 to 1977.

22. Roskam, J.; "Airplane Flight Dynamics and Automatic Flight Controls": Roskam Aviation and Engineering Corporation; Ottawa, Kansas;(Rte 4,Box 274);1979

23. Renz, R. R. L.; "Flight Test Instrumentation Certification Report"; KU-FRL-407-2; University of Kansas Flight Research Laboratory; Lawrence, Kansas; 1980.

24. Wuest, W.; "Measurements of Flow Speed and Flow Direction by Aerodynamic Probes and Vanes"; AGARD Conf. Proc. 32; 1967.

25. Ikhtiari, Paul A.; Marth, V. G.; "Trailing Cone Static Pressure Measurement Device"; Journal of Aircraft, Vol. 1, No. 2, pgs. 93-94; March 1969.

26. Lewis, J. R.; "TRS-80 Performance, Evaluation by Program Timing"; Byte Magazine; pgs. 84-94; McGraw-Hill; New York; March 1980.

27. Renz, R. R. L.; Mosser, M. A.; "Digital Tape Qualifying Procedure for KU-FRL Instrumentation Package"; KU-FRL-407-5; University of Kansas Flight Research Laboratory; Lawrence, Kansas; 1981.

28. Gainer, T. G.; Hoffman, S.; "Summary of Transformation Equations and Equations of Motion Used in Free-Flight and Wind-Tunnel Data Reduction and Analysis"; NASA SP-3070; 1972.

29. Gupta, N. K.; Hall, W. E.; "Input Design for Identification of Aircraft Stability and Control Derivatives"; NASA CR-2493; 1975.

30. Suit, W. T.; Cannaday, R. L.; "Comparison of Stability and Control Parameters for a Light, Single Engine, High-Winged Aircraft Using Different Flight Test and Parameter Estimation Techniques"; Nasa Tech Memorandum 80163; 1979.

31. Kershner, D. D.; "Miniature Flow-Direction and Airspeed Sensor for Airplanes and Radio-Controlled Models in Spin Studies"; NASA TP-1467; 1979.

32. McLaren, I.; "Calibration Methods for the Accurate Assessment of the Static and Dynamic Performance of Some Flight Test Instrument"; Her Majesty's Stationery Office; London, England; 1964.

33. Maine, R. E.; Iliff, K. W.; "User's Manual for MMLE-3, A General Fortran Program for Maximum Likelihood Parameter Estimation"; NASA TP 1563; 1980.

10.1 Instrumentation System Reports

KU-FRL Number	Title	Date
407-1	A Literature Survey of Performance and Stability Flight Testing	1979
407-2	Flight Test Instrumentation Certification Report	1980
407-3	Progress Report on Phase I: Development of a Simple, Self-Contained Flight Test Data Acquisition System	1980
407-4	Calibration of MDAS-16 Analog-to-Digital Converter	1981
407-5	Digital Tape Qualifying Procedure for KU-FRL Instrumentation Package	1981
407-6	Progress Report on Phase II: Development of a Simple, Self-Contained Flight Test Data Acquisition System	1981
407-P1	Development of a Simple, Self-Contained Flight Test Data Acquisition System. (Paper presented at Society of Flight Test Engineers, Atlanta, Georgia.)	1980
407-P2	A Microcomputer Based Data Acquisition System for Use in Flight Testing of General Aviation Airplanes. (Paper presented at IEEE Mid America Electronics Conference, Kansas City.)	1980
407-P3	Development of a Simple, Self-Contained Flight Test Data Acquisition System. (Paper presented at SAE Business Aircraft Meeting, Wichita, Kansas.)	1981

APPENDIX A

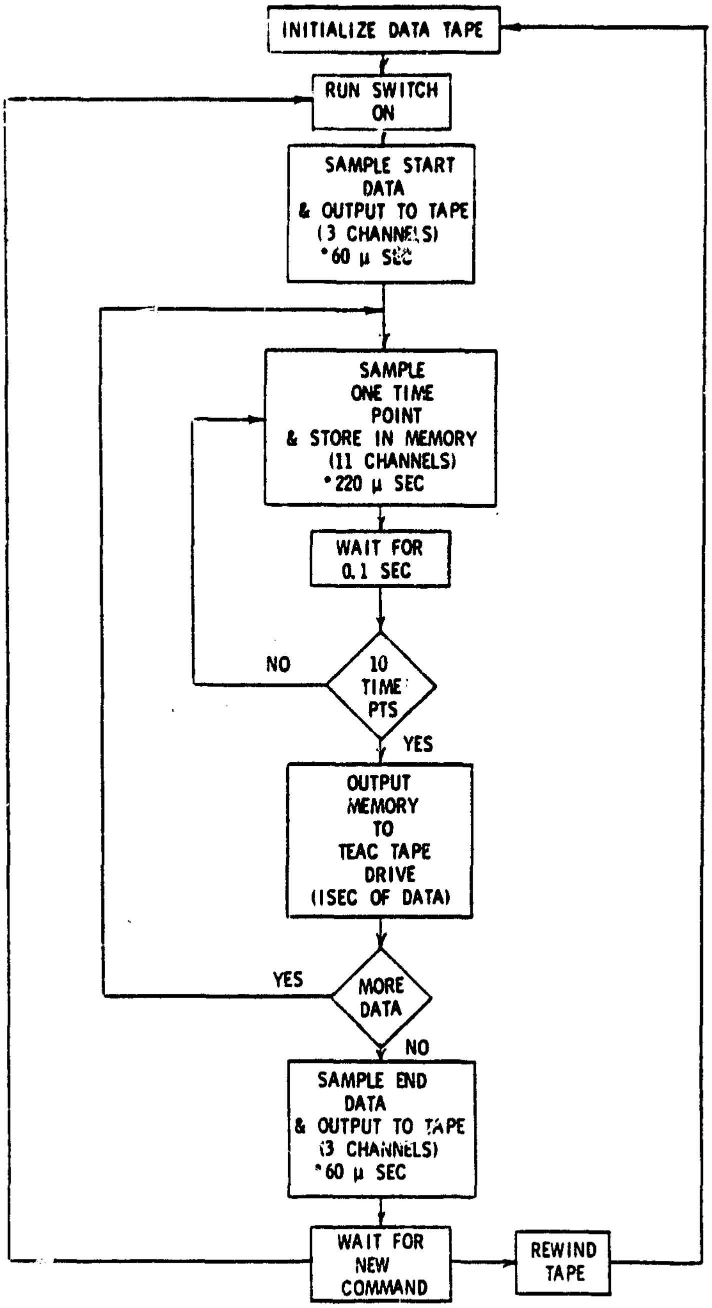
PROGRAMS

This appendix includes descriptions, flow charts, and listings of the computer programs required by this flight test system.

A.1) DATA ACQUISITION PROGRAM

Description: This program, which runs on the AIM 65, collects and saves the measured state time histories. The information is collected and stored on the cassette tape in one-second real-time blocks. The data for each channel is coded as two binary eight-bit words totalling sixteen bits. The first word holds the eight most significant bits. The second word holds the four most significant bits and the four least significant bits. This gives a redundancy check of the highest order bits.

Flowchart:



* 20 µ sec between successive channels

PROGRAM LISTING

```
;DATA ACQUISITION
;
RNCNT=0
BLKCNT=2
BUFCNT=4
IBUF=5
OBUF=7
CNT=9
BUF1=$200
BUF2=$300
KDDRA2=$A481
KDDRB2=$A483
KDRA2=$A480
KDRB2=$A482
DBR=$9008
WDC=$9009
CDR=$900A
MDRO=$900B
CSR=$900C
ESR=$900D
ISR=$900E
MDR1=$900F
WRT=$C1
WTM=$C2
ERA=$C3
SLE=$C9
REW=$CA
NRDY=$10
FPT=$04
DA=$20
DBRE=$40
CCE=$80
UDRB=$A000
UACR=$A00B
UIER=$A00E
UT1L=$A004
UT1CH=$A005
UT2L=$A008
UT2H=$A009
UIFR=$A00D
BIT5=$20
LOADK=$EF
RECK=$BF
CLOSEK=$DF
TIME1H=$27
TIME1L=$10
--------------------
*=$0400
LDA #$92
STA MDRO
LDA #1
STA RNCNT
LDA #0
STA RNCNT+1
LDA #$FF
STA KDDRA2
LDA #0
STA KDDRB2
STA KDRA2
LDA #$C0
STA UACR
START
LDA #$12
STA MDRO
LDA #REW
JSR COMD
LDA #REW
JSR COMD
MAIN
JSR GKEY
CMP #LOADK
BNE MAIN
LDA #REW
JSR COMD
LDA #SLE
JSR COMD
MAIN2
JSR GKEY
CMP #RECK
BEQ RECORD
CMP #CLOSEK
BNE MAIN2
CLOSE
LDA #WTM
JSR COMD
LDA #WTM
JSR COMD
LDX #12
CLOSE1
LDA #ERA
JSR COMD
DEX
BNE CLOSE1
JMP START
RECORD
LDA #0
STA IBUF
STA OBUF
LDA #>BUF1
STA IBUF+1
LDA #>BUF2
STA OBUF+1
JSR ENDREC
LDA #>INT
STA $A405
LDA #<INT
STA $A404
LDA #$C0
STA UIER
LDA #<$C34E
STA UT1L
LDA #>C34E
STA UT1CH
CLI
LDA #0
STA BLKCNT
STA BLKCNT+1
REC1
JSR SWAP
JSR WRITE
REC2
JSR GKEY
CMP #RECK
BNE RECX
LDA BUFCNT
CMP #220
BNE REC2
INC BLKCNT
BNE REC1
INC BLKCNT+1
JMP REC1
RECX
LDA BUFCNT
CMP #220
BNE RECX
LDA #$40
STA UIER
JSR SWAP
INC BLKCNT
BNE RECX1
INC BLKCNT+1
RECX1
JSR WRITE
JSR ENDREC
JSR SWAP
LDA #$FF
STA BLKCNT
STA BLKCNT+1
JSR WRITE
INC RNCNT
BNE RECX2
INC RNCNT+1
RECX2
JMP MAIN2
```

PROGRAM LISTING (continued)

```
--------------------
GKEY
LDA KDRB2
PHA
LDA #TIME1L
STA UT2L
LDA #TIME1H
STA UT2H
GKEY1
LDA UIFR
AND #BIT5
BEQ GKEY1
PLA
CMP KDRB2
BNE GKEY
RTS
--------------------
COMD
PHA
LDA ESR
COMD1
LDA CSR
AND #NRDY
BNE COMD1
LDA CSR
AND #FPT
BNE COMD1
PLA
STA CDR
COMD2
LDA ISR
AND #CCE
BEQ COMD2
RTS
--------------------
ENDREC
LDY #0
ENDR1
LDX #10
LDA $8000
JSR WAIT
ENDR2
LDA $9001
JSR WAIT
DEX
BNE ENDR2
LDX #5
ENDR3
LDA $8001
JSR WAIT
LDA $8002
STA (IBUF),Y
INY
LDA $8003
STA (IBUF),Y
INY
DEX
BNE ENDR3
CPY #220
BNE ENDR1
RTS
```

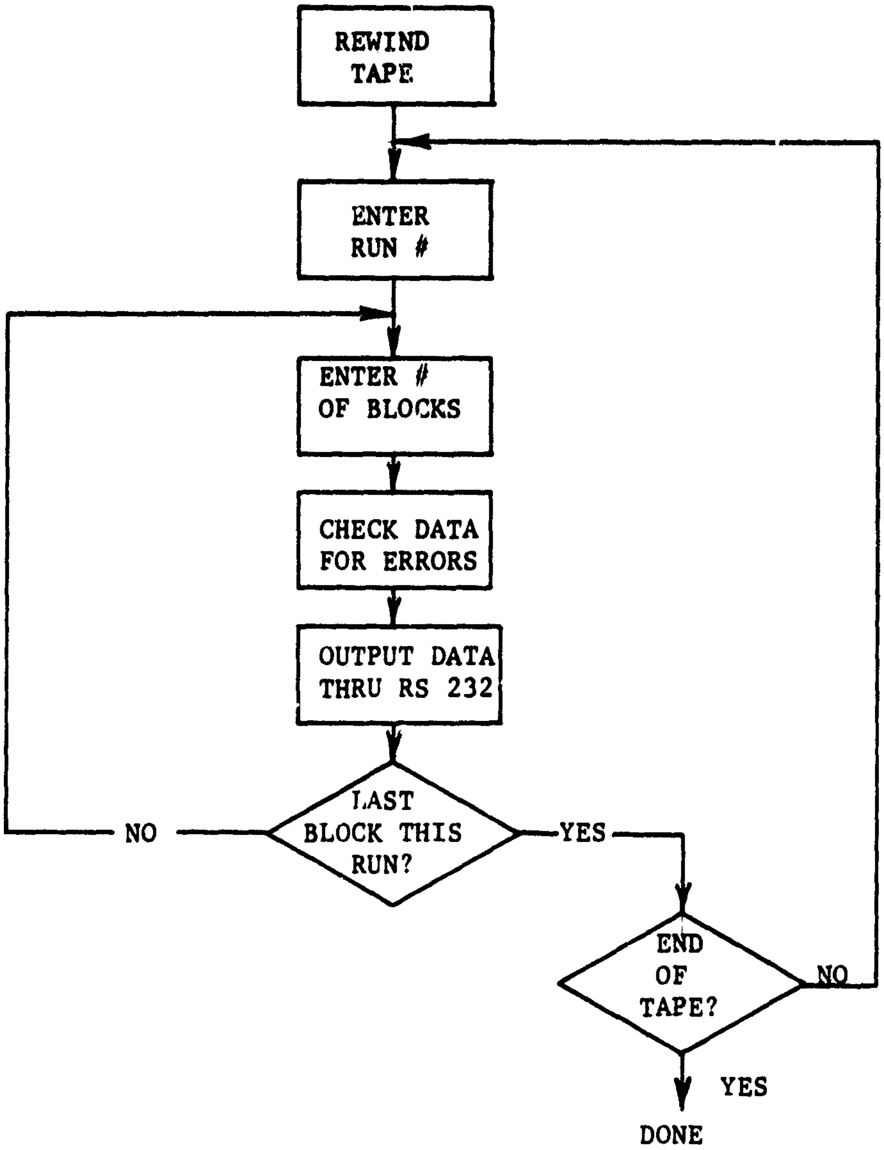
```
--------------------
WAIT
JSR WAITX
WAITX
RTS
--------------------
WRITE
LDA ESR
LDA #224
STA WDC
LDA #WRT
STA CDR
LDA RNCNT
JSR WWORD
LDA RNCNT+1
JSR WWORD
LDA BLKCNT
JSR WWORD
LDA BLKCNT+1
JSR WWORD
LDY #0
WRITE1
LDA (OBUF),Y
JSR WWORD
INY
CPY #220
BNE WRITE1
WRITE2
JMP COMD2
--------------------
SWAP
LDA OBUF+1
PHA
LDA IBUF+1
STA OBUF+1
PLA
STA IBUF+1
LDA #0
STA BUFCNT
RTS
--------------------
WWORD
PHA
WWORD1
LDA ISR
AND #DBRE
BEQ WWORD1
PLA
STA DBR
RTS
```

```
--------------------
INT
PHA
LDA UDRB
BPL INTEX
TYA
PHA
LDY BUFCNT
LDA #11
STA CNT
LDA $8000
JSR WAIT
ILOOP
LDA $8002
STA (IBUF),Y
LDA $8001
INY
LDA $8003
STA (IBUF),Y
INY
DEC CNT
BNE ILOOP
STY BUFCNT
PLA
TAY
INTEX
LDA UT1L
PLA
RTI
 END
```

A.2) DATA TRANSFER

Description: This program allows the AIM 65 to read the information stored in the DATA ACQUISITION program. The information is passed to the MINC 11/03 computer using the RS 232 port.

Flowchart:



PROGRAM LISTING

```
                    ;DATA RECOVERY
RNCNT=0
BLKCNT=2
VRUN=4
VBLK=5
CNT=6
BUF1=$200
DBR=$9008
WDC=$9009
CDR=$900A
MDRO=$900B
CSR=$900C
ESR=$900D
ISR=$900E
MDR1=$900F
SLP=$C8
RDL=$C4
REW=$CA
NRDY=$10
TDRE=$02
CR=$0D
SCR=$9006
SDR=$9007
LOADC=$4C
READC=$52
CLOSEC=$43
INALL=$E993
NUMA=$EA46
READM=$E93C
OUTALL=$E9BC
OUTPUT=$E97A
UTIL=$A004
UTICH=$A005
UACR=$A00B
-------------
*=$300
CCE
.BYTE $80
DA
.BYTE $20
MO
.BYTE CR,'TAPE ERROR',$A0
MRUN
.BYTE CR,'WHICH RUN NUMBER',$BF
MBLK
.BYTE CR,'HOW MANY BLOCKS',$BF
MEND
.BYTE CR,'LAST BLO[illegible] HIS RU',$CE
MINV
.BYTE CR,'INVALID COMMAN',$C4
MERR1
.BYTE CR,'FILE MARK FOUN',$C4
MRNCNT
.BYTE CR,'RUN NUMBER',$A0
MERROR
.BYTE CR,'DATA ERROR',$BF
TEMPO
.BYTE $00
--------------------
*=$400
RESETB
LDA #$92
STA MDRO
LDA #$C0
STA UACR
LDA #$68       ;$68=300BAUD
               ;$34=600
               ;$1A=1200
               ;$0D=2400
```

```
STA UT1L
LDA #0

STA UT1CH
LDA #$11
EOR #$FF
STA SCR
              ;
MAIN
JSR GCOM
CMP #LOADC
BEQ MAIN2
JSR INVAL
JMP MAIN
MAIN2
LDA #$12
STA MDRO
LDA #REW
JSR COMDA
LDA #SLP
JSR COMDA
MAIN3
JSR GCOM
CMP #READC
BEQ READ
CMP CLOSEC
BEQ CLOSE
JSR INVAL
JMP MAIN3


              ;
CLOSE
LDA #REW
JSR COMDA
LDA #REW
JSR COMDA
JMP MAIN
              ;
READ
LDY #MRUN-MO
JSR MESS
JSR GCNT
CMP #0
BEQ CLOSE
STA VRUN
READ1
JSR RBLK
BCS CLOSE
LDA BLKCNT
ORA BLKCNT+1
BNE READ3
LDY #MRNCNT-MO
JSR MESS

LDA RNCNT
JSR NUMA
READ3
LDA RNCNT
CMP VRUN
BCC READ1
BEQ READ2
LDA #REW
JSR COMDA
LDA #SLP
JSR COMDA
JMP READ1
READ2
LDY #MBLK-MO
JSR MESS
```

```
JSR GCNT
CMP #0
BEQ CLOSE
STA VBLK
                   ;CHANGE TO NOPS
                   ;TO TRANSMIT COUNTS
SENDB
JMP SENDB1
LDA RNCNT
JSR SEND
LDA RNCNT+1
JSR SEND
LDA BLKCNT
JSR SEND

LDA BLKCNT+1
JSR SEND
SENDB1
LDX #0
CNVT
LDA BUF1,X         ;COMPARE HIGH BITS
AND #$F0
STA TEMPO
INX
LDA BUF1,X
AND #$F0
CMP TEMPO
BEQ CNVT1
JSR FIX            ;IF BAD FIX HERE
CNVT1
DEX
LDA BUF1,X         ;CONVERT TO PRINTABLE
SEC                ;CHARACTERS
ROR BUF1,X
LSR BUF1,X
AND #$3
CLC
ROL A
ROL A
ROL A
ROL A
STA TEMPO
INX
LDA BUF1,X
AND #$0F
ORA TEMPO
ORA #$40
STA BUF1,X
INX
CPX #220
BNE CNVT
LDY #0
SENDB2
LDA BUF1,Y
JSR SEND
INY
CPY #220
BNE SENDB2
LDA BLKCNT
CMP #$FF
BNE SENDB3

CMP BLKCNT+1
BEQ END
SENDB3
DEC VBLK
BNE SENDB5
SENDB4
LDY #MBLK-MO
```

```
JSR MESS
JSR GCNT
CMP #0
BEQ CLOSE1
STA VBLK
SENDB5
JSR RBLK
BCS CLOSE1
JMP SENDB
CLOSE1
JMP CLOSE
END
LDY #MEND-MO
JSR MESS
JMP MAIN3
--------------
MESS
LDA MO,Y
PHA
AND #$7F
JSR OUTPUT
INY
PLA
BPL MESS
RTS
--------------
GCOM
JSR READM
JSR OUTALL
RTS
--------------
INVAL
LDY #MINV-MO
JSR MESS
RTS
--------------
GCNT
LDA #0
STA CNT
GCNT1
JSR INALL
JSR DPACK
BCC GCNT1
LDA CNT
RTS
--------------
DPACK
CMP #'0'
BCC RSPAC
CMP #$3A
BCS RSPAC
AND #$0F
PHA
LDA CNT
ASL A
ASL A
CLC

ADC CNT
ASL A
STA CNT
PLA
CLC
ADC CNT
STA CNT
CLC
RTS
RSPAC
SEC
RTS
```

```
--------------
SEND
PHA
SEND1
LDA SCR
AND #TDRE
BNE SEND1
PLA
EOR #$FF        ;CHG FOR KNOWN CHAR
STA SDR
RTS
-------------
RBLK
LDA CSR
AND #NRDY
BNE RBLK
LDA #224
STA WDC
LDA #RDL
STA CDR
JSR RWORD
BCS RBLK2
STA RNCNT
JSR RWORD
BCS RBLK2
STA RNCNT+1
JSR RWORD
BCS RBLK2
STA BLKCNT
JSR RWORD
BCS RBLK2
STA BLKCNT+1
LDY #0
RBLK1
JSR RWORD
BCS RBLK2
STA BUF1,Y
INY
CPY #220
BNE RBLK1
JMP COMDA2.
RBLK2
JMP COMDA4
--------------
COMDA
PHA
LDA ESR
COMDA1
LDA CSR
AND #NRDY
BNE COMDA1
PLA
STA CDR
COMDA2
LDA ISR
AND CCE
BEQ COMDA2
COMDA4
LDA CSR
PHA
AND #2
BEQ COMDA5
LDY #MERR1-MO
JSR MESS
PLA
SEC
RTS
COMDA5
PLA
```

```
AND #$81
BNE COMDA3
CLC
RTS
COMDA3
LDY #MO-MO
PHA
JSR MESS
PLA
JSR NUMA
LDA ESR
JSR NUMA

CLC
RTS
--------------
RWORD
LDA ISR
BIT CCE
BNE RWORD2
BIT DA
BEQ RWORD
LDA DBR

CLC
RTS
RWORD2
SEC
RTS
-------------------
FIX
LDY #MERROR-MO
JSR MESS
RTS
-------------------
                          ;INITIAL SET UP
*=$700
LDA #$C0
STA UACR
LDA #$60
STA UTIL
LDA #$00
STA UTICH
JMP RESETB
 END
```

A.3) DATA RECEIVE

Description: This program accepts raw data from the AIM 65. This raw, formatted data is collected in files to be used in the ENGINEERING CONVERSION routine.

Program listing

```
      PROGRAM AIMIN
C
C  AIM TO MINC PROGRAM
C   WRITTEN BY MARK A MOSSER
C
C   This program inputs data from the AIM - 65  through SLU-1
C    as characters (22 at a time ) to fill a 600 X 22 character
C    array . When full this array is outputed to the user
C    specified file.
C
      DIMENSION IADDR (4),IDATA(600,22),ICHAR(22)
C
      TYPE *,'THIS PROGRAM READS 22 CHARACTER WORDS FROM SLU1   AT A'
      TYPE *,'TIME AND PLACES THESE WORDS ON A FILE'
      TYPE *,' '
      TYPE 7
7     FORMAT (' WHAT IS THE NAME OF YOUR OUTPUT FILE ?'/)
      CALL ASSIGN (1,,-1,NEW)
      TYPE *,'1ST I WILL ATTACH SLU 1'
      IERR=MTATCH(2)
      TYPE 998,IERR
998   FORMAT ('IERR = ',I2)
      TYPE *,'NEXT I WILL SET IT UP FOR READING'
      IADDR(1) = "50010
      IADDR(2) = 0
      IADDR(3) = 0
      IADDR(4) = 0
      IERR = MTSET(2,IADDR(1))
      TYPE 998,IERR
C
      TYPE *,'NOW YOU HAVE 2 CHOICES . STOP OR READ IN DATA'
10    TYPE *,'1 = READ IN DATA '
      TYPE *,'2 = STOP'
      TYPE *,'$WHICH ?'
      ACCEPT 5,IFCT
5     FORMAT (I1)
      IF (IFCT .GT. 2) GOTO 10
      GOTO (100,200),IFCT
C
100   TYPE *,'                         READ FUNCTION'
      DO 110 I=1,600
      DO 120 J=1,22
      IERR = MTIN(2,ICHAR(J),1)
      IDATA(I,J)=ICHAR(J)
120   CONTINUE
      TYPE *,'COUNT = ',I
110   CONTINUE
      TYPE *,'WAIT FOR DISK OUTPUT'
      DO 135 K=1,600
      WRITE (1,169) (IDATA(K,L),L=1,22)
169   FORMAT (22A1)
135   CONTINUE
      GOTO 10
C
C  STOP
C
200   TYPE *,'STOPPING'
      TYPE *,'DETATCH INPUT PORT'
      IERR = MTDTCH(2)
      TYPE 998,IERR
      STOP
      END
```

PRECEDING PAGE BLANK NOT FILMED

A.4) ENGINEERING CONVERSION

Description: This process presently uses two programs which run seperately. The first routine, AIMCNV, converts the raw coded AIM 65 data to voltages. The AIMCNV program makes use of a macro assembly language routine (CONVRT) which performs the bit manipulations necessary to turn the AIM 65 coded data into a form useable by the MINC Fortran programs.The second routine ,EGRCNV, converts the voltages into engineering units.

Program listing:

```
C
C  AIM TO VOLT CONVERSION PROGRAM
C  IT MUST BE LINKED TO CONVRT TO WORK
C
           PROGRAM AIMCNV
           DIMENSION IN(22),VOLT(11),IVOLT(11)
           TYPE *,' AIM FORMAT TO VOLT CONVERSION '
           TYPE *,' INSERT A DATA DISKETTE'
5          TYPE *,' What is the full name of the INPUT file?'
           TYPE *
           CALL ASSIGN(1,,-1,RDO)
10         TYPE *,' What is the full name of the OUTPUT file?'
           TYPE *
           CALL ASSIGN (2,,-1,NEW)
C
20         TYPE *,' How many SECONDS (10 time pts) do you want?'
           ACCEPT *,NUM
C
C   convert this mess
C
           NUM=NUM*10
           DO 100 I=1,NUM
           READ (1,109,END=300) (IN(J),J=1,22)
109        FORMAT(22A1)
           DO 150 K=1,11
           KH=(K*2)-1
           KL=K*2
           CALL CONVRT(IN(KH),IN(KL),IVOLT(K))
           VOLT(K)=IVOLT(K)/16
150        VOLT(K)=(VOLT(K)*.002435632861)+.06330029324
           WRITE (2,119) (VOLT(J),J=1,11)
119        FORMAT (11F12.9)
           TYPE *,' COUNT=',I
100        CONTINUE
C
C   Here is the series of quostions
C
           TYPE *,' More in this file?'
           ACCEPT 209,NORY
209        FORMAT(A1)
           IF (NORY .EQ. 'Y') GOTO 20
           TYPE *,' Another OUTPUT file from this INPUT file?'
           ACCEPT 209,NORY
           IF (NORY .EQ. 'Y') GOTO 210
           TYPE *,' Another INPUT file in this OUTPUT file?'
           ACCEPT 209,NORY
           IF (NORY .EQ. 'Y') GOTO 310
           TYPE *,' Are you done?'
           ACCEPT 209,NORY
           IF (NORY .EQ. 'N') GOTO 5
           GOTO 500
210        CALL CLOSE(2)
           GOTO 10
300        TYPE *,' END OF INPUT FILE ERROR'
```

```
310     CALL CLOSE(1)
        TYPE *,' What is the next INPUT file name?'
        TYPE *
        CALL ASSIGN(1,,-1,RDO)
        GOTO 20
500     TYPE *,' GOODBYE'
        STOP
        END
```

```
        .TITLE  CONVRT
        .SBTTL  CONVERSION ROUTINE
        .GLOBL  CONVRT
CONVRT: CLC
        CLR     R1
        BIC     #177700,@2(R5)  ;PEEL OFF 1ST 2 BITS
        BIC     #177700,@4(R5)  ;OF THE ARGS
        MOVB    @2(R5),R1       ;PUT HIGH 6 BITS INTO R1
        ASH     #6,R1           ;SHIFT LEFT 6
        ADD     @4(R5),R1       ;ADD IN LOW 6 BITS
        MOV     R1,@6(R5)       ;PLACE IN RESULT ARG
        ROL     @6(R5)
        ROL     @6(R5)          ;ROTATE LEFT 4
        ROL     @6(R5)
        ROL     @6(R5)
TST:    CMP     @6(R5),#100000  ;SEE IF POSITIVE OR NEGATIVE
        BMI     NEGA
POS:    COM     @6(R5)          ;POSITIVE NUMBERS ARE 1'S COMPLIMENTED
        ADD     @6(R5),#1       ;THEN ADD 1
        RTS     PC              ;DONE
NEGA:   NEG     @6(R5)          ;NEGATIVE #'S ARE 2'S COMPLIMENTED
        RTS     PC
        .END
```

```
            PROGRAM EGRCNV
      C
      C  Volt to Engineering Units conversion program
      C  By Mark A. Mosser  Dec. 1980
      C
            COMMON VOLTS(10,11),ENG(10,11)
      10    TYPE *,' Volts to Engineering Units conversion'
            TYPE *,' INSERT DATA DISK '
            TYPE *,' What is the full name of your INPUT file?'
            TYPE *
            CALL ASSIGN(1,,-1,RDO)
      30    TYPE *,' What is the full name of your OUTPUT file?'
            TYPE *
            CALL ASSIGN(2,,-1,NEW)
      C
      C  input first block (T,Pd,Ps) ,convert & output
      C
            READ(1,109) ((VOLTS(J,K),K=1,11),J=1,10)
      109   FORMAT(11F12.9)
            CALL TOTPSD(T,PS,PD)
      119   FORMAT(3E12.4)
            READ (1,109)((VOLTS(J,K),K=1,11),J=1,10)
            CALL VTOEGR
      100   READ(1,109,END=300,ERR=1000)((VOLTS(J,K),K=1,11),J=1,10)
            WRITE (2,129)((ENG(J,K),K=1,11),J=1,10)
      129   FORMAT(11E12.4)
            CALL VTOEGR
            GOTO 100
      C
      C   LOOP EXITED BY EOF IN READ
      C
      300   WRITE (2,119) T,PS,PD
            CALL TOTPSD(T,PS,PD)
            WRITE (2,119) T,PS,PD
            CALL CLOSE(1)
            CALL CLOSE(2)
```

```
        TYPE *,' ANOTHER FILE? '
        ACCEPT 777,NORY
777     FORMAT(A1)
        IF( NORY .EQ. 'Y') GOTO 10
        GOTO 500
C
C ERROR
C
1000    TYPE *,' ERROR IN READ'
500     STOP
        END
```

```
C
C    Subroutine TOTPSD
C        This subroutine converts the data to T (temperature)
C      PS (static pressure) & PD (dynamic pressure)
C
        SUBROUTINE TOTPSD(T,PS,PD)
        COMMON VOLTS(10,11),ENG(10,11)
        II=2
        JJ=1
        T=0
        PS=0
        PD=0
C
C    Loop to fill T,PS,PD
C
11      T=T+VOLTS(II,JJ)
        JJ=JJ+1
        IF (JJ .LE. 11) GOTO 21
        JJ=JJ-11
        II=II+1
21      PD=PD+VOLTS(II,JJ)
        JJ=JJ+1
        IF (JJ .LE. 11) GOTO 31
        JJ=JJ-11
        II=II+1
31      PS=PS+VOLTS(II,JJ)
        JJ=JJ+3
        IF (JJ .LE. 11) GOTO 11
        IF (II .EQ. 10) GOTO 101
        JJ=JJ-11
        II=II+1
        GOTO 11
C
C    Average T,PD,PS
C
101     T=T/20
        PD=PD/20
        PS=PS/20
C
C    Convert to engineering units
C
        T=((((280*T)-487)-273.16)*(9/5))+32
        IF(PD .LE. 4.526) GOTO 206
        PD=6658.67-(1436.61*PD)
        GOTO 216
206     PD=2648.58-(550.6*PD)
216     IF (PS .LE. 2.303) GOTO 226
        PS=(12919.896*PS)-19754.521
        GOTO 236
226     PS=(9451.796*PS)-11767.49
236     CONTINUE
        RETURN
        END
```

```
C
C     Subroutine VTOEGR
C       This subroutine converts the data from volts to engineering
C     units. It then reorders them as follows:
C     theta,q,Az,Ax,deltaE,phi,p,Ay,r,deltaA,deltaR
C
        SUBROUTINE VTOEGR
        COMMON VOLTS(10,11),ENG(10,11)
12      DO 102 IA=1,10
        ENG(IA,1)=(VOLTS(IA,9)*.2111)+.00738
        ENG(IA,2)=-((VOLTS(IA,6)+.001)/.09988)/57.3
        ENG(IA,3)=((VOLTS(IA,5)+.002)/1.001)-1.0
        ENG(IA,4)=(VOLTS(IA,3)-.002)/2.499
        ENG(IA,5)=(VOLTS(IA,10)*6.24+2.6)/57.3
        ENG(IA,6)=(VOLTS(IA,2)*.3117)+.01342
        ENG(IA,7)=-((VOLTS(IA,8)+.0035)/.09933)/57.3
        ENG(IA,8)=VOLTS(IA,4)/10.027
        ENG(IA,9)=-((VOLTS(IA,7)-.011)/.10134)/57.3
        ENG(IA,10)=(-(VOLTS(IA,11)*4.73091)+.55548)/57.3
        ENG(IA,11)=((VOLTS(IA,1)*7.52845)+8.20487)/57.3
102     CONTINUE
        RETURN
        END
```

A.5) QUICK LOOK PLOTS

Description: The QUICK LOOK PLOT program is used as an aid in choosing appropriate flight data for further analysis. The routine collects the engineering units data and plots it on a graphics CRT terminal.

Program listing:

```
C  * * * * * * * * * * * * * * * * * * * * *
C  *                                       *
C  *  MAIN PROGRAM FOR PLOTTING MMLE DATA   *
C  *                                       *
C  *  USE FOLLOWING SUBROUTINES TO LINK :  *
C  *  INIT,PLOT55,FOUR,GRID,GRAPH.         *
C  * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/STATUS/ISTAT(16)
      DIMENSION IARRAY(512),HH(11,1),DATA(240,11)
      COMMON /FIVE/ K1,K2,K3,K4,N
      COMMON/FIVEA/ DATA
      BYTE YES,NO,ANS,NAME(15)
C
      DATA YES,NO /'Y','N'/
      DATA ISTAT/16*0/
1     CONTINUE
C     . . . . . . . . . . . . . . . . . . . . .
      CALL INIT
C
      TYPE 5
5     FORMAT(' TYPE IN NAME OF FILE WITH MEASURED DATA'/)
25    FORMAT(14A1)
      ACCEPT 25, (NAME(I),I=1,14)
      OPEN(UNIT=2,NAME=NAME,TYPE='OLD',ACCESS='SEQUENTIAL',
     1     READONLY,FORM='UNFORMATTED')
      DO 26 I=1,14
      NAME(I) =' '
26    CONTINUE
      TYPE 6
6     FORMAT(' TYPE IN NUMBER OF TIME POINTS'/)
      READ(5,7) N
7     FORMAT(I3)
C
C     . . . . . . . . . . . . . . . . . . . . .
C     READ IN DATA FROM FILE
      DO 21 I=1,N
      READ(2) (HH(K,1),K=1,11)
99    CONTINUE
      DO 21 J=1,11
      DATA(I,J)=HH(J,1)
21    CONTINUE
C
C     . . . . . . . .
C
223   CONTINUE
      CALL INIT
22    FORMAT(11E12.4)
24    FORMAT(A1)
C
C     . . . . . . . . . . . . . . . . . . . . .
C
      CALL GRID(500,45)
C

101   CONTINUE
      CALL FOUR(1)
      CALL FOUR(2)
      CALL FOUR(3)
C
C     . . . . . . . .
C
```

```
      TYPE 51
 51   FORMAT(' DO YOU WANT TO TAKE ANOTHER LOOK AT THE DATA?(Y/N)')
      READ(5,24) ANS
      IF(ANS.EQ.YES) GO TO 223
      TYPE 52
 52   FORMAT(' DO YOU WANT TO LOOK AT ANOTHER DATA FILE?(Y/N)')
      READ(5,24) ANS
      CALL CLOSE(2)
      IF(ANS.EQ.YES) GO TO 1
C
C     . . . . . . . . . . . . . . . .
C
      CALL PLOT55(2,512,1+2+4+32+64,ISTAT)
      CALL PLOT55(0,-1,0,ISTAT)
      RETURN
      END
```

```
      SUBROUTINE INIT
      COMMON/STATUS/ISTAT(16)
      DATA ISTAT/16*0/
      CALL PLOT55(13,72,,ISTAT)
      CALL PLOT55(13,74,,ISTAT)
      CALL PLOT55(2,1+512,,ISTAT)
      RETURN
      END
```

```
      SUBROUTINE FOUR(NZ)
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * *
C     * THIS SUBROUTINE WILL DISPLAY FOUR VARIABLES ON  *
C     * THE CRT EVERY TIME IT IS CALLED FROM THE MAIN   *
C     * PROGRAM.                                        *
C     * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/STATUS/ISTAT(16)
      INTEGER IARRAY(512)
      DIMENSION DATA(240,11)
      COMMON /FIVE/ K1,K2,K3,K4,N
      INTEGER GAIN(11)
      COMMON/FIVEA/ DATA
      DATA GAIN/150,90,39,157,225,75,90,157,90,225,225/
C     . . . . . . . . . . . . . . . . . . . . .
C
      IF(NZ.EQ.2.) GO TO 52
      IF(NZ.EQ.3.) GO TO 53
C
      K1 = 3
      K2 = 2
      K3 = 1
      K4 = 5
      GO TO 54
 52   CONTINUE
      K1 = 9
      K2 = 11
      K3 = 4
      K4 = 5
      GO TO 54
 53   CONTINUE
      K1 = 8
      K2 = 7
      K3 = 6
      K4 = 10
 54   CONTINUE
C
      DO 41 K=1,N
      IARRAY(2*K) = DATA(K,K1)*.57295*GAIN(K1)+45
      IARRAY(2*K-1) =DATA(K,K2)*.57295*GAIN(K2)+90
 41   CONTINUE
      CALL GRAPH(2*N,IARRAY)
```

```
C       . . . . . . . . . . . . . . . . . . . . .
        DO 42 K=1,N
        IARRAY(2*K) = DATA(K,K3)*.57295*GAIN(K3)+135
        IARRAY(2*K-1) = DATA(K,K4)*.57295*GAIN(K4)+180
42      CONTINUE
        CALL GRAPH(2*N,IARRAY)
C
C       . . . . . . . . . . . . . . . . . . . . . . .
C
        CALL PLOT55(9,25,1,ISTAT)
        CALL PLOT55(12,,'@** QUICK LOOK DATA PLOTS ***',ISTAT)
C
        IF(NZ.EQ.1.) GO TO 31
        IF(NZ.EQ.2.) GO TO 32
        IF(NZ.EQ.3.) GO TO 33
31      CONTINUE
        CALL PLOT55(9,50,4,ISTAT)
        CALL PLOT55(12,,'ELEVATOR POSN. 20 DEG.',ISTAT)
        CALL PLOT55(9,50,8,ISTAT)
        CALL PLOT55(12,,'PITCH ATTITUDE, 30 DEG.',ISTAT)
        CALL PLOT55(9,50,13,ISTAT)
        CALL PLOT55(12,,'PITCH RATE, 50 DEG/SEC.',ISTAT)
        CALL PLOT55(9,50,17,ISTAT)
        CALL PLOT55(12,,'NORMAL ACCEL., 2 G.',ISTAT)
C
        GO TO 34
32      CONTINUE
        CALL PLOT55(9,50,4,ISTAT)
        CALL PLOT55(12,,'ELEVATOR POSN., 20 DEG.',ISTAT)
        CALL PLOT55(9,50,8,ISTAT)
        CALL PLOT55(12,,'LONGITUDINAL ACCEL., .5 G',ISTAT)
        CALL PLOT55(9,50,13,ISTAT)
        CALL PLOT55(12,,'RUDDER POSN., 20 DEG.     ',ISTAT)
        CALL PLOT55(9,50,17,ISTAT)
        CALL PLOT55(12,,'YAW RATE, 50 DEG/SEC.',ISTAT)
        GO TO 34
33      CONTINUE
        CALL PLOT55(9,50,4,ISTAT)
        CALL PLOT55(12,,'AILERON POSN., 20 DEG.',ISTAT)
        CALL PLOT55(9,50,8,ISTAT)
        CALL PLOT55(12,,'BANK ANGLE, 60 DEG.         ',ISTAT)
        CALL PLOT55(9,50,13,ISTAT)
        CALL PLOT55(12,,'ROLL RATE, 50 DEG/SEC.',ISTAT)
        CALL PLOT55(9,50,17,ISTAT)
        CALL PLOT55(12,,'LATERAL ACCEL., .5 G.',ISTAT)
34      CONTINUE
        CALL PLOT55(9,0,20,ISTAT)
        DO 1 J=1,5000
        DAMMY= COS(45)
1       CONTINUE
        RETURN
        END
```

```
        SUBROUTINE GRID(IDX,IDY)
        COMMON/STATUS/ISTAT(16)
        CALL PLOT55(2,1+512,,ISTAT)
        CALL PLOT55(9,0,0,ISTAT)
        CALL PLOT55(10,,,ISTAT)
        CALL PLOT55(2,1+32+64,,ISTAT)
        DO 3 I=1,512,IDX
3       CALL PLOT55(5,I-1,1,ISTAT)
        DO 4 I=1,236,IDY
4       CALL PLOT55(4,1,I-1,ISTAT)
        RETURN
        END
```

```
        SUBROUTINE GRAPH(N,IARRAY)
        COMMON/STATUS/ISTAT(16)
        DIMENSION IARRAY(512)
        NUMBER=ISTAT(8)/8
        CALL PLOT55(7,0,0,ISTAT)
        CALL PLOT55(8,512,0,ISTAT)
        CALL PLOT55(2,1+(NUMBER+1)*2,(NUMBER+1)*10,ISTAT)
        CALL PLOT55(3,-N,IARRAY,ISTAT)
        CALL PLOT55(1,1-NUMBER,,ISTAT)
        CALL PLOT55(9,10,1,ISTAT)
        END
```

PRECEDING PAGE BLANK NOT FILMED

A.6) DETAILED ENGINEERING CONVERSION

Description: The CRINST program performs the detailed corrections for instrument offsets from the body axes. Biases on the accelerometers are also removed in the corrections.

Program listing:

```
      PROGRAM CRINST
C.... PROGRAM TO MODIFY THE RAW ENGINEERING
C.... DATA FOR INSTRUMENT CORRECTIONS
C.... THAI  MEASURED FROM BODY TO INST AXES
C.... XBAR, YBAR, AND ZBAR FROM BODY TO INST AXES
      BYTE NAME(15)
      DIMENSION FI(11),FIM1(11),FIP1(11),DATA(600,11)
      DATA DGR,G /57.29578,32.174/
      DATA THAI /-5.696/
      DATA XBAR,YBAR,ZBAR /+0.052,+1.179,+1.630/
C.... TRANSDUCER POSTIONS RECALCULATED ON 6-FEB-81
    4 FORMAT(14A)
    5 FORMAT(I10,8F12.4)
C.... SET LAST BYTE OF CHARACTER STRING TO NULL
      NAME(15)=0
C.... ENTER THE FILE NAME FOR THE DATA TO BE CORRECTED
   10 FORMAT(' ENTER THE FILE NAME FOR THE RAW ENGINEERING DATA',/)
      TYPE 10
      ACCEPT 4,(NAME(I),I=1,14)
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',ACCESS='SEQUENTIAL',
     .    READONLY,FORM='FORMATTED',RECORDSIZE=132)
C.... ENTER THE FILE NAME FOR THE DATA TO BE SAVED ON
C.... CLEAR OUT OLD FILENAME
      DO 22 I=1,14
      NAME(I)=' '
   22 CONTINUE
   20 FORMAT(' ENTER THE FILE NAME TO HOLD THE CONVERTED DATA',/)
      TYPE 20
      ACCEPT 4,(NAME(I),I=1,14)
      OPEN(UNIT=2,NAME=NAME,TYPE='NEW',ACCESS='SEQUENTIAL',
     .     FORM='UNFORMATTED',
     .     BUFFERCOUNT=2)
C.... READ DATA FROM HP DATA FILE
   28 FORMAT(' ENTER THE NUMBER OF TIME POINTS TO BE CORRECTED',/)
      TYPE 28
   29 FORMAT(I10)
      ACCEPT 29,IEND
   30 FORMAT(11E12.4)
      DO 31 I=1,IEND
      READ(1,30)(DATA(I,J),J=1,11)
   31 CONTINUE
      CLOSE(UNIT=1)
C.... TRANSFER FIRST TWO DATA POINTS
      DO 35 I=1,11
      FIM1(I)=DATA(1,I)
   35 CONTINUE
C.... CORRECT FOR SIGN ERRORS IN CALIBRATIONS
      FIM1( 4)=-FIM1( 4)
      FIM1( 6)=-FIM1( 6)
      FIM1( 8)=-FIM1( 8)
      FIM1( 9)=-FIM1( 9)
      FIM1(11)=-FIM1(11)
C.... CORRECT FOR GYRO MISALINEMENT
      COSTHI=COS(THAI/DGR)
      SINTHI=SIN(THAI/DGR)
      FIM1(1)=FIM1(1)-THAI/DGR
      PI= FIM1(7)*COSTHI+FIM1(9)*SINTHI
      RI=-FIM1(7)*SINTHI+FIM1(9)*COSTHI
      FIM1(7)=PI
      FIM1(9)=RI
C.... WRITE THESE VALUES TO THE OUTPUT FILE
      WRITE(2)(FIM1(I),I=1,11)
```

```
C.... TRANSFER NEXT DATA POINT
      DO 36 I=1,11
      FI(I) =DATA(2,I)
   36 CONTINUE
C.... CORRECT FOR SIGN ERRORS IN CALIBRATIONS
      FI( 4)  =-FI( 4)
      FI( 6)  =-FI( 6)
      FI( 8)  =-FI( 8)
      FI( 9)  =-FI( 9)
      FI(11)  =-FI(11)
C.... ADD CORRECTION FOR GYRO MISALINEMENT
      FI(1)=FI(1)-THAI/DGR
      PI= FI(7)*COSTHI+FI(9)*SINTHI
      RI=-FI(7)*SINTHI+FI(9)*COSTHI
      FI(7)=PI
      FI(9)=RI
C.... PRINT OUT AVERAGE VALUES FOR THI1 AND PHI1
      THA1=((FIM1(1)+FI(1))/2.)*DGR
      PHI1=((FIM1(6)+FI(6))/2.)*DGR
   40 FORMAT(' THA1  (IN DEG) = ',F12.6,' PHI1  (IN DEG) = ',F12.6)
      TYPE 40,THA1,PHI1
      KOUNT=2
      COSTHA=COS(THA1/DGR)
      SINTHA=SIN(THA1/DGR)
      COSPHI=COS(PHI1/DGR)
      SINPHI=SIN(PHI1/DGR)
C.... START CORRECTION LOOP
   50 CONTINUE
      KOUNT=KOUNT+1
C.... IF KOUNT IS GREATER THAN IEND GO TO 1000
      IF(KOUNT.GT.IEND)GO TO 1000
C.... TRANFER NEXT TIME POINT
      DO 37 I=1,11
      FIP1(I)=DATA(KOUNT,I)
   37 CONTINUE
C.... CORRECT FOR SIGN ERRORS IN CALIBRATION
      FIP1( 4)=-FIP1( 4)
      FIP1( 6)=-FIP1( 6)
      FIP1( 8)=-FIP1( 8)
      FIP1( 9)=-FIP1( 9)
      FIP1(11)=-FIP1(11)
C.... CORRECT FOR MISALINEMENT ANGLE
      FIP1(1)=FIP1(1)-THAI/DGR
      PI= FIP1(7)*COSTHI+FIP1(9)*SINTHI
      RI=-FIP1(7)*SINTHI+FIP1(9)*COSTHI
      FIP1(7)=PI
      FIP1(9)=RI
C.... GET AN BACK TO ORIGINAL SIGNAL
      FI(3) =FI(3)+1.00
C.... COMPUTE PDOT, QDOT, AND RDOT
      PDOT  =(FIP1(7)-FIM1(7))/0.2
      QDOT  =(FIP1(2)-FIM1(2))/0.2
      RDOT  =(FIP1(9)-FIM1(9))/0.2
C.... CORRECT ACCELERATIONS FOR OFFSET
      P     =FI(7)
      Q     =FI(2)
      R     =FI(9)
      AXP1  =FI(4)*COSTHI
      AXP2  =FI(3)*SINTHI
      AXP3  =(R*R+Q*Q)*XBAR/G
      AXP4  =(P*Q-RDOT)*YBAR/G
      AXP5  =(P*R+QDOT)*ZBAR/G
      AXFIX =AXP1-AXP2+AXP3-AXP4-AXP5
      AX    =AXFIX
      AY    =FI(8)       -(P*Q+RDOT)*XBAR/G+(P*P+R*R)*YBAR/G
     .       -(Q*R-PDOT)*ZBAR/G
      AZ    =-FI(4)*SINTHI-FI(3)*COSTHI-(P*R-QDOT)*XBAR/G
     .       -(Q*R+PDOT)*YBAR/G+(P*P+Q*Q)*ZBAR/G
      FI(3) =-AZ
      FI(4) = AX
      FI(8) = AY
```

```
C.... CORRECT FOR ACCELEROMETER BIAS
      FI(3) =FI(3)-COSTHA*COSPHI
      FI(4) =FI(4)-SINTHA
      FI(8) =FI(8)+COSTHA*SINPHI
C.... WRITE VALUES ON OUTPUT
C.... TYPE OUT KOUNTER
      TYPE 5,KOUNT
      WRITE(2)(FI(I),I=1,11)
C.... BUCKET BRIGADE VALUES THRU TIME
      DO 100 I=1,11
      FIM1(I)=FI(I)
      FI(I)  =FIP1(I)
  100 CONTINUE
      GO TO 50
 1000 CONTINUE
C.... TRANSFER LAST DATA POINT
      WRITE(2)(FI(I),I=1,11)
C.... CLOSE DATA FILE
      STOP
      END
```

A.7) MMLE BONES ROUTINES

This appendix describes the MMLE programs. The first program required is the one that sets up the input matrices, as well as defining for the MMLE program which parameters it is to estimate. The MMLE programs, as well as their output format is also presented.

A.7.1) MMLE SETUP

Description: The setup program is an interactive program which sets up the input data for the MMLE BONES routine. Non-dimensional derivatives, geometric, and inertia data are input and used to form the initial estimate to the MMLE program.

Program listing:

```
      PROGRAM SETUP
C.... THIS PROGRAM SETS UP THE DATA USED IN BONES MMLE.
C.... DIMENSIONAL DERIVATIVES ARE BUILT UP FROM NON-DIMENSIONAL
C.... INPUT DATA AND AIRPLANE GEOMETRIC DATA.
C.... DEFAULT VALUES  (IF THEY EXIST)  ARE SHOWN AFTER EACH QUESTION.
      DIMENSION A(5,4),B(5,4),AA(5,4),BB(5,4),AP(8,4),BP(8,3)
      DIMENSION ZERO(4),BIAS(4),D1(7,7)
      DOUBLE PRECISION CASE,TEMP
      BYTE BANNER(4,80)
      DATA VALUE,IVALUE,AA,BB,AP,BP/0.,0,20*0.,20*0.,32*0.,24*0./
      DATA D1,BIAS,ZERO/49*0.,4*0.,4*0./
      DATA CASE,TEMP/'        ','        '/
      DATA BANNER/320*' '/
C.... SET DEFAULT VALUES
      NN     =200
      ITR    =10
      MZ     =7
      MAPR   =0
      HH     =0.10
      EPS    =0.0
      TIME   =0.0
      ALPHA  =0.0
      XLA    =1.0
C.... UNIT 1 WILL BE THE FILE NUMBER OF THE FILE FOR
C.... THE DATA DISK WHICH IS ASSUMED ON DY1:
C.... OPEN UNIT 1
    2 FORMAT(80A1)
    9 FORMAT(' ENTER A BANNER OF UP TO FOUR LINES.')
      TYPE 9
    1 FORMAT(' ENTER LINE: ',I1)
      DO 3 I=1,4
      TYPE 1,I
      ACCEPT 2,(BANNER(I,J),J=1,80)
    3 CONTINUE
   10 CONTINUE
   30 FORMAT(' ENTER 'LONG' OR 'LATR' FOR THE TYPE OF CASE',/,
     .       ' TO BE SET UP.')
      TYPE 30
   40 FORMAT(1A8)
      ACCEPT 40,CASE
      IF(CASE.EQ.'LONG')OPEN (UNIT=1,NAME='DY1:MMLELO.DAT',TYPE='NEW',
     .     RECORDSIZE=96,INITIALSIZE=50,DISPOSE='SAVE')
      IF(CASE.EQ.'LATR')OPEN (UNIT=1,NAME='DY1:MMLELD.DAT',TYPE='NEW',
     .     RECORDSIZE=96,INITIALSIZE=50,DISPOSE='SAVE')
      IF((CASE.NE.'LONG').AND.(CASE.NE.'LATR'))GO TO 10
```

```
C.... ERROR TRAP IF RESPONCE IS NOT 'LONG' OR 'LATR'
C.... BASIC DATA FOR EITHER LONGITUDINAL OR LATERAL-DIRECTIONAL CASE
   50 FORMAT(' ENTER THE NUMBER OF DATA POINTS TO BE PROCESSED.',/,
     .       ' (DEFAULT IS 200)')
      TYPE 50
   60 FORMAT(2F10.0)
   61 FORMAT(1I5)
      ACCEPT 61,IVALUE
      IF(IVALUE.GT.0)NN=IVALUE
      IVALUE=0
   70 FORMAT(' ENTER THE NUMBER OF ITERATIONS TO BE PERFORMED.',/,
     .       ' (DEFAULT IS 10)')
      TYPE 70
      ACCEPT 61,IVALUE
      IF(IVALUE.GT.0)ITR=IVALUE
      IVALUE=0
   80 FORMAT(' ENTER THE NUMBER OF OBSERVATIONS.',/,
     .       ' (DEFAULT IS 7)')
      TYPE 80
      ACCEPT 61,IVALUE
      IF(IVALUE.GT.0)MZ=IVALUE
      IVALUE=0
   90 FORMAT(' ENTER THE CONTROL NUMBER FOR THE APRORI OPTION.',/,
     .       ' (DEFAULT IS 0;  WHICH IS NO APRORI VALUES)')
      TYPE 90
      ACCEPT 61,IVALUE
      MAPR  =0
      IF(IVALUE.NE.0)MAPR=IVALUE
      IVALUE=0
  100 FORMAT(' ENTER THE DELTA TIME INCREMENT.',/,
     .       ' (DEFAULT IS 0.10)')
      TYPE 100
      ACCEPT 60,VALUE
      IF(VALUE.GT.0.)HH=VALUE
      VALUE =0.
  110 FORMAT(' ENTER THE VALUE FOR EPS.',/,' (DEFAULT IS 0.0)')
      TYPE 110
      ACCEPT 60,VALUE
      IF(VALUE.GT.0.)EPS=VALUE
      VALUE =0.
  120 FORMAT(' ENTER THE VALUE FOR TIME.',/,' (DEFAULT IS 0.0)')
      TYPE 120
      ACCEPT 60,VALUE
      IF(VALUE.GT.0.)TIME=VALUE
      VALUE =0.
  130 FORMAT(' ENTER THE VALUE FOR ALPHA.',/,' (DEFAULT IS 0.0)')
      TYPE 130
      ACCEPT 60,VALUE
      IF(VALUE.GT.0.)ALPHA=VALUE
      VALUE =0.
  140 FORMAT(' ENTER THE VALUE FOR XLA.',/,' (DEFAULT IS 1.0)')
      TYPE 140
      ACCEPT 60,VALUE
      IF(VALUE.GT.0.)XLA=VALUE
      DO 141 I=1,4
      WRITE(1,2)(BANNER(I,J),J=1,80)
  141 CONTINUE
      WRITE(1,150)NN,ITR,MZ,MAPR
      WRITE(1,160)HH,EPS,TIME,ALPHA,XLA
  150 FORMAT(7I10)
  160 FORMAT(8F10.4)
C.... ENTER THE MASS AND GEOMETRIC DATA
  170 FORMAT(' ENTER THE AIRPLANE WEIGHT.  (IN LBS)')
      TYPE 170
      ACCEPT 60,WEIGHT
      AMSS  =WEIGHT/32.174
  180 FORMAT(' ENTER THE AIRPLANE WING AREA.  (IN FT**2)')
      TYPE 180
      ACCEPT 60,S
  190 FORMAT(' ENTER THE AIRPLANE CBAR.  (IN FT)')
      TYPE 190
      ACCEPT 60,CBAR
```

```
  195 FORMAT(' ENTER THE WING SPAN.  (IN FT)')
      TYPE 195
      ACCEPT 60,SPAN
  200 FORMAT(' ENTER THE ALTITUDE OF THE FLIGHT/RUN.  (IN FT)')
      TYPE 200
      ACCEPT 60,H
C.... COMPUTE ATMOSPHERIC CONDITIONS FROM APPROXIMATE RELATIONS
      TA    =518.7-H*0.00358
      IF(TA.LT.390.)TA=390.
      PA    =2116.22*(1.-0.0000068784*H)**5.2532
      RHO   =PA/(1716.56*TA)
      AVEL   =49.02*SQRT(TA)
C.... ENTER THE STEADY-STATE FLIGHT CONDITIONS
  210 FORMAT(' ENTER THE STEADY STATE VELOCITY.  (IN FT/SEC)')
      TYPE 210
      ACCEPT 60,U1
      CL1   =2.*WEIGHT/(RHO*U1*U1*S)
C.... ASSUME L/D OF 10.
      CD1   =CL1/10.
C.... ASSUME CXT1=CD1
      CXT1  =CD1
  220 FORMAT(' ENTER THE STEADY STATE THETA.  (IN DEG)',/,
     .       ' (DEFAULT IS 0.0)')
      DGR   =57.29578
      TYPE 220
      ACCEPT 60,THA
      THA   =THA/DGR
  230 FORMAT(' ENTER THE STEADY STATE BANK ANGLE.  (IN DEG)',/,
     .       ' (DEFAULT IS 0.0)')
      TYPE 230
      ACCEPT 60,PHI
      PHI   =PHI/DGR
  240 FORMAT(' ENTER THE STEADY STATE ANGLE OF ATTACK.  (IN DEG)',/,
     .       ' (DEFAULT IS THETA)')
      TYPE 240
      VALUE =0.
      ALP   =THA*DGR
      ACCEPT 60,VALUE
      IF(VALUE.NE.0.)ALP=VALUE
      ALP   =ALP/DGR
      SINALP=SIN(ALP)
      COSALP=COS(ALP)
      SINTHA=SIN(THA)
      COSTHA=COS(THA)
      SINPHI=SIN(PHI)
      COSPHI=COS(PHI)
      TANTHA=SINTHA/COSTHA
C.... ENTER THE INERTIAL DATA
  260 FORMAT(' ENTER IYYB.  (IN SLUG*FT**2)')
      TYPE 260
      ACCEPT 60,AIY
  270 FORMAT(' ENTER IXXB.  (IN SLUG*FT**2)')
      TYPE 270
      ACCEPT 60,AIX
  280 FORMAT(' ENTER IZZB.  (IN SLUG*FT**2)')
      TYPE 280
      ACCEPT 60,AIZ
C.... SPLIT FOR CASES
      IF(CASE.EQ.'LONG')GO TO 300
      IF(CASE.EQ.'LATR')GO TO 500
      STOP
C.... LONGITUDINAL CASE
  300 CONTINUE
  310 FORMAT(' ENTER CDU, 0 OR 1. ')
      TYPE 310
  311 FORMAT(' ( 1 IF THIS IS A VARIABLE; 0 OTHERWISE )')
      TYPE 311
      ACCEPT 60,CDU,AA(2,2)
  320 FORMAT(' ENTER CXTU.')
      TYPE 320
      ACCEPT 60,CXTU
  330 FORMAT(' ENTER CDA, 0 OR 1.')
      TYPE 330
      ACCEPT 60,CDA,AA(2,3)
```

```
  340 FORMAT(' ENTER CDDE, 0 OR 1.')
      TYPE 340
      ACCEPT 60,CDDE,BB(2,1)
  350 FORMAT(' ENTER CLU, 0 OR 1.')
      TYPE 350
      ACCEPT 60,CLU,AA(3,2)
  360 FORMAT(' ENTER CLA, 0 OR 1.')
      TYPE 360
      ACCEPT 60,CLA,AA(3,3)
  370 FORMAT(' ENTER CLDE, 0 OR 1.')
      TYPE 370
      ACCEPT 60,CLDE,BB(3,1)
  380 FORMAT(' ENTER CMAD.')
      TYPE 380
      ACCEPT 60,CMAD
  390 FORMAT(' ENTER CMQ, 0 OR 1.')
      TYPE 390
      ACCEPT 60,CMQ,AA(1,1)
  400 FORMAT(' ENTER CMU, 0 OR 1.')
      TYPE 400
      ACCEPT 60,CMU,AA(1,2)
  410 FORMAT(' ENTER CMTU.')
      TYPE 410
      ACCEPT 60,CMTU
  420 FORMAT(' ENTER CMA, 0 OR 1.')
      TYPE 420
      ACCEPT 60,CMA,AA(1,3)
  430 FORMAT(' ENTER CMTA.')
      TYPE 430
      ACCEPT 60,CMTA
  440 FORMAT(' ENTER CMDE, 0 OR 1.')
      TYPE 440
      ACCEPT 60,CMDE,BB(1,1)
C.... DEFINE DIMENSIONAL DERIVATIVES
      Q1     = RHO*U1*U1/2.0
      XU     = Q1*S*(CXTU+2.*CXT1-CDU-2.*CD1)/(AMSS*U1)
      XA     =-Q1*S*(CDA-CL1)/AMSS
      XDE    =-Q1*S*CDDE/AMSS
      ZU     =-Q1*S*(CLU+2.*CL1)/(AMSS*U1*U1)
      ZA     =-Q1*S*(CLA+CD1)/(AMSS*U1)
      ZDE    =-Q1*S*CLDE/(AMSS*U1)
      AMQ    = Q1*S*CBAR*CBAR*(CMAD+CMQ)/(2.*AIY*U1)
      AMU    = Q1*S*CBAR*(CMU+CMTU)/(AIY*U1)
      AMA    = Q1*S*CBAR*(CMA+CMTA)/AIY
      AMDE   = Q1*S*CBAR*CMDE/AIY
C.... DEFINE A MATRIX ELEMENTS
      A(1,1)=AMQ
      A(1,2)=AMU
      A(1,3)=AMA
      A(1,4)=0.0
      A(2,1)=0.0
      A(2,2)=XU
      A(2,3)=XA
      A(2,4)=-COSTHA*32.174
      A(3,1)=1.0
      A(3,2)=ZU
      A(3,3)=ZA
      A(3,4)=-SINTHA*COSPHI*32.174/U1
      A(4,1)=COSPHI
      A(4,2)=0.0
      A(4,3)=0.0
      A(4,4)=0.0
C.... DEFINE B MATRIX ELEMENTS
      B(1,1)=AMDE
      B(1,2)=0.0
      B(1,3)=0.0
      B(2,1)=XDE
      B(2,2)=0.0
      B(2,3)=0.0
      B(3,1)=ZDE
      B(3,2)=0.0
      B(3,3)=0.0
```

```
      B(4,1)=0.0
      B(4,2)=0.0
      B(4,3)=0.0
C.... ALL ELEMENTS OF THE AA MATRIX ARE DEFINED
C.... DEFINE ADDITIONAL ELEMENTS OF THE BB MATRIX
      BB(1,3)=1.0
      BB(2,3)=1.0
      BB(3,3)=1.0
      BB(4,3)=1.0
C.... DEFINE AP MATRIX  (ASSUMED ORDER OF THE OBSERVATION VECTOR IS:
C.... Q, U, ALPHA, THETA, QDOT, AX-AXBIAS, AND AN-ANBIAS)
      DO 450 I=1,5
      DO 460 J=1,4
      AP(I,J)=1.0
  460 CONTINUE
  450 CONTINUE
      AP(6,2)=1.0/32.174
      AP(7,3)=-U1/32.174
C.... DEFINE BP MATRIX  (ASSUMED ORDER OF THE CONTROL VECTOR IS:
C.... DE, BIAS)
      DO 470 I=1,5
      DO 480 J=1,3
      BP(I,J)=1.0
  480 CONTINUE
  470 CONTINUE
      DO 490 I=1,3
      BP(6,I)=1.0/32.174
      BP(7,I)=-U1/32.174
  490 CONTINUE
C.... SKIP LATERAL DIRECTIONAL INPUT CASE
      GO TO 700
C.... LATERAL DIRECTIONAL CASE
  500 CONTINUE
  510 FORMAT(' ENTER CLP, 0 OR 1.')
      TYPE 510
  511 FORMAT(' ( 1 IF THIS VARIES, 0 OTHERWISE )')
      TYPE 511
      ACCEPT 60,CLP,AA(1,1)
  520 FORMAT(' ENTER CLR, 0 OR 1.')
      TYPE 520
      ACCEPT 60,CLR,AA(1,2)
  530 FORMAT(' ENTER CLB, 0 OR 1.')
      TYPE 530
      ACCEPT 60,CLB,AA(1,3)
  540 FORMAT(' ENTER CLDA, 0 OR 1.')
      TYPE 540
      ACCEPT 60,CLDA,BB(1,1)
  550 FORMAT(' ENTER CLDR, 0 OR 1.')
      TYPE 550
      ACCEPT 60,CLDR,BB(1,2)
  560 FORMAT(' ENTER CNP, 0 OR 1.')
      TYPE 360
      ACCEPT 60,CNP,AA(2,1)
  570 FORMAT(' ENTER CNR, 0 OR 1.')
      TYPE 570
      ACCEPT 60,CNR,AA(2,2)
  580 FORMAT(' ENTER CNB, 0 OR 1.')
      TYPE 580
      ACCEPT 60,CNB,AA(2,3)
  590 FORMAT(' ENTER CNDA, 0 OR 1.')
      TYPE 590
      ACCEPT 60,CNDA,BB(2,1)
  600 FORMAT(' ENTER CNDR, 0 OR 1.')
      TYPE 600
      ACCEPT 60,CNDR,BB(2,2)
  610 FORMAT(' ENTER CYB, 0 OR 1.')
      TYPE 610
      ACCEPT 60,CYB,AA(3,3)
  620 FORMAT(' ENTER CYDA, 0 OR 1.')
      TYPE 620
      ACCEPT 60,CYDA,BB(3,1)
  630 FORMAT(' ENTER CYDR, 0 OR 1.')
      TYPE 630
      ACCEPT 60,CYDR,BB(3,2)
```

```
C.... DEFINE DIMENSIONAL DERIVATIVES
      Q1     = RHO*U1*U1/2.0
      BLP    = Q1*S*SPAN*SPAN*CLP/(2.*AIX*U1)
      BLR    = Q1*S*SPAN*SPAN*CLR/(2.*AIX*U1)
      BLB    = Q1*S*SPAN*CLB/AIX
      BNP    = Q1*S*SPAN*SPAN*CNP/(2.*AIZ*U1)
      BNR    = Q1*S*SPAN*SPAN*CNR/(2.*AIZ*U1)
      BNB    = Q1*S*SPAN*CNB/AIZ
      YB     = Q1*S*CYB/(AMSS*U1)
      BLDA   = Q1*S*SPAN*CLDA/AIX
      BLDR   = Q1*S*SPAN*CLDR/AIX
      BNDA   = Q1*S*SPAN*CNDA/AIZ
      BNDR   = Q1*S*SPAN*CNDR/AIZ
      YDA    = Q1*S*SPAN*CYDA/(AMSS*U1)
      YDR    = Q1*S*SPAN*CYDR/(AMSS*U1)
C.... DEFINE A MATRIX ELEMENTS
      A(1,1)=BLP
      A(1,2)=BLR
      A(1,3)=BLB
      A(1,4)=0.0
      A(2,1)=BNP
      A(2,2)=BNR
      A(2,3)=BNB
      A(2,4)=0.0
      A(3,1)=SINALP
      A(3,2)=-COSALP
      A(3,3)=YB
      A(3,4)=32.174*COSTHA*COSPHI/U1
      A(4,1)=1.0
      A(4,2)=COSPHI*TANTHA
      A(4,3)=0.0
      A(4,4)=0.0
C.... DEFINE B MATRIX ELEMENTS
      B(1,1)=BLDA
      B(1,2)=BLDR
      B(1,3)=0.0
      B(2,1)=BNDA
      B(2,2)=BNDR
      B(2,3)=0.0
      B(3,1)=YDA
      B(3,2)=YDR
      B(3,3)=0.0
      B(4,1)=0.0
      B(4,2)=0.0
      B(4,3)=0.0
C.... ALL ELEMENTS OF THE AA MATRIX ARE DEFINED
C.... DEFINE ADDITIONAL ELEMENTS OF THE BB MATRIX
      BB(1,3)=1.0
      BB(2,3)=1.0
      BB(3,3)=1.0
      BB(4,3)=1.0
C.... DEFINE AP MATRIX  (ASSUMED ORDER OF THE OBSERVATION VECTOR IS:
C.... P, R, BETA, PHI, PDOT, RDOT, AND AY-AYBIAS)
      DO 650 I=1,6
      DO 660 J=1,4
      AP(I,J)=1.0
  660 CONTINUE
  650 CONTINUE
      AP(7,3)=U1/32.174
C.... DEFINE AP MATRIX  (ASSUMED ORDER OF THE CONTROL VECTOR IS:
C.... DA, DR, AND BIAS)
      DO 670 I=1,6
      DO 680 J=1,3
      BP(I,J)=1.0
  680 CONTINUE
  670 CONTINUE
      DO 690 I=1,3
      BP(7,I)=U1/32.174
  690 CONTINUE
  700 CONTINUE
      MU     =3
      MZ     =4
      MY     =7
C.... ECHO DATA BACK
```

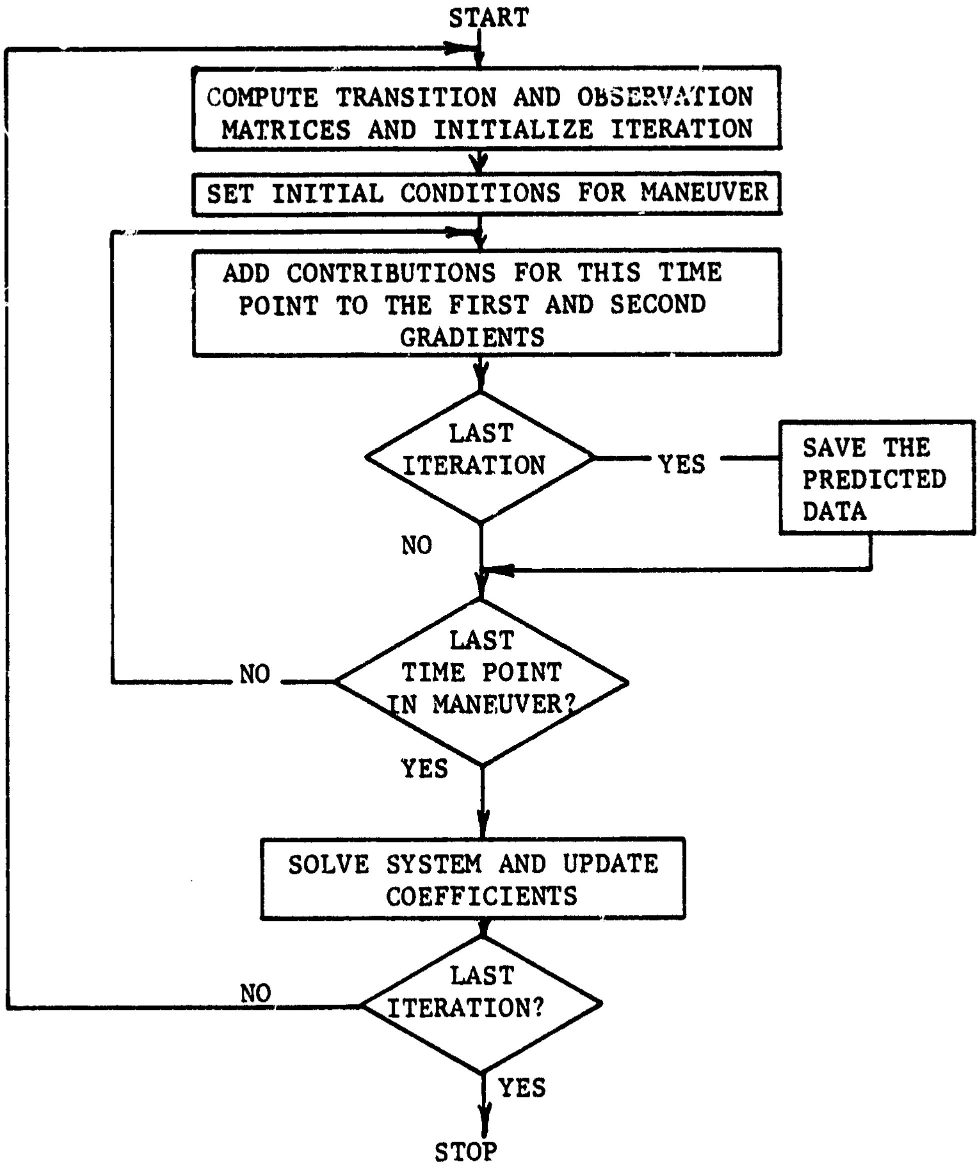
```
  710 FORMAT('  AIRPLANE INPUT DATA',/,
     .'  WING AREA           (IN FT**2)        = ',F12.4,/,
     .'  WEIGHT              (IN LBS)          = ',F12.4,/,
     .'  WING SPAN           (IN FT)           = ',F12.4,/,
     .'  CBAR                (IN FT)           = ',F12.4,/,
     .'  AIRSPEED            (IN FT/SEC)       = ',F12.4,/,
     .'  DENSITY             (IN SLUG/FT**3)   = ',F12.4,/,
     .'  ALPHA1              (IN RAD)          = ',F12.4,/,
     .'  THETA1              (IN RAD)          = ',F12.4,/,
     .'  PHI1                (IN RAD)          = ',F12.4,/,
     .'  IYY                 (IN SLUG*FT**2)   = ',F12.4,/,
     .'  IXX                 (IN SLUG*FT**2)   = ',F12.4,/,
     .'  IZZ                 (IN SLUG*FT**2)   = ',F12.4,/,
     .'  CL1                                   = ',F12.4,/,
     .'  CD1                                   = ',F12.4,/,
     .'  CXT1                                  = ',F12.4,//)
      TYPE 710,S,WEIGHT,SPAN,CBAR,U1,RHO,ALP,THA,PHI,AIY,AIX,
     .           AIZ,CL1,CD1,CXT1
C.... SPLIT FOR CASES
      IF(CASE.EQ.'LONG')GO TO 750
      IF(CASE.EQ.'LATR')GO TO 850
      STOP
C.... LONGITUDINAL CASE
C.... WRITE OUTPUT TO DISPLAY, GET WEIGHTING FACTORS, AND FINISH FILE
  750 CONTINUE
  760 FORMAT('                       LONGITUDINAL DERIVATIVES',/,
     .' CDU    ',F8.4,'                       XU    ',F8.4,/,
     .' CXTU   ',F8.4,/,
     .' CDA    ',F8.4,'                       XA    ',F8.4,/,
     .' CDDE   ',F8.4,'                       XDE   ',F8.4,/,
     .' CLU    ',F8.4,'                       ZU    ',F8.4,/,
     .' CLA    ',F8.4,'                       ZA    ',F8.4,/,
     .' CLDE   ',F8.4,'                       ZDE   ',F8.4,/,
     .' CMU    ',F8.4,'                       MU    ',F8.4,/,
     .' CMAD   ',F8.4,/,
     .' CMQ    ',F8.4,'                       MQ    ',F8.4,/,
     .' CMA    ',F8.4,'                       MA    ',F8.4,/,
     .' CMDE   ',F8.4,'                       MDE   ',F8.4,////)
      TYPE 760,CDU,XU,CXTU,CDA,XA,CDDE,XDE,CLU,ZU,CLA,ZA,
     .           CLDE,ZDE,CMU,AMU,CMAD,CMQ,AMQ,CMA,AMA,CMDE,AMDE
C.... GET THE WEIGHTING MATRIX DIAGONAL VALUES
  770 FORMAT(' ENTER THE WEIGHTING FACTOR FOR Q')
      TYPE 770
      ACCEPT 60,D1(1,1)
  780 FORMAT(' ENTER THE WEIGHTING FACTOR FOR VELOCITY.')
      TYPE 780
      ACCEPT 60,D1(2,2)
  790 FORMAT(' ENTER THE WEIGHTING FACTOR FOR ALPHA.')
      TYPE 790
      ACCEPT 60,D1(3,3)
  800 FORMAT(' ENTER THE WEIGHTING FACTOR FOR THETA.')
      TYPE 800
      ACCEPT 60,D1(4,4)
  810 FORMAT(' ENTER THE WEIGHTING FACTOR FOR QDOT.')
      TYPE 810
      ACCEPT 60,D1(5,5)
  820 FORMAT(' ENTER THE WEIGHTING FACTOR FOR AX.')
      TYPE 820
      ACCEPT 60,D1(6,6)
  830 FORMAT(' ENTER THE WEIGHTING FACTOR FOR AN.')
      TYPE 830
      ACCEPT 60,D1(7,7)
C.... SKIP PAST LATERAL DIRECTIONAL CASE
      GO TO 950
C.... LATERAL DIRECTIONAL CASE
C.... WRITE OUTPUT TO DISPLAY, GET WEIGHTING FACTORS, AND FINISH FILE
  850 CONTINUE
  860 FORMAT('                  LATERAL DIRECTIONAL DERIVATIVES',/,
     .' CYB    ',F8.4,'                       YB    ',F8.4,/,
     .' CYDA   ',F8.4,'                       YDA   ',F8.4,/,
     .' CYDR   ',F8.4,'                       YDR   ',F8.4,/,
     .' CLB    ',F8.4,'                       LB    ',F8.4,/,
     .' CLP    ',F8.4,'                       LP    ',F8.4,/,
     .' CLR    ',F8.4,'                       LR    ',F8.4,/,
     .' CLDA   ',F8.4,'                       LDA   ',F8.4,/,
```

```
     .' CLDR   ',F8.4,'                          LDR    ',F8.4,/,
     .' CNB    ',F8.4,'                          NB     ',F8.4,/,
     .' CNP    ',F8.4,'                          NP     ',F8.4,/,
     .' CNR    ',F8.4,'                          NR     ',F8.4,/,
     .' CNDA   ',F8.4,'                          NDA    ',F8.4,/,
     .' CNDR   ',F8.4,'                          NDR    ',F8.4,////)
      TYPE 860,CYB,YB,CYDA,YDA,CYDR,YDR,CLB,BLB,CLP,BLP,CLR,BLR,
     .       CLDA,BLDA,CLDR,BLDR,CNB,BNB,CNP,BNP,CNR,BNR,CNDA,
     .       BNDA,CNDR,BNDR
C.... GET THE WEIGHTING MATRIX DIAGONAL VALUES
  870 FORMAT(' ENTER THE WEIGHTING FACTOR FOR P.')
      TYPE 870
      ACCEPT 60,D1(1,1)
  880 FORMAT(' ENTER THE WEIGHTING FACTOR FOR R.')
      TYPE 880
      ACCEPT 60,D1(2,2)
  890 FORMAT(' ENTER THE WEIGHTING FACTOR FOR BETA.')
      TYPE 890
      ACCEPT 60,D1(3,3)
  900 FORMAT(' ENTER THE WEIGHTING FACTOR FOR PHI.')
      TYPE 900
      ACCEPT 60,D1(4,4)
  910 FORMAT(' ENTER THE WEIGHTING FACTOR FOR PDOT.')
      TYPE 910
      ACCEPT 60,D1(5,5)
  920 FORMAT(' ENTER THE WEIGHTING FACTOR FOR RDOT.')
      TYPE 920
      ACCEPT 60,D1(6,6)
  930 FORMAT(' ENTER THE WEIGHTING FACTOR FOR AY.')
      TYPE 930
      ACCEPT 60,D1(7,7)
  950 CONTINUE
C.... WRITE MATRICES TO FILE
  960 FORMAT(2I10)
 1160 FORMAT(7F12.6)
      WRITE(1,960)4,4
      DO 970 I=1,4
      WRITE(1,1160)(A(I,J),J=1,4)
  970 CONTINUE
      WRITE(1,960)4,MU
      DO 980 I=1,4
      WRITE(1,1160)(B(I,J),J=1,MU)
  980 CONTINUE
      WRITE(1,960)4,4
      DO 1000 I=1,4
      WRITE(1,1160)(AA(I,J),J=1,4)
 1000 CONTINUE
      WRITE(1,960)4,MU
      DO 1010 I=1,4
      WRITE(1,1160)(BB(I,J),J=1,MU)
 1010 CONTINUE
      WRITE(1,960)7,4
      DO 1020 I=1,7
      WRITE(1,1160)(AP(I,J),J=1,4)
 1020 CONTINUE
      WRITE(1,960)7,MU
      DO 1030 I=1,7
      WRITE(1,1160)(BP(I,J),J=1,MU)
 1030 CONTINUE
      WRITE(1,960)7,7
      DO 1040 I=1,7
      WRITE(1,1160)(D1(I,J),J=1,7)
 1040 CONTINUE
      WRITE(1,1160)(ZERO(I),I=1,4)
      WRITE(1,1160)(BIAS(I),I=1,4)
      STOP
          END
```

A.7.2) MAIN MMLE PROGRAMS

Description: The main program of the MMLE BONES routines acts as a controller in calling the subroutines as needed. Initially, it reads the input data for the starting conditions of the test case. If all states are measured a least squares process is used to compute the initial estimate of the derivatives. If the states are not completely measured this feature must be skipped over or errors in the solution of updates to the coefficients will result.

Flowchart:



Program listing:

```
        PROGRAM MAIN
C       *************************************************************
C
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       *                                                           *
C       *                       BONES - FRL                         *
C       *                       ===========                         *
C       *                                                           *
C       *   -   -   -   -   -   -   -   -   -   -   -   -   -   -   *
C       *                                                           *
C       *      MAIN PROGRAM OF THE MAXIMUM LIKELIHOOD ESTIMATOR     *
C       *      TECHNIQUE, (MMLE). THIS PROGRAM IS DERIVED FROM      *
C       *      THE 'BONES' PROGRAM THAT WAS ORIGINALLY DEVELOPED    *
C       *      BY NASA. THE FOLLOWING SUBROUTINES ARE REQUIRED      *
C       *      FOR THE OPERATION OF THIS PROGRAM :                  *
C       *      GIRL, EAT, CRAMER, SPIT1, REDUCE, MULT, OUTPUT       *
C       *      ADD, MAKE, ZOT, LOAD, LOAD1, SPIT, SOLVE, AND        *
C       *      DIAGIN.                                              *
C       *      THE OUTPUT OF THE PROGRAM IS TWO FILES THAT          *
C       *      CONTAINS THE MATRICES [A] AND [B] FOR EACH           *
C       *      ITERATION AND THE ESTIMATED TIME RESPONSES           *
C       *      RESPECTIVELY.  SEE THE SPECIFIC INSTRUCTIONS         *
C       *      OF THIS PROGRAM FOR FURTHER INFORMATION              *
C       *      CONCERNING THE INPUT AND OPERATION OF THIS MMLE      *
C       *      PROGRAM.                                             *
C       *      THIS MODIFIED 'BONES' PROGRAM WAS WRITTEN BY:        *
C       *                    ALEX KOTSABASIS                        *
C       *                                                           *
C       *                                      DATE 22-NOV-80       *
C       *                                                           *
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C       NEWTON-RAPHSON METHOD FOR OBTAINING STABILITY DERIVATIVES
C               LONG: ALPHA, Q, V, THETA, AN, QDOT, AX
C               L-D : P, R, BETA, PHI, PDOT, RDOT, AY
C
        COMMON MAX,MA,MAM,MAT,Z,U,D2,E1,APHI,DUM,PHI1,D1,A,B,AA,BB,
     2     BJI,XJI,SUM,FB,XT1,ZERO,D54,DD4,E,XTX,CCC,BIAS,
     3     IZE,IBIAS,IC,XLA,APR,MAPR,XT4,JKMM,XT5,AP,BP
        COMMON HH,ENN,MX,MUMX,RB,IA,JK,NB,EPS,PDB,I,MUMX1,
     2     TRACE,K,IJ,TIME,RDB,MXP1,JKM,LM,TT,JKMM1,
     3     ALPHA,NNM1,MZ,J,KJ,BD,LL,L,NN,MU,MUX,EN,PP,AAA,KM,FAC,
     4     ITR,NI,XT2,XT3,MZM1
        COMMON/MATAB/ALX,BLX,ERX
        COMMON/CONST/ KABC,KCDF
        COMMON/TRNSFR/ XL(300,7)
        DIMENSION ALX(20,10),BLX(20,10),ERX(10,10)
        DIMENSION XT5(25),APR(25)
        DIMENSION AP(8,4), BP(8,3), XT4(4)
        DIMENSION Z(7,3),U(3,3),D2(7),DD4(5,4),BIAS(5),APHI(5,4),
     2     XT1(7),PHI1(5,4),D1(8,7),A(5,4),B(5,4),AA(5,4),
     3     BB(5,4),BJI(25,4),XJI(25,7),SUM(25,25),PB(25),
     4     DUM(25,4),XT2(7),ZERO(5),D54(5,4),XT3(7)
        BYTE INAME(15),COMNT(80,4)
        COMMON/ANSWER/ KBUGG
C
        BYTE LOG,DIR,ANS
        DATA LOG,DIR/'L','D'/
C       - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C       - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
        TYPE 3511
  3511  FORMAT(///////////,20X,'* * * THE MMLE PROGRAM * * *'//)
        TYPE 3512
  3512  FORMAT(10X,'THINGS YOU HAVE TO KNOW TO RUN THIS PROGRAM:',/
     1         ,10X,'1.IS IT A LOGITUDINAL OR A LATERAL DIRACTIONAL RUN?',
     2        /,10X,'2.NAME OF FILE WITH INITIAL CONDITIONS.'/
     3         ,10X,'3.NAME OF FILE WITH MEASURED DATA.'/
     4         ,10X,'4.HOW TO DESIGNATE OUTPUT FILES.'////)
C
        TYPE 3513
```

```
3513  FORMAT(/////,10X,'INDICATE TYPE OF RUN:',/
     1       ,10X,'IF LONGITUDINAL TYPE 'L'......',/
     2       ,10X,'IF LATERAL-DIRECTIONAL TYPE 'D'......',/
     3       ,10X,'SELECT RUN:')
C
      READ(5,3514) ANS
3514  FORMAT(A1)
C
C
5     CONTINUE
83    CONTINUE
      NI = 25
      XXXX = 1.0
      MAX = 5
      MA = 4
      MZ = 7
C
C
C  ATTACH DATA FILE CONTAINING MATRICES AND INT. CONSTANTS.
      TYPE 119
119   FORMAT(//,10X,'ENTER DATA FILE NAME WITH INITIAL CONDITIONS,',
     1  /,10X,'AND MATRICES A, B, AA, AP, ETC.')
128   FORMAT(14A1)
      ACCEPT 128,(INAME(IABC),IABC=1,14)
      INAME(15)=0
      OPEN(UNIT=2,NAME=INAME,TYPE='OLD',ACCESS='SEQUENTIAL',
     1     READONLY,FORM='FORMATTED',RECORDSIZE=132)
C  CLEAR OUT OLD FILE NAME
      DO 127,IABC=1,14
      INAME(IABC)=' '
127   CONTINUE
C
1700  FORMAT(10X,10E12.4)
3777  FORMAT(12X,7I10)
3700  FORMAT(12X,8F10.4)
700   FORMAT(8F10.4)
777   FORMAT(7I10)
1010  FORMAT(10E12.4)
1011  FORMAT(I3,9E12.4)
1012  FORMAT(12E12.4)
      FACT = 1.0
      BLANC = 0.0
C  READ COMMENTS FROM INPUT FILE (4 LINES OF 80 CHARACTERS)
      READ(2,1301) ((COMNT(I,J),I=1,80),J=1,4)
1301  FORMAT(80A1)
C  READ STATMENTS
      READ(2,777) NN,ITR,MZ,MAPR
      READ(2,700) HH,EPS,TIME,ALPHA,XLA
C  LOAD MATRICES
      ENN = NN
      KCDF = NN
      KABC = ITR
      CALL LOAD(4,A,B,AA,BB)
      MAX = 8
      MA = 7
      CALL LOAD(3,AP,BP,D1,D1)
C
C      . . . . . . . . . . . . . . . . .
C
C
      MAX = 5
      MA = 4
      NNM1 = NN-1
      MU = B(MAX,2)+ .01
      MX = A(MAX,2)+ .01
C  READ IN ZEROS' AND BIASES.
C
      READ (2,700) (ZERO(I),I=1,MX),(BIAS(IA),IA=1,MX)
C
C      - - - - - - - - - - - - - - - - - - - - - - -
C
      CLOSE(UNIT=2)
```

```
C
      MXP1 = MX + 1
      MZM1 = MZ - 1
      MUX = MU + MX
      MUMX = MU*MX
      MUMX1 = MUMX + 1
      YY = 0.0
      XX = 1.0
C  ADD BIASES AND ZEROS'
      DO 49 I=1,MX
      XT4(I) = 0.0
      XT3(I) = 0.0
C       XX = XX + ZERO(I) + BIAS(I)
      XX = XX
      DO 48 J=1,MU
      YY = YY + AA(I,J) + BB(I,J)
48    XX = XX + AA(I,J) + BB(I,J)

C
C     - - - - - - - - - - - - - - - - - - - - - - - - - -
C
      YY = YY + AA(I,MX)
49    XX = XX + AA(I,MX)
C
      JKMM = YY + .01
      JKM = XX + .01
      JKMM1 = JKM - 1
      SUM(NI,1) = JKM
      SUM(NI,2) = JKM
      MAX = NI
      MA = NI
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
C  INITIALIZE MATRICES TO ZERO
C
      CALL ZOT(SUM)
C  SELECT APRIORI OPTION THRU MAPR.
C
      IF(MAPR) 176,178,177
C
C  READ IN APRIORI MATRIX
C
177   DO 261 IB=1,JKM
      DO 663 IA=1,JKM
      SUM(IB,IA)=0.0
663   CONTINUE
      DO 261 IA=1,JKM
261   SUM(IB,IA)=SUM(IB,IA)
      APR(IB) = SUM(IB,IB)
      GO TO 178
176   CONTINUE
C
      DO 664 IA=1,JKMM1
      APR(IA)=0.0
664   CONTINUE
      DO 263 IA=1,JKM
263   APR(IA) = APR(IA)*FACT
178   CONTINUE
C
C  ENTER NAME OF DATA FILE WITH MEASURED FLIGHT TEST DATA
C
      TYPE 139
139   FORMAT(//,10X,'ENTER FILE NAME CONTAINING THE MEASURED DATA')
C
C  ATTACH STATMENT FOR FILE CONTAINING MEASURED DATA
C
      ACCEPT 128,(INAME(IABC),IABC=1,14)
      OPEN(UNIT=4,NAME=INAME,TYPE='OLD',ACCESS='SEQUENTIAL',
     1       FORM='UNFORMATTED',READONLY)
C  REWIND TAPE
      REWIND 4
C
```

```
C     - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C     READ IN DATA AND PRINT OUT INITIAL CONDITIONS
C     BIASES AND ZERO'S
C
 1302 FORMAT('- - - - - - - - - - - - - - - - - - - - - - - - - - - -',
     1      ' - - - - - - - - - - - - - - - - - - - - - - - - - - -')
 1303 FORMAT(24X,'. . . . . . . .  INITIAL CONDITIONS  . . . . . . . .'/)
 1304 FORMAT(10X,'NUMBER OF DATA POINTS :                ',I3,
     1         8X,'MAXIMUM NUMBER OF ITERATIONS :      ',I3,
     2     /,10X,'DATA SAMPLING INTERVAL :         ',F10.4,
     3         5X,'FIRST DATA POINT AT TIME :      ',F10.4,
     4     /,10X,'DIAGONAL MULTIPLYING FACTOR :  ',F10.4,
     5         5X,'NUMBER OF STATES :                   ',I3,/)
 1325 FORMAT(10X,'DIAGONAL ELEMENTS OF THE WEIGHTING MATRIX D1:',/
     1          5X,7F13.3,/)
 1326 FORMAT(/,10X,'ESTIMATES OF THE [A] AND [B] MATRICES',/)
C
 1305 FORMAT(10X,'INITIAL INPUT MATRICES [A] AND [B].',/,
     1       10X,'A STAR (*) FOLLOWING THE VALUE OF A MATRIX',/,
     2       10X,'ELEMENT INDICATES THAT THE RESPECTIVE DERIVATIVE',/,
     3       10X,'IS NOT ESTIMATED BY THE MMLE METHOD.'/)
 1306 FORMAT(/,10X,'STABILITY MATRIX [A]')
 1307 FORMAT(/,10X,'CONTROL MATRIX [B]')
 1309 FORMAT(10X,'ITERATION',I3,' WAS COMPLETED'/)
C    PRINT OUT INPUT DATA
C
      PRINT 1302
      DO 1407 J=1,4
      PRINT 1421, (COMNT(I,J),I=1,80)
 1421 FORMAT(10X,80A1)
 1407 CONTINUE
      PRINT 1302
      PRINT 1303
      PRINT 1304, NN,ITR,HH,TIME,XLA,MZ
      PRINT 1302
      PRINT 703
 703  FORMAT(/,10X,'ZEROS AND BIASES')
      PRINT 1700, (ZERO(I),I=1,MX)
      PRINT 1700, (BIAS(IA),IA=1,MX)
      PRINT 1325,(D1(IBCD,IBCD),IBCD=1,7)
      PRINT 1302
C  SET MAX AND MA TO [A] AND [B] DIMENSIONS
      MAX=5
      MA =4
      PRINT 1305
      PRINT 1306
      CALL SPIT1(A,AA,1)
      PRINT 1307
      CALL SPIT1(B,BB,1)
C
C
C
C
C
C  STARTING ITERATION LOOP
C
      TT = TIME - HH
      DO 1 LM=1,NN
      TT = TT + HH
C
      U(MU,1) = 1.0
 1    CONTINUE
C     - - - - - - - - - - - - - - - - - - - - - - - - - -
C
      DO 272 IA =1,NI
      XT5(IA) = 0.0
 272  PB(IA) = 0.0
      IZE =1
      DO 276 IA=1,MX
      IF(ZERO(IA))277,276,277
 277  IZE = IZE + 1
 276  CONTINUE
```

```
C
C       - - - - - - - - - - - - - - - - - - - - - - - - -
C
C    MAIN LOOP FOR 'INT' NUMBER OF ITERATIONS
C
        DO 12 LL = 1,ITR
C
C  REWIND TAPE FOR EVERY ITERATION STEP
C
        REWIND 4
C
        TYPE 345, LL
345     FORMAT(///' NUMBER OF ITERATION PRESENTLY COMPUTED :',I3,//)
C
        MAX = 5
        MA = 4
        DO 31 JK =1,5
        DO 31 IK=1,4
        KT = IK+4*(JK-1)
        ALX(KT,LL) = A(JK,IK)
        BLX(KT,LL) = B(JK,IK)
31      CONTINUE
C
C
C       CALL SPECIAL MATRIX OUTPUT ROUTINE
C
        IF(LL.EQ.1) GO TO 1308
        PRINT 1326
        PRINT 1306
        CALL SPIT1(A,AA,LL)
        PRINT 1307
        CALL SPIT1(B,BB,LL)
1308    CONTINUE
C
C
        MAX = 5
        MAT = 5
        MAM = 4
        CALL EAT (A,HH,PHI1,APHI,D54,DD4)
        U(3,1) = 1.0
C
C
        U(3,2) = 1.0
        U(3,3) = 1.0
        XJI(NI,1) = JKM
        XJI(NI,2) = MX
        BJI(NI,1) = JKM
        BJI(NI,2) = MX
        SUM(NI,1) = JKM
        SUM(NI,2) = JKM
        MA = NI
C
C  INITIALIZE AND READ DATA FROM TAPE
C
        DO 778 IJK=1,JKM
        DO 778 JKL=1,IJK
778     SUM(IJK,JKL) = 0.0
        MAX = NI
        CALL ZOT(XJI)
C
C  READ IN THE FIRST TWO SERIES OF MEASURED DATA
C  FROM THE DATA TAPE.
C
C
        XT1(3) = 0.0
        XT2(3) = 0.0
        XT1(5) = 0.0
        XT2(5) = 0.0
        XT1(6) = 0.0
        XT2(6) = 0.0
```

```
C
      KBUGG = 1
C
      IF(ANS.EQ.LOG) GO TO 2012
C
C
      READ (4) DXY,DXY,DXY,DXY,DXY,XT1(4),XT1(1),XT1(7),XT1(2),
     1              U(1,1),U(2,1)
      READ (4) DXY,DXY,DXY,DXY,DXY,XT2(4),XT2(1),XT2(7),XT2(2),
     1             U(1,2),U(2,2)
C
C
      GO TO 2013
 2012 CONTINUE
      XT1(2) = 0.0
      XT2(2) = 0.0
      XT1(3) = 0.0
      XT2(3) = 0.0
      XT1(5) = 0.0
      XT2(5) = 0.0
      U(2,1) = 0.0
      U(2,2) = 0.0
C
      KBUGG = 0.0
C
      READ (4) XT1(4),XT1(1),XT1(7),XT1(6),U(1,1),DXY,DXY,DXY,
     1              DXY,DXY,DXY
C
      READ (4) XT2(4),XT2(1),XT2(7),XT2(6),U(1,2),DXY,DXY,DXY,
     1              DXY,DXY,DXY
C
 2013  CONTINUE
C
C
      IC = 0.0
      DO 51 I=1,MX
   51 XJI(JKM,I) = XT2(I)
      IF(LL-1) 64,65,64
   64 DO 66 IA=1,MX
      IF(ZERO(IA))67,66,67
   67 IC = IC + 1
      XT3(IA) = XT3(IA) + PB(JKM-IZE + IC)
      XT1(IA) = XT1(IA) + XT3(IA)
      XJI(JKM,IA) = XJI(JKM,IA) + XT3(IA)
      XT2(IA) = XJI(JKM,IA)
   66 CONTINUE
      IC = 0.0
C
C  ADD BIASES
C
      DO 166 IA=1,MX
      IF(BIAS(IA))167,166,167
  167 IC = IC + 1
      XT4(IA) = XT4(IA) + PB(JKMM+IC)
      XT1(IA) = XT1(IA) - XT4(IA)
      XT2(IA) = XT2(IA) - XT4(IA)
      XJI(JKM,IA) = XT2(IA)
  166 CONTINUE
   65 CONTINUE
C
C  MAIN MMLE LOOP
C
      DO 260 IA=1,JKMM
      XT5(IA) = XT5(IA)+PB(IA)
  260 CONTINUE
      DO 13 IA=1,MZ
      D2(IA) = 0.0
      Z(IA,1) = XT1(IA)
      Z(IA,2) = XT2(IA)
   13 CONTINUE
      IC =0.0
C  ZERO SPLIT
      DO 62 I=1,MX
```

```
      IF(ZERO(I))63,62,63
63    IC = IC+1
      XJI(JKM-IZE + IC,I) = 1.0
62    CONTINUE
      CALL GIRL
      MAX = NI
      MA = NI
C
C  OUTPUT OF ITERATION LOOP
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
      DO 325 IA=1,JKM
325   SUM(IA,IA) = SUM(IA,IA)*XLA
C     CALL SPIT(SUM)
      SUM(NI,1) = JKM-1
      SUM(NI,2) = JKM-1
      PRINT 1309,LL
      PRINT 1302
      IF(LL-ITR) 269,268,268
268   CALL CRAMER(MU,MX,MZ,NI)
C     CALL SPIT(SUM)
      CALL OUTPUT
      PRINT 1302
      STOP
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
269   CONTINUE
      CALL SOLVE(SUM,PB)
      NB = SUM(NI,1) + 0.01
      IJ = 0.0
      DO 18 I=1,MX
      DO 21 J=1,MU
      IF(BB(I,J))22,21,22
22    IJ = IJ +1
      B(I,J) = B(I,J) + PB(IJ)
21    CONTINUE
      DO 18 J=1,MX
      IF(AA(I,J))19,18,19
19    IJ = IJ + 1
      A(I,J) = A(I,J) + PB(IJ)
18    CONTINUE
12    CONTINUE
      GO TO 83
      RETURN
      END
```

Subroutine GIRL

Description: Subroutine GIRL performs the parameter identification.

Important variables;

SUM	Contains the second gradient in lower triangular and diagonal locations and off-diagonal a priori weighting in upper triangular. Diagonal a priori weightings are stored in APR. The first gradient appears as an extra column in SUM (the JKM column)
XJI	$\nabla_c(z_i - y_i)$
PHI1	$e^{A\Delta t}$
APHI	$\int_0^{\Delta t} e^{At}\ dt$
Z,U	measured values of observations and controls
XT1,XT2	computed values for observations
XT3	variable initial conditions on states
XT4	variable bias on the observations other than states
XT5	difference between estimated coefficients and the a priori values
PB	solution vector for the change in the estimates of the coefficients
MX	number of states
MZ	number of observations

Subroutine listing;

```
      SUBROUTINE GIRL
C
      COMMON MAX,MA,MAM,MAT,Z,U,D2,E1,APHI,DUM,PHI1,D1,A,B,AA,BB,
     1       BJI,XJI,SUM,PB,XT1,ZERO,D54,DD4,E,XTX,CCC,BIAS,
     2       IZE,IBIAS,IC,XLA,APR,MAPR,XT4,JKMM,XT5,AP,BP
      COMMON HH,ENN,MX,MUMX,RB,IA,JK,NB,EPS,PDB,I,MUMX1,
     1       TRACE,K,IJ,TIME,RDB,MXP1,JKM,LM,TT,JKMM1,
     2       ALPHA,NNM1,MZ,J,KJ,BD,LL,L,NN,MU,MUX,EN,PP,AAA,
     3       KM,FAC,ITR,NI,XT2,XT3,MZM1
      DIMENSION AP(8,4),BP(8,3),XT4(4)
      DIMENSION XT5(25),APR(25)
      DIMENSION Z(7,3),U(3,3),D2(7),DD4(5,4),BIAS(5),APHI(5,4),
     1          XT1(7),PHI1(5,4),D1(8,7),A(5,4),B(5,4),AA(5,4),
     2          BB(5,4),BJI(25,4),XJI(25,7),SUM(25,25),PB(25),
     3          DUM(25,4),XT2(7),ZERO(5),D54(5,4),XT3(7)
C
      COMMON/ANSWER/ KBUGG
      COMMON/EQDATA/ ANPT
C
      DIMENSION XL(300,7)
      COMMON/TRNSFR/XL
      COMMON/MATAB/ ALX,BLX,ERX
      DIMENSION ALX(20,10),BLX(20,10),ERX(10,10)
C . . . . . . . . . . . . . . . . . . . . . . .
C
777   FORMAT(7I10)
606   FORMAT(10X,10E12.4)
1011  FORMAT(I3,9E12.4)
1012  FORMAT(12E12.4)
      ANPT=FLOAT(NNM1)+1.0
```

```
C  TIME LOOP
      TT = TIME + MH
      DO 41 I=2,NNM1
      TT = TT + MH
      DO 28 JK=1,JKM
      DO 28 J=MXP1,MZ
28    XJI(JK,J) = 0.0
      DO 170 IA = 1,MX
170   XJI(JKM,IA) = XT2(IA)
C
C  READ MEASURED DATA FROM DATA FILE
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
      IF(KBUGG.EQ.0.0) GO TO 3071
C
      READ(4) DXY,DXY,DXY,DXY,DXY,Z(4,3),Z(1,3),Z(7,3),Z(2,3),
     1                U(1,3),U(2,3)
      Z(3,3) = 0.0
      Z(5,3) = 0.0
      Z(6,3) = 0.0
C
      GO TO 3011
3071  CONTINUE
C
C     LONGITUDINAL
C
      READ(4) Z(4,3),Z(1,3),Z(7,3),Z(6,3),U(1,3),DXY,DXY,DXY,DXY,
     1                DXY,DXY
C
C
      Z(2,3) = 0.0
      Z(5,3) = 0.0
      Z(3,3) = 0.0
      U(2,3) = 0.0
C
C     . . . . . . . . . .
3011  CONTINUE
C
C
      DO 171 IA=1,MX
171   Z(IA,3) = Z(IA,3)-XT4(IA)
      MAX = NI
      MA = 4
      CALL ZOT(BJI)
      JK = 0
      DO 44 J=1,MX
      DO 43 K=1,MU
      BJI(JKM,J) = BJI(JKM,J)+B(J,K)*(U(K,3)+U(K,2))*0.5
      IF(BB(J,K))45,43,45
45    JK = JK +1
      XJI(JK,J+MX) = U(K,2)*BP(J+MX,K)
      BJI(JK,J) =0.5*(U(K,2)+U(K,1))
43    CONTINUE
      DO 44 K = 1,MX
      IF(AA(J,K))46,44,46
46    JK = JK + 1
      IF(LL-1)4,4,4
2     CONTINUE
      BJI(JK,J)= 0.5*(Z(K,2)+Z(K,1))
      XJI(JK,J+MX) = Z(K,2)*AP(J+MX,K)
      GO TO 44
4     CONTINUE
      BJI(JK,J) = (XT2(K) + XT1(K))*0.5
      XJI(JK,J+MX) = XT2(K)*AP(J+MX,K)
44    CONTINUE
      MAX = NI
      MA = 4
      MAM = 4
      MAT = 5
      XJI(NI,2) = MX
      CALL MULT(XJI,PHI1,XJI,DUM)
      CALL MULT(BJI,APHI,DUM,DUM)
      CALL ADD(1.0,DUM,1.0,XJI,XJI)
```

```
        XJI(NI,2) = MZ
        IBIAS = 0.0
        DO 162 IA = 1,MX
        IF(BIAS(IA))163,162,163
163     IBIAS = IBIAS + 1
        DO 175 IB=1,MZ
175     XJI(JKMM+IBIAS,IB) = 0.0
        XJI(JKMM+IBIAS,IA) = 1.0
162     CONTINUE
        JKMM1 = JKM -1
        DO 7 JK = 1,JKMM
        DO 7 L = MXP1,MZ
        DO 7 K = 1,MX
        XJI(JK,L) = XJI(JK,L) + A(L-MX,K)*XJI(JK,K)*AP(L,K)
7       CONTINUE
        DO 9 L =MXP1,MZ
        XJI(JKM,L) = 0.0
        DO 8 K=1,MU
        XJI(JKM,L) = XJI(JKM,L)+B(L-MX,K)*U(K,3)*BP(L,K)
8       CONTINUE
        DO 9 K=1,MX
        XJI(JKM,L) = XJI(JKM,L)+A(L-MX,K)*XJI(JKM,K)*AP(L,K)
9       CONTINUE
        DO 3 J=1,MZ
        XT1(J) = XT2(J)
        XT2(J) = XJI(JKM,J)
        XJI(JKM,J) = Z(J,3)-XT2(J)
3       CONTINUE
        DO 27 K=1,MZ
        D2(K) = D2(K) + XJI(JKM,K)**2
27      CONTINUE
        MAX = NI
        MA = 7
81      CONTINUE
C        PRINTS OUT TIME HISTORIES
C        TYPE 606, (XT2(IA),IA=1,7),TT
        IF(LL.LT.ITR) GO TO 80
        DO 1013 IK=1,7
        XL(I,IK) = XT2(IK)
1013    CONTINUE
80      CONTINUE
        DO 91 J=1,JKM
        DO 91 I4=J,JKM
        DO 92 K=1,MZ
92      SUM(I4,J) = SUM(I4,J)+XJI(I4,K)*D1(K,K)*XJI(J,K)
91      CONTINUE
        DO 69 IA=1,MZ
        Z(IA,1) = Z(IA,2)
69      Z(IA,2) = Z(IA,3)
        U(1,1) = U(1,2)
        U(2,1) = U(2,2)
        U(1,2) = U(1,3)
        U(2,2) = U(2,3)
41      CONTINUE
C
        PRINT 607, SUM(JKM,JKM)
607     FORMAT(/,10X,'WEIGHTED ERROR SUM = ',F12.4)
C        TYPE 606, SUM(JKM,JKM)
        MAX = 8
        MA = 7
C        CALL SPIT(D1)
        PRINT 608
608     FORMAT(/,10X,'WEIGHTED ERRORS:',/)
        PRINT 606,(D2(IA)*D1(IA,IA),IA=1,MZ)
        TYPE 606,(D2(IA),IA=1,MZ)
        DO 2101 IA=1,MZ
        ERX(IA,LL) = D2(IA)
2101    CONTINUE
        DO 888 IJK = 1,JKM
888     SUM(IJK,JKM) = SUM(JKM,IJK)
        IF(MAPR) 180,181,180
180     DO 182 IB=1,JKM
```

```
      SUM(IB,JKM) =-XTS(IB)*APR(IB)+SUM(IB,JKM)
      SUM(IB,IB) = SUM(IB,IB)+APR(IB)
      IBM1 = IB-1
      DO 182 IA =1,IBM1
182   SUM(IB,IA) = SUM(IB,IA) + SUM(IA,IB)
181   CONTINUE
531   FORMAT(////'     END OF ITERATION '//)
      TYPE 531
      RETURN
      END
```

Subroutine EAT

Description: Subroutine EAT computes $e^{A\Delta t}$ and $\int_0^{\Delta t} e^{At}\,dt$ using the Taylor series expansion to ten terms. These are returned as PHI1 and APHI1 respectively.

subroutine listing:

```
      SUBROUTINE EAT (A,T,PHI,APHI,A2,A3)
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C     THIS SUBROUTINE COMPUTES THE TRANSITION MATRIX
C     AND IT'S INTEGRAL USING A TAYLOR SERIES EXPANSION
C     TO 10 TERMS.
C
C     A = STABILITY MATRIX
C     T = DELTA TIME INCREMENT
C     PHI = TRANSITION MATRIX
C     APHI = INTEGRAL OF THE TRANSITION MATRIX
C     A2 = DUMMY MATRIX
C     A3 = DUMMY MATRIX
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION A(1),PHI(1),A2(1),APHI(1),A3(1)
C
C  CALLS MULTIPLICATION AND ADDITION SUBROUTINES
C
C
      MAX2 = MAX*2
      II = A(MAX)
      JJ = A(MAX2)
      PHI(MAX) = A(MAX)
      PHI(MAX2) = A(MAX)
C
C  INITIALIZE TO ZERO AND CREATE NEW MATRICES
C
      CALL ZOT(PHI)
      CALL MAKE(APHI,PHI)
      CALL MAKE(A3,PHI)
      MI = -MAX
      DO 1 I=1,II
      MI = MI+MAX
      MII = MI+I
      PHI(MI+I) = 1.0
1     CONTINUE
      CALL MAKE(A2,PHI)
      G = 1.0
      DO 2 I=1,10
      BB = I
      G = G*T/BB
      CALL ADD(1.,APHI,G,A2,APHI)
      CALL MULT(A,A2,A2,A3)
      CALL ADD(1.,PHI,G,A2,PHI)
2     CONTINUE
      DO 10 I=1,II
      DO 10 J=1,I
```

```
      JI = (I-1)*MAX+J
      IJ = (J-1)*MAX+I
      TEMP = PHI(IJ)
      PHI(IJ) = PHI(JI)
      PHI(JI) = TEMP
      TEMP = APHI(IJ)
      APHI(IJ) = APHI(JI)
 10   APHI(JI) = TEMP
C      CALL SPIT(PHI)
C      CALL SPIT(APHI)
      RETURN
      END
```

Subroutine ZOT

Description: Subroutine ZOT intializes the elements of a matrix to zero.

Subroutine listing:

```
      SUBROUTINE ZOT(X)
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C     THIS SUBROUTINE SETS ALL ELEMENTS OF A MATRIX
C     TO ZERO.
C
C     X : MATRIX TO BE ZEROED
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION X(1)
C
      MAX2 = MAX * 2
      IIM1 = X(MAX) -1.0
      JJM1 = X(MAX2) -1.0
      LEND = JJM1*MAX+1
      DO 1 L=1,LEND,MAX
      KEND = L + IIM1
      DO 1 K=L,KEND
 1    X(K) = 0.0
      RETURN
      END
```

Subroutine LOAD

Description: Subroutine LOAD loads matrices from the input file.

Subroutine listing:

```
      SUBROUTINE LOAD (N,A,B,C,D)
C
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C     THIS SUBROUTINE LOADS MATRICES A,B,C AND D FROM
C     AN INPUT FILE. THE VARIABLE N SPECIFIES THE NUMBER
C     OF MATRICES TO BE LOADED.
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
      REAL A(1),B(1),C(1),D(1)
C
      CALL LOAD1(A)
      IF(N.LT.2) RETURN
      CALL LOAD1(B)
      IF(N.LT.3) RETURN
      CALL LOAD1(C)
      IF(N.LT.4) RETURN
      CALL LOAD1(D)
      RETURN
      END
```

Subroutine LOAD1

Description: Subroutine LOAD1 actually loads the matrix from the input file.

Subroutine listing:

```
      SUBROUTINE LOAD1(A)
C  ROUTINE CALLED BY LOAD LOADS MATRIX A FROM FILE
C
      COMMON MAX
      REAL A(1)
      READ(2,100) II,JJ
100   FORMAT(8X,I2,I10)
      KE =(JJ-1)*MAX
      DO 10 I=1,II
      KEND = I+KE
10    READ(2,1001) (A(K),K=I,KEND,MAX)
      A(MAX) = II
      A(MAX*2) = JJ
1001  FORMAT(8F12.6)
      RETURN
      END
```

Subroutine ADD

Description: Subroutine ADD adds scalar multiples of two matrices, $Z = g\ X + h\ Y$.

Subroutine listing:

```
      SUBROUTINE ADD (G,X,H,Y,Z)
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C     THIS SUBROUTINE ADDS SCALAR MULTIPLES OF TWO
C     MATRICES AS FOLLOWS:
C
C          [Z] = G*[X] + H*[Y]   WITH : G = 1.0
C     ( NO CHECKING IS MADE FOR MATRIX COMPATIBILITY )
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION X(1),Y(1),Z(1)
C
      MAX2 = MAX * 2
      II = X(MAX)
      JJ = X(MAX2)
      JEND = (JJ-1)*MAX+1
      IIM1 = II-1
      DO 53 J=1,JEND,MAX
      KEND = J+IIM1
      DO 53 K=J,KEND
53    Z(K) = X(K)+H*Y(K)
      Z(MAX) = X(MAX)
      Z(MAX2) = X(MAX2)
      RETURN
      END
```

Subroutine MAKE

Description: Subroutine MAKE moves a copy of the matrix Y into X.

Subroutine listing:

```
      SUBROUTINE MAKE(X,Y)
C
C     - - - - - - - - - - - - - - - - - - - - - - - -
C     THIS SUBROUTINE GENERATES A MATRIX X THAT IS
C     A COPY OF MATRIX Y.
C
C     X : NEW MATRIX, COPY OF Y
C     Y : MATRIX TO BE COPIED
C     - - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION X(1),Y(1)
C
      MAX2 = MAX*2
      IIM1 = Y(MAX) -1.
      JJM1 = Y(MAX2) -1.
      LEND = JJM1*MAX +1
      DO 1 L=1,LEND,MAX
      KEND = L+IIM1
      DO 1 K=L,KEND
1     X(K) = Y(K)
      X(MAX) = Y(MAX)
      X(MAX2) = Y(MAX2)
      RETURN
      END
```

Subroutine MULT

Description: Subroutine MULT computes the matrix product ,C = A B. The matrix C can not be the same as matrix A or B.

Subroutine listing:

```
      SUBROUTINE MULT (A,B,C,D)
C  MULTIPLIES A AND B AND PUTS THE PRODUCT
C  IN C AND D (USING SUB.MAKE)
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION A(1),B(1),C(1),D(1)
C .   .   .   .   .   .   .   .   .   .
      MAX2 = MAX*2
      MIX2 = MIX*2
      II = A(MAX)
      JJ = A(MAX2)
      KK = B(MIX2)
      JE = (JJ-1)*MAX
      KE = (KK-1)*MAX
      DO 20 I=1,II
      KEND = KE+I
      JEND = JE+I
      L = 1
      DO 20 K=I,KEND,MAX
      D(K) = 0.0
      JB = L
C  INITIALIZATION LOOP
      DO 10 J=I,JEND,MAX
      D(K) = A(J)*B(JB)+D(K)
```

```
 10   JB = JB + 1
 20   L = L + MIX
      D(MAX) = A(MAX)
      D(MAX2) = B(MIX2)
C  COPY D INTO C
      CALL MAKE(C,D)
      RETURN
      END
```

Subroutine SPIT

Description: Subroutine SPIT prints out a marix.

Subroutine listing:

```
      SUBROUTINE SPIT(X)
C
C     SUBROUTINE USED FOR THE PRINTOUT OF MATRICES
C
C     - - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION X(1)
100   FORMAT(13X,' DIMENSION ',8X,I3,' BY ',I3)
101   FORMAT(12X,10E12.4)
C
      MAX2 = MAX*2
      II = X(MAX)
      JJ = X(MAX2)
      PRINT 100, II,JJ
      KE = (JJ-1)*MAX
      DO 1 I=1,II
      KEND = I+KE
1     PRINT 101, (X(K),K=I,KEND,MAX)
      RETURN
      END
```

Subroutine SPIT1

Description: Subroutine SPIT1 prints out the A and B matrices with "*" 's to show which of the parameters have been allowed to vary.

Subroutine listing:

```
      SUBROUTINE SPIT1(X,XX,KI)
C
C     SUBROUTINE USED FOR THE PRINTOUT OF MATRICES
C
C     - - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MAX1,MIX1,MIX
      DIMENSION X(1),XX(1)
      BYTE CHAR(4)
100   FORMAT(10X,'DIMENSION ',I3,' BY',I3)
101   FORMAT(10X,5(PE12.4,A1))
C
C
      MAX2 = MAX*2
      II = X(MAX)
      JJ = X(MAX2)
      PRINT 100, II,JJ
      KE = (JJ-1)*MAX
      DO 1 I=1,II
```

```
      KEND = I+KE
      DO 2 K=I,KEND,MAX
      CHAR((K-I)/MAX+1)=' '
      IF(XX(K).EQ.0.)CHAR((K-I)/MAX+1)='*'
2     CONTINUE
1     PRINT 101,((X(K),CHAR((K-I)/MAX+1)),K=I,KEND,MAX)
      RETURN
      END
```

Subroutine SOLVE

Description: Subroutine SOLVE solves the system of linear equations, A x = b where A is symmetrical. Only the lower triangular and diagonal elements of A are used. The b vector is assumed to be stored in the N+1 column of A, where N is the dimension of the system.

Subroutine listing:

```
      SUBROUTINE SOLVE(A,X)
C
C  SOLVES SYSTEM AX = B WHERE A SYMMETRIC MATRIX
C  AND B A MATRIX IN N+1 COLUMN OF A
C
      REAL A(25,25),X(25)
      CALL REDUCE(A)
      N = A(25,1)
      NM1 = N-1
      NP1 = N+1
C  MULTIPLY MATRICES, (L) * (B) . . .
      DO 70 I=2,N
      X(I) = A(I,NP1)
      IM1 = I-1
      DO 70 J=1,IM1
70    X(I) = X(I)+A(I,J)*A(J,NP1)
C  MULTIPLY BY (DI)
C
      A(1,NP1) = A(1,NP1)/A(1,1)
      DO 80 I=2,N
80    A(I,NP1) = X(I)/A(I,I)
C  MULTIPLY BY (L*) TO FORM (L*)*(DI)*(L)*(B)
C
      DO 90 I=1,NM1
      X(I) = A(I,NP1)
      IP1 = I+1
      DO 90 J=IP1,N
90    X(I) = X(I)+A(J,I)*A(J,NP1)
      X(N) = A(N,NP1)
C
      RETURN
      END
```

Subroutine DIAGIN

Description: Subroutine DIAGIN obtains the diagonal elements of the inverse of a symmetric matrix.

Subroutine listing:

```
      SUBROUTINE DIAGIN(A)
C
C  FIND DIAGONAL ELEMENTS OF A INVERSE FOR SYMMETRIC A
C
      REAL A(25,25)
      CALL REDUCE(A)
      N = A(25,1)
      NM1 = N-1
      DO 90 I=1,NM1
      A(I,I) = 1.0/A(I,I)
      IP1 = I+1
      DO 90 J=IP1,N
90    A(I,I) =A(I,I)+A(J,I)**2/A(J,J)
      A(N,N) = 1.0/A(N,N)
      RETURN
      END
```

Subroutine REDUCE

Description: Subroutine REDUCE factors a symmetric matrix A by Cholesky's matrix decomposition.

Subroutine listing:

```
      SUBROUTINE REDUCE(A)
C  REDUCES SYMMETRIC MATRIX A STORED IN LOWER TRIANGULAR LOCATIONS
C  TO THE FORM (LI)*(D)*(LI*) WHERE L IS A LOWER TRINGULAR MATRIX
C  WITH UNITY DIAGONAL TERMS, D IS A DIGONAL MATRIX, I DENOTES
C  INVERSE AND * TRANSPOSE.
C
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
      REAL A(25,25)
      N = A(25,1)
      NM1 = N-1
C  MAIN DO LOOP
      DO 20 K=1,NM1
      KP1 = K+1
      KM1 = K-1
      AKKI = 1.0/A(K,K)
C
      DO 20 I=KP1,N
      AKKIK = A(I,K)*AKKI
      DO 10 J=I,N
10    A(J,I) = A(J,I)-AKKIK*A(J,K)
      A(I,K) =-AKKIK
      IF(KM1.EQ.0.0) GO TO 20
      DO 15 J=1,KM1
15    A(I,J) = A(I,J)-AKKIK*A(K,J)
20    CONTINUE
C
C  (L) IS NOW STORED IN LOWER TRIANGULAR PART OF (A)
C  EXCEPT OF MATRIX DIAGONAL, WHICH CONTAINS D
      RETURN
      END
```

Subroutine CRAMER

Description: Subroutine CRAMER computes the confidence levels of the estimated derivatives.

Subroutine listing:

```
      SUBROUTINE CRAMER(MU,MX,MZ,NI)
C
C     - - - - - - - - - - - - - - - - - - - - - - -
C
C     THIS SUBROUTINE COMPUTES THE CRAMER-RAO BOUNDS
C     ALSO KNOWN AS THE CONFIDENCE LEVELS OF THE
C     ESTIMATED DERIVATIVES.
C     MU = NUMBER OF CONTROL INPUTS
C     MX = NUMBER OF STATES
C     MZ = NUMBER OF OBSERVATIONS
C     NI = MAX. NUMBER OF UNKNOWNS (25)
C
C     - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON MAX,MA,MAM,MAT,Z,U,D2,E1,APHI,DUM,PHI1,D1,A,B,AA,BB,
     1       BJI,XJI,SUM,PB,XT1,ZERO,D54,DD4,E,XIX,CCC,BIAS,
     2       IZE,IBIAS,IC,XLA,APR,MAPR,XT4,JKMM,XT5,AP,BP
      COMMON /EQDATA/ ANPT
      DIMENSION AC(5,4),BC(5,4)
      DIMENSION XT5(25),APR(25)
      DIMENSION AP(8,4),BP(8,3),XT4(4)
      DIMENSION Z(7,3),U(3,3),D2(7),DD4(5,4),BIAS(5),APHI(5,4),
     1           XT1(7),PHI1(5,4),D1(8,7),A(5,4),B(5,4),AA(5,4),
     2           BB(5,4),BJI(25,4),XJI(25,7),SUM(25,25),PB(25),
     3           DUM(25,4),XT2(7),ZERO(5),D54(5,4),XT3(7)
C
C     NORMALLY THE APRIORI CONTRIBUTION TO HESSIAN IS SUBTRACTED
C     FOR THIS COMPUTATION BUT THIS ROUTINE ASSUMES NO APRIORI
C     OPTIONS ARE BEING USED AND HENSE THERE ARE NO CONTRIBUTIONS
C
      AC(5,1) = MX
      AC(5,2) = MX
      BC(5,1) = MX
      BC(5,2) = MU
      JKMM1 = SUM(NI,1) + 1.01
C
C     STORE WEIGHTED ERROR SUM IN ERRSUM
      ERRSUM = SUM(JKMM1,JKMM1)/ANPT
C
C     OBTAIN DIAGONAL ELEMENTS OF INVERSE
C
      CALL DIAGIN(SUM)
C
C     COMPUTE CRAMER-RAO BOUNDS
C
      WTS = 0.0
      DO 1 I=1,MZ
      IF(D1(I,I).NE.0.0) WTS = WTS + 1.0
    1 CONTINUE
      COEFF = ERRSUM/WTS
      PRINT 10,ERRSUM,COEFF,WTS
   10 FORMAT(' ERRSUM = ',F12.4,' COEFF = ',F12.4,' WTS = ',F12.4)
      L = 0
      DO 2 I=1,MX
      DO 3 J=1,MU
      BC(I,J) = 0.0
      IF(BB(I,J).NE.1.) GO TO 3
      L = L+1
      BC(I,J) = SQRT(ABS(SUM(L,L))*COEFF)
    3 CONTINUE
```

```
      DO 4 J=1,MX
      AC(I,J) = 0
      IF(AA(I,J).NE.1.) GO TO 4
      L = L+1
      AC(I,J) = SQRT(ABS(SUM(L,L))*COEFF)
4     CONTINUE
2     CONTINUE
      MAX = 5
      PRINT 6
6     FORMAT(' AC MATRIX')
      CALL SPIT(AC)
      PRINT 7
7     FORMAT(' BC MATRIX')
      CALL SPIT(BC)
      RFTURN
      END
```

Subroutine OUTPUT

Description: Subroutine OUTPUT provides the output of time histories and matrices to user defined files for later plotting.

Subroutine listing:

```
      SUBROUTINE OUTPUT
C     - - - - - - - - - - - - - - - - - - - - - - -
C     THIS SUBROUTINE WILL PROVIDE MMLE RESULTS
C     IN A FILE TO BE SPECIFIED BY THE USER.
C     THE FILE WILL CONTAIN INFORMATION ABOUT
C     THE MATRICES A AND B AT EACH ITERATION.
C     - - - - - - - - - - - - - - - - - - - - - - -
C
      COMMON/MATAB/ ALX,BLX
      COMMON/TRNSFR/ XL
      COMMON/CONST/ ITR,NN
      BYTE INAME(15)
      DIMENSION ALX(20,10),BLX(20,10)
      DIMENSION XL(300,7)
C
C
      TYPE 10
10    FORMAT(/' ENTER FILE NAME FOR OUTPUT OF MMLE MATRICES'/,'*')
      ACCEPT 11,(INAME(IAB),IAB=1,14)
11    FORMAT(14A1)
      INAME(15)=0
      OPEN(UNIT=2,NAME=INAME,TYPE='NEW',ACCESS='SEQUENTIAL',
     1    FORM='FORMATTED',BUFFERCOUNT=2)
C     . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
35    FORMAT(10E12.4)
40    FORMAT(' *** MATRIX A ***'/)
50    FORMAT(' *** MATRIX B *** '/)
60    FORMAT('  ITERATION ... ',I2,/)
70    FORMAT(//,' ESTIMATED TIME RESPONSES'//)
      DO 130 I=1,ITR
      WRITE(2,60) I
      WRITE(2,40)
      DO 30 J=1,14,4
      WRITE(2,35) (ALX(J-1+K,I),K=1,4)
30    CONTINUE
      WRITE(2,50)
      DO 131 J=1,14,4
      WRITE(2,35) (BLX(J-1+K,I),K=1,3)
131   CONTINUE
130   CONTINUE
C . . . . . . . . . . . .
      CALL CLOSE(2)
```

```
C
        TYPE 80
80      FORMAT(/' TYPE IN FILE THAT WILL CONTAIN',
       1         ' LAST INERATION TIME RESPONSES.'/,'$')
        ACCEPT 11,(INAME(IAB),IAB=1,14)
        OPEN(UNIT=3,NAME=INAME,TYPE='NEW',ACCESS='SEQUENTIAL',
       1     FORM='UNFORMATTED',BUFFERCOUNT=2)
C
C
        DO 330 N=2,NN
        WRITE(3) (XL(N,I),I=1,7)
330     CONTINUE
        WRITE(3)
        RETURN
        END
```

A.7.3) MMLE OUTPUT FORMAT

Following is an example and description of the MMLE output.

Longitudinal:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    KU FRL BONES MMLE RESULTS
                    CESSNA 172     LONGITUDINAL CAFE
                    3000. FT ALT. AT 176. FPS AIRSPEED
                    FLIGHT 19/10/80  RUN 23
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 . . . . . . . .  INITIAL CONDITIONS  . . . . . . . .

NUMBER OF DATA POINTS :              240        MAXIMUM NUMBER OF ITERATIONS :        9
DATA SAMPLING INTERVAL :             0.1000     FIRST DATA POINT AT TIME :          0.0000
DIAGONAL MULTIPLYING FACTOR :        1.0000     NUMBER OF STATES :                    7

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ZEROS AND BIASES
  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
DIAGONAL ELEMENTS OF THE WEIGHTING MATRIX D1:
 100.000        0.000        0.000       700.000        0.009       40.000       5.000

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
INITIAL INPUT MATRICES [A] AND [B].
A STAR (*) FOLLOWING THE VALUE OF A MATRIX
ELEMENT INDICATES THAT THE RESPECTIVE DERIVATIVE
IS NOT ESTIMATED BY THE MMLE METHOD.


STABILITY MATRIX [A]
DIMENSION    4 BY  4                                          1
 -0.5863E+01   0.0000E+00* -0.2130E+02   0.0000E+00*
  0.0000E+00*  0.0000E+00*  0.1934E+02  -0.3216E+02*
  0.1000E+01*  0.0000E+00* -0.2595E+01  -0.4560E-02*
  0.9995E+00*  0.0000E+00*  0.0000E+00*  0.0000E+00*

CONTROL MATRIX [B]
DIMENSION    4 BY  3                    2
 -0.3063E+02   0.0000E+00*  0.0000E+00
 -0.5915E+01   0.0000E+00*  0.0000E+00
 -0.2409E+00   0.0000E+00*  0.0000E+00
  0.0000E+00*  0.0000E+00*  0.0000E+00

WEIGHTED ERROR SUM = 2197421.2500

WEIGHTED ERRORS:
                                                                                    3
  0.1481E+04  0.0000E+00  0.0000E+00  0.2193E+07  0.0000E+00  0.1147E+02  0.2508E+04
ITERATION  1 WAS COMPLETED

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

1
$$\begin{bmatrix} M'_q & M'_u & M'_\alpha & M'_\theta \\ 0 & X'_u & X'_\alpha & -g\cos(\theta_1) \\ \frac{Z_q + U_1}{U_1 - Z_{\dot{\alpha}}} & Z'_u & Z'_\alpha & \frac{-g}{U_1 - Z_{\dot{\alpha}}}\sin(\theta_1)\cos(\phi_1) \\ \cos(\phi_1) & 0 & 0 & 0 \end{bmatrix}$$

2
$$\begin{bmatrix} M_{\delta_E}' & M_{\delta_c}' & M'_o \\ X_{\delta_E}' & X_{\delta_c}' & X'_o \\ Z_{\delta_E}' & Z_{\delta_c}' & Z'_o \\ 0 & 0 & \dot{\delta}'_o \end{bmatrix}$$

3 $(q,U,\alpha,\theta,A_X,A_N)$

Lateral:

```
. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    KU FRL BONES MMLE RESULTS
                    CESSNA 172    LATERAL-DIRECTIONAL CASE
                    3000. FT. ALT. AT 176. FPS  AIRSPEED                              .
                    FLIGHT 19/10/80    RUN 45
. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                . . . . . . . .  INITIAL CONDITIONS  . . . . . . . .

NUMBER OF DATA POINTS :              140       MAXIMUM NUMBER OF ITERATIONS :       9
DATA SAMPLING INTERVAL :             0.1000    FIRST DATA POINT AT TIME :      0.0000
DIAGONAL MULTIPLYING FACTOR :        1.0000    NUMBER OF STATES :                   7

. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ZEROS AND BIASES
  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
DIAGONAL ELEMENTS OF THE WEIGHTING MATRIX D1:
  60.500       40.000        0.000      150.000        0.000        0.000      100.000

. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
INITIAL INPUT MATRICES [A] AND [B].
A STAR (*) FOLLOWING THE VALUE OF A MATRIX
ELEMENT INDICATES THAT THE RESPECTIVE DERIVATIVE
IS NOT ESTIMATED BY THE MMLE METHOD.


STABILITY MATRIX [A]
DIMENSION    4 BY   4
 -0.9749E+01   0.1791E+01  -0.1815E+02   0.0000E+00*     1
 -0.3386E+00  -0.1117E+01   0.7214E+01   0.0000E+00*
  0.3943E-01* -0.9992E+00* -0.1737E+00   0.1827E+00*
  0.1000E+01*  0.3946E-01*  0.0000E+00*  0.0000E+00*

CONTROL MATRIX [B]
DIMENSION    4 BY   3                          2
  0.3630E+02   0.2998E+01   0.0000E+00
 -0.5882E+01  -0.7291E+01   0.0000E+00
  0.0000E+00   0.3750E+01   0.0000E+00
  0.0000E+00*  0.0000E+00*  0.0000E+00

WEIGHTED ERROR SUM =   52015.3281

WEIGHTED ERRORS:
                                                                                        3
  0.2387E+03  0.8692E+03  0.0000E+00  0.1239E+05  0.0000E+00  0.0000E+00  0.3852E+05
ITERATION  1 WAS COMPLETED
```

$$
{}^{1}\begin{bmatrix} L'_p & L'_r & L'_\beta & 0.0 \\ N'_p & N'_r & N'_\beta & 0.0 \\ \sin(\alpha_1) & -\cos(\alpha_1) & Y'_\beta & \frac{g}{U_1}\cos(\theta_1)\cos(\phi_1) \\ 1.0 & \cos(\phi_1)\tan(\theta_1) & 0.0 & 0.0 \end{bmatrix}
\qquad
{}^{2}\begin{bmatrix} L_{\delta_A}' & L_{\delta_r}' & L'_o \\ N_{\delta_A}' & N_{\delta_r}' & N'_o \\ Y_{\delta_A}' & Y_{\delta_r}' & Y'_o \\ 0.0 & 0.0 & \dot{\phi}'_o \end{bmatrix}
$$

$^{3}(p, r, \beta, \phi, \dot{p}, \dot{r}, A_Y)$

PRECEDING PAGE BLANK NOT FILMED

A.8) TIME HISTORY PLOTTING

Description: The PLOT03 program collects the time histories of both the actual inflight measurements, and the predicted states of the MMLE BONES program. These are then plotted on the graphics CRT terminal. This process allows the user to observe the fit to the measured states. For hard copy plots the time histories are transfered to the KU-FRL's Hewlett Packard computer.

Program listing:

```
C  * * * * * * * * * * * * * * * * * * * * * *
C  *                                         *
C  *  THIS PROGRAM CAN BE USED TO PLOT       *
C  *  MEASURED AND ESTIMATED DATA.           *
C  *  DATE 18-NOV-80                         *
C  *  THE REQUIRED SUBROUTINES ARE:          *
C  *  PLOT55,INIT AND GRAPH. LINK AS         *
C  *  FOLLOWS : PLOT=MAIN,PLOT55,INIT,       *
C  *  GRAPH (CR).                            *
C  * * * * * * * * * * * * * * * * * * * * * *
C
         COMMON/STATUS/ISTAT(16)
         DIMENSION IARRAY(512),HH(9,1),RDATA(240,9),EDATA(240,9)
         DIMENSION GAIN(9),GAIN1(9),GAIN2(9)
         COMMON/FIVEA/ DATA,GAIN
         BYTE YES,NO,ANS1,LOG,DIR,ANS2,NAME(15)
         DATA LOG,DIR /'L','D'/
         DATA YES,NO /'Y','N'/
         DATA ISTAT/16*0/
         DATA GAIN1/114.6,114.6,143.2,47.7,57.3,57.3,100.,143.2,143.2/
         DATA GAIN2/114,6,2.5,143.2,95.5,57.3,100.,25.,143.2,0./
C
C        - - - - - - - - - - - - - - - - - - - - - - -
C
         TYPE 1
   1     FORMAT(/////'          HELLOOOOOO.........',////,
        1'      THIS IS A PLOTTING PROGRAM USED TO PLOT'/,
        2'      MEASURED VERSUS ESTIMATED TIME HISTORIES'//,
        3'      INSTRUCTIONS: '/,
        4'      1. TYPE IN TYPE OF MANOEUVRE.'/,
        5'      2. TYPE IN NUMBER OF TIME POINTS N (0-250).'/,
        6'      3. INDICATE NAME OF FILE CONTAINING MEASURED DATA.'/,
        7'      4. INDICATE NAME OF FILE CONTAINING ESTIMATED DATA.'/,
        8'      5. WHEN DATA IS PLOTTED HIT 'CR' TO CONTINUE.'/,
        9'      6.                    GOOD LUCK.......   ALEX'//)
C
C
C
         TYPE 301
   301   FORMAT(' INDICATE TYPE OF MANOEUVRE,'/
        1       ,' 'L' FOR LONGITUDINAL'/
        2       ,' 'D' FOR LATERAL DIRECTIONAL'/)
         READ(5,302) ANS1
   302   FORMAT(A1)
C
C
C
C        READ IN NUMBER OF TIME POINTS
C
```

```
 10     FORMAT(I3)
 20     FORMAT(' TYPE IN NUMBER OF TIME POINT'/)
        TYPE 20
        READ(5,10) N
 C
 C      READ DATA FILE WITH MEASURED DATA
 C
        TYPE 30
 30     FORMAT(' TYPE IN NAME OF DATA FILE WITH MEASURED DATA'/)
 C
        NAME(15) = 0
 31     FORMAT(14A1)
        ACCEPT 31, (NAME(I),I=1,14)
        OPEN(UNIT=2,NAME=NAME,TYPE='OLD',ACCESS='SEQUENTIAL',
       1     READONLY,FORM='UNFORMATTED')
        DO 32 I=1,14
        NAME(I) =' '
 32     CONTINUE
 C
 C
 40     FORMAT(12E12.4)
        DO 50 I=1,N
        IF(ANS1.EQ.LOG) GO TO 310
 C
 C
 C
        READ(2) DMY,DMY,DMY,DMY,DMY,HH(4,1),HH(1,1),HH(7,1),
       1            HH(2,1),HH(8,1),HH(9,1)
        HH(3,1) = 0.0
        HH(5,1) = 0.0
        HH(6,1) = 0.0
        GO TO 311
 310    CONTINUE
        READ(2) HH(4,1),HH(1,1),HH(7,1),HH(6,1),HH(8,1),DMY,
       1            DMY,DMY,DMY,DMY,DMY
 C
 C
        HH(2,1) = 0.0
        HH(3,1) = 0.0
        HH(5,1) = 0.0
        HH(9,1) = 0.0
 C
 C
 311    CONTINUE
        DO 50 J=1,9
        RDATA(I,J) = HH(J,1)
 50     CONTINUE
 C
 C
 C      - - - - - - - - - - - - - - - - - - - - - - - - -
 C
        TYPE 60
 60     FORMAT(' TYPE IN NAME OF DATA FILE WITH ESTIMATED DATA'/)
 C
        ACCEPT 31, (NAME(I),I=1,14)
        OPEN(UNIT=3,NAME=NAME,TYPE='OLD',ACCESS='SEQUENTIAL',
       1     READONLY,FORM='UNFORMATTED')
        DO 81 K=1,14
        NAME(I) =' '
 81     CONTINUE
 70     FORMAT(12X,9E12.4)
        DO 80 K=1,N-2
        READ(3) (HH(L,1),L=1,7)
        DO 80 J=1,7
        EDATA(K+2,J) = HH(J,1)
 80     CONTINUE
 C
 C
        DO 399 J=1,7
        EDATA(1,J) = 0.0
```

```
      EDATA(2,J) = 0.0
399   CONTINUE
C
      DO 500 I=1,N
      EDATA(I,8)=0.0
      EDATA(I,9)=0.0
500   CONTINUE
      IARRAY(2) = 0
      IARRAY(3) = 0
113   CONTINUE
C     - - - - - - - - - - - - - - - - - - - - - - -
C
      TYPE 100
100   FORMAT('  * * * SELECT VARIABLES * * *'///)
C     LONGITUDIANAL CASE
      IF(ANS1.EQ.DIR) GO TO 102
      TYPE 90
90    FORMAT(/,'      VARIABLE     RANGE +/-'//,
     1' 1. PITCH RATE -      50 DEG/SEC'/,
     2' 2. AIRSPEED -        20 FT/SEC'/,
     3' 3. ANGLE OF ATTACK - 20 DEG '/,
     4' 4. PITCH ATTITUDE -  30 DEG'/,
     5' 5. PITCH RATE ACCEL. - 50 DEG/SEC**2'/,
     6' 6. LONGITUDINAL ACCEL. - .5 G '/,
     7' 7. NORMAL ACCEL. -         2 G '/,
     8' 8. ELEVATOR PSN. - 20 DEG '/,
     9' 9.   * * *   (BLANK)'//)
C
C     LATERAL DIRECTIONAL
C
102   CONTINUE
      IF(ANS1.EQ.LOG) GO TO 103
      TYPE 91
91    FORMAT(/' VARIABLE    RANGE +/-'//,
     1' 1. ROLL RATE - 25 DEG/SEC'/,
     2' 2. YAW RATE  - 25 DEG/SEC'/,
     3' 3. SIDESLIP ANGLE - 20 DEG'/,
     4' 4. BANK ANGLE      - 60 DEG'/,
     5' 5. ROLL RATE ACCEL. - 50 DEG/SEC**2'/,
     6' 6. YAW RATE ACCEL.  - 50 DEG/SEC**2'/,
     7' 7. LONGITUDINAL ACCEL. - .5 G'/,
     8' 8. AILERON DEFLECTION - 20 DEG'/,
     9' 9. RUDDER DEFLECTION  - 20 DEG'//)
C
C     - - - - - - - - - - - - - - - - - - - - - -
C
103   CONTINUE
C
      TYPE 11
11    FORMAT(' INDICATE VARIABLE NO. FOR TOP PLOT'/)
      ACCEPT *,KT
      TYPE 12
12    FORMAT(' INDICATE VARIABLE NO. FOR BOTTOM PLOT'/)
      ACCEPT *,KB
105   CONTINUE
      IF(ANS1.EQ.LOG) GO TO 411
      DO 421 I=1,9
      GAIN(I) = GAIN1(I)
421   CONTINUE
411   CONTINUE
      IF(ANS1.EQ.DIR) GO TO 402
      DO 422 I=1,9
      GAIN(I) = GAIN2(I)
422   CONTINUE
402   CONTINUE
C
C
```

```
C       - - - - - - - - - - - - - - - - - - - - -
C       CLEAR CRT AND FORM GRID FOR PLOTTING
C
        CALL INIT
        CALL PLOT55(2,1+2+32+64+128,,ISTAT)
        DO 110 K=1,235,50
        CALL PLOT55(4,1,K-1,ISTAT)
  110   CONTINUE
        CALL PLOT53(4,1,229,ISTAT)
        CALL PLOT55(5,0,1,ISTAT)
C
C       - - - - - - - - - - - - - - - - - - -
C
C       FORM THE MEASURED AND ESTIMATED RESPNSES
C
        DO 130 I=1,N
        IARRAY(2*I) = RDATA(I,KT)*GAIN(KT)+150
        IARRAY(2*I-1) = RDATA(I,KB)*GAIN(KB)+50
  130   CONTINUE
        CALL PLOT55(9,20,2,ISTAT)
        CALL PLOT55(12,,'* * *  TIME HISTORIES  * * * ',ISTAT)
        CALL PLOT55(9,50,4,ISTAT)
        CALL PLOT55(12,,' MEASURED DATA ',ISTAT)
        CALL GRAPH(2*N,IARRAY)
        DO 140 I=1,N
        IARRAY(2*I) = EDATA(I,KT)*GAIN(KT)+150
        IARRAY(2*I-1) = EDATA(I,KB)*GAIN(KB)+50
  140   CONTINUE
        CALL PLOT55(9,50,4,ISTAT)
        CALL PLOT55(12,,' ESTIMATED DATA',ISTAT)
        CALL GRAPH(2*N,IARRAY)
C       - - - - - - - - - - - - - - - - - - - -
C
        CALL PLOT55(9,50,4,ISTAT)
        CALL PLOT55(12,,'                     ',ISTAT)
        KFLAG = 1
C
C       - - - - - - - - - - - - - - - - - - - - - -
C
        IF(ANS1.EQ.DIR) GO TO 699
C
        KFLAG1 = 0
C
C       LONGITUDINAL LABELS
C
        CALL PLOT55(9,50,6,ISTAT)
        IF(KT.EQ.1) GO TO 601
        IF(KT.EQ.2) GO TO 602
        IF(KT.EQ.3) GO TO 603
        IF(KT.EQ.4) GO TO 604
        IF(KT.EQ.5) GO TO 605
        IF(KT.EQ.6) GO TO 606
        IF(KT.EQ.7) GO TO 607
        IF(KT.EQ.8) GO TO 608
C
  610   CONTINUE
        KFLAG1 = 1
C
        CALL PLOT55(9,50,16,ISTAT)
        IF(KB.EQ.1) GO TO 601
        IF(KB.EQ.2) GO TO 602
        IF(KB.EQ.3) GO TO 603
        IF(KB.EQ.4) GO TO 604
        IF(KB.EQ.5) GO TO 605
        IF(KB.EQ.6) GO TO 606
        IF(KB.EQ.7) GO TO 607
        IF(KB.EQ.8) GO TO 608
C
  601   CALL PLOT55(12,,' Q  +/- 25 DEG/SEC        ',ISTAT)
        GO TO 640
```

```
602   CALL PLOT55(12,,' V  +/- 20 FEET/SEC       ',ISTAT)
      GO TO 640
603   CALL PLOT55(12,,' ALPHA  +/- 20 DEG        ',ISTAT)
      GO TO 640
604   CALL PLOT55(12,,' THETA  +/- 30 DEG        ',ISTAT)
      GO TO 640
605   CALL PLOT55(12,,' Q DOT  +/- 50 DEG/SEC**2',ISTAT)
      GO TO 640
606   CALL PLOT55(12,,' AX  +/- .5 G            ',ISTAT)
      GO TO 640
607   CALL PLOT55(12,,' AN  +/- 2 G             ',ISTAT)
      GO TO 640
608   CALL PLOT55(12,,' DE  +/- 20 DEG          ',ISTAT)
C
640   CONTINUE
      IF(KFLAG1.EQ.0) GO TO 610
C
      GO TO 751
C
C
699   CONTINUE
C
C
C     LATERAL DIRECTIONAL LABELS
C
      KFLAG2 = 0
C
      CALL PLOT55(9,50,6,ISTAT)
      IF(KT.EQ.1) GO TO 701
      IF(KT.EQ.2) GO TO 702
      IF(KT.EQ.3) GO TO 703
      IF(KT.EQ.4) GO TO 704
      IF(KT.EQ.5) GO TO 705
      IF(KT.EQ.6) GO TO 706
      IF(KT.EQ.7) GO TO 707
      IF(KT.EQ.8) GO TO 708
      IF(KT.EQ.9) GO TO 709
C
710   CONTINUE
C
      KFLAG2 = 1
C
      CALL PLOT55(9,50,16,ISTAT)
      IF(KB.EQ.1) GO TO 701
      IF(KB.EQ.2) GO TO 702
      IF(KB.EQ.3) GO TO 703
      IF(KB.EQ.4) GO TO 704
      IF(KB.EQ.5) GO TO 705
      IF(KB.EQ.6) GO TO 706
      IF(KB.EQ.7) GO TO 707
      IF(KB.EQ.8) GO TO 708
      IF(KB.EQ.9) GO TO 709
C
C     - - - - - - - - - - - - - - -
C
701   CALL PLOT55(12,,' P  +/- 25 DEG/SEC          ',ISTAT)
      GO TO 740
702   CALL PLOT55(12,,' R  +/- 25 DEG/SEC          ',ISTAT)
      GO TO 740
703   CALL PLOT55(12,,' BETA  +/- 20 DEG           ',ISTAT)
      GO TO 740
704   CALL PLOT55(12,,' PHI  +/- 60 DEG            ',ISTAT)
      GO TO 740
705   CALL PLOT55(12,,' P DOT  +/- 50 DEG/SEC**2  ',ISTAT)
      GO TO 740
706   CALL PLOT55(12,,' R DOT  +/- 50 DEG/SEC**2  ',ISTAT)
      GO TO 740
707   CALL PLOT55(12,,' AY  +/- .5 G               ',ISTAT)
      GO TO 740
708   CALL PLOT55(12,,' DA  +/- 20 DEG             ',ISTAT)
      GO TO 740
709   CALL PLOT55(12,,' DR  +/- 20 DEG             ',ISTAT)
```

```
C
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C
  740 CONTINUE
      IF(KFLAG2.EQ.0) GO TO 710
C
C     LOGIC FOR GENERATING NEW PLOTS END TERMINATING
  751 CONTINUE
      READ(5,180) KR
  180 FORMAT(I2)
      CALL INIT
C
C
      TYPE 210
  210 FORMAT('         DO YOU WANT TO REPLACE TOP PLOT? (Y/N)'/)
      READ(5,220) ANS2
  220 FORMAT(A1)
      IF(ANS2.EQ.NO) KFLAG = 0
      IF(ANS2.EQ.NO) GO TO 230
C
      TYPE 100
      IF(ANS1.EQ.LOG)      TYPE 90
      IF(ANS1.EQ.DIR)      TYPE 91
      TYPE 240
  240 FORMAT('         INDICATE NEW VARIABLE NUMBER '/)
      ACCEPT *,KT
  230 CONTINUE
      TYPE 309
  309 FORMAT('         DO YOU WANT TO REPLACE BOTTOM PLOT? (Y/N)'/)
      READ(5,220) ANS2
      IF(ANS2.EQ.NO) GO TO 400
      TYPE 320
  320 FORMAT('         INDICATE NEW VARIABLE NUMBER'/)
      ACCEPT *,KB
      GO TO 105
  400 CONTINUE
      IF(KFLAG.EQ.0.) GO TO 410
      GO TO 105
  410 CONTINUE
C
C    - - - - - - - - - - - - - - - - - - - - - - -
C
      CALL PLOT55(2,512,1+2+4+32+64,ISTAT)
      CALL PLOT55(0,-1,0,ISTAT)
      RETURN
      END
```

```
      SUBROUTINE INIT
      COMMON/STATUS/ISTAT(16)
      DATA ISTAT/16*0/
      CALL PLOT55(13,72,,ISTAT)
      CALL PLOT55(13,74,,ISTAT)
      CALL PLOT55(2,1+512,,ISTAT)
      RETURN
      END
```

```
      SUBROUTINE GRAPH(N,IARRAY)
      COMMON/STATUS/ISTAT(16)
      DIMENSION IARRAY(512)
      NUMBER=ISTAT(8)/8
      CALL PLOT55(7,0,0,ISTAT)
      CALL PLOT55(8,512,0,ISTAT)
      CALL PLOT55(2,1+(NUMBER+1)*2,(NUMBER+1)*10,ISTAT)
      CALL PLOT55(3,-N,IARRAY,ISTAT)
      CALL PLOT55(1,1-NUMBER,,ISTAT)
      CALL PLOT55(9,10,1,ISTAT)
      END
```

A.9) CESSNA PROGRAMS

This appendix contains listings of the programs used in the Cessna spin test program.

A.9.1) DATA ACQUISITION

Description: This is an assembly language program for the AIM 65. The program is essentially the same as the one of Appendix A.1, the differences being

- no start and end data is taken;
- Channels 0-14 are sampled continuously every 0.1 secs when the "RUN/STBY" switch is on RUN;
- data is output to the TEAC tape drive every 0.5 secs.

Program listing:

```
;DATA ACQUISITION
;
RNCNT=0
BLKCNT=2
BUFCNT=4
IBUF=5
OBUF=7
CNT=9
BUF1=$200
BUF2=$300
KDDRA2=$A481
KDDRB2=$A483
KDRA2=$A480
KDRB2=$A482
DBR=$9008
WDC=$9009
CDR=$900A
MDRO=$900B
CSR=$900C
ESR=$900D
ISR=$900E
MDR1=$900F
WRT=$C1
WTM=$C2
ERA=$C3
SLE=$C9
REW=$CA
NRDY=$10
FPT=$04
DA=$20
DBRE=$40
CCE=$80
UDRB=$A000
UACR=$A00B
UIER=$A00E
UT1L=$A004
UT1CH=$A005
UT2L=$A008
UT2H=$A009
UIFR=$A00D
BIT5=$20
LOADK=$EF
RECK=$BF
CLOSEK=$DF
TIME1H=$27
TIME1L=$10
--------------------
*=$0400
LDA #$92
STA MDRO
LDA #1
STA RNCNT
LDA #0
STA RNCNT+1
LDA #$FF
STA KDDRA2
LDA #0
STA KDDRB2
STA KDRA2
LDA #$C0
STA UACR
START
LDA #$12
STA MDRO
LDA #REW
JSR COMD
LDA #REW
JSR COMD
MAIN
JSR GKEY
CMP #LOADK
BNE MAIN
LDA #REW
JSR COMD
LDA #SLE
JSR COMD
MAIN2
JSR GKEY
CMP #RECK
BEQ RECORD
CMP #CLOSEK
BNE MAIN2
CLOSE
LDA #WTM
JSR COMD
LDA #WTM
JSR COMD
LDX #12
CLOSE1
LDA #ERA
JSR COMD
DEX
BNE CLOSE1
JMP START
RECORD
LDA #0
STA IBUF
STA OBUF
LDA #>BUF1
STA IBUF+1
LDA #>BUF2
STA OBUF+1
LDA #>INT
STA $A405
LDA #<INT
STA $A404
LDA #$C0
STA UIER
LDA #<$C34E
STA UT1L
LDA #>C34E
STA UT1CH
CLI
LDA #0
STA BLKCNT
STA BLKCNT+1
REC1
JSR SWAP
JSR WRITE
REC2
JSR GKEY
CMP #RECK
BNE RECX
LDA BUFCNT
CMP #150
BNE REC2
INC BLKCNT
BNE REC1
INC BLKCNT+1
JMP REC1
RECX
LDA BUFCNT
CMP #150
BNE RECX
LDA #$40
```

```
STA UIER
JSR SWAP
LDA #$FF
STA BLKCNT
STA BLKCNT+1
JSR WRITE
INC RNCNT
BNE RECX2
INC RNCNT+1
RECX2
JMP MAIN2
--------------------
GKEY
LDA KDRB2
PHA
LDA #TIME1L
STA UT2L
LDA #TIME1H
STA UT2H
GKEY1
LDA UIFR
AND #BIT5
BEQ GKEY1
PLA
CMP KDRB2
BNE GKEY
RTS
--------------------
COMD
PHA
LDA ESR
COMD1
LDA CSR
AND #NRDY
BNE COMD1
LDA CSR
AND #FPT
BNE COMD1
PLA
STA CDR
COMD2
LDA ISR
AND #CCE
BEQ COMD2
RTS
```

```
--------------------
WAIT
JSR WAITX
WAITX
RTS
--------------------
WRITE
LDA ESR
LDA #154
STA WDC
LDA #WRT
STA CDR
LDA RNCNT
JSR WWORD
LDA RNCNT+1
JSR WWORD
LDA BLKCNT
JSR WWORD
LDA BLKCNT+1
JSR WWORD
LDY #0
WRITE1
LDA (OBUF),Y
JSR WWORD
INY
CPY #150
BNE WRITE1
WRITE2
JMP COMD2
--------------------
SWAP
LDA OBUF+1
PHA
LDA IBUF+1
STA OBUF+1
PLA
STA IBUF+1
LDA #0
STA BUFCNT
RTS
--------------------
```

```
--------------------
WWORD
PHA
WWORD1
LDA ISR
AND #DBRE
BEQ WWORD1
PLA
STA DBR
RTS
--------------------
INT
PHA
LDA UDRB
BPL INTEX
TYA
PHA
LDY BUFCNT
LDA #15
STA CNT
LDA $8000
JSR WAIT
ILOOP
LDA $8002
STA (IBUF),Y
LDA $8001
INY
LDA $8003
STA (IBUF),Y
INY
DEC CNT
BNE ILOOP
STY BUFCNT
PLA
TAY
INTEX
LDA UT1L
PLA
RTI
 END
```

A.9.2) DATA READBACK

Description: This program is used to read the data off the AIM 65 system's tape drive to be sent to the ground-based system. The program is similar to the one of Appendix A.2, the differences being

- no error checking is done by the AIM 65;
- all 12 BITS of recorded data are transferred just as they have been recorded.

Program listing:

```
                ;DATA RECOVERY
RNCNT=0
BLKCNT=2
VRUN=4
VBLK=5
CNT=6
BUF1=$200
DBR=$9008
WDC=$9009
CDR=$900A
MDRO=$900B
CSR=$900C
ESR=$900D
ISR=$900E
MDR1=$900F
SLP=$C8
RDL=$C4
REW=$CA
NRDY=$10
TDRE=$02
CR=$0D
SCR=$9006
SDR=$9007
LOADC=$4C
READC=$52
CLOSEC=$43
INALL=$E993
NUMA=$EA46
READM=$E93C
OUTALL=$E9BC
OUTPUT=$E97A
UTIL=$A004
UTICH=$A005
UACR=$A00B
----------------
*=$300
CCE
.BYTE $80
DA
.BYTE $20
MO
.BYTE CR,'TAPE ERROR',$A0
MRUN
.BYTE CR,'WHICH RUN NUMBER',$BF
MBLK
.BYTE CR,'HOW MANY BLOCKS',$BF
MEND
.BYTE CR,'LAST BLOCK THIS RU',$CE
MINV
.BYTE CR,'INVALID COMMAN',$C4
MERR1
.BYTE CR,'FILE MARK FOUN',$C4
MRNCNT
.BYTE CR,'RUN NUMBER',$A0
-------------
*=$400
RESETB
LDA #$92
STA MDRO
LDA #$C0
STA UACR
LDA #$68      ;$68=300BAUD
              ;$34=600
              ;$1A=1200
              ;$0D=2400
```

```
STA UTIL
LDA #0

STA UTICH
LDA #$11
EOR #$FF
STA SCR
                 ;
MAIN
JSR GCOM
CMP #LOADC
BEQ MAIN2
JSR INVAL
JMP MAIN
MAIN2
LDA #$12
STA MDRO
LDA #REW
JSR COMDA
LDA #SLP
JSR COMDA
MAIN3
JSR GCOM
CMP #READC
BEQ READ
CMP CLOSEC
BEQ CLOSE
JSR INVAL
JMP MAIN3

                 ;
CLOSE
LDA #REW
JSR COMDA
LDA #REW
JSR COMDA
JMP MAIN
                 ;
READ
LDY #MRUN-MO
JSR MESS
JSR GCNT
CMP #0
BEQ CLOSE
STA VRUN
READ1
JSR RBLK
BCS CLOSE
LDA BLKCNT
ORA BLKCNT+1
BNE READ3
LDY #MRNCNT-MO
JSR MESS

LDA RNCNT
JSR NUMA
READ3
LDA RNCNT
CMP VRUN
BCC READ1
BEQ READ2
LDA #REW
JSR COMDA
LDA #SLP
JSR COMDA
```

```
JMP READ1
READ2
LDY #MBLK-MO
JSR MESS
JSR GCNT
CMP #0
BEQ CLOSE
STA VBLK
                    ;CHANGE TO NOPS
                    ;TO TRANSMIT COUNTS
SENDB
JMP SENDB1
LDA RNCNT
JSR SEND
LDA RNCNT+1
JSR SEND
LDA BLKCNT
JSR SEND

LDA BLKCNT+1
JSR SEND
SENDB1
LDY #0
SENDB2
LDA BUF1,Y
JSR SEND
INY
CPY #160
BNE SENDB2
LDA BLKCNT
CMP #$FF
BNE SENDB3

CMP BLKCNT+1
BEQ END
SENDB3
DEC VBLK
BNE SENDB5
SENDB4
LDY #MBLK-MO
JSR MESS
JSR GCNT
CMP #0
BEQ CLOSE1
STA VBLK
SENDB5
JSR RBLK
BCC SENDB
CLOSE1
JMP CLOSE
END
LDY #MEND-MO
JSR MESS
JMP MAIN3
```

```
--------------
MESS
LDA MO,Y
PHA
AND #$7F
JSR OUTPUT
INY
PLA
BPL MESS
RTS
--------------
GCOM
JSR READM
JSR OUTALL
RTS
--------------
INVAL
LDY #MINV-MO
JSR MESS
RTS
--------------
GCNT
LDA #0
STA CNT
GCNT1
JSR INALL
JSR DPACK
BCC GCNT1
LDA CNT
RTS
--------------
DPACK
CMP #'0'
BCC RSPAC
CMP #$3A
BCS RSPAC
AND #$0F
PHA
LDA CNT
ASL A
ASL A
CLC

ADC CNT
ASL A
STA CNT
PLA
CLC
ADC CNT
STA CNT
CLC
RTS
RSPAC
SEC
RTS
```

```
--------------
SEND
PHA
SEND1
LDA SCR
AND #TDRE
BNE SEND1
PLA
EOR #$FF
STA SDR
RTS
--------------
RBLK
LDA CSR
AND #NRDY
BNE RBLK
LDA #164
STA WDC
LDA #RDL
STA CDR

JSR RWORD
BCS RBLK2
STA RNCNT
JSR RWORD
BCS RBLK2
STA RNCNT+1
JSR RWORD

BCS RBLK2
STA BLKCNT
JSR RWORD
BCS RBLK2
STA BLKCNT+1
LDY #0
RBLK1
JSR RWORD
BCS RBLK2
STA BUF1,Y
INY
CPY #160
BNE RBLK1
JMP COMDA2
RBLK2
JMP COMDA4
```

```
--------------
COMDA
PHA
LDA ESR
COMDA1
LDA CSR
AND #NRDY
BNE COMDA1
PLA
STA CDR
COMDA2
LDA ISR
AND CCE
BEQ COMDA2
COMDA4
LDA CSR
PHA
AND #2
BEQ COMDA5
LDY #MERR1-MO
JSR MESS
PLA
SEC
RTS
COMDA5
PLA
AND #$81
BNE COMDA3
CLC
RTS
COMDA3
LDY #MO-MO
PHA
JSR MESS
PLA
JSR NUMA
LDA ESR
JSR NUMA

CLC
RTS
--------------
RWORD
LDA ISR
BIT CCE
BNE RWORD2
BIT DA
BEQ RWORD
LDA DBR

CLC
RTS
RWORD2
SEC
RTS
 END
```

A.9.3) DATA RECEIVE

Description: This program is written on the Hewlett Packard 9825 of Cessna Aircraft Company. The program receives data from its RS232 port. The program is the same as the one of Appendix A.3 of Reference 2.

Program listing:

```
0: "COMP. TERMINAL WITH CONTROL KEYS AND AUTO FUNCTIONS trk0;file5":
1: fmt 1,c1,z;1→I
2: sfg 2
3: 11→Q
4: wtc Q,1
5: wtb Q,37
6: 0→J;1→K
7: wtc Q,0
8: dim L$[106],C[0:255],D$[300,30],A$[80],Q$[100]
9: if H=1;""→L$
10: gsb "string"
11: oni Q,"in"
12: eir Q,4
13: buf "in",L$,1
14: tfr Q,"in",1
15: on key "KEY"
16: "15":if flg7;trk 0;cfg 7;sfg 6
17: if flg6;for L=1 to 203;for N=1 to 70;num(D$[L,N,N])→P;wtb Q,P
18: if flg6;char(P)→A$[Z+1→Z,Z]
19: if flg6;dsp A$[max(1,len(A$)-31),max(32,len(A$))];next N;0→Z;wtb Q,13
20: if flg6;next L;cfg 6;sfg 7
21: gto 16
22: "KEY":key→C;if C=0;kret
23: if C=66 or C=194;sfg 9;0→C;kret
24: if flg9 and C[C]>64 and C[C]<90;wtb Q,C[C]-64;cfg 9;kret
25: if flg9;dsp "NOT A VALID CONTROL KEY",C[C];cfg 9;0→C;kret
26: if Z=79;13→C
27: if C[C]=1008;sfg 7;0→C;kret
28: if C[C]=1001;sfg 2;0→C;0→J;1→K;kret
29: if C[C]=1000;cfg 2;0→C;kret
30: wtb Q,C[C]→P;char(P)→A$[Z+1→Z,Z]
31: dsp A$[max(1,len(A$)-31),max(32,len(A$))]
32: if C[C]=13;" "→A$;0→Z;dsp A$
33: kret
34: "in":
35: if flg2;L$[1,1]→Q$[K,K];K+1→K;dsp (J+1)/5
36: if flg2 and K=31;Q$[1,30]→D$[J+1];J+1→J;1→K;" "→Q$
37: buf "in"
38: if C=7 or C=135;gsb "break"
39: eir Q,4
40: tfr Q,"in",1
41: iret
42: "string":
43: for I=1 to 58
44: I→C[I]
45: next I
46: for I=78 to 87
47: I-30→C[I]
48: next I
49: for I=88 to 96
50: 32→C[I]
51: next I
52: for I=97 to 122
53: I→C[I]
54: next I
55: for I=123 to 175
```

```
56: 32→C[I]
57: next I
58: for I=176 to 185
59: I-144→C[I]
60: next I
61: for I=186 to 224
62: 32→C[I]
63: next I
64: for I=225 to 250
65: num(char(I-160))→C[I]
66: next I
67: 13→C[141];10→C[138]
68: for I=206 to 216
69: I-158→C[I]
70: next I
71: 60→C[172];123→C[123];94→C[94];32→C[7]
72: 47→C[47]→C[175];40→C[40]→C[168];41→C[41]→C[169];92→C[222];62→C[174]
73: 101→C[96]→C[224];125→C[125]→C[253];43→C[43]→C[171];45→C[45]→C[173]
74: 27→C[65]→C[193];63→C[63]
75: 64→C[183];91→C[184];93→C[185]
76: 39→C[176];8→C[20]→C[148];61→C[61]→C[189]
77: 59→C[59]→C[187];58→C[191];46→C[88]→C[216];7→C[7]→C[135]
78: 1001→C[76]→C[220];1000→C[75]→C[219];1010→C[74]→C[218]
79: 1009→C[73]→C[218];1008→C[72]→C[217]
80: 44→C[217]→C[89]
81: ret
82: "break":buf "in";eir Q,4
83: wtc Q,1;wtb Q,8;wait 200;wtb Q,37;wtc Q,0
84: wtc Q,1;wtb Q,37;wtc Q,0
85: ret
86: end
87: for I=1 to 5
88: wrt 706,D$[I,1,30],I
89: next I
90: for I=1 to 200
91: dsp D$[I];wait 50;next I;stp
*3167
```

A.9.4) DATA PLOTTING

Description: This program is used by the Hewlett Packard 9825 computer of Cessna Aircraft Company to convert and plot out the flight test results. (Sample plots are presented as Figures 7.6.)

Program listing:

```
0: "FNG UNITS CON (quick look data plt) trk1;file 6":gto "START";0→Y→Z
1: "CON":Z+1→Z
2: for K=1 to 29 by 2;num(D$[I,K])→U;num(D$[I,K+1])→V
3: (K-1)/2+1→H
4: shf(U,-4)→U;band(V,15)→V;ior(U,V)→U
5: if bit(11,U);ior(U,-4096)→U
6: .06528-.002427983U→M[H]
7: next K;ret
8: "START":706→R
9: dim F[2],D$[300,30],M[15]
10: dim A[15,50],A$[20]
11: ent "FIRST POSTION OF DYNAMIC DATA?",r5,"LAST POSITION OF DYNAMIC",r6
12: 10r5/2+1→F[1];10r6/2→F[2];fxd 1
13: ent "TRK # TEMP DATA ",T;if flg13;gto +4
14: ent "FILE #?",F
15: dsp "tape CONTINUE";stp
16: trk T;ldf F,D$
17: "C":0→D→r8→B;cfg 1,2
```

```
18: ent "START TIMF X AXIS",r0;ent "CONTINUE AT TIME",r8
19: if ((F[2]-F[1])/10+r8→r11)>30;dsp "TOO BIG DYNAMIC",r11;wait 7000;gto 11
20: dsp "Mount B size paper and CONTINUE";stp
21: fmt ;wrt 705,"OP";red 705,r2,r3,r4,r5;r4-r2→r2;r5-r3→r3
22: if r2>15650 or r2<15550 or r3>9650 or r3<9550;beep;dsp "check P1-P2";stp
23: psc 705
24: if F[2]-F[1]>50;F[2]→D;F[1]+49→F[2]
25: gto +2
26: if F[2]>D;D→F[2]
27: for I=1 to 15;for J=1 to 50;0→A[I,J];next J;next I
28: fmt 1,z,f6.1;fmt 2,2f4.1,f6.0,f6.1
29: fmt 3,"  AL-R BTA-L  AL-L      q      r      P     da     dr     de",z
30: fmt 4," TAB-L TAB-R Az Ay     Hp BTA-R"
31: "wrt R+.3;wrt R+.4":
32: for I=F[1] to F[2];I-F[1]+1→r1
33: gsb "CON"
34: "cal Alpha - RH":-.9656+22.5313M[1]→A[1,r1]
35: "cal Beta - LH":.4759-12.03M[2]→A[2,r1]
36: "cal Alpha - LH":-1.377-25.7185M[3]→A[3,r1]
37: "cal q":10.045+72.0222M[4]→A[4,r1]
38: "cal r":1.8175-39.9680M[5]→A[5,r1]
39: "cal P":-3.0776-57.2165M[6]→A[6,r1]
40: "cal da":1.5759+4.13373M[7]-.1511M[7]→A[7,r1]
41: "cal dr":-.45125+10.8905M[8]→A[8,r1]
42: "cal de":-7.12932+7.35987M[9]→A[9,r1]
43: "cal TAS - LH":-2.31826+115.913M[10]→A[10,r1]
44: "cal TAS - RH":-.95755+119.69424abs(M[11])→A[11,r1]
45: if M[11]<0 and not flg1;.1r1+r8+r0→B;sfg 2
46: "cal Az":10M[12]→A[12,r1]
47: "cal Ay":10M[13]→A[13,r1]
48: "cal Hp":-1000+5000M[14]→A[14,r1]
49: "cal Beta - RH":-1.0342+12.4306M[15]→A[15,r1]
50: "for J=1 to 11;wrt R+.1,A[J,r1];next J":
51: "wrt R+.2,A[12,r1],A[13,r1],A[14,r1],A[15,r1]":
52: if flg2;sfg 1
53: dsp I/10;next I
54: pen
55: for I=1 to 15;jmp I
56: scl -4,72,-500,600;gto +15
57: scl -48,30,-250,230;gto +14
58: scl -4,72,-480,480;gto +13
59: scl -48,30,-2200,200;gto +12
60: scl -4,72,-3800,1000;gto +11
61: scl -4,72,-4400,400;gto +10
62: scl -4,72,-20,460;gto +9
63: scl -4,72,-140,340;gto +8
64: scl -4,72,-80,400;gto +7
65: scl -48,30,-660,300;gto +6
66: scl -48,30,-520,440;gto +5
67: scl -4,72,-30,18;gto +4
68: scl -48,30,-7.5,16.5;gto +3
69: scl -48,30,-2000,46000;gto +2
70: scl -48,30,-200,280;gto +1
71: 50→Z;for J=50 to 1 by -1;dsp A[I,J],J
72: if A[I,J]=0;J-1→Z;next J
73: pen# 2
74: for J=1 to Z;plt .1J+r0+r8→r6,A[I,J]
75: if r6=B;pen# 3;wrt 705,"UC99,0,8,0,-16,0,8,-99";pen;cplt -1,0;pen# 2
76: next J;pen;dsp J
77: next I
78: pen#
79: if D>0;F[2]+1→F[1];if D>F[2];F[1]+49→F[2];5+r8→r8;gto 26
80: 0→Y→Z;gto 11
*22683
```

APPENDIX B

TRANSFORMATION OF AXES SYSTEMS

This appendix shows the correlation between several axes systems.

Much information contained in this section is taken directly from Reference 28, which deals in depth with the problem of the different axes systems used in airplane analysis.

PRECEDING PAGE BLANK NOT FILMED

There are primarily five axes systems used in airplane analysis. These are described here.

1) Body Axes

"The orthogonal body-axes system is fixed within the vehicle with the X-axis along the longitudinal center line of the body, the Y-axis normal to the plane of symmetry, and the Z-axis in the plane of symmetry. This is the axes system about which aircraft instruments are usually mounted. Its main advantage in motion calculations is that vehicle moments of inertia about the axes are constant, so that the $\dot{I}$ terms can be omitted from the equations of motion. It is the logical system to which to refer velocities, accelerations, and stability and control parameters in the study of aircraft handling qualities because the pilot's orientation with respect to this frame is fixed."* *(This is the axes system used in this report.)*

2) Principal Axes

"The principal axes are an orthogonal body-fixed system for which the products of inertia are zero. The X and Z principal axes lie in the plane of symmetry; the angle between the X body axis and the X principal axes is usually small so that in many cases the body axes can be assumed to coincide with the principal axes."*

---

*From Reference 28.

3) Flight Stability Axes

"The flight stability axes (sometimes referred to as vehicle stability axes) are an orthogonal body-axes system fixed to the vehicle, the X-axes of which is alined with the relative wind vector when the vehicle is in a steady-state trim condition but then rotates with the vehicle after a disturbance as the vehicle changes angle of attack. This system is preferred in many stability studies because, as with other body-fixed axes, the moments of inertia about the axes remain constant and also because the motions defined are primarily those about the flight path rather than about body reference lines."*
(This is the axes system used in Reference 22.)

4) Wind-Tunnel Stability Axes

"The wind-tunnel stability axes are the system about which most wind-tunnel data are obtained. For this system the X-axis is in the same horizontal plane as the relative wind at all times . . . . The angle α between the X-axis of this system and the X-body axes is variable. (It is a constant α_0 for the flight stability axes.) This means that vehicle moments of inertia about the X-axis change. It also means that additional terms are required in the transformation equations for static-stability derivatives and for u,v,w derivatives when data are transferred to or from the wind axes or the wind-tunnel stability axes."*

---

*From Reference 28.

C-3

5) Wind Axes

"The wind axes are the system generally used in calculating motions of the vehicle as a point mass. The X-axis for this system is alined with the relative wind at all times so that vehicle moments of inertia about this axis change. As with the wind-tunnel stability axes, additional terms . . . are required in the transformation to or from the wind axes and either the body, principal, or flight stability axes, since the angle . . . between the X wind axis and the X-axis of either of these systems is variable. Also, since the lateral angle . . . between the X-axes is variable, there are additional terms . . . required in the transformations for some of the lateral derivatives between the wind axes and either of the other axes systems."*

The correlation between these axes systems is perhaps best summarized by Table B.1.

Table B.1 Designation of Force and Moment Coefficients for Different Axes Systems*

Component	Coefficients for axes system –			
	Body or principal	Flight stability	Wind-tunnel stability	Wind
X-axis force	C_X or $-C_A$	$C_{X,s}$	$-C_D$	$-C_D$
Y-axis force	C_Y	$C_{Y,s}$	C_Y	C_C
Z-axis force	C_Z or $-C_N$	$C_{Z,s}$	$-C_L$	$-C_L$
X-axis moment (roll)	C_{ℓ}	$C_{\ell,s}$	$C_{\ell,wt}$	$C_{\ell,w}$
Y-axis moment (pitch)	C_m	$C_{m,s}$	$C_{m,wt}$	$C_{m,w}$
Z-axis moment (yaw)	C_n	$C_{n,s}$	$C_{n,wt}$	$C_{n,w}$

---

*From Reference 28.

Transformation from the flight stability axes (as used in Reference 22) to the body axes used in this report involves accounting for the steady-state angle of attack (α_1). The following equation takes care of this by correcting the inertias. This is the only change required.

$$\begin{bmatrix} I_{xx,s} \\ I_{zz,s} \\ I_{xz,s} \\ I_{yy,s} \end{bmatrix} = \begin{bmatrix} \cos^2\alpha_1 & \sin^2\alpha_1 & (-)\sin^2\alpha_1 & 0 \\ \sin^2\alpha_1 & \cos^2\alpha_1 & \sin^2\alpha_1 & 0 \\ \frac{1}{2}\sin^2\alpha_1 & (-)\frac{1}{2}\sin^2\alpha_1 & \cos^2\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{xx} \\ I_{zz} \\ I_{xz} \\ I_{yy} \end{bmatrix} \qquad [B.1]$$

NOTE: "s" denotes stability axes; no subscript denotes body axes.

NASA Langley (Reference 30) and NASA Dryden (References 12-16) both use the body axes system. They both, however, use different designations. NASA Langley uses the X, Y, Z, ℓ, m, n designation; NASA Dryden, the A, Y, N, ℓ, m, n designation. The parameters will be presented in the X, Y, Z, ℓ, m, n system in this report. Table B.2 shows the correlation between both these systems.

The symbols (i.e., Z_α', etc.) in the definition column of Table B.2 are those as predicted by the MMLE "BONES" program. For conversion from normal stability parameters (as per Reference 22) to these state vector derivatives, the reader is referred back to Tables 5.2 and 5.3.

For rigorous conversion between the various axes systems, the reader is referred to Reference 28.

Table B.2 Comparison of Non-Dimensional Derivatives

LONGITUDINAL

KU-FRL NASA Langley designation	NASA Dryden designation	DEFINITION
C_{Z_α}'	$-C_{N_\alpha}'$	$\dfrac{Z_\alpha' \, mU_1}{\bar{q}_1 \, S}$
C_{Z_u}'	$-C_{N_u}'$	$\dfrac{Z_u' \, mU_1}{\bar{q}_1 \, S}$
$C_{Z_{\delta_{E,c}}}'$	$-C_{N_{\delta_{E,c}}}'$	$\dfrac{Z_{\delta_{E,c}}' \, mU_1}{\bar{q}_1 \, S}$
C_{Z_o}'	$-C_{N_o}'$	$\dfrac{Z_o' \, mU_1}{\bar{q}_1 \, S}$
C_{m_α}'	C_{m_α}'	$\dfrac{M_\alpha' \, I_{yy}}{\bar{q}_1 \, S\bar{c}}$
C_{m_q}'	C_{m_q}'	$\dfrac{M_q' \, 2U_1 \, I_{yy}}{\bar{q}_1 \, S\bar{c}^2}$
C_{m_u}'	C_{m_u}'	$\dfrac{M_u' \, U_1 \, I_{yy}}{\bar{q}_1 \, S\bar{c}}$
$C_{m_{\delta_{E,c}}}'$	$C_{m_{\delta_{E,c}}}'$	$\dfrac{M_{\delta_{E,c}}' \, I_{yy}}{\bar{q}_1 \, S\bar{c}}$
C_{m_o}'	C_{m_o}'	$\dfrac{M_o' \, I_{yy}}{\bar{q}_1 \, S\bar{c}}$
C_{X_α}'	$-C_{A_\alpha}'$	$\dfrac{X_\alpha' \, m}{\bar{q}_1 \, S}$
C_{X_u}'	$-C_{A_u}'$	$\dfrac{X_u' \, mU_1}{\bar{q}_1 \, S}$
$C_{X_{\delta_{E,c}}}'$	$-C_{A_{\delta_{E,c}}}'$	$\dfrac{X_{\delta_{E,c}}' \, m}{\bar{q}_1 \, S}$
C_{X_o}'	$-C_{A_o}'$	$\dfrac{X_o' \, m}{\bar{q}_1 \, S}$

LATERAL

KU-FRL NASA-Langley/-Dryden designation	Definition
C_{ℓ_p}'	$\dfrac{L_p' \, 2I_{xx} \, U_1}{\bar{q}_1 \, Sb^2}$
C_{ℓ_r}'	$\dfrac{L_r' \, 2I_{xx} \, U_1}{\bar{q}_1 \, Sb^2}$
C_{ℓ_β}'	$\dfrac{L_\beta' \, I_{xx}}{\bar{q}_1 \, Sb}$
$C_{\ell_{\delta_A}}'$	$\dfrac{L_{\delta_A}' \, I_{xx}}{\bar{q}_1 \, Sb}$
$C_{\ell_{\delta_R}}'$	$\dfrac{L_{\delta_R}' \, I_{xx}}{\bar{q}_1 \, Sb}$
C_{y_β}'	$\dfrac{Y_\beta' \, mU_1}{\bar{q}_1 \, S}$
$C_{y_{\delta_A}}'$	$\dfrac{Y_{\delta_A}' \, mU_1}{\bar{q}_1 \, S}$
$C_{y_{\delta_R}}'$	$\dfrac{Y_{\delta_R}' \, mU_1}{\bar{q}_1 \, S}$
C_{n_p}'	$\dfrac{N_p' \, 2U_1 \, I_{zz}}{\bar{q}_1 \, Sb^2}$
C_{n_r}'	$\dfrac{N_r' \, 2U_1 \, I_{zz}}{\bar{q}_1 \, Sb^2}$
C_{n_β}'	$\dfrac{N_\beta' \, I_{zz}}{\bar{q}_1 \, Sb}$
$C_{n_{\delta_A}}'$	$\dfrac{N_{\delta_A}' \, I_{zz}}{\bar{q}_1 \, Sb}$
$C_{n_{\delta_R}}'$	$\dfrac{N_{\delta_R}' \, I_{zz}}{\bar{q}_1 \, Sb}$