AD NUMBER

AD869887

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; OCT 1969. Other requests shall be referred to Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH 45433.

AUTHORITY

AFFDL ltr, 7 Jan 1975

AFFDL-TR-69-93

ANGLE-OF-ATTACK COMPUTATION STUDY

J.B. DENDY
K.G. TRANSIER
SPERRY FLIGHT SYSTEMS DIVISION
SPERRY RAND CORPORATION

TECHNICAL REPORT AFFDL-TR-69-93
OCTOBER 1969

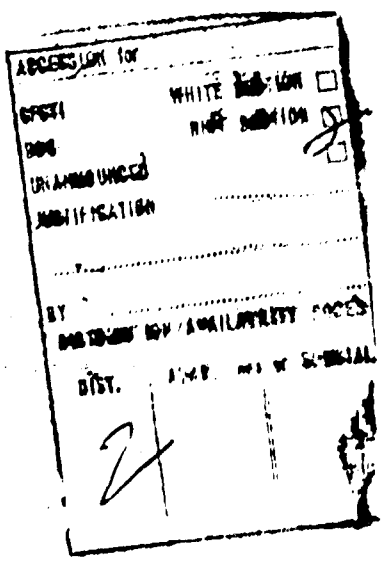
AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the AF Flight Dynamics Laboratory (FDCL), Wright-Patterson AFB, Ohio 45433.

The distribution of this report is limited because the report concerns technology identifiable with the DOD strategic embargo list.



Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AFFDL-TR-69-93

ANGLE-OF-ATTACK COMPUTATION STUDY

J.B. DENDY
K.G. TRANSIER
SPERRY FLIGHT SYSTEMS DIVISION
SPERRY RAND CORPORATION

TECHNICAL REPORT AFFDL-TR-69-93
OCTOBER 1969

FOREWORD

This report was prepared by Sperry Flight Systems Division, Sperry Rand Corporation, Phoenix, Arizona, on U.S. Air Force Contract F33615-69-C-1178. The work was administered under the direction of the Control Elements Branch of the Flight Controls Division, Air Force Flight Dynamics Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, by John E. Houtz, Project Engineer. Work on the contract was performed between 1 December 1968 and 31 July 1969. The manuscript was submitted by the authors in August 1969.

Messrs. J. B. Dendy and K. G. Transier were the principal investigators. Assistance was received from Messrs. L. G. Hofmann and I. L. Ashkenas of Systems Technology, Inc in the area of equation derivations.

This report concludes the work on the contract. The Sperry Flight Systems Division report number is 71-00004-00-00, and has been reviewed and approved by Sperry Flight Systems Division.

This technical report has been reviewed and is approved.

H. W. Basham
Chief, Controls Elements Branch
Flight Controls Division
AF Flight Dynamics Laboratory

ABSTRACT

This report discusses methods of computing angle of attack by inference using combinations of data from presently available, on-board sensors, thereby eliminating the need for external vanes or probes. Equations were derived from which computed angle of attack could be extracted. Those equations that were impractical from a mechanization standpoint were eliminated, leaving three candidate methods. These three methods were then analyzed with respect to errors arising from mathematical simplifications and errors due to imperfect sensor information. Two of the candidate methods provide inertial angle of attack, and will provide acceptable accuracy for low-performance aircraft applications. The third method will provide a high-quality, air-mass-related angle of attack. A mechanization of a high-quality angle of attack system using a small, special purpose digital computer is described. Using measurements of normal acceleration, longitudinal acceleration, elevator position, flap position, throttle position, airspeed, Mach, and dynamic pressure, the system will provide a high-quality angle of attack measurement applicable to any high-performance aircraft and competitive with current vane and probe transducers.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont)

SECTION I
INTRODUCTION

True airframe angle of attack finds many applications as an input to the sophisticated systems on-board the latest generation of aircraft. It has been used for weapons guidance computation, autopilot functions, speed command, stall warning, and gust alleviation. This multitude of uses requires an accurate, reliable measurement of angle of attack (α). Unfortunately, the techniques for measuring α have not progressed far beyond the Wright brothers' vane transducer used in the first powered flights.

Externally mounted angle-of-attack transducers have had a rather poor history of reliability and accuracy. To be effective, the device must be mounted in the free airstream to minimize fuselage flow effects. This necessarily exposes the transducer to all the severe environmental conditions experienced by the aircraft. Human error has also made a considerable contribution to α transducer unreliability. Most mounting positions have made the transducer a convenient handle or step for ground crews and pilots.

The inaccuracies exhibited by vanes and probes can be attributed to the extreme range of environment over which operation is required and to static errors due to the mounting position. Sophisticated approaches to solving these problems have produced some useful results, but at considerable expense.

The major drawbacks of externally mounted transducers can be overcome by computing angle of attack from combinations of internally mounted sensors. The concept of computed α is not a new one - it had been proposed as early as 20 years ago. However, it has never received a thorough evaluation, particularly in light of the highly improved computing techniques now available.

This technical report presents the results of a study of computed α and the mechanization thereof. The principal objectives of the study are to:

- Derive equations that can be used to compute angle of attack from existing, internally mounted sensors (Section II)
- Select those methods (equations) which appear to be practical and determine the errors that occur when simplifying assumptions are made (Section III)
- Determine the effect of sensor errors on the α computation when using the practical equations (Section IV)
- Determine the mechanization of an optimum system for computing angle of attack, using one or more of the methods generated and refined under the first three objectives (Section V). The system should maintain 0.1-degree accuracy through ± 75 degrees of pitch and roll, ± 10 degrees of side-slip, and +8g and -5g of acceleration.

SECTION II

DERIVATION OF EQUATIONS

1. BASIC DEFINITIONS

Before proceeding to the derivations, an explanation of axis systems would be expedient. Aircraft equations of motion are given in terms of a body-fixed axis system with its origin at the aircraft center of mass. The x and z axes in the plane of symmetry are oriented such that the Euler angles locating the body-fixed frame are those angles indicated by the vertical gyro, θ and ϕ, shown in Figure 1.

From Reference 1*:

$$Ax_{cg} = \dot{U} + QW - RV + g \sin\theta \tag{1}$$

$$Ax_{cg} = X(\alpha) + X(\beta) + X_{\delta_T}\,\delta_T \tag{2}$$

$$Ay_{cg} = \dot{V} + RU - PW - g \cos\theta \sin\phi \tag{3}$$

$$Ay_{cg} = Y_\beta\,\beta + Y_{\delta_r}\,\delta_r \tag{4}$$

$$Az_{cg} = \dot{W} + PV - QU - g \cos\theta \cos\phi \tag{5}$$

$$Az_{cg} = Z(\alpha) + Z_{\delta_e}\,\delta_e + Z_{\delta_T}\,\delta_T \tag{6}$$

$$\dot{P} - \frac{I_{xz}}{I_x}\dot{R} + \frac{I_z - I_y}{I_x}QR - \frac{I_{xz}}{I_x}PQ = L_\beta\,\beta + L_p\,P + L_r\,R + L_{\delta_r}\,\delta_r \tag{7}$$

$$\dot{Q} + \frac{I_x - I_z}{I_y}PR + \frac{I_{xz}}{I_y}(P^2 - R^2) = M_{\dot{\alpha}}\,\dot{\alpha} + M_q\,Q + M_{\delta_e}\,\delta_e + M_{\delta_T}\,\delta_T \tag{8}$$

$$\dot{R} - \frac{I_{xz}}{I_z}\dot{P} + \frac{I_y - I_x}{I_z}PQ + \frac{I_{xz}}{I_z}QR = N(\beta) + N_r\,R + N_p\,P + N_{\delta_r}\,\delta_r + N_{\delta_a}\,\delta_a \tag{9}$$

Stability derivatives with respect to speed are omitted because this effect is accounted for through use of the varying dynamic pressure implicit in the dimensional stability derivatives. Details of the aircraft simulation are presented in the Appendix.

---

*See list of references.

Care must be taken to distinguish between velocities determined with respect to an inertial frame and air that is moving with respect to an inertial reference frame. The former quantity is denoted by the velocity components U, V, W, in the body axis directions; while U_a, V_a, and W_a will denote corresponding velocity components with respect to the (moving) air mass. These are related to the gust velocity components u_g, v_g, and w_g seen by a body-axis observer as follows:

$$U - u_g = U_a \tag{10}$$

$$V - v_g = V_a \tag{11}$$

$$W - w_g = W_a \tag{12}$$

The total velocity with respect to the air mass is

$$\overline{V}_a = \left(U_a^2 + V_a^2 + W_a^2\right)^{1/2} \tag{13}$$

The true aerodynamic angle of attack, α, and angle of side-slip, β, are defined by

$$\alpha = \tan^{-1} \frac{W_a}{U_a} \tag{14}$$

$$\beta = \sin^{-1} \frac{V_a}{\overline{V}_a} \tag{15}$$

Other useful relations are

$$U_a = \overline{V}_a \cos \beta \cos \alpha \tag{16}$$

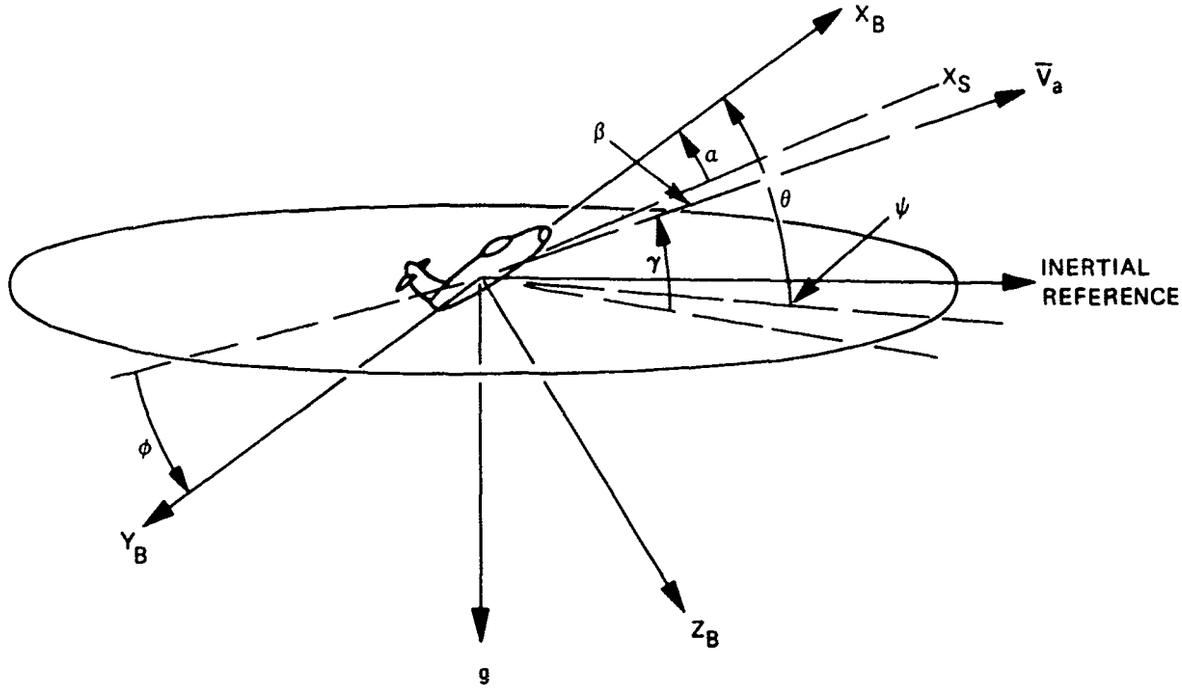
$$V_a = \overline{V}_a \sin \beta \tag{17}$$

$$W_a = \overline{V}_a \cos \beta \sin \alpha \tag{18}$$

In the absence of atmospheric turbulence, the α and β measurements correspond with the inertial values.

$$\alpha_i = \tan^{-1} \frac{W}{U} \tag{19}$$

$$\beta_i = \sin^{-1} \frac{V}{\overline{V}} \tag{20}$$



X_B, Y_B, Z_B – THE BODY-AXIS SYSTEM CONSISTS OF RIGHT-HANDED, ORTHOGONAL AXES WHOSE ORIGIN IS FIXED AT THE NOMINAL AIRCRAFT CENTER OF GRAVITY. THE SYSTEM'S ORIENTATION REMAINS FIXED WITH RESPECT TO THE AIRCRAFT, THE X_B AND Z_B AXES BEING IN THE PLANE OF SYMMETRY. BECAUSE THE EXACT ALIGNMENT OF X_B AXIS IS ARBITRARY, HERE IT IS TAKEN ALONG THE BODY CENTERLINE REFERENCE.

719-2-1

Figure 1
Aircraft Axis System

2. INTEGRATION OF THE AIRSPEED EQUATIONS

The most direct method of obtaining α_i is the solution of three simultaneous, first-order, nonlinear differential equations.

$$\dot{U} = Ax_{cg} - QW + RV - g \sin \theta \tag{21}$$

$$\dot{V} = Ay_{cg} - RU + PW + g \cos \theta \sin \phi \tag{22}$$

$$\dot{W} = Az_{cg} - PV + QU + g \cos \theta \cos \phi \tag{23}$$

This requires measurements of Ax_{cg}, Ay_{cg}, Az_{cg}, P, Q, R, θ, and ϕ. The equations are integrated to solve for U, V, and W, after which Equation (19) is solved. One serious drawback to this method is the requirement for inertial quality computation.

3. ALTITUDE RATE EQUATION

Compensated barometric altitude rate, $\dot{h}_i$, assuming constant altitude isobars, is given by

$$\dot{h}_i = U \sin \theta - V \cos \theta \sin \phi - W \cos \theta \cos \phi \tag{24}$$

Substituting inertial values for U, V, and W similar to Equations (16) through (18) results in

$$\dot{h}_i = \bar{V}_i \cos \beta_i \cos \alpha_i \sin \theta - \bar{V}_i \sin \beta_i \cos \theta \sin \phi$$
$$- \bar{V}_i \cos \beta_i \sin \alpha_i \cos \theta \cos \phi \tag{25}$$

In practice, $\bar{V}_a$ would be substituted for $\bar{V}_i$, and α_i would be given by

$$\alpha_i = \sin^{-1} \frac{-A \cos \theta \cos \phi + \sin \theta \sqrt{\cos^2 \theta \cos^2 \phi + \sin^2 \theta - A^2}}{\cos^2 \theta \cos^2 \phi + \sin^2 \theta} \tag{26}$$

where

$$A = \frac{\dot{h}_i}{\bar{V}_a \cos \beta_i} + \tan \beta_i \cos \theta \sin \phi \tag{27}$$

Equation (26) requires measurement of θ, ϕ, $\dot{h}$, $\bar{V}_a$, and β_i.

4. ELIMINATION OF $\dot{U}$, U, $\dot{V}$, V, $\dot{W}$, AND W

Differentiate Equations (16) through (18) and substitute in Equations (1) through (3).

$$\dot{\overline{V}}_a \cos\beta_i \cos\alpha_i - \dot{\beta}_i \overline{V}_a \sin\beta_i \cos\alpha_i - \dot{\alpha}_i \overline{V}_a \cos\beta_i \sin\alpha_i$$

$$= Ax_{cg} - Q \overline{V}_a \cos\beta_i \sin\alpha_i + R \overline{V}_a \sin\beta_i - g \sin\theta \quad (28)$$

$$\dot{\overline{V}}_a \sin\beta_i + \dot{\beta}_i \overline{V}_a \cos\beta_i$$

$$= Ay_{cg} - R \overline{V}_a \cos\beta_i \cos\alpha_i + P \overline{V}_a \cos\beta_i \sin\alpha_i$$

$$+ g \cos\theta \sin\phi \quad (29)$$

$$\dot{\overline{V}}_a \cos\beta_i \sin\alpha_i - \dot{\beta}_i \overline{V}_a \sin\beta_i \sin\alpha_i + \dot{\alpha}_i \overline{V}_a \cos\beta_i \cos\alpha_i$$

$$= Az_{cg} - P \overline{V}_a \sin\beta_i + Q \overline{V}_a \cos\beta_i \cos\alpha_i + g \cos\theta \cos\phi \quad (30)$$

Eliminating P, Q, and R yields

$$\dot{\overline{V}}_a \cos^2\beta_i \cos^2\alpha_i - \beta_i \overline{V}_a \sin\beta_i \cos\beta_i \cos^2\alpha_i$$

$$- \dot{\alpha}_i \overline{V}_a \cos^2\beta_i \sin\alpha_i \cos\alpha_i + \dot{\overline{V}}_a \sin^2\beta_i + \dot{\beta}_i \dot{\overline{V}}_a \cos\beta_i \sin\beta_i$$

$$+ \dot{\overline{V}}_a \cos^2\beta_i \sin^2\alpha_i - \dot{\beta}_i \overline{V}_a \sin\beta_i \cos\beta_i \sin^2\alpha_i$$

$$+ \dot{\alpha}_i \overline{V}_a \cos^2\beta_i \sin\alpha_i \cos\alpha_i$$

$$= Ax_{cg} \cos\alpha_i \cos\beta_i + Ay_{cg} \sin\beta_i$$

$$+ Az_{cg} \sin\alpha_i \cos\beta_i - g \sin\theta \cos\alpha_i \cos\beta_i$$

$$+ g \cos\theta \sin\phi \sin\beta_i + g \cos\theta \cos\phi \sin\alpha_i \cos\beta_i \quad (31)$$

When collecting terms, $\dot{\alpha}_i$ and $\dot{\beta}_i$ are eliminated, leaving

$$Ax_{cg} \cos \alpha_i \cos \beta_i + Ay_{cg} \sin \beta_i + Az_{cg} \sin \alpha_i \cos \beta_i = \dot{\bar{V}}_a \left(\cos^2 \beta_i \cos^2 \alpha_i + \sin^2 \beta_i + \cos^2 \beta_i \sin^2 \alpha_i\right) + g\left(\sin \theta \cos \alpha_i \cos \beta_i + \cos \theta \sin \phi \sin \beta_i + \cos \theta \cos \phi \sin \alpha_i \cos \beta_i\right) \quad (32)$$

The first term on the right-hand side of Equation (32) reduces to $\dot{\bar{V}}_a$, and the second term to $g \frac{\dot{h}_i}{\bar{V}_a}$ Thus

$$Ax_{cg} \cos \alpha_i \cos \beta_i + Ay_{cg} \sin \beta_i + Az_{cg} \sin \alpha_i \cos \beta_i = \dot{\bar{V}}_a + g \frac{\dot{h}_i}{\bar{V}_a} \quad (33)$$

which can be solved for α_i using measurements of Ax_{cg}, Ay_{cg}, Az_{cg}, β_i, $\dot{\bar{V}}_a$, $\bar{V}_a$, and $\dot{h}_i$.

5. EQUATING LEFT-HAND SIDES OF LIFT EQUATION IN STILL AIR

Combining Equations (28) and (30) and reducing,

$$(Ax_{cg} + \bar{V}_a \sin \beta_i R - g \sin \theta) \sin \alpha_i - (Az_{cg} - \bar{V}_a \sin \beta_i P + g \cos \theta \cos \phi) \cos \alpha_i + \bar{V}_a \cos \beta_i \dot{\alpha}_i = \bar{V}_a \cos \beta_i Q \quad (34)$$

This can be solved for α_i in terms of β_i, Ax_{cg}, Az_{cg}, P, Q, R, $\bar{V}_a$, θ, and ϕ.

6. LONGITUDINAL AND NORMAL ACCELERATION EQUATIONS

Equations (28), the longitudinal acceleration equation, and (30), the normal acceleration equation, both contain higher-order terms and do not appear promising on an individual basis unless many geometric assumptions are made. No steady-state solution for α_1 exists, nor can one be obtained by summing these two equations.

7. PITCHING MOMENT EQUATION

The aircraft pitching moment equation is given by

$$M_{\delta_T} \delta_T + M(\alpha) + M_{\dot{\alpha}} \dot{\alpha} + M_q Q + M_{\delta_e} \delta_e = Q + \frac{I_x - I_z}{I_y} PQ + \frac{I_{xz}}{I_y} (P^2 - R^2) \quad (35)$$

This differential equation can be solved for α without explicitly measuring Q if $M(\alpha) = M_\alpha \alpha$. The solution then requires that P, Q, R, δ_e, δ_T, m, and q be measured. The method for approximate solution is presented in Section III.

8. NORMAL FORCE EQUATION

The apparent simplicity of the aircraft normal force equation recommends its use for computing α.

$$Az_{cg} = Z(\alpha) + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T \tag{36}$$

$Z(\alpha)$ is a stability derivative that is known with high precision. Its large magnitude tends to relax the requirements for accuracy in representing Z_{δ_e} and Z_{δ_t}. Measurements of q and mass are also required.

Directly related to the normal force equation is the lift equation.

$$\frac{1}{m} (L_\alpha \alpha + L_{\delta_e} \delta_e) = Ax_{cg} \sin \alpha - Az_{cg} \cos \alpha - Z_{\delta_T} \delta_T \cos \alpha \tag{37}$$

The comments for the normal force equation apply here also. Equations (36) and (37) are mainly different in that (37) requires an extra measurement (Ax_{cg}), and the solution for α is made somewhat more complex because the right-hand-side terms are multiplied by functions of α.

9. SUM OF NORMAL FORCE AND PITCHING MOMENT EQUATIONS

Equations (35) and (36) can be summed to yield

$$Az_{cg} - X_a Q = Z(\alpha) + Z_{\delta_e} \delta_e + M_{\delta_T} \delta_T + Z_{\delta_T} \delta_T - X_a M_{\dot{\alpha}} \dot{\alpha} - X_a M(\alpha)$$

$$- X_a M_q Q - X_a M_{\delta_e} \delta_e + X_a \frac{I_x - I_z}{I_y} PQ$$

$$+ X_a \frac{I_{xz}}{I_y} (P^2 - R^2) \tag{38}$$

If X_a is chosen such that $X_a = (Z_{\delta_e}/M_{\delta_e})$, then the δ_e term in Equation (38) becomes zero. Furthermore,

$$Az_{cg} - \frac{Z_{\delta_e}}{M_{\delta_e}} \dot{Q} = A'_z \tag{39}$$

where A_Z' is the output of a normal accelerometer located at $X_a = Z_{\delta_e}/M_{\delta_e}$, and $Y_a = Z_a = 0$. Equation (38) requires measurements of P, Q, R, A_Z', δ_T, $\dot{\delta}_T$, and mass along with accurate knowledge of $Z(\alpha)$, Z_{δ_T}, Z_{δ_e}, $M(\alpha)$, M_{δ_e}, $M_{\dot{\alpha}}$, M_{δ_T}, M_q, I_x, I_z, I_y, and I_{xz}.

SECTION III

ERROR ANALYSIS

Section II developed a multitude of equations from which aircraft angle of attack could be extracted. The following section examines these equations with respect to complexity, practicality, and accuracy in an attempt to determine a feasible system.

1. EQUATIONS YIELDING INERTIAL α

Integration of Equations (21) through (23) will produce a high-quality inertial α, provided inertial quality computer and sensors are used. From a cost standpoint alone, this method is quite impractical. In addition, any attempt to simplify the equations will result in cumulative errors that quickly become intolerable.

Equation (25), using inertial altitude rate and Euler angle measurements, appears to be a promising algorithm. It has a steady-state solution, thus preventing cumulative error buildup. Furthermore, the sensor requirements are minimal except for a possible β measurement. This equation (repeated here for convenience with $\overline{V}_a$ substituted for $\overline{V}_i$) is known henceforth as Method I.

$$\dot{h}_i = \overline{V}_a (\cos \beta_i \cos \alpha_i \sin \theta - \sin \beta_i \cos \theta \sin \phi$$

$$- \cos \beta_i \sin \alpha_i \cos \theta \cos \phi$$

A thorough study of this equation will be presented following the selection of all other candidate methods.

Equation (33) also appears to be a useful algorithm. It requires an air data system, three accelerometers, and a β measurement. Like Equation (25), it provides a steady-state solution, thus relaxing the mechanization requirements. Equation (33) will be designated as Method II.

$$Ax_{cg} \cos \alpha_i \cos \beta_i + Ay_{cg} \sin \beta_i + Az_{cg} \sin \alpha_i \cos \beta_i = \dot{\overline{V}}_a + g \frac{\dot{h}_i}{\overline{V}_a}$$

Equation (34) has no steady-state solution for α_i and is therefore subject to cumulative errors. In addition, it is considerably more complex than Methods I and II, requiring accelerometers, rate gyros, Euler angles, β, and air data measurements. Thus, Equation (34) is not recommended for α computation.

2. EQUATIONS YIELDING TRUE α

Equation (35), the pitching moment equation, is fairly complicated as given. However, several assumptions can be made to considerably simplify the equation. First, assume that P, Q, and R are small, making their products negligible and $M(\alpha) = M_\alpha \alpha$. Equation (35) then becomes

$$M_{\delta_T} \delta_T + M_\alpha \alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q Q + M_{\delta_e} \delta_e = \dot{Q} \tag{40}$$

Solving for $\dot{\alpha}$ yields

$$\dot{\alpha} = \frac{1}{M_{\dot{\alpha}}} \left(\dot{Q} - M_{\delta_T} \delta_T - M_\alpha \alpha + M_q Q - M_{\delta_e} \delta_e \right) \tag{41}$$

Taking some mathematical liberties with Equation (41),

$$\alpha \approx \frac{1/M_{\dot{\alpha}} \ (S - M_q)}{S + M_\alpha / M_{\dot{\alpha}}} Q - \frac{M_{\delta_e}/M_{\dot{\alpha}}}{S + M_\alpha / M_{\dot{\alpha}}} \delta_e - \frac{M_{\delta_T} \delta_T / M_{\dot{\alpha}}}{S + M_\alpha / M_{\dot{\alpha}}} \tag{42}$$

indicating that $\dot{Q}$ need not be measured. The equation does require measurements of Q, δ_e, δ_T, mass, and Mach number and accurate knowledge of $M_{\dot{\alpha}}$, M_α, M_q, M_{δ_T}, and M_{δ_e}. Two basic factors limit the usefulness of Equation (42). First, and most important, the computed value of α is an approximation at the outset. Second, the equation requires accurate knowledge of four aircraft stability derivatives as a function of Mach and mass, which leads to a complex mechanization. In view of the simpler, more accurate methods available, Equation (42) is not recommended for computing α.

The normal force equation, Equation (36), is given by

$$AZ_{cg} = Z(\alpha) + Z_{\delta_e} \delta_e + Z_{\delta_T} \delta_T$$

As was stated previously, the relative simplicity of this equation recommends its use in α computation. It is henceforth known as Method III and will be analyzed in greater depth later.

The lift equation is similar in principle to the normal force equation but has additional complexities. Since there is nothing to recommend its use over Method III, it is discarded. The same can be said for summing the normal force and pitching moment equations.

3. ATMOSPHERIC TURBULENCE

Before examining the errors that can develop due to simplifying assumptions, atmospheric turbulence effects should be considered. Both Methods I and II compute inertial α, which is in error due to the contribution of gust velocities to U_a and W_a. Whether the α error is serious depends on the mission requirements, the severity of the turbulence, and the aircraft response.

The requirements for a high-frequency α response come primarily from the weapons computer and/or the gust alleviation system input. The weapons delivery problem is complicated by the effect of aircraft velocity. If weapons delivery takes place at relatively high velocities, then effects of gusts on α can be sufficiently small to be ignored. Whether inertial α is sufficient for weapons delivery will be a function of the individual aircraft and mission requirements.

The requirements for α at low speed are generally dictated by some form of speed command system in which high-frequency excitation is undesirable. Therefore, a low pass filtered inertial α may be adequate as an input. Once again, the requirements of the specific system must be known to determine the applicability of inertial α computation.

To illustrate the problem, several simulation runs were made using the test aircraft. The results of two such runs are presented below.

Flight Condition	Mach	Altitude	Turbulence*	$N_{z_{cg}}$ (RMS g's)	E_{11}** (RMS degrees)	E_{12} (peak degrees)	E_{21} (RMS degrees)	E_{22} (peak degrees)
1	0.25	Sea level	Slight	0.1	0.35	0.8	0.34	0.8
2	0.9	Sea level	Very heavy	0.35	0.24	0.56	0.24	0.60

*Reference 2

**E_{11} and E_{12} – Method I α error

E_{21} and E_{22} – Method II α error

4. APPROXIMATIONS

Of the many different methods of inferential α computation, only three remain that are accurate and yet relatively simple. A disadvantage (turbulence) of Methods I and II has already been discussed. These two methods do, however, have one distinct advantage over the third: their universal applicability to any aircraft type without calibration. Method III, on the other hand, will require accurate knowledge of at least two aerodynamic coefficients for each aircraft type.

The approximations that can be made are limited by the wide range of flight parameters over which α must be computed. Obviously, no small-angle approximations can be made for either θ or ϕ, which can be as large as 75 degrees.

The effects of approximations are best illustrated by exercising the simulated aircraft through large-angle maneuvers wherein the principal equation parameters become large. The aircraft response is shown in Figures 2, 3, and 4. As each approximation is considered, the error produced should be compared with the aircraft response to gain insight into the approximation effects with respect to each of the equation parameters.

Side-slip angle can be approximated by $\sin \beta_i = \beta_i$ and $\cos \beta_i = 1$. This causes Methods I and II [Equations (25) and (33)] to become, respectively,

$$\dot{h}_i = \overline{V}_a \left(\cos \alpha_i \sin \theta - \beta_i \cos \theta \sin \phi - \sin \alpha_i \cos \theta \cos \phi\right) \tag{43}$$

and

$$Ax_{cg} \cos \alpha_i + \beta_i \, Ay_{cg} + Az_{cg} \sin \alpha_i = \dot{\overline{V}}_a + g \frac{\dot{h}_i}{\overline{V}_a} \tag{44}$$

The effect of this approximation on α computation is shown in Figure 5. Since the error which occurs is small, this approximation is useful for all but the most stringent system requirements.

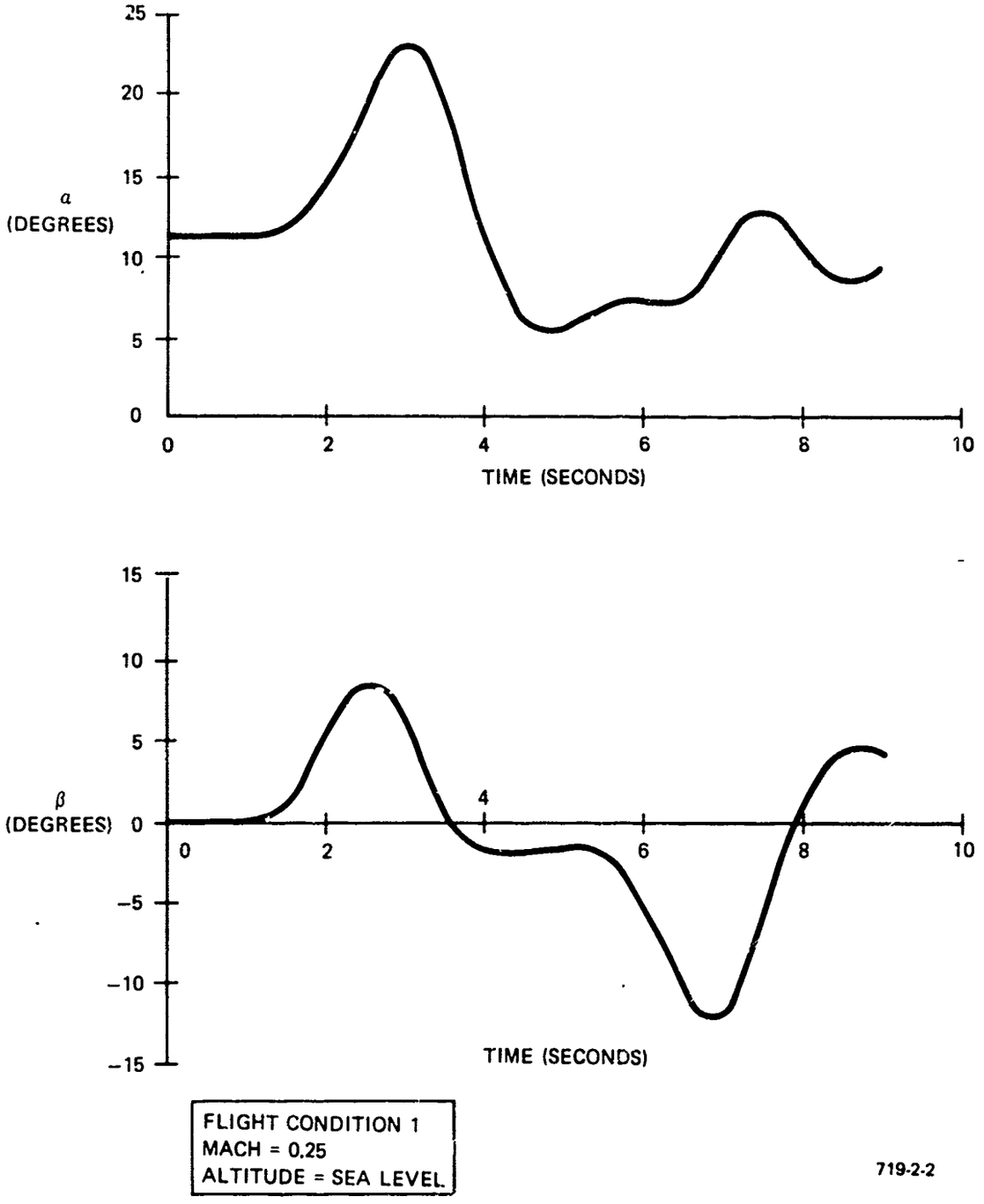


Figure 2
Aircraft α and β Response

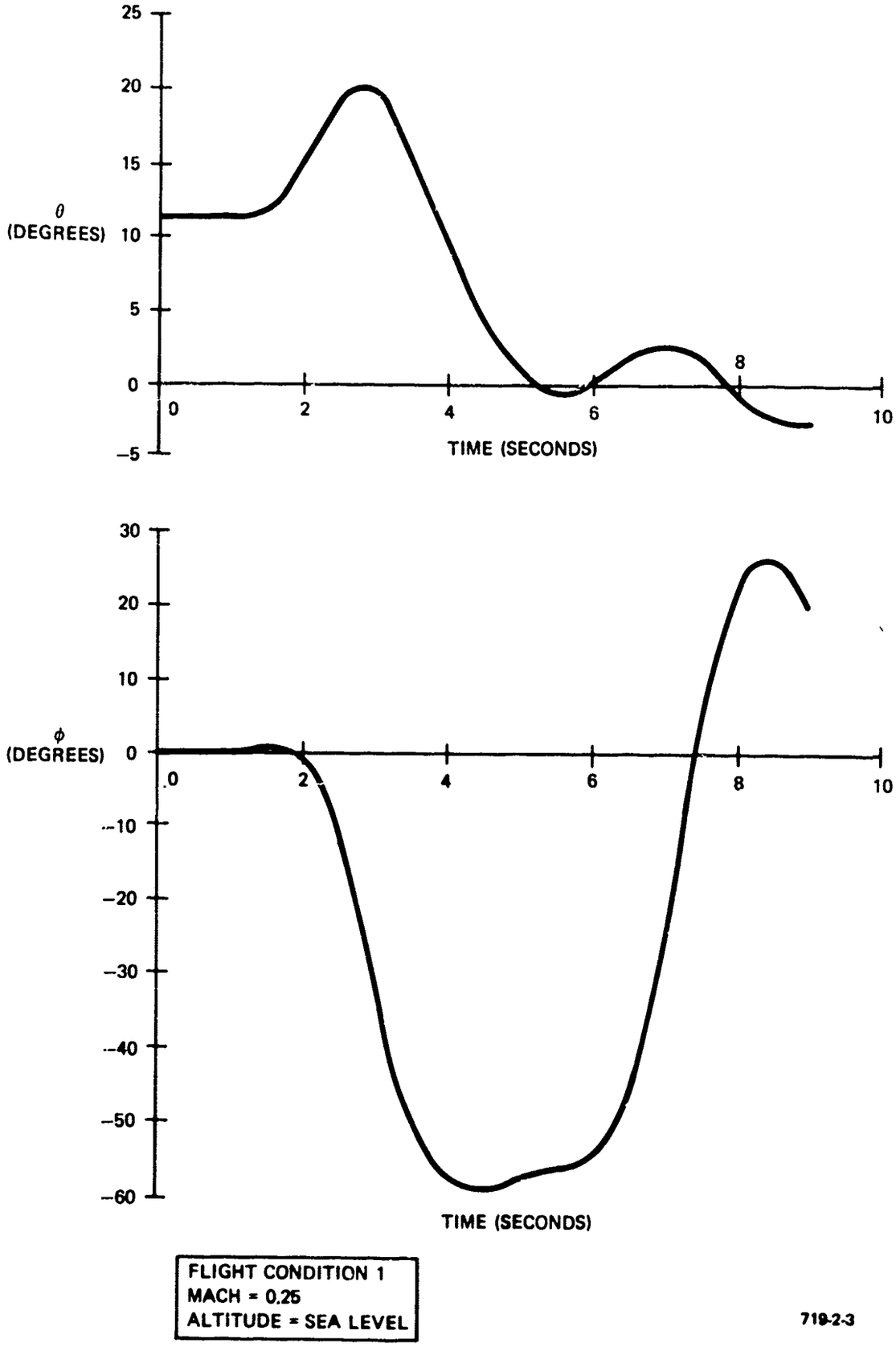
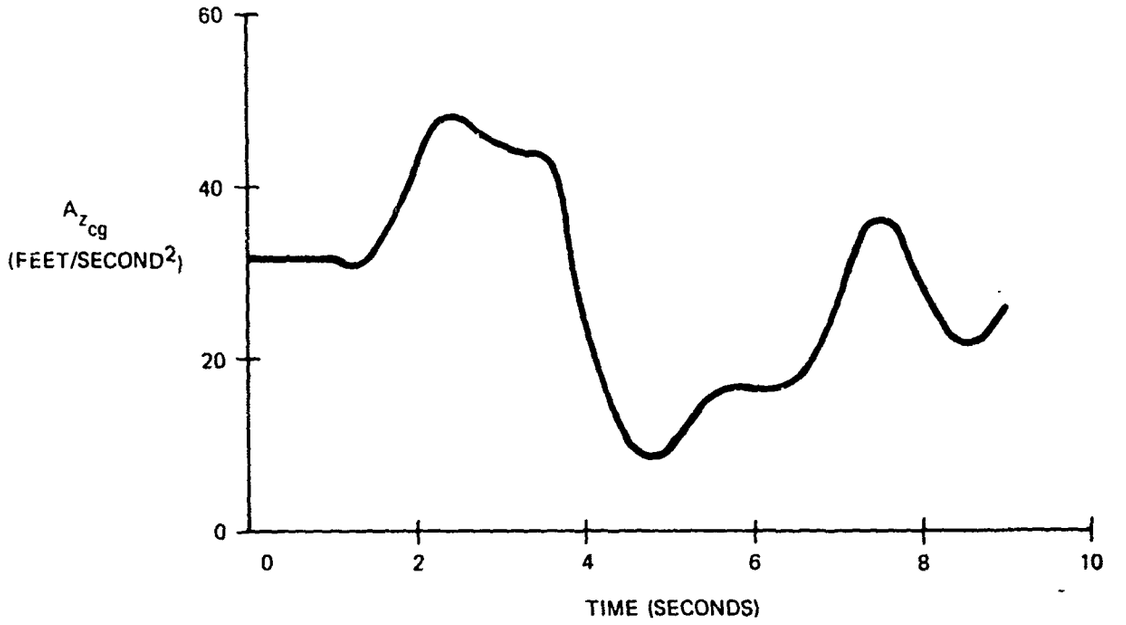
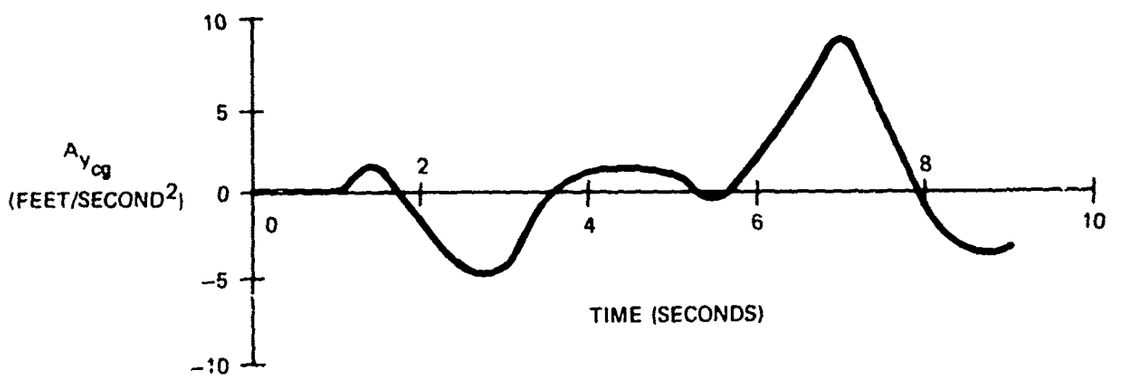


Figure 3
Aircraft θ and Ø Response





FLIGHT CONDITION 1
MACH = 0.25
ALTITUDE = SEA LEVEL

719-2-4

Figure 4
Aircraft $A_{z_{cg}}$ and $A_{y_{cg}}$ Response

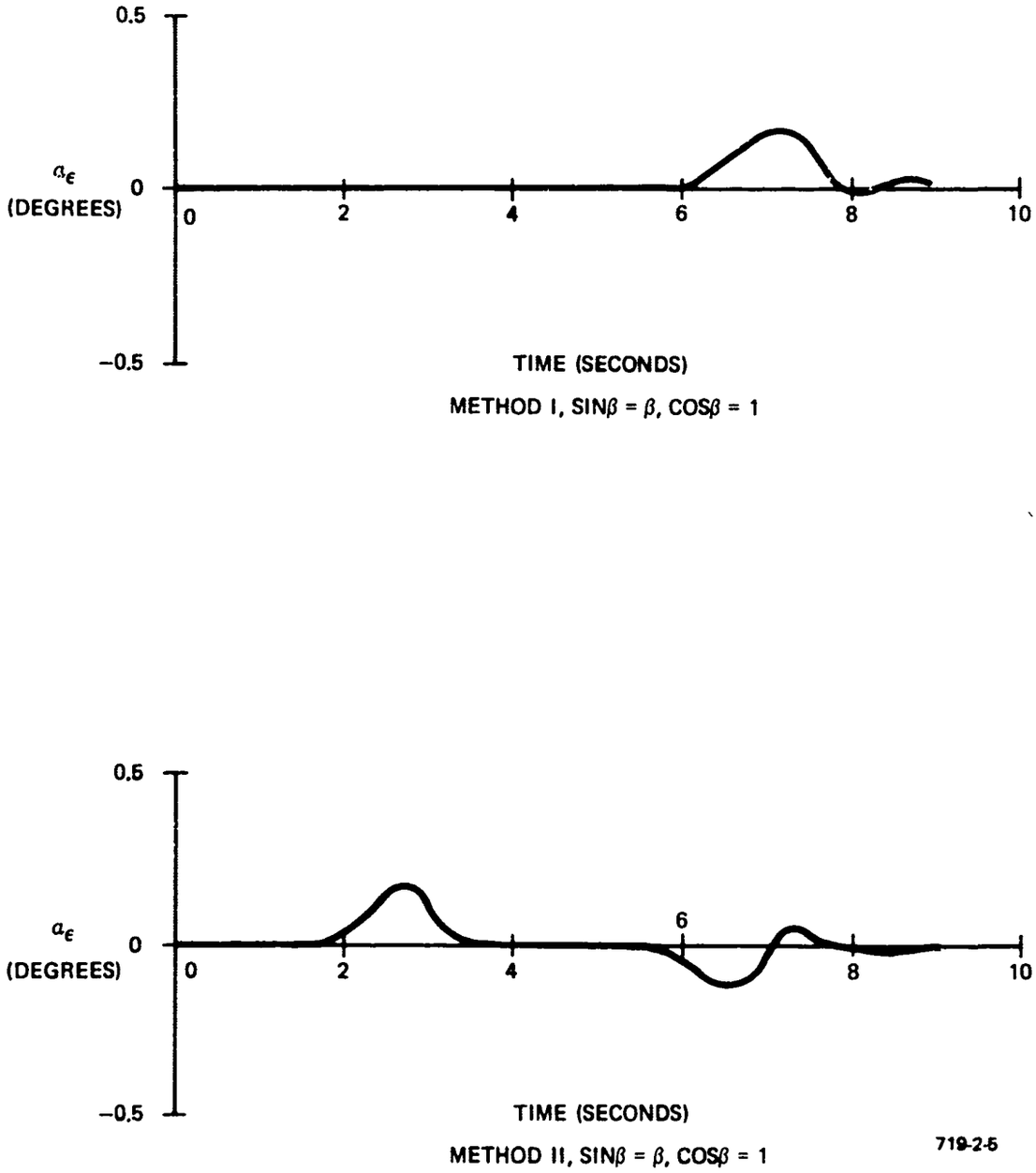


Figure 5
Errors Due to Side-Slip Angle Approximation

If side-slip angle is completely ignored, Equations (25) and (33) simplify to

$$\dot{h}_i = \overline{V}_a \, (\cos \alpha_i \sin \theta - \sin \alpha_i \cos \theta \cos \phi) \tag{45}$$

and

$$Ax_{cg} \cos \alpha_i + Az_{cg} \sin \alpha_i = \dot{\overline{V}}_a + g \frac{\dot{h}_i}{\overline{V}_a} \tag{46}$$

This last assumption results in considerable error under certain conditions. The error occurs in Method I when β is large (10 degrees) and a roll angle is present. As much as 1.5 degrees of error can result if ϕ is only 5 degrees. Larger bank angles (30 to 40 degrees) can produce considerably large errors (8 to 10 degrees). This is presented graphically in Figure 6.

Method II will be in error if β is large and a lateral acceleration is present. For the particular case shown in Figure 6, the error in Method II is smaller than that of Method I, but this is not always true. Larger lateral accelerations will produce equivalent errors in Method II.

Geometric approximations for sin α and cos α simplify the mechanization requirements considerably. However, the errors produced are unacceptable for all but the lowest quality system. These errors are illustrated in Figures 7, 8, 9, and 10.

Method III does not lend itself to geometric simplifications. Those parameters which are not critical include $C_{z_{\delta_e}}$ and Z thrust. Depending on aircraft type, Z thrust may be zero. The representation of $C_{z_{\delta_e}}$ in the mechanization may require only a simple constant gain. On the other hand, $C_z(\alpha)$ must be precisely represented, including the effects of any high-lift devices.

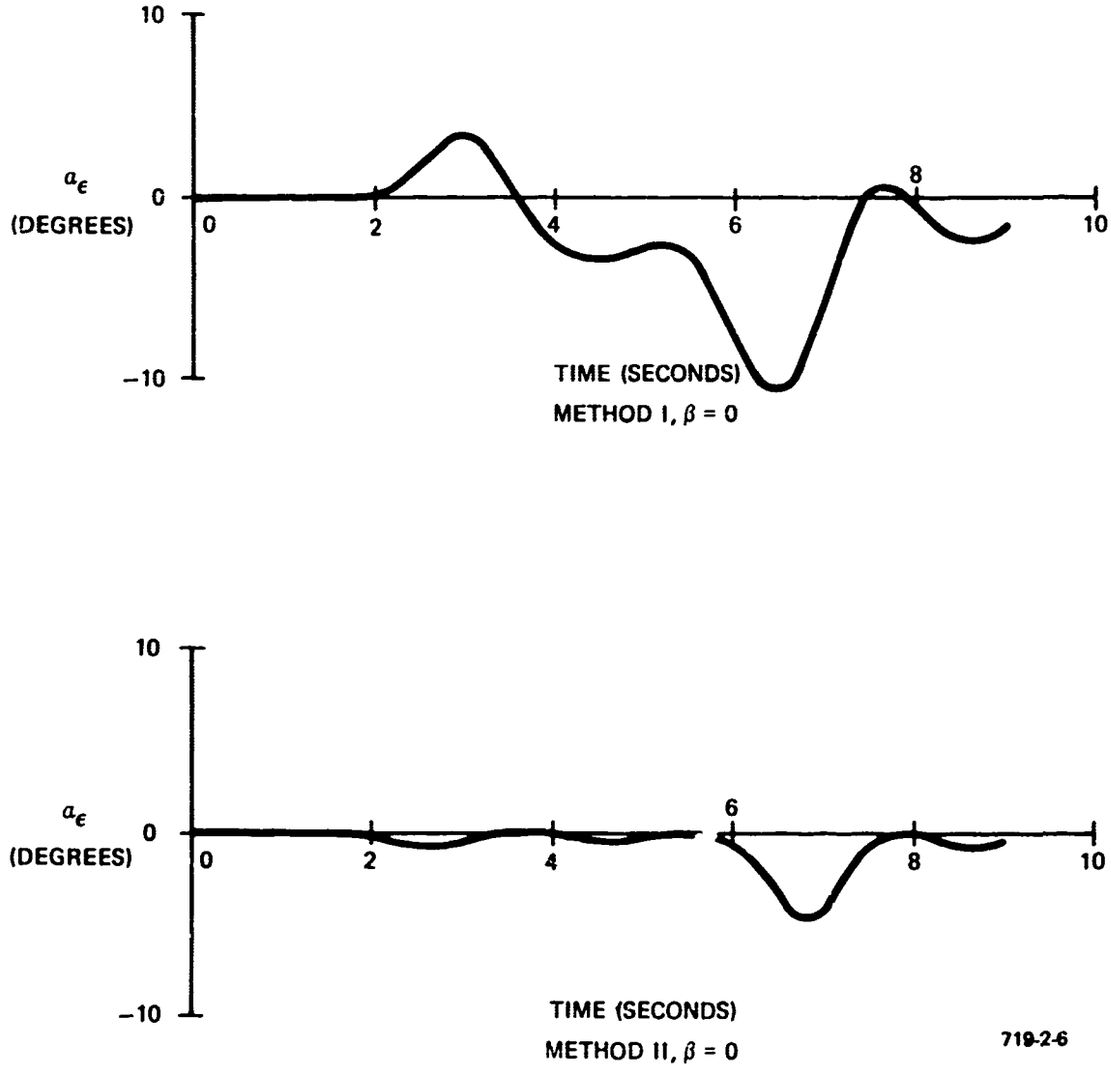
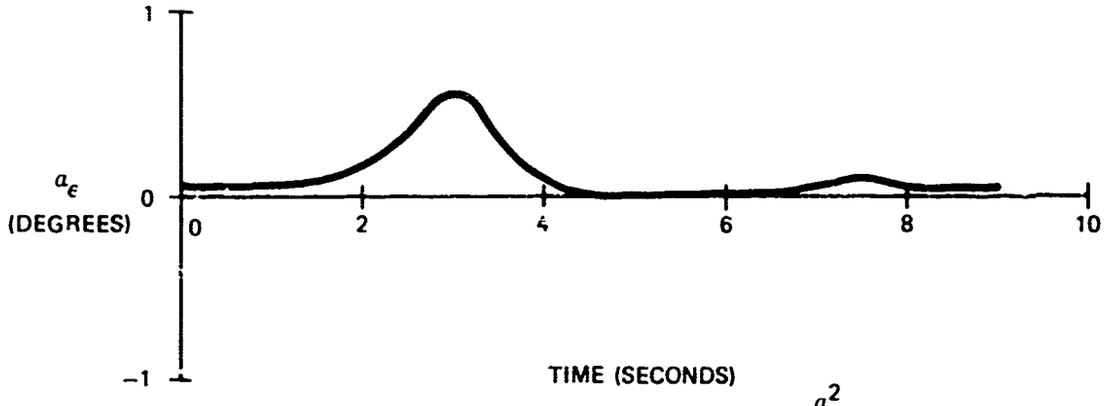
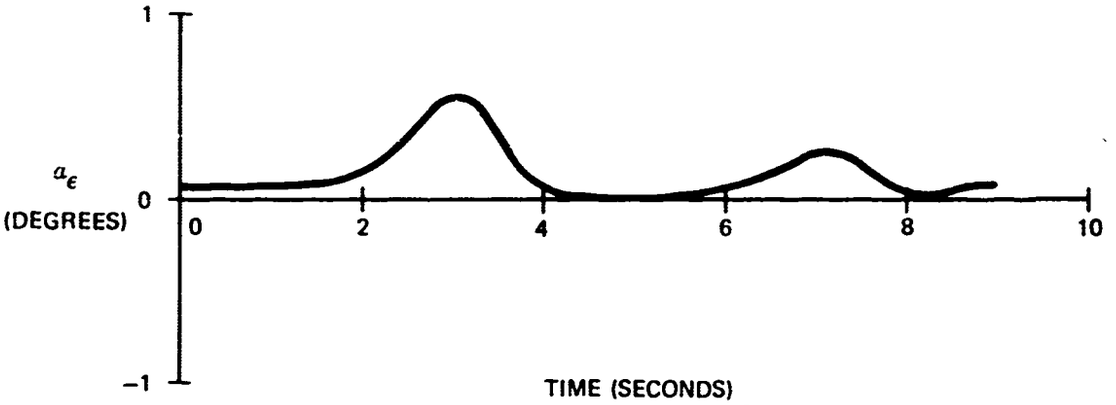


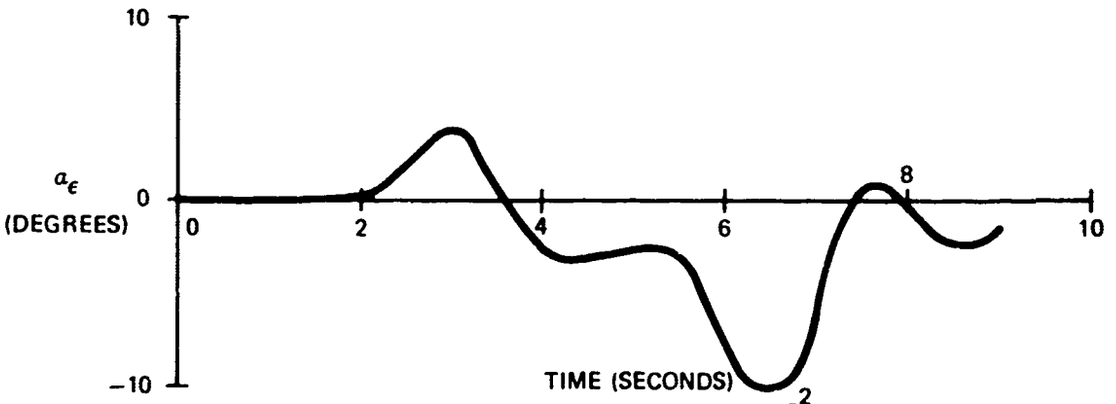
Figure 6
Errors Due to Ignoring Side-Slip



METHOD I, SINα = α, COSα = $1 - \frac{\alpha^2}{2}$



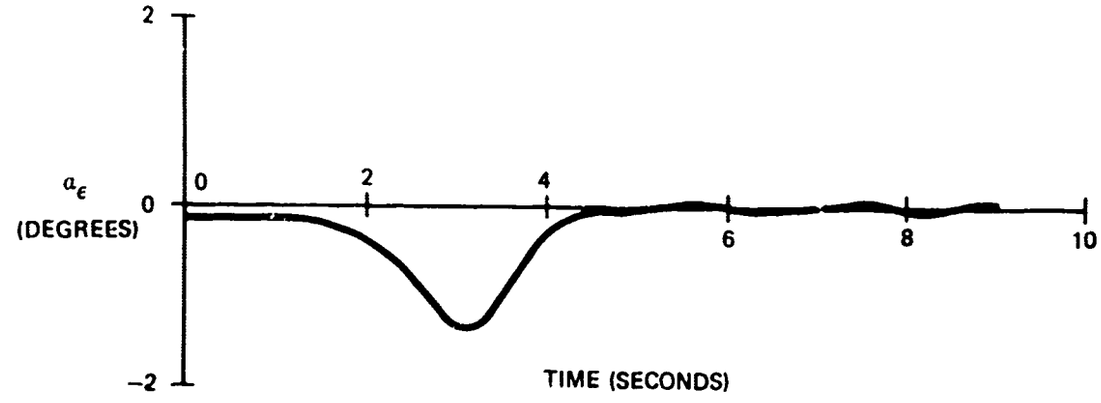
METHOD I, SINα = α, COSα = $1 - \frac{\alpha^2}{2}$, SINβ = β, COSβ = 1



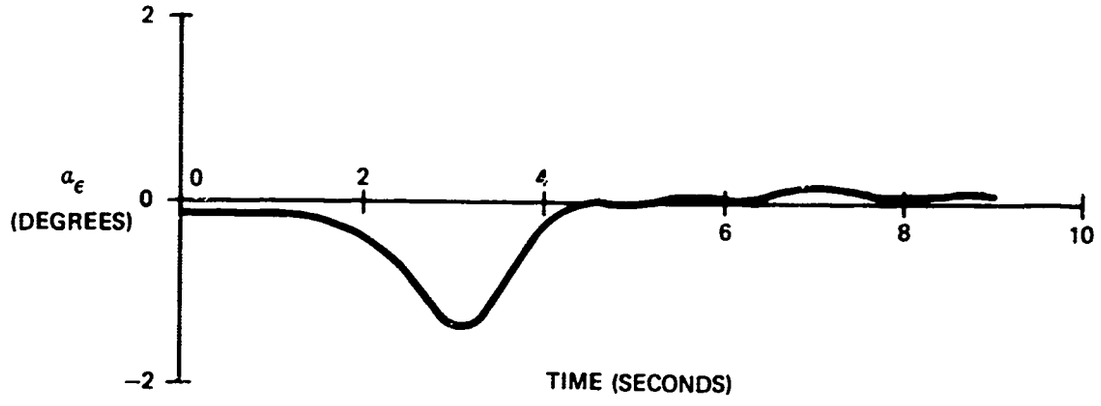
METHOD I, SINα = α, COSα = $1 - \frac{\alpha^2}{2}$, β = 0

719-2-7

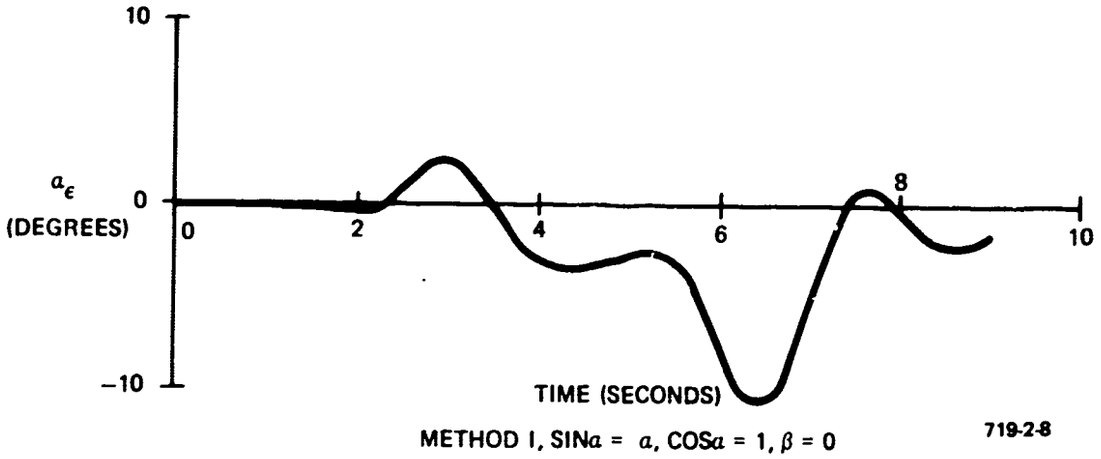
Figure 7
Error in Method I
when Sin α = α and Cos α = $1 - \frac{\alpha^2}{2}$



METHOD I, SINα = α, COSα = 1



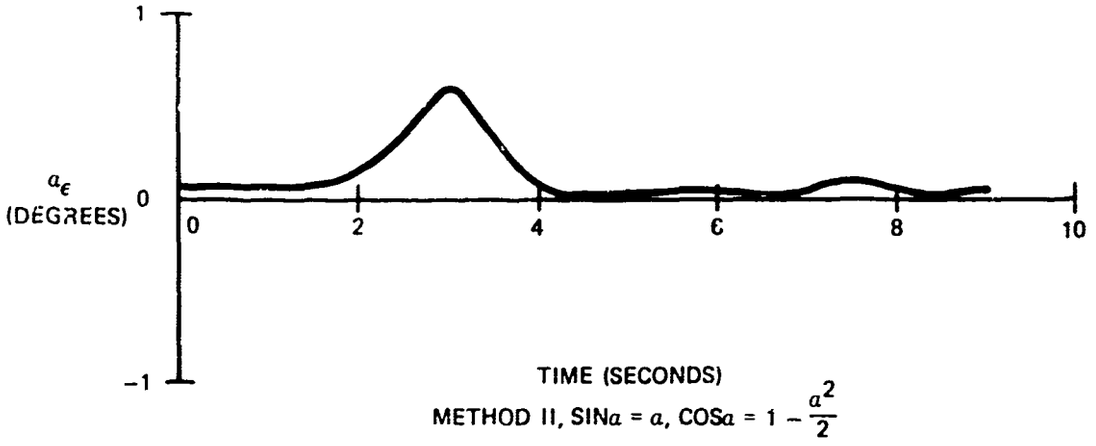
METHOD I, SINα = α, COSα = 1, SINβ = β, COSβ = 1

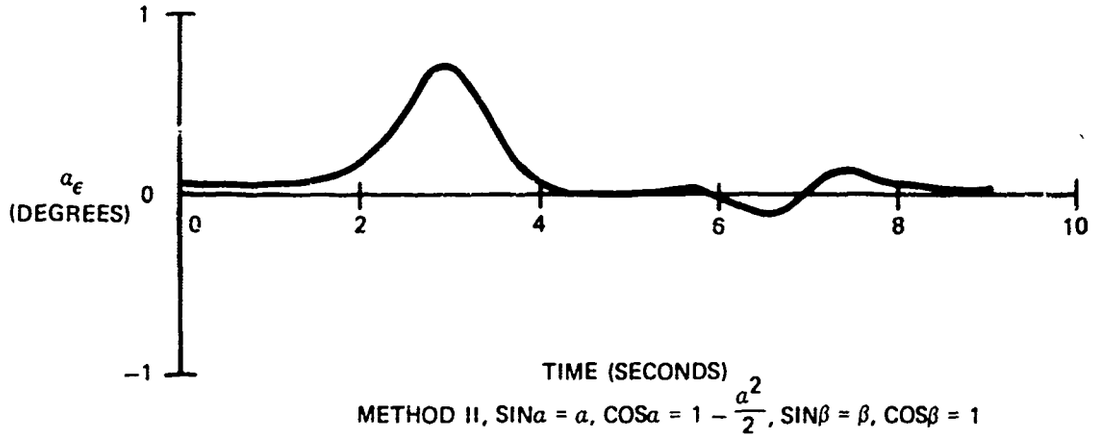


METHOD I, SINα = α, COSα = 1, β = 0

719-2-8

Figure 8
Error in Method I
when Sin α = α and Cos α = 1





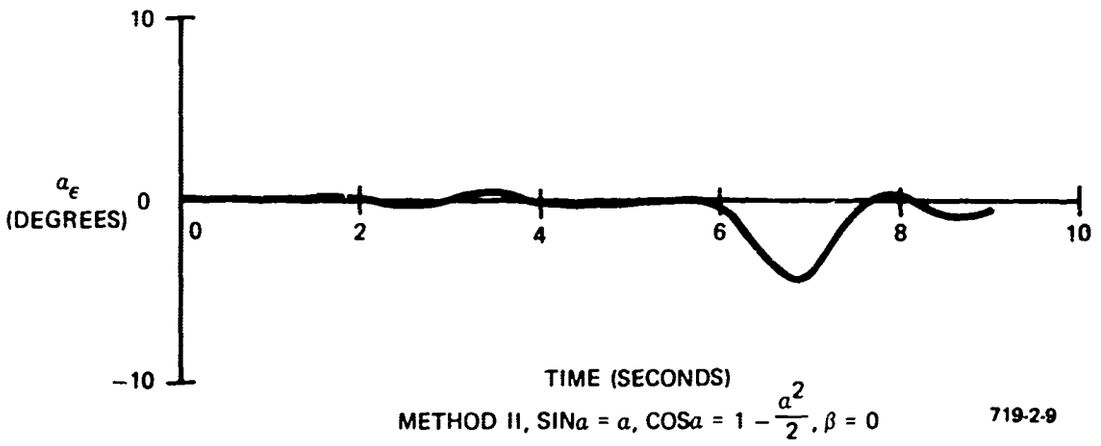
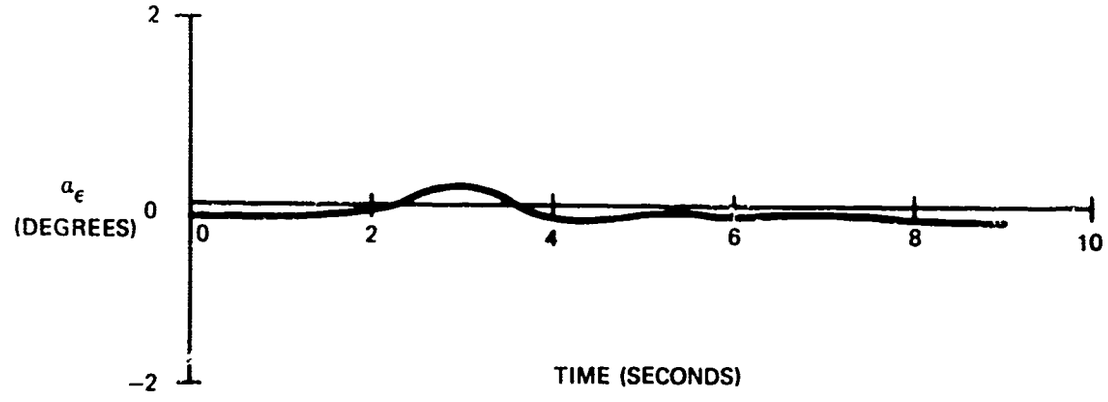
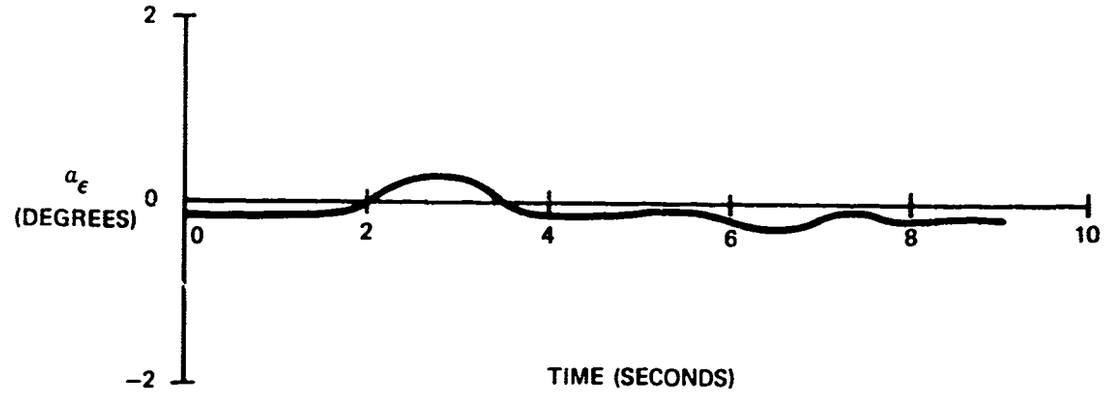


Figure 9
Error in Method II
when Sin $\alpha = \alpha$ and Cos $\alpha = 1 - \frac{\alpha^2}{2}$





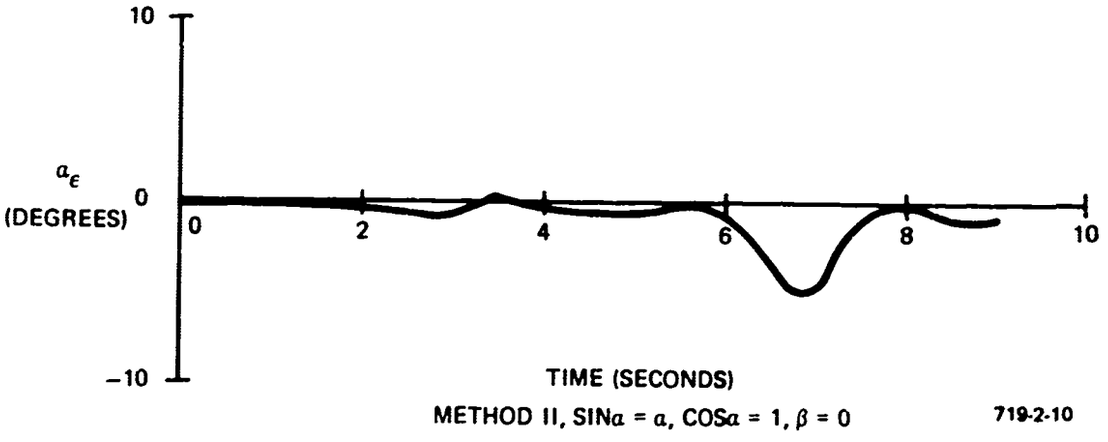


Figure 10
Error in Method II
when Sin α = α and Cos α = 1

SECTION IV

SENSOR ANALYSIS

1. INTRODUCTION

To gain insight into the effects of sensor-induced errors, the study has been separated into maneuvering (dynamic) and non-maneuvering (static) flight. The static case uses reduced equations to reveal the basic accuracy to be expected from each computing method. The dynamic equations, which illustrate the effects of each equation parameter, have been evaluated on a worst-case basis since a completely analytical approach proved unenlightening.

Ideally, the sensor inputs to the α computer should have zero error. Any improvement in basic sensor accuracy will improve the α computation. However, it is beyond the scope of this program to improve basic sensor designs so as to increase their accuracy. We can at best point out those sensors which are critical to the accurate computation of α. Operations involving sensor outputs such as filtering, quickening, and mode conversion are considered where improvements in α computation are obtained.

2. STATIC ERRORS

Examining the first two methods of α computation under conditions of wings-level, constant-altitude flight reveals their basis for computing steady-state α. In the absence of input offsets in static flight, Equations (25) and (36) reduce to, respectively,

$$-\sin \alpha_i \cos \theta + \cos \alpha_i \sin \theta = 0 \tag{47}$$

and

$$Ax_{cg} \cos \alpha_i + Az_{cg} \sin \alpha_i = 0 \tag{48}$$

Equation (47) merely states that for wings-level, constant-altitude flight, $\alpha_i = \theta$. Thus, the static computation of α_i in Method I is simply the output of the vertical reference. This implies that the α computation can be no better than the vertical reference accuracy when using Method I.

Equation (48) has reduced α_i in Method II to the arctangent of the ratio of Ax_{cg} to Az_{cg}. But in static flight, $Ax_{cg} = -g \sin \theta$ and $Az_{cg} = g \cos \theta$, and again, $\alpha_i = \theta$. In this case, the accuracy of the α computation is dependent on the basic accelerometer accuracies and their alignments with the X and Z body axes.

The third computation method for α does not reduce in the static case, but the equation allows a mathematical approach to the error analysis, which will be discussed subsequently.

Table I shows the effects of static sensor errors on the three methods of α computation for the six flight conditions. In each case, one sensor was given a 5-percent offset from the trim value, while the other sensor outputs remained accurate. The values presented reveal that in Methods I and II the α computation accuracy is directly proportional to the accuracy of the input sensors. In Method III it is apparent that three of the seven parameters are not critical; those being thrust, $C_{z_{\delta_e}}$, and elevator position. This is of particular importance with respect to thrust since it may be difficult to accurately measure in complex engines.

TABLE I

ANGLE-OF-ATTACK ERROR AT TRIM DUE TO STATIC SENSOR ERRORS

Flight Condition*	Method	α_{trim}	θ_{ε}	$Ax_{cg_{\varepsilon}}$	$T_{z_{\varepsilon}}$	$\delta_{e_{\varepsilon}}$	q_{ε}	m_{ε}	$C_{z_{\delta_{e_{\varepsilon}}}}$	$Az_{cg_{\varepsilon}}$	$C_{z(\alpha)_{\varepsilon}}$
1	I	11.2	0.560**	0.544							
	II									0.479	
	III				0.046	0.054	0.506	0.482	0.054	0.460	0.533
3	I	2.1	0.105	0.105							
	II									0.100	
	III				0.058	0.018	0.086	0.082	0.018	0.028	0.100
7	I	2.9	0.145	0.145							
	II									0.138	
	III				0.099	0.016	0.128	0.123	0.016	0.028	0.139
8	I	7.5	0.375	0.375							
	II									0.354	
	III				0.046	0.038	0.337	0.322	0.038	0.290	0.358
10	I	7.5	0.375	0.375							
	II									0.354	
	III				0.206	0.019	0.355	0.339	0.019	0.560	0.358
11	I	0.8	0.040	0.040							
	II									0.039	
	III				0.023	0.009	0.031	0.030	0.009	0.053	0.038

*Flight Condition 1 Mach = 0.25 Altitude = Sea level
Flight Condition 3 Mach = 0.9 Altitude = Sea level
Flight Condition 7 Mach = 1.1 Altitude = 15,000 feet
Flight Condition 8 Mach = 0.6 Altitude = 35,000 feet
Flight Condition 10 Mach = 0.154 Altitude = Sea level (landing configuration)
Flight Condition 11 Mach = 2.5 Altitude = 35,000 feet

**Error in computed α in degrees. Sensors were given a 5-percent output error.

3. DYNAMIC ERRORS

The analysis of dynamic errors considers the unreduced Equations (25), (33), and (36) in the presence of maneuvering flight. A worst-case study of all three methods is followed by a mathematical error analysis of Method III.

3.1 Worst-Case Study

A worst-case analysis of sensor errors has been used due to the relatively fruitless results obtained from a purely mathematical approach. Some special situations in which the equations reduce to a reasonable complexity will be mathematically analyzed later.

As an example of the complexity resulting from a mathematical approach, consider the solution of Equation (25) using the identity

$$\cos \alpha_i = \left(1 - \sin^2 \alpha_i\right)^{1/2} \tag{49}$$

Equation (25) then becomes

$$\frac{\dot{h}_i}{V_a} + \sin \beta_i \cos \theta \sin \phi = \left[-\sin \alpha_i \cos \theta \cos \phi + \left(1 - \sin^2 \alpha_i\right)^{\frac{1}{2}} \sin \theta\right] \cos \beta_i \tag{50}$$

When solved for α_i, this yields

$$\alpha_i = \sin^{-1} \frac{A \cos \theta \cos \phi + \sin \theta \left(\cos^2 \theta \cos^2 \phi + \sin^2 \theta - A^2\right)^{1/2}}{\cos^2 \theta \cos^2 \phi + \sin^2 \theta} \tag{51}$$

where

$$A = \frac{\dot{h}_i}{V_a \cos \beta_i} + \tan \beta_i \cos \theta \sin \phi \tag{52}$$

The sensitivity of α_i to the individual parameters is determined by taking the partial derivative of Equation (51) with respect to the subject parameter. The derivative of arcsin is given by

$$d\alpha = \left(1 - u^2\right)^{-1/2} du \tag{53}$$

Equation (53) applied to Equation (51) for any of the parameters leads to a highly complex expression that provides no insight into the parameter effects. Furthermore, if a quantitative evaluation is attempted, realistic values must be assigned to each parameter. This either limits the analysis to one or two points, or inflates the task beyond reasonable bounds.

The worst-case analysis procedure consisted of examining the effects of an erroneous sensor output during large aircraft maneuvers. Flight Conditions (FC) 1 and 3 were used to provide both low- and high-performance cases. Large bank angles and side-slip angles were produced using FC-1, whereas FC-3 provided large normal and lateral accelerations. In each instance, the maneuvers were limited to produce realistic parameter values. "Worst case" must be qualified by stating that although the maneuvers may not represent a perfect worst-case situation with respect to α computation, they do exercise all aircraft parameters beyond normal conditions without resorting to an unrecoverable attitude.

In each simulation run, one sensor was given a 5-percent scale error while all others remained accurate. The influence of sensor inaccuracy was then observed by recording the error between the computed and true α. The aircraft response for FC-1 (Mach 0.25 at sea level) is shown in Figure 11. The errors induced in each of the three computing methods are presented in Figures 12 through 14. The high-performance case, FC-3 (Mach 0.9 at sea level), is shown in Figures 15 through 18.

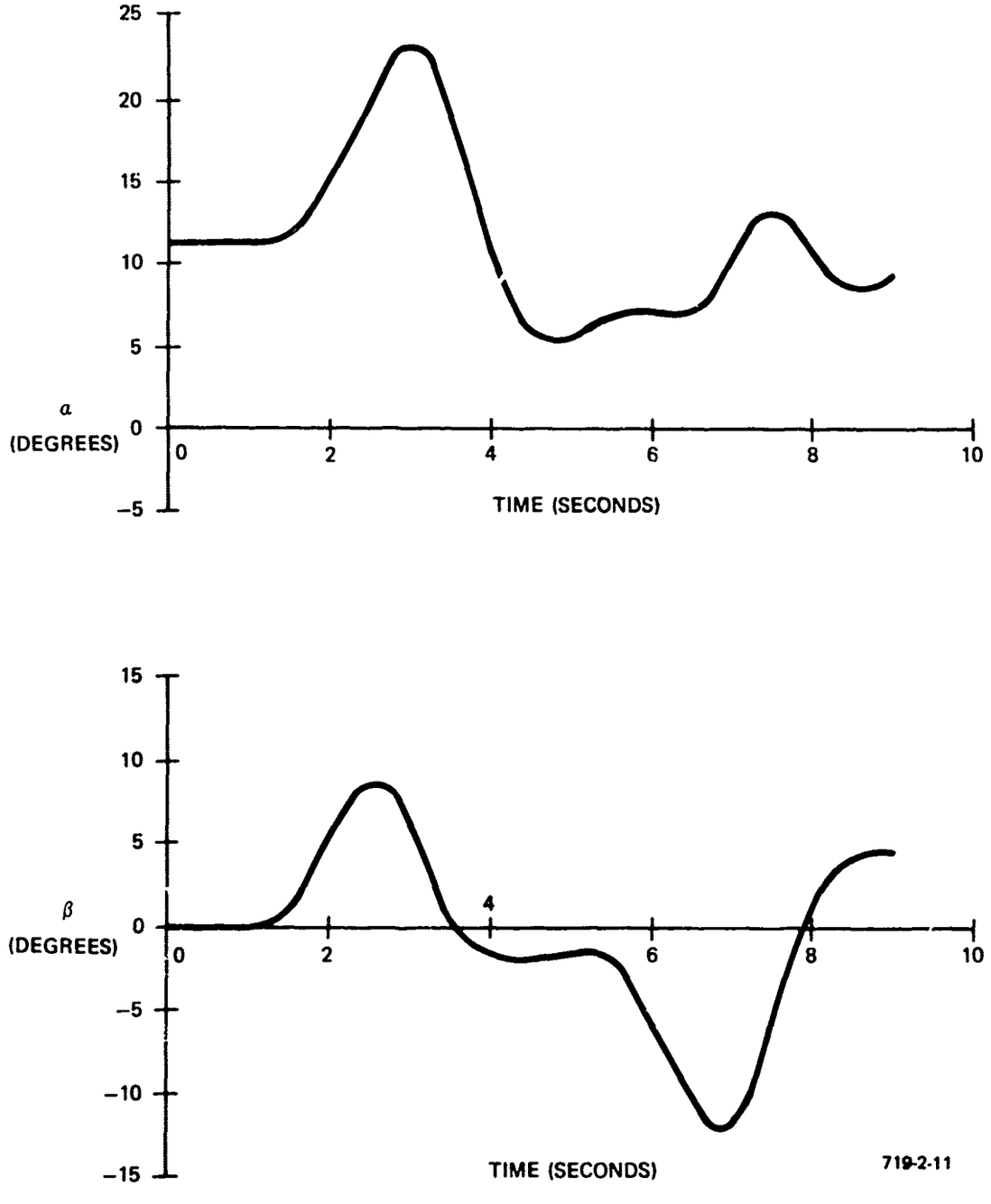


Figure 11
Aircraft Response, Flight Condition 1
(a) α and β

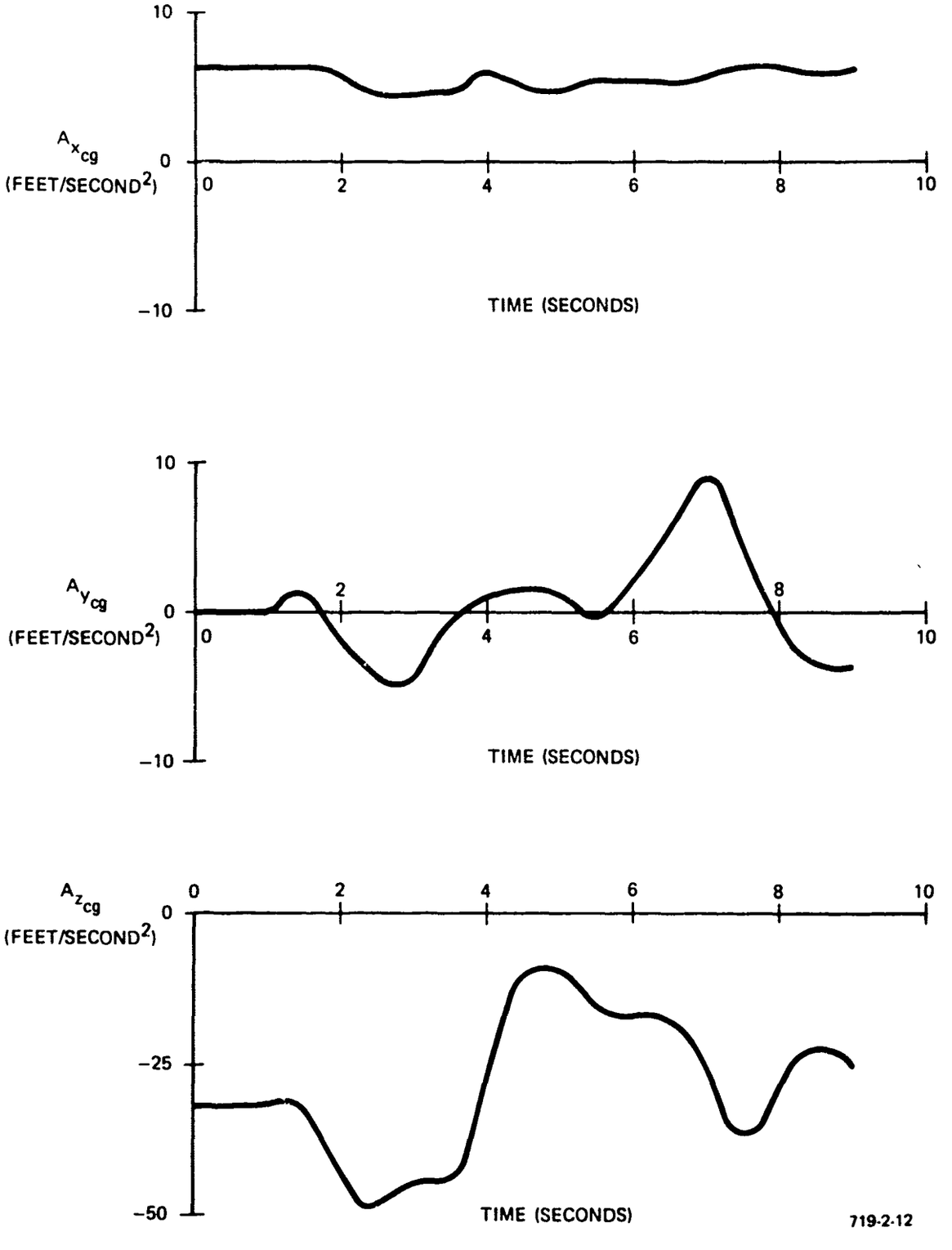


Figure 11 - Continued
(b) $A_{x_{cg}}$, $A_{y_{cg}}$, and $A_{z_{cg}}$

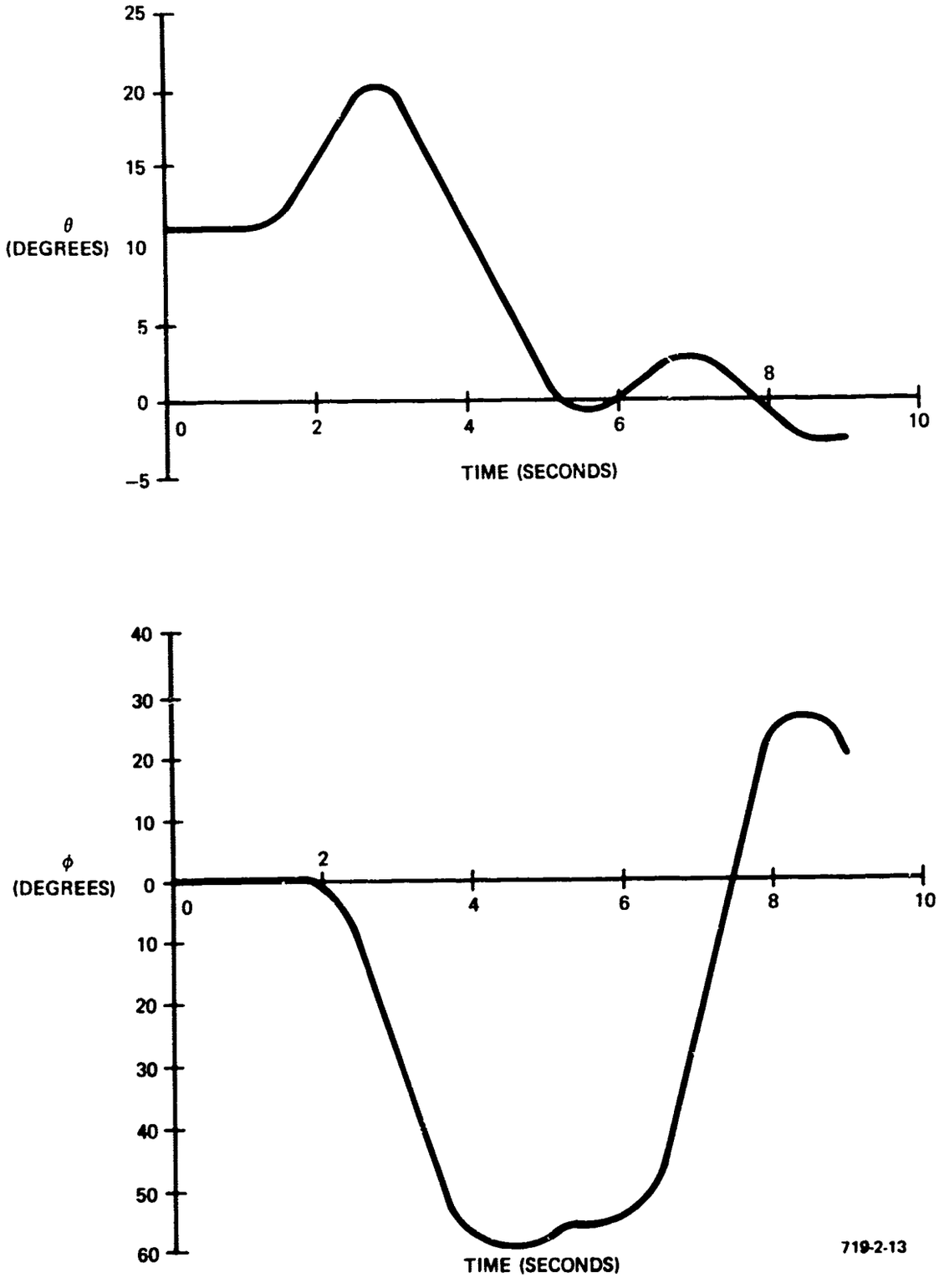


Figure 11 - Continued
(c) θ and Ø

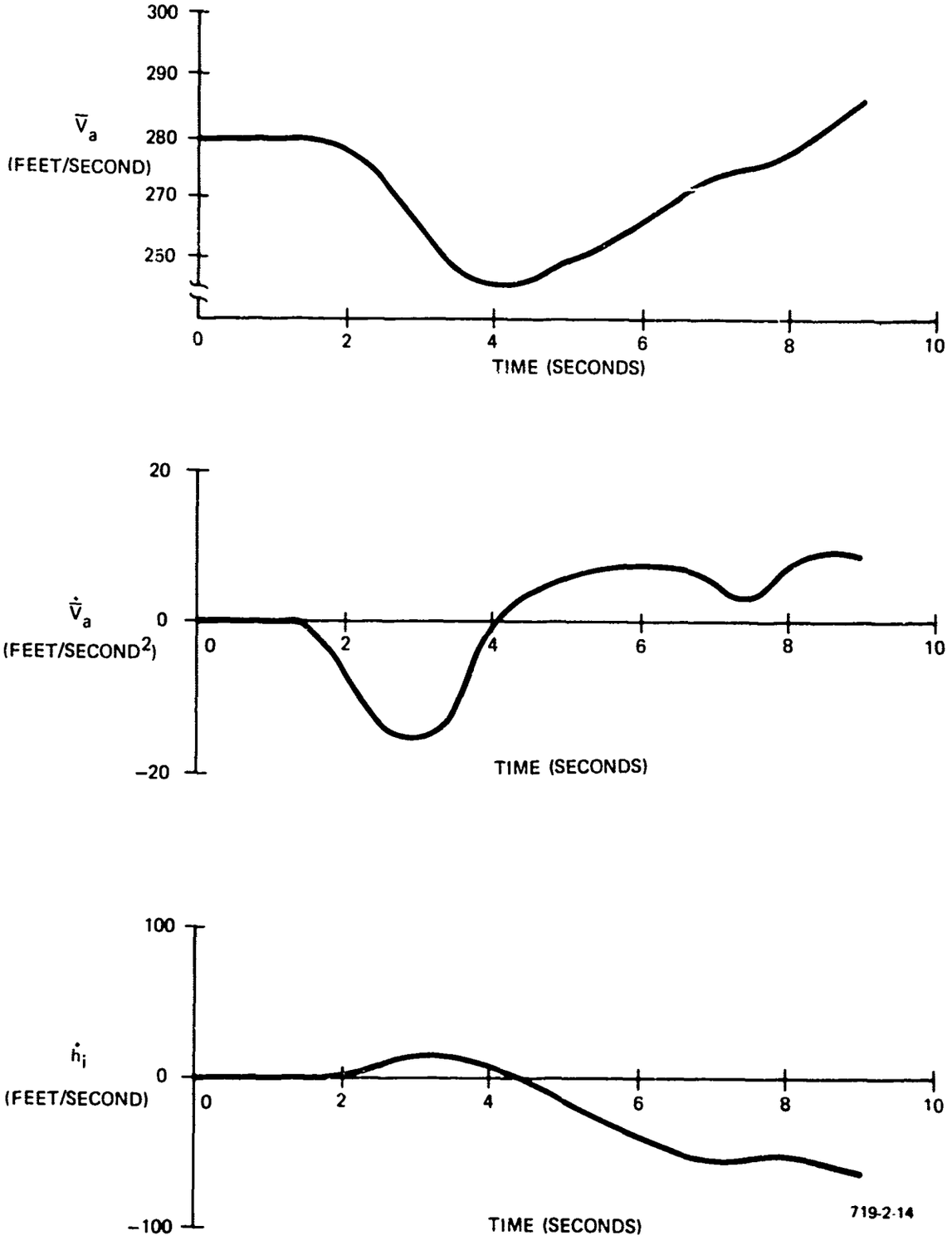


Figure 11 - Concluded
(d) $\bar{V}_a$, $\dot{\bar{V}}_a$, and $\dot{\bar{h}}_i$

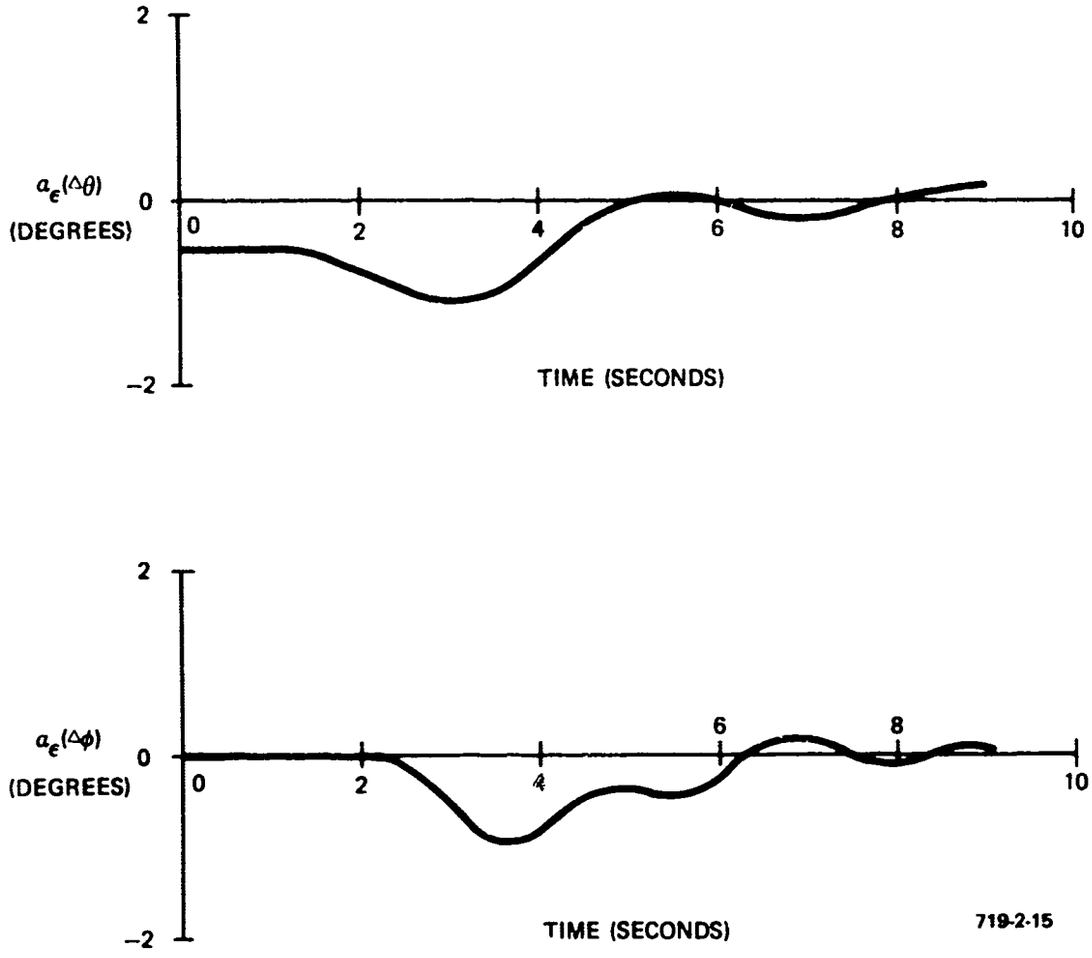


Figure 12
Sensor Errors
Flight Condition 1, Method I

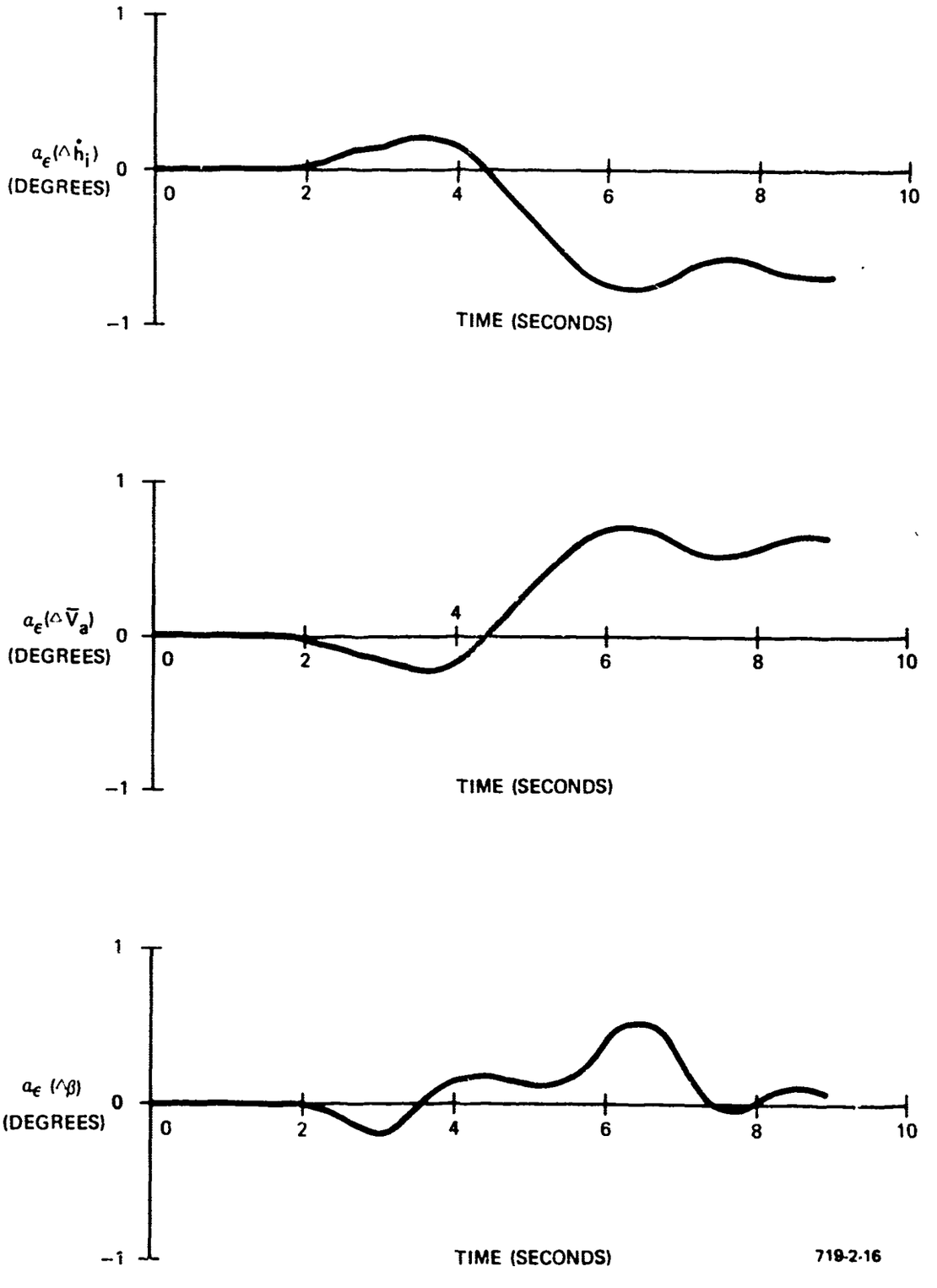


Figure 12 - Concluded

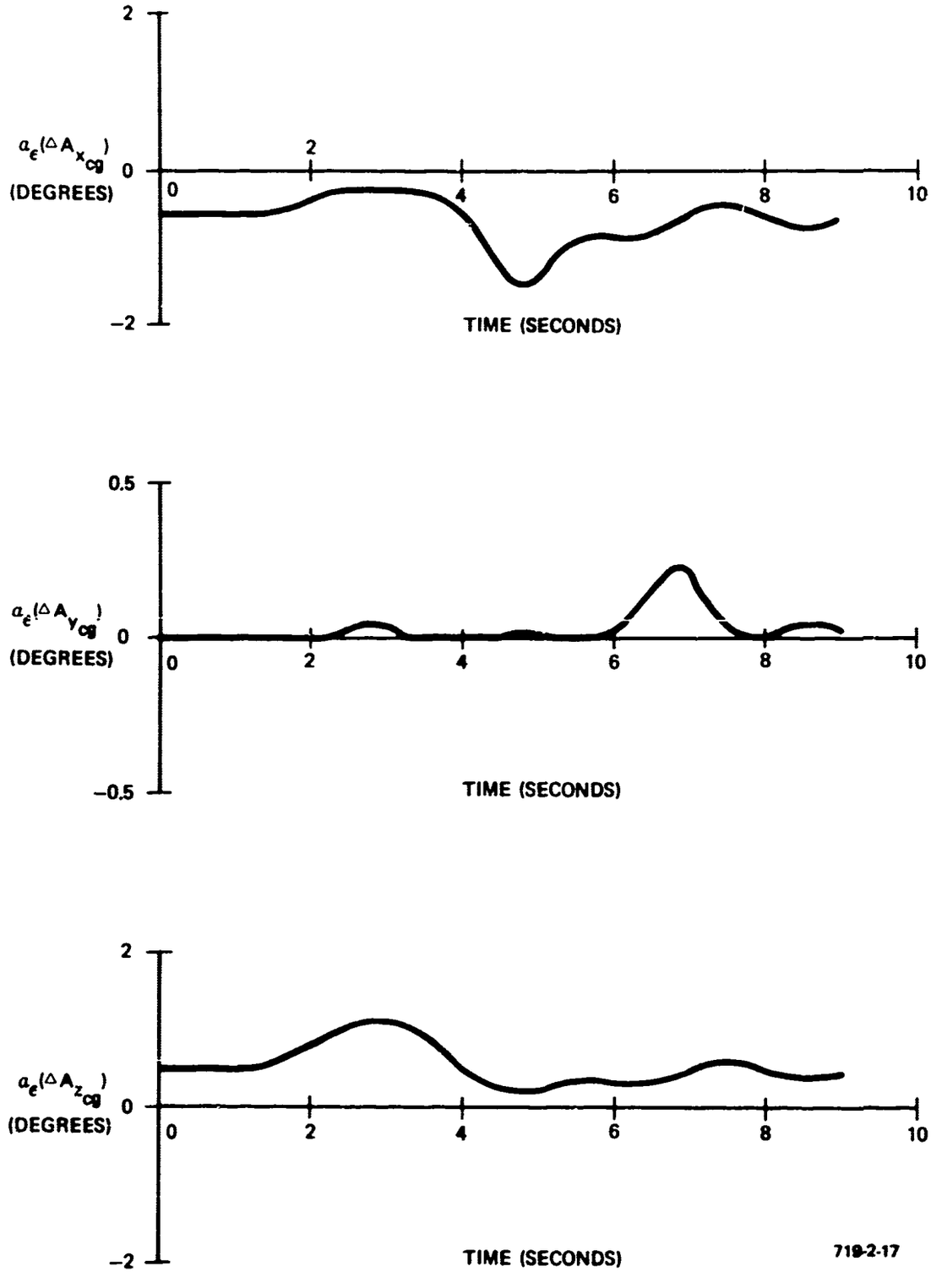


Figure 13
Sensor Errors
Flight Condition 1, Method II

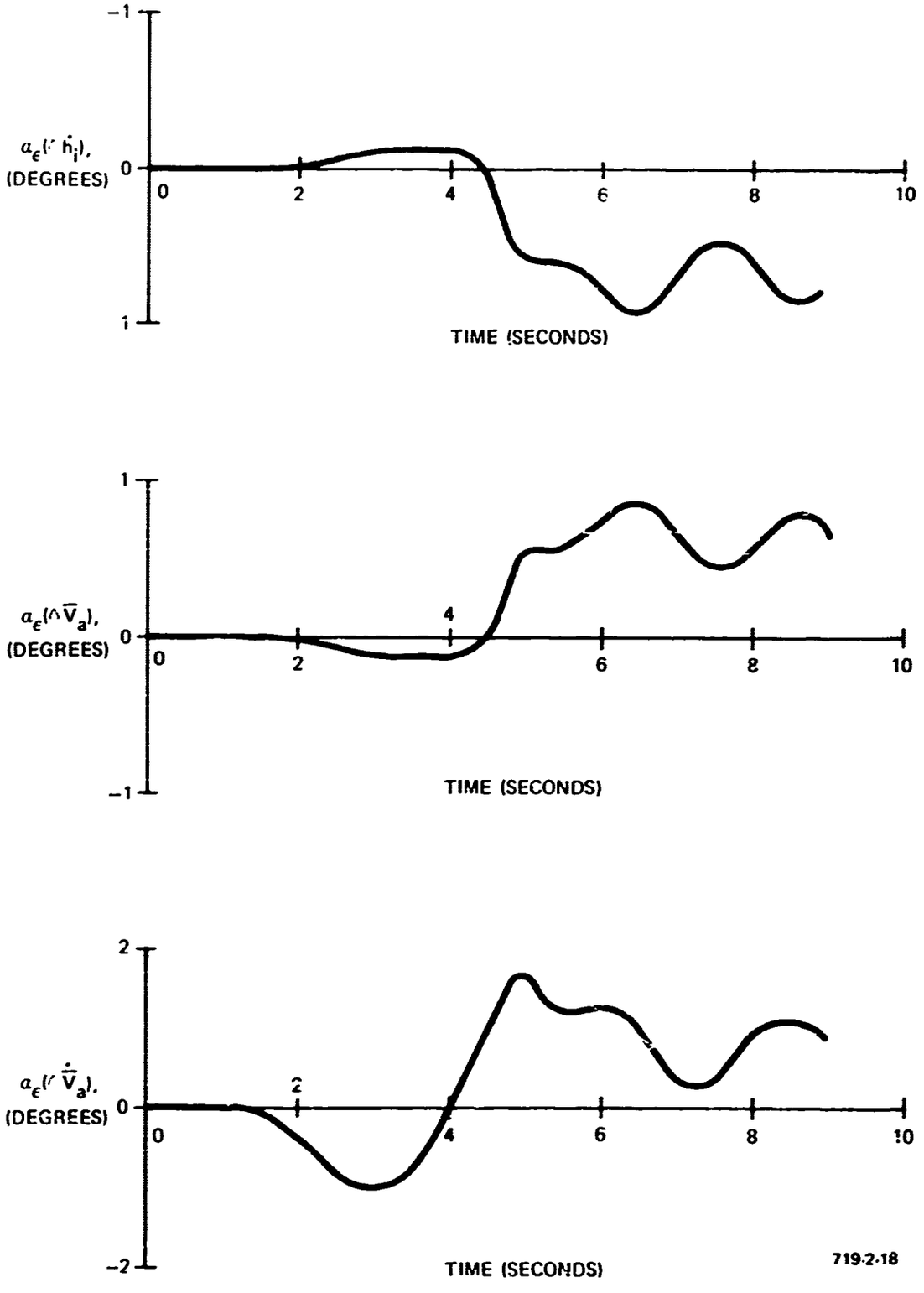


Figure 13 - Continued

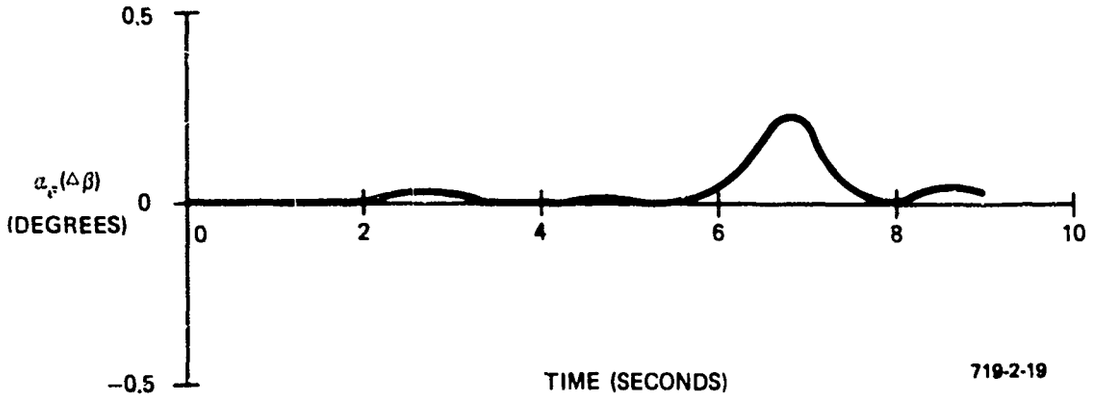


Figure 13 - Concluded

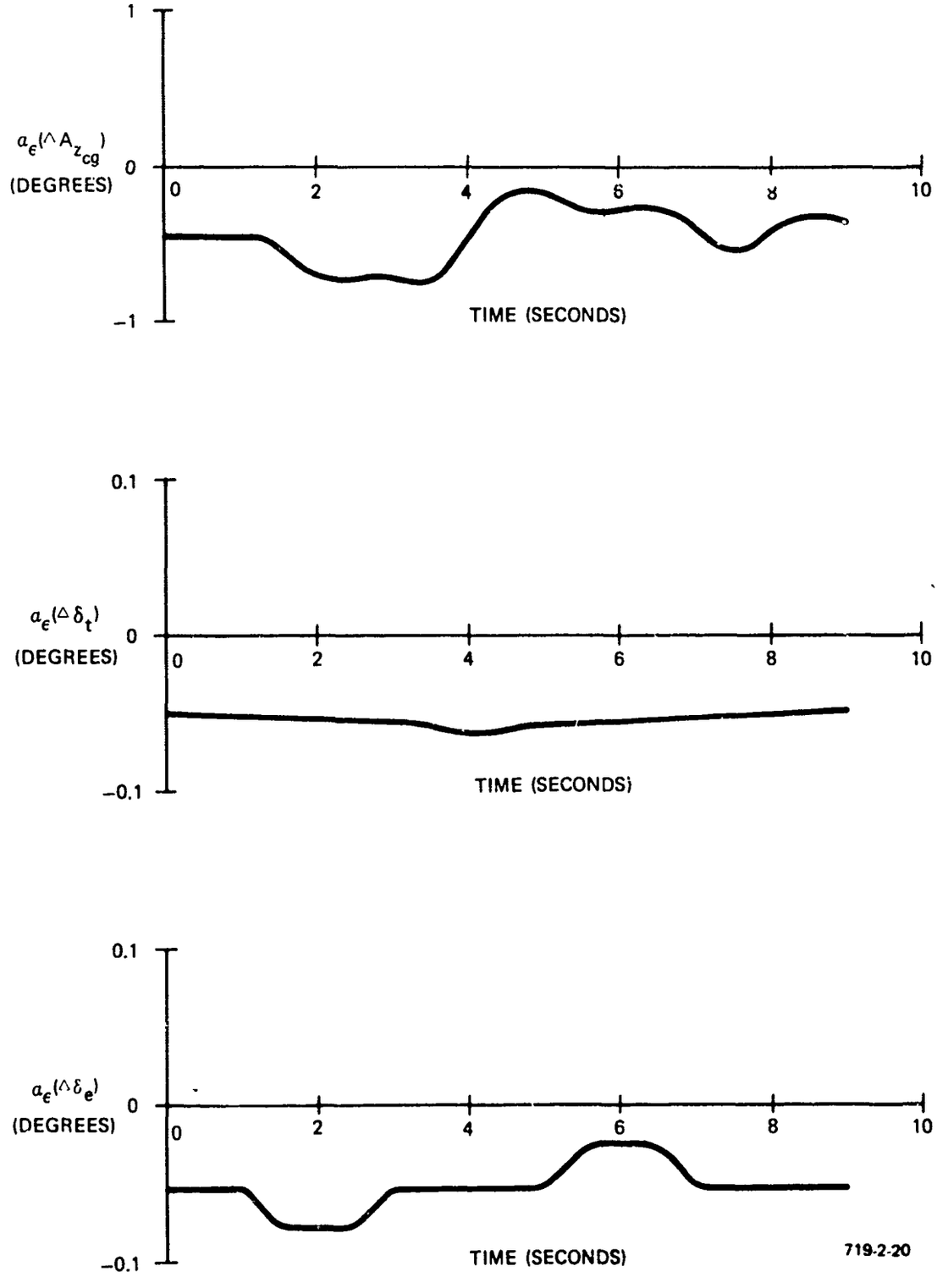


Figure 14
Sensor Errors
Flight Condition 1, Method III

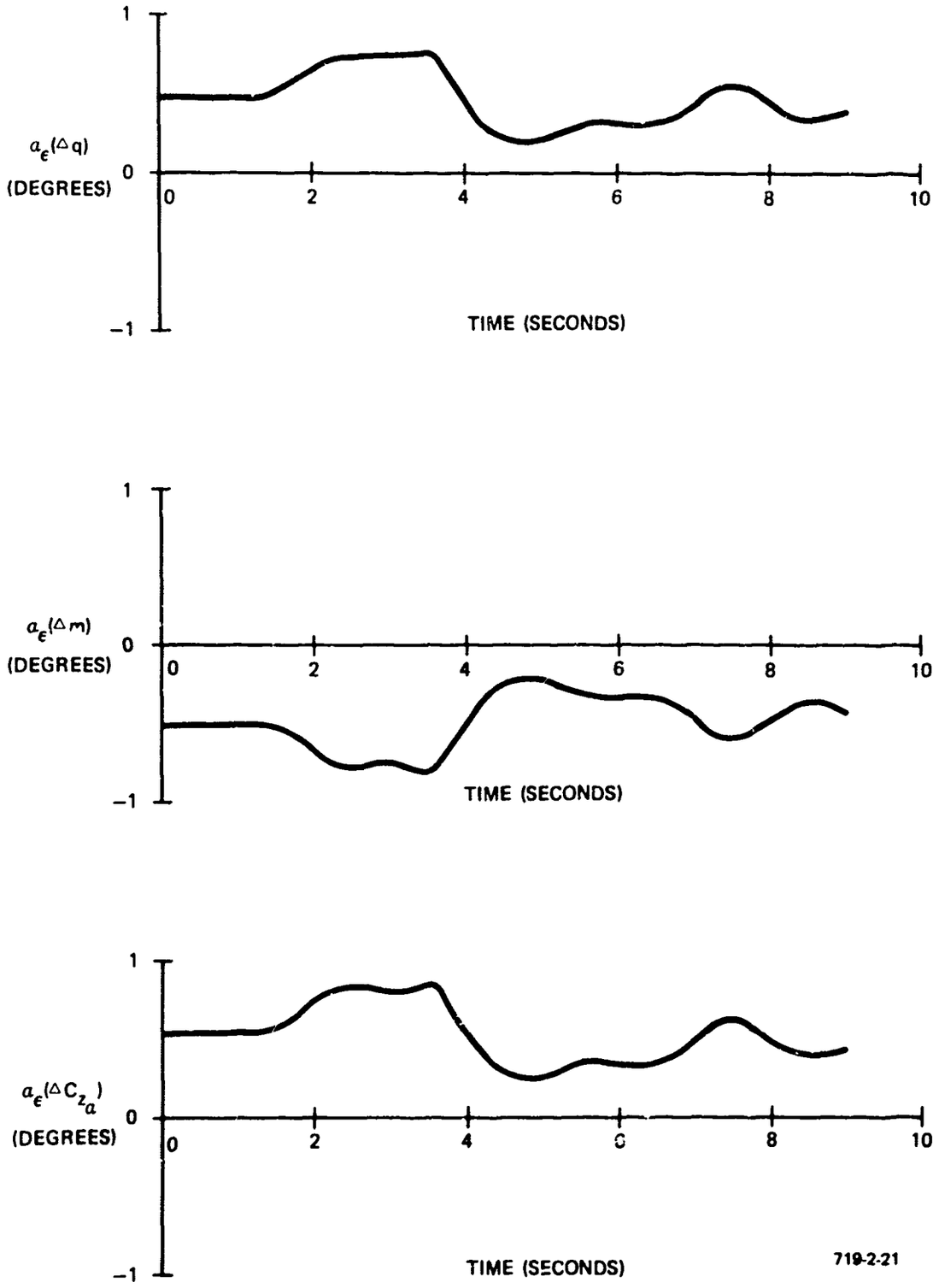


Figure 14 - Concluded

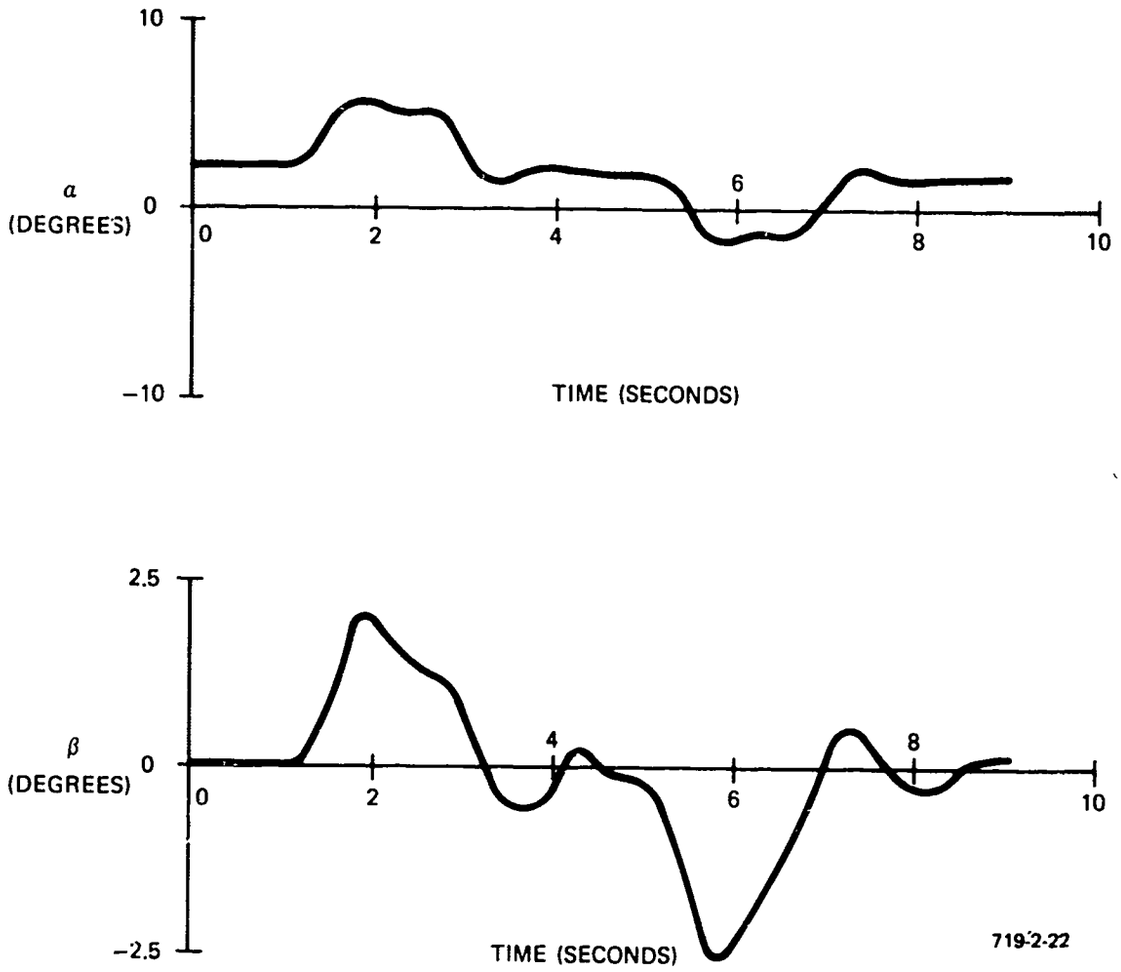


Figure 15
Aircraft Response, Flight Condition 3
(a) α and β

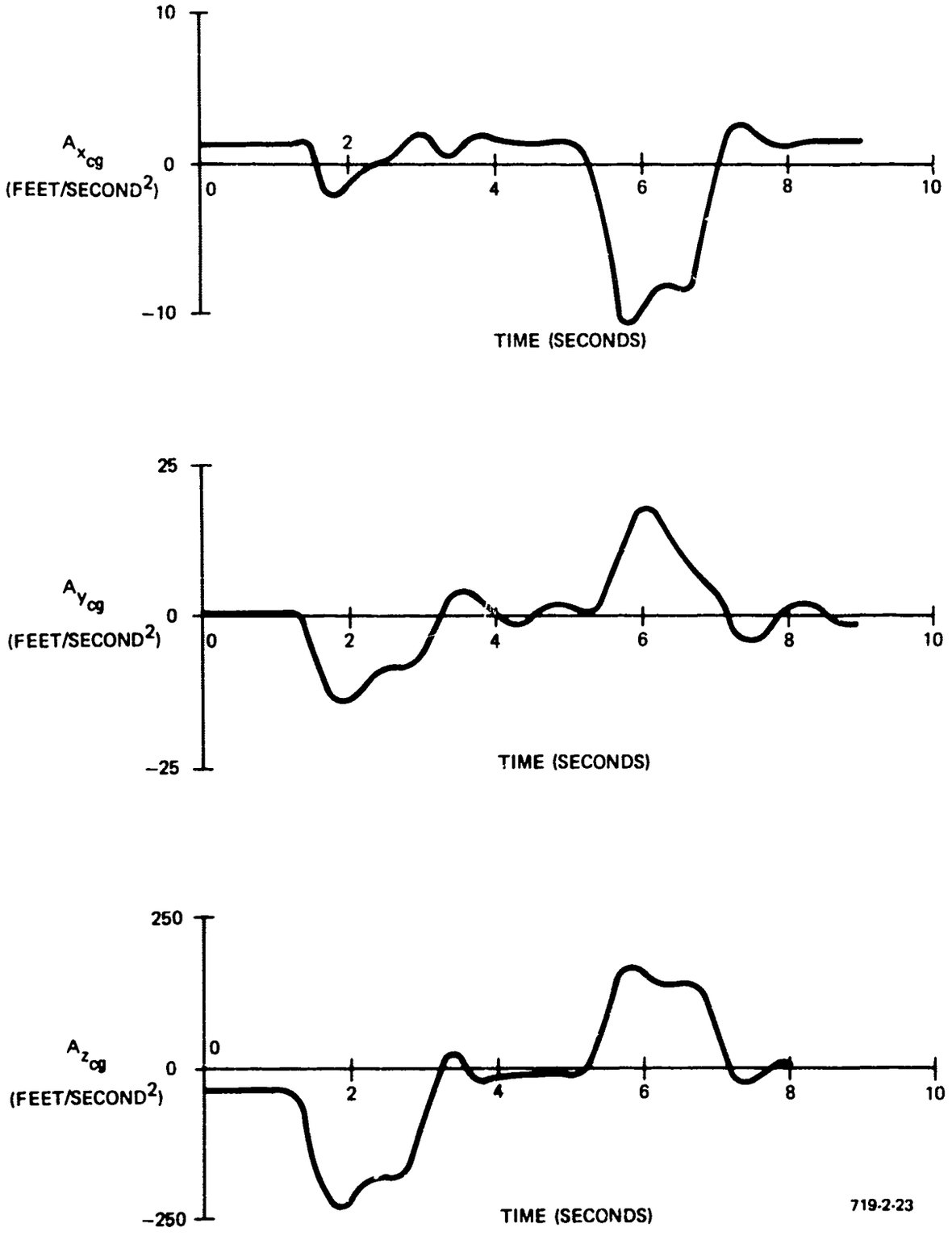


Figure 15 - Continued
(b) $A_{x_{cg}}$, $A_{y_{cg}}$, and $A_{z_{cg}}$

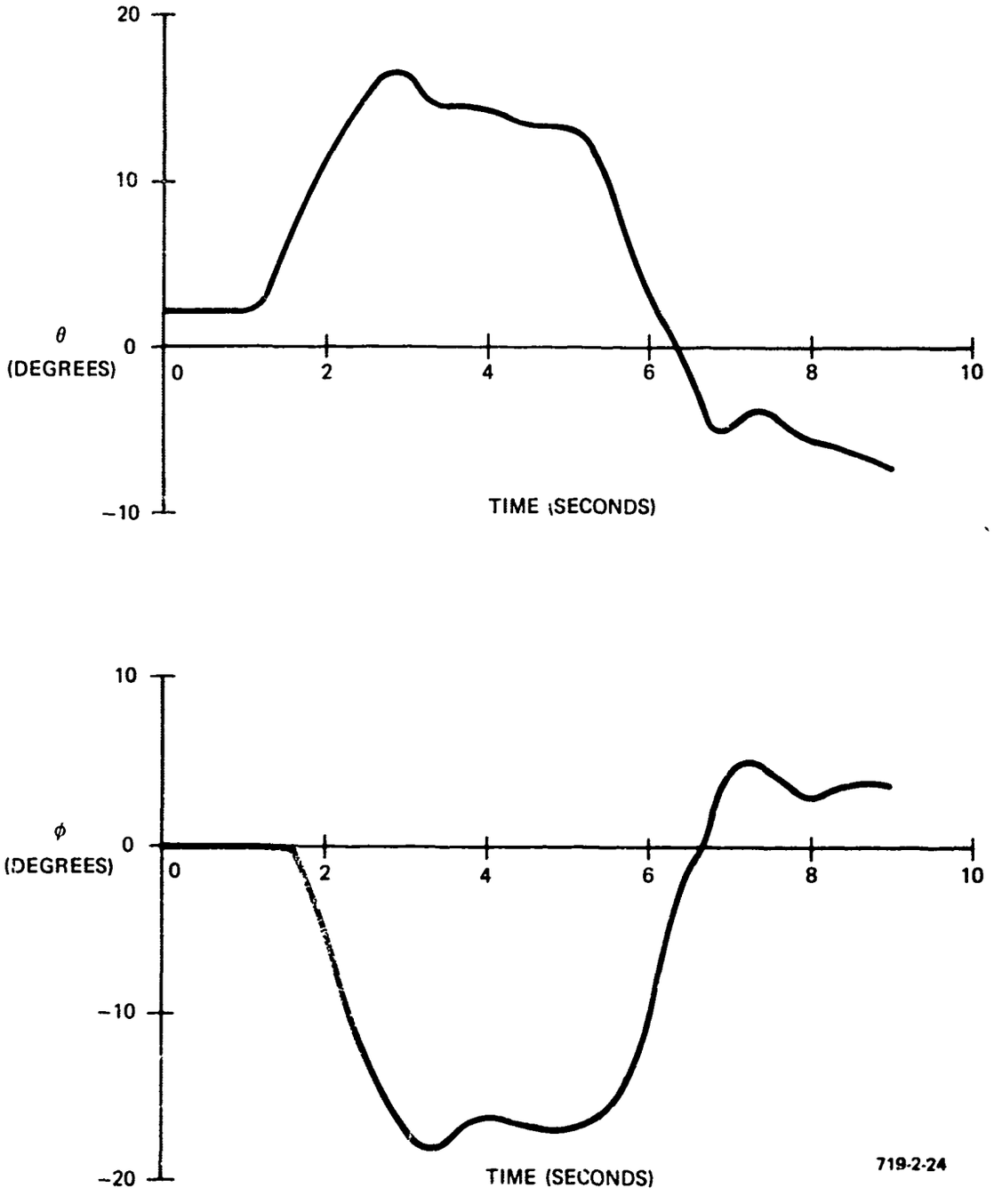


Figure 15 - Continued
(c) θ and Ø

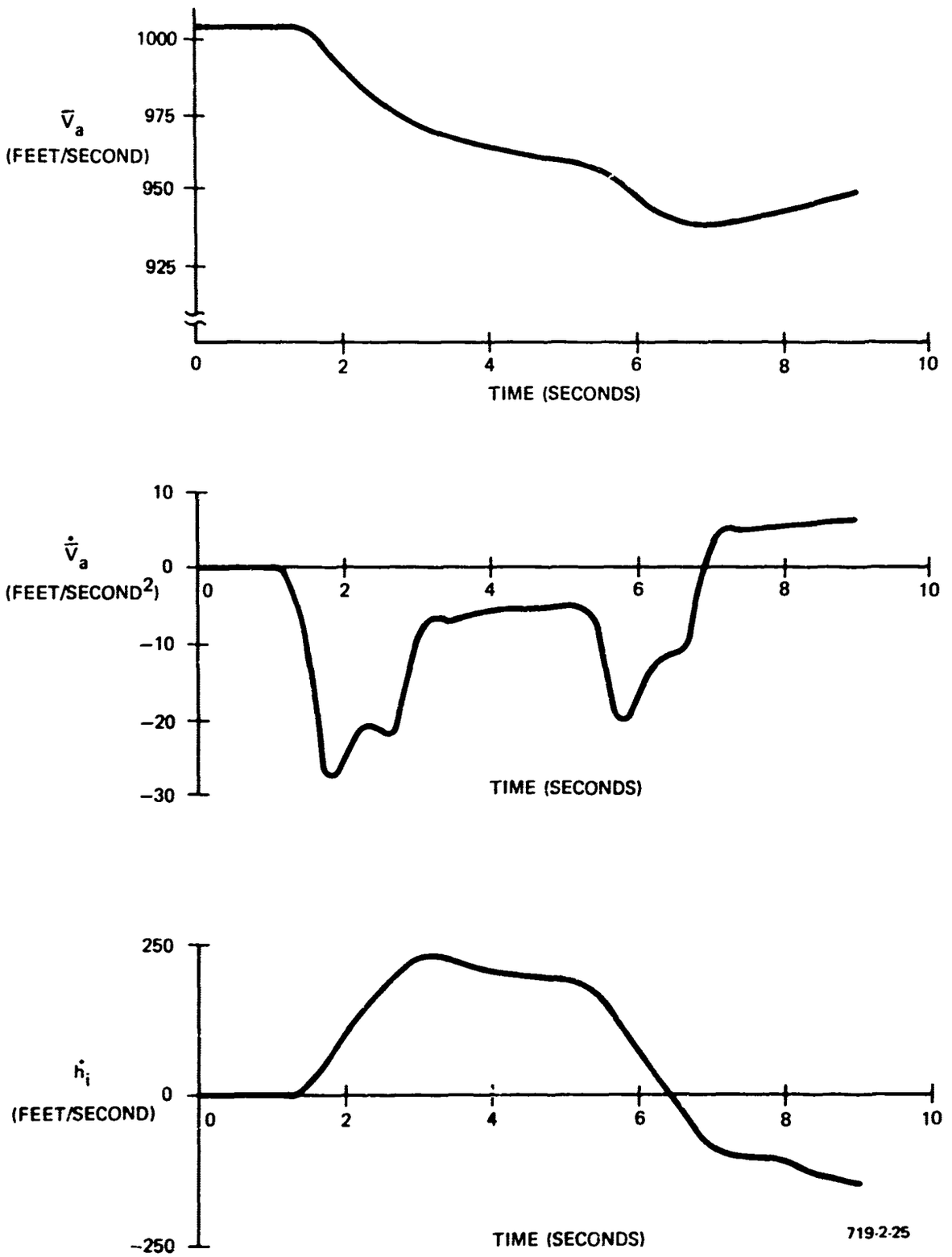


Figure 15 - Concluded
(d) $\bar{V}_a$, $\dot{\bar{V}}_a$, and $\dot{\bar{h}}_i$

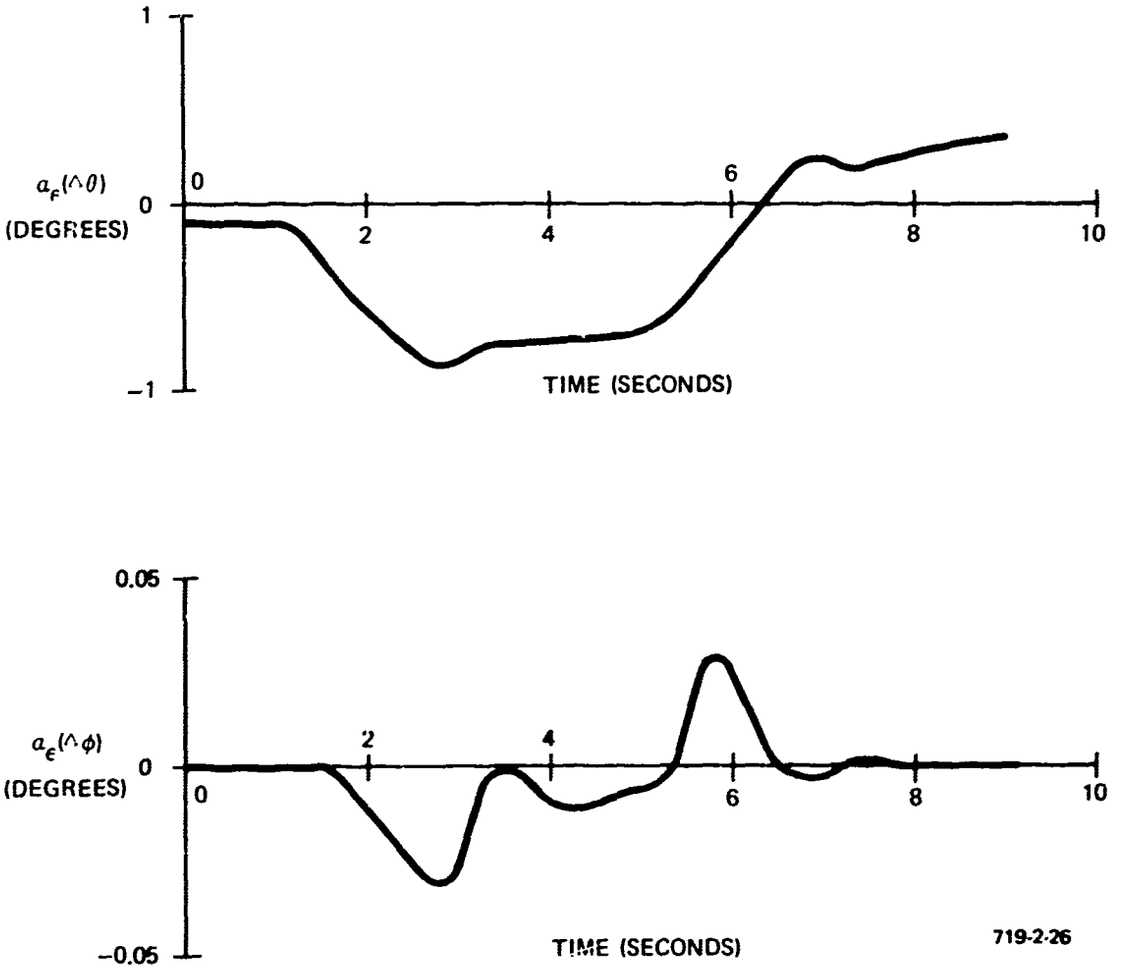


Figure 16
Sensor Errors
Flight Condition 3, Method I

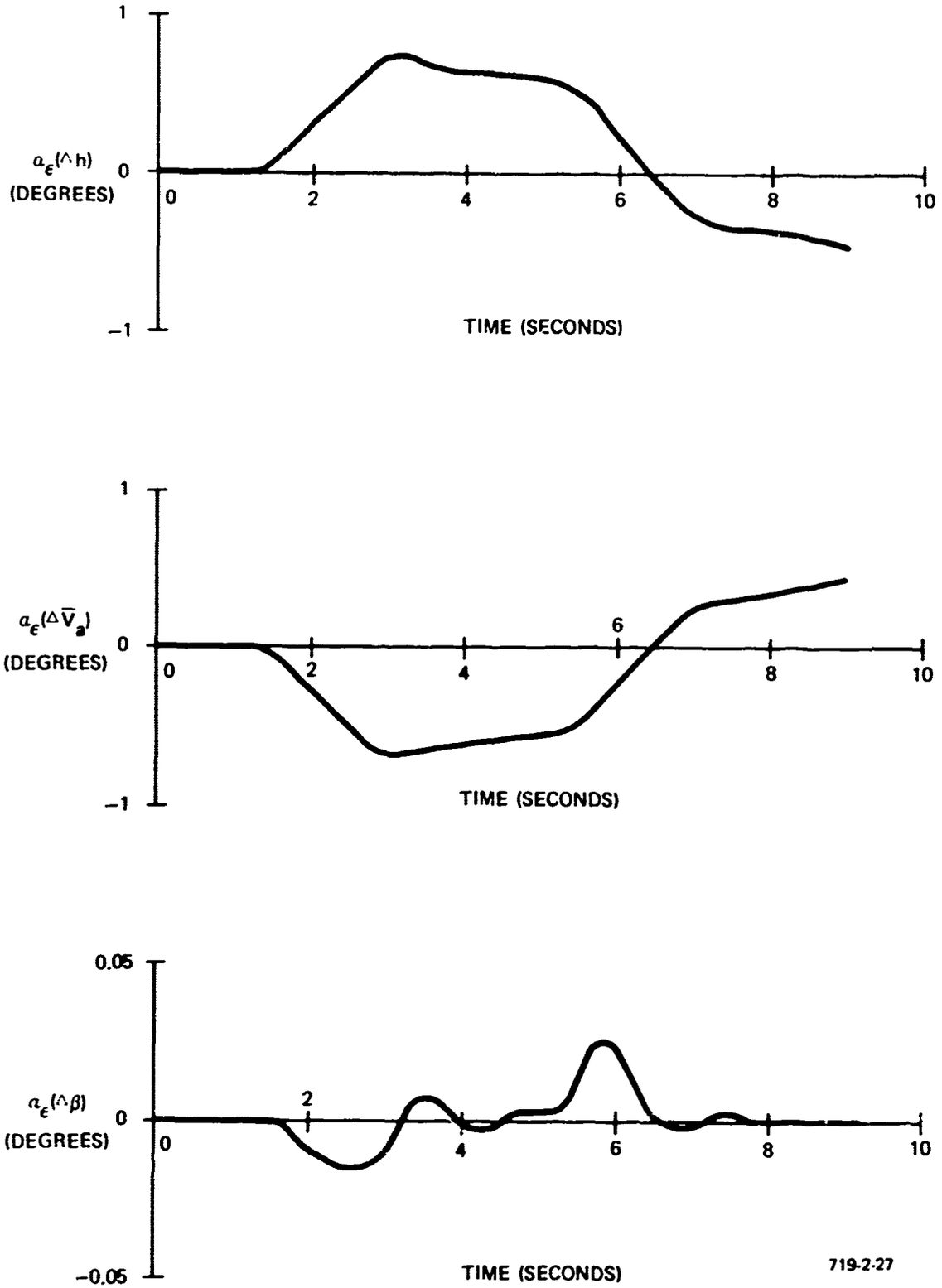


Figure 16 - Concluded

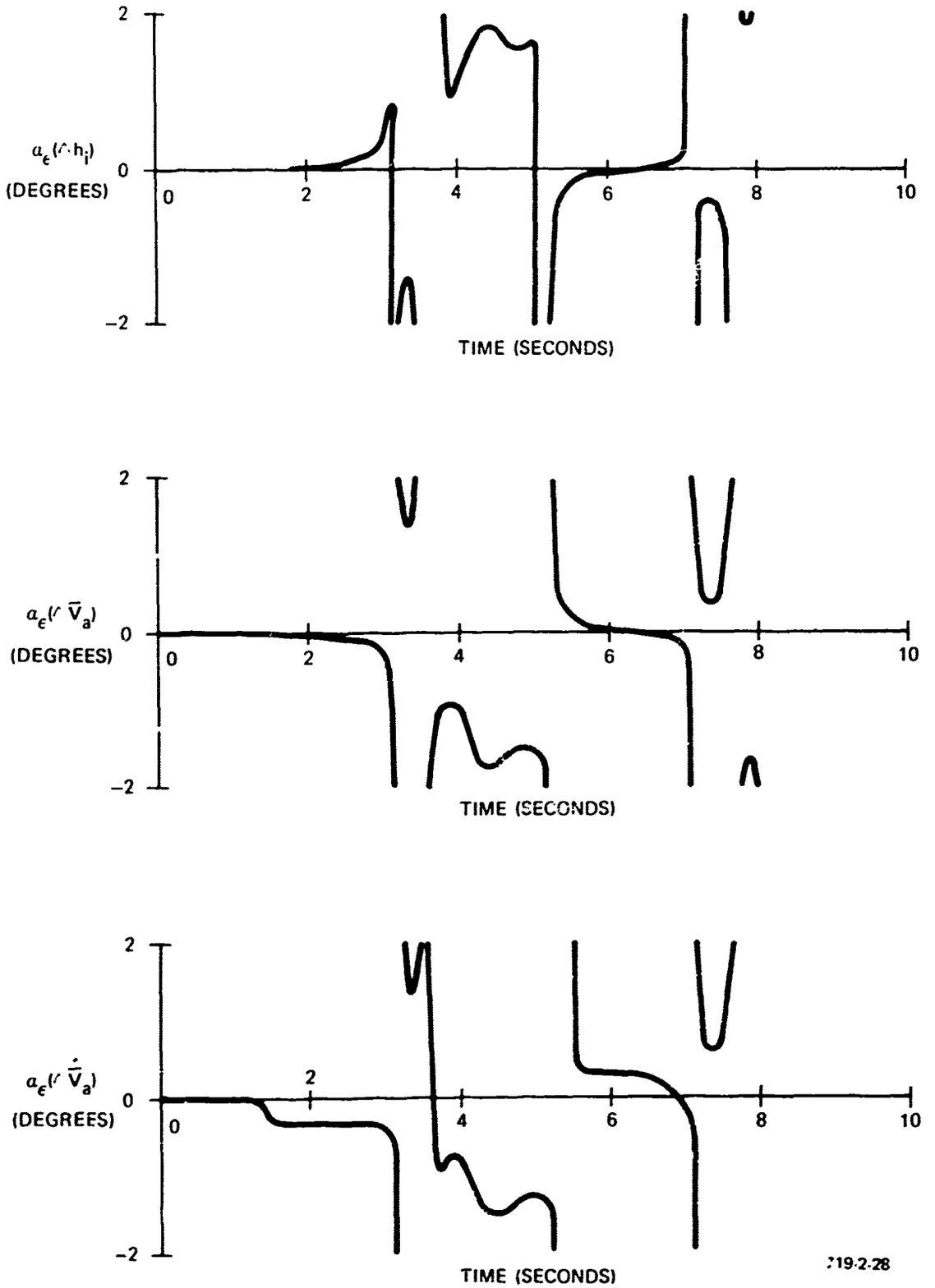


Figure 17
Sensor Errors
Flight Condition 3, Method II

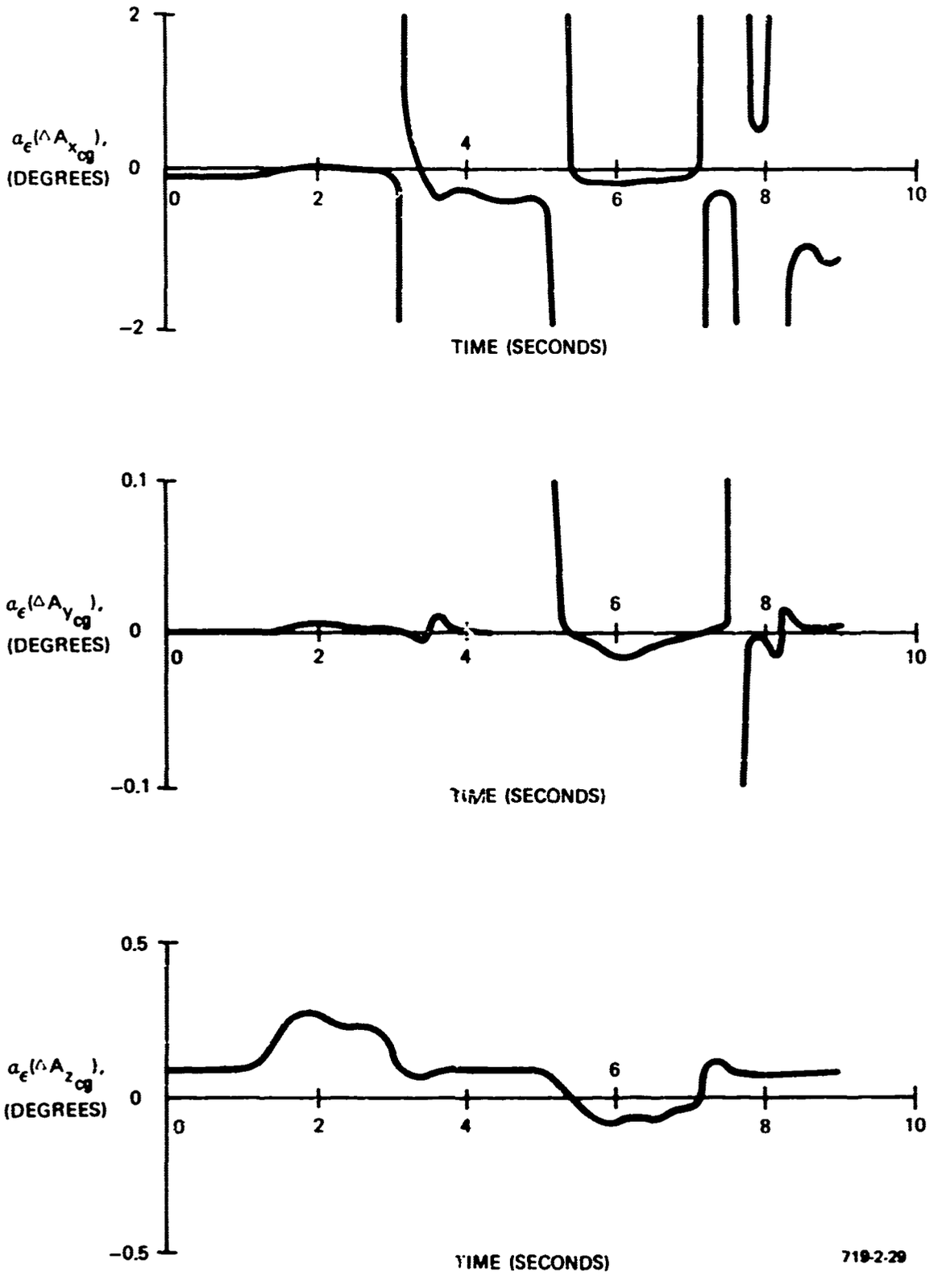


Figure 17 - Continued

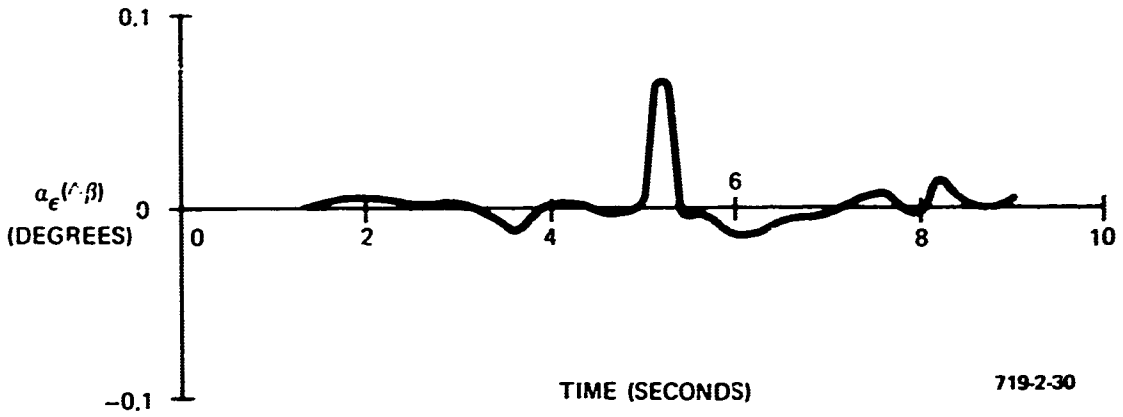


Figure 17 - Concluded

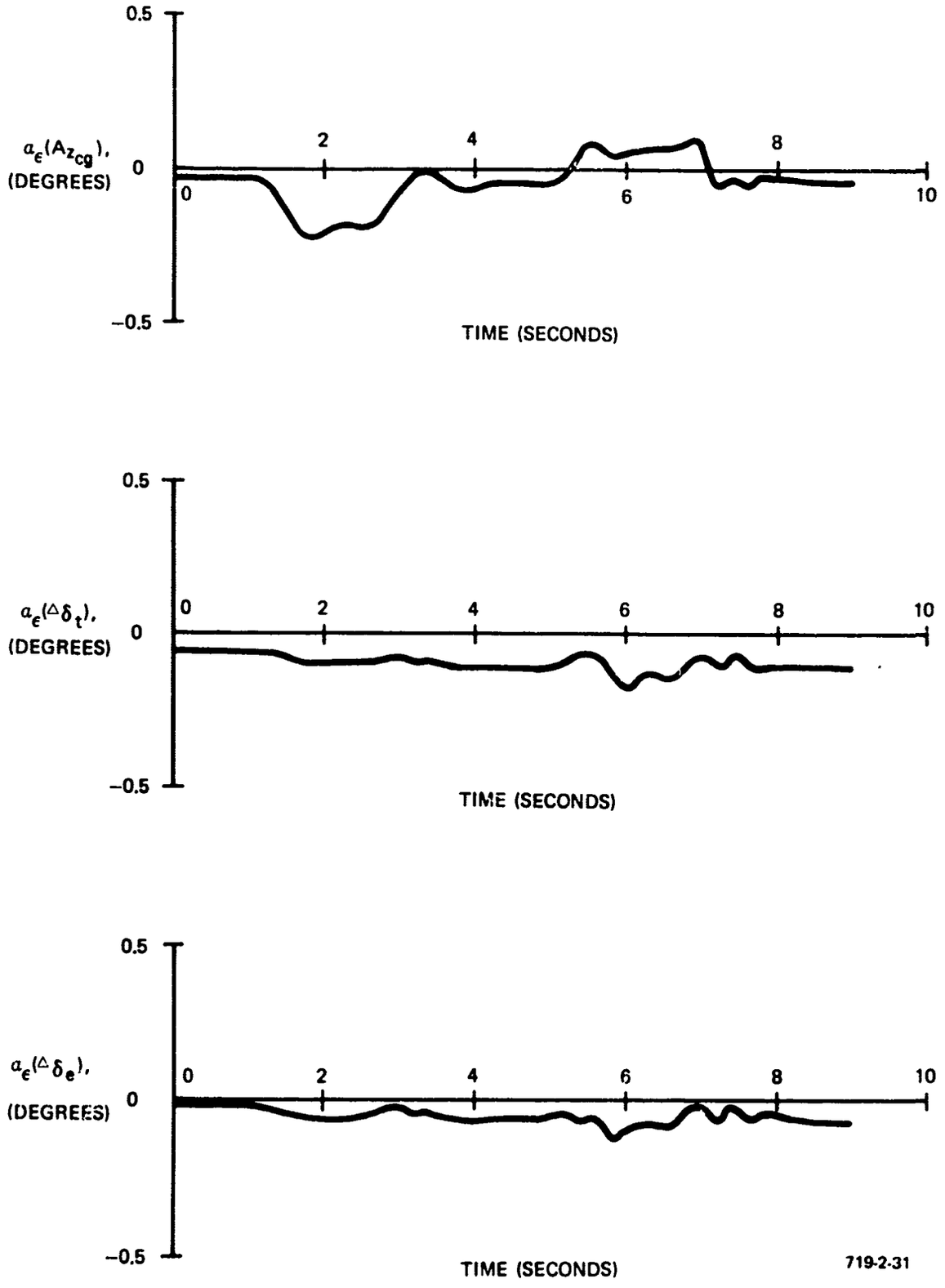


Figure 18
Sensor Errors
Flight Condition 3, Method III

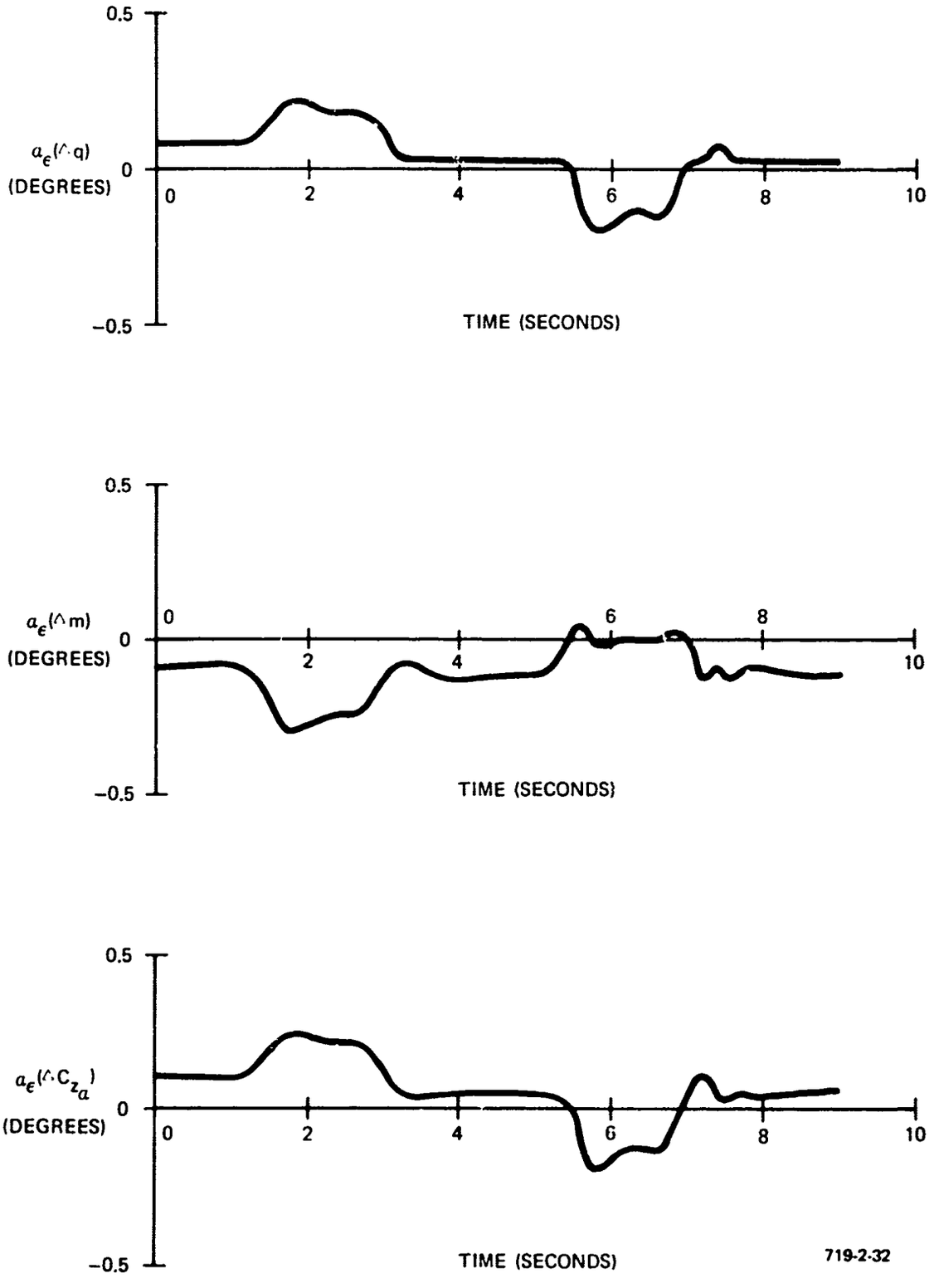


Figure 18 - Concluded

β is the only parameter in Method I that does not require high accuracy. Bank angle is not critical at high speeds but does require accuracy at low speeds. If any parameter can be considered the most critical, it is the vertical reference in the longitudinal axis, θ. For the major portion of most flights, $\alpha \approx \theta$; and if θ is not accurate, α will seldom be correct. Altitude rate and airspeed are important primarily when there is a negative flight path angle and θ is close to zero. In this case, $\gamma \approx \frac{\dot{h}}{\bar{V}_a}$ and $\alpha \approx \gamma$. In view of the magnitude of error caused by each of the sensors, at least 1-percent accuracy is required for all but the side-slip angle measurement. A β measurement of 5 percent should be sufficient in any case.

Two parameters in Method II, Ay_{cg} and β, require only 5-percent accuracy. The remaining sensors, Ax_{cg}, Az_{cg}, $\dot{h}_i$, $\bar{V}_a$, and $\dot{\bar{V}}_a$, all require a 1-percent measurement.

The large errors occurring in Method II with errors in $\dot{h}_i$, $\bar{V}_a$, $\dot{\bar{V}}_a$, Ay_{cg}, and Ax_{cg} are correlated with normal acceleration (Az_{cg}) passing through or close to zero. The discontinuities are explained mathematically by examining Equation (53) when Az_{cg} is 0. Solving for α_i,

$$\alpha_i = \sin^{-1} \frac{\left(A_x^2 - K^2\right)^{1/2}}{A_x} \tag{54}$$

where

$$K = \frac{g\dot{h}_i}{\bar{V}_a} + \dot{\bar{V}}_a - \sin \beta_i \, Ay_{cg} \tag{55}$$

The sensitivity to errors in K is found by taking the partial derivative of α_i.

$$\frac{\partial \alpha_i}{\partial K} = \frac{-1}{\sqrt{Ax^2 - K^2}} \tag{56}$$

When Az_{cg} is close to zero, α_i is small, and from Equation (54), $Ax^2 - K^2$ must be small. This makes the denominator of Equation (56) small, and the sensitivity of α_i to changes in K extremely large. The solution to this problem is to inhibit computation of α_i when using Method II if Az_{cg} is close to zero. This is discussed in Section V.

3.2 Method III Sensitivity Analysis

The independence of the parameters in Method III permits a useful mathematical sensitivity analysis. The particular item of interest is the RMS error that would occur if the sensors were each in error by a specified amount. The RMS error is given by

$$\left[\left(\frac{\partial\alpha}{\partial Az}\,\Delta Az\right)^2+\left(\frac{\partial\alpha}{\partial C_{z_\alpha}}\,\Delta C_{z_\alpha}\right)^2+\left(\frac{\partial\alpha}{\partial T}\,\Delta T\right)^2+\left(\frac{\partial\alpha}{\partial\delta_e}\,\Delta\delta_e\right)^2+\left(\frac{\partial\alpha}{\partial C_{z_{\delta_e}}}\,\Delta C_{z_{\delta_e}}\right)^2+\left(\frac{\partial\alpha}{\partial m}\,\Delta m\right)^2+\left(\frac{\partial\alpha}{\partial q}\,\Delta q\right)^2\right]^{1/2}=\varepsilon_{RMS} \qquad (57)$$

Method III has been evaluated at trim for FC-1 and FC-3 assuming each of the sensors and parameters is in error by 1 percent. The resulting computed angle-of-attack errors were 0.20 degree for α = 11.2 degrees, and 0.034 degree for α = 2.1 degrees. A reasonable limit to the accuracy of Method III seems to be approximately 2 percent, assuming that all parameters can be measured to 1 percent.

4. SENSOR IMPROVEMENTS

The following discussion compares sensor requirements with respect to α computation against state-of-the-art sensor performance characteristics. It is already apparent from the static error and sensitivity analyses that the 0.1-degree accuracy requirement will be difficult to meet with any of the three computation methods. For this reason, a new set of constraints is proposed based on the α measurement requirements for the latest generation of high-performance aircraft. These requirements are listed below.

Aircraft Velocity	Accuracy (degree)	Threshold (degree)	Time Constant (second)
90 to 125 knots	±0.5	0.2	0.075
125 knots to Mach 3.0	±0.2	0.1	0.075

4.1 Vertical Gyro

The vertical gyro provides measurement of Euler angles θ and ϕ. Except in a low-grade, low-cost system, it is inadequate for angle-of-attack computation due to its precession under sustained accelerations.

The long-term precession can be overcome by combining Methods I and II. Method I will provide good, short-term accuracy (using a vertical gyro) while Method II is best in the long term. Thus, a mechanization of both methods --

passing the first through a high pass filter and the second through a low pass filter and then summing the two -- will provide a high-quality inertial α with good dynamic response. Low passing Method II has the added advantage of filtering the $\dot{\bar{V}}_a$ signal, which tends to be noisy. Figure 19 is a block diagram of a simplified mechanization of this combination.

4.2 Inertial Navigation System

Euler angles θ and ϕ can be obtained from an Inertial Navigation System (INS) with sufficient accuracy for the α computation. It is expected that the availability of an INS will increase with each new generation of aircraft. The INS does not suffer from long-term accelerations and is therefore preferred over the vertical gyro in Method I.

4.3 Air Data System

The air data system is required for all three α computation methods. The necessary outputs are $\bar{V}_{air}$, $\dot{\bar{V}}_{air}$, $\dot{h}_i$, Mach, and ρ. The air data system accuracy and frequency response are marginal with respect to α computation. Particularly at the lower speeds, the airspeed inaccuracy can be significant.

At altitude, the lag in the true airspeed measurement can be considerable if the pressure transducers are remote from the source. It would be desirable to compensate for this lag. Let the measurement of true airspeed be $\bar{V}_m$ where

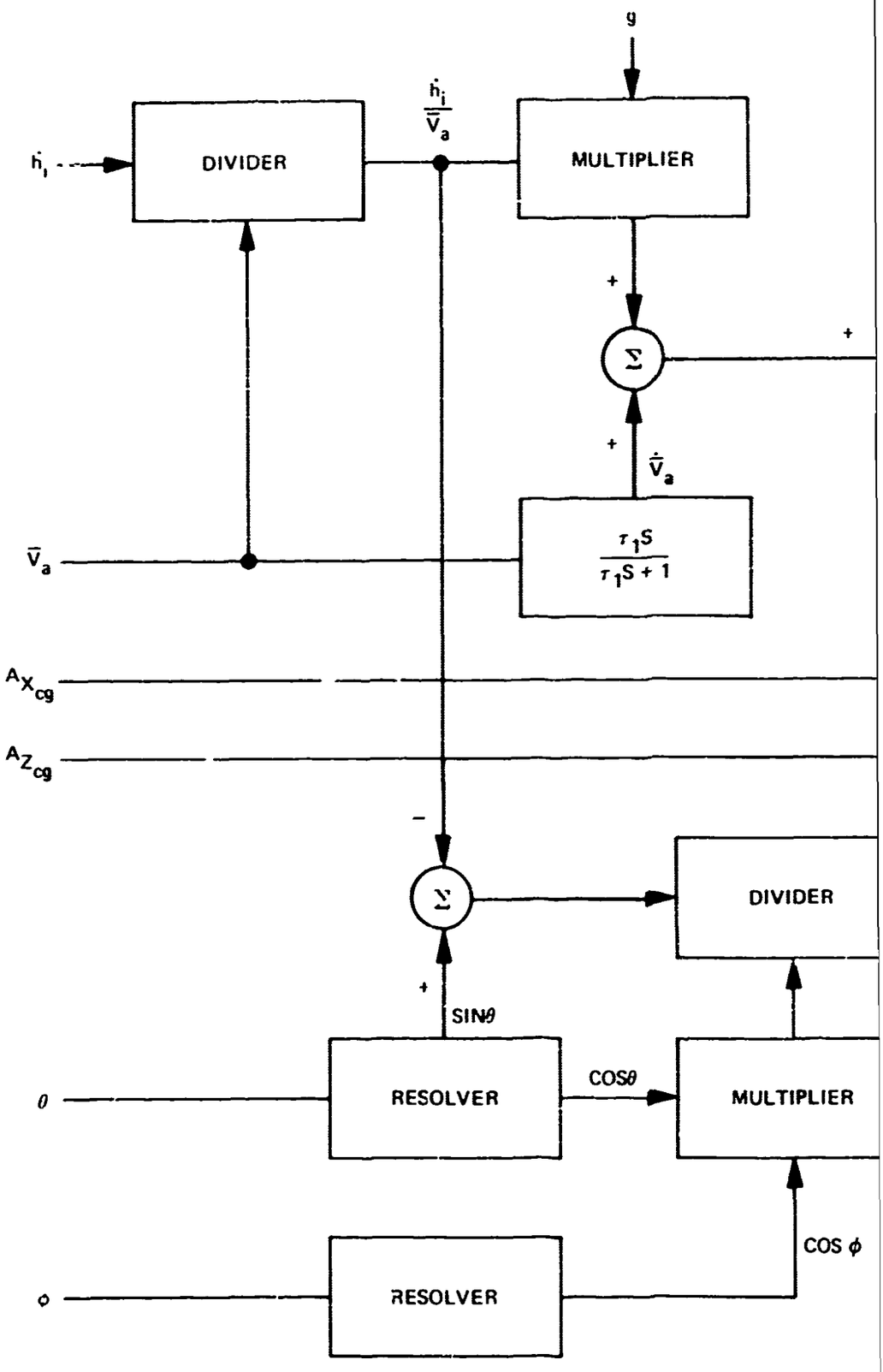
$$\bar{V}_m \doteq \frac{\bar{V}_a}{T(\rho)\ s + 1} \tag{58}$$

The time constant, T, will be principally a function of density for a given installation. The measurement can be partially compensated for plumbing lag, introducing a $\dot{\bar{V}}_a$ term as shown below.

$$\bar{V}_a \approx \bar{V}_m \frac{T_{(\rho)}\ s + 1}{T_1\ s + 1} + T_1 \frac{\left(Ax_{cg} - g \sin\theta\right)}{T_1\ s + 1} \tag{59}$$

Airspeed, altitude rate, Mach number, and air density are standard air data outputs. Airspeed rate is obtained by pseudo differentiation of the airspeed signal.

The accuracy of Methods I and II could be greatly improved in turbulence if air-mass-related altitude rate were available. Unfortunately, for all other applications inertial altitude rate is the required quantity, and the Air Data System (ADS) is designed to minimize the effects of turbulence on $\dot{h}_i$. Enhancing these turbulence effects would require relocating the static pressure part to a turbulence-sensitive position.


g
$\frac{\dot{h}_i}{\bar{V}_a}$
$\dot{h}_i$
DIVIDER
MULTIPLIER
+
Σ
+
+
$\dot{\bar{V}}_a$
$\frac{\tau_1 S}{\tau_1 S + 1}$
$\bar{V}_a$
$A_{X_{cg}}$
$A_{Z_{cg}}$
−
Σ
DIVIDER
+
SINθ
θ
RESOLVER
COSθ
MULTIPLIER
COS φ
φ
RESOLVER

A

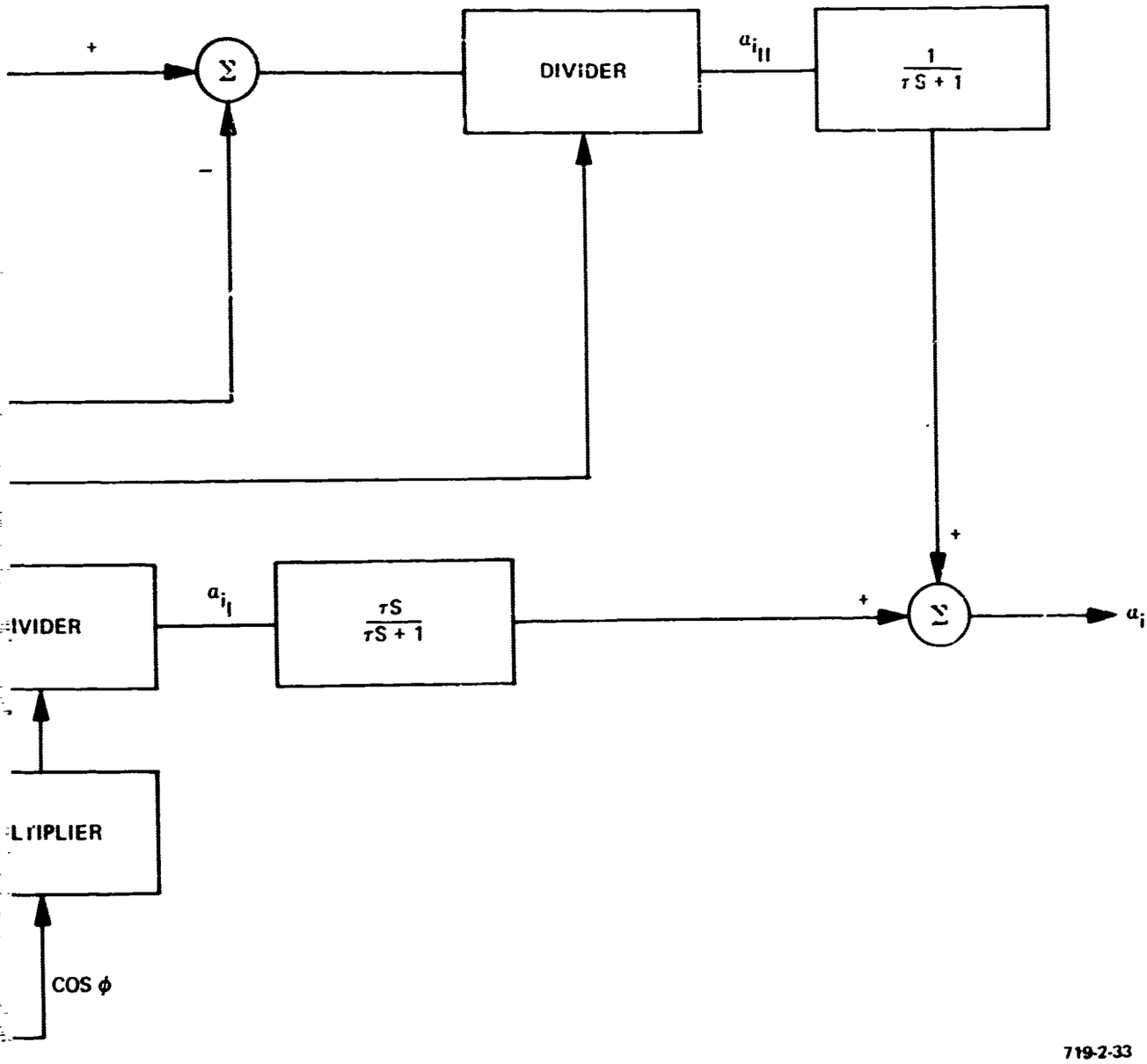


Figure 19
Simplified Combination of Methods I and II

B

Dynamic pressure, $\frac{1}{2} \rho \bar{V}_a^2$, can be calculated within the α computer if it is not available as an ADS output.

4.4 Side-Slip Angle

The measurement of β presents an interesting problem in that a probe or vane must be located in the airstream. If β is not available, Methods I and II can exhibit more than 1.5 degrees of error during conditions of large β and small bank angles or lateral accelerations.

Side-slip can be calculated internally in a manner similar to α computation. The least complex method is the Y-force Equation (4).

$$Ay_{cg} = Y_\beta \beta + Y_{\delta_r} \delta_r$$

The relative simplicity of this equation would recommend its use. However, it has one serious flaw: the value of Y_β may be extremely difficult to obtain and is generally known with poor precision. Nevertheless, the equation still remains the best choice if β is to be calculated.

4.5 Accelerometers

Method II requires accelerometers located at the aircraft center of gravity in all three body axes. Most types of accelerometers will meet the requirements of α computation. One item that deserves particular attention is the misalignment between the accelerometer's sensitive axis and the desired body axis. The error can be geometrically compensated if the angles of misalignment are known. (This subject is well documented in Reference 3.) Additionally, the accelerometers must be mounted as close to the aircraft center of gravity as is practical.

4.6 Elevator Position

Measurement of aircraft elevator position does not present any formidable problems. The measurement is generally available with more than sufficient accuracy on most advanced aircraft.

4.7 Thrust

Measurement of thrust is required only if the thrust line is not parallel to the X body axis. Even slight misalignments may be tolerable since the measurement is not critical. In the event that thrust misalignment is significant, throttle position may provide sufficient accuracy for α computation.

4.8 Mass

No direct measurement of mass exists other than takeoff mass on at least one of the latest generation of cargo aircraft. Mass can be calculated using fuel flow measurements or the lift equation. Use of the lift equation appears to be the most promising technique and will be discussed in Section V. In any system using calculated mass, a takeoff initial condition may have to be inputed manually or through a strain gauge/landing gear measurement system.

5. QUICKENING DEVICES

The term "quickening devices" has been used in conjunction with angle-of-attack systems development by the U.S. Navy. The basic purpose of the quickening device was to minimize the lag in the pilot response during carrier landings. This was accomplished by putting a lead network in the display to reflect the result of the pilot input before the aircraft fully responded. Quoting from the flight evaluation final report*: "The purpose of the concept was to enable the pilot to control α (or analogously airspeed) during the carrier approach by minimizing the inherent time lag in the pilot-airplane-α control loop. The modification was termed 'instantaneous angle-of-attack display (IAAD)' with a 'quickening' characteristic of the display."

As it turned out, quickening devices did not appear to be a promising innovation. Again quoting from the flight test report: "The IAAD is unsuitable for service use from the standpoint of monitoring ease and resultant pilot performance because of three basic discrepancies, i.e., excessive system sensitivity, lack of situation display and inadequate system input design for airplane response characteristics." Further difficulties of quickened α were detailed in the flight test report.

Quickening devices as they relate to angle-of-attack computation do not appear to offer any advantages with respect to system performance.

6. SENSOR STUDY CONCLUSIONS

In general, airframe angle of attack can be accurately computed using currently available on-board sensors. Since the weakest link in the computing system is the ADS, any efforts to improve sensor operation with respect to α computation should be concentrated on air data functions.

---

*Reference 4.

Filtering except for producing airspeed rate is confined to low pass networks designed to limit sensor frequency response and noise output. The airspeed rate signal is derived from the airspeed signal through a lead-lag network.

Ultimately, the sensor requirement will be dictated by the mission requirements with respect to α. Systems not requiring α for weapons computation or gust alleviation may considerably relax the need for high-quality sensors.

SECTION V

MECHANIZATION STUDIES

1. DISCUSSION OF THE EQUATIONS

1.1 Method I, the Euler Angle Equation

The Euler angle equation, (25), has a serious difficulty which limits its usefulness in high-quality, angle-of-attack computation. For accurate computation, a knowledge of side-slip angle (β) is required. Since one of the goals of an angle-of-attack computer is to avoid the need for an α vane, it is unrealistic to assume that a vane is available for β measurement. Therefore, β must be computed, and an accurate computation of β is at least as difficult as computing α. A possibility might be to use Method II, the accelerometer equation, to compute side-slip angle and Method I to compute angle of attack (or vice versa). On examination, however, both Methods I and II reduce, during straight and level flight, to the trivial case

$$0 \sin \beta_i + 0 \cos \beta_i = 0 \tag{60}$$

The most practical method available for computation of β is the Y-force equation, (4). This equation requires a knowledge of mass, the computation of which is quite complex.

It is concluded that Method I is most useful in a simplified system where side-slip angle can be neglected. If it is assumed that α_i is small enough to permit small-angle approximations, the following equation results.

$$\alpha_i = \frac{1}{\cos \theta \cos \phi} \left(\sin \theta - \frac{\dot{h}}{\bar{V}_a} \right) \tag{61}$$

Since the limits on θ and ϕ are ± 75 degrees, there is no problem of division by zero. The airspeed, $\bar{V}_a$, is zero only when the airplane is on the ground and the system is not operational. A block diagram (Figure 20) shows one possible mechanization.

1.2 Method II, the Accelerometer Equation

Some of the comments for Method I also apply to Method II, Equation (33). The computed angle of attack is inertial rather than air-mass related, and the need exists for computation of β. In addition, airspeed rate, which is not normally available from the air data computer, is required. Although it is noisy and somewhat difficult to use, pseudo rate can be derived and used for $\dot{\bar{V}}_a$. True airspeed, $\bar{V}_a$, would be used in mechanization rather than inertial velocity, V, because of availability.

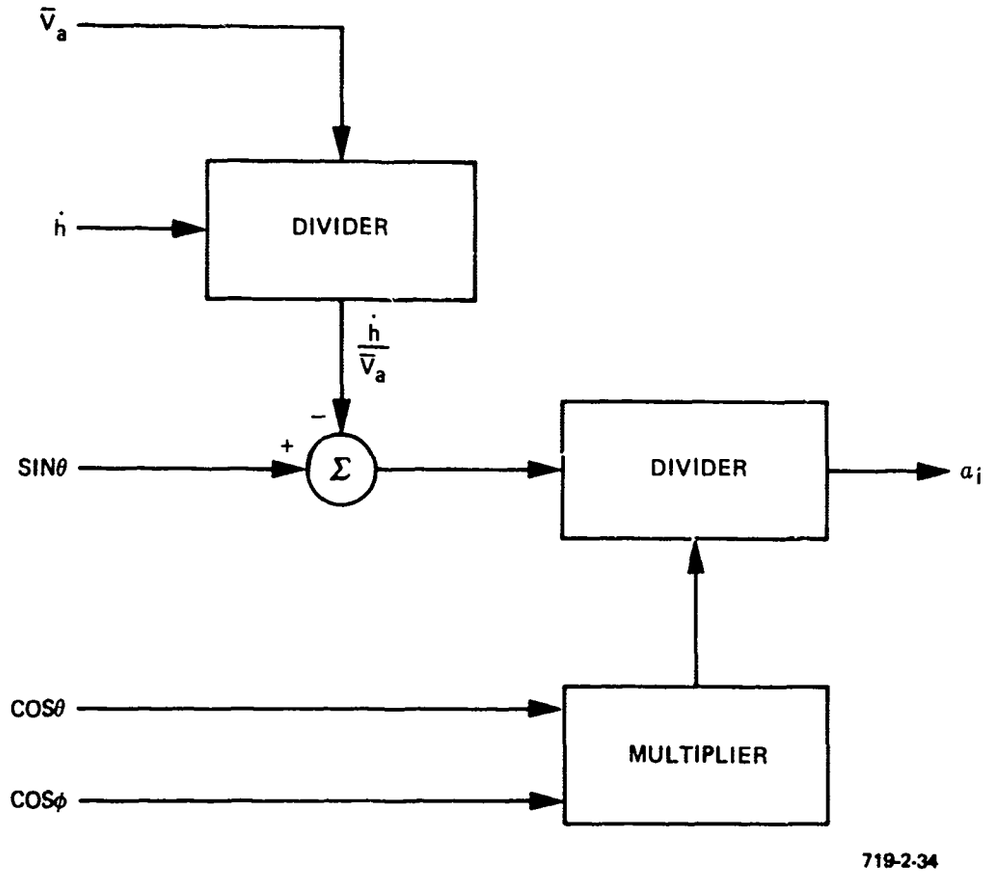


Figure 20
Method J Simplified

For these reasons, Method II is not recommended for high-quality α computation. However, in a simpler, more economical system, a simplified version of this method can be used. By assuming $Ay_{cg} \sin \beta_i = 0$, $\cos \beta_i = 1$, $\sin \alpha_i = \alpha_i$, and $\cos \alpha_i = 1$, the following equation is obtained.

$$\alpha_i = \frac{1}{Az_{cg}} \left(\frac{\dot{V}_a}{V_a} + g \frac{\dot{h}}{V_a} - Ax_{cg} \right) \tag{62}$$

This equation can be solved explicitly for α_i.

A difficulty occurs when attempting to divide by zero. Az_{cg} can go to zero in flight during a zero g maneuver. This can be handled by logically detecting when a low value of Az_{cg} exists and clamping the angle-of-attack output to the value previously computed, or possibly, by clamping to the known zero-lift angle of attack for that flight condition.

Static errors (errors during trimmed flight) are produced by the approximations $\cos \alpha_i = 1$ and $\sin \alpha_i = \alpha_i$. If this approximation is not made, the equation cannot be solved explicitly for α_i, but must be solved implicitly.

Implicit computation is a closed-loop technique whereby the computed variable converges to the proper value through a continuous adjustment. A common example is the performance of division by using a multiplier in a high-gain feedback loop. As with any closed-loop system, stability can be a problem. In the divider example, if the divisor changes sign, static instability results. An approach to implicit computation designed to minimize some error criterion (for example, mean square error) can avoid the stability problem and minimize errors at difficult points such as the indeterminate zero divided by zero.

Assume that the given equation takes the form

$$A(t) - B(\alpha) = 0 \tag{63}$$

where it is desired to evaluate α. In the implicit solution, α will be adjusted to minimize the mean square error between A(t) and B(α). In other words,

$$A(t) - B(\alpha) = e \tag{64}$$

the error to be minimized. Define the mean square error

$$E = e^2 \tag{65}$$

Minimum E occurs when the derivative of E with respect to α equals zero. Considering Figure 21, which shows E and $\partial E/\partial\alpha$ plotted as a function of α in the vicinity of minimum E, note that

- When $\partial E/\partial\alpha$ is negative, α is too small
- When $\partial E/\partial\alpha$ is positive, α is too large
- When $\partial E/\partial\alpha = 0$, α is optimum

It follows that α will seek its optimum value provided

$$\frac{d\alpha}{dt} = -K \frac{\partial E}{\partial \alpha} \tag{66}$$

where K is some positive constant.

Taking the derivative of Equation (65),

$$\frac{\partial E}{\partial \alpha} = 2e \frac{\partial e}{\partial \alpha} \tag{67}$$

and from Equation (64)

$$\frac{\partial e}{\partial \alpha} = - \frac{\partial B(\alpha)}{\partial \alpha} \tag{68}$$

Thus,

$$\frac{d\alpha}{dt} = 2Ke \frac{\partial B(\alpha)}{\partial \alpha} \tag{69}$$

An example of this technique is shown in Figure 22, an implementation of Method II assuming $\beta_i = 0$. Writing the equation in the form of Equation (63),

$$\frac{\dot{gh}}{\bar{V}_a} + \dot{\bar{V}}_a - \left(Az_{cg} \sin \alpha_i + Ax_{cg} \cos \alpha_i \right) = 0 \tag{70}$$

For this,

$$\frac{\partial B(\alpha)}{\partial \alpha} = Az_{cg} \cos \alpha_i - Ax_{cg} \sin \alpha_i \tag{71}$$

The equation mechanized in Figure 22 is

$$\frac{d\alpha i}{dt} = K \left(Az_{cg} \cos \alpha_i - Ax_{cg} \sin \alpha_i \right) \left(\frac{\dot{gh}}{\bar{V}_a} + \dot{\bar{V}}_a - Az_{cg} \sin \alpha_i - Ax_{cg} \cos \alpha_i \right) \tag{72}$$

This technique provides a stable computation when Az_{cg} passes through zero and reverses polarity. Although the accuracy of solution is sensitive to sensor errors at $Az_{cg} = 0$, the computational method is valid.

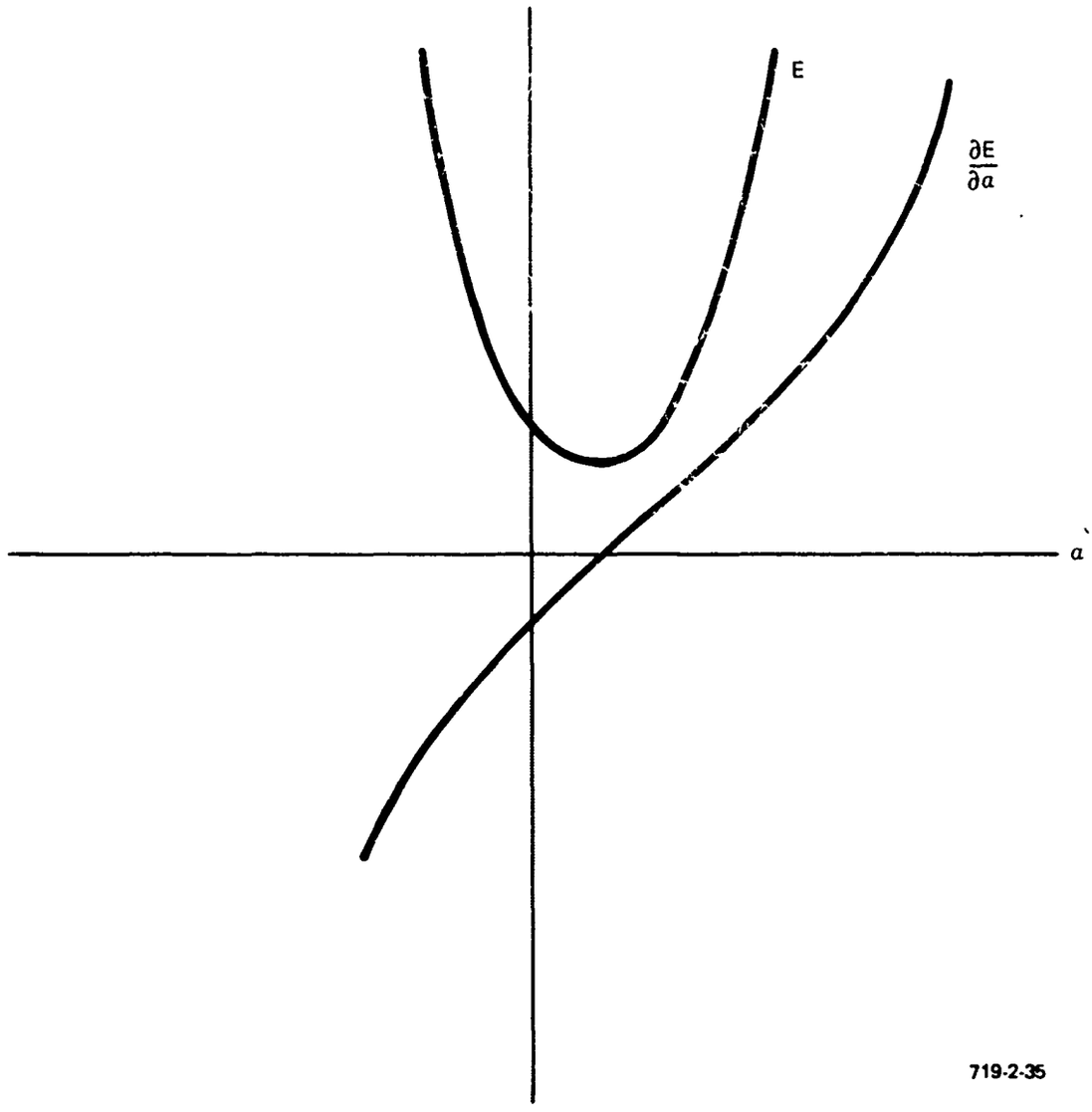
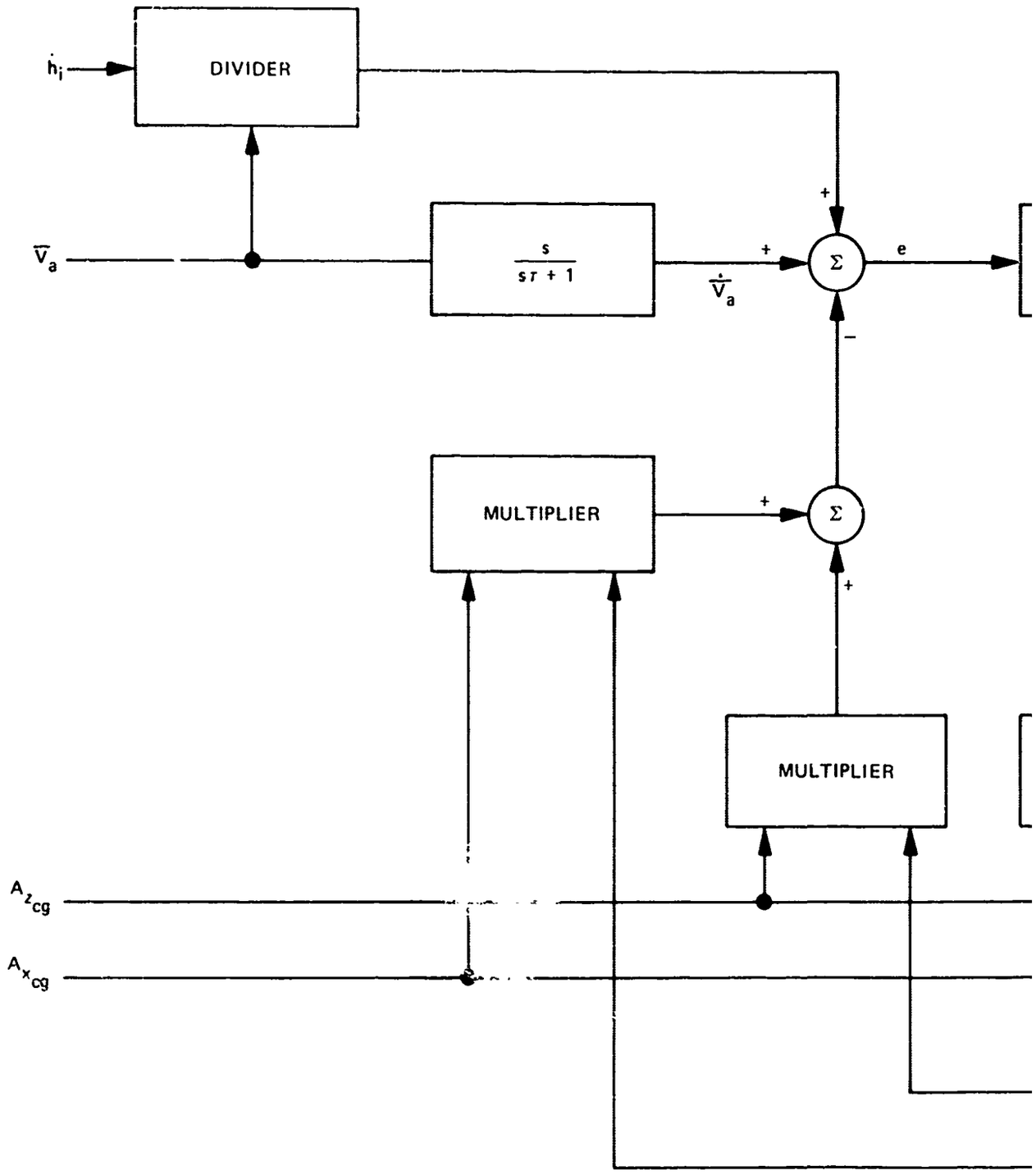


Figure 21
Variation of E and $\frac{\partial E}{\partial \alpha}$ as a Function of α


$\dot{h}_i$
DIVIDER
$\bar{V}_a$
$\frac{s}{s\tau + 1}$
$\dot{\bar{V}}_a$
+
+
e
Σ
−
MULTIPLIER
+
Σ
+
MULTIPLIER
$A_{z_{cg}}$
$A_{x_{cg}}$

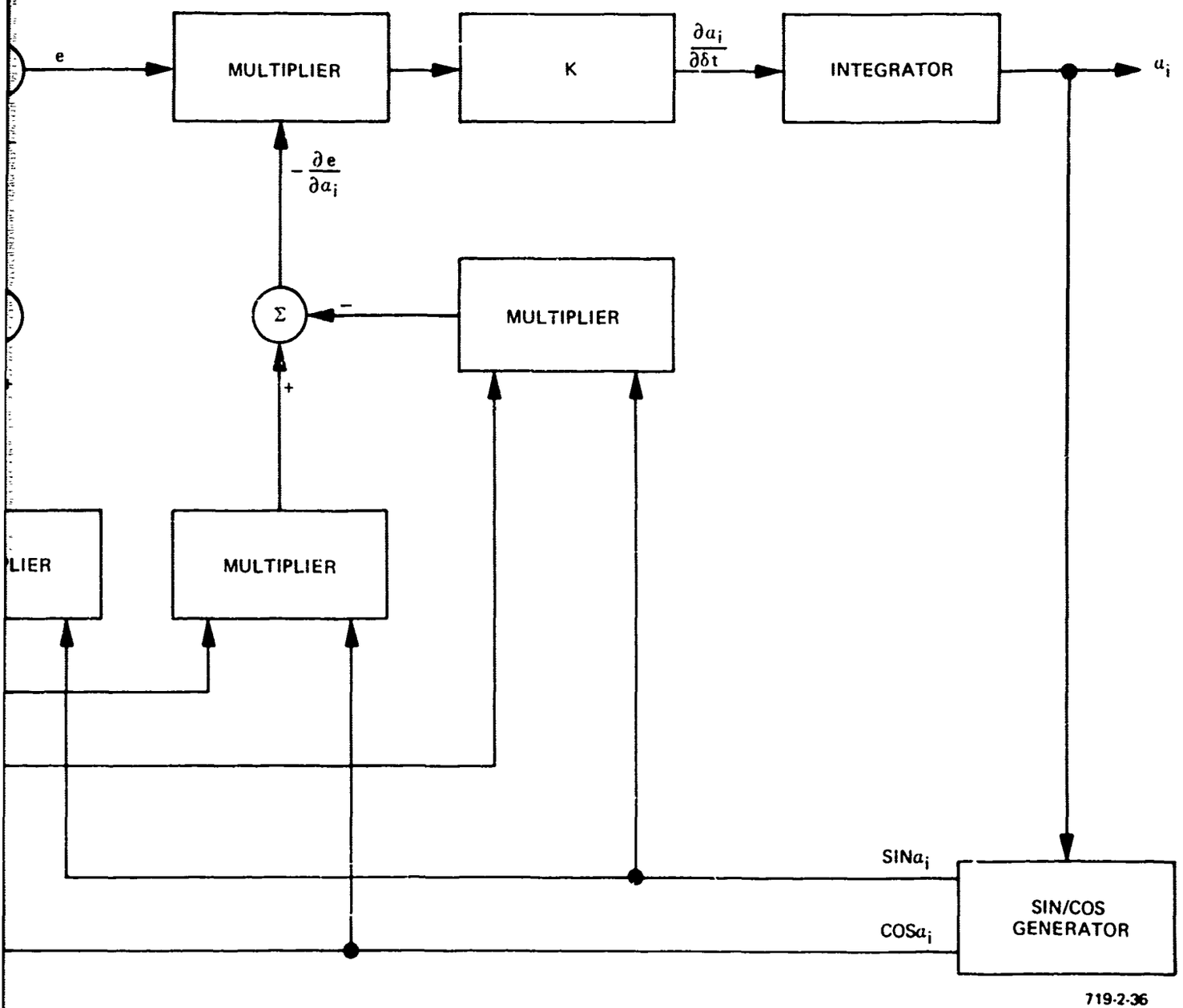


Figure 22
Implicit Computation, Method II

B

1.3 Method III, the Z-Force Equation

Equation (36), the z-force equation, is the best method for high-quality, angle-of-attack computation. As for mechanization, difficulty arises in that aircraft mass must be continuously computed, and aerodynamic coefficients, which vary with Mach and airplane configuration, must be computed.

The most practical method for computing mass is through the z-force equation. However, this equation cannot be used to compute both mass and angle of attack. The lift equation could be used to compute mass, but this is just the z-force equation rotated by the angle α, and is not, for practical purposes, sufficiently independent of the z-force equation to permit derivation of two unknown quantities.

The solution proposed is to use Method I or II to compute inertial angle of attack, and use this in the z-force equation to compute mass. Then with the z-force equation, a good, high-quality, dynamic angle of attack can be computed. Statically, the resulting angle of attack can be no better than the previously computed, independent angle of attack used in computing mass. Therefore, this α_i must be quite accurate statically.

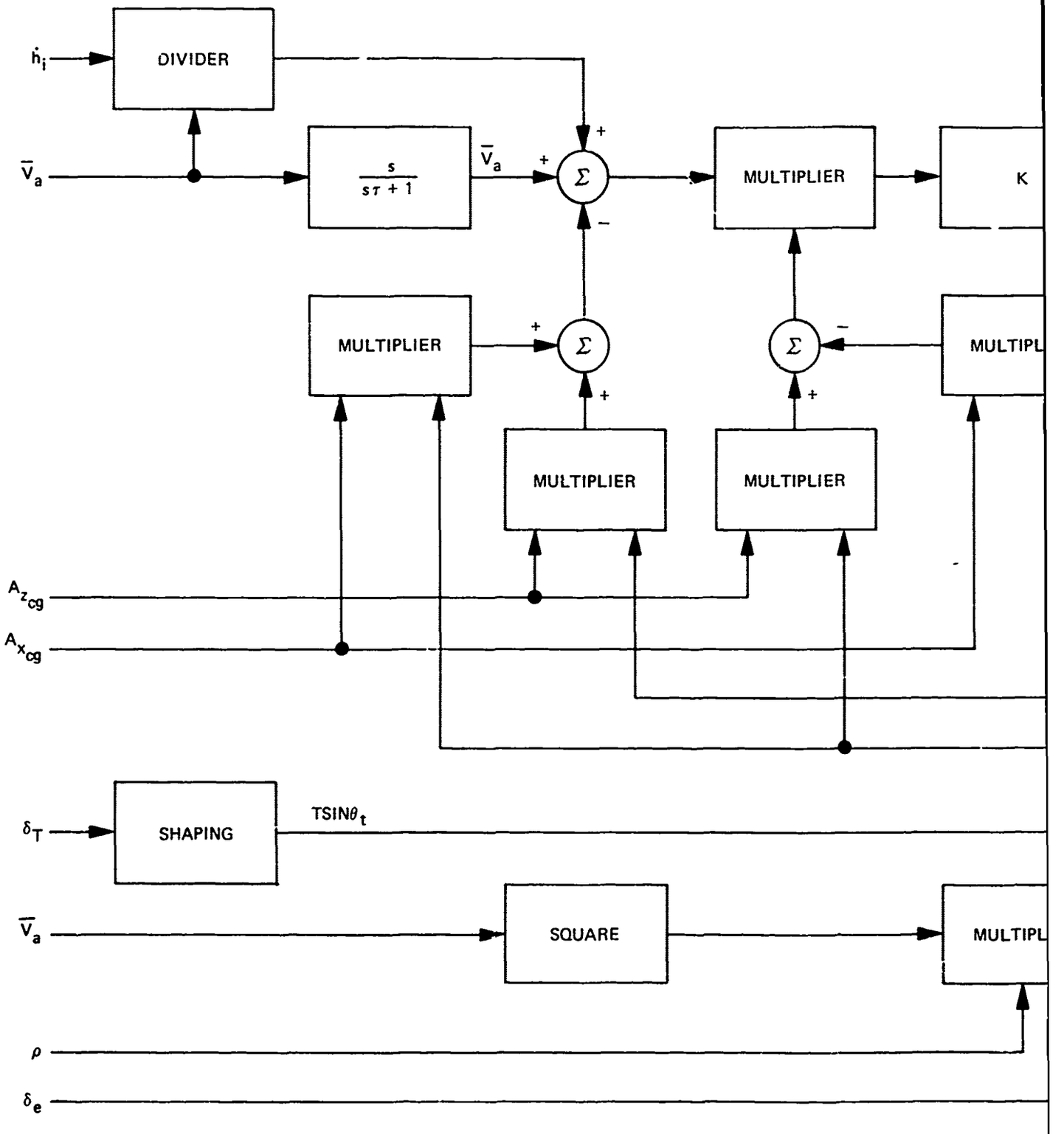
With Method II used to compute α_i, the simplifying assumptions $Ay_{cg} \sin \beta_i = 0$ and $\cos \beta_i = 1$ are permissible because they are dynamic terms. However, $\sin \alpha_i$ and $\cos \alpha_i$ must be computed, or at least more closely approximated than simply $\sin \alpha_i = \alpha_i$ and $\cos \alpha_i = 1$. As shown in subsequent paragraphs, if a Digital Differential Analyzer (DDA) is used for the computations, derivation of the sine and cosine is a simple matter. But if a special purpose, whole-number computer or an analog computer is used, the approximations $\cos \alpha = 1 - a^2/2$ and $\sin \alpha = \alpha - \alpha^3/6$ are easier to mechanize.

Figure 23 shows the block diagram of this approach. The complexity is such that it is feasible to consider special purpose digital computing methods to mechanize the system. Indeed, the accuracy required to obtain a maximum error of 0.1 degree when α ranges from 0 to 20 degrees is better than 0.5 percent. This is not achievable using analog circuitry at reasonable cost.

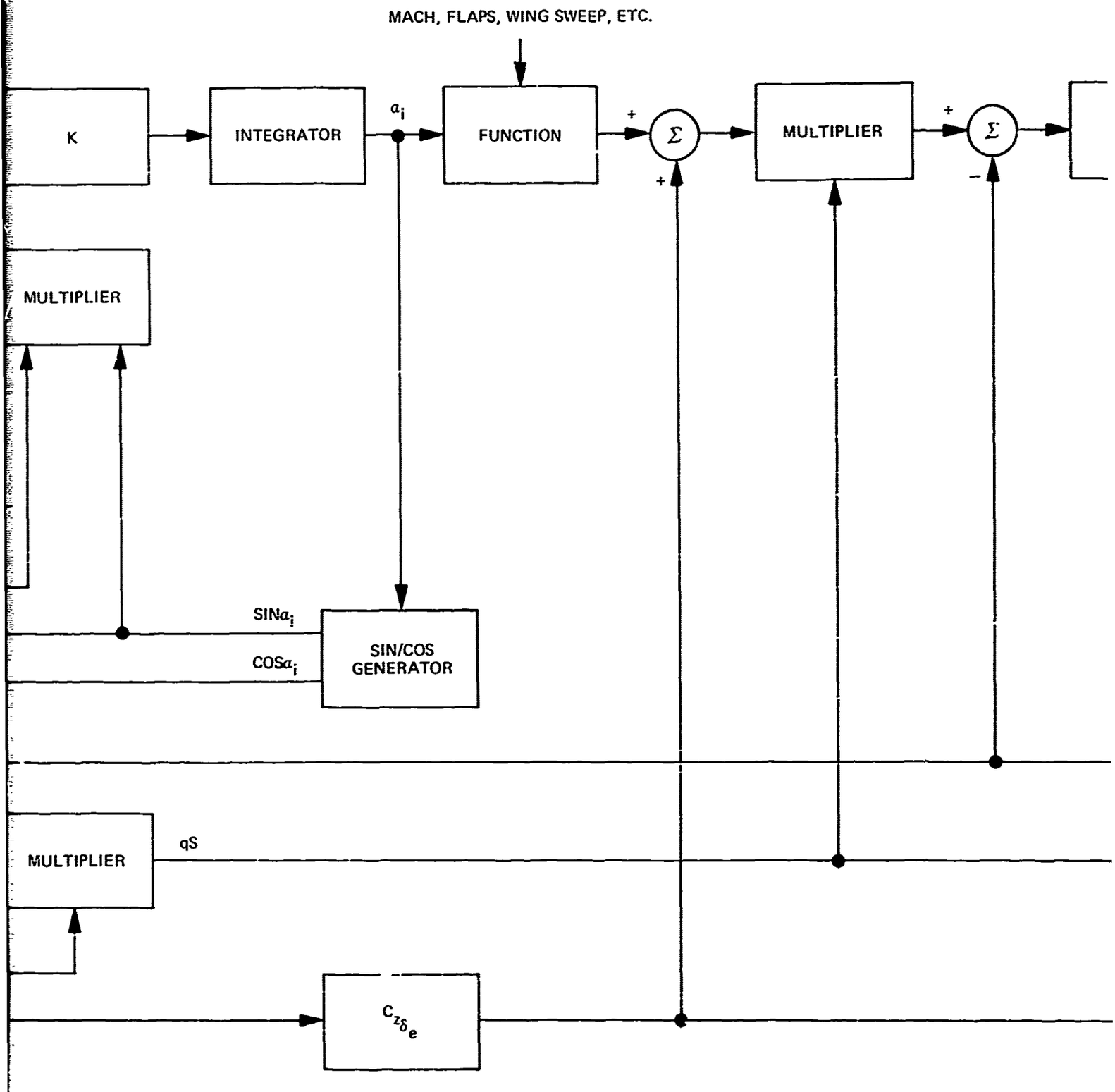
Computation of the aerodynamic coefficient, $C_z(\alpha, Ma)$, can be achieved by tables and interpolation in a digital computer. Alternatively, if a functional relationship can be empirically determined, expressing C_z as a function of powers of α and Mach, the function can be computed directly. Assuming this function can be expressed with sufficient accuracy and without undue complexity, this approach would be well suited for either DDA or whole-number digital computation.

As previously mentioned, care must be taken when dividing by quantities that can go to zero. In this system, only Az_{cg} can go to zero when the system is operating. This can best be handled logically by detecting small or zero Az_{cg} and holding the appropriate variables constant.

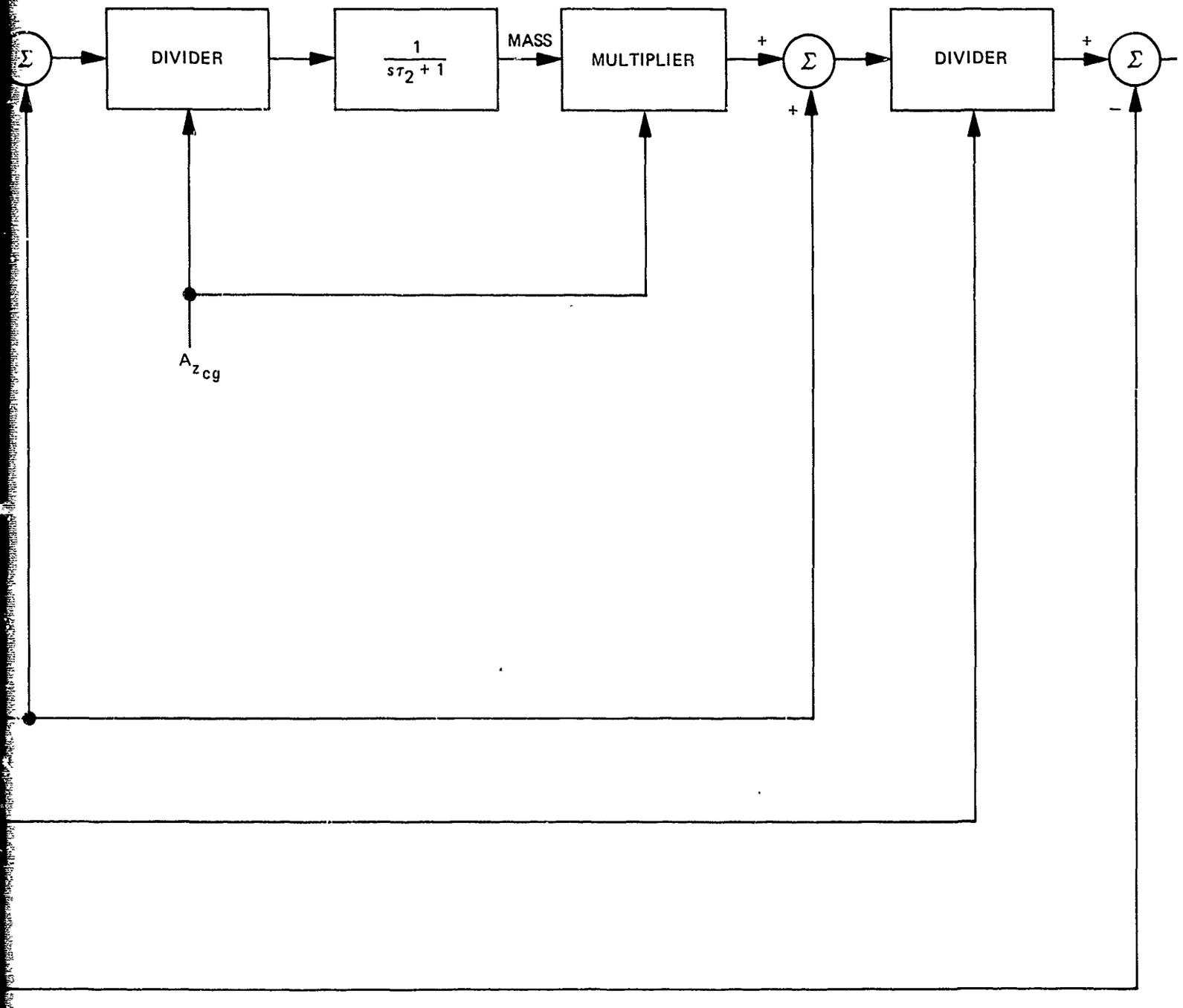
The thrust term must be included if the thrust line is not the same as the x body axis. It is assumed that thrust is a function of throttle position only. A simple polynomial of second or third degree should be sufficiently accurate since the thrust term is small, and the resulting error in computed α will be insignificant.


$\dot{h}_i$
DIVIDER
$\bar{V}_a$
$\frac{s}{s\tau + 1}$
$\bar{V}_a$
+
+
Σ
−
MULTIPLIER
K
MULTIPLIER
+
Σ
+
Σ
−
MULTIPL
+
MULTIPLIER
MULTIPLIER
$A_{z_{cg}}$
$A_{x_{cg}}$
δ_T
SHAPING
$TSIN\theta_t$
$\bar{V}_a$
SQUARE
MULTIPL
ρ
δ_e

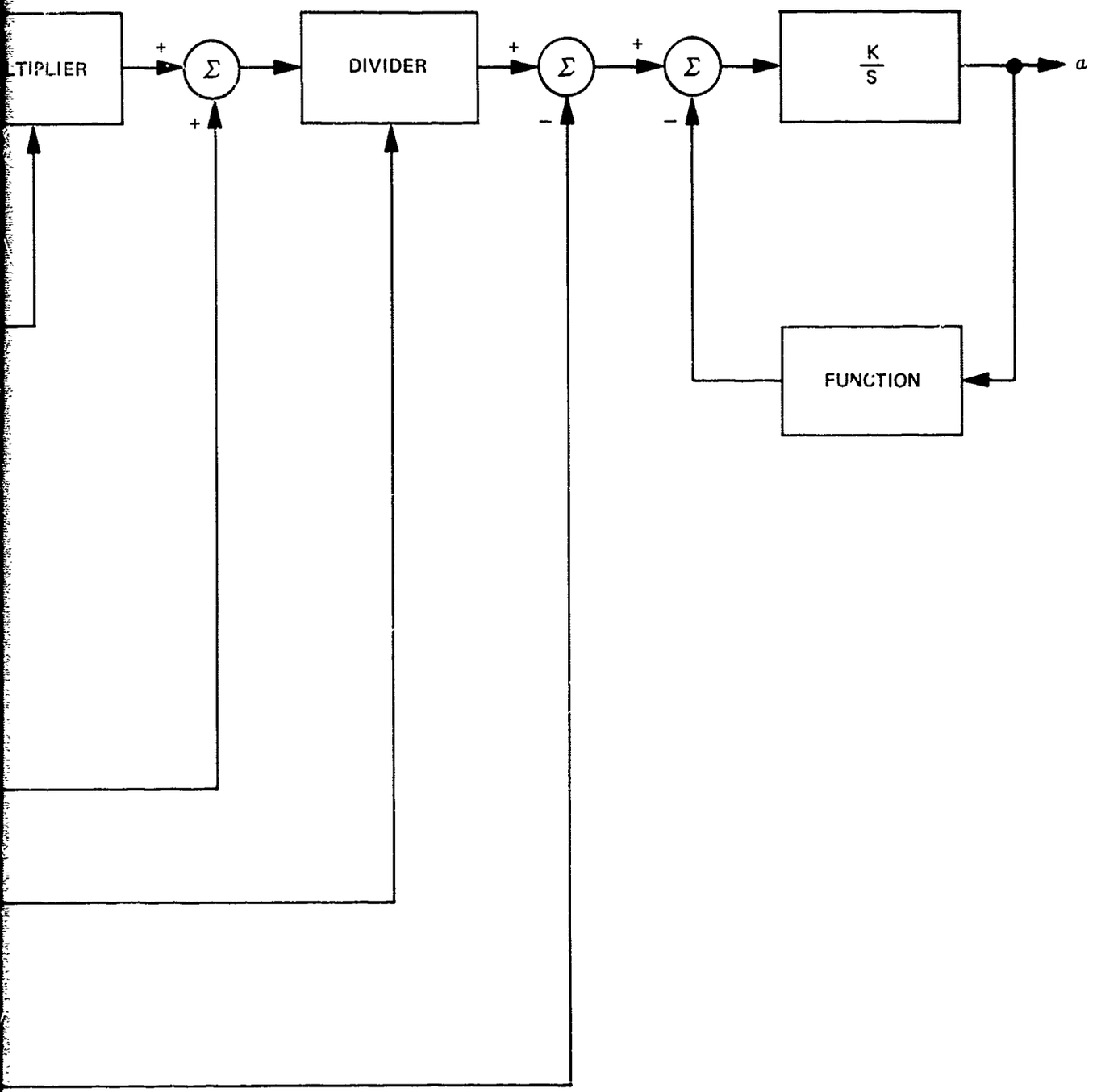
A


MACH, FLAPS, WING SWEEP, ETC.
K
INTEGRATOR
α_i
FUNCTION
+
Σ
+
MULTIPLIER
+
Σ
−
MULTIPLIER
SINα_i
COSα_i
SIN/COS GENERATOR
MULTIPLIER
qS
$C_{z_{\delta_e}}$

B



C



719-2-37

Figure 23
High-Accuracy,
Angle-of-Attack Computer

2. ANALOG COMPUTATION

Analog techniques provide the best means for producing an economical angle-of-attack system. This is due to the high cost of the analog-digital interface equipment as well as the greater complexity of digital systems. Analog circuitry has accuracy limitations that make it unsuitable for the high-quality system, but it is entirely adequate for applications such as transports and cargo airplanes.

The simplified versions of Methods I or II could readily be mechanized using analog circuitry, with certain advantages over digital hardware. The almost infinite resolution of the analog voltages permits the use of extremely high gains in implicit computation loops. In digital implicit loops, finite resolution of the data results in limit cycle oscillations at high-loop gains. Another advantage of analog systems is the ease of interface: analog sensors and displays are widely used at present, while their digital counterparts are just beginning to enter service. Analog systems will continue to hold an economic advantage at the interface until the majority of the interfacing units are digital.

The multiplier is the key component in the analog circuitry. The multipliers found most in practice are pulse-width modulation and quarter-square types, although the variable transconductance type is becoming more widely used. The quarter-square multipliers are standard in analog computer facilities because of their high-accuracy capabilities, but they are very expensive. Pulse-width multipliers are used extensively in flight control equipment for gain changing. They are relatively inexpensive, but accuracy is limited to approximately 1 percent. They also have the disadvantage of low bandwidth since the output must be filtered to remove the carrier frequency component.

Both hybrid and microcircuit multipliers of the variable transconductance type have been recently introduced on the market. These multipliers only achieve accuracies of about 1 percent but they are very economical, and it is reasonable to assume that their accuracy will improve as the market for multipliers expands. Several major semiconductor companies are developing low-cost multipliers of this type.

The accuracy of multipliers over a temperature range is a much more difficult problem than that of operational amplifiers. In an amplifier, stable resistors are used to define the gain; but in a multiplier, gain (scale factor) is at least partially determined by the stability of semiconductor components. In addition, there are offset errors that vary with temperature. Due to the

lack of detailed specifications on recently available models, it is difficult to determine actual performance over temperature without testing.

A circuit diagram of Method II simplified ($\beta_i = 0$, $\sin \alpha_i = \alpha_i$, $\cos \alpha_i = 1$) is shown in Figure 24. Equation (62) is multiplied through by $\bar{V}_a$ and Az_{cg} and solved implicitly, resulting in a slightly simpler mechanization than solving for a_i explicitly. Explicit computation requires two divide operations, which are really implicit computations using multipliers. Hence, the technique employed is simpler because it has only one implicit loop rather than two.

The problem of static instability due to polarity changes of Az_{cg} is handled by switching polarity in the loop. Two polarity detectors are used having a region of overlap around $Az_{cg} = 0$. Thus, in this region the input to the integrator is shorted to ground, and the output remains constant during the zero g maneuver.

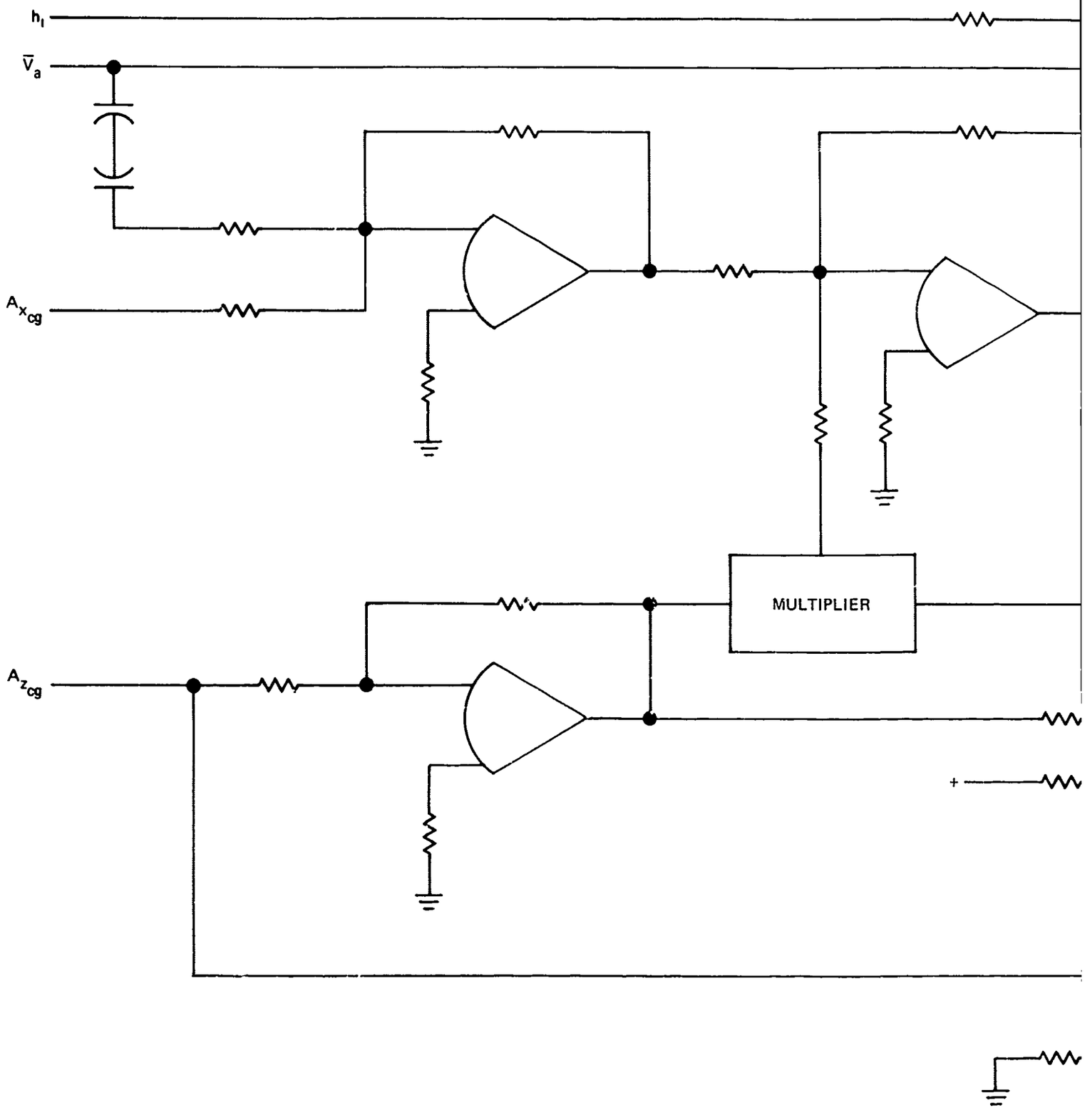
This mechanization requires eight encapsulated circuit modules, including two multiplier modules and a power supply module. The type of multiplier is not specified, but accuracy is specified to be 1 percent over the temperature range, without external calibration. It is assumed that such a multiplier will be available in the near future at a cost of $100 each in quantity.

The packaged angle-of-attack computer using the analog mechanization of Method II would weigh approximately 1.0 pound, occupy a volume of 32 cubic inches, and consume 3 watts of power. The sales price is estimated to be $600 to $800 each.

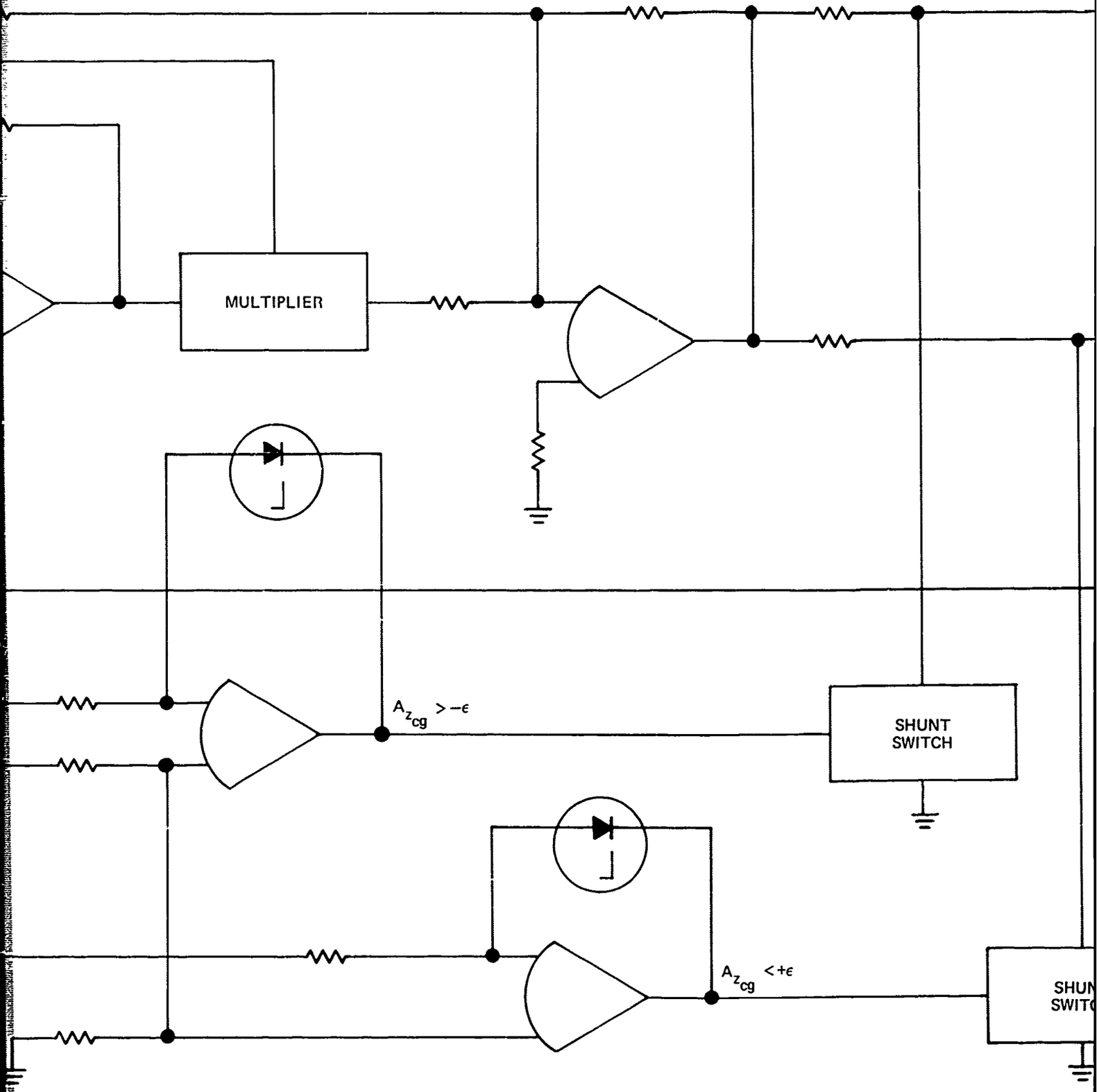
Method I is slightly more complex than Method II. It requires Scott-Tee transformers to convert synchro pitch and roll data to sines and cosines, and good demodulators to convert the sines and cosines to dc voltages. Three multipliers are required instead of two. Twelve modules are required; they weigh approximately 1.5 pounds, occupy a volume of 48 cubic inches, and consume 4 watts. The cost is estimated to be $800 to $1000 each.

3. DIGITAL DIFFERENTIAL ANALYZER

When mechanizing a problem that is small or moderate in complexity and requires accurate nonlinear or functional computations, the DDA is a prime contender. A DDA is a special purpose digital computer that performs incremental computation by simply updating the values previously computed rather than recomputing the entire function each cycle. Any operation that can be expressed in terms of differential equations can be readily mechanized.



A


MULTIPLIER
$A_{z_{cg}} > -\epsilon$
SHUNT SWITCH
$A_{z_{cg}} < +\epsilon$
SHUN SWITC

B

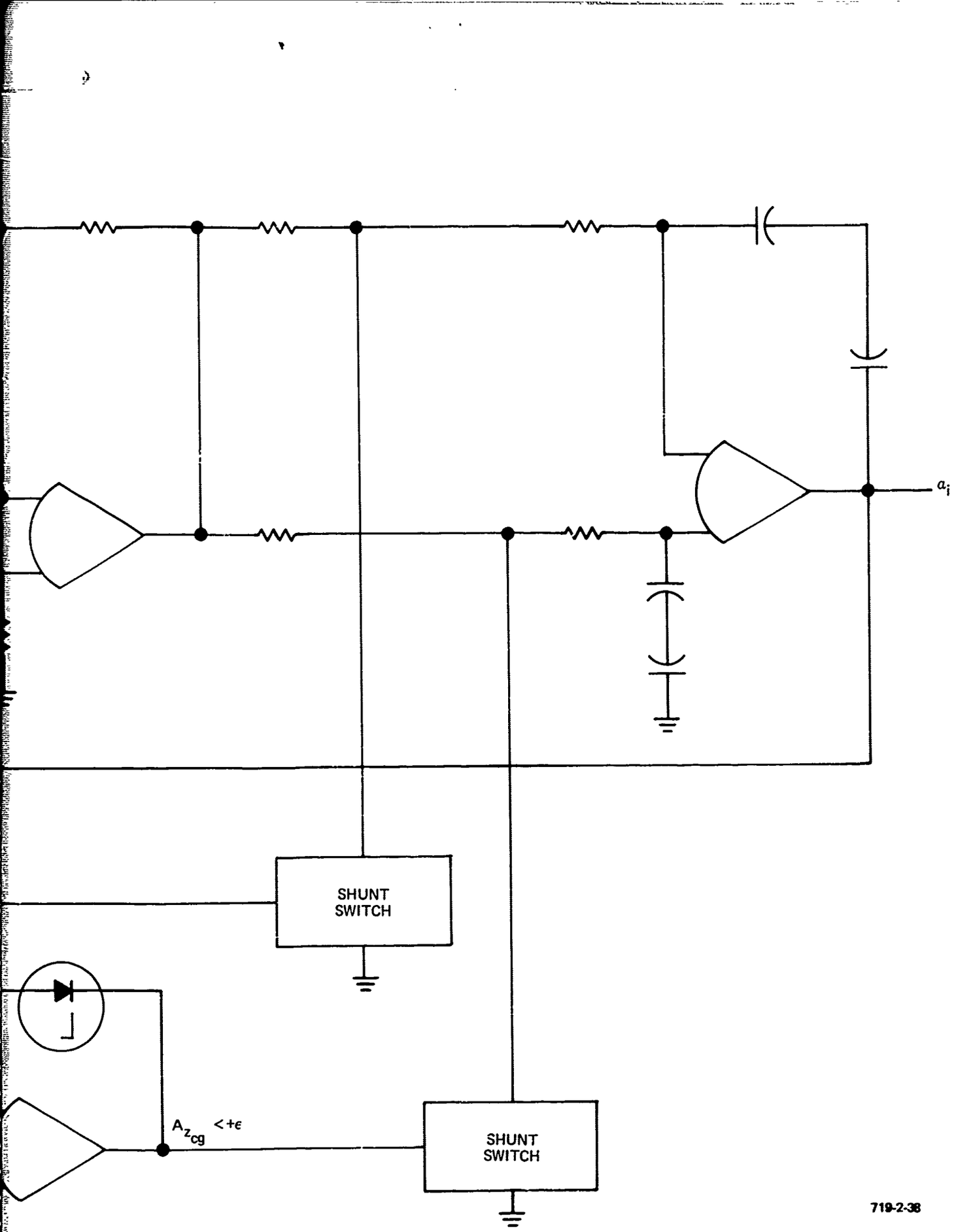


Figure 24
Circuit Diagram of
Method II Simplified

C

In many respects, the DDA is like an analog computer. The basic computing element is an integrator. A DDA system can be described by means of a "patch" diagram showing the interconnection of computing elements to solve a given set of differential equations. The information flow between integrators is in the form of pulses representing changes to be made to the problem variables.

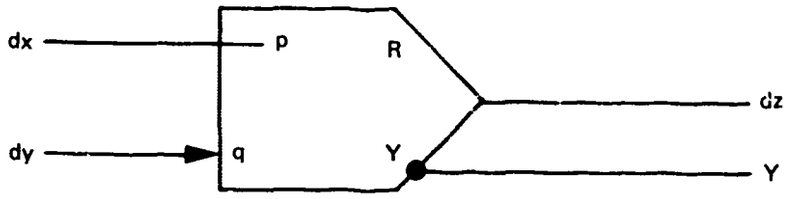
3.1 Basic DDA Operation

The symbol for the integrator is shown in Figure 25. The basic integrator consists of two registers and two adder/subtracters. The Y-register contains the current value of the dependent variable (Y), and the R-register holds the remainder resulting from the previous operation. The inputs and outputs of the integrators are incremental variables but in the figure are written as the differential variables which they approximate. When a dy increment appears at the input, Y is incremented by a constant scale factor q (decremented if the sign of dy is negative). When a dx increment appears, Y is added to R (subtracted if the sign of dx is negative). Whenever the value of R + Y exceeds p (the capacity of the R-register), an output dz pulse is generated and the R-register holds the remainder. When the size of the increments is small compared to the whole-word problem variables, the numerical solution of differential equations can be quite accurate.

A DDA servo is used to sum incremental signals and provide the output in incremental form. It is also used as a high-gain element in the same manner that an open-loop amplifier is used in an analog computer. In operation, the contents of a register in the DDA servo is incremented by each of several dy inputs. A zero detection circuit examines the contents of the register and provides an output whenever the contents is anything other than zero. Sometimes the output is gated and the polarity controlled by a dx input signal. The servo symbol is similar to the integrator symbol with the word "Servo" written in it.

3.2 Reversibility

Due to the finite levels of quantization, a DDA computing system will exhibit systematic error when the polarity of the independent variable is reversed. The error may build up indefinitely during computation and result in large accumulative errors. Fortunately, there are methods of handling this problem, making it possible to eliminate the problem completely in most cases and reducing it to very small levels in others. In general, a system of DDA integrators will be reversible without error if each DDA unit is reversible and if all computations in the DDA system due to a single input pulse are completed before a second input pulse arrives. These conditions are always sufficient but may not be necessary for a particular system.



$$\int dz = \frac{1}{p} \int Y dx$$

$$Y = Y_0 + qy = Y_0 + q \int dy$$

WHERE

p = OVERFLOW POINT

q = Y INCREMENT

Y_0 = INITIAL VALUE OF Y

719-2-39

Figure 25
Symbol and Equation for DDA Integrator

The best way to make a DDA unit reversible is to mechanize it to perform trapezoidal integration rather than rectangular integration. This has the added advantage that more accurate integration is performed. However, true trapezoidal integration is not always possible because it can only occur when there is no time delay associated with computation of the inputs to the integrator. In any closed-loop computation, there will be at least one integrator whose inputs were computed during the previous computation cycle and hence, have a time delay of one sampling period. Even so, a form of trapezoidal integration that greatly reduces any errors can be mechanized.

Controlled timing satisfies the requirements that all computations due to a single input pulse be completed before a second input pulse arrives. If the computer is fast enough to complete all computations within the time between two successive increments of any variable changing at its maximum rate, the requirement is met. This is sometimes very difficult to accomplish in practice.

Achieving reversible computation depends to a great extent on the type of function being computed. Of the functions most used in the alpha computer, the sin/cos function, polynomial evaluation, and multiplication functions are easily made reversible, while division is very difficult.

3.3 Incremental Computation

There are certain consequences of performing incremental computation that must be considered. First, in order to get the computation started, all the data registers must be filled with their proper initial conditions. This means that fixed constant numbers are stored in the machine so that when power is initially applied these numbers are automatically set into the registers.

Second, the sensor input data to the machine must be converted, not only from Analog-to-Digital (A/D) but also to incremental form. This is commonly done in conjunction with a standard A/D converter. The increment generator (Figure 26) consists of storage space for each input variable in a data memory, an adder/subtracter, and a comparator. An analog input to the A/D converter is compared to the accumulated value of the increments of that quantity held in storage. If they are not equal an increment is generated, positive if the input is larger and negative if the input is smaller than the stored value. The increment is then added to the stored value, which is put back in storage to await the next conversion cycle.

If an error is made in a DDA due to noise, power supply transient, etc, the DDA will be in error for all subsequent time due to its incremental nature. Subsequent computations are merely changes to what has gone on before, and when

an error goes in, it remains in. There are two ways to attack this problem. One is to detect the existence of certain check points of an input and set the computer directly to the corresponding values. For example, if sine and cosine of an input angle are being computed, the zero degree point might be designated the check point. Whenever the angle passes through zero, the sine can be set to zero and the cosine to one immediately. The incremental computation then carries on from that point. If any errors have been made, they are corrected; if not, no change is made in the data.

Another method of correcting any transient errors that may have occurred is to set the computer registers, including the increment generator storage registers, to a specific set of points with the outputs and all intermediate variables correctly corresponding to the values set in the increment generator registers. Then the computer slews up to the set of values corresponding to the actual inputs at the maximum computer slew rate. This operation can be repeated periodically to ensure that no error remains in the computer longer than a fixed amount of time.

These two techniques require additional hardware over that needed to perform the actual computations. However, they have both been successfully used in DDA practice.

3.4 Rate Limiting

The frequency response of a DDA is limited in that the problem variable can change by only one increment per computing cycle. If the input variables are changing at a very high rate or at such a rate that the computed output must vary rapidly, the DDA may contribute a considerable lag. The two factors that affect the maximum rate the computer can handle are the computer cycle time and the size of the increments.

Since a problem variable may change by one increment per cycle, the larger the increment, the greater the rate of change of the variable for a given cycle time. However, large increments cause poor computational accuracy and lead to limit cycle oscillations in closed-loop systems.

The computer cycle time depends on the computer organization and on the clock rate at which the circuitry operates. Assuming the system is operating at the maximum clock frequency that the circuitry can tolerate, the frequency response will depend only on the organization.

Organization refers to the degree of parallelism in the computer. Arithmetic processing can be either serial or parallel, and processing of the integrators either sequential or simultaneous. Economy demands a minimum of

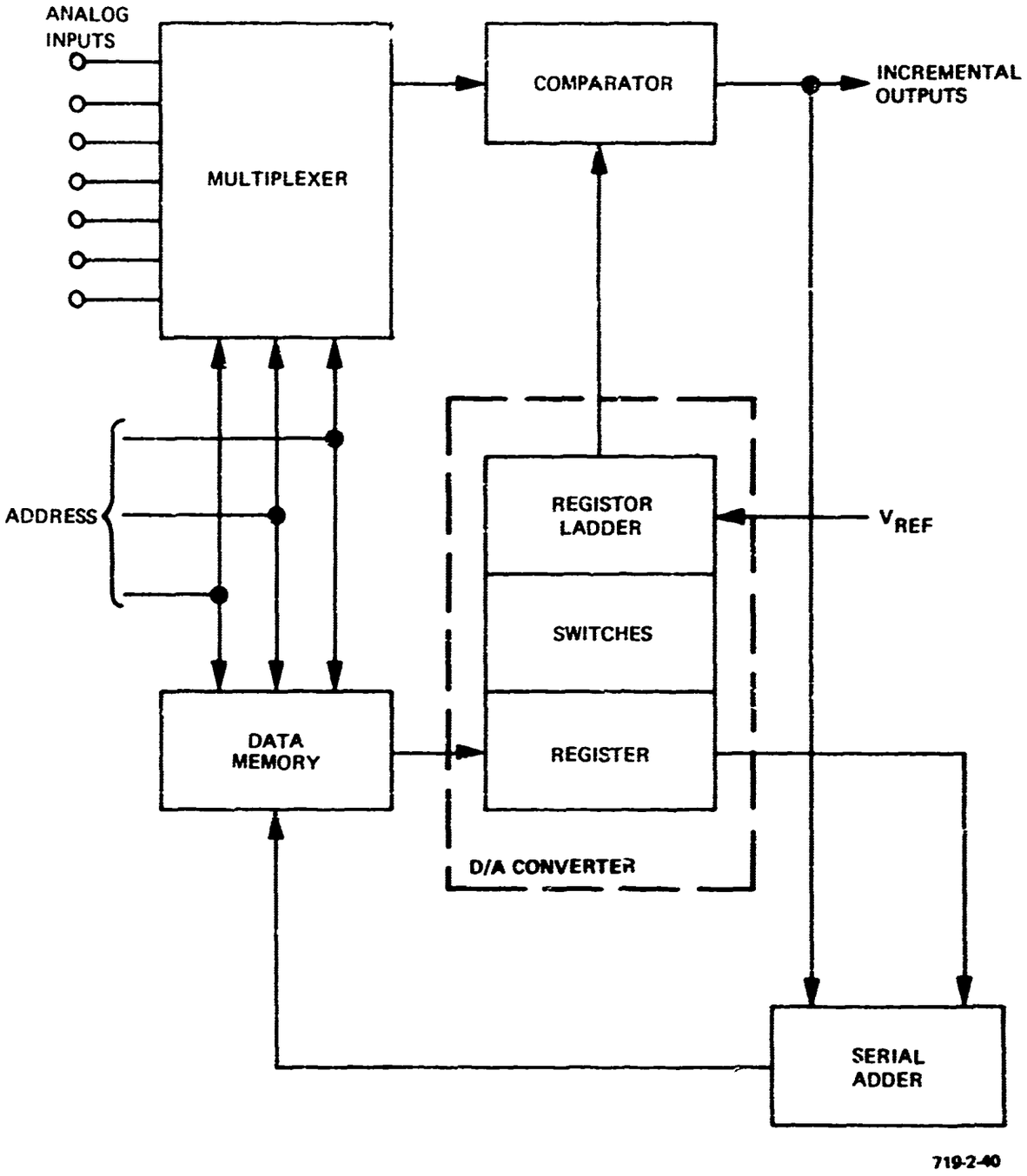


Figure 26
Analog-to-Digital Converter
and Increment Generator

lism, but this is achieved at the expense of speed. Most DDA's use serial arithmetic, but processing of the integrators varies from all-sequential to all-parallel. If a configuration of 32 integrators is not fast enough for a particular application, the DDA can be organized into two groups of 16 integrators operating in parallel to double the speed. The cost would actually be quite small since the memory requirements are the same for both systems. Only an additional arithmetic unit, consisting of several adders and logical circuits, would be required.

3.5 DDA Mechanization

The DDA mechanization of the system in Figure 23 is developed for estimating cost, size, and performance. The design is based on the use of Metal-On-Silicon (MOS) shift registers for the data registers, and read-only memory for program control and constant storage. Hence, the arithmetic is serial. It is assumed that the integrators will be processed sequentially.

A clock rate of 5 MHz will be used. A word length of 20 bits is dictated by resolution requirements (10 data bits, 1 sign bit, and 9 bits for scale factor resolution). The resulting time required to perform one integration is 4 microseconds.

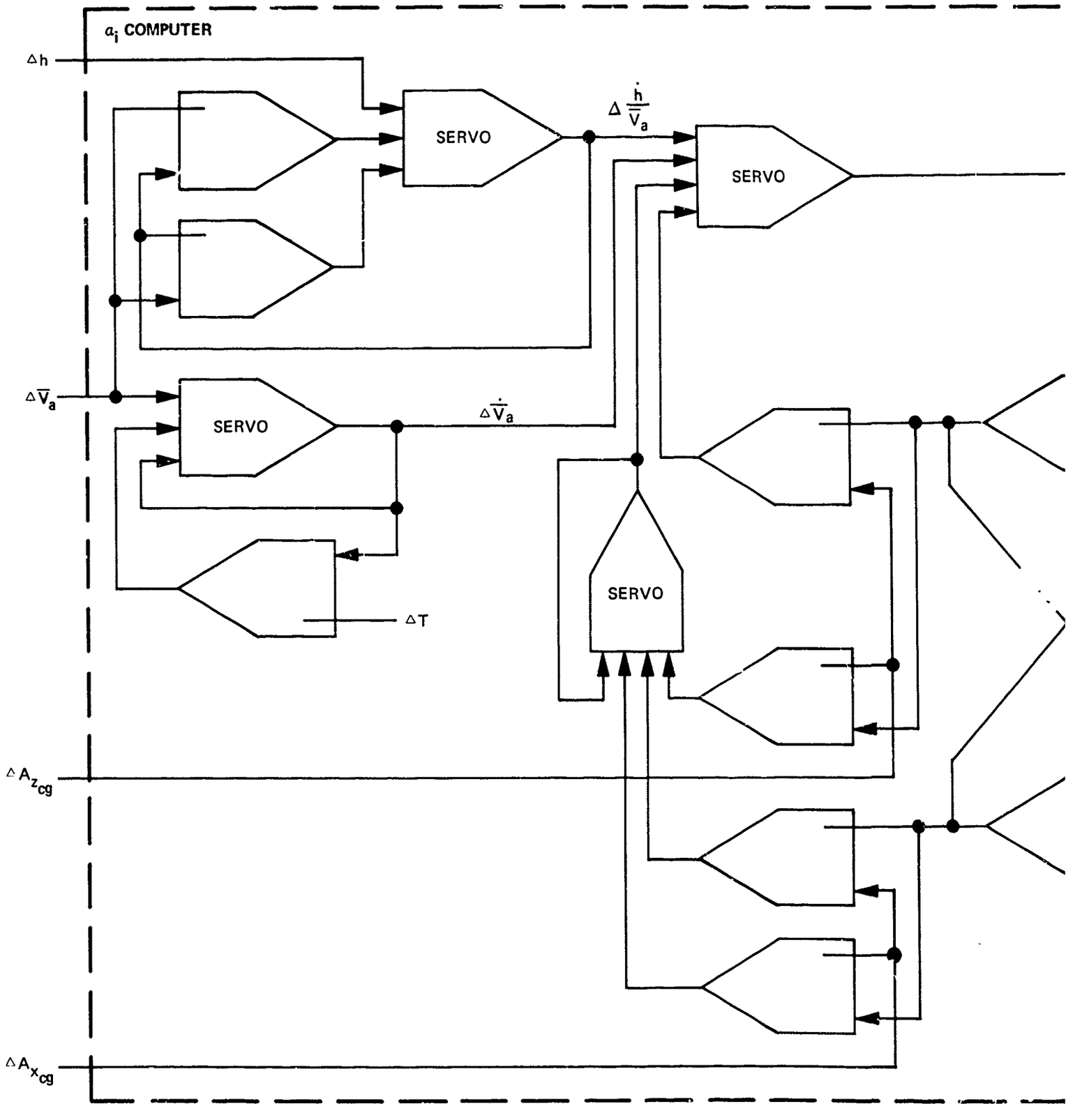
Figure 27 is a DDA patch diagram of the computer. As indicated by the figure, 30 integrators and 13 DDA servos are needed. A machine built with only the minimum number of units would be totally inflexible and without growth capability. Hence, a minimum of 32 integrators and 16 servos is used for the estimates in this study.

Given the clock rate, number of integrators, and word length, the slewing rate capability of the machine can be determined. For an all-sequential machine, the computer cycle time is 4 microseconds per integrator multiplied by 48 integrators, or 192 microseconds. Assuming an increment of 0.025 degree, the output slewing rate is then 130 degrees per second. Thus, it appears that a sequential process DDA will handle the angle-of-attack problem.

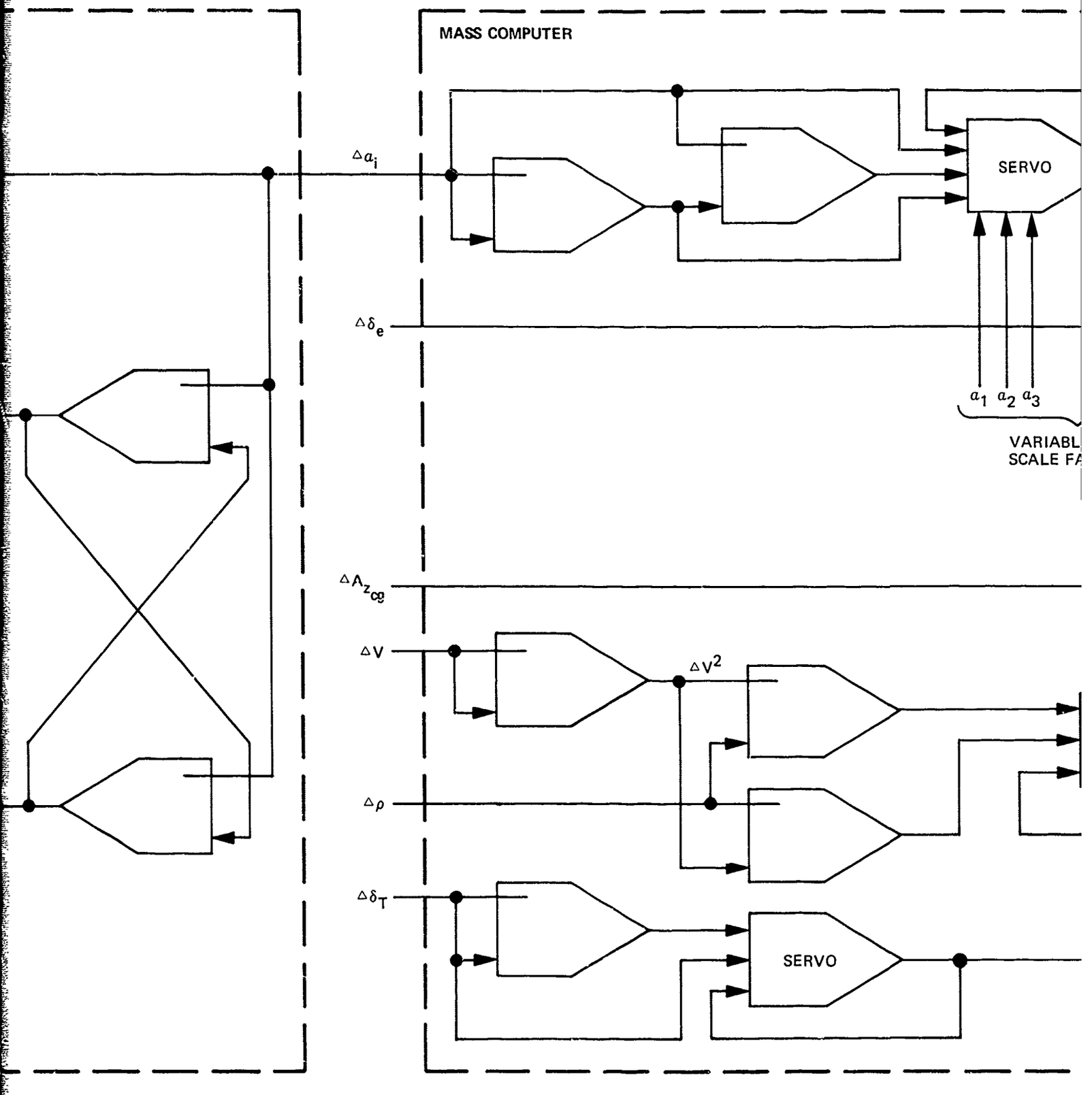
3.6 System Organization

Figure 28 shows the organization of the DDA. It consists mainly of interface equipment, the arithmetic unit, timing and control circuits, and various memories.

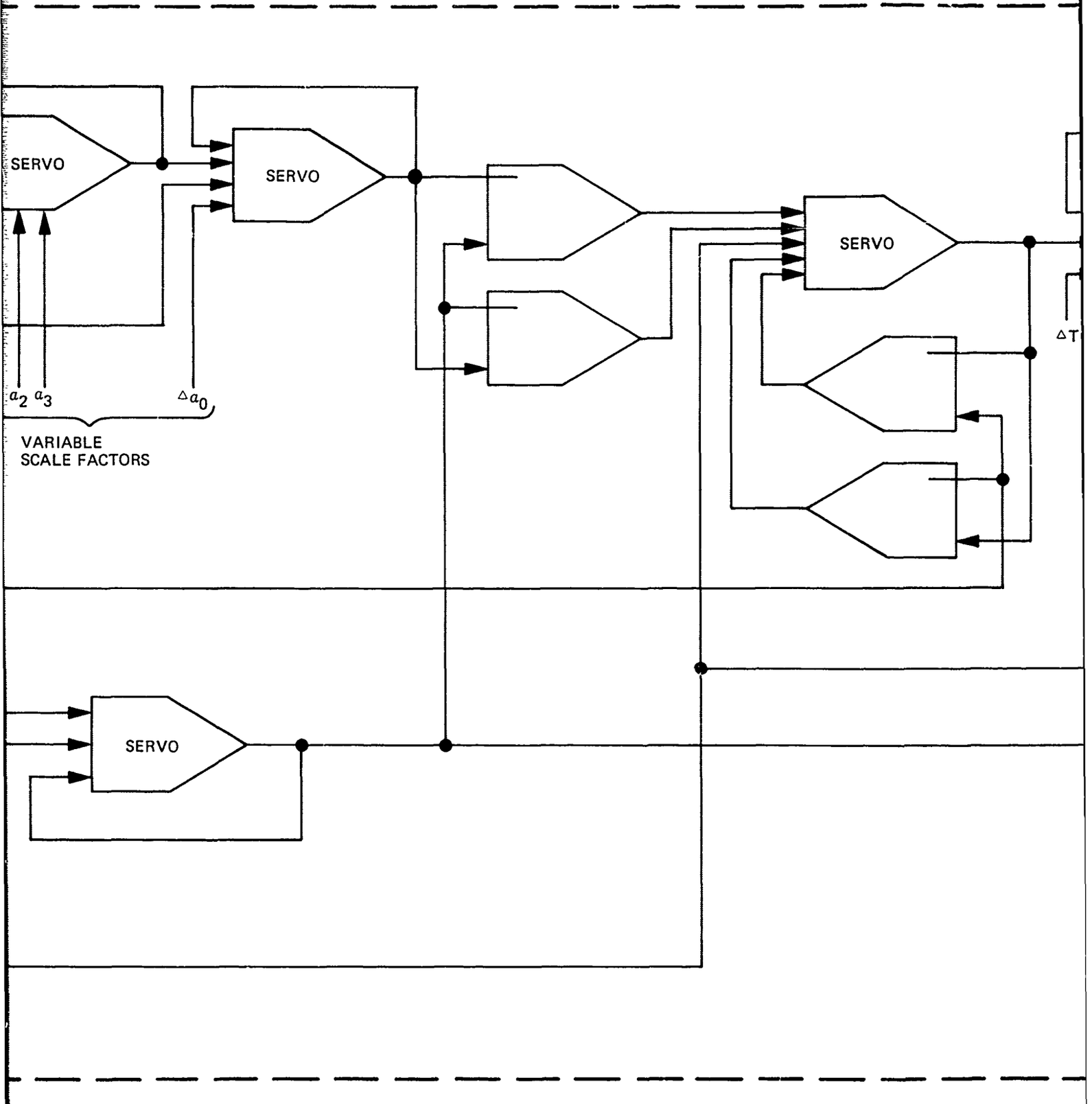
The interface equipment contains a multiplexer capable of handling a minimum of eight inputs, an A/D converter, an increment generator, and a D/A converter. The A/D converter can be time shared for D/A conversions if used with sample-and-hold output amplifiers.


a_i COMPUTER
Δh
SERVO
$\Delta \frac{\dot{h}}{\overline{V}_a}$
SERVO
$\Delta \overline{V}_a$
SERVO
$\Delta \dot{\overline{V}}_a$
SERVO
ΔT
$\Delta A_{z_{cg}}$
$\Delta A_{x_{cg}}$

A


MASS COMPUTER
Δa_i
SERVO
$\Delta \delta_e$
a_1 a_2 a_3
VARIABL
SCALE FA
$\Delta A_{z_{cg}}$
ΔV
ΔV^2
$\Delta \rho$
$\Delta \delta_T$
SERVO

B


SERVO
SERVO
SERVO
SERVO
a_2 a_3
Δa_0
VARIABLE
SCALE FACTORS
ΔT

C

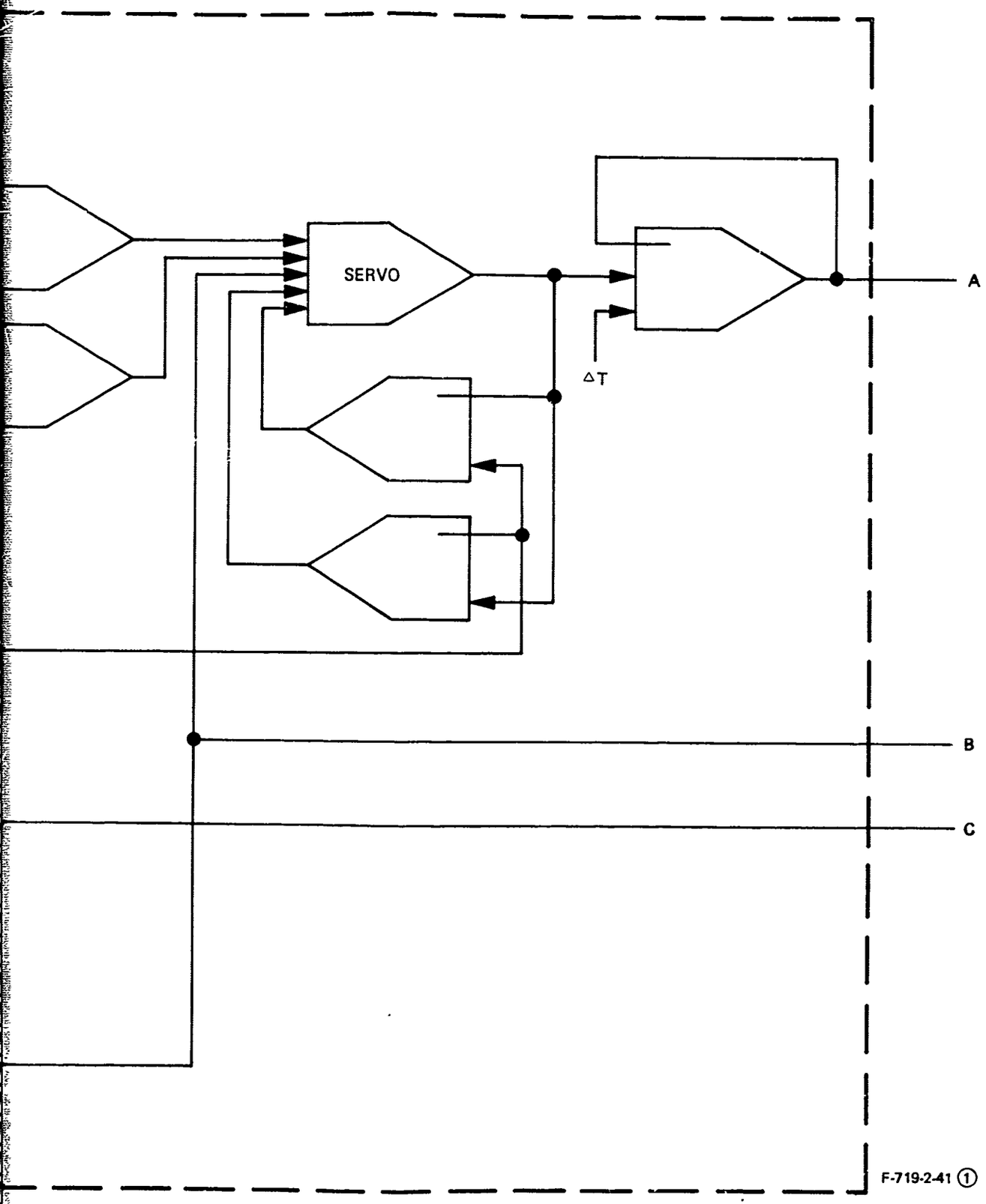
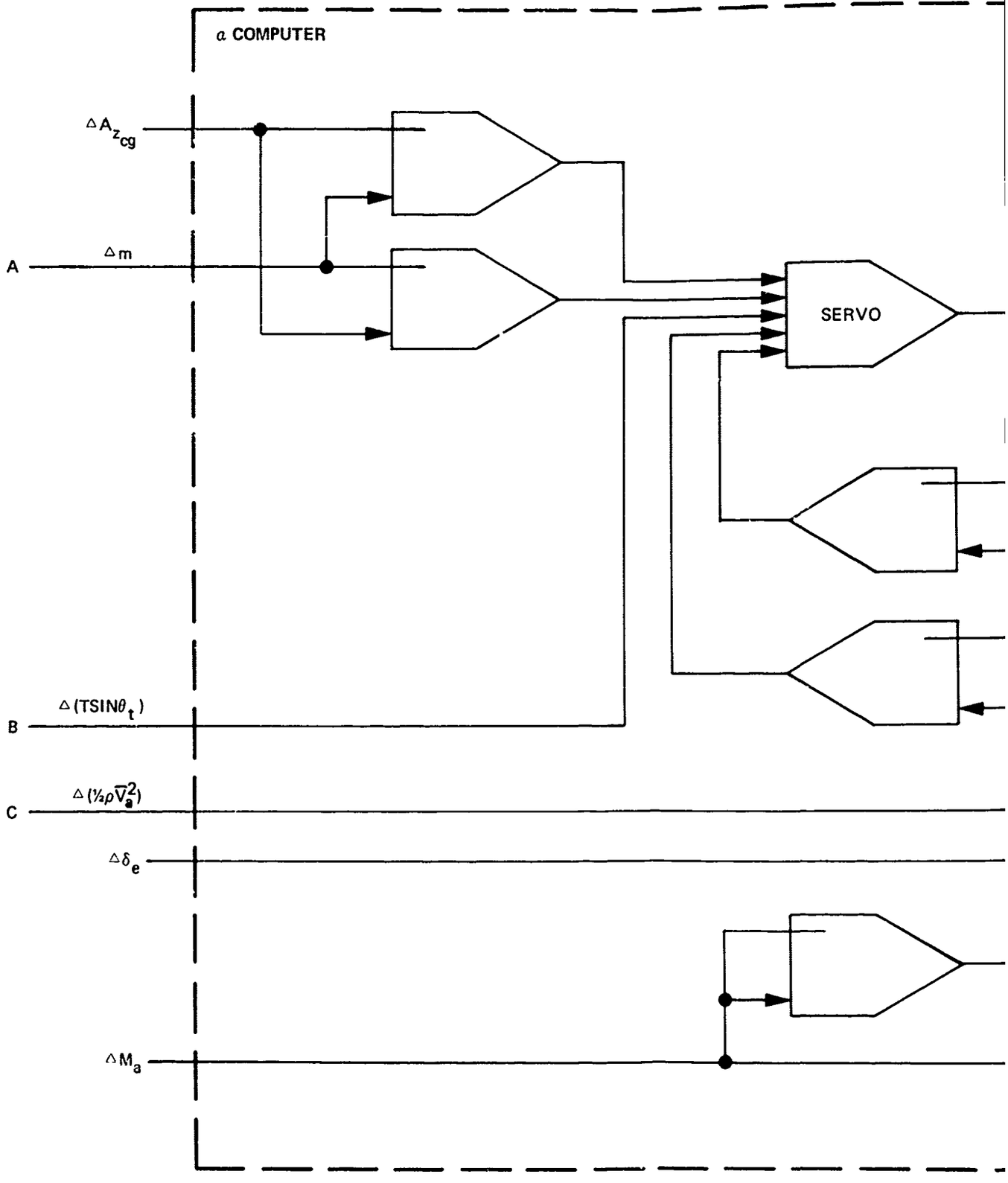
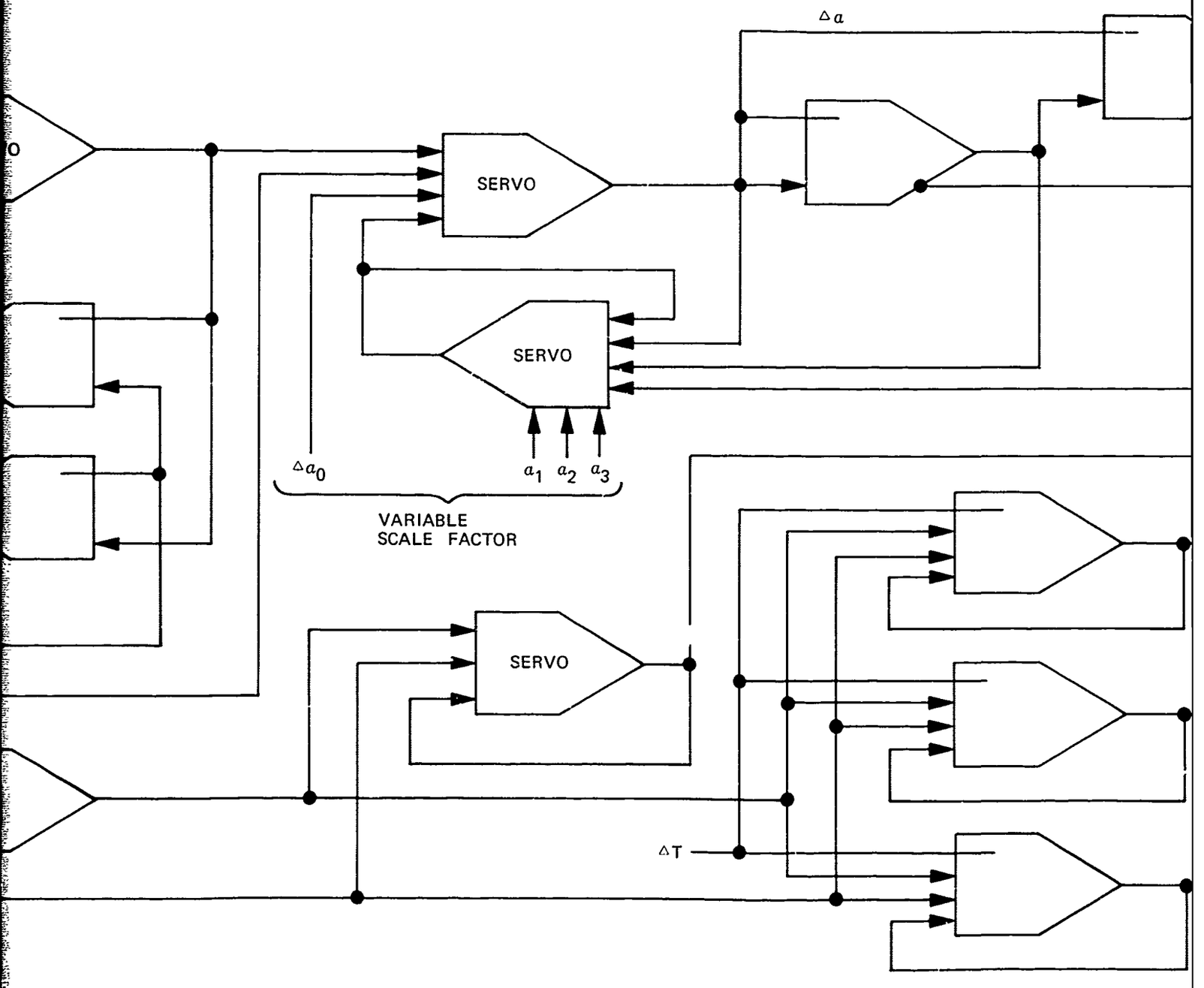


Figure 27
DDA Patch Diagram

D


a COMPUTER
$\Delta A_{z_{cg}}$
A
Δm
SERVO
B
$\Delta (TSIN\theta_t)$
C
$\Delta (\frac{1}{2}\rho \bar{V}_a^2)$
$\Delta \delta_e$
ΔM_a

A


Δa
SERVO
SERVO
a_1 a_2 a_3
Δa_0
VARIABLE
SCALE FACTOR
SERVO
ΔT

B

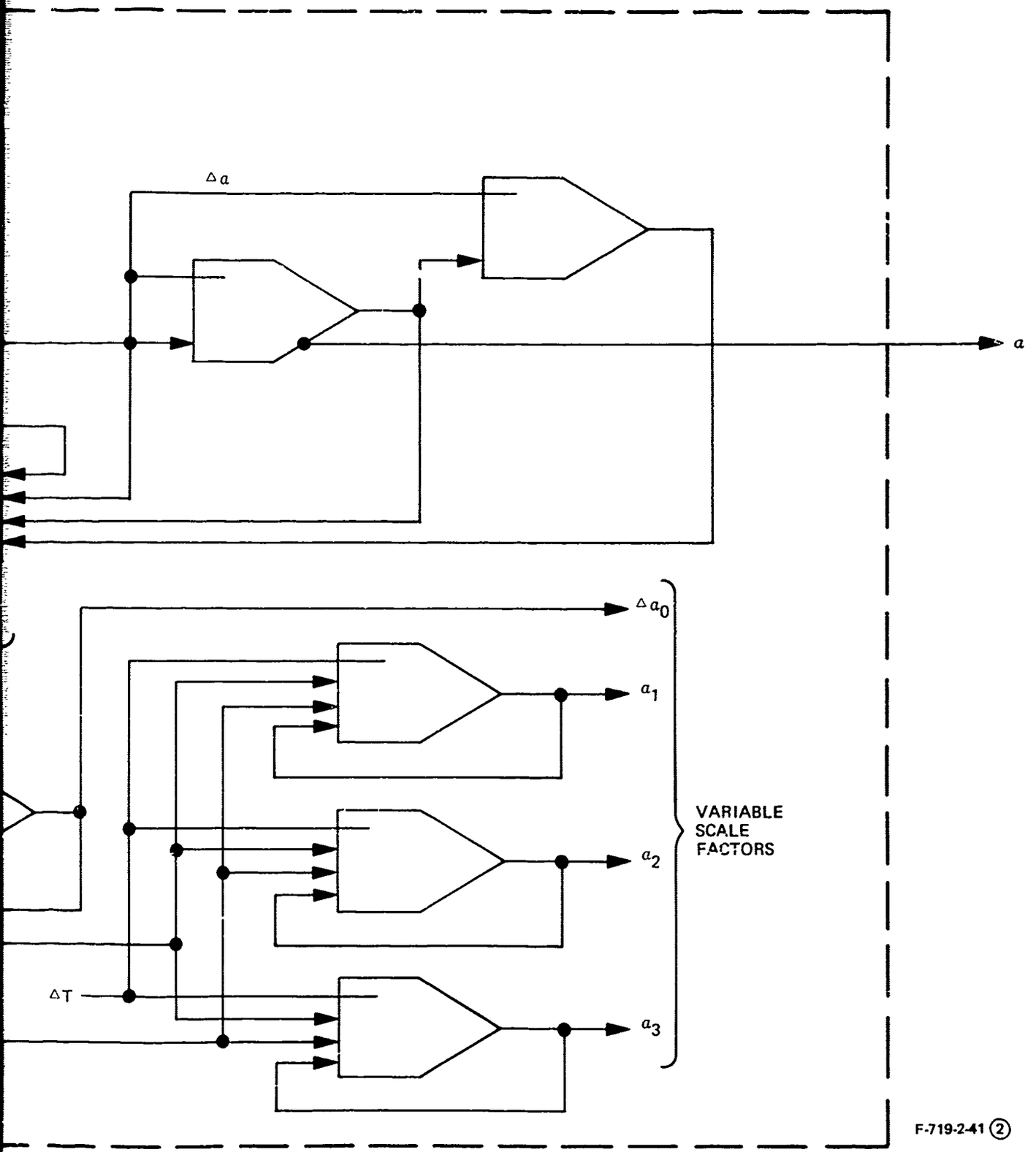


Figure 27 - Concluded

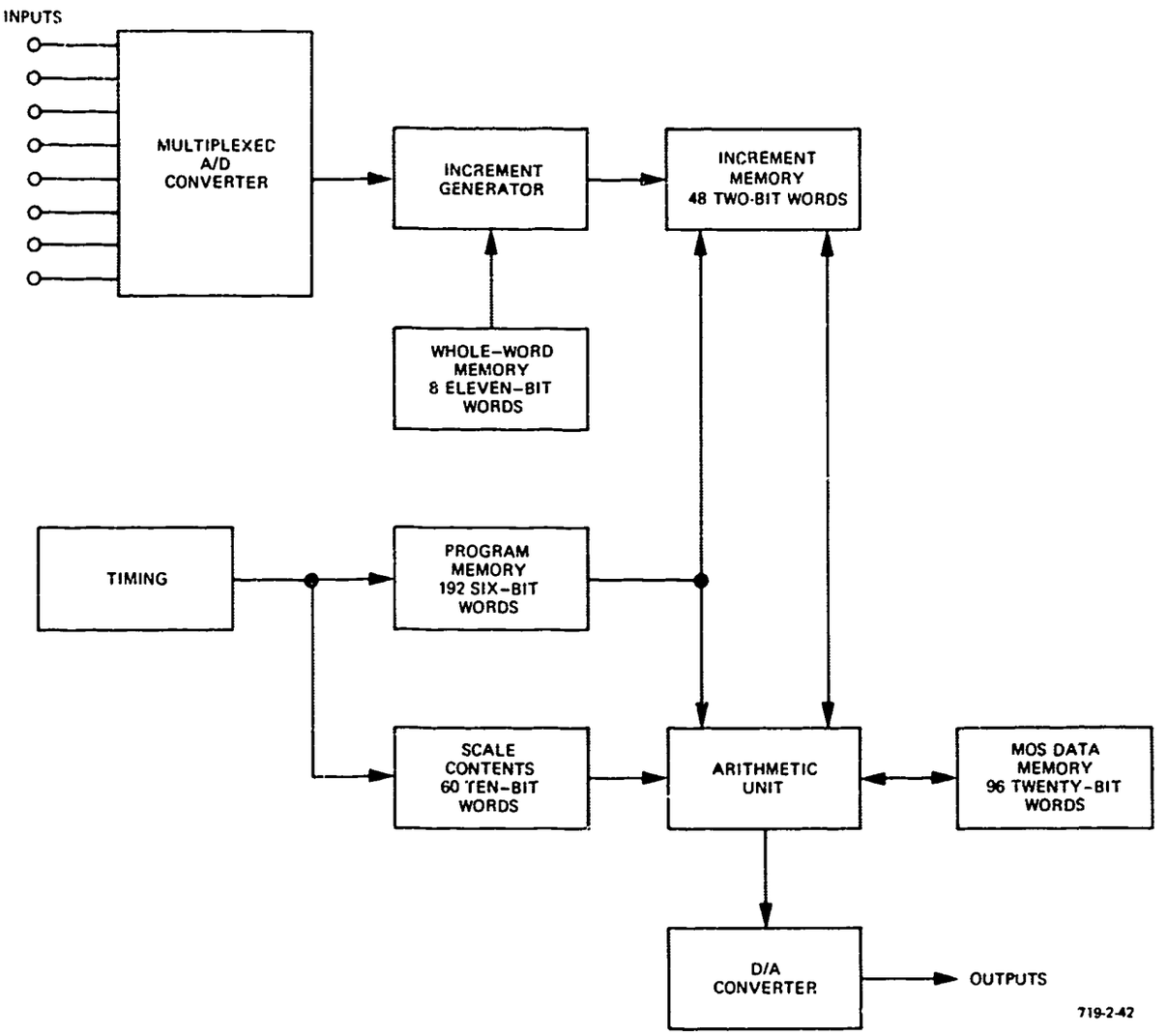


Figure 28
DDA Block Diagram

The arithmetic unit consists of four serial adder/subtracters, overflow detection logic, and zero detection logic. This circuitry is time shared by all integrators and servos.

The timing circuitry consists of two counters (a bit counter and a word counter) and two decoders. The bit counter and decoder define bit times within a word, while the word counter and decoder define word times within a computer cycle.

There are five distinct memories in the system. The MOS data memory is a pair of long shift registers containing all the Y and R registers of the integrators and servos. The increment generator, whole-word memory is a long shift register storing the whole-word digital inputs from the A/D converter. The increment memory stores the outputs from the increment generator and each DDA integrator and servo. All of the memories discussed so far are shift register types. The program memory and scale constant memory are solid-state, read-only types. The program memory provides addresses to the increment memory at the proper times to select outputs from integrators as inputs to other integrators required to execute the program. Scale constants are also gated by timing signals into the arithmetic unit as needed.

The physical characteristics of the DDA are estimated as follows:

- Size 175 cubic inches
- Weight 6 pounds
- Power 25 watts

The cost estimate for the DDA is based on the use of large-scale MOS memories, and Transistor-Transistor Logic (TTL) including Medium-Scale Integrated (MSI) circuits. The parts are full military temperature range types, and the prices assume quantity purchase. On this basis, the sales price of the DDA system in production would be approximately $4000.

4. WHOLE-NUMBER COMPUTER

A small, special purpose, whole-number digital computer would be ideal for performing the angle-of-attack computations. The only question concerns the economics of this approach when compared with the DDA, which is normally considered the simplest digital approach to a small problem. A detailed analysis of the computer requirements has been made, permitting realistic estimates of the hardware requirements. The following paragraphs describe a hypothetical computer that has the capability required for computing angle of attack and some reserve capacity for self-test routines and growth. This model is used for estimating purposes comparing it to the DDA design of Paragraph 3.

4.1 Advantages of Whole-Number Computation

Whole-number computers have several important advantages over incremental computers. A whole-number computer has greater flexibility to perform a wide variety of computations, logical operations, and self-test routines. The basic DDA solves only differential equations, and must have considerable extra hardware to provide any different type of operation.

The analog-digital interface is simpler for the whole-number computer because no increment generator is required. In addition, since each output is recomputed each cycle, any error caused by noise or power supply transient will be corrected the next cycle.

The whole-number computer will normally require much less data memory than the DDA. In the latter unit, all problem variables are stored in data memory so they can be updated each computer cycle. In a whole-number computer, most computed variables are required only temporarily; that is, until they are used in some other computation. Then they are discarded, and the memory locations in which they were stored are available for new variables.

There is no reversibility problem with a whole-number computer, nor is there rate limiting, except where a rate limit is deliberately programmed. The frequency response is limited only by the computation cycle rate and sampling theory considerations.

4.2 Computational Requirements

Referring to Figure 23, the majority of computations are multiplications and additions. There are also a sizable number of scaling operations required; that is, multiplication by constants. Scaling can often be accomplished by shift operations, which save a great deal of time. A few divisions are required, and if the hardware is included to provide for multiplication, the mechanization of the divide instruction costs very little more.

The z-force coefficient can be computed by polynomial approximation as in the DDA, or alternatively, by using storage tables and interpolating to determine the value between stored points. Interpolation is a common technique for function evaluation, with several approaches that can be used. The interpolation routine consists of passing a polynomial through several points surrounding the input point. The polynomial may be of any degree. The first-degree (linear) interpolation is easiest to compute but requires the largest number of stored data points. Higher-degree interpolations can achieve the required accuracy with fewer stored points, but the computation is more complex and requires more stored program instructions.

The limiting case is where there are no stored data points at all, only coefficients of a high-degree polynomial describing the entire curve. This is the polynomial approximation approach adopted in the DDA system. The polynomial approximation is better than tables/interpolation in the whole-number computer as well as the DDA because it is very easy to program the polynominal evaluation for a reasonably smooth curve such as $C_z(\alpha)$. If the function had a large number of wiggles and sharp bends, then a tables/interpolation approach would be more efficient.

4.3 Computer Organization (Figure 29)

A simple, straightforward computer is envisioned to take maximum advantage of newly emerging solid-state memory technology. The computer is a serial, binary, 2's complement machine operating at a basic clock rate of 2 MHz. It can perform an addition in 16 microseconds and a multiplication in 144 microseconds, including access to memory. There are 10 instructions, some of which can be modified by use of a hardware index register. The characterisitics of the computer are summarized in Table II.

It is estimated that the computer can perform the angle-of-attack program at a rate exceeding 50 times per second, with a considerable margin for growth. This is based on the use of serial arithmetic, 16-bit words, and a 2-MHz clock rate. The total number of instructions to be executed is estimated at 150, broken down as follows:

Additions*	120 at 16 microseconds	=	1,920
Multiplications	85 at 144 microseconds	=	12,240
Divisions	5 at 288 microseconds	=	1,440
*Add, Subtract, Load, and Store			15,600 microseconds

15,600 microseconds per cycle is equivalent to about 64 iterations per second. The computer should not be made faster than necessary since this will add to its cost.

TABLE II

WHOLE-NUMBER COMPUTER CHARACTERISTICS

Data Type	Serial
	Binary
	2's Complement
	Fixed Point
Number of Instructions	Ten
Computing Time	Add: 16 Microseconds
	Multiply: 144 Microseconds
Memory	256 Words, 10-Bit, Read-Only
	31 Words, 16-Bit, Scratchpad
Input/Output	Direct Access to Scratchpad
Hardware Type	MOS Memories
	TTL Logic
Weight	4.5 Pounds
Size	12 Cubic Inches
Power	20 Watts

4.3.1 Instruction Format

An instruction word consists of 10 bits, subdivided as shown below:

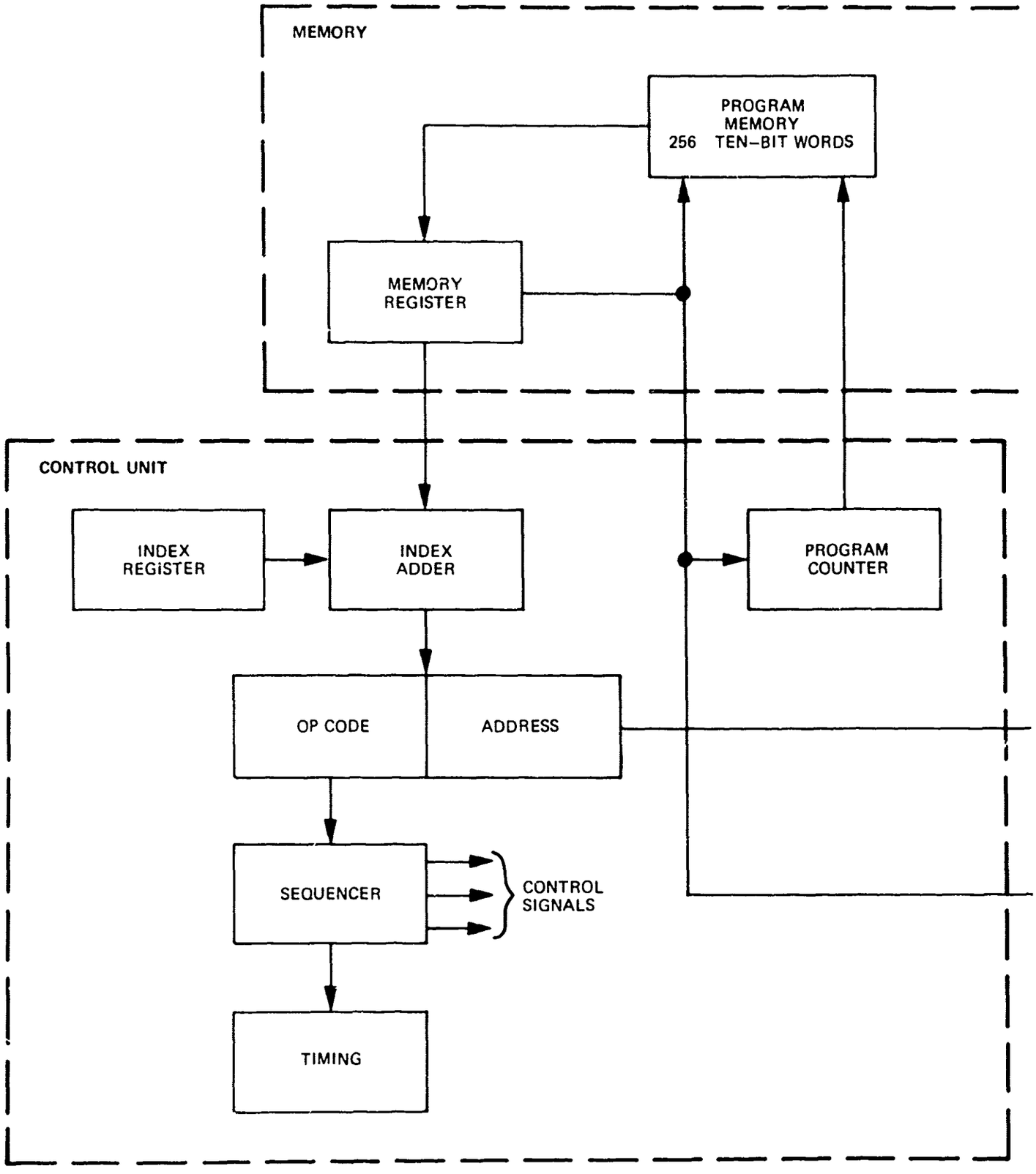
10	9	8	7	6	5	4	3	2	1
Index	Order				Address				

Bits 1 through 5 specify the location in data memory to which the instruction refers; bits 6 through 9 identify the order to be performed. A "one" in bit position 10 causes the contents of the index register to be added to the address before executing the instruction. If bit 10 is "zero", the address is not modified. A list of instructions is presented in Table III.

TABLE III

INSTRUCTION LIST

Bit No.	Instruction
1	Load
2	Store
3	Add
4	Subtract
5	Multiply
6	Divide
7	Jump
8	Jump on Negative
9	Shift Right (sign extend, end off)
10	Shift Left (circularly)


MEMORY
PROGRAM MEMORY 256 TEN–BIT WORDS
MEMORY REGISTER
CONTROL UNIT
INDEX REGISTER
INDEX ADDER
PROGRAM COUNTER
OP CODE
ADDRESS
SEQUENCER
CONTROL SIGNALS
TIMING

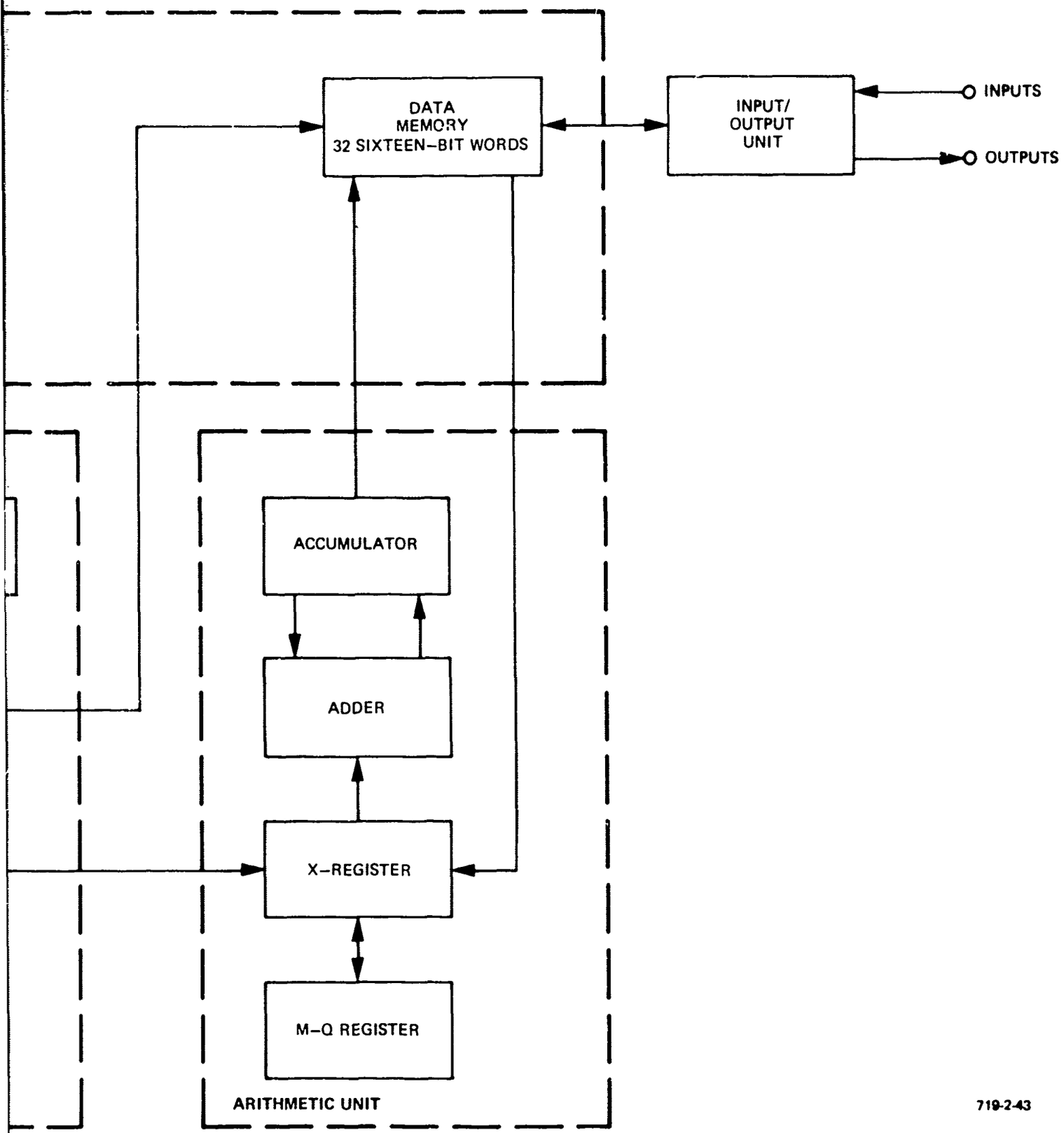


Figure 29
Whole-Number Computer
Block Diagram

B

4.3.2 Control Unit

The control unit obtains the instruction to be executed from storage and then controls the various transfer gates necessary to execute this instruction. The unit consists of an index register, an index adder, a program counter, an instruction register, timing counters, and a sequencer. The sequencer is the most complex part of the control unit; it decodes the instruction and generates the proper sequence of timing signals to execute the instruction. An index register, which requires very little hardware, is included to facilitate the setting up of loops. This is highly useful in programming polynomials and in self-test routines.

The data memory is small and can be addressed with only 5 bits. The instruction word is 10 bits in length: 5 for operation (op) code and index, and 5 for addressing. There are 256 program memory words, which takes a minimum of 8 bits for addressing. One of the 32 data memory addresses is reserved for use when addressing the program memory. If this address appears with an op code, the computer takes the next word in the instruction sequence as the operand address. Thus, a 10-bit address is available for addressing the program memory.

A real-time clock provides a signal that interrupts the program and makes it return to the beginning of the computation cycle. This internally generated signal occurs once every 20 milliseconds. The interrupt operation not only serves to time the required problem solution rate, but also provides a automatic recovery from improper program loops which may arise due to a temporary internal failure.

4.3.3 Arithmetic Unit

The arithmetic unit performs the various operations required in the solution of a given problem. Two arithmetic shift registers and a serial adder/subtracter are the essential components. Addition is accomplished by adding a number from the data memory to the number in the accumulator. This sum is held in the accumulator.

When a multiplication is performed, the number in memory is the multiplicand, and the number in the accumulator is transferred to the M-Q register and used as the multiplier. The multiplicand is successively added to the accumulator contents and shifted in accordance with the "one" bit in the multiplier. Upon completion of the operation, the most significant half of the product is held in the accumulator.

When a divide operation is performed, the contents of the accumulator is the dividend and the specified word from memory is the divisor. Division is accomplished using the nonrestoring method. When the operation is complete, the quotient is in the accumulator.

4.3.4 Memory

The memory is divided into two parts: the data memory and the program memory. Both are solid-state types, well suited to large-scale-integration techniques.

A small, random-access data memory is used as a scratchpad. It includes certain addressable arithmetic registers, temporary storage for intermediate computations, and buffer storage for the input/output unit. It contains 31 sixteen-bit words.

The program memory is read-only storing the fixed program and problem constants. It consists of 256 ten-bit words.

4.3.5 Input/Output Unit

A straightforward, flexible input/output unit has space assigned in the data memory location for each A/D converter input word and each D/A output word. The Input/Output (I/O) unit accesses the data memory directly, putting in words as they are converted and taking out data as needed for the D/A converter. The program uses the data in these specified locations without reference to the I/O unit. The data transfers occur in synchronism with the computer clock, but the converters themselves operate asynchronously. They need operate only fast enough to make each conversion approximately once per computation cycle.

4.4 Cost Estimate

The cost estimate is based on the use of MOS solid-state memories, TTL logic including MSI, full military temperature-range parts, and quantity purchase prices. The current sales price of the digital system in production would be approximately $3000.

The failure rate of conventional transducers deserves further discussion. Vanes and probes typically advertise MTBF's on the order of 10,000 hours. These numbers never reflect what really happens to a probe or vane after mounting on an aircraft. The transducer is used as a handle, step, brace, etc. No mounting position has yet been devised to eliminate the handling and environmental problems. Thus, it would appear that the 10,000-hour calculated MTBF must be judged with extreme prudence.

Semiconductor prices have shown a considerable shift downward each year. An accurate prediction of price in 3 to 4 years is difficult to obtain, but it would not be unreasonable to reduce the $3000 figure by an amount ranging from $250 to $500. At the same time, the recent sophisticated vane and probe prices have approached $2000. With the more stringent environmental requirements expected for future generation aircraft, vane and probe prices may well approach

the computer cost. When the additional aspect of reliability is injected, the computer (MTBF $\approx$ 10,000 hours) cost effectiveness compares favorably with conventional transducers (MTBF ranging from 37 to 2500 hours).

One of the largest items of expense is the analog-digital interface equipment. Eventually this can be greatly reduced by interfacing with only digital equipment. Digital air data computers and accelerometers having pulse outputs presently exist. In addition, instruments and displays that accept digital inputs have been developed. If only a digital interface is required, the price estimate for the angle-of-attack computer is $2200.

5. SAMPLED-DATA COMPUTER

In Reference 5, a sampled-data computer technique is described in which the primary problem variables are represented as analog voltages, and amplifier gains are represented by digital and discrete control functions. The potential advantages of this technique are increased flexibility and programmability of a digital computer wi hout the penalty of A/D and D/A conversion equipment. The digital gain control function is achieved by converting the digital input into a pulse width that is used to pulse-width modulate the analog signal. Multiplications are performed by averaging the pulse-width modulated signal in an integrator circuit. The digital pulse-width modulator is time shared by all the using analog amplifiers in the system.

Nonlinear functions can be computed in the sampled-data computer. By incorporating an analog comparator in the basic computer, it is possible to evaluate integrals of the form

$$z(x) = \frac{1}{w} \int_0^X y(x') \, d\, x'$$

where x and w are independent analog variables, and y(x') is any function of the machine variable x'. With such integrals, it is possible to perform a great variety of functions such as logarithms, exponentials, and sine-cosines, as well as analog multiplication and division.

The sampled-data computer is not suitable for angle-of-attack computation because

- It is not very accurate when compared to a good-quality analog system; it is limited in accuracy to approximately 5 percent.
- The digital pulse-width modulator, even when multiplexed and time shared, adds greatly to the complexity. The only advantage is flexibility of programming, which has very little value in either the simple- or high-quality angle-of-attack computer systems.

SECTION VI

CONCLUSIONS

There are three practical methods of computing aircraft angle of attack from internally mounted, previously available sensors. The first method uses a vertical gyro, an air data computer, and a side-slip angle transducer to produce inertial α. The second method requires three body-axis mounted accelerometers, an air data computer, and a side-slip angle transducer and, like the first method, yields inertial α. The third method uses a normal accelerometer, an elevator position transducer, a flap transducer, a throttle position transducer, and an air data computer to produce true α. A calculation of mass along with accurate knowledge of two key airframe parameters is also required.

Of these three approaches, only Method III has the ultimate capability of computing errorless angle of attack under all conditions of flight; the first two methods exhibit errors when flying in turbulence. On the other hand, Method III is considerably more complex than the other two, requiring knowledge of aircraft parameters, a mass calculation, and a sophisticated mechanization. The two inertial methods are easily mechanized and are universally adaptable to any aircraft type.

It is concluded that a combination of Methods I and II would be adequate for transport applications. Fighter-type aircraft would require a mechanization of the third method, while bomber types fall in between but lean to the third method.

The initial cost of an angle-of-attack computer will probably exceed that of a vane or probe transducer. However, the superior reliability of the computer is expected to more than compensate for the price differential when maintenance costs are considered. The angle-of-attack computer also offers potentially higher accuracy than the external devices.

APPENDIX

AIRCRAFT DATA

The data presented in this appendix represents the aerodynamic characteristics of a high-performance tactical fighter aircraft. It is a composite from several different data sources and does not necessarily represent a particular aircraft model. Of the eleven flight conditions available, six were used in the study program and only those six are presented.

TABLE IV

AERODYNAMIC DATA FOR SIMULATED AIRCRAFT

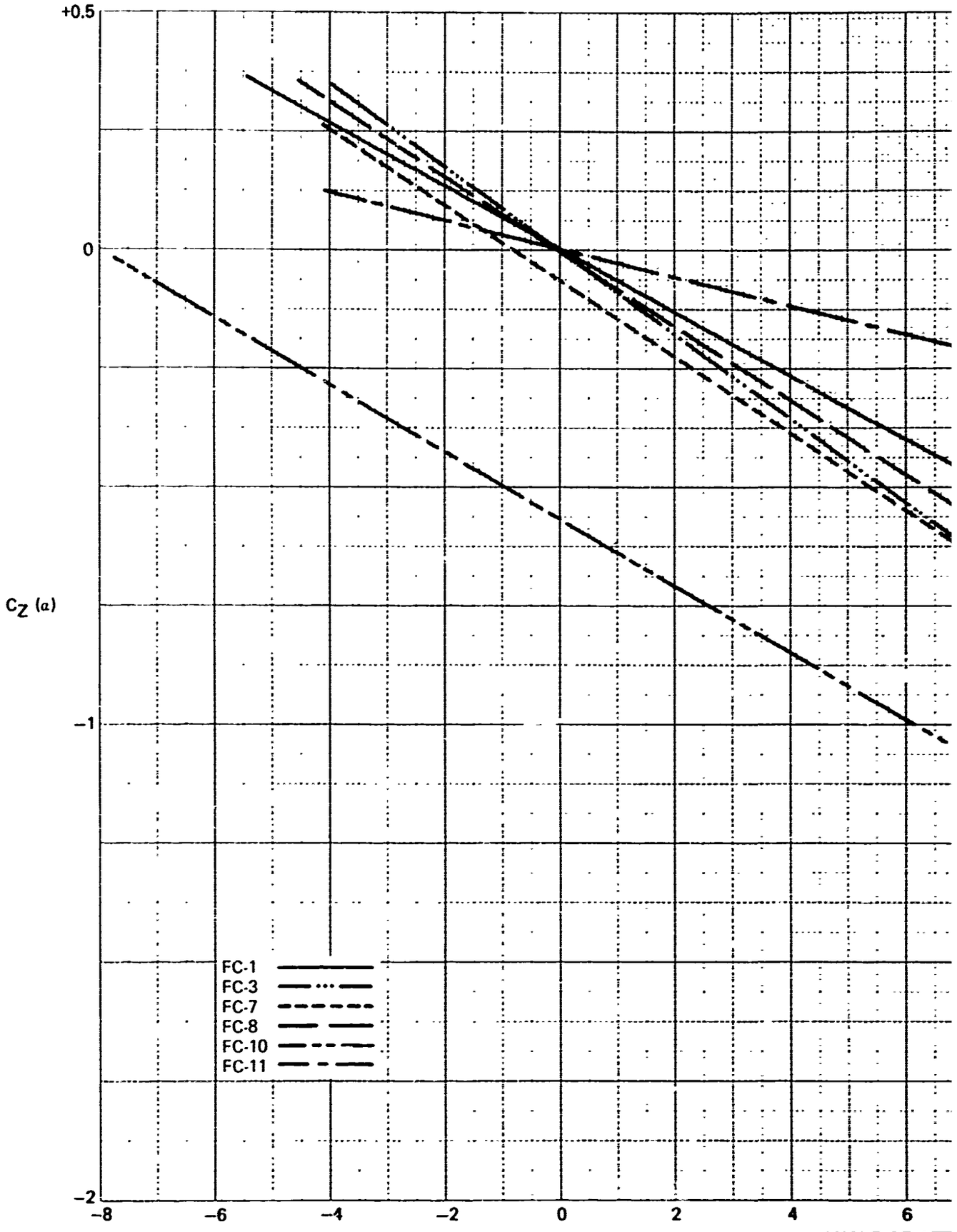
Parameter	Flight Condition					
	1	3	7	8	10	11
h(ft)	0	0	15,000	35,000	0	35,000
Mach	0.25	0.9	1.1	0.6	0.154	2.5
P(slug/ft^3)	0.002378	0.002378	0.001496	0.000736	0.002378	0.000736
$\bar{V}_a$ (ft/sec)	279.3	1004.6	1164.5	584	171.9	2429.2
$q = P\bar{V}_a^2/2$(lb/ft^2)	92.70	1200	1010	126	35.2	2170
α_o (deg)	11.2	2.1	2.9	7.5	7.5	0.8
U_o (ft/sec)	274	1004	1163	579	170.5	2429
W_o (ft/sec)	54.2	36.8	58.9	76.2	22.4	33.9
δ_{e_o} (deg)	-7.4	-3.8	-4.95	-5.5	-5.5	-1.8
C_{y_β}	-0.885537	-0.779997	-0.905967	-0.714662	-0.900031	-0.643002
Cy_{δ_a}	-0.014978	-0.013005	-0.004499	-0.022528	-0.006670	-0.004640
Cy_{δ_r}	0.235050	0.094996	0.039988	0.225283	-0.282005	0.062601
$C_{z_{\delta_e}}$	-0.568179	-0.480165	-0.393634	-0.624148	-0.600101	-0.150010
Cm_{δ_e}	-0.855987	-0.710445	-0.634982	-0.950073	-1.14000	-1.31291
$Cm_{\dot{\alpha}}$	-0.660375	-0.839857	1.197740	-0.820280	-1.00135	2.00191
Cm_q	-3.787501	-3.542457	-4.056669	-4.140056	-6.12003	-4.99989
C_{L_β}	-0.123596	-0.079564	-0.070374	-0.113761	-0.164400	-0.006850
$C_{L_{\delta r}}$	0.026750	0.012195	0.007703	0.027221	0.007410	0.016651
CL_p	-0.289997	-0.393388	-0.405630	-0.312417	-0.441001	-0.181412
$CL_{\delta a}$	0.053330	0.019453	0.011381	0.058537	-0.077100	0.009411
CL_r	0.184332	0.069535	0.063858	0.147335	0.500020	0.032899
CN_β	0.090368	0.083167	0.114795	0.075179	0.126700	0.021700
$CN_{\delta r}$	-0.101924	-0.045250	-0.023636	-0.099188	-0.123200	-0.015897
$CN_{\delta a}$	0.008559	0.001806	0.002284	0.011366	-0.017000	0.001655
CN_p	-0.000126	0.020995	0.040992	-0.021406	-0.012700	-0.002981
CN_r	-0.304765	-0.322117	-0.377810	-0.305616	-0.668000	-0.146700

Constants:

$s = 375$ ft^2 $\quad$ $b = 38.7$ ft $\quad$ $c = 10.8$ ft

$W = 21,889$ lb $\quad$ $M = 680$ slugs $\quad$ cg at 30 percent MGC

$I_x = 13,635$ slug-ft^2 $\quad$ $I_y = 58,966$ slug-ft^2 $\quad$ $I_z = 67,560$ slug-ft^2 $\quad$ $I_{xz} = 2,933$ slug-ft^2


+0.5
0
−1
−2
C_Z (a)
−8
−6
−4
−2
0
2
4
6
ANGLE OF ATT
FC-1
FC-3
FC-7
FC-8
FC-10
FC-11

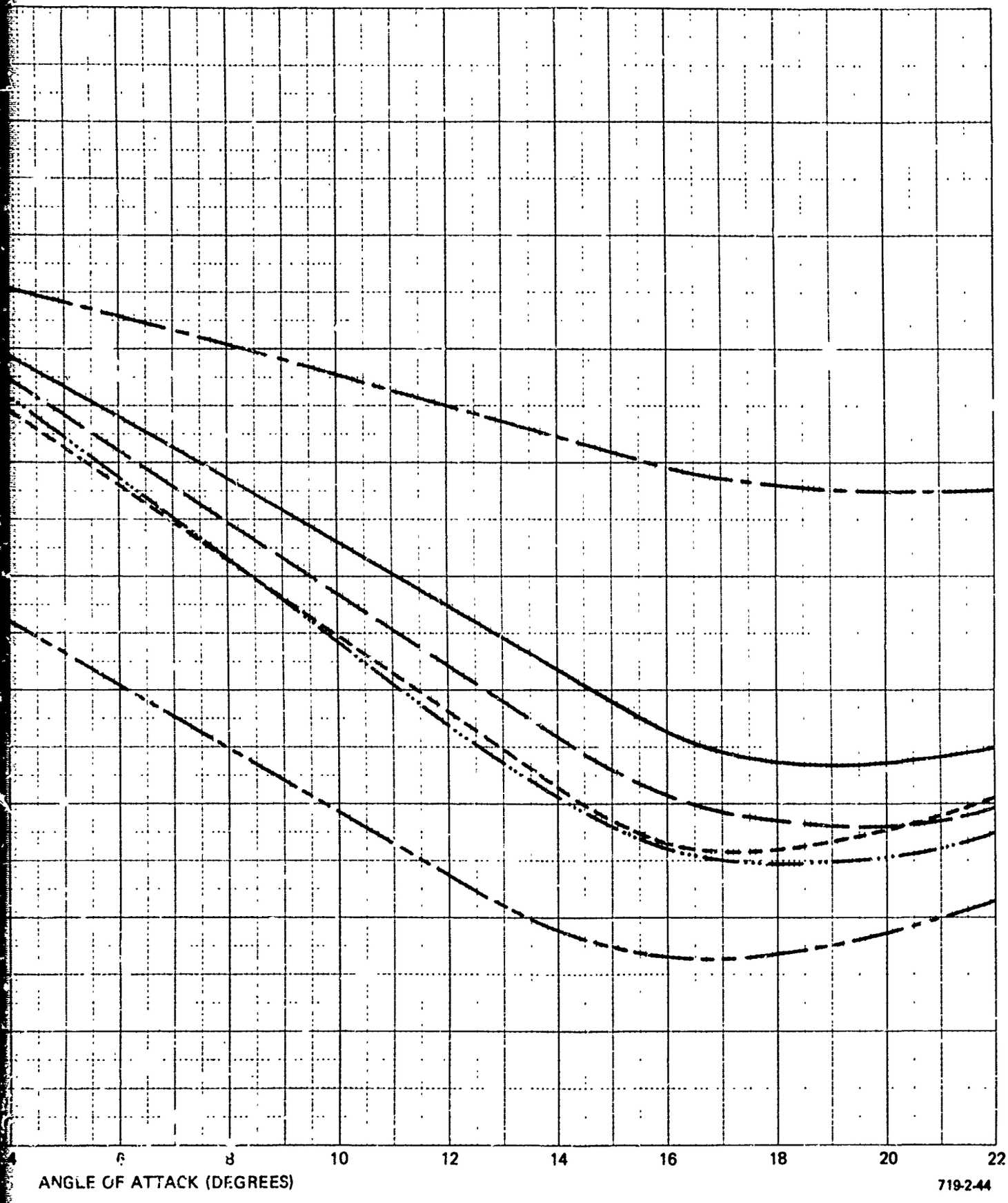


Figure 30
Stability Derivative C_z (α) versus α

B

REFERENCES

1. McRuer, D.T., I. Ashkenas, and D. Graham, Aircraft Dynamics and Automatic Control, Systems Technology, Inc., August 1968.

2. Atmospheric Turbulence and Its Relation to Aircraft, Proceedings of a Symposium Held at the Royal Aircraft Establishment, Farnborough, England, November 1961.

3. Renty, P.E., Evaluation and Utilization of Airplane Flight Loads Data, AFFDL-TR-68-52 Part II, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, May 1968.

4. Peters, W.J., and T.S. Momiyama, Flight Evaluation of the Instantaneous Angle of Attack Display Final Report, FT2222-13R-63, Naval Air Test Center, U.S. Naval Air Station, Patuxent River, Maryland, July 1963.

5. Close, K.E., S.P. Liden, J. Williams, and J.B. Dendy, Use of Sampled-Data Computing Techniques for Advanced [illegible]y Flight Controls, Naval Air Development Center Final Report, AD-824283L, May 1967.

BIBLIOGRAPHY

Atmosphereic Turbulence and Its Relation to Aircraft, Proceedings of a Symposium Held at the Royal Aircraft Establishment, Farnborough, England, November 1961.

Baldwin, A.W., B.L. Perry, R.H. Shields, and H.P. Birmingham, A Method for the Establishment of the Quickening Terms for an Improved Binary Angle of Attack Display, Engineering Psychology Branch, Applications Research Division, U.S. Naval Research Laboratory, Washington D.C., July 1962.

Brink, P.L., The Determination of the Angles of Attack and Skid from the Measurement of Aircraft Accelerations, NAVORD Report 1392, U.S. Naval Ordnance Plant, Indianapolis, Indiana, September 1951.

Close, K.E., S.P. Liden, J. Williams, and J.B. Dendy, Use of Sampled-Data Computing Techniques for Advanced Navy Flight Controls, Naval Air Development Center Final Report, AD-824283L, May 1967.

Cover, J.H., N.F. Witte, and L.W. Seely, Angle of Attack and Skid Measurements for Aircraft Rocket Fire Control Purposes, U.S. Naval Ordnance Test Station, China Lake, California, November 1951.

Emerson, F.M., F.H. Gardner, G.D. Gruenwald, R. Olshausen, and L.V. Sloma, Study of Systems for True Angle of Attack Measurement, WADC Technical Report 54-267, Wright Air Development Center, Air Research and Development Command, United States Air Force, Wright-Patterson Air Force Base, Ohio, May 1955.

McRuer, D.T., I. Ashkenas, and D. Graham, Aircraft Dynamics and Automatic Control, Systems Technology, Inc., August 1968.

Peters, W.J., and T.S. Momiyama, Flight Evaluation of the Instantaneous Angle of Attack Display Final Report, FT2222-13R-63, Naval Air Test Center, U.S. Naval Air Station, Patuxent River, Maryland, July 1963.

Renty, P.E., Evaluation and Utilization of Airplane Flight Loads Data, AFFDL-TR-68-52 Part II, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, May 1968.

Sansom, F.J., and H.E. Peterson, Mimic Programming Manual, SEG-TR-67-31, Systems Engineering Group, Aeronautical Systems Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, July 1967.

Unclassified

Security Classification

DOCUMENT CONTROL DATA · R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Sperry Flight Systems Division
Sperry Rand Corporation
Phoenix, Arizona

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
None

3. REPORT TITLE
Angle-of-Attack Computation Study

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report, 1 December 1968 to 31 July 1969

5. AUTHOR(S) (Last name, first name, initial)
Dendy, Joseph, B.
Transier, Kent, G.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1969	102	5

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
F33615-69-C-1178	71-0004-00-00
b. PROJECT NO. 8222	
c. Task No. 822207	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	AFFDL-TR-69-93

10. AVAILABILITY/LIMITATION NOTICES
"This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the AF Flight Dynamics Laboratory (FDCL)."

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Air Force Flight Dynamics Laboratory Wright-Patterson Air Force Base, Ohio

13. ABSTRACT

This report discusses methods of computing angle of attack by inference using combinations of data from presently available, on-board sensors, thereby eliminating the need for external vanes or probes. Equations were derived from which computed angle of attack could be extracted. Those equations that were impractical from a mechanization standpoint were eliminated, leaving three candidate methods. These three methods were then analyzed with respect to errors arising from mathematical simplifications and errors due to imperfect sensor information. Two of the candidate methods provide inertial angle of attack, and will provide acceptable accuracy for low-performance aircraft applications. The third method will provide a high-quality, air-mass-related angle of attack. A mechanization of a high-quality angle of attack system using a small, special purpose digital computer is described. Using measurements of normal acceleration, longitudinal acceleration, elevator position, flap position, throttle position, airspeed, Mach, and dynamic pressure, the system will provide a high-quality angle of attack measurement applicable to any high-performance aircraft and competitive with current vane and probe transducers.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
On-Board Sensors						
Inertial Angle of Attack						
Air-Mass-Related Angle of Attack						
Digital Computer						
Aircraft Applications						
Normal Acceleration						
Longitudinal Acceleration						
Elevator Position						
Flap Position						
Throttle Position						
Airspeed						
Mach						
Dynamic Pressure						