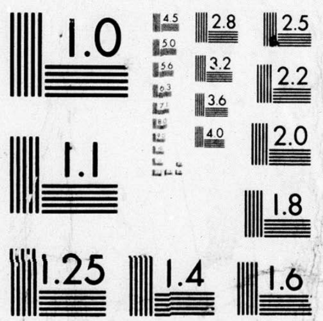
AD-A041 676 OMNEMII INC VIENNA VA F/G 12/1
DEVELOPMENT OF A CONTINUOUS PERFORMANCE MEASURE FOR MANUAL CONT--ETC(U)
APR 77 E M CONNELLY, R M ZESKIND, G P CHUBB F33615-75-C-5088

UNCLASSIFIED AMRL-TR-76-24 NL

1 OF 2
AD
A041676


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

AMRL-TR-76-24

2

12

DEVELOPMENT OF A CONTINUOUS PERFORMANCE MEASURE FOR MANUAL CONTROL

E. M. CONNELLY
R. M. ZESKIND

OMNEMII INC.
410 PINE STREET, S. E., SUITE 200
VIENNA, VIRGINIA 22180

G. P. CHUBB

AEROSPACE MEDICAL RESEARCH LABORATORY

COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

APRIL 1977

Approved for public release; distribution unlimited

AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the 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.

Please do not request copies of this report from Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Documentation Center should direct requests for copies of this report to:

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

TECHNICAL REVIEW AND APPROVAL

AMRL-TR-76-24

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Charles Bates

CHARLES BATES, JR.
Chief
Human Engineering Division
Aerospace Medical Research Laboratory

AIR FORCE - 20 JUN 77 - 100

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 AMRL-TR-76-24	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 DEVELOPMENT OF A CONTINUOUS PERFORMANCE MEASURE FOR MANUAL CONTROL.		5. TYPE OF REPORT & PERIOD COVERED 9 Final Report. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 E. M. Connelly*, R. M. Zeskind*, and G. P. Chubb**		8. CONTRACT OR GRANT NUMBER(s) 15 F33615-75-C-5088 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS * Omnemii Inc. 410 Pine St. (Suite 200) Vienna VA 22180 ** Aerospace Medical Research Lab, WPAFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62202F 7184-13-04 16 17
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratory Aerospace Medical Division, AFSC Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE 11 Apr[redacted]77 13. NUMBER OF PAGES 167
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 166p.		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Optimal Control Theory, Tracking Manual Control, Performance Measurement, Man-Machine Relationships, Human Engineering, Systems Effectiveness

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The performance measurement concept typically applied to manual control systems employs a summary measure which provides a numerical score to represent performance of the total control problem. While summary measures are necessary for evaluating total performance they do not reveal information about control actions that occur during the control problem. If performance can be measured and evaluated continuously throughout the control problem each control action can be evaluated rapidly and thus individually. Also if over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391 705

Block 20

the continuous measurement and evaluation can be related to the summary measure, each control action can be evaluated as an entity, as well as, part of the total control problem.

The desired continuous performance measure (CPM) provides a continuous indication of the correct motion of the aircraft at each point in a mission segment and what is also important, an indication of the significance of any motion errors with respect to the summary measure performance selected. This application of the continuous performance measure is termed the "Mission Model" since the model can provide both reference aircraft motion and an evaluation of the flight control errors. The objective of the research reported here is to evaluate the feasibility of continuous performance measures for aircraft systems, and the development of the necessary computational tools.

This report provides a description of the test mission, mission segments, and segment specifications. A mathematical model for the aircraft dynamics along with the development of the continuous performance measure for a mission segment are included. A computational technique for solving the CPM for a mission segment is also included.

SUMMARY

Man-in-the-loop simulation is used both to evaluate proposed system designs and to train new operators. Historically, measures of the man's performance have often been confounded with measures of the system he is asked to control. While the intent has been to get a measure which has operational meaning and predictive validity, confounding system performance with operator performance can obscure important information. The intent here was to explore the feasibility of developing a measure that would provide a better scoring procedure for manned system simulation, whether for research or training: a procedure which reflected the impact of inappropriate operator action which did not also include penalties for factors beyond the operator's control. For example, summary measures such as RMS error or integrated absolute error can confound operator control actions with turbulence induced excursions from a desired flight path. Also, summarized measures do not provide any guidance as to how one makes the best of a bad situation. They do not prescribe a desired or appropriate course of action sensitive to the objectives of the task and conditional upon the circumstances prevailing when action is required.

Optimal control theory was used as the basis for formulating the continuous performance measurement approach developed here. The original goal was to develop a measure that was sensitive to the information displayed to the operator so one could "assign cause" for inappropriate actions taken by the operator. It was also desirable to develop the measure for all segments of the mission. It was soon discovered that such ambitions were beyond the scope of this effort and attention was devoted to simpler issues and a single, well defined mission segment. The report documents the overall mission analysis but develops the performance measure only for the cruise phase.

One of the impediments was the problem context chosen for study. The optimal control theory requires the formulation of an explicit model in order to define what actions are appropriate if one wishes to achieve specified objectives. The model includes a description of the task the operator is asked to perform. Since the measure would potentially be applicable to a planned series of experiments, the model of the aircraft control task used to develop the continuous performance measure was borrowed directly from the real-time simulation program developed for those studies.

This report describes the mission that was of interest and rationalizes the model proposed for experimental studies. While the

specifics of the continuous performance measure developed, implemented, and briefly studied here are based upon that mission and aircraft model, the general philosophy and approach are believed to be applicable to a broad class of problems. The specifics of the continuous performance measure appropriate for these other applications will require a similar development: quantification of mission objectives, descriptive modeling of the task (the aircraft or plant dynamics), derivation of the optimal feedback control law (including solution techniques appropriate to whatever model is developed), and finally, the construction and implementation of the continuous performance measure.

Additional work will be required to explore the unanswered questions and to gain experience with these measures in contrasting them with the more conventional measures now employed. While their utility in human factors research was the original justification for the development of continuous performance measures, they also appear attractive as measures useful in a training environment where an instructor wishes to single out appropriate and inappropriate student actions, calling attention to the impact these actions have on mission effectiveness.

The findings of this study substantiate the feasibility of developing a continuous performance measure. Unfortunately, they also confirm the "curse of dimensionality" alluded to by Bellman: complicated problems lead to tedious calculations if solvable at all. The measure as presented in this report is not calculable in real-time as had been desired, but insight has been gained that may lead to more practical and faster solutions if not an entire new approach. Until further experience is gained in applying the technique, no conclusions can be made about the total merits of the method. While the results were inconclusive, they should encourage further development, particularly for mission phases not treated here.

PREFACE

This study was initiated by the Human Engineering Systems, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio. The research was conducted by Omnemii, Inc., Springfield, Virginia, under contract F33615-75-C-5088. The work is in support of Air Force Project Number 7184, "Human Engineering for Air Force Systems," Task 718413, "Man Machine Models for System Performance." Research sponsored by this contract was performed between October 1974 and 1975, under contract number F33615-75-C-5088.

Omnemii's principal investigator was Mr. E. M. Connelly, Dr. R. M. Zeskind accomplished the majority of the analytical work and Mr. R. F. Comeau and Mr. Ignacio Huerta completed the computer programming. The Air Force project engineer was Mr. Gerald P. Chubb.

The authors wish to thank Dr. Richard A. Miller, Assistant Professor of Industrial and Systems Engineering at Ohio State University for his thoughtful and provocative critique of the theoretical implications of this work and for his foresight in anticipating the pragmatic difficulties of trying to apply the theory in practice. The authors are also indebted to Mr. P. V. Kulwicki of the Aerospace Medical Research Laboratory for his review of the mathematical model of the aircraft borrowed for and documented in this study.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTD)

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

A performance measurement concept typically employed for manual control systems uses a summary measure which provides a single numerical score to represent performance of the total control problem. While summary measures are necessary for evaluating total problem performance, for instance, in comparing performance of competitive systems, summary measures do not reveal performance information about control actions that occur during the control problem. If performance can be measured and evaluated continuously throughout the control problem, each control action (or series of actions), whether continuous or discrete can be evaluated rapidly and thus individually. Also, if the continuous measurement and evaluation can be related to the summary measure thereby indicating the effect of a control action on the associated summary measure, each control action can be evaluated as an entity, as well as, part of the total control problem. This performance measurement concept specifically permits identification and evaluation of the significance of operator error patterns, and identification of critical and sensitive regions of the control problem.

This type of measure - a continuous performance measure (CPM) - is a tool which the authors believe could be used to increase the efficiency of experiments, training, and design of manual control systems. For example, the tool can allow evaluation of experimental results on a portion of a control problem with respect to the effect on the total mission. Rapid evaluation of control errors can facilitate training efficiency, and knowledge of control sensitivity to total mission performance permits concentration of design effort on critical areas. This report documents research on a method of continuous performance evaluation of manual flight control systems.

The desired CPM provides a continuous indication of the correct motion of the aircraft at each point in a mission segment - thus providing flight criteria against which actual aircraft motion can be compared - and what is also important, an evaluation of the significance of any motion errors with respect to the summary performance measure selected. This application of continuous performance measurement is termed "Mission Model" since the model can provide both the reference aircraft motion and an evaluation of flight control errors. The objective of the research reported here is to investigate the feasibility of continuous performance measure for aircraft systems, and demonstrate the development of the necessary computational tools.

The mission model considered here is based on an optimal control concept which relates performance during the mission to summary performance. When a summary measure is selected for a mission or mission segment, optimal control theory can be used to determine for each aircraft state in the mission segment, the optimal control and associated optimal solution trajectories. The term "optimal solution trajectories" used here means the aircraft motion trajectories that minimize the summary performance measure selected for the mission segment. Also the term "aircraft state" refers to a set of state variables that provide a complete description of the aircraft (to the extent it is represented by the aircraft model used) by identifying values for all positional and rotational variables, as well as, their velocities.

In order to understand how optimal control theory can be used to determine the instantaneous effect of control actions on the summary measure, consider a control problem, shown in Figure 1, where the aircraft is presently at Point A and the control objective is to direct the aircraft to Point B. When the aircraft reaches Point B the segment control problem is completed and performance for that segment is evaluated according to the selected summary measure. Only those flights which satisfy the control requirement by reaching Point B (or within a specified tolerance) are considered for scoring. If one (or more) solution trajectory from Point A to Point B incurs a minimum value of the selected summary measure, that trajectory (or those trajectories) is the optimal one sought. The optimal trajectory, say Path 1 of Figure 1, may be considered as a reference path from Point A and each point along the path to Point B. Optimal control theory is the mathematical tool that can be used to find the optimal trajectory from each point in the region of interest in the mission segment.

To determine what control is "optimal" it is necessary to construct several model components. These include: (1) a representation of the goal of the current task in terms of a set of objectives and weights reflecting their relative importance, (2) a representation of the system being controlled, e.g., the equations of motion for an aircraft of interest (suitably detailed for the nature of the problem to be addressed), and (3) a representation of the physical or other constraints that limit where the system can go or what controls can be applied. Mathematical techniques are used to solve this set of models to define a rule for choosing the control which leads to the "best" outcome for these given objectives, system, and constraints. Applying this rule leads to an optimal trajectory.

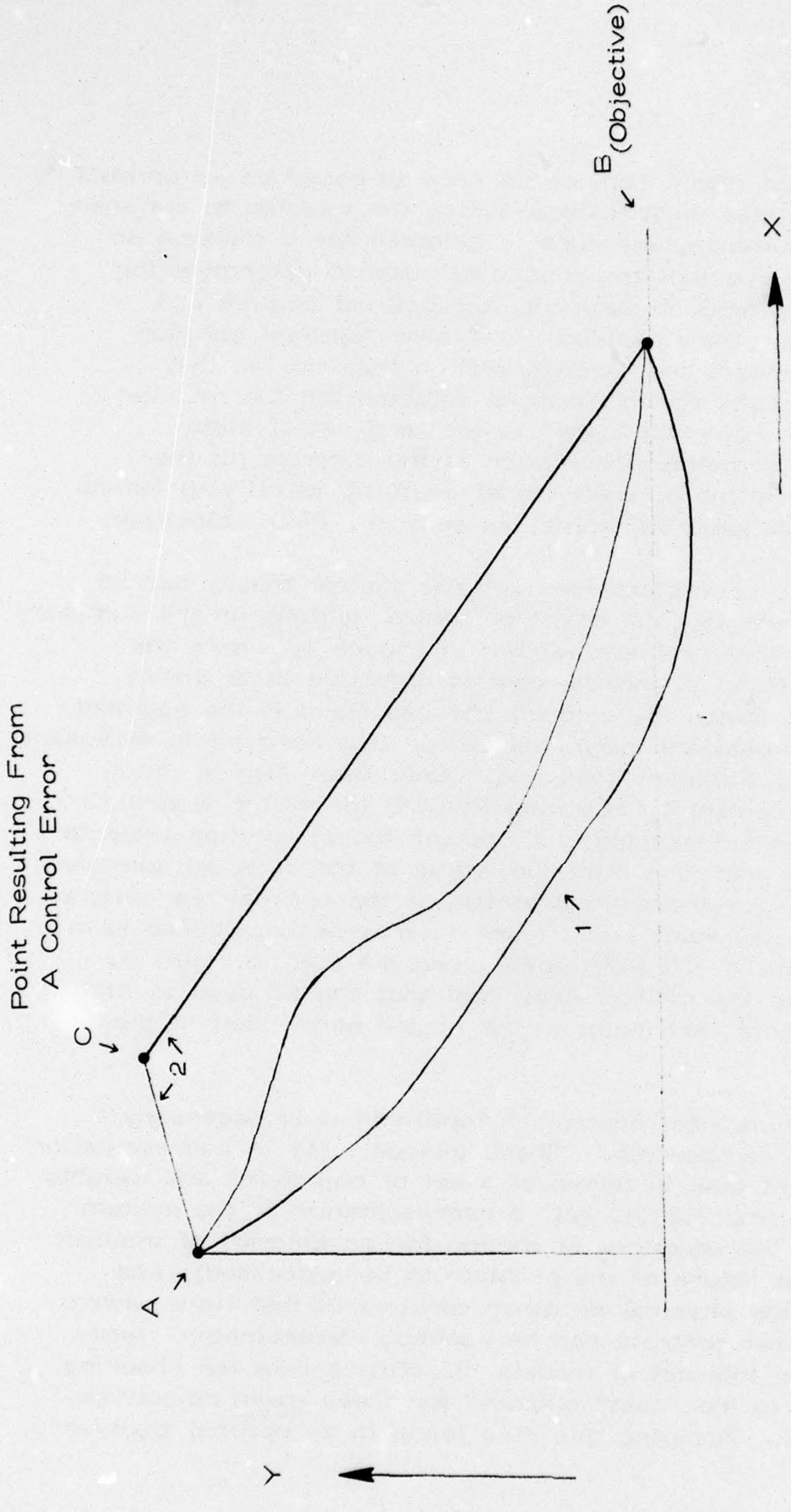


FIGURE 1 EXAMPLE AIRCRAFT TRAJECTORIES

The theory assumes the objective, system, and constraints as mathematically represented are an accurate and complete description. To the extent the models fall short, so will the theory's ability to meet one's subjective or intuitive notion of "optimal." For example, if "my" objectives differ from "your" objectives - or if we but weight the same objectives differently, then the theory produces a control rule for "me" that may differ from the control rule for "you." While this sensitivity is often desirable, it is apparent that acceptance of the defined "optimum" requires concurrence in each element of the model. If agreement is not reached, the theory does not apply except to each separate model proposed, in which case it must be expected that the results will differ if the models do. In short, what is optimal is relative to the associated models, and here we must assume there is no argument with the objectives, system, and constraints as posed. If there were argument, the first step would be to change the models proposed for whatever element was questioned, revise it appropriately, and then proceed with optimization with the assurance that agreement was reached. The reader is therefore asked to accept the models as proposed simply to facilitate the subsequent analysis. It is to be understood that exceptions to the model would demand reformulations specific to each reader's criticisms - a task obviously impossible a priori - which would, at least in principle, dispense with these criticisms. In a sense, criticism of the specific model is peripheral to the main theme of this report. The emphasis is on an approach not a specific application much less a single result. On the other hand, much of the specifics developed here must be repeated as the models are changed. The philosophy is what is generalizable. This report attempts to demonstrate the feasibility of implementing that philosophy in a given context, the participating critiques (namely the authors) having been satisfied - at least at this juncture.

The optimal control analysis can also provide error sensitivity information. Error sensitivity weighting is important because it allows direct evaluation of control errors and further reveals the regions of high control error sensitivity in the mission segment. In order to see how the error sensitivity is determined, assume that the pilot controlling the aircraft at Point A does not provide the correct control and as a result the aircraft moves to Point C. Thereafter, the pilot employs optimal control directing the aircraft along the optimal path from Point C to Point B. Note that Path C-B can be totally different from Path A-B. Except for the control error of short duration which moved the aircraft from A to C, the pilot used optimal control. The difference between the summary measures for Paths 1 (the optimal solution from Point A to B) and 2 (the trajectory from Point A to C and the optimal solution from C to B) must be the increase in the summary measure due to the initial control error. There cannot be a decrease in the summary measure since that would imply that trajectory 1 is not optimal. The amount of increase in the summary measure value due to the initial control error is the effect of the control error on the summary measure. The type of mission model considered here provides this sensitivity weighting for each incremental aircraft motion and thus provides an instantaneous weighting of control errors.

Military aircraft missions are not usually defined solely with a summary measure, but typically are defined by a flight profile consisting of a series of flight maneuvers along with objective values for appropriate state variables such as velocity, heading, altitude, and rate of climb. Frequently, a mission can be segmented so that a consistent set of flight variable specifications can be defined throughout each segment. Segments may be and typically are defined by a reference flight path along which some state variable values are given. Desired terminal conditions indicating the preferred state variable values at the end of the segment may also be available. The total mission may be viewed as a series of segments where the end flight conditions of one segment are the initial conditions for the next segment. Thus in order to construct a segmented Mission Model, the mission segment specifications must be converted to a summary measure and any required flight constraints.

Conversion of mission specifications into a summary measure and reference trajectories requires construction of a penalty function (cost index function in control theory terminology) which identifies the relative importance of:

1. Deviation from the desired terminal state, and

2. Variable rates of change, control actions, and deviations from reference trajectories occurring along the solution path.

The cost function is selected by study of the requirements of each mission segment. It reflects the nature of the objectives for the segment and the relative importance of each. For example, in some segments straight and level flight is desired at a specified heading, altitude, and velocity. In other segments a constant climb or dive may be desired. In yet other segments, more complex coordinated maneuvering may be required. The cost or penalty function is constructed to incorporate these reference maneuvers so that if the aircraft flys along the reference path, zero penalty is incurred. For instance, if constant altitude (Z) and velocity (V) are desired, the terms

$$I = \int_0^T (W_1(V-V_R)^2 + W_2(Z-Z_R)^2 \ldots)dt$$

might be used, where V_R and Z_R are the reference values, and W_1, W_2 are weighting values. This expression proposes that the task objective will be to minimize the weighted squared deviations or errors. These weighted squared errors are then integrated (if time is a continuous variable, summed if time were treated as a discrete variable) from the start of the segment (t = 0) to its end (t = T). The result is a single number (I).

The cost function also evaluates trajectories not along the reference-trajectories, with initial conditions and displacements that occur due to control errors or wind disturbances. But recall that whatever the present state, there is an optimal solution from that state to the terminal point. Optimal solutions from points not on the reference trajectories are termed "Preferred trajectories" to distinguish them from the reference trajectories. Calculation of preferred trajectories is accomplished, as described previously, from optimal control theory employing the cost function and the aircraft equations. Thus the form of the cost function and the weighting constant values influence the calculation of preferred trajectories. The effect of a given weighting term or relative values of terms is usually not known prior to calculation of the preferred trajectories. Initial selection of weighting values might alternatively be accomplished by asking experienced pilots and/or other personnel experienced in the mission performance requirements to select weighting values or at least order the terms according to importance to mission success. Thus selection of weighting values may be an iterative process involving initial weighting selection, computation and evaluation of preferred trajectories, followed by an adjustment of the weighting values, etc.

1.1 Method of Approach

The mission selected for analysis is an aircraft operating in a Close Air Support Night Attack Mission where the overall mission was divided into segments: cruising, climbing, etc. Each segment of the mission may be considered as a separate problem in itself. The overall mission can be modeled as a sequence of sub-problems which the pilot must solve, where the terminal conditions of one segment serve as the initial conditions of the next segment. Individual segments are then cast in the form of an appropriate optimal control problem, the solution of which yields an optimal control law as a function of the problem state variables, i.e., feedback control law.

Once optimal trajectories and the feedback control law have been obtained for a given segment, a corresponding continuous performance measure (CPM) function is found for that segment. The CPM gives an instantaneous measure of actual man-machine system performance as contrasted to preferred or optimal performance.

The work was divided into the following subtasks:

1. Analysis of a Close Air Support Night Attack Mission to develop segments and segment specifications.

2. Formulate a cost index for an example mission segment.

3. Develop the optimum feedback control law for that segment.

4. Evaluate segment trajectories using the optimal feedback control.

5. Develop the continuous system performance measure and associated computer algorithms.

6. Using the above system performance measure, evaluate the segment performance using non-optimal control, that is, use a non-optimal autopilot model for the feedback control law.

7. Demonstrate the continuous performance measurement technique.

1.2 Overview of Report

Section 2 provides a description of the mission, mission segments, and segment specifications. The mathematical model for aircraft dynamics is given in Section 3 along with a description of the aircraft state variables selected for this problem. Section 4 contains the development of the continuous performance measure (CPM) including construction of CPM functions, and an illustrative example. Section 5 presents a specific application of the method to a mission segment. Conclusions and recommendations are presented in Section 6.

2.0 MISSION MODELS

This section includes a description of the Close Air Support Night Attack Mission, and a development of a mathematical model for the mission. The mission was broken into its segments, which can be treate as separate flight control problems. Each mission segment has its own reference and performance index functions.

2.1 Close Air Support Night Attack Mission

The Close Air Support Night Attack Mission was defined to be a night/clear VFR* attack against a power plant. Refueling is done en route to the target where rendezvous with a tanker is accomplished via UFH/ADF* procedures. Missile evasion is done en route to target to avoid radar detection and encounters with surface-to-air missiles (SAMs). After using FLIR* to locate the target, type MF-84 "dumb bombs" are dropped. The escape employs terrain-following until the forward-edge-of-battle-area (FEBA)* is reached. TACAN* navigation during the penetration segment is followed by a GCA landing (Figures 2 and 3 illustrates the mission). The mission segments and segment elements are as follows:

1. Preflight and Takeoff

Mission briefing, weapons selection, aircraft preflight, all communications equipment turned on, inertial system set up, engine start and systems checks, taxi checks, pre-takeoff checks, arming completed, takeoff accomplished. Takeoff speed is 140 kts.

2. Accelerate and Climb

Gear and flaps retracted, build speed to 330 kts at 5,000 ft., radar and radar homing and warning receiver (RHAW) turned on, TACAN turned on, external fuel tanks turned on, checks of jammers zero delay lanyard unhooked. After level off, set power, and check all systems. Climb speed is 330 kts.

3. Rendezvous and Refuel

Navigate to air refueling initial point with inertial systei contact tanker on UHF, and join tanker in formation using UHF/ADF

*Acronyms are defined in the Glossary for readers not familar with these terms; the Glossary is in Appendix A.

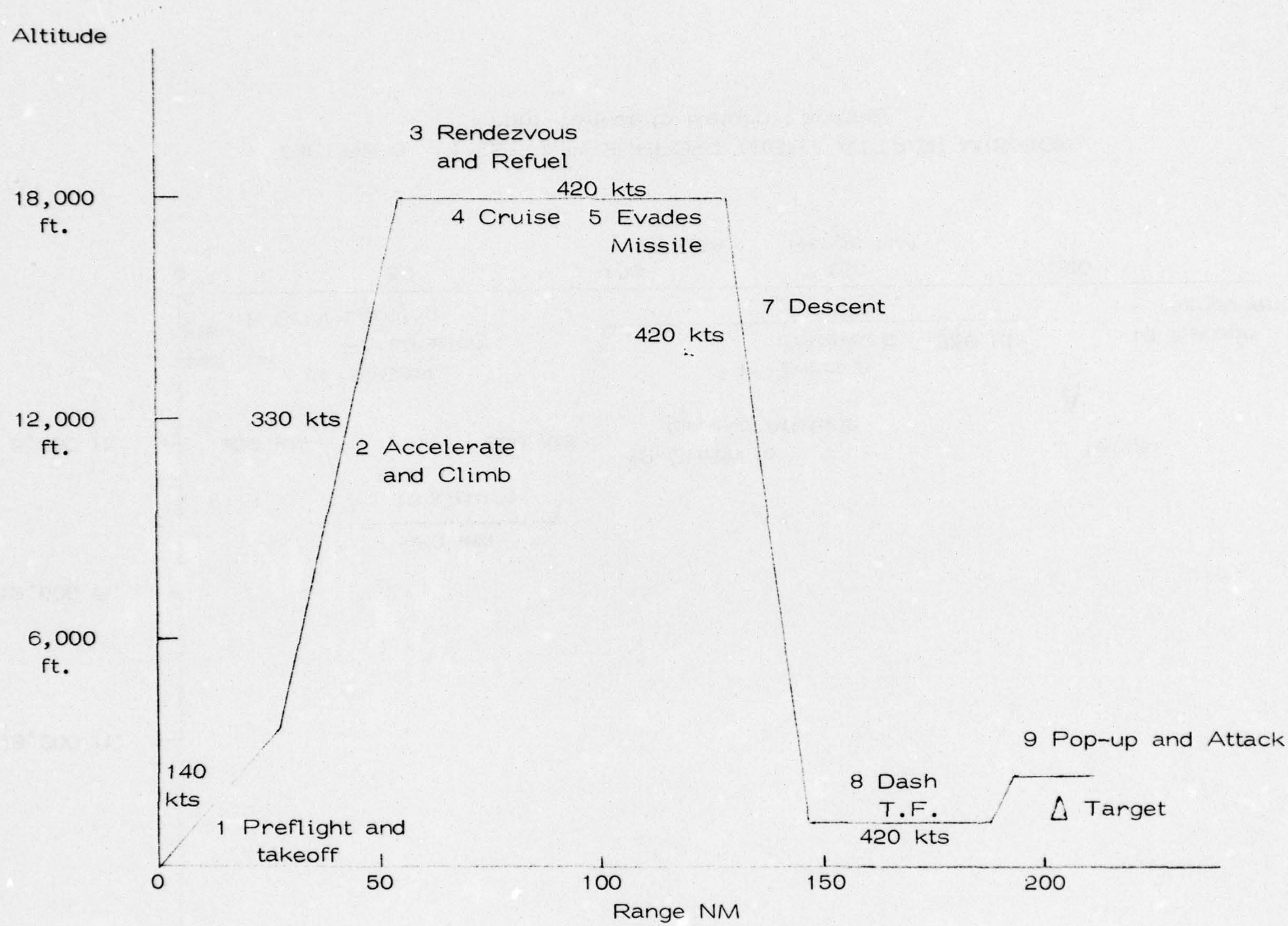


FIGURE 2 CLOSE AIR SUPPORT NIGHT ATTACK MISSION
(From Takeoff to Target)

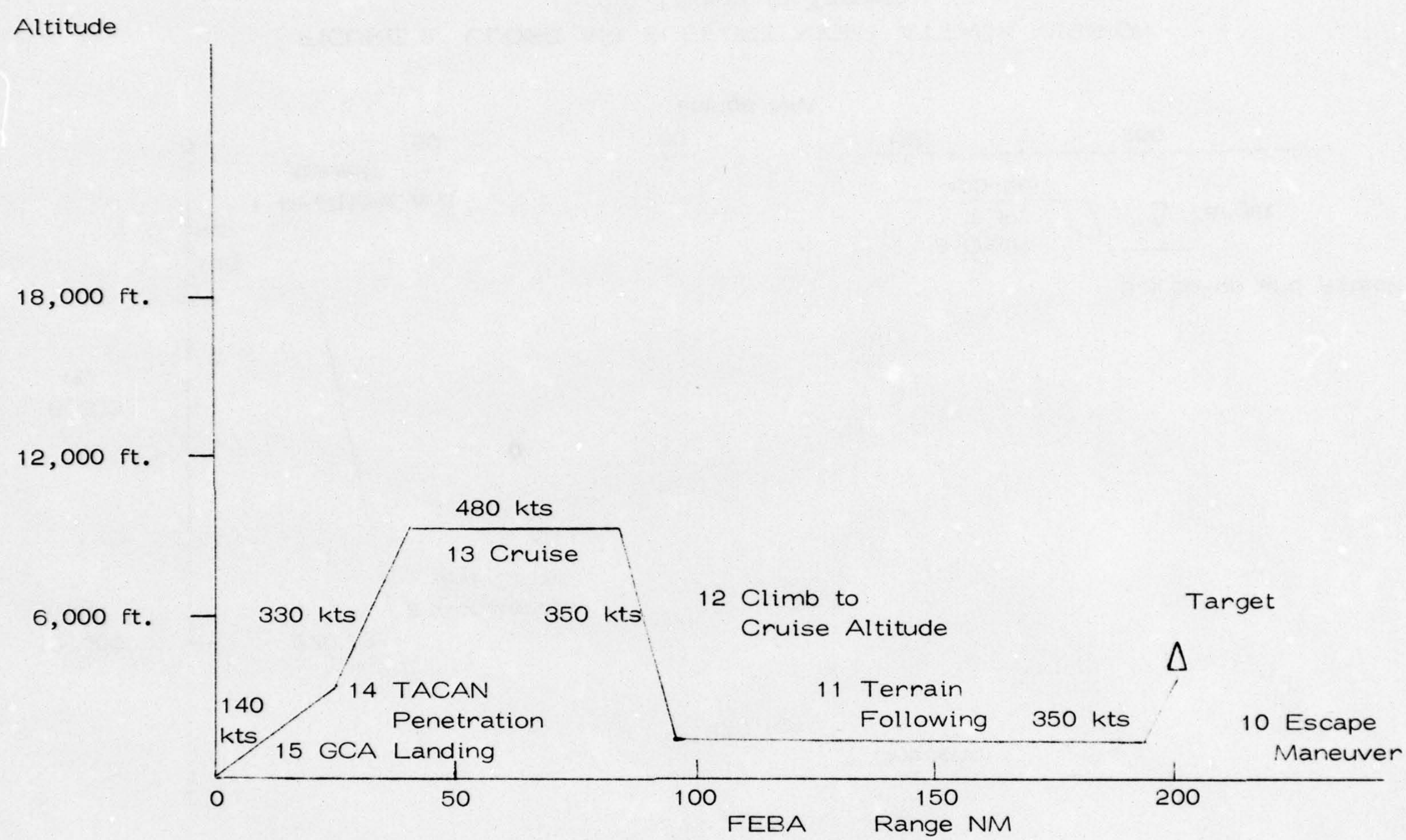


FIGURE 3 CLOSE AIR SUPPORT NIGHT ATTACK MISSION
(From Target to Return Landing)

procedures. Complete refueling and depart for target, navigating with inertial system.

4. Cruise

Cruise at an altitude of 18,000 ft. at a speed of 420 kts.

5. Missile Evasion

Arm MK-84's (dumb bombs). RHAW indicates missile threat, and ECM is turned on, successfully avoiding the threat.

6. Step 6 was deleted from the mission.

7. Descent

Descend to minimum terrain-following, VFR altitude. Use DOPPLER and RADAR for navigation. Determine if weather will permit use of FLIR for target identification and tracking. Descent speed is 420 kts.

8. Dash (Terrain Following)

Turn on laser designator and laser spot tracker; maintain speed at 420 kts.

9. Pop-up and Attack

Perform pop-up maneuver to 3,000 ft., 4 nm from target. Set HUD to bomb and select proper armament switches. Identify target with FLIR. Select desired attack mode. Make approach and release MK-84's, hitting the target.

10. Escape Maneuver

At release, initiate 45° bank right turn, hold until bomb impact, then pull into dive to descend to terrain following (TF) altitude; turn jammers to standby.

11. Terrain Following

Fly TF at 350 kts, using FLIR. Navigate toward home.

12. Climb to Cruise Altitude

Start climb to cruise altitude of 18,000; climb speed is 350 kts. Check in with GCI sight on UHF, and report mission results to combat operations center on HF/SSB.*

13. Cruise

Manage fuel and safety armament. Aircraft is passed from GCI to approach control. Pilot is cleared for TACAN/GCA approach and landing. Perform descent checklist. Cruise speed is 480 kts.

14. Penetration

Make TACAN penetration. Descent speed is 330 kts.

15. GCA Landing

Make GCA approach and landing. Landing speed is 140 kts. Dearm and complete after landing checklist. Debrief maintenance and intelligence personnel.

The approximate average distance, average velocity, time, change in altitude and rate of climb are given in Table 1 for each segment of the Close Air Support Night Attack Mission. These numbers give an indication of the desired aircraft average performance for each segment of the mission.

2.2 Mission Model

Each segment of the overall mission can be considered as a mission itself with its own performance index, reference functions (when specified), and terminal conditions. The pilot can fly each segment of the mission as a separate problem where the terminal conditions of one segment are the initial conditions of the next segment. The overall mission problem is considered to be the sequenced collection of problems from the segments. Table 2 lists the terminal conditions, possible reference functions, inequality constraints, and performance factors for each segment of the Close Air Support Night Attack Mission.

The performance factors are, in general, different for each segment of the mission, however, the segments can be classified into a few categories. The first is the climbing or descending type segment

*See Glossary, Appendix A

SEGMENT	TYPE	DIS (NM)	REF. VEL* (KNOTS)	TIME (MIN)	ΔALT (FT)	RATE OF CLIMB (FT/MIN)
1	Preflight and Takeoff	23.4	330	4.25	5,000	1,176
2	Accelerate and Climb	29.6	330	5.38	13,000	2,416
3	Rendezvous and Refuel	Navigates to air refueling initial point. Joins tanker in formation.				
4	Cruise	81.2	420	11.60	-0-	-0-
5	Evade Missile	Continuous cruise, but with a secondary task or may include an evasive maneuver.				
7	Descent	12.5	420	1.79	-17,000	-9,550
8	Dash	40.6	420	5.80	-0-	-0-
9	Pop-up and Attack	4.0	420	0.57	2,000	3,508
10	Escape Maneuver	3.1	420	0.44	-2,000	-4,545
11	Terrain Following	103.1	350	17.67	-0-	-0-
12	Climb to Cruise Alt.	9.3	350	1.59	17,000	10,691
13	Cruise	53.1	480	6.64	-0-	
14	Penetration	12.5	330	2.27	-13,000	-5,726
15	GCA Landing	21.8	330 (140)	3.96	-5,000	1,262
TOTAL		394.2		61.96		

*Reference velocity is the velocity given with the mission description, in some cases it is a reference for the segment and in other cases it is an objective for the segment.

TABLE 1 AVERAGE BEHAVIOR FOR EACH SEGMENT OF THE MISSION

SEGMENT	TYPE	TERMINAL CONDITIONS	POSSIBLE REFERENCE FUNCTIONS	INEQUALITY CONSTRAINTS	PERFORMANCE FACTORS
1	Preflight & Takeoff	5000' = Z 330 Kts = V $\psi = \psi_R$	$Y = 0$ $X = +[28.4]\ Z$	$Z \leq 0$ $\lvert X - AZ \rvert \leq N_1$ Takeoff cone	1. Tracking $X = AZ$ 2. Fuel
2	Accelerate & Climb	18,000' = Z 330 Kts = V $\psi = \psi_{R_2}$	$Y = 0$ $X = -[12.4]\ Z + 80017$	$Z \leq 0$ on α or γ on roll, pitch and yaw	1. Tracking $X = AZ + B$ 2. Fuel 3. Reaching terminal condition
3	Rendezvous & Refuel	18,000' = Z(t) $X = X(t)$ $Y = Y(t)$ $\gamma = 0$ $\phi = 0$			Pre-rendezvous maintaining Z ; Post rendezvous (during refueling) tracking error.
4	Cruise	Z = 18,000' $\psi = \psi_{R_4}$ V = 420 Kts	Z = 18,000' $Y = 0$ V = 420 Kts		1. $\dot{Z} = 0$, $\dot{V} = 0$, $\psi = 0$ 2. Minimize fuel 3. Terminal cond.
5	Evade Missile	Continues cruise, but with a secondary task or may include an evasive maneuver.			

A Glossary of symbols used in this Table is given in Appendix A.

TABLE 2 GENERAL CONSTRAINTS ON RATES AND ANGLES

SEGMENT	TYPE	TERMINAL CONDITIONS	POSSIBLE REFERENCE FUNCTIONS	INEQUALITY CONSTRAINTS	PERFORMANCE FACTORS
7	Descent	$\psi = \psi_{R_7}$ $V = 420$ Kts	$X = a_7 Z + b_7$ $Y = 0$	$Z \leq 0$ $\lvert \alpha \rvert \leq$ constant $\lvert V \rvert \leq$ constant	Terminal condition Track $X = a_7 Z + b_7$
8	Dash	$Z = Z_R, Y = Y_R$ $\psi = \psi_{R_8}$ $V = 420$ Kts	$Z = f(X_1\ Y) + E$ Terrain Tracking	$Z \geq Z$ Min $V \leq V$ Max	Minimum time
9	Pop-up & Attack	Climb $X = X(t_1)$ $Y = Y(t_1)$ Level Attack $\dot{Z}_R = 0$ $X = X(t_1)$ $Y = Y(t_1)$	 $X = X(t)$ $Y = Y(t)$ $X = X(t)$ $Y = Y(t)$	 $Z > Z$ Min	
10	Escape Maneuver	Predescribed path			

TABLE 2 GENERAL CONSTRAINTS ON RATES AND ANGLES (CONTD)

SEGMENT	TYPE	TERMINAL CONDITIONS	POSSIBLE REFERENCE FUNCTIONS	INEQUALITY CONSTRAINTS	PERFORMANCE FACTORS
11	Terrain Following	←	Same as 8	→	
12	Climb to Cruise Alt	←	Same as 2	→	
13	Cruise	←	Same as 4	→	
14	Penetration	←	Similar to 2	→	
15	GCA Landing	$Z = 0$ $V = 0$ $\psi = \psi_{R_{10}}$	Z = glide path	$Z \leq 0$ $V \geq V_R$ $Y = Y_R \pm K$	$\dot{Z} = \dot{Z}_{R_{10}}$ $\dot{V} = 0$

TABLE 2 GENERAL CONSTRAINTS ON RATES AND ANGLES (CONCLUDED)

where the objective is to reach some terminal altitude, or velocity and altitude, while the performance is evaluated by comparing actual trajectories with reference trajectories. A second type of segment is the cruising segment. In this type of segment the performance is evaluated based on how well the pilot maintains specified altitude, heading and velocity while reaching the terminal position. A third type of segment is a prescribed maneuver, which could include escape maneuvers, refueling maneuvers, and possibly pop-up and attack maneuvers. Finally, a fourth type of segment is terrain following where desired performance requires completing the segment in minimum time and maintaining a minimal distance above the ground.

Differences between segments of the same type would occur in the initial conditions, terminal conditions, and possibly the inclusion of inequality constraints. Although each typical segment of the mission appears to be quite different, a generalized performance index common to many segments can be used. Equation 2.1 is the proposed generalized performance index.

$$J = [X_R(t_f) - X(t_f)]^T S [X_R(t_f) - X(t_f)] + \int_{t_o}^{t_f} \Big\{ (X_R(t) - X(t))^T Q (X_R(t) - X(t)) + (U_R(t) - U(t))^T R (U_R(t) - U(t)) + (\dot{X}(t))^T W (\dot{X}(t)) \Big\} dt \tag{2.1}$$

This is a measure of performance over a segment, where t_f is the final time, X is the vector of state variables of the system, U is a vector of control variables of the system, and S, W, Q, and R are weighting matrices which can be selected by the method described in the introduction. $X_R(t)$ is the reference state and $U_R(t)$ is the control in that reference state. The generalized performance index is made up of a term that weights the difference of the state variables from a specified reference state, penalizes excessive control, and penalizes high rates of change of the state variables, effectively suggesting "smooth" transitions in changing states. Another term might also be added to penalize "jerky" control actions, but since the optimal feedback control law will define the control rule as a function of states, the penalty (W) on state-change rates ($\dot{X}$) serves virtually the same purpose.

The difference between the performance index for a cruise segment and a climbing segment could be reflected in differences in the weighting matrices, reference functions and terminal conditions. Essentially, the performance index is chosen such that the resulting optimal paths (solution trajectories) represent the desired flight path. If a flight path is specified as virtually mandatory (very high penalty weights on errors or excursions from that path), then that reference path is "optimal." Thus the optimal control forces the desired path to be reference path. The distinction between "desired" and "reference" paths arises from not being able to maintain the reference path due to other perturbing factors (e.g. turbulence). Then the optimal path may not be identical with the reference path once the segment has begun (i.e., $t > t_o$).

3.0 MATHEMATICAL MODEL OF AIRCRAFT DYNAMICS

General mathematical models of the dynamics of an aircraft moving through the atmosphere have been developed for use in simulations and design of aircraft control systems (e.g., Etkin, 1972 and Fogarty and Howe, 1969). For purposes of this analysis, the aircraft is modeled by the simplified equations for aircraft dynamics contained in the subroutine ADCOMP from the ALL DIGITAL COCKPIT DISPLAY SYSTEM PROGRAMS obtained from the Human Engineering Division, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio. This aircraft model was derived as a first attempt to provide a fairly reasonable and realistic task to relatively naive subjects. For non-flyers it is a demanding task and believable. For flyers and knowledgeable engineers, the simulation is anything but real and represents an expedient compromise to obtain a workable set up for controlled, laboratory experiments. Since the current effort was focused on developing a new technique for scoring performance, the CPM methodology has been developed for this artificial aircraft: the ADCOMP subroutine. The rationale for this decision is that once developed, initial experience with CPM can be gained as real-time laboratory experiments are conducted using the ADCOMP driven simulations. For other simulations or for real aircraft, a more elaborate model would have to be defined (and parameter values determined) specific to that application. Since ADCOMP was not documented by its developer, an attempt is made here to rationalize the given design of the aircraft model.

3.1 Assumptions

For the model of the aircraft incorporated in the equations of the given ADCOMP subroutine, the following assumptions apply.

- The aircraft is traveling at a speed less than MACH 3,
- The thrust vector is aligned with the fuselage reference line.
- The vehicle is a rigid body having a plane of symmetry, i.e., the right side of the aircraft is configured the same as the left (i.e., same size, weight and shape of components and attachments - fuel pods, weapons, etc.)*,

*While this is true in many cases, it may not be true for some segments where a weapon releases from one wing but none from the other.

- The atmosphere is at rest relative to the earth, i.e., the wind is zero,
- The earth is considered a plane fixed in space, i.e., a flat earth,
- The reference axes are a north, east and down system fixed to the earth,
- The side-slip angle is neglected, i.e., assumed to be zero,
- The aircraft is assumed to be a point mass, in that moments of inertia are ignored,
- The rate of change of roll angle is approximately proportional to stick position,
- The equations describing angular acceleration are neglected,
- The rate of change of the angle of attack is approximated as being proportional to stick position plus a term due to lift,
- The reference frame for the aircraft is a combined wind and body axes system, and
- The rudder is automatically set to give coordinated turns.

Under the assumptions given above, the very complicated set of equations for aircraft dynamics given on Pages 149-150 of Anderson and Moore (1971) reduce down to the simplified set of equations used in the ADCOMP subroutine. Generally, these assumptions are valid for the mission segments considered since the aircraft flights are over short distances and at relatively low speeds.

Figures 4 and 5 show the notation for the angles, forces and associated reference frames for the aircraft.

L =	Lift	α =	Angle of attack	ψ =	Heading angle
D =	Drag	γ =	Flight path angle	ϕ =	Roll angle
V =	Velocity	m =	Mass of the aircraft	g =	Gravitational constant (32 ft./sec.^{2})
MT =	Maximum thrust	w =	Weight of the aircraft		

$$\left.\begin{matrix} X_e \\ Y_e \\ -Z_e \end{matrix}\right\} = \text{Earth fixed coordinates}$$

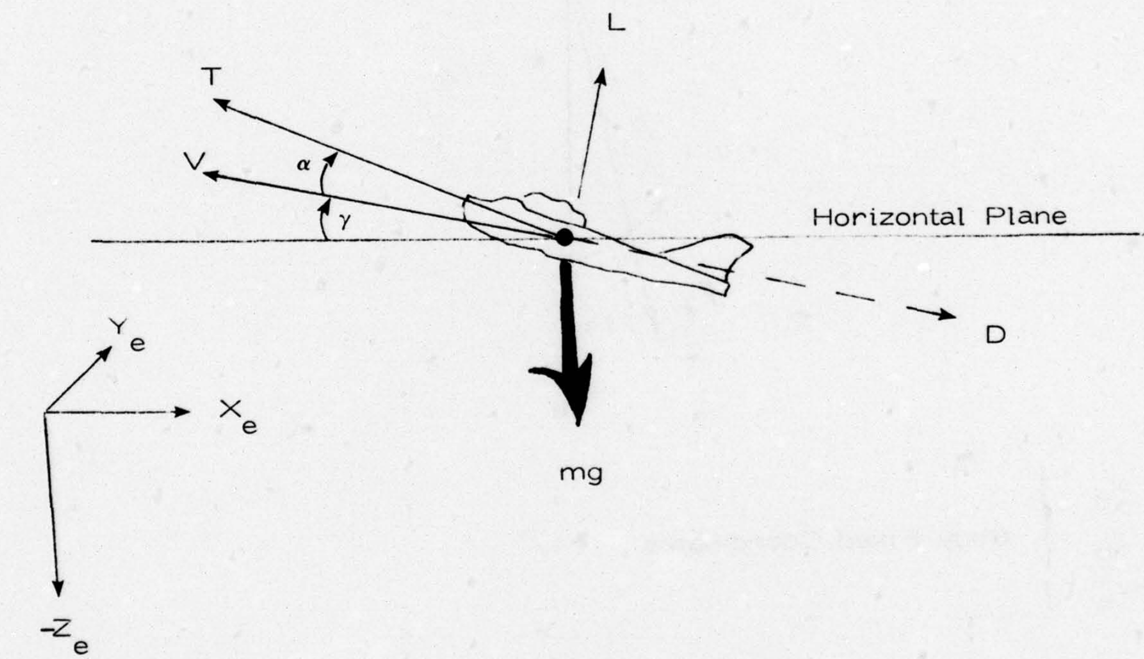
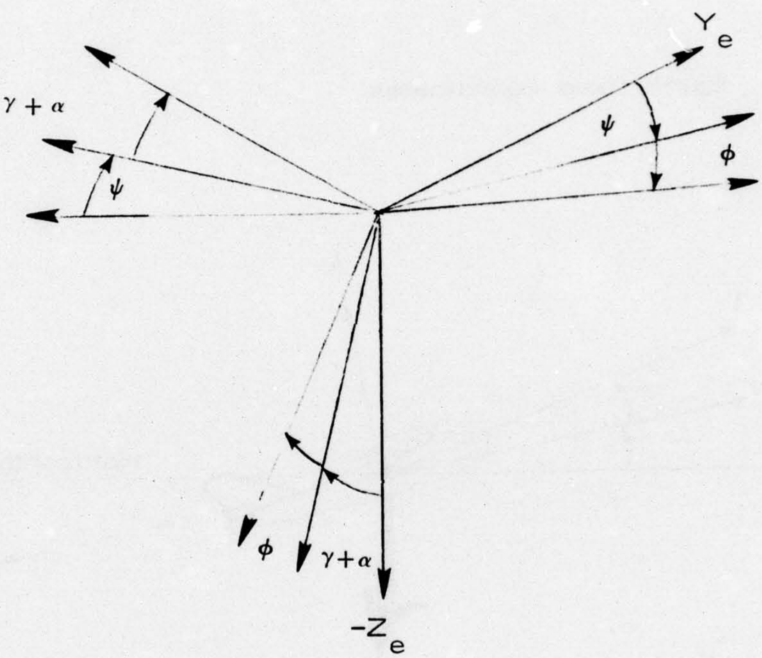


FIGURE 4 REFERENCE FRAMES



$\left.\begin{array}{l} x_b \\ y_b \\ -z_b \end{array}\right\}$ Body Fixed Coordinates

FIGURE 5 CONVENTIONAL AIRCRAFT EULER ANGLES

3.2 The Aircraft Model

The resulting set of simplified equations for the aircraft dynamics are:

$$\dot{X}_e = V \cos \gamma \cos \psi \tag{3.1}$$

$$\dot{Y}_e = V \cos \gamma \sin \psi \tag{3.2}$$

$$\dot{Z}_e = V \sin \gamma \tag{3.3}$$

$$\dot{\phi} = \mu_2 \tag{3.4}$$

$$\dot{\psi} = \frac{L \sin \phi}{m V \cos \gamma} \tag{3.5}$$

$$\dot{\gamma} = \frac{L \cos \phi - W \cos \gamma}{m V} \tag{3.6}$$

$$\dot{\alpha} = \mu_1 + (L/W - 1)\ (AL1) \tag{3.7}$$

$$\dot{V} = \frac{\mu_3\ (MT) \cos \alpha - D - W \sin \gamma}{m} \tag{3.8}$$

where μ_1 is functionally related to the pilot's pitch input (fore-aft stick movement)

μ_2 is functionally related to the pilot's roll input (side-to-side stick movement)

and μ_3 is functionally related to the pilot's throttle settings.

The parameter AL1 is a constant taking on the values:

$$AL1 = \begin{cases} \dfrac{-K}{MAXG} \quad , \text{ for } \dfrac{L}{W} \geq 1 \\ \\ \dfrac{+K}{MING} \quad , \text{ for } \dfrac{L}{W} < 1 \end{cases} \tag{3.9}$$

where K is a scaling parameter which is used to specify the aircraft pitching rates with respect to lift to weight ratio. In ADCOMP, K was an input that was read into the computer when the real-time simulation was executed. Here K was assumed to be 1.0 for convenience. Also, the values of MAXG and MING are artificial constraints. Since there is no motion in the ADCOMP simulation, subjects can "pull" unrealistically high "G" levels. This equation prevents subjects' unreasonable inputs from "blowing up" the ADCOMP simulation. While MAXG refers to the *largest allowable positive* acceleration, MING refers to the largest allowable negative acceleration. Chosen properly, MAXG (+15) and MING (-5) prevent the execution of "impossible" turns, dives, and climbs. Even so, the values used in ADCOMP are quite large, which is one of its unrealistic features. The values can be readily changed, however. The maximum thrust is given by:

$$MT = \Big[A_{MT} \quad (2327 + 0.172\, Z_e - 0.0000031\, Z_e^2) \ \frac{V}{SS} + 11500 - 0.25\, Z_e \Big] \tag{3.10}$$

where

$$A_{MT} = \begin{cases} 2 \text{ , if the afterburner is on} \\ \\ \frac{1}{2} \text{ , if the afterburner if off} \end{cases} \tag{3.11}$$

and SS is the speed of sound which is a function of altitude.

The ADCOMP subroutine uses an aircraft weight (W) of 17,000 lbs or a mass (M) of 528 slugs. The drag force (D) is a function of the altitude, velocity and angle of attack. The lift force (L) is a function of angle of attack, altitude, thrust, and velocity. Appendix A contains defining equations for the drag and lift forces.

The control variables (or inputs) for the model are μ_1 (t) which controls the rate of change of the angle of attack and is proportional to longitudinal stick position, μ_2 (t) which controls the rate of change of the roll angle and is proportional to lateral stick position, and μ_3 (t) which is normalized percent throttle. In practice these inputs come from the pilot or from an autopilot.

The dynamic variables for the model are:

$\alpha(t)$ = Angle of attack (radians)

$V(t)$ = Air speed of aircraft (feet/second)

$X_e(t)$ = X-position of aircraft with respect to earth (feet)

$Y_e(t)$ = Y-position of aircraft with respect to earth (feet)

$Z_e(t)$ = Altitude of aircraft (feet)

$\phi(t)$ = Roll angle (radians)

$\psi(t)$ = Heading angle (radians)

$\gamma(t)$ = Flight path angle (radians)

Appendix A contains a summary of these equations.

For various segments of the mission these equations can be further simplified. For example, for a cruising segment if a constant altitude is assumed, many of the coefficients that change with altitude can be approximated by a constant over a small range of altitude change. The equations can then be reduced to a simpler form for a cruising problem.

3.3 State Space Formulation of Aircraft Equations

The set of differential equations describing the aircraft dynamics given in Section 3.2 can be put into the form of vector/matrix differential equations if state variable notation is used (Padulo and Arbib, 1974). This notation simplifies the description of the optimal control and CPM developments.

The differential equations describing the aircraft dynamics can, in general, be written in vector/matrix form by the state equation:

$$\underline{\dot{X}}(t) = F(\underline{X})\,\underline{X}(t) + G(\underline{X})\,\underline{\mu}(t) \tag{3.12}$$

where $\mu(t)$ is the three-dimensional control vector given by:

$$\mu(t) = \begin{bmatrix} \mu_1(t) \\ \mu_2(t) \\ \mu_3(t) \end{bmatrix} \tag{3.13}$$

and X(t) is the system state vector. Wernli and Cook (1975) contains a discussion of the "apparent linearization" technique that rationalizes equation 3.12. The choice of which of the aircraft dynamic variables to include in the state vector may vary depending on the segment of the mission. For example, in a cruising segment the X-position of the aircraft with respect to earth, X_e, should not be included in the state vector $\underline{X}$, since steady state cruising conditions do not depend on X_e. This point is discussed further in Section 5. $F(\underline{X})$ is the system matrix whose elements are a function of the state variables and $G(\underline{X})$ is the control matrix whose elements are also a function of the state variables. Since $F(\underline{X})$ and $G(\underline{X})$ are not unique but change from problem to problem (i.e., system A requires a different model than system B, or mission phase I for system A requires a different model than phase II, etc.), the generalized forms apply to specific cases only when the numbers appropriate to the system/problem at hand have been defined and entered into the matrices. However, even in this general form, the matrix equations can often be "solved" for the general case so that a specific solution is immediately available as soon as the numeric values for the matrix entries become available. When this is possible, the technique is indeed powerful, since for those cases where the problem of interest "fits" one of the general cases already formulated and solved, then the answers for the problem of interest are obtained relatively easily. However, this is most often possible in cases where the linearized models reasonably define the steady state behavior of the system. The theory of linear systems is then applicable and provides a well developed set of solution techniques many of which make use of the matrix or linear algebra.

For nonlinear cases, the same conceptual scheme is taken for setting up the state variable equations in cannonical (i.e., a predefined "standard," and typically "simple") form, but the matrix algebra does not apply, the solutions are typically not known in advance, and the techniques for solving the equations may not be readily available. In these cases it is often necessary to use a simulation and recursive or iterative solution techniques to arrive at the definition of an optimal control rule and the associated optimal trajectory. Because of these often formidable difficulties, modelers often choose to study a linearized model first and develop nonlinear representations only after they have exhausted the insights to be gained from the simpler linearized model.

The choice of the $F(\underline{X})$ and $G(\underline{X})$ matrix is not unique, but can frequently be put into the form given by Equation 3.12. In certain applications it may be of advantage to pick one form of $F(\underline{X})$ over another (Wernli and Cook, 1975).

The aircraft state equation given by Equation 3.12 together with the performance index given in Section 2.2 by Equation 2.1 define an optimal control problem. The solution to this optimal control problem is the first step in finding the CPM.

3.4 Simplified State Equation for Constant Altitude Cruising (Mission Segment 4)

The aircraft equations described in the previous sections can be further simplified whenever the aircraft is flown at near constant altitude. Segment 4 provides a constant altitude reference flight path, and it is assumed the aircraft will be near that reference altitude. As a result, aircraft coefficients which are functions of altitude can be replaced by a constant value for Segment 4 analysis. The assumptions for the simplified equation which are used in Section 5 of this report to find an optimal control for Segment 4 are as follows:

- The altitude Z_e is approximately 18,000 feet over the entire segment
- The after-burner is off during the entire segment

Then for Z_e = 18,000 feet, the following variables were assigned specific values as designated below:

Air Temperature	=	-5.26^{o} F
D_1	=	0.87604 (dimensionless)
Air Pressure	=	1055.4212 lbs/sq ft
Air Density	=	1.3539×10^{-3} slugs/ft^3
Speed of Sound	=	1046.9171 ft/sec.

The mach of the aircraft is

$$M_1 = (9.5519 \times 10^{-4})\ V$$

The dynamic pressure is

$$Q = (6.7695 \times 10^{-4})\ V^2,$$

Dynamic pressure affects the calculation of lift and drag forces as described in Appendix A.

The maximum thrust (with after-burner off) is

$$MT = 2.1103V + 3500,$$

and thrust is

$$T = [2.1103\ V + 3500.]\ \mu_3.$$

The coefficient of lift is

$$CL = CL1 + (CL2)\ \alpha$$

where

$$CL1 = \begin{cases} 0 & \text{if } \alpha \geq 0.6 \\ 1.8 & \text{if } 0.4 \leq \alpha < 0.6 \\ 0.1 & \text{if } 0.4 > \alpha \end{cases}$$

and

$$CL2 = \begin{cases} 0 & \text{if } \alpha \geq 0.6 \\ -2.0 & \text{if } 0.4 \leq \alpha < 0.6 \\ +2.5 & \text{if } 0.4 > \alpha \end{cases}$$

where α is the angle of attack in radians. The implied restrictions are a feature of the ADCOMP subroutine as originally designed. As with the choice of values for MING and MAXG, it is not clear why the designer of ADCOMP chose these exact limits, but it appears the intent was to preclude grossly unrealistic inputs to the model even if naive subjects inadvertently induced such inputs. The coefficient of drag is given by

$$CD = 0.0327 + 0.135\alpha + 1.6875\ \alpha^2$$

The drag then is given by:

$$D = (0.1367)\ V^2\ (0.0327 + 0.135\alpha + 1.6875\,\alpha^2)$$

These equations for drag (D) and the coefficient of drag (CD) are also only applicable for cases where the angle of attack (α) is less than 0.4, which is an acceptable assumption at least for the cruise segment of the mission.

The component of applied force normal to the flight path is:

$$L = [0.13673\ (CL1)]\ V^2 + [0.13673\ (CL2)]\ \alpha V^2 + [2.1103\ V + 3500.]\ \sin(\alpha\mu_3)$$

Assuming, as stated previously, that K = 1, then the parameter AL1 takes the values:*

$$AL1 = \begin{cases} -0.0666, & \text{for } \dfrac{L}{17,000} \geq 1 \\ -0.0400, & \text{for } \dfrac{L}{17,000} < 1 \end{cases}$$

Using the above approximations the aircraft dynamic equations, become for the cruising segment:

$$\dot{X}_e = V \cos\gamma \cos\psi \tag{3.14}$$

$$\dot{Y}_e = V \cos\gamma \sin\psi \tag{3.15}$$

$$\dot{Z}_e = V \sin\gamma \tag{3.16}$$

$$\dot{\phi} = \mu_2 \tag{3.17}$$

*The calculation of these values is more fully described in Appendix A.

$$
\begin{aligned}
\dot{\psi} \quad &= \quad (2.589 \times 10^{-4})\ CL1\ \frac{\sin\phi}{\cos\gamma}\ V \\
&\quad + (2.589 \times 10^{-4})\ CL2\ \frac{\sin\phi}{\cos\gamma}\ \alpha\ V \\
&\quad + \left[(3.997 \times 10^{-3}) + \frac{6.629}{V}\right] \frac{\sin\phi \sin\alpha}{\cos\gamma}\ \mu_3 \qquad (3.18)
\end{aligned}
$$

$$
\begin{aligned}
\dot{\gamma} \quad &= \quad \left[(3.997 \times 10^{-3}) + \frac{6.629}{V}\right] \cos\phi\ \sin(\alpha\mu_3) \\
&\quad - 32.2\ \frac{\cos\gamma}{V} + (2.589 \times 10^{-4})\ CL2\ \cos\phi\ \alpha\ V \\
&\quad + (2.589 \times 10^{-4})\ CL1\ \cos\phi\ V \qquad (3.19)
\end{aligned}
$$

$$
\begin{aligned}
\dot{\alpha} \quad &= \quad \dot{\mu}_1 - AL1 + AL1\ \left[(1.241 \times 10^{-4})\ V + 0.206\right] \sin(\alpha\mu_3) \\
&\quad + \left[(8.041 \times 10^{-6})\ AL1\right]\left[CL1\ (V^2)\right] \\
&\quad + \left[(8.041 \times 10^{-6})\ AL1\right]\ CL2\ (\alpha V^2)] \qquad (3.20)
\end{aligned}
$$

$$
\begin{aligned}
\dot{V} \quad &= \quad \left[(3.997 \times 10^{-3})\ V + 6.629\right] \cos(\alpha\mu_3) \\
&\quad - 32.2 \sin\gamma - (8.466 \times 10^{-6})\ V^2 \\
&\quad - (3.495 \times 10^{-5})\ \alpha V^2 - (4.369 \times 10^{-4})\ \alpha^2\ V^2 \qquad (3.21)
\end{aligned}
$$

Again the reader is cautioned that the above equations apply for the restricted values of (α) that govern the computation of values for CL1 and CL2. These equations are put into state variable notation in Section 5.

4.0 A CONTINUOUS PERFORMANCE MEASURE FOR MAN-MACHINE SYSTEMS

A continuous performance measure (CPM) for aircraft flight control systems is developed in this section. A CPM is developed by applying optimal control theory to the manual control problem in order to establish the required flight reference (criteria) and significance of deviation-from-criteria information. To illustrate the concepts and techniques used, an example problem is presented.

4.1 Performance Measurement Requirements

A motivation and rationale for developing a CPM is presented in the introduction. The desired properties of the CPM are summarized as follows:

1. The measure should allow comparison of present performance with respect to preferred performance where the preferred performance is defined by the system's motion in state space (trajectories) under the optimal control law.

2. The optimum control against which performance is being evaluated should be determined in terms of a system performance (cost) index which in turn is selected by examination of the associated optimal state space trajectories.

3. The measure should allow instantaneous (state related) performance measurement, as well as, average performance measurement over an arbitrary time interval within the task.

4. The measure should allow determination of critical regions in the system state space. Critical regions refer to regions that are particularly sensitive to accurate operator control. The performance measurement must allow both theoretical determination and experimental determination of high cost sensitivity regions in state space.

The first step in developing the CPM for manual control systems is to formulate mathematically the objective of the control task as a performance index J. The performance index J is a summary measure for the task. Examples of some types of J where this might be appropriate include cases where performance is defined in terms of a penalty or "cost" function: minimize the time to complete the task, minimize the fuel expended, minimize the error, etc.

The differential equations describing the system to be controlled together with J constitute an optimal control problem. If a solution to the optimal control problem exists and can be found in the form of the optimal feedback control law, then the control to be applied to the system in any state is determined such that J will have a minimum value. For example, if J represents the time to complete the control task, then the solution to the related optimal control problem will yield a feedback control law which results in the task being completed in the minimum time possible.

Any non-optimal control applied to the system by the operator will result in a larger value for J. The CPM developed in this section is based on the instantaneous effect of a non-optimal control applied to the system at any time during the task. This is done by comparing the effect of the non-optimal control on J as opposed to the effect if an optimal control had been applied to the system. By doing this, a CPM is developed which gives an instantaneous measure of the operator's performance as compared with optimal or "best possible" performance.

Again the reader is cautioned that "optimum" is defined (or influenced) by the terms one places in the performance index and the weights used in the scoring matrices. While these may reflect objective quantities (fuel, time, etc.) or the engineer's judgment (large penalty weights placed on altitudes "close to the ground"), it was also proposed that these could be subjective weights if one wished, thereby reflecting a single pilot's a priori goals, or a training instructor's criteria, or an operating command's policy. Consequently, the comparison the CPM makes to the "best possible" performance is always relative to the nature of the goals one explicitly puts into the performance index. The issue of which goals are in some sense "best" is yet another issue, and one that is beyond the scope of this discussion. Here it is assumed the goals have been appropriately chosen and accurately reflected in the performance index.

If the operator is using the optimal control, then the value of the CPM is zero. If the operator uses non-optimal control, the CPM is positive and its value is equal to the significance of the control error. This sensitivity property of the CPM is demonstrated in the following sections.

4.2 <u>Related Optimal Control Problem</u>

The system to be controlled is described by a vector/matrix differential equation of the general form:

$$\dot{X}(t) = f\left[X(t),\ U(t)\right] \tag{4.1}$$

where $X(t)$ is the vector of state variables, $U(t)$ is the vector of control variables, and f is some function of X and U. It is assumed that the objectives of a segment of the mission can be analytically expressed as the minimization of a scalar performance index of the general form:

$$J\left[X(t),\ U(t)\right] = \int_{t_o}^{t_f} E\left[X(t),\ U(t)\right] dt \tag{4.2}$$

where t_o is the initial time and t_f is the final time of the problem. E is a positive definite function, that is, $E\left[X(t),\ U(t)\right] > 0$ for all values of $X \neq 0$ and $U \neq 0$. Examples of positive definite functions that arise in some typical control problems are:

1. E is a quadratic function of X and U, that is

$$E(X,U) = X^T(t)\ Q\ X(t) + U^T(t)\ R\ U(t) \tag{4.3}$$

where R is a positive definite matrix and Q is a non-negative definite matrix*

2. minimum time problems, where

$$E(X,U) = 1 \tag{4.4}$$

*The differential, integral, and matrix calculus are used in the remainder of this chapter and the reader is assumed to have a reading knowledge of the notation (see Padulo and Arbib (1975), Sage (1968), Bryson and Ho (1976), Athans and Falb (1966) or Anderson and Moore (1971) for engineering application and Pipes (1963), Bellman (1970) or Gantmacher (1959) for mathematical development).

The related optimal control problem is to find a feedback control law which transfers the system of Equation 4.1 from any initial state to a given terminal condition and which minimizes the performance index of Equation 4.2. Assuming a solution exists and can be found analytically, the optimal feedback control law can be written in functional form as:

$$U^*(t) = \beta\left[X(t),\ t\right] \tag{4.5}$$

Equation 4.5 specifies the optimal control to be applied for every state of the system. That is, equation 4.5 asserts that the optimal control ($U^*(t)$) is some as yet unspecified but derivable function (β), and the only required inputs or arguments to that function are the states of the system at time (t), the $X(t)$, and the actual or current time (t). Using this optimal control will result in the minimum value for J, which will be denoted as J^*. This is, by definition, the best performance possible for this segment of the mission.

4.3 "Cost-to-Go" Function

Consider the performance index of Equation 4.2, but with the integral evaluated in two parts. This can be written as the sum

$$J = \int_{t_o}^{t_1} E\left[X(t),\ U(t)\right]dt + \int_{t_1}^{t_f} E\left[X(t),\ U(t)\right]dt \tag{4.6}$$

The first part of the sum is the integral evaluated between the initial time, t_o, and an intermediate time t_1, where $t_o \leq t_1 \leq t_f$. Call this the "cost accumulated" at time t_1. The second part of the sum is the integral evaluated between time t_1 and the final time t_f. By definition, this integral is called the "cost-to-go" at time t_1 and is denoted by the symbol θ, i.e.,

$$\theta\left[X(t_1),\ U(t)\right] = \int_{t_1}^{t_f} E\left[X(t),\ U(t)\right]dt \tag{4.7}$$

Using the optimal feedback control law, we can replace $U(t)$ with $U^*(t)$ to define the optimal cost to go $(\theta^*(X(t_1), U^*(t)))$, but by Equation 4.5, we can also replace $U^*(t)$ with $\beta\left[X(t), t\right]$, so Equation 4.7 finally becomes

$$\theta^*\left[X(t_1)\right] = \int_{t_1}^{t_f} E\left[X(t), \beta(X(t), t)\right] dt \tag{4.8}$$

Notice that the optimal "cost-to-go" is the value of the integral when the optimal feedback control law is the control the operator decides to use from time t_1 to t_f and, therefore, <u>θ^* depends only on the state</u>. It follows from Equation 4.8 that the "cost-to-go" evaluated at $t_1 = t_f$ is

$$\theta^*\left[X(t_f)\right] = 0 \tag{4.9}$$

and if evaluated at $t_1 = t_o$ then

$$\theta^*\left[X(t_o)\right] = J^*(t_f). \tag{4.10}$$

Note also that from Equation 4.7

$$\left.\frac{d\theta}{dt}\right|_{t = t_1} = -E\left[X(t_1), U(t_1)\right] \tag{4.11}$$

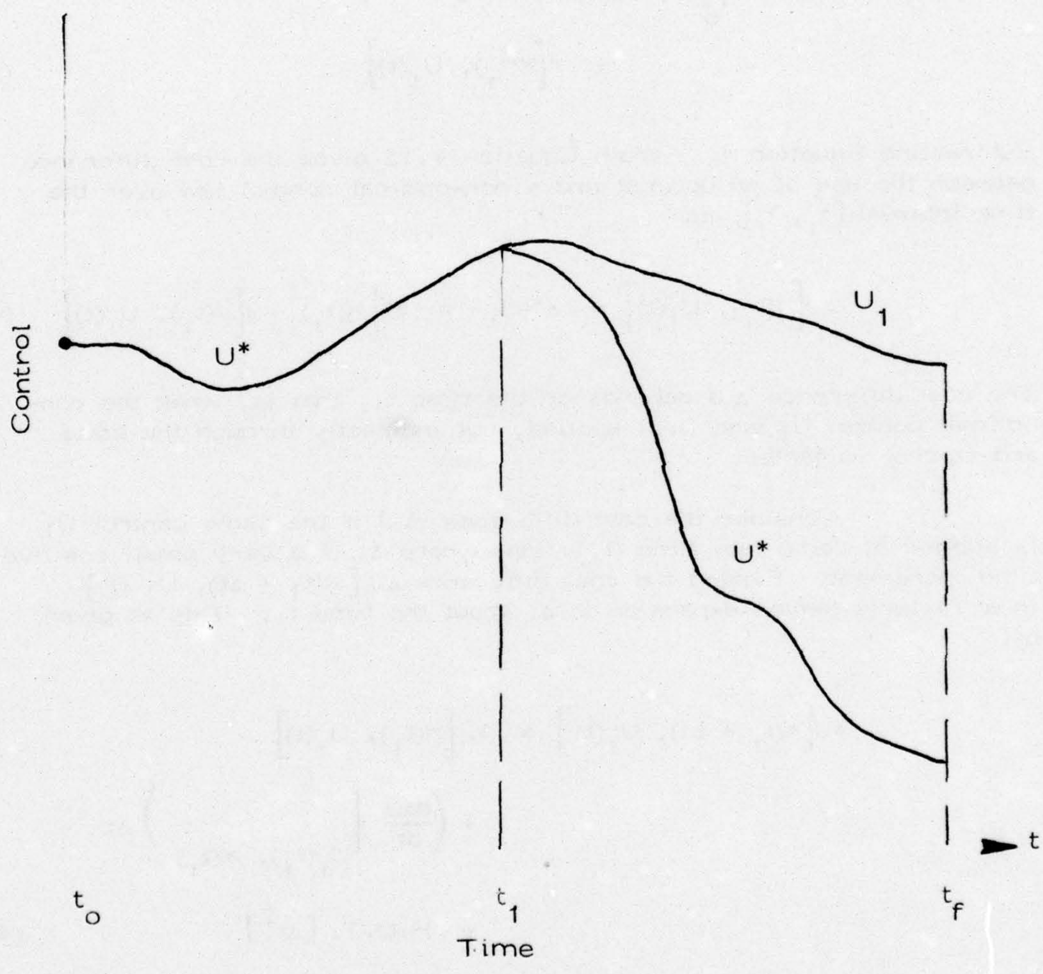
4.4 Continuous Performance Measure

Consider the effect of two different control laws on the performance index by comparing trajectories and cost values for two solutions with different control laws but the same initial conditions. Assume the first control law used is the optimal feedback control, while the second control law is such that between time t_o and t_1 the optimal control $U^*(t)$ is applied, but between time t_1 and t_f a non-optimal control $U_1(t)$ is applied. This is shown in Figure 6. The value of the performance index using the first control, i.e., the optimal control U^* is

FIGURE 6

TWO DIFFERENT CONTROL POLICIES

ONE OPTIMAL, ONE NON-OPTIMAL



$$J^* = \int_{t_o}^{t_f} E\left[X(t),\ U^*(t)\right]dt + \theta^*\left[X(t_1)\right] \tag{4.12}$$

The value of the performance index using the second control law is

$$J_1 = \int_{t_o}^{t_1} E\left[X(t),\ U^*(t)\right]dt + \theta\left[X(t_1),\ U_1(t)\right] \tag{4.13}$$

Subtracting Equation 4.13 from Equation 4.12 gives the cost difference between the use of an optimal and a non-optimal control law over the time interval $[t_1,\ t_f]$, as

$$\Delta J\left[X(t_1),\ U_1(t)\right] = J^* - J_1 = \theta^*\left[X(t_1)\right] - \theta\left[X(t_1),\ U_1(t)\right] \tag{4.14}$$

The cost difference ΔJ depends on the time t_1, that is, *when the non-optimal* control U_1 was first applied, but indirectly through the state and control variables.

Consider the cost difference ΔJ if the same control U_1 is applied at some late time $(t_1 + \Delta t)$, where Δt is a very small positive time increment. Expand the cost difference $\Delta J\left[X(t_1 + \Delta t),\ U_1\ (t)\right]$ in a Taylor's series expansion in Δt about the time t_1. This is given by:

$$\begin{aligned} \Delta J\left[X(t_1 + \Delta t),\ U_1(t)\right] &= \Delta J\left[X(t_1),\ U_1(t)\right] \\ &+ \left(\frac{d\Delta J}{dt}\Bigg|_{U_1(t_1),\ X(t_1)}\right)\Delta t \\ &+ \text{H.O.T.}\left[\Delta t^2\right] \end{aligned} \tag{4.15}$$

The last term on the righthand side of Equation 4.15 represents higher order terms (H.O.T.) that depend on (Δt^2) or larger powers. Rearranging

Equation 2.15 and dividing by Δt, the incremental cost difference is given by:

$$\frac{\Delta J\left[X(t_1 + \Delta t),\ U_1(t)\right] - \Delta J\left[X(t_1),\ U_1(t)\right]}{\Delta t}$$

$$= \left.\frac{d\Delta J}{dt}\right|_{U(t_1),\ X(t_1)} + \frac{H.O.T.\ (\Delta t^2)}{\Delta t} \tag{4.16}$$

Equation 4.16 represents the incremental cost difference between the use of the optimal control U^* and the non-optimal control U_1 over the time interval t_1 to $(t_1 + \Delta t)$. In other words, the incremental cost difference is the increase in the value of the performance index due to the use of the non-optimal control U_1 during the time interval t_1 to $(t_1 + \Delta t)$.

Define the Continuous Performance Measure (CPM at time t_1) as:

$$\phi\left[X(t_1),\ U_1(t)\right] = \lim_{\Delta t \to o} \frac{\Delta J\left[X(t_1 + \Delta t),\ U_1(t)\right] - \Delta J\left[X(t_1),\ U_1(t)\right]}{\Delta t} \tag{4.17}$$

Making use of Equations 4.11, 4.14, and 4.16 in Equation 4.17, the CPM evaluated at time t_1 using control $U_1(t_1)$ is given by:

$$\phi\left[X(t_1),\ U_1(t_1)\right] = \left.\frac{d\phi^*\left[X(t)\right]}{dt}\right|_{U_1(t_1),\ X(t_1)}$$

$$- \left.\frac{d\phi}{dt}\right|_{U_1(t_1),\ X(t_1)} = \left.\frac{d\phi^*\left[X(t)\right]}{dt}\right|_{U_1(t_1),\ X(t_1)}$$

$$+ E\left[X(t_1),\ U_1(t_1)\right] \tag{4.18}$$

The continuous performance measure ϕ of Equation 4.18 can be interpreted as the instantaneous increase in the value of the performance index due to the use of the non-optimal control U_1 at time t_1. If one now considers any time t between t_o and t_f, i.e., $t_o \leq t \leq t_f$, and any admissible control U(t), the CPM of Equation 4.18 generalizes to

$$\phi\left[X(t),\ U(t)\right] = \left.\frac{d\phi^*\left[X(t)\right]}{dt}\right|_{U(t),\ X(t)} + E\left[X(t),\ U(t)\right] \tag{4.19}$$

Equation 4.19 can be evaluated at each point in time to yield a continuous metric of performance which only depends on the present state and present value of control.

4.5 Properties of CPM

In this section, several properties of the CPM are presented.

1. The CPM, $\phi\left[X(t),\ U(t)\right]$ is zero when evaluated using the optimal control law of Equation 4.5 i.e.,

$$\phi\left[X(t),\ U^*(t)\right] = 0, \text{ for } t_o \leq t \leq t_f \tag{4.20}$$

2. Using the optimal feedback control law, Equation 4.19 is

$$\left.\frac{d\phi^*\left[X(t)\right]}{dt}\right|_{U^*,X} + E\left[X(t),\ U^*(t)\right] = 0 \tag{4.21}$$

However, if the "cost-to-go" in Equation 4.21 is only a function of the state variables, then

$$\left.\frac{d\theta^*[X(t)]}{dt}\right|_{U^*,X} = \left.\left[\frac{\partial\theta^*[X(t)]}{\partial X}\right]^T \dot{X}(t)\right|_{U^*,X}$$

$$= \left(\frac{\partial\theta^*[X(t)]}{\partial X}\right)^T f\left[X(t),\ \beta\,(X(t))\right] \tag{4.22}$$

The last substitution is based upon Equations 4.1 and 4.5. Using Equation 4.22 in Equation 4.21, yields the partial differential equation whose solution is the optimal "cost-to-go" function as:

$$\left(\frac{\partial\theta^*}{\partial X}\right)^T f\left[X(t),\ \beta\,(X(t))\right] + E\left[X(t),\ \beta\,(X(t))\right] = 0 \tag{4.23}$$

where $\beta\left[X(t)\right]$ is the optimal feedback control law.

3. From the definition of θ and the assumption that $E[X,\ U] > 0$, this implies

   $\theta^*[X(t)] \geq 0.$

4. Property 1 above implies that

   $\phi[X(t),\ U(t)] \geq 0$ if $U \neq U^*$.

5. The integral of the CPM over the problem time interval, $[t_o,\ t_f]$ is given by

   $$\int_{t_o}^{t_f} \phi\left[X(t),\ U(t)\right] dt = J(U) - J^*(U^*)$$

   which is the difference between the performance index evaluated using the operators control and that obtained using the optimal control law.

If the performance index reflects a model of the operator's goal aspirations, then $J(U) - J^*(U^*)$ implies the degree of dissatisfaction which may be experienced when performance falls short of the operator's objectives. If instead the performance index is based upon an instructor's criteria, then $J(U) - J^*(U^*)$ reflects the operator's earned score for less than perfect performance. If some "ideal" resource expenditure is reflected in the performance index, then $J(U) - J^*(U^*)$ reflects the wastefulness of non-optimal or sub-optimal control rules or policies and the behavior guided by these. So again, a specific interpretation depends upon the a priori specification of the objectives captured in the performance index.

4.6 Application of the Continuous Performance Measure

Generation, use, and interpretation of the continuous performance measure $\phi\,(X,U)$ involves the following steps (the system equation, performance index and control constants are assumed to be given):

1. Obtain an analytic formulation of the optimal feedback control function $U^*\,[X(t)]$ and the "cost-to-go" function $\theta^*[X(t)]$.

2. Form the performance measure $\phi\,(X(t), U(t))$

$$= \left.\frac{d\theta^*[X(t)]}{dt}\right|_{U(t),\;X(t)} + E\big[X(t),\;U(t)\big]$$

   where U is the operator's present control action at each instant of time (t).

3. Make the following observations:

   a. $\phi\,(X(t), U(t)) = 0$ if the operator is using optimal control,

   b. $\phi\,(X(t), U(t)) \geq 0$ if $[U(X(t))] \neq U^*[X(t)]$

      (if $\phi\,(X(t), U(t)) = 0$ for $U \neq U^*$, then control sensitivity to cost is zero),

c. ϕ (X(t), U(t)) indicates instantaneous performance directly,

d. ϕ (X, U) is a state related measure of cost,

e. $\dfrac{\partial\phi(X, U)}{\partial U}$ is a measure of the control sensitivity at each point X in state space (for fixed t) and therefore, weights the importance of control errors.

f. Let,

$$\overline{\phi} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \phi\left[X(t),\ U(t)\right] dt$$

This is a measure of average performance over the interval $[t_1, t_2]$,

g. Let,

$$\phi^2 = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left\{\phi\left[X(t),\ U(t)\right]\right\}^2 dt$$

This is a measure of average squared performance.

h. Then define,

$$s^2 = \phi^2 - (\overline{\phi})^2$$

This is a measure or index of performance variability.

i. Then $s = \sqrt{\phi^2 - (\overline{\phi})^2} = \text{R.M.S.}\left(\phi\left[X(t),\ U(t)\right]\right)$

which is an index of performance variability also. Similarly, higher order moments could provide indeces of performance asymmetry (skewness and kurtosis).

4.7 Continuous Performance Measure Illustrative Example

This section is devoted to the solution of an example problem in order to illustrate the concepts and techniques introduced in previous sections. In this section, the CPM is found for the infinite-time Linear Regulator Problem.

4.7.1 CPM For Linear Regulator Problem

Consider the standard infinite-time linear regulator problem (Sage, 1968, Bryson and Ho, 1967, Athans and Falb, 1966 and Anderson and Moore, 1971). Given the state equation as:

$$\dot{X}(t) = A\,X(t) + B\,U(t) \tag{4.24}$$

with arbitrary initial condition $X(0) = X_0$ where A and B are constant matrices, X(t) is the state vector, and U(t) is the control vector. The performance index is given by:

$$J = \tfrac{1}{2} \int_0^{\infty} \left[X^T(t)\,Q\,X(t) + U^T(t)\,R\,U(t) \right] dt \tag{4.25}$$

where Q and R are positive definite symmetric constant matrices.

The optimal control problem is to find the feedback control law $U^* = U(X)$ which transfers the system given by Equation 4.24 from any arbitrary initial state X_0 to the destination, while minimizing the performance index of Equation 4.25.

The well-known solution to this problem (Sage, 1968, Bryson and Ho, 1967, Athans and Falb, 1966 and Anderson and Moore, 1971) is the optimal feedback control law given by:

$$U^*(t) = -R^{-1}\,B^T\,K\,X(t) \tag{4.26}$$

where K is the positive definite symmetric constant matrix which is the solution of the matrix equation:

$$KBR^{-1}\,B^T\,K - KA - A^T K - Q = 0 \tag{4.27}$$

The minimum value of the performance index is given by:

$$J^* = \tfrac{1}{2}\,X^T(0)\,K\,X(0) \tag{4.28}$$

4.7.1.1 "Cost-to-Go" Function: $\theta^* [X(t)]$

The "cost-to-go" function must satisfy Equation 4.21 along an optimal trajectory. For this example:

$$E\left[X(t),\ U(t)\right] = \tfrac{1}{2}\left[X^T(t)\, Q\, X(t) + U^T(t)\, R\, U(t)\right] dt$$

but by Equation 4.26:

$$U^*(t) = -R^{-1}\, B^T\, X(t)$$

so $E[X(t),\ U^*(t)]$ becomes:

$$\tfrac{1}{2}\left[X^T(t)\, Q\, X(t) + \left(-R^{-1}\, B^T\, X(t)\right)^T R \left(-R^{-1}\, B^T\, X(t)\right)\right]$$

which may be simplified, using the matrix calculus (Gantmacher 1959), to the expression

$$E\left[X(t),\ U^*(t)\right] = \tfrac{1}{2}\, X^T(t)\left[Q + K\, B\, R^{-1}\, B^T\, K\right] X(t) \qquad (4.29)$$

Substituting eq. 4.26 in eq. 4.24,

$$\left.\frac{d\theta^*}{dt}\right|_{U^*,X} = \left[\frac{\partial\theta^*[X(t)]}{\partial X}\right]^T \left.\dot{X}(t)\right|_{U^*,X}$$

$$= \left(\frac{\partial\theta^*}{\partial X}\right)^T \left[A - BR^{-1}\, B^T\, K\right] X(t) \qquad (4.30)$$

So now the differential equation from which the optimal "cost-to-go" function derives (as defined by eq. 4.23) can be expressed by combining equations 4.29 and 4.30.

$$\left(\frac{\partial\theta^*}{\partial X}\right)^T \left[A - BR^{-1}\, B^T\, K\right] X(t)$$
$$+ \tfrac{1}{2}\, X^T(t)\left[Q + K\, B\, R^{-1}\, B^T\, K\right] X(t) \qquad (4.31)$$

Along the optimal trajectory, this expression is equal to zero (which implies that either both 4.29 and 4.30 are zero or that one is the negative of the other, i.e., they balance out or null one another). Further, by definition of θ^*, we know that at $x(t) = 0$ $\theta^*(X(t)) = 0$. This is a boundary condition imposed upon eq. 4.31.

The "cost-to-go" function which is the solution to Equation 4.31 is

$$\theta^* \left[X(t)\right] = \tfrac{1}{2} X^T(t)\, K\, X(t) \tag{4.32}$$

where K is the positive definite symmetric constant matrix which is the solution of Equation 4.27. Since K is positive definite, then $\theta^* [X(t)] > 0$ for any $X(t) \neq 0$; and

$$\theta^* \left[X(0)\right] = \tfrac{1}{2} X^T(0)\, K\, X(0) = J^*$$

and

$$\theta^* \left[0\right] = 0$$

4.7.1.2 Continuous Performance Measure $\phi\,[X(t),\ U(t)]$

The Continuous Performance Measure (CPM) evaluated at the present state and control is given by Equation 4.19 and is repeated here for convenience as Equation 4.33

$$\phi\left[X(t),\ U(t)\right] = \left.\frac{d\theta^*[X(t)]}{dt}\right|_{U(t)} + E\left[X(t),\ U(t)\right] \tag{4.33}$$

Borrowing from eq. 4.29 and 4.30, 4.33 may be written

$$\phi\left[X(t),\ U(t)\right] = \frac{\partial\theta}{\partial X}^{T}\left[A - B\ R^{-1}\ B\ K\right] X(t)$$

$$+ \ \tfrac{1}{2}\ X^{T}(t)\left[Q\ +\ K\ B\ R^{-1} B\ K\right] X(t)$$

which with approprite manipulation using the matrix calculus becomes

$$\phi\left[X(t),\ U(t)\right] = \tfrac{1}{2}\ X^{T}(t)\left[K\ B\ R^{-1}\ B^{T}\ K\right] X(t)$$

$$+ \ X^{T}(t)\ K\ B\ U(t)$$

$$+ \ \tfrac{1}{2}\ U^{T}(t)\ R\ U(t)\,. \qquad (4.34)$$

Note that if the optimal control U^* given by Equation 4.26 is used, then $\theta\left[X(t),\ U^*(t)\right] = 0$. That is, the CPM evaluated using the optimal control is zero.

Assume that the operator's present control action, U(t), can be written as some deviation from the optimal; that is,

$$U(t) = U^*(t) + e(t) \qquad (4.35)$$

where $U^*(t)$ is the optimal feedback control given by Equation 4.26 and e(t) is the control error. Note that the control error is the difference between the operator's control action at the present time and the optimal control action for the present state.

Using Equation 4.35 in Equation 4.34 the CPM in terms of the control error is given by:

$$\phi\left[X(t),\ U^*(t) + e(t)\right] = \phi\left[e(t)\right] = \tfrac{1}{2}\ e^{T}(t)\ R\ e(t) \qquad (4.36)$$

Since it was assumed that R is chosen as a positive definite matrix, this implies that $\phi[e(t)] > 0$ for $e(t) \neq 0$ and $\phi[e(t)] = 0$ for $e(t) = 0$. Hence the CPM is a positive definite function.

4.7.1.3 Sensitivity of Continuous Performance Measure

The sensitivity of the CPM to small variations in the operator's control action can be found by taking the partial derivative of Equation 4.34 with respect to U(t), which for this example is:

$$\frac{\partial\phi[X(t),\ U(t)]}{\partial U(t)} = B^T K X(t) + R\ U(t)$$

Note that the sensitivity of the CPM to the control is proportional to both the present state and the present control action for this linear-regulator example. This implies that non-optimal operator action is more serious in some states that in others.

The sensitivity of the CPM to small variations in the control error e(t) can be found for this example by taking the partial derivative of Equation 4.36 with respect to e(t), which is:

$$\frac{\partial\phi[e(t)]}{\partial e(t)} = R\ e(t)$$

Note that for this example the sensitivity of the CPM is directly proportional to the present control error weighted by the matrix R regardless of the present state. This implies that the changes in the performance index will reflect a cost weighted penalty for inappropriate action but will not again penalize him for being in some undesirable state. So long as he makes the best of a bad situation, he can keep the performance index "down," i.e., by minimizing his own errors (by responding in a manner appropriate to the specified and quantitative objectives) he can effectively produce a minimal performance index as he was instructed or set out to do.

5.0 APPLICATION OF CPM TO A CRUISING SEGMENT OF THE MISSION

In this section, the theoretical results and methodology of the preceeding three sections are used to find a continuous performance measure for a cruise segment (Segment 4) of the aircraft mission.

First, the optimal control problem is formulated for the cruising problem. This involves the formulation of the state equations and performance index. Next, an approximate solution of the optimal control problem is given for the cruising segment (Segment 4) of the aircraft mission. Based on the approximate optimal control law, the CPM for the cruising segment is derived. Preliminary computational results are presented for the CPM when a non-optimal control policy (an auto-pilot) is used to fly the aircraft in Segment 4.

5.1 Cruise Problem Formulation for Segment 4

The formulation of an optimal control problem for the cruising segment of the aircraft mission is developed in this section.

5.1.1 Selection of a Performance Index

A generalized performance index, common to many segments of a mission, was described in Section 2, and is given by:

$$
\begin{aligned}
J = & \left[X_R(t_f) - X(t_f)\right]^T S \left[X_R(t_f) - X(t_f)\right] \\
& + \int_{t_o}^{t_f} \Big\{ \left[X_R(t) - X(t)\right]^T Q \left[X_R(t) - X(t)\right] \\
& \qquad + \left[U_R(t) - U(t)\right]^T R \left[U_R(t) - U(t)\right] \\
& \qquad + \left[\dot{X}(t)\right]^T W \dot{X}(t) \Big\} \, dt
\end{aligned}
\tag{5.1}
$$

where t_o is the initial time, $X_R(t)$ and $U_R(t)$ are possible reference state and control functions, t_f is the final time, X is the vector of state variables of the system, U is a vector of control variables of the system, and S, W, Q, and R are weighting matrices. The generalized performance index is made up of a term that penalizes state variable errors, excessive control, and large rates of change of state variables.

For a general cruising mission segment, the assumed objective is to maintain constant heading, altitude, and velocity over a given distance with small changes in the control and state variables that may be required for any error correction. The final time (t_f) and the error in the terminal state ($X(t_f)_R - X(t_f)$) is assumed not to be of primary importance. The problem is terminated when the aircraft has travelled a specified distance over the earth. Therefore, Equation 5.1 is adapted to a cruising problem by not penalizing the error in the final state, i.e., let the weighting matrix S be all zero elements; and by letting the final time t_f be undefined, that is, $t_f = \infty$. In order to simplify the development, the rate of change of the state variables are not penalized in the performance index, i.e., the weighting matrix W has all zero elements. The objective for Segment 4 of the mission is to cruise at 18,000 feet altitude at a speed of 708.87 feet/sec. (420 knots). The direction to target is selected as due East, so that the aircraft should travel along the X_e - axis. The corresponding reference aircraft heading is taken as zero radians, ($\psi = 0$). The segment flight problem is terminated when the aircraft has traveled East for fifty nautical miles, (X_e = 303,805.75 feet (50 nm.)).

The desired steady-state flight condition for Segment 4 of the mission (straight and level flight due east) is the reference trajectory. When the aircraft flys along the reference trajectory all variables are constant except for X_e which is changing at a rate of 708.87 feet/sec. Figure 7 shows this reference trajectory in a vertical plane indicating altitude versus X-position of the aircraft with respect to earth. Also shown are several preferred (optimal) trajectories for several non-reference initial conditions.

Choose as the vector of state variables for this problem the 7-dimensional vector $\underline{X}$, defined by

$$\underline{X}(t) = \begin{bmatrix} \alpha\ (t) \\ \phi\ (t) \\ \psi\ (t) \\ \gamma\ (t) \\ V\ (t) \\ Y_e\ (t) \\ Z_e\ (t) \end{bmatrix} \tag{5.2}$$

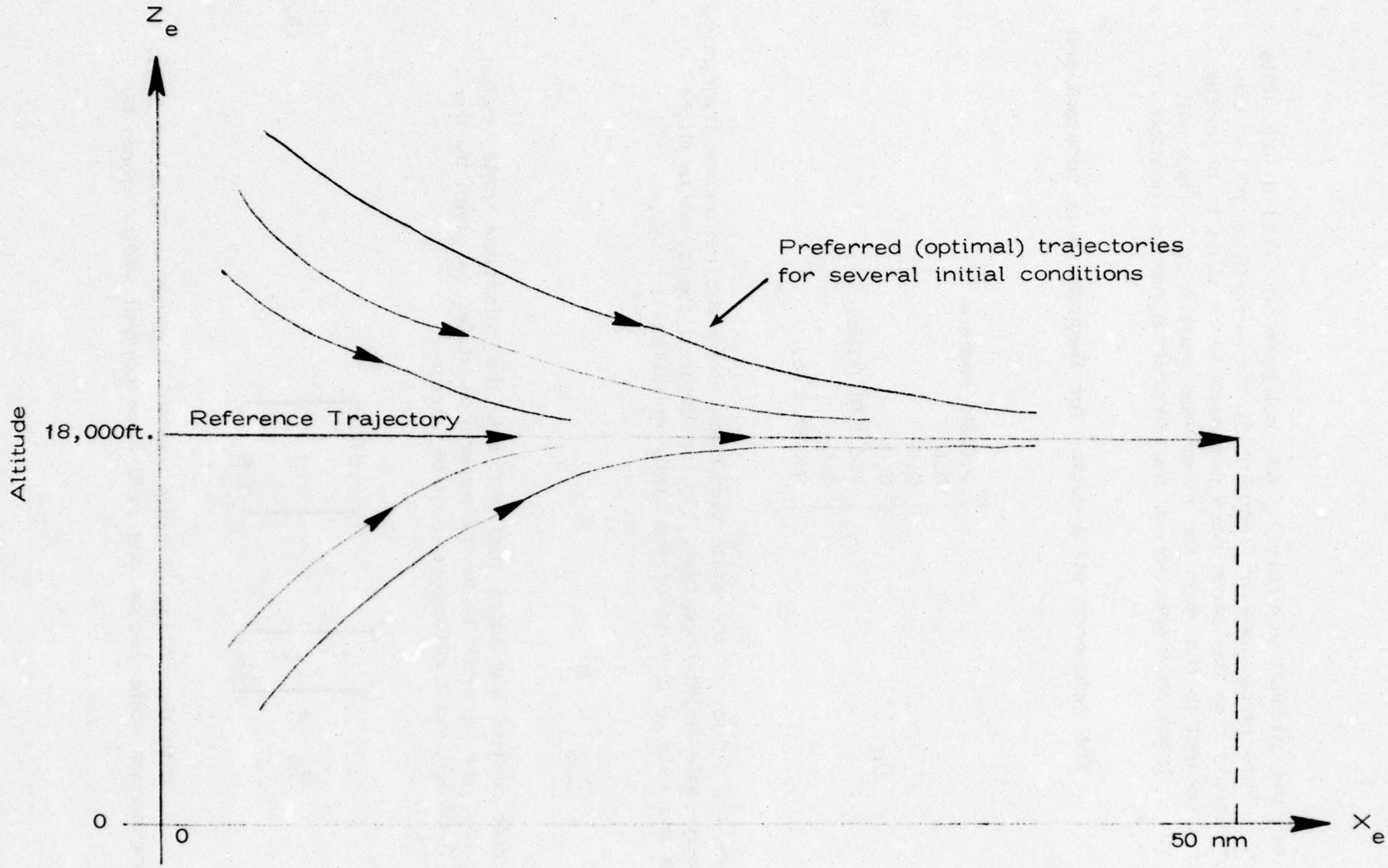


FIGURE 7 X-POSITION OF AIRCRAFT WITH RESPECT TO EARTH

Notice that the aircraft variable X_e has not been included in the state vector $\underline{X}$. This is because in Segment 4, the X-position of the aircraft with respect to the earth does not need to be used to generate a feedback control to maintain the reference trajectory. This will become clear from an inspection of the aircraft dynamic equation in Section 3.4.

The reference state vector for Segment 4 is constant and given by

$$\underline{X}_R = \begin{bmatrix} 0.0594 \text{ radians} \\ 0.0 \\ 0.0 \\ 0.0 \\ 708.876 \text{ ft./sec.} \\ 0.0 \\ 18000.0 \text{ ft.} \end{bmatrix} \tag{5.3}$$

These are the values of the state variables along the reference trajectory in a steady-state flight condition. In the desired steady-state flight condition the rate of change of the state variables is zero,

$$\dot{\underline{X}}_R = 0$$

The aircraft control variables that maintain the reference state vector $\underline{X}_R$, provided the aircraft is in the reference state, is given by the constant 3-dimensional reference control vector

$$\underline{U}_R = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \quad \begin{bmatrix} 0.0702 \\ 0.0 \\ 0.541 \end{bmatrix} \tag{5.4}$$

With the choice of state variables given in Equation 5.2, and the reference state vector and reference control vector given by

Equations 5.3 and 5.4 respectively, the performance index for the cruising problem of Segment 4 is given by Equation 5.5.

$$J = \tfrac{1}{2} \int_0^{\infty} \Big\{ (X_R - X(t))^T \; Q \; (X_R - X(t)) + (U_R - U(t))^T R \, (U_R - U(t)) \Big\} \, dt \tag{5.5}$$

The selection of the weighting matrices Q and R are discussed in Section 5.4.3.

The performance index for cruising (segment 4) given by Equation 5.5 tends to limit excessive control element displacements and insures that the reference trajectory is an optimal trajectory. The 1/2 term in front of the integral is merely a scaling factor for convenience.

5.1.2 State Variable Formulation of Aircraft Equation for Segment 4 of the Mission

The simplified set of equations (3.14 thru 3.21) that represent the model of aircraft dynamics for the cruising segment of the mission (Segment 4) are given in Section 3.4. The model of aircraft dynamics is put into state variable form as:

$$\dot{X}_e = V \cos \gamma \cos \psi \tag{5.6}$$

and the vector/matrix differential equation

$$\dot{X}(t) = F(x) \, X(t) + G(x) \, U(t) \tag{5.7}$$

where X(t) is the 7-dimensional state vector defined by Equation 5.2, and U(t) is the 3-dimensional control vector defined by

$$\underline{U}(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ U_3(t) \end{bmatrix} \tag{5.8}$$

The matrix F(x) is the (7 by 7) dimensioned system matrix whose elements are a function of the state vector, where

$$F(X) = \begin{bmatrix} f_{11}(x) & 0 & 0 & 0 & f_{15}(x) & 0 & f_{17}(x) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_{31}(x) & 0 & 0 & 0 & f_{35}(x) & 0 & 0 \\ f_{41}(x) & 0 & 0 & 0 & f_{45}(x) & 0 & f_{47}(x) \\ f_{51}(x) & 0 & 0 & 0 & f_{55}(x) & 0 & f_{57}(x) \\ 0 & 0 & 0 & 0 & f_{65}(x) & 0 & 0 \\ 0 & 0 & 0 & 0 & f_{75}(x) & 0 & 0 \end{bmatrix}$$

and G(X) is the (7 by 3) - dimensioned control matrix whose elements are a function of the state vector, where:

$$G(X) = \begin{bmatrix} 1 & 0 & g_{13}(x) \\ 0 & 1 & 0 \\ 0 & 0 & g_{33}(x) \\ 0 & 0 & g_{43}(x) \\ 0 & 0 & g_{53}(x) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The non-zero elements of F(X) and G(X) are:

$$f_{11} = (8.041 \times 10^{-6})\,(AL1)\,(CL2)\,V^2$$

$$f_{15} = (8.041 \times 10^{-6})\,(AL1)\,(CL1)\,V$$

$$f_{17} = -AL1/Z_e$$

$$f_{31} = (2.589 \times 10^{-4}\ CL2)\,\frac{\sin\phi}{\cos\gamma}\ V$$

$$f_{35} = (2.589 \times 10^{-4}\ CL1)\,\frac{\sin\phi}{\cos\gamma}$$

$$f_{41} = (2.589 \times 10^{-4})\,(CL2)\cos\phi\, V$$

$$f_{45} = (2.589 \times 10^{-4})\,(CL1)\cos\phi$$

$$f_{47} = -\frac{32.2\cos\gamma}{V Z_e}$$

$$f_{51} = -(3.4952 \times 10^{-5})\,V^2$$

$$f_{55} = -(4.3689 \times 10^{-4})\,\alpha^2\, V\ -(8.466 \times 10^{-6})\,V$$

$$f_{57} = -\frac{32.2\sin\gamma}{Z_e}$$

$$f_{65} = \cos\gamma\sin\psi$$

$$f_{75} = \sin\gamma$$

$$g_{13} = (AL1)\left[(1.241 \times 10^{-4})\,V + 0.2059\right]\ \sin\alpha$$

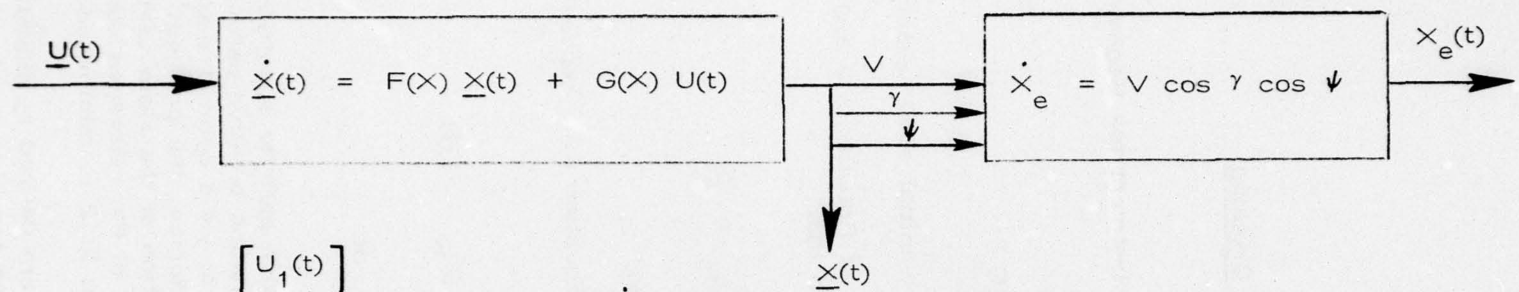
$$g_{33} = \left[(3.997 \times 10^{-3}) + \frac{6.6288}{V}\right]\ \frac{\sin\alpha\sin\phi}{\cos\gamma}$$

$$g_{43} = \left[(3.997 \times 10^{-3}) + \frac{6.6288}{V}\right]\ \sin\alpha\cos\phi$$

$$g_{53} = \left[(3.997 \times 10^{-3})\,V + 6.6288\right]\ \cos\alpha$$

Figure 8 shows a block diagram of the structure of the aircraft model used for Segment 4. Note that in the steady state flight condition, that is on the reference trajectory, Equation 5.7 becomes:

$$\dot{X}_R = F(X_R)\, X_R + G(X_R)\, U_R = 0 \tag{5.9}$$



$$\underline{U}(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ U_3(t) \end{bmatrix} = \text{Control Vector}$$

$$X_e = \text{X-position of Aircraft With Respect to Earth}$$

$$\underline{X}(t) = \begin{bmatrix} \alpha(t) \\ \phi(t) \\ \psi(t) \\ \gamma(t) \\ V(t) \\ Y_e(t) \\ Z_e(t) \end{bmatrix} = \text{State Vector}$$

FIGURE 8 BLOCK DIAGRAM SHOWING STRUCTURE OF AIRCRAFT MODEL FOR CRUISING PROBLEM

5.2 Optimal Control Problem For Cruising

5.2.1 Statement of the Problem

The optimal control problem statement for Segment 4 of the mission is as follows: Given:

1, the state equation

$$\dot{X}(t) = F(X)\, X(t) + G(X)\, U(t) \tag{5.10}$$

described in Section 5.1.2, with an arbitrary initial state vector $X(0)$,

2, the reference state vector of Equation 5.3 and the reference control vector of Equation 5.4 such that

$$\dot{X}_R = F(X_R)\, X_R + G(X_R)\, U_R = 0 \tag{5.11}$$

is satisfied along the reference trajectory, and

3, the performance index of Equation 5.5, repeated for convenience,

$$J = \tfrac{1}{2} \int_0^\infty \left[(X_R - X(t))^T\, Q\, (X_R - X)(t) \right.$$

$$\left. + (U_R - U(t))^T\, R\, (U_R - U(t)) \right] dt \tag{5.12}$$

where it is assumed that R is a (3 x 3) positive definite symmetric matrix of constants and Q is an (7 x 7) symmetric positive definite matrix of constants whose values are chosen to yield optimal trajectories which are the preferred aircraft trajectories, the problem is to find the feedback control law, U as a function of the state variables, such that any initial state $X(0)$ is transferred to the reference state X_R and the performance index, J, of Equation 5.12 is minimized.

Note that the optimization problem defined by Equation 5.10 and 5.12 is similar to the standard linear tracking problem of optimal control theory (Bryson and Ho, 1967 or Athans and Falb, 1966). The problem formulated here differs because the F and G matrices elements are functions of the state variables. This is also analogous to the classical compensatory tracking task used in laboratory studies of manual control, where F and G constitute the so called "plant dynamics." Here the plant dynamics are non-linear and time varying, again because of F and G being functions of the state variables.

5.2.2 Approach to a Solution

There are several different approaches which can be taken to solve this non-linear optimization problem. Applying Pontryagin's maximum principle will yield a set of necessary conditions for the solution to the optimization problem. However, these necessary conditions would be in the form of a set of non-linear differential equations which must be solved for given boundary conditions. Although this approach is feasible for finding open-loop control (control as a function of time), it is not practical for developing feedback control laws (control as a function of state).

A second approach is to use the method of continuous dynamic programming developed by Bellman which yields a sufficient condition for the optimum. This condition is in the form of a partial differential equation known as the Hamilton-Jacobi-Bellman equation. In general, the Hamilton-Jacobi-Bellman equation cannot be easily solved; however when it can, the control is determined as a function of the state variables, i.e., feedback control. A detailed discussion and derivation of Pontryagin's maximum principle and the Hamilton-Jacobi-Bellman equation can be found in Sage (1968), Bryson and Ho (1967), Athans and Falb (1966) and Anderson and Moore (1971).

The optimal control problem for the cruising segment can be approximately solved using the Hamilton-Jacobi-Bellman equation. However, the standard approach is modified so that this partial differential equation need not be solved directly. This modified approach has been successfully used on other types of optimal control problems (Zeskind and Vimolranich, 1973).

5.2.3 Hamilton-Jacobi-Bellman Equation For Optimal Cruising Problem

Equation 5.13 is the Hamilton-Jacobi-Bellman equation, where $H(X, U, \frac{\partial J}{\partial X}, t)$ is the Hamiltonian.

$$\frac{\partial J}{\partial t} = -\min_{U} H(X, U, \frac{\partial J}{\partial X}, t) \tag{5.13}$$

It states that the partial derivative of the optimal performance index with respect to time is equal to the negative of the Hamiltonian evaluated along the optimal trajectory, that is, evaluated using the minimizing value of U. The Hamiltonian for the problem considered here is

$$H = \tfrac{1}{2} [X_R - X(t)]^T Q [X_R - X(t)]$$

$$+ \tfrac{1}{2} (U_R - U(t)^T) R (U_R - U(t))$$

$$+ \left(\frac{\partial J}{\partial X}\right)^T [F(X) X(t) + G(X) U(t)] \tag{5.14}$$

Minimizing the Hamiltonian with respect to the control, the matrix calculus allows us to obtain:

$$\frac{\partial H}{\partial U} = RU - R U_R + G(X)^T \frac{\partial J}{\partial X} = 0 \tag{5.15}$$

Since by assumption R is positive definite, it has an inverse. Therefore, the optimal control is given by:

$$U^*(t) = U_R - R^{-1} G^T(X) \frac{\partial J}{\partial X} \tag{5.16}$$

Equation 5.16 is the control which minimizes the Hamiltonian, since

$$\frac{\partial^2 H}{\partial U^2} = R > 0 \tag{5.17}$$

Note that Equation 5.16 gives the optimal feedback control in terms of $\frac{\partial J}{\partial X}$ and not in terms of J. This point is discussed later and is the key to the solution of the optimal control problem.

Since the system given by Equation 5.10 and the Q and R matrices of Equation 5.12 are time invariant, and since the optimization is for a process considered over an infinite duration, it follows that the performance index will depend only upon the state variables. This implies that

$$\frac{\partial J}{\partial t} = 0 \tag{5.18}$$

Substituting Equation 5.16 into Equation 5.14 and using Equation 5.18, the Hamilton-Jacobi-Bellman equation for this problem becomes:

$$\tfrac{1}{2}(X_R - X)^T Q (X_R - X) - \tfrac{1}{2}\left(\frac{\partial J}{\partial X}\right)^T G R^{-1} G^T \frac{\partial J}{\partial X} + \left(\frac{\partial J}{\partial X}\right)^T F X + \left(\frac{\partial J}{\partial X}\right)^T G U_R = 0 \tag{5.19}$$

Note that the notational dependency of F and G on X, and that of X on t has been dropped at this point for convenience. From here on in the discussion, F is used instead of F(X), G instead of G(X) and X instead of X(t).

Adding and substracting $\left(\frac{\partial J}{\partial X}\right)^T F(X)\, X_R$ in Equation 5.19, and grouping terms, the Hamilton-Jacobi-Bellman equation becomes

$$\tfrac{1}{2}(X_R - X)^T Q (X_R - X) - \tfrac{1}{2}\left(\frac{\partial J}{\partial X}\right)^T G R^{-1} G^T \frac{\partial J}{\partial X} + \left(\frac{\partial J}{\partial X}\right)^T F(X - X_R) + \left(\frac{\partial J}{\partial X}\right)^T [F X_R + G U_R] = 0 \tag{5.20}$$

Notice that Equation 5.20 is similar to the Hamilton-Jacobi-Bellman equation for the standard linear regulator problem (Athans and Falb, 1966), except for the last term which involves $[F(X) X_R + G(X) U_R]$.

5.2.4 An Approximate Solution to the Optimal Control Problem

Equation 5.20 is a non-linear first order partial differential equation for the optimal performance index. If this equation can be solved for J, as a function of X, a feedback control law can be obtained. However, from inspection of Equation 5.16 the optimal feedback control law does not depend directly on J, but depends on $\frac{\partial J}{\partial X}$. Therefore, it is the solution of $\left(\frac{\partial J}{\partial X}\right)$ in terms of X which is really of interest in finding a feedback control law. From this point of view, Equation 5.20 can be considered as a non-linear equation in the unknown $\left(\frac{\partial J}{\partial X}\right)$.

From physical insight into the nature of the problem and from inspection of the structure of Equation 5.20, assume that the following approximate relationship holds

$$\frac{\partial J}{\partial X} \cong K(X)\,(X - X_R) \tag{5.21}$$

for values of X in the neighborhood of X_R.

Equation 5.21 gives an approximation to $\left(\frac{\partial J}{\partial X}\right)$ for "reasonable" values of X(t), where K(X) is an (7 x 7) symmetric matrix whose elements are a function of the state variables. Substituting Equation 5.21 into Equation 5.20, the Hamilton-Jacobi-Bellman equation can be written as:

$$\begin{aligned} &\tfrac{1}{2}\,(X_R - X)^T\,[Q + F^T K + KF - K\,GR^{-1}\,G^T K]\,(X_R - X) \\ &+ (X - X_R)^T\,K\,[F\,X_R + G\,U_R] = 0 \end{aligned} \tag{5.22}$$

Choose the matrix K(X), such that for every value of X, it is the positive definite solution to the matrix equation:

$$Q + F^T K + KF - K\,GR^{-1}\,G^T K = 0 \tag{5.23}$$

Equation 5.23 is the steady state matrix Riccati equation (Anderson and Moore, 1971), and is a function of X. Appendix C presents an iterative method for solving this equation for each value of X.

A unique positive definite solution to Equation 5.23 exists if the coefficient matrices of the system are controllable. Thus, Equation 5.23 has a unique positive definite solution if the pair $F(X(t))$, $G(X(t))$ is chosen such that the matrix

$$M = [\, G(X) \;\vdots\; F(X)\, G(X) \;\vdots\; \ldots \;\vdots\; F^{6}(X)\, G(X) \,]$$

has rank 7 for all values of X in the range of interest. Wernli and Cook (1975) contains a discussion of this type of equation.

Using Equation 5.23, Equation 5.22 reduces to

$$(X-X_R)^T K \, (FX_R + GU_R) \cong 0 \tag{5.24}$$

If Equation 5.24 is approximately zero for the range of values of X of interest in the problem, then Equation 5.21 gives a good approximation for $\frac{\partial J}{\partial X}$. Notice that as $X(t)$ approaches X_R the approximation becomes better and better, since $[\, F(X)\, X_R + G(X)\, U_R \,] \rightarrow 0$ as $X \rightarrow X_R$.

If the range of X in the problem is restricted to values for which Equation 5.24 is true, the approximate optimal feedback control law is given by:

$$U^*(t) \cong U_R - R^{-1} G^T(X)\, K(X)\, (X(t) - X_R) \tag{5.25}$$

As $X(t)$ approaches X_R, the optimal control approaches U_R and Equation 5.25 becomes a better and better approximation to the exact optimal feedback control law for the optimization problem posed in Section 5.2.1.

5.2.5 Structure of the Closed-Loop System

Using the approximate optimal feedback control law of Equation 5.25 results in the closed-loop system

$$\dot{X}(t) = F(X)\, X(t) + G(X)\, U_R - GR^{-1} G^T K \, (X(t) - X_R) \tag{5.26}$$

Adding and subtracting $F(X)\, X_R$ to the right hand side of Equation 5.26 and rearranging the terms, the closed loop system reduces to

$$\dot{X}(t) = + [F - GR^{-1} G^T K] (X(t) - X_R)$$

$$+ F(X)\, X_R + G(X)\, U_R \tag{5.27}$$

Equation 5.27 shows that the structure of the closed loop system is such that the rate of change of the state variables is linearly related to the state error $[X(t) - X_R]$. Therefore, if the system is at the reference state it will stay there. This implies that the reference trajectory for a cruising problem is an optimal solution to the problem presented. This concurs with intuition, i.e., the mathematics have led to no surprises for the relatively simple problem of maintaining a cruise attitude. Figure 9 shows a block diagram of the closed-loop system.

5.2.6 Stability of the Closed-Loop System

The stability of the closed-loop system is discussed in this section. Again, if the mathematical derivations are to agree with our intuition, we rationalize that the derived solution should prove to yield stable results since we are examining the cruise phase which intuitively should be steady and smooth. Liapunov's direct method is applied to the system for values of X in the neighborhood of the reference state X_R. (Padulo and Arbib (1974) contains a discussion of Liapunov's direct method, while Anderson and Moore, 1971 contains an application to optimal feedback control problems.)

The reference state X_R is an equilibrium point of the closed-loop system, since from Equation 5.27, $X(t) = 0$ when $X(t) = X_R$. Choose as a Liapunov function for the equilibrium point X_R

$$V(X) = \tfrac{1}{2} [X(t) - X_R]^T K(X) [X(t) - X_R] \tag{5.28}$$

which is positive definite since $K(X)$ is the positive definite solution of the algebraic Riccati equation. Differentiating $V(X)$ with respect to time and using Equation 5.27 for $X(t)$, yields

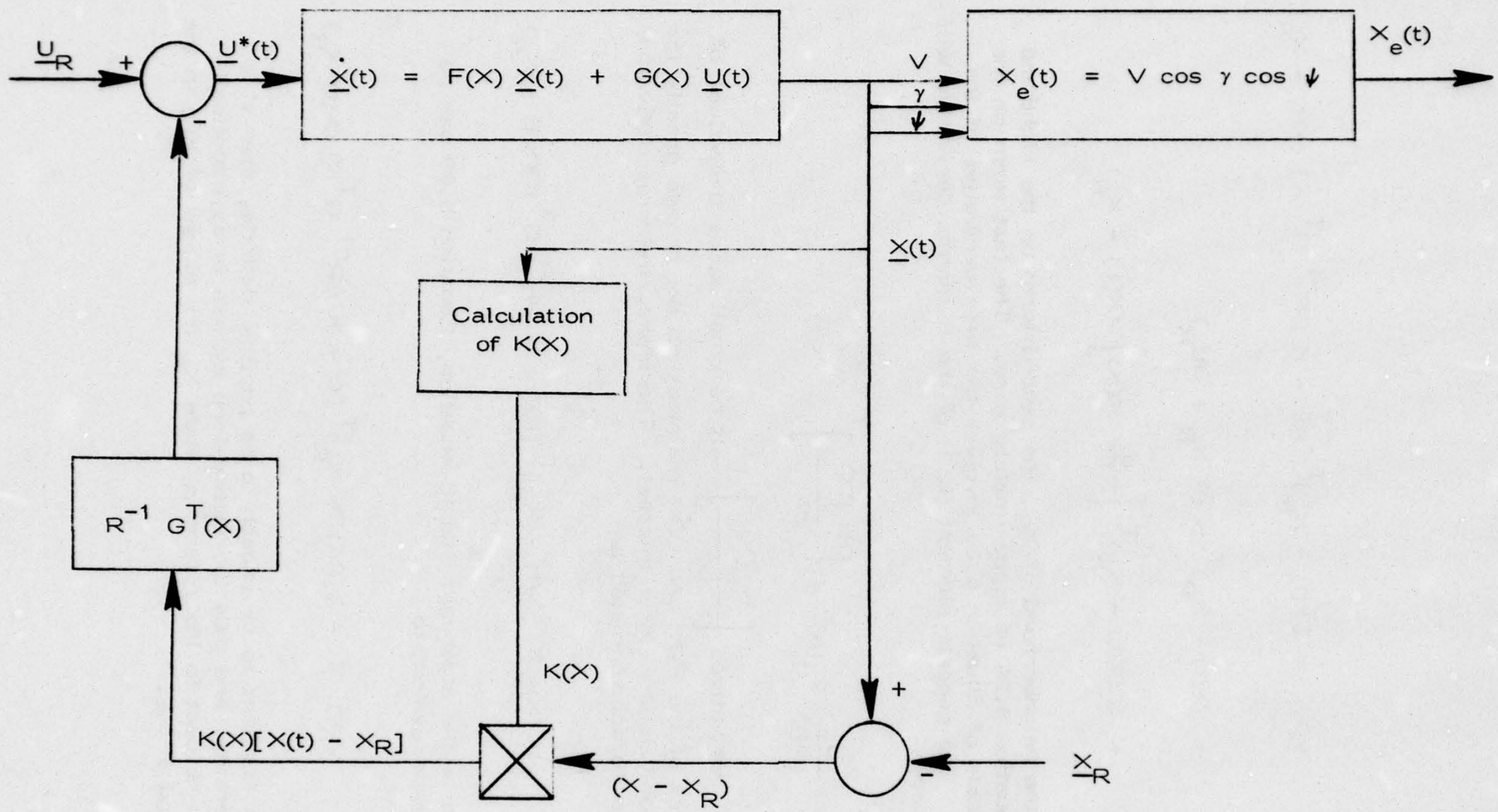


FIGURE 9 BLOCK DIAGRAM OF APPROXIMATE OPTIMAL FEEDBACK CONTROL FOR CRUISING (SEGMENT 4)

$$\dot{V}(X) = [X(t) - X_R]^T [KF - K\,GR^{-1}\,G^T K]\,(X(t) - X_R)$$

$$+ \; (X(t) - X_R)^T K\,(F\,X_R + GU_R)$$

$$+ \; \tfrac{1}{2}\,(X(t) - X_R)^T \left[\frac{d}{dt} K(X)\right] (X(t) - X_R) \qquad (5.29)$$

For X in the neighborhood of X_R, the second term on the righthand side of Equation 5.29 is approximately zero. The last term on the righthand side of Equation 5.29 involves the time derivative of the matrix K. The general element (i, j) of the K matrix derivative will be of the form

$$\frac{d}{dt}\,k_{ij}(X) = \left[\frac{\partial k_{ij}(X)}{\partial X}\right]^T \dot{X}(t)$$

The partial derivatives $\left[\frac{\partial k_{ij}(X)}{\partial X}\right]$ will be small since the values of the elements of the F(X) and G(X) matrices do not change greatly for the range of values of X of interest. Therefore, the time derivative of V(X) can be approximated as

$$\dot{V}(X) \cong [X(t) - X_R]^T [KF - K\,GR^{-1}\,G^T K](X(t) - X_R) \qquad (5.30)$$

Making use of the algebraic Riccati equation, Equation 5.30 can be shown to be equivalent to

$$\dot{V}(X) \cong -\tfrac{1}{2}\,(X(t) - X_R)^T [Q + K\,GR^{-1}\,G^T K]\,(X(t) - X_R) \qquad (5.31)$$

Since Q is assumed to be chosen to be positive definite, then $\dot{V}$ is negative definite, and thus the closed-loop system is asymptotically stable with respect to the reference state X_R for values of X(t) in the neighborhood of X_R.

5.3 CPM For Cruising Segment of Aircraft Mission

The approximate optimal feedback control law found in Section 5.2 can now be used with the results presented in Section 4 to find the CPM for Segment 4 of the aircraft mission. First, the "cost-to-go" function is derived. Next the approximate CPM is derived based on the approximate optimal control.

5.3.1 "Cost-to-Go" Function, $\theta^*[X(t)]$

In Section 4 it was shown that the "cost-to-go" function $\theta^*[X(t)]$ evaluated using optimal control satisfied the following equation:

$$\left.\frac{d\,\theta^*[X(t)]}{dt}\right|_{U^*(t)} + E[X(t), U^*(t)] = 0 \tag{5.32}$$

Since θ is only a function of the state,

$$\left.\frac{d\,\theta^*[X(t)]}{dt}\right|_{U^*(t)} = \left.\frac{\partial\theta^*}{\partial X}^T \dot{X}(t)\right|_{U^*(t)} \tag{5.33}$$

Using Equations 5.33, 5.10, 5.12 and 5.25, Equation 5.32 for this example reduces to

$$\left.\frac{d\,\theta^*[X(t)]}{dt}\right|_{U^*(t)} + E[X(t), U^*(t)] \tag{5.34}$$

$$= \left[\left(\frac{\partial\theta^*}{\partial X}\right)^T - (X - X_R)^T K\right](F - GR^{-1}G^T K)(X - X_R)$$

$$+ \left(\frac{\partial\theta^*}{\partial X}\right)^T (F X_R + G U_R) = 0$$

As an approximation to $\left(\frac{\partial \theta^*}{\partial X}\right)$ choose

$$\frac{\partial \theta^*}{\partial X} \cong K(X) \, (X(t) - X_R) \tag{5.35}$$

Using the approximation of Equation 5.35, Equation 5.34 becomes

$$(X - X_R)^T \, K \, [F \, X_R + G \, U_R] \cong 0 \tag{5.36}$$

If the left-hand side of Equation 5.36 is approximately zero then the choice of $\frac{\partial \theta^*}{\partial X}$ defined by Equation 5.35 is a good approximation. Notice that Equation 5.36 is the same as Equation 5.24 and hence, the comments for Equation 5.24 also apply to Equation 5.36.

5.3.2 CPM For General Cruising Problem

The continuous performance measure was developed in Section 4 for a general class of problems. The CPM is given by Equation 4.19 which for this specific problem can be written as:

$$\phi[X(t), U(t)] = \left(\frac{\partial \theta^*}{\partial X}\right)^T \dot{X}(t) \Bigg|_{U(t)} + E[X(t), U(t)] \tag{5.37}$$

since θ^* only depends on the state. Making use of Equation 5.7 for $\dot{X}$, Equation 5.5 for $E[X(t), U(t)]$ and Equation 5.35 for $\frac{\partial \theta^*}{\partial X}$, the CPM for this example is given by:

$$\begin{aligned} \phi[X(t), U(t)] &= \tfrac{1}{2} (X_R - X)^T \, Q \, (X_R - X) \\ &+ \tfrac{1}{2} (U_R - U)^T \, R \, (U_R - U) \\ &+ (X - X_R)^T \, K \, (FX_R + GU_R) \end{aligned} \tag{5.38}$$

If the approximate optimal control given by Equation 5.25 is used in Equation 5.38 the CPM is approximately equal to zero; since Equation 5.36 is approximately zero for the values of X of interest.

Assume that the operators present control action can be written as the approximate optimal control plus a control error, e(t), that is:

$$U(t) = U^*(t) + e(t) \tag{5.39}$$

Using Equation 5.39 in Equation 5.38, the CPM can be written in terms of the control error as:

$$\phi[e(t)] = \tfrac{1}{2} e^T(t)\, R\, e(t) + (X - X_R)^T K \left[FX_R + GU_R\right] \tag{5.40}$$

However, since it is assumed that Equation 5.36 holds for the value of X considered,

$$\phi[e(t)] \cong \tfrac{1}{2} e^T(t)\, Re(t) \tag{5.41}$$

Note that the CPM given by Equation 5.41 depends only on the control error.

5.4 Computer Simulation

In order to demonstrate the CPM technique developed in the preceeding sections, a digital computer program was written to solve the aircraft equations, optimal control law, auto-pilot control law, and CPM for Segment 4 of the mission. The aircraft equations used in this simulation are given in Section 5.1. The optimal control law and CPM implemented are those developed in Sections 5.2 and 5.3 respectively. Auto-pilot equations were used to generate a control vector U(t) that would fly the aircraft in a stable but non-optimal manner in order to demonstrate the measurement capability of the CPM.

A FORTRAN digital computer program written to demonstrate the CPM applied to Segment 4 of the mission is documented in this section. Appendix B contains a more detailed description and a listing of the program.

5.4.1 Auto-pilot

The auto-pilot designed to correct initial aircraft errors and bring the aircraft to the steady state flight conditions, is as follows:

The control $U_1(t)$ representing the longitudinal stick position which controls the rate of change of angle of attack is proportional to the altitude error Z_e, the rate of change of altitude, $\dot{Z}_e$, and the rate of change of the flight path angle, $\dot{\gamma}$. However the altitude error was hard limited by:

$$\Delta Z_e(t) = \begin{cases} +\ 100.0, & \text{if } (Z_{e_R} - Z_e(t)) > 100.0 \\ (Z_{e_R} - Z_e(t)), & \text{if } -100.0 \leq (Z_{e_R} - Z_e(t)) \leq 100.0 \\ -\ 100.0, & \text{if } (Z_{e_R} - Z_e(t)) < -\ 100.0 \end{cases} \tag{5.42}$$

where the reference altitude Z_{e_R} equals 18,000 feet for Segment 4. The auto-pilot control U_1 was expressed as follows:

$$U_1(t) = 0.001 \ \Delta Z_e - 0.001 \ \dot{Z}_e(t) - 4.0 \ \dot{\gamma}(t) \tag{5.43}$$

If $\alpha(t) > 0.2$ and $U_1(t) > 0$, then $U_1(t)$ was redefined $U_1(t) = 0$. Similarly, if $\alpha(t) < -0.2$ and $U_1(t) < 0$, then $U_1(t)$ was again set to zero. This limiting process keeps the aircraft model from producing an excessive angle of attack.

The control $U_2(t)$ that represents lateral stick position (to control the rate of change of roll angle ϕ) is given by:

$$U_2(t) = 0.1 \left[\psi_R - \psi(t)\right] - 2.2 \ \dot{\psi}(t) \tag{5.44}$$

where ψ_R is the reference heading, which for this problem is zero radians. In order to limit aircraft roll angle (ϕ) to 40° or less, $U_2(t)$ is set to zero if either $\phi(t) > 0.69812$ and $U_2(t)$ of Equation 5.44 is greater than zero or if $\phi(t) < -0.69812$ and $U_2(t)$ of Equation 5.44 is less that zero.

The control $U_3(t)$ that represents normalized percent throttle is given by:

$$U_3(t) = U_3(0) - \int_0^t \Big\{ 0.001\ (V(t) - V_R) + 0.120\ \dot{V}(t) \Big\}\ dt \tag{5.45}$$

where V_R is the reference velocity which for Segment 4 is 708.876 feet/sec. (420 knots) and $U_3(0)$ is the initial throttle setting. Since $U_3(t)$ is normalized percent throttle, the value given by Equation 5.45 must be between 0.0 and 1.0, thus

$$U_3(t) = \begin{cases} 0.0 \text{ if } U_3(t) < 0.0 \\ U_3(t) \text{ if } 0.0 \leq U_3(t) \leq 1.0 \\ 1 \text{ if } U_3(t) > 1.0 \end{cases}$$

This auto-pilot corrects for velocity, heading, and altitude. It does not correct any error in the y-position of the aircraft.

5.4.2 FORTRAN Computer Program

A brief description of the FORTRAN computer program is given in this subsection. Appendix B contains a more detailed discussion and documentation of this program. Figure 10 shows a simplified flow chart of the computer program structure.

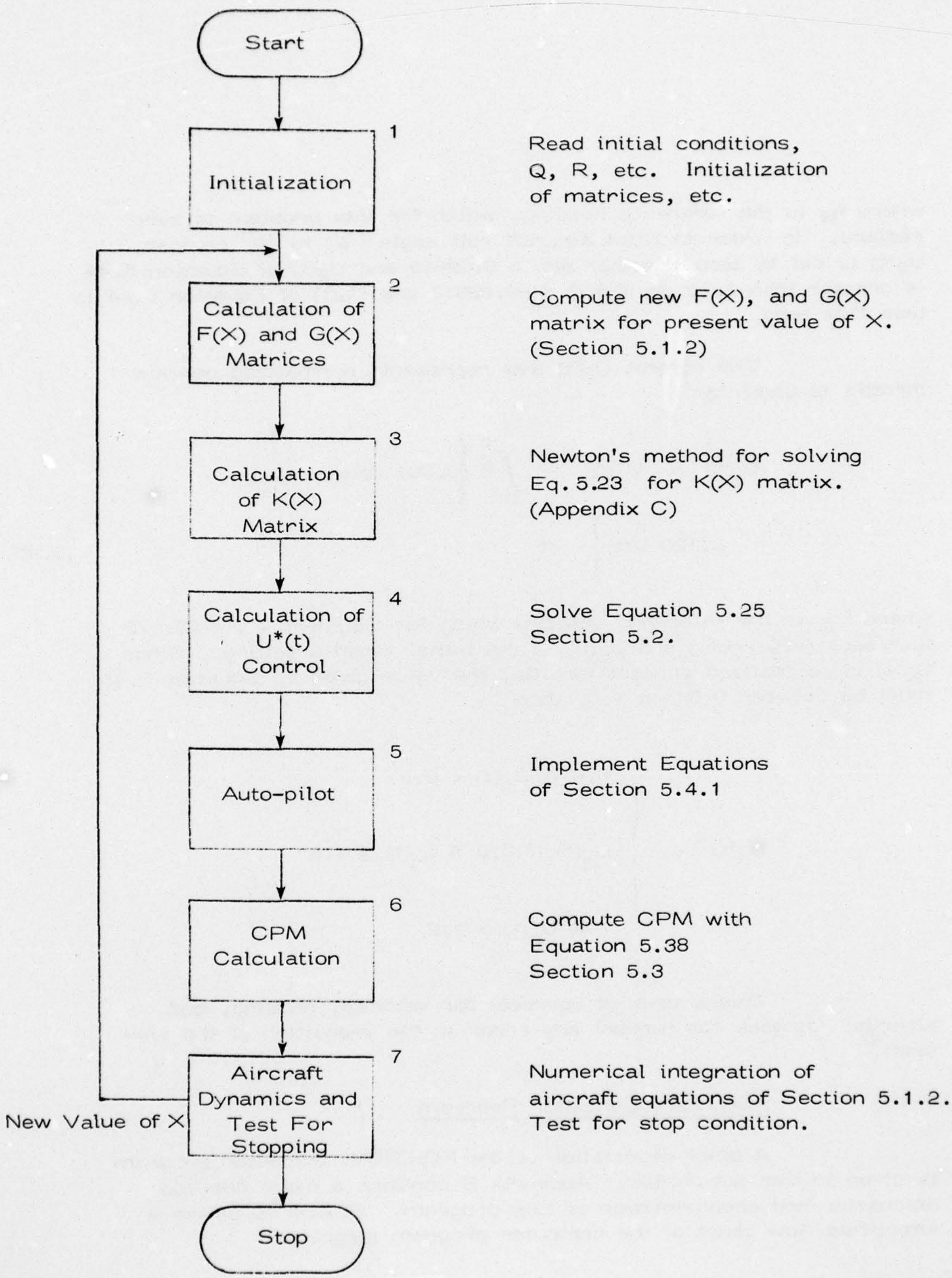


FIGURE 10 FLOW CHART OF COMPUTER PROGRAM STRUCTURE

In the initialization section of the program, variables are defined, dimensioned and initialized and the initial conditions, reference conditions and diagonal elements of the Q and R matrices are read. In the second section, the elements of the F matrix and G matrix are calculated for the present value of the state variables, X. The defining relationship for the non-zero elements of these matrices are given in Section 5.1.2.

Block 3 of coding implements an iterative method of computing for the present value of X, and the matrix K(X), which is the solution to Equation 5.23. The matrices F(X) and G(X) calculated in the previous section of the program are used in this section. Appendix C contains a detailed discussion of an iterative technique for algebraic steady state Riccati equation computations.

Once a numerical value for the elements of the matrix K(X) has been found, the approximate optimal control law is calculated. Equation 5.25 of Section 5.2 is implemented in the program. However, since in the development of Section 5.2 no limits were placed on the control variables, it was deemed appropriate to use the same limits used in the auto-pilot.

Block 5 of the computer program implements the auto-pilot equations discussed in Section 5.4.1. Values of the control variables are calculated based on the present values of the aircraft variables.

Next, (Block 6) values of the auto-pilot control and the optimal control are used to compute the CPM from Equation 5.41. In the final section of the program, the aircraft variables are updated. First, either the auto-pilot control or the approximate optimal control is chosen to control the aircraft. This is done by logical comparison based on the value of an input parameter that was read in the initialization section of the program. Then the next value of the aircraft dynamic variables is calculated. This is done by numerically integrating the differential equations of the aircraft given in Section 3. Rectangular numerical integration is used with a step size of 0.5 seconds. The program then loops back and new values of the aircraft variables are used to update F(X) and G(X). If, however, the aircraft has flown for the predetermined amount of time, the program terminates execution and stops.

5.4.3 Example of Program Output

The output of the computer program is presented for a typical run for purposes of documentation and demonstration. The initial conditions chosen for the aircraft dynamic variables at the start of Segment 4 of the mission are:

$$\alpha(0) = 0.1 \text{ radians } (5.7 \text{ degrees})$$

$$\phi(0) = 0.1 \text{ radians}$$

$$\psi(0) = 0.1 \text{ radians } (5.7 \text{ degrees})$$

$$\gamma(0) = 0.1 \text{ radians}$$

$$V(0) = 708.80 \text{ ft/sec.}$$

$$Y_e(0) = 0.0 \text{ ft.}$$

$$Z_e(0) = 17{,}000 \text{ ft.}$$

$$X_e(0) = 0.0 \text{ ft.}$$

Thus the aircraft model starts out 1,000 feet below the reference with a misalignment of approximately 5.7 degrees in heading. The attitude is pitched up and rolling slightly to the right. The initial velocity is approximately that of the reference. The reference values of these variables for a steady state flight condition in Segment 4 are given in Section 5.1.1. The initial conditions chosen for the state variables are in the neighborhood of the reference state variables. So that, the approximate optimal control of Section 5.2.4 is applicable.

The Q and R matrix element values were chosen to give the solution trajectories desired; but the exact relationship between Q and R element values and the solution trajectory characteristics is not known before the optimal solution is obtained. As an initial guess, the values chosen are:

$$R = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-1} & 10^{-3} & 0 & 0 & 10^{-3} & 0 \\ 0 & 10^{-3} & 10^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-1} & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-4} \end{bmatrix}$$

The R matrix was chosen as the positive definite diagonal matrix with the three diagonal elements all equal to 10. This was done because it was felt that the error between the control (u) and reference control U_r should be weighed more heavily than the state errors as weighted by the Q matrix. The purpose for this choice was to try to keep the optimal control values (U^*) small. Also it was felt that one control should be weighted the same as any other, that is there did not seem to be any reason for unequal weights. Although one could develop rational arguments for other weights, this issue was not explored.

The form of the Q matrix was chosen for the following reasons. It was decided to weight the angle of attack (α) error the heaviest to keep the aircraft model from excessive angles of attack. Excessive angles of attack can lead to instability, so this variable is of some concern. The assigned weight reflects the seriousness or gravity of allowing the aircraft to assume high angles of attack. Roll angle, heading angle, flight path angle, and velocity were all weighted the same, (10^{-1}) but a factor of ten less than angle of attack. The y position and altitude were both weighed by 10^{-4}, since it was felt that these states could be corrected slowly over the segment. Note they are weighted 1/1000 times the weighting of roll angle, heading angle, flight path angle and velocity; and 1/10,000 times the weighting of angle of attack.

The off-diagonal terms in the Q matrix were included to introduce a weighted coupling of heading error and y-position error into the roll angle in order to allow the aircraft to roll and to correct for heading and y position error. Again, relatively small weights were used reflecting moderate preference rather than grave concern.

The Q matrix is positive definite and symmetric. This choice of Q and R matrices is only a preliminary one used for demonstration of the computational technique. In order to determine the suitable values of the elements of Q and R, the simulation would have to be run with the optimal control "flying" the aircraft model. From inspection of the resulting trajectories, the Q and R matrices could then be modified and the simulation repeated with these new values. This process is repeated until the elements of Q and R are those which give the desired solution trajectories. However, due to lack of time, only the trajectory with the initial condition and values of Q and R given above was run to demonstrate the program.

The simulation was run for 30 seconds of flight time on an IBM 370-155 digital computer. In the present version of the computer program the execution time is greater than real time. Several suggestions are presented in Section 6 to remedy this situation to explore the feasibility of calculating the CPM in real time.

Figure 11 is a sample of the printout for one iteration, 0.5 seconds of flight time, through the program. Figures 12 through 19 show plots of the aircraft variables resulting from the auto-pilot control. Figure 20 is a plot of the CPM versus flight time for this non-optimal, auto-pilot control. Summary measures, mean and variance are given on the figure.

The mean was calculated using the time weighted average:

$$\bar{\phi} \simeq \frac{\Delta t}{t_2 - t_1} \sum_{i=0}^{N-1} \phi(i)$$

where

$$t_2 = N \Delta t$$

and

$$\Delta t = 0.5 \text{ secs.}$$

$$N = 60$$

with

$$\sum_{i=0}^{N-i} \phi\,(i) = 133.2281$$

The variance was calculated from the expectation formula:

$$\hat{\sigma}^2 = E\left[(\bar{\phi} - \phi(i))^2\right]$$

$$= E\left[(\phi(i))^2\right] - \left[E(\phi(i))\right]^2$$

The expectation of the squared scores was again calculated using the time weighted average:

$$E\left[(\phi(i))^2\right] \simeq \frac{\Delta t}{t_2 - t_1} \sum_{i=0}^{N-1} [\phi(i)]^2$$

With $\sum [\phi(i)]^2 = 355.1372$,

the variance is $= 5.919 - 4.935 = 0.9839$

Alternately, one might use the statistical formulas that follow:

$$\bar{\phi} = \frac{\sum_{i=0}^{N-1} \phi(i)}{N} = \frac{133.2881}{60} = 2.22$$

and

$$\hat{\sigma}^2 = \frac{N \sum_{i=0}^{N-1} \left[\phi(i)\right]^2 - \left[\sum_{i=0}^{N-1} \phi(i)\right]^2}{N\ (N-1)}$$

$$= \frac{60(355.1372) - (133.2881)^2}{3540}$$

$$= 1.001$$

In either case, the variability in performance as reflected by $\hat{\sigma}^2$ is relatively small. No attempt was made to explore higher order moments. These would reflect the degree of assymmetry in the measure, i.e., whether performance was skewed to the higher or lower scores, or was symmetrical about the mean.

```
INITIAL CONDITIONS FOLLOWS

   THE INITIAL STATE VECTOR FOLLOWS
   X(1) =  0.100000E 00
   X(2) =  0.100000E 00
   X(3) =  0.100000E 00
   X(4) =  0.100000E 00
   X(5) =  0.708800E 03
   X(6) =  0.0
   X(7) =  0.0
   X(8) =  0.170000E 05
 REFERANCE X FOLLOWS
     0.000000      0.0          0.0          0.0          708.818848  303806.000000     0.0          18000.000000
 Q DIAGICNAL ASSIGNMENTS FOLLOW
     1.000000      0.100000     0.100000     0.100000     0.100000     0.0               0.000100     0.000100
 R DIAGINIAL DEFINITIONS FOLLOW
     10.000000     10.000000    10.000000
THE G MATRIX IS

    0.100000000E 01     0.0                 -0.195385725E-02

    0.0                 0.100000000E 01     0.0

    0.0                 0.0                 0.133714741E-03

    0.0                 0.0                 0.132603082E-02

    0.0                 0.0                 0.541460037E 01

    0.0                 0.0                 0.0

    0.0                 0.0                 0.0

    0.0                 0.0                 0.0

THE F MATRIX IS

 -0.669192E 00   0.0          0.0          0.0          -0.377648E-04   0.0          0.0          0.391764E-05

  0.0            0.0          0.0          0.0           0.0            0.0          0.0          0.0

  0.460305E-01   0.0          0.0          0.0           0.259766E-05   0.0          0.0          0.0

  0.456479E 00   0.0          0.0          0.0           0.257606E-04   0.0          0.0         -0.265894E-05

 -0.175598E 02   0.0          0.0          0.0          -0.909737E-02   0.0          0.0         -0.189096E-03

  0.0            0.0          0.0          0.0           0.990033E 00   0.0          0.0          0.0

  0.0            0.0          0.0          0.0           0.993345E-01   0.0          0.0          0.0

  0.0            0.0          0.0          0.0           0.998333E-01   0.0          0.0          0.0
```

FIGURE 11 SAMPLE PRINTOUT

```
THE FC MATRIX IS

  -0.135469E 01   0.224880E-04  -0.187690E-03  -0.186132E-02   0.550060E-01  -0.983535E-13   0.242671E-05  -0.581311E-05
   0.253799E-04  -0.104994E-01  -0.104994E-03  -0.109248E-07  -0.482351E-05   0.205880E-17  -0.104994E-03   0.451183E-09
   0.461016E-01  -0.570205E-08  -0.241461E-08  -0.239436E-07  -0.711346E-05   0.392158E-17  -0.566586E-09   0.596220E-09
   0.457183E 00  -0.565465E-07  -0.239453E-07  -0.237446E-06  -0.705432E-04   0.388898E-16  -0.561876E-08  -0.265302E-05
  -0.125583E 02  -0.401470E-03  -0.170008E-03  -0.168583E-02  -0.692839E 00   0.276111E-12  -0.398923E-04  -0.147117E-03
   0.0            0.0            0.0            0.0            0.990033E 00   0.0            0.0            0.0
   0.0            0.0            0.0            0.0            0.993345E-01   0.0            0.0            0.0
   0.0            0.0            0.0            0.0            0.998333E-01   0.0            0.0            0.0
CL1=      0.999999642E-01

CL2=      0.250000000E 01

L  =      0.243666289E 05

AL1=      0.665999651E-01
```


FIGURE 11 SAMPLE PRINTOUT

(CONTINUED)

IDE A MATRIX

0.684657E 01	-0.224047E-03	0.187726E-02	0.186167E-01	-0.549019E 00	0.982963E-12	-0.241843E-04	0.972205E-04
-0.253799E-03	0.104994E 00	0.104994E-02	0.109248E-06	0.482351E-04	-0.205880E-16	0.104994E-02	-0.451184E-08
0.155139E-02	0.104995E-02	0.104998E 00	0.224249E-06	0.184973E-04	0.214752E-15	0.515069E-09	0.434997E-08
0.193518E-01	0.970763E-07	0.224251E-06	0.105001E 00	0.183422E-03	0.212967E-14	0.933459E-08	-0.110410E-06
-0.562860E 00	0.452336E-04	0.180791E-04	0.179274E-03	0.770276E-01	-0.309479E-13	0.449573E-05	-0.471596E-05
0.109638E-11	-0.321777E-16	0.263044E-15	0.260862E-14	-0.426063E-13	0.115999E-23	-0.348443E-17	0.264208E-16
-0.272495E-04	0.104995E-02	0.639009E-09	0.105628E-07	0.479717E-05	-0.227563E-17	0.104999E-03	-0.453222E-09
0.912006E-04	-0.410018E-08	0.507572E-08	-0.892527E-07	-0.404184E-05	0.255689E-16	-0.412509E-09	0.105001E-03

```
EX(1) = -0.400000E-01     XXD(1) = -0.502686E-01

EX(2) = -0.100000E 00     XXD(2) = -0.105987E-02

EX(3) = -0.100000E 00     XXD(3) =  0.164361E-02

EX(4) = -0.100000E 00     XXD(4) =  0.209418E-01

EX(5) =  0.190430E-01     XXD(5) = -0.341174E 00

EX(6) =  0.303806E 06     XXD(6) = -0.168532E-01

EX(7) =  0.0              XXD(7) = -0.189162E-02

EX(8) =  0.100000E 04     XXD(8) = -0.190112E-02

UOP BEFORE LIMITS
UOP (2) = -0.105987E-02
UOP(3) =  0.117664E 01
UOP(1) = -0.212280E-01
UOP
 -0.212280E-01  -0.105987E-02   0.100000E 01
```

FIGURE 11 SAMPLE PRINTOUT
(CONTINUED)

THE TRANSPOSE OF KGR-1GT-FT FOLLOWS

0.135486E 01	-0.254687E-04	0.195103E-03	0.193482E-02	-0.563901E-01	0.109717E-12	-0.273378E-05	0.520988E-05
-0.224047E-04	0.104994E-01	0.104995E-03	0.970763E-08	0.452335E-05	-0.321777E-17	0.104995E-03	-0.410017E-09
-0.460998E-01	0.608068E-08	0.246532E-08	0.244467E-07	0.711383E-05	-0.539218E-17	0.604614E-09	-0.511200E-09
-0.457166E 00	0.603013E-07	0.244483E-07	0.242434E-06	0.705468E-04	-0.534735E-16	0.599587E-08	0.265387E-05
0.126810E 02	0.428129E-03	0.173579E-03	0.172124E-02	0.692865E 00	-0.379653E-12	0.425697E-04	0.153103E-03
0.0	0.0	0.0	0.0	-0.990033E 00	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-0.993345E-01	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-0.998333E-01	0.0	0.0	0.0

FIGURE 11 SAMPLE PRINTOUT
(CONTINUED)

TEST OF GOODNESS OF K
THE FOLLOWING MATRIX SHOULD BE APPROX. ZERO

0.393795E 01	-0.219254E-03	0.206785E-02	0.205064E-01	-0.228113E 00	-0.938345E-12	-0.242161E-04	0.586227E-04
-0.210764E-03	0.988975E-01	0.989015E-03	0.389911E-06	0.564012E-04	0.385005E-16	0.988974E-03	-0.557315E-08
0.196452E-02	0.989012E-03	0.999995E-0	-0.366585E-05	0.937703E-04	-0.186405E-15	-0.106300E-06	-0.140024E-07
0.194817E-01	0.354588E-06	-0.366585E-05	0.999036E-01	0.929964E-03	-0.184859E-14	0.390439E-07	-0.418048E-06
-0.199774E 00	0.598383E-04	0.925812E-04	0.918167E-03	0.149916E-01	0.768742E-13	0.594653E-05	0.246117E-05
-0.949205E-12	0.420983E-16	-0.198534E-15	-0.196689E-14	0.898289E-13	-0.119597E-24	0.436363E-17	-0.112183E-18
-0.232930E-04	0.988974E-03	-0.105939E-06	0.426212E-07	0.560235E-05	0.400680E-17	0.998895E-04	-0.546898E-09
0.603571E-04	-0.571482E-08	-0.124947E-07	-0.403096E-06	0.158253E-05	-0.368546E-18	-0.561908E-09	0.100001E-03

UPLT

0.100000E 00 -0.999999E-02 0.650009E 00

CONT. PERFORMANCE MEASURE = 0.686347E 00

TIME	X1	X2	X3	X4	X5	X6	X7	X8
0.5	0.748656E-01	0.994700E-01	0.103289E 00	0.110015E 00	0.707798E 03	0.350868E 03	0.352041E 02	0.170354E 05

FIGURE 11 SAMPLE PRINTOUT
(CONCLUDED)

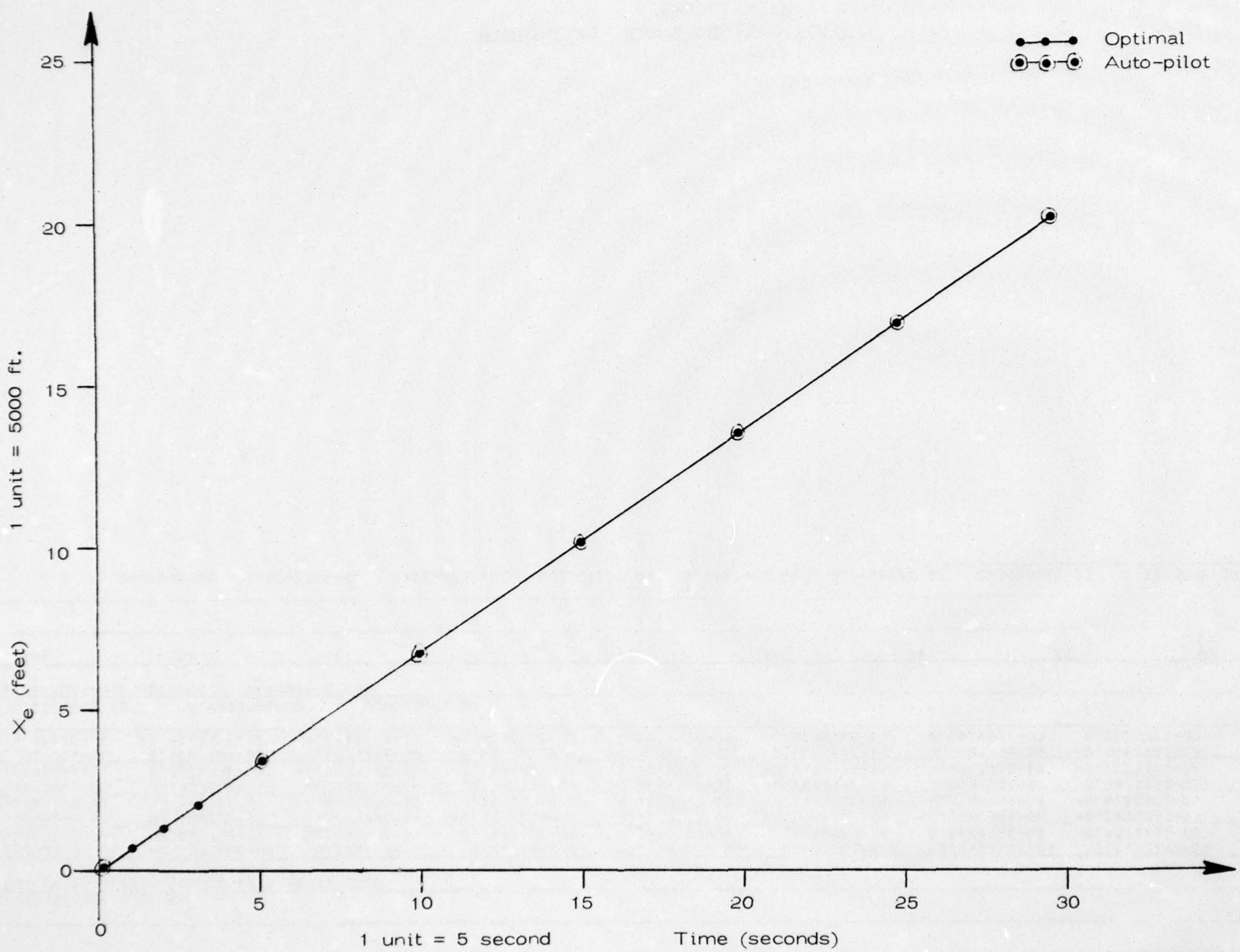
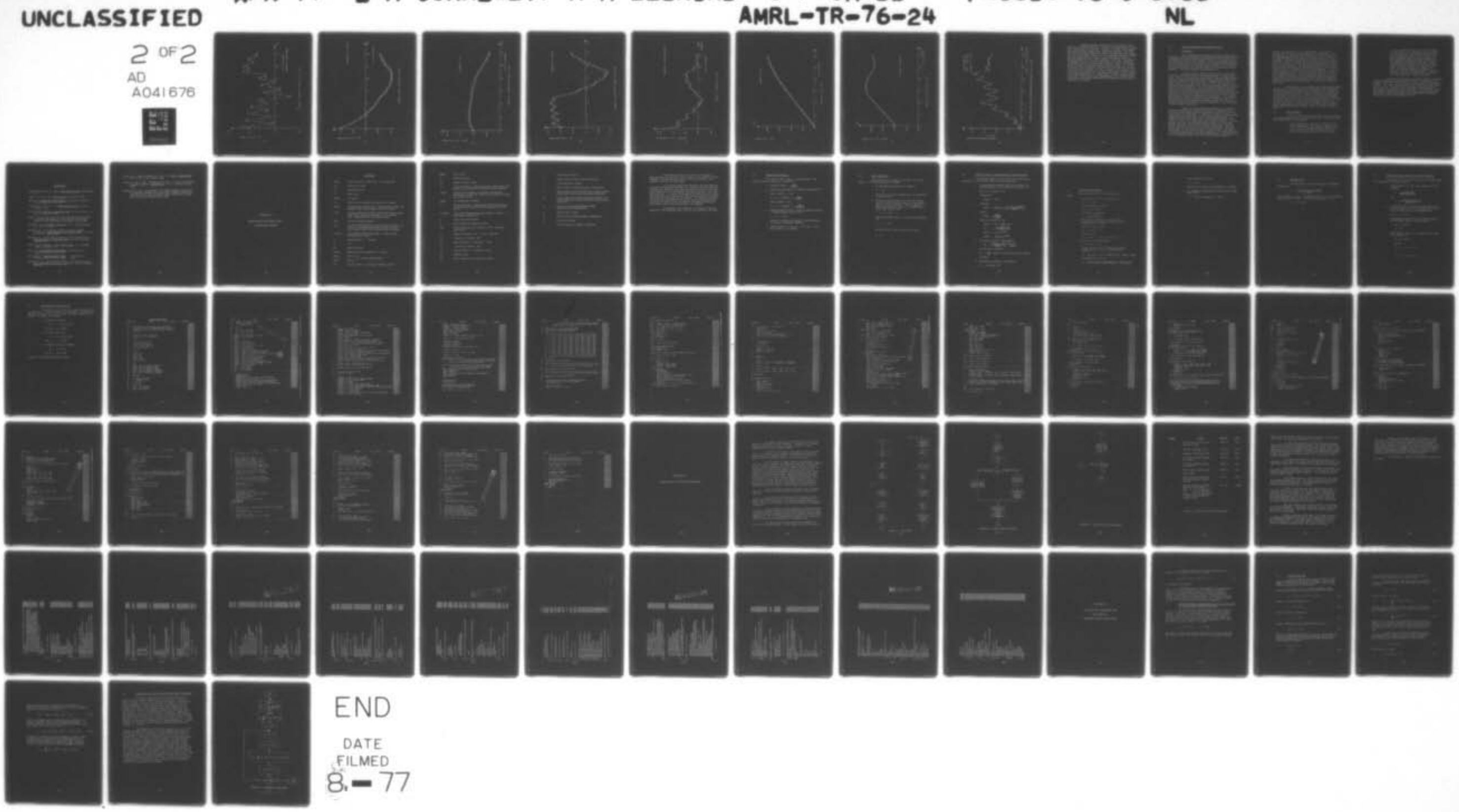


FIGURE 12 X_e vs. TIME

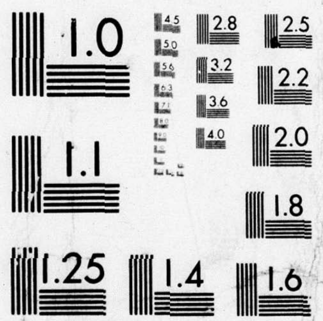
AD-A041 676 OMNEMII INC VIENNA VA F/G 12/1
DEVELOPMENT OF A CONTINUOUS PERFORMANCE MEASURE FOR MANUAL CONT--ETC(U)
APR 77 E M CONNELLY, R M ZESKIND, G P CHUBB F33615-75-C-5088

UNCLASSIFIED AMRL-TR-76-24 NL

2 OF 2
AD
A041676



END
DATE
FILMED
8-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

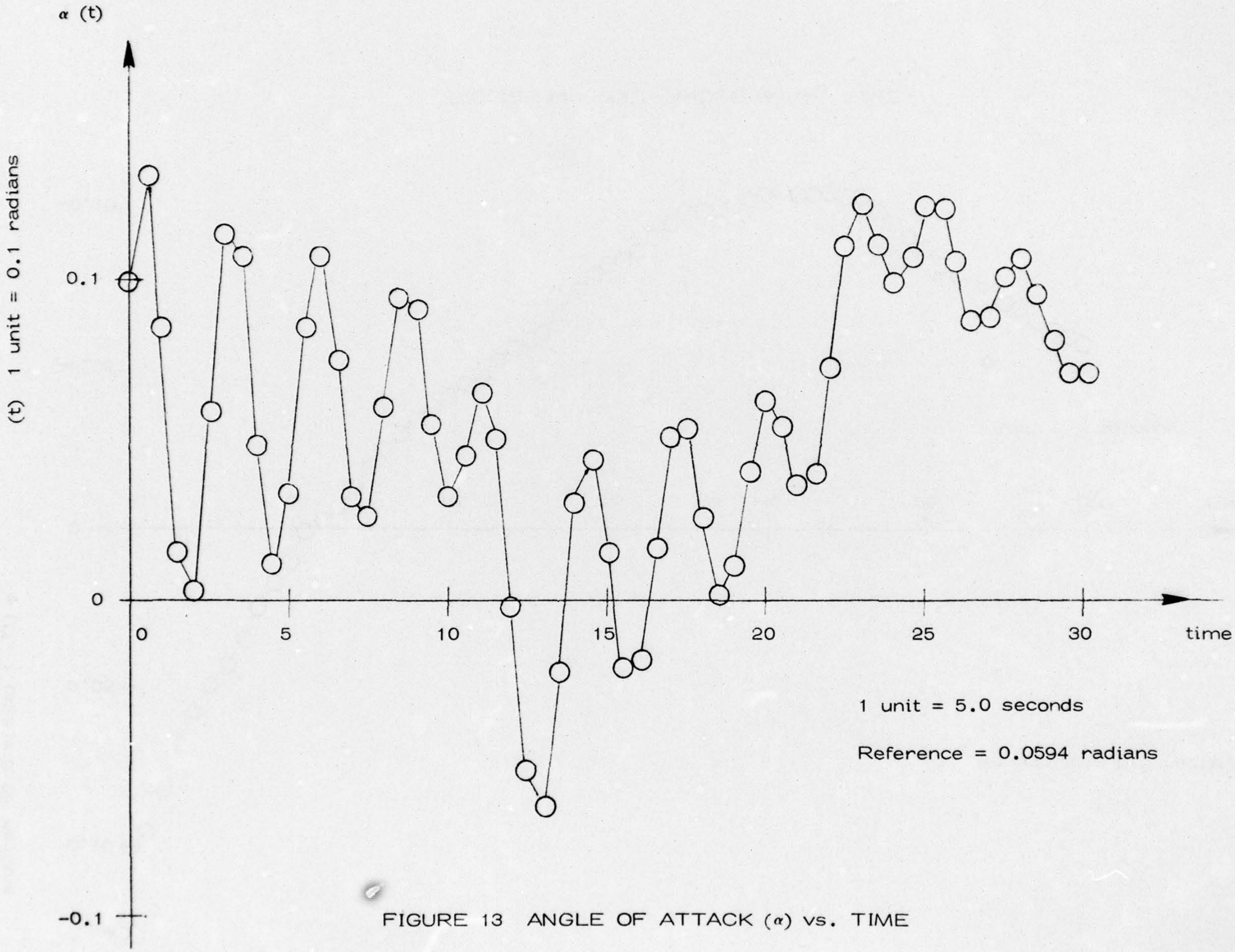


FIGURE 13 ANGLE OF ATTACK (α) vs. TIME

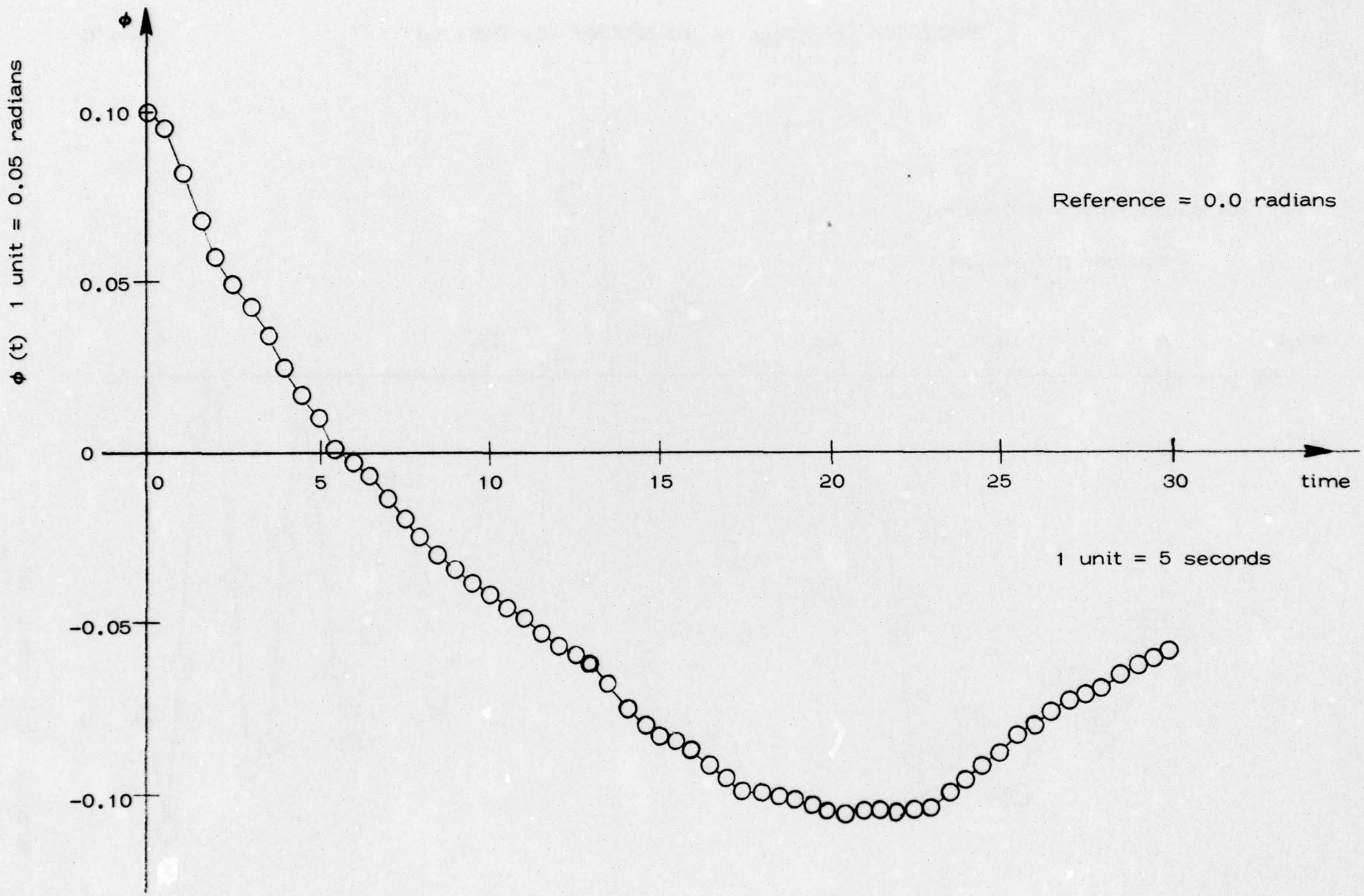


FIGURE 14 ROLL ANGLE (φ) vs. TIME

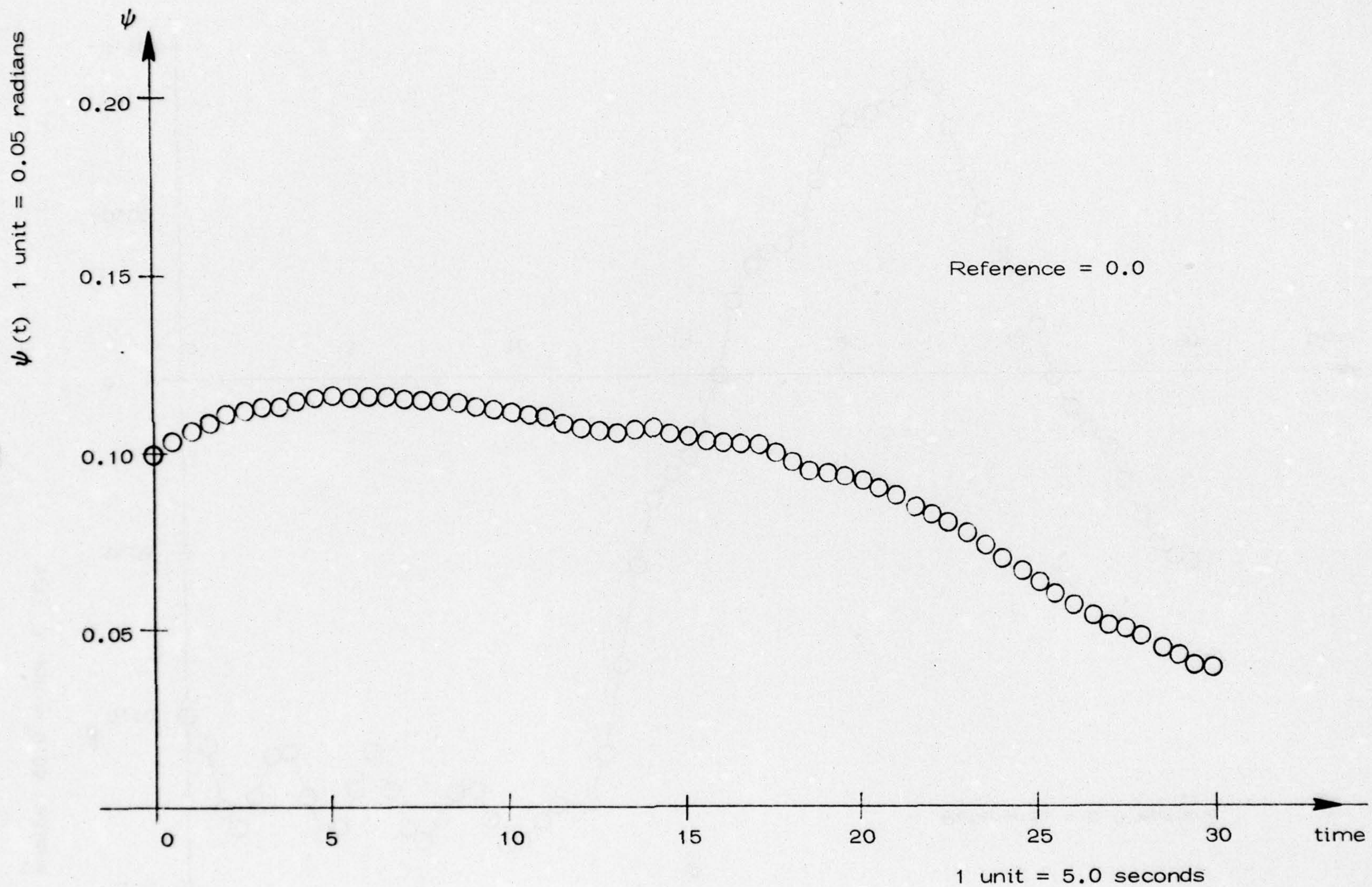


FIGURE 15 HEADING ANGLE (ψ) vs. TIME

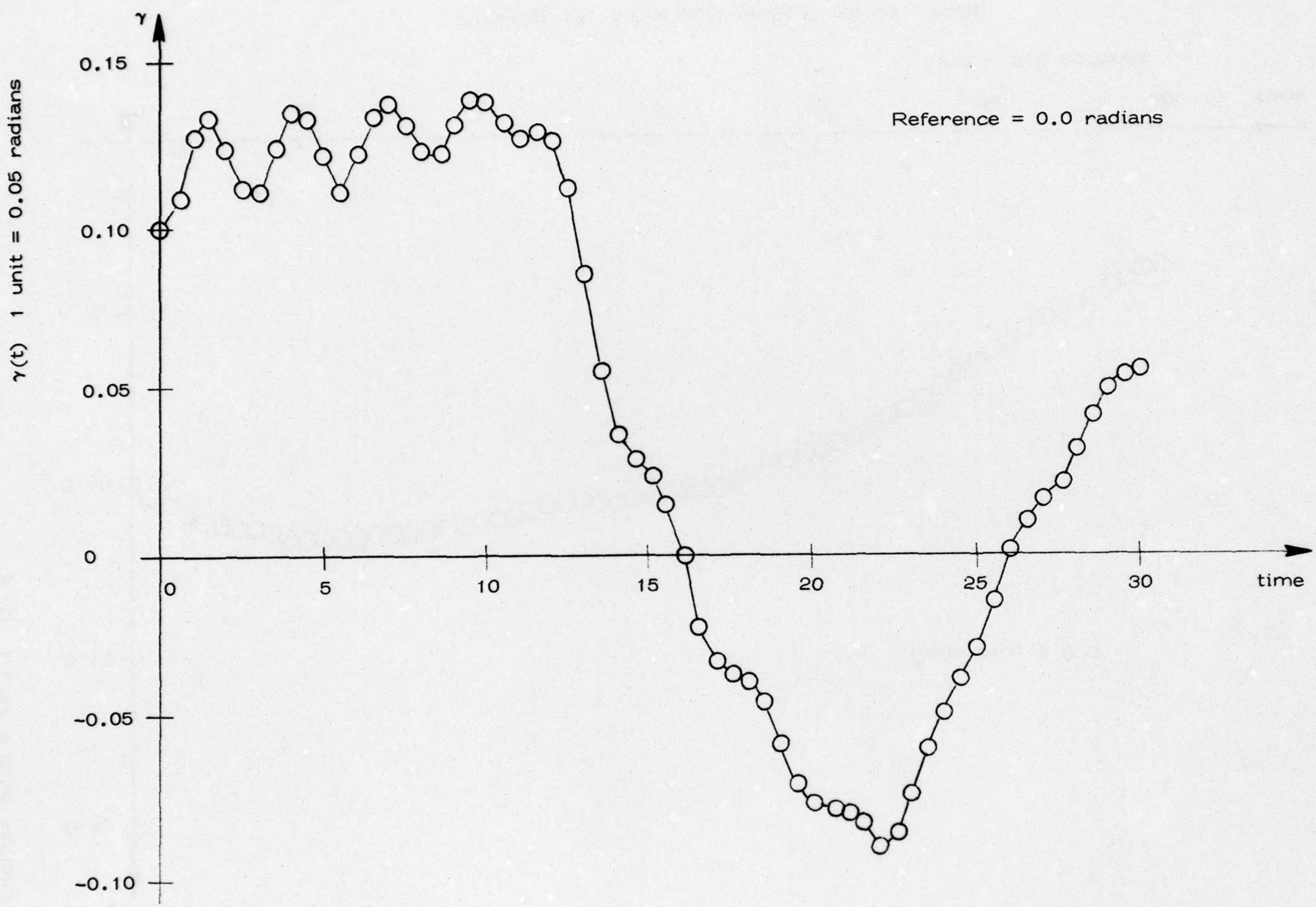


FIGURE 16 FLIGHT PATH ANGLE (γ) vs. TIME

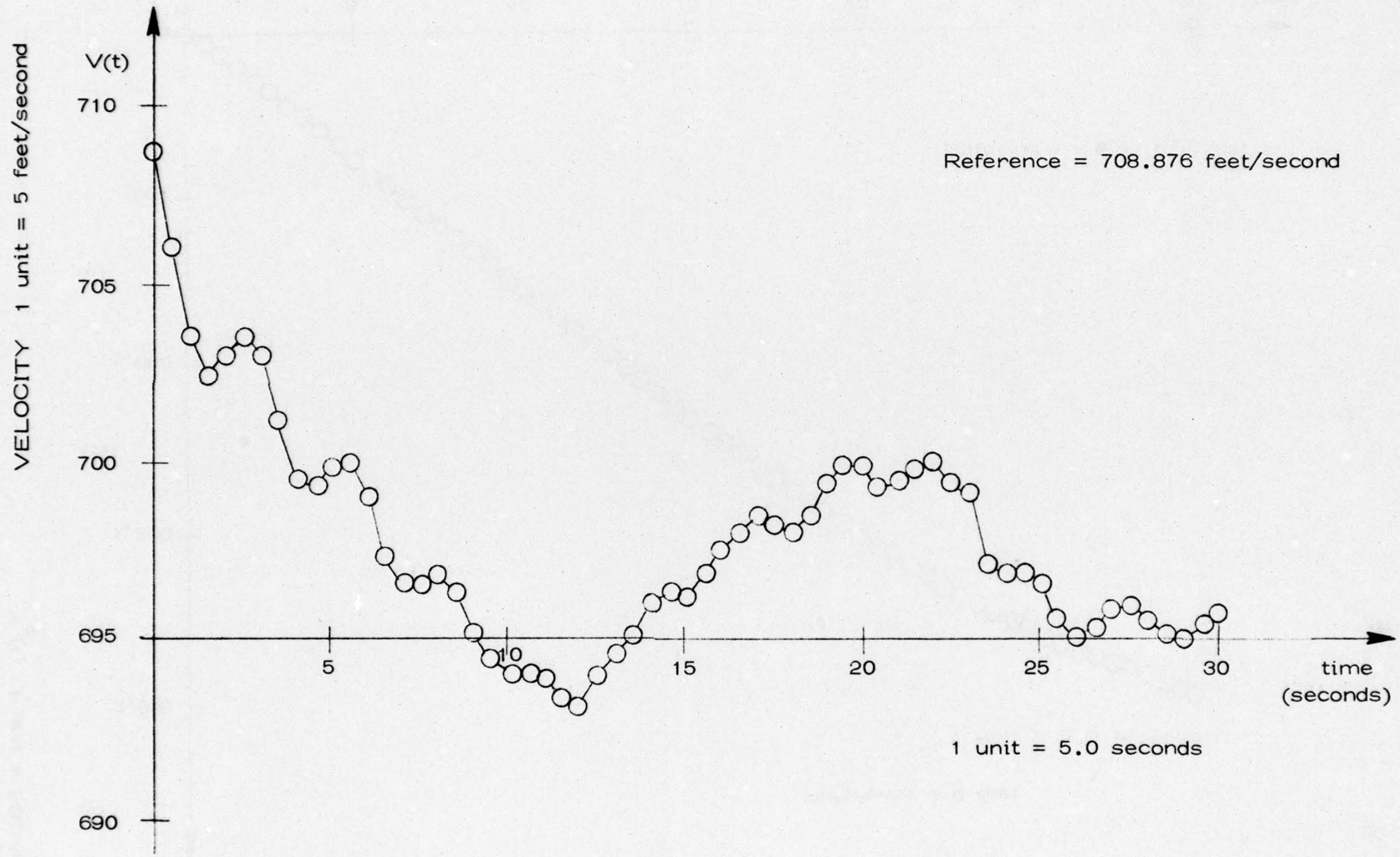


FIGURE 17   VELOCITY vs. TIME

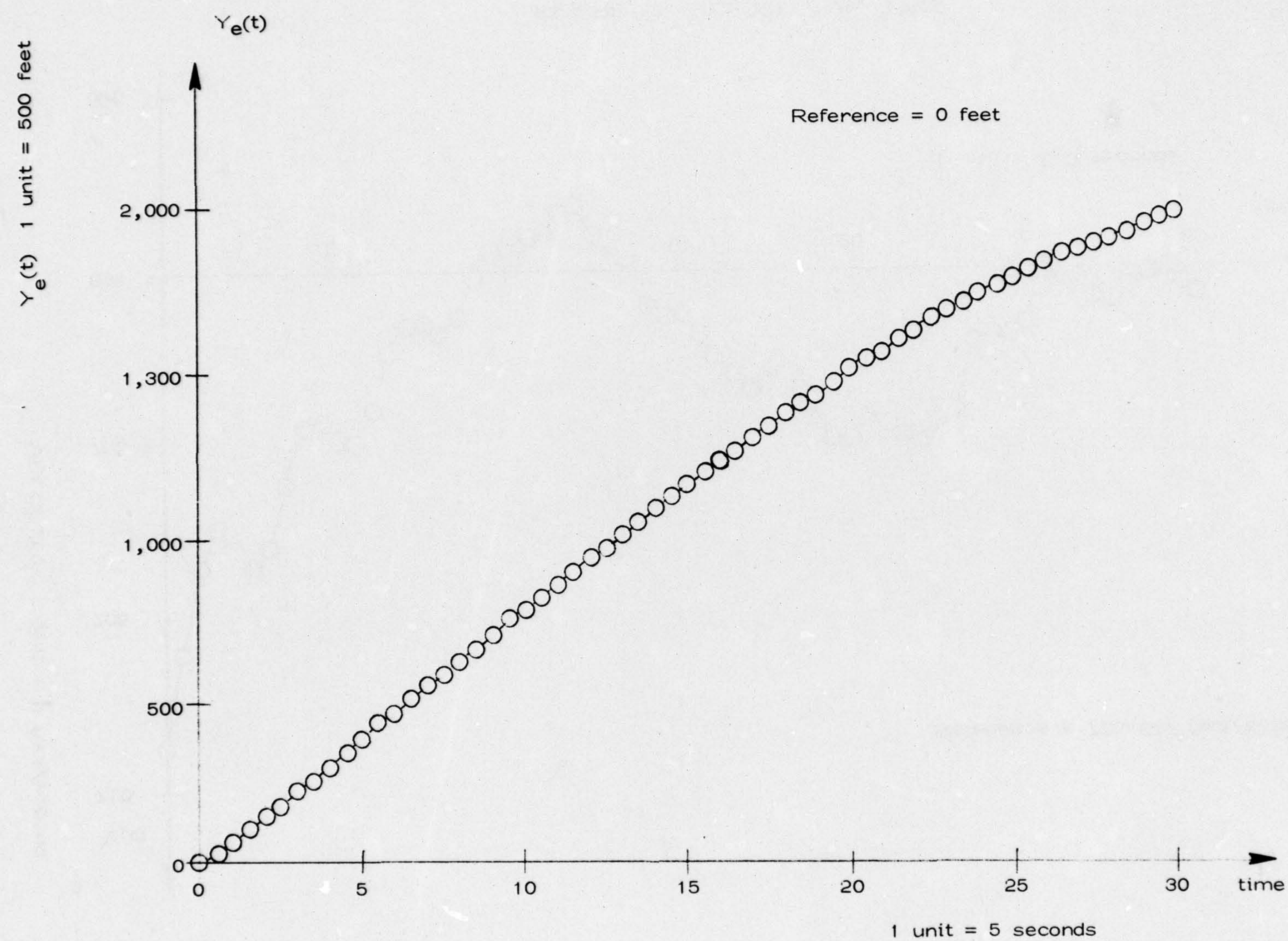


FIGURE 18 Y_e vs. TIME

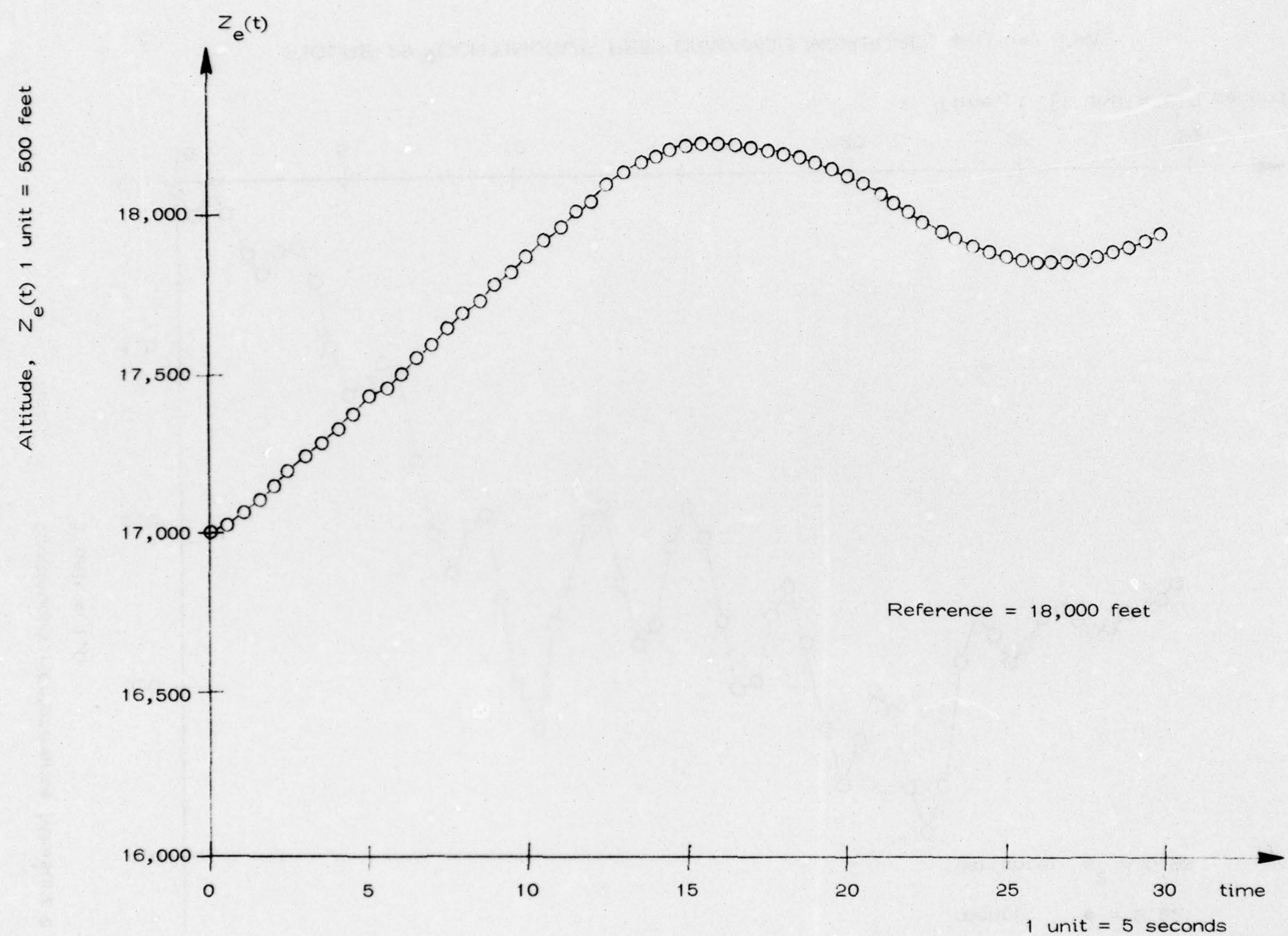


FIGURE 19 ALTITUDE Z_e vs. TIME

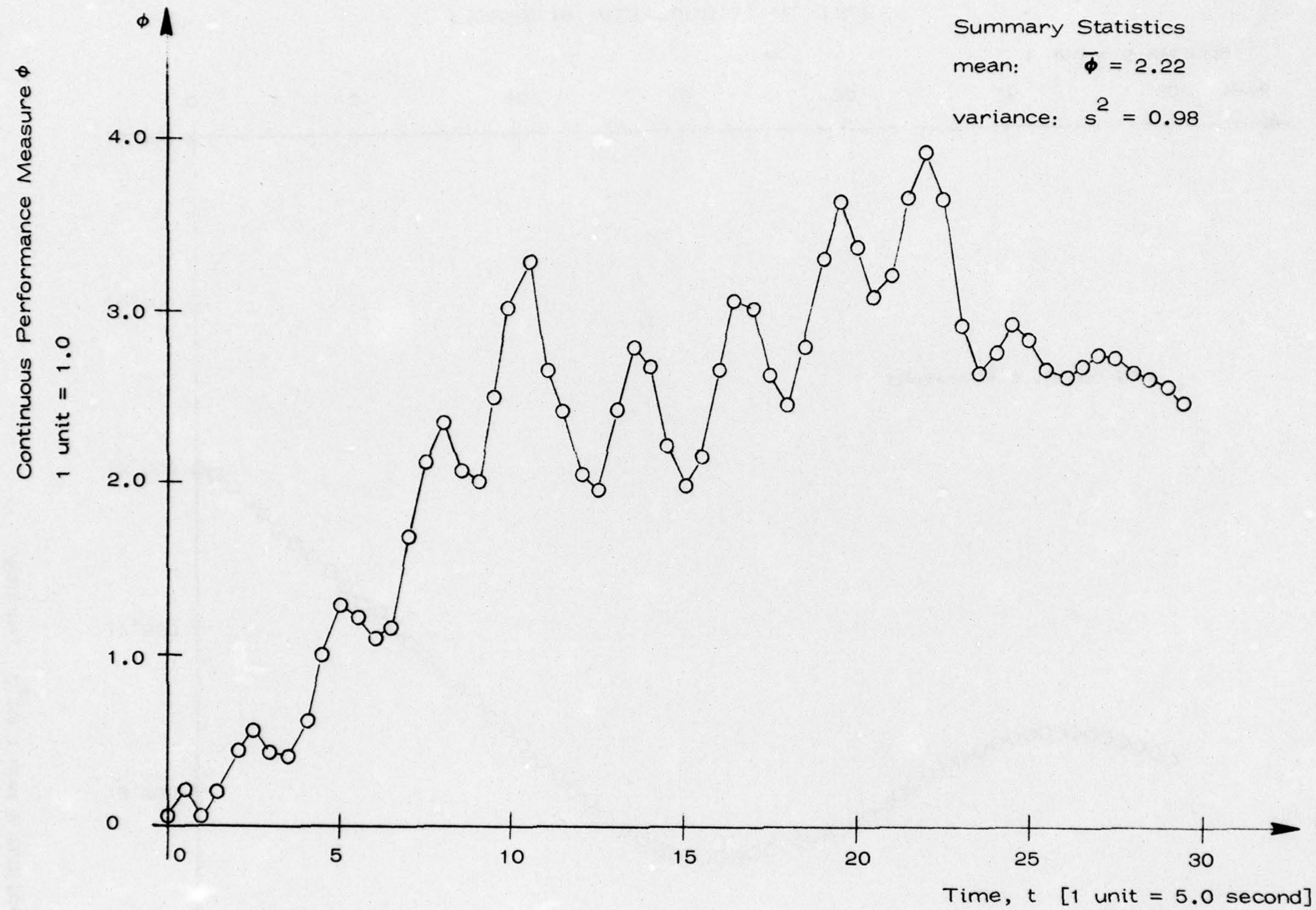


FIGURE 20 CONTINUOUS PERFORMANCE MEASURE, $\phi(t)$ vs. TIME

Inspecting Figures 12 through 19, the autopilot rapidly corrects the initial altitude error, bringing the aircraft model to 18,000 feet in approximately 15 seconds. However, in Figure 18 the altitude correction oscillates slightly about 18,000 feet reference. The autopilot rolls the aircraft model over slowly to correct the initial heading error to zero as seen in Figure 14. The autopilot's attempt to correct the initial misalignment in angle of attack and flight path angle is highly oscillatory as can be seen in Figures 13 and 16. The autopilot is attempting to bring γ to zero radians and α to 0.0594 radians but undershoots and then overshoots these reference values. The throttle control is slowly varying the velocity as seen in Figure 17. Over the 30 seconds of this simulation, the aircraft velocity was about 15 feet/second below the reference. As seen in Figure 18, the autopilot does not correct the y-position error of the aircraft model. This is because it was not designed to do so. While this is a "poor" autopilot design, it provided a good check on the reasonableness of the computer output and assured a non-optimal performance for evaluation via the CPM, as reflected in Figure 20.

6.0 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

This investigation of continuous performance measurement (CPM) (and continuous performance evaluation) shows that summary measures developed from mission segment specifications, can be converted into an instantaneous performance measure. This can be accomplished with optimal control theory by either linearizing the plant (aircraft) equations as is most frequently done in optimal control problems, or solving, at least by approximation, the optimal control for the non-linear aircraft equations. The latter approach was explored here.

Selection of the Q weighting matrix which is part of the summary performance measure can be accomplished in at least three ways. One way is to ask experienced pilots or other personnel familiar with the mission performance to select numerical values reflecting the relative importance of each flight factor. Another way is to pick Q matrix values with a simple form, say with 1's on the diagonal and 0's elsewhere, and subsequently solve for the corresponding optimal control law and aircraft trajectories. The third approach (really a variation of either of the two previous approaches) would be to reset the values of the Q matrix after examining the resulting "optimal" aircraft trajectories. Systematic trial and error adjustments to the Q matrix entries would then produce a variety of trajectories for examination. If one had pre-defined notions of what the desired trajectory should look like, the Q values might be approximated by iteratively adjusting the Q entries in directions known (from the preliminary trial-and-error runs) to produce "more desirable" trajectories.

There was not sufficient time available on the contract to thoroughly investigate the relationship between the selected Q matrix and the resultant optimal solution trajectories so that little can be concluded about that relationship. It would be of interest to identify optimal response characteristics with Q and R matrix element values. An additional area of work which was not investigated extensively is the region of validity of the approximate optimal control solution. It should be noted, however, that the approximation improves as the aircraft position approaches the reference flight path and becomes the exact solution when the aircraft is on the reference flight path. Thus, the region where the approximation holds to a given degree is the space immediately surrounding the reference flight path. The reason the control law developed becomes only approximately optimal when the aircraft is off the reference path is that the aircraft dynamics change as a function of the deviation from the reference path. If the aircraft dynamics were constant the solution would not be approximate. Thus, the preferred way to pick the aircraft dynamic (F and G) matrices and corresponding state variables would be to render matrices F and G as constant as possible as a function of the deviation from the reference flight path. However, available time did not allow a thorough investigation of the benefits to be obtained choosing various alternative structures of the aircraft equations. Also, the alternate approach to the problem would have been to linearize the representation of the aircraft model and proceed with an exact solution for the linear representation. The results could be subjected to sensitivity analysis to determine how the linearization affected the results. Further comparisons might lead to insights as to whether linearization of a non-linear phenomenon was a good or bad compromise versus the need to approximate (rather than determine exactly) the solution for the non-linear model. These are rather formidable issues and were not addressed here. They should be explored in future work.

A major problem in implementing the continuous performance measure on-line using the non-linear aircraft model is the computational load requires excessive computer time. With the approach developed to the point described in this report, the computational load is extreme and may prevent real time solution. However, it should be recognized that the computation described here solves for both the approximate optimal control law and the CPM. But the optimal control law can be precomputed and stored, since only the CPM need be implemented on-line. For example, the function K, which provides the feedback control law gains, can be represented by a pre-computed function of the state variables which might be evaluated more rapidly on-line to implement the desired CPM.

6.2 Recommendations

In order to realize the benefits available from using a continuous performance measure to evaluate manual flight control performance, the following steps are recommended:

1. The computation time required to compute the CPM should be reduced so that it can be computed in real time. This might be accomplished by approximating K as a function of the state variables and/or improving the computational efficiency of the program.

2. The error weighting of the CPM must be evaluated by examination of the continuous scoring of flights by human subjects. Note that the trajectory evaluation involving adjustment of the Q matrix is to obtain satisfactory optimal aircraft trajectories - the trajectories obtained when the aircraft motion is governed by the optimal control law. These trajectories were referred to as preferred trajectories and serve as continuous criteria for the CPM. As indicated above, the weighting of deviations from the continuous criteria - deviations that occur when the aircraft motion is governed by a human operator - must be evaluated.

As shown by the section on sensitivity analysis, the control deviation weighting is governed by the R matrix. Consequently, the credibility of the CPM rests on the values one employs in the objective function. Once again, the performance score rests upon proper selection of criteria. The goals have to be defined and agreed upon if performance measurement is to be meaningful. CPM does not resolve the problem of choosing the goals, but it does provide a performance index which is inextricably linked to the quantification of objectives, however one wishes to accomplish that task.

REFERENCES

Anderson, B.D.O. and J.B. Moore, Linear Optimal Control, Prentice-Hall, 1971.

Athans, M. and P.L. Falb, Optimal Control, McGraw-Hill, 1966.

Bellman, R., Introduction to Matrix Algebra, 2nd edition, McGraw-Hill Book Company, New York, 1970.

Bryson, A.E., Jr. and Y.C. Ho, Applied Optimal Control, Ginn and Blasedell, 1967.

Etkin, Bernard, Dynamics of Atmospheric Flight, John Wiley & Sons, Inc., New York, New York, 1972.

Fogarty, L.E. and R.M. Howe, "Computer Mechanization of Six-Degree of Freedom Flight Equations," NASA CR-1344, National Aeronautics and Space Administration, Washington, DC, May 1969.

Gantmacher, F.R., The Theory of Matrices, Vol. 1, Chelsea Publishing Co., New York, 1959.

Kleinman, D.L., "On an Iterative Technique for Riccati Equation Computations," IEEE Transactions on Automatic Control, Vol. AC-13, pp. 114-115, February 1968.

Kleinman, D.L., "An Easy Way to Stabilize a Linear Constant System," IEEE Transactions on Automatic Control, Vol. AC-15, pp. 692, December 1970.

Padulo, Louis and Michael A. Arbib, System Theory, W. B. Saunders Company, Philadelphia, 1974.

Pipes, L.A., Matrix Methods for Engineering, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.

Sage, Andrew P., Optimum Systems Control, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, Chapter 4, 1968.

Sandell, N.R., Jr., "On Newton's Method for Riccati Equation Solution," IEEE Transactions on Automatic Control, Vol. AC-19, pp. 254-255, June 1974.

Smith, R.A., "Matrix Equation XA + BX = C," SIAM J. Applied Math, Vol. 16, No. 1, pp. 198-201, 1968.

Wernli, A. and G. Cook, "Suboptimal Control for the Nonlinear Quadratic Regulator Problem," AUTOMATICA: Vol. II, January 1975, pp. 75-84.

Zeskind, R.M. and V. Vimolvanich, "An Optimal Feedback Control Law for Regulator Problems with Linear State Inequality Constraints," Presented at 1973 Joint Automatic Control Conference, The Ohio State University, Columbus, Ohio, 1973.

APPENDIX A

AERODYNAMIC EQUATIONS FROM SUBROUTINE ADCOMP

GLOSSARY

AREA	Area of aircraft in square feet - set to 202 $(ft)^2$
CD	Coefficient of drag
CL	Coefficient of lift
D	Drag in lbs.
DENS	Air density
DT	Frame time in seconds (or fractions of a second)
FEBA	Forward Edge of Battle Area, a hypothetical boundary line separating safe and potentially hostile territories
FLIR	Forward Looking Infra Red, a sensor often used at night when visible light is not available; a very sensitive "heat" related detection and display system
GCA	Ground Controlled Approach
GCI	Ground Controlled Intercept, radar operator "vectors" an aircraft to some desired location by telling the pilot what altitude, heading and airspeed should be attained
HF/SSB	High Frequency/Single Side Band - a type of radio transmitter and receiver
K	Scaling factor in α equation
L	Lift
M_1	Mach of aircraft
MASS	Weight of aircraft divided by G - SLUGS
MAXG	Set to +15 } Aircraft scaling factors
MING	Set to -5 } Aircraft scaling factors
MT	Maximum thrust - a function of altitude, velocity

PRES	Air Pressure
Q	Dynamic pressure
SS	Speed of sound in ft/sec
TA	Terrain Avoidance - altering course to avoid terrain when altitude to be maintained is lower than terrain features
TACAN	Tactical Air Navigation, a navigation aid originally developed by and for the military but now used in commercial aviation as well
TEMP	Air temperature in degrees
TF	Terrain Following - staying close to the ground, diving and climbing but maintaining same heading (in contrast to TA)
UHF/ADF	Ultra High Frequency/Automatic Direction Finding, a radio based navigation aid
V	Velocity of aircraft ft/sec
$\dot{V}$	Rate of change of velocity in ft/sec
VFR	Visual Flight Rules (in contrast to IFR - instrument flight rules)
W	Weight of aircraft in lbs. - set to 17,000 lbs.
X_e	X position of aircraft - feet
$\dot{X}_e$	Rate of change in X coordinate - ft/sec
Y_e	Y position of aircraft - feet
$\dot{Y}_e$	Rate of change in Y coordinate - ft/sec
Z_e	Altitude in feet
$\dot{Z}_e$	Rate of change in Z_e coordinate in ft/sec

α	Attack angle in radians
$\dot{\alpha}$	Rate of change of attack angle in radians/sec
γ	Flight path angle - radians
$\dot{\gamma}$	Rate of change of flight path angle in radians/sec
μ_1	Input in radians/sec, controls the rate of change of attack (proportional to fore and aft movements of the stick)
μ_2	Input in radians/sec, controls the rate of change of roll angle (proportional to lateral or side-to-side movements of the stick)
μ_3	Input as normalized percentage of throttle (proportional to throttle position)
ψ	Heading angle - radians
$\dot{\psi}$	Rate of change of heading angle in radians/sec
ϕ	Roll angle in radians
$\dot{\phi}$	Rate of change of roll angle in radians/sec

This appendix presents a summary of the model of flight dynamics that was used in this contract. The computer program ADCOMP, which provides the coding for the flight dynamics, was supplied to Omnemii by AMRL/HEB.

The following aerodynamic equations were extracted from the FORTRAN subroutine ADCOMP, ALL DIGITAL COCKPIT DISPLAY SYSTEM, and represent those equations which determine the simulated flight characteristics for the display presented to subjects in the real-time simulation. The subroutine computes all flight values from initial values and from three analog inputs representing control stick positions and throttle setting executed by the subject. The ADCOMP subroutine is tied to other subroutines which monitor subject performance (read and record the analog inputs) and generate the displays themselves. These other routines are not described in this appendix.

The following set of equations is a listing in order of execution of the computation steps involved in the subroutine ADCOMP.

1.0 ANGLE OF ATTACK, α

The angle of attack, alpha, is computed first. The following steps are executed in computing α.

1. Compute GLOAD $= \dfrac{L}{\text{Weight}}$

2. Compute the derivative of α based on GLOAD and μ_1.

   When GLOAD $\geq$ 1, set

   $$\dot{\alpha} = \mu_1 - (\text{GLOAD} - 1) \frac{K}{\text{MAXG}}$$

   When GLOAD < 1 set

   $$\dot{\alpha} = \mu_1 + (\text{GLOAD} - 1) \frac{K}{(\text{MING})\,5}$$

3. Compute present value of α from the previous value of α and the present value of $\dot{\alpha}$ as:

   $$\alpha = \alpha + \dot{\alpha}\,DT$$

   where DT is the time increment. This constitutes a numerical rectangular integration.

4. Test the value of α, and if $\alpha < -.2$, set $\alpha = -0.2$. This limits α at $-.2$ radians.

2.0 ROLL ANGLE, ϕ

Next, the program computes a new value for the roll angle, ϕ. The following steps are executed.

1. Set the negative derivative of roll angle to

   $$\dot{\beta} = U_2$$

   where U_2 is a control variable which is proportional to stick position.

2. Compute the present value of ϕ from the previous value of β and the present value of $\dot{\beta}$ using rectangular numerical integration. When the absolute value of the flight path angle, γ, is greater than or equal to $\frac{\pi}{2}$ radians, then

   $$\beta = \beta + \dot{\beta}\,DT + \pi$$

   When the absolute value of γ is less than $\frac{\pi}{2}$ radians, set

   $$\beta = \beta + \dot{\beta}\,DT$$

   Now convert the β values to positive roll angle:

   $$\phi = -\beta$$

3.0 COMPUTATION OF ENVIRONMENTAL PARAMETERS

The ADCOMP subroutine next computes the environmental parameters for the aircraft model in the following steps:

1. The parameters associated with the atmosphere are calculated based on the altitude of the aircraft, Z_e.

   When $Z_e \geq 35{,}300$ ft., set

   Temperature:

   $$TEMP = -67.0$$

   Pressure:

   $$PRES = 489.456 \exp\left(\frac{-(Z_e - 35{,}300.)}{20930.}\right)$$

   Density:

   $$DENS = \frac{PRES}{673946.}$$

   When $Z_e < 35{,}300$ ft., set

   $$TEMP = 59 - (.00357\ Z_e)$$

   $$D_1 = 1 - \left(\frac{0.00357\ Z_e}{518.4}\right)$$

   $$PRES = 2116\ D_1^{5.256}$$

   $$DENS = .002378\ D_1^{4.256}$$

2. The speed of sound is calculated as:

   $$SS = \left(\frac{PRES\ 1.406}{DENS}\right)^{.5} \left(\frac{ft.}{sec.}\right)$$

   and the MACH number as

   $M_1 = \frac{V}{SS}$, where V is the velocity of the aircraft in ft/sec.

3. The dynamic pressure is calculated as:

   $$Q = 0.5\ DENS\ (V)^2$$

4.0 CALCULATION OF FORCES

The forces acting on the aircraft are calculated as follows:

1. The coefficient of LIFT is first calculated by:

   $CL = .1 + 2.5\,\alpha$

   Then the coefficient of DRAG by:

   $CD = 0.03 + .27(CL)^2$

   The DRAG is computed from

   $D = (Q)\,(CD)\,(AREA)$

2. After the first computed value of CL is used to compute CD and D, CL and α are modified according to the value of α, as follows:

   When $\alpha \geq .4$ and $\alpha < .6$, set

   $CL = 1 - 2.\,(\alpha - .4)$

   When $\alpha \geq .6$, set

   $CL = 0$ and $\alpha = .6$; if $V < 100$, set $\alpha = 0$.

3. The thrust is computed as follows:

   If $V < 1$, set $V = 1.0$

   Compute maximum thrust, MT, depending on whether the after-burner is on or off. When after-burner is on, set

   $$MT = 2\left[\left(2327. + .172Z_e - .0000031(Z_e)^2\right)M_1 + (11500. - .25Z_e)\right]$$

   When after-burner is off, set

   $$MT = \frac{\left((2327. + .172Z_e - .0000031(Z_e)^2\right)M_1}{2} + \frac{(11500. - .25Z_e)}{2}$$

Next compute thrust from:

$$T = \mu_3 \, MT$$

where μ_3 is a control input proportional to throttle.

4. The component of applied force normal to the flight path (Lift) is:

$$L = ((Q) (CL) (AREA)) + T \sin(\alpha)$$

5.0 VELOCITY, V

The derivative of the aircraft velocity is computed according to:

$$\dot{V} = \frac{T \cos(\alpha) - D - W \sin(\gamma)}{MASS}$$

where MASS is in slugs. The present value of V is next computed from the previous value of V and the present value of $\dot{V}$ as:

$$V = V + \dot{V}\,DT$$

6.0 HEADING ANGLE ψ AND FLIGHT PATH ANGLE γ

The present value of heading angle ψ and flight path angle γ are computed in the following steps:

1. The derivatives of ψ and γ are computed from the equations

$$\dot{\psi} = \frac{L \sin(\phi)}{MASS \cos(\gamma)\ V}$$

and

$$\dot{\gamma} = \frac{L \cos(\phi) - W \cos(\gamma)}{(MASS)\ V}$$

2. The present value of ψ and γ are computed from the previous values of ψ and γ the present values of $\dot{\psi}$ and $\dot{\gamma}$, but with limits in the following way:

Set $\gamma_1 = \gamma + \dot{\gamma}\ DT$

If the absolute value of γ_1 is less than $\frac{\pi}{2}$ radians, set the present values of ψ and γ equal to

$$\psi = \psi + \dot{\psi}\ DT$$

$$\gamma = \gamma_1$$

If the absolute value of γ_1 is greater than or equal to $\frac{\pi}{2}$ radians, set

$$\psi = \psi + \dot{\psi}\ DT + \pi$$

and set

$$\gamma = -\pi - \gamma, \text{ if } \gamma_1 < 0$$

or

$$\gamma = \pi - \gamma_1, \text{ if } \gamma_1 \geq 0$$

7.0 POSITION OF THE AIRCRAFT

The position of the aircraft with respect to the earth is calculated next. In each step below, the derivative is calculated first and then the present value is found from rectangular numerical integration. The steps are as follows:

1. X-position of aircraft

$$\dot{X}_e = V \cos(\gamma) \cos(\psi)$$

and $X_e = X_e + \dot{X}_e DT$

2. Y-position of aircraft

$$\dot{Y}_e = V \cos(\gamma) \sin(\psi)$$

and $Y_e = Y_e + \dot{Y}_e DT$

3. Z-position of aircraft (altitude)

$$\dot{Z}_e = V \sin(\gamma)$$

and $Z_e = Z_e + \dot{Z}_e DT$

A listing of the ADCOMP subroutine follows.

ADCOMP SUBROUTINE

```
/ G LEVEL  20                    MAIN             DATE = 74275             16/11/33

      C
      C
      C         THIS SUBROUTINE COMPUTES ALL OF THE MATH USED
      C         TO GENERATE NEW XY COORDINATE POSITIONS OF ALL
      C         THE DISPLAYS THAT ARE GENERATED DURING A SIMULATED
      C         MISSION .
      C
      C
      C         EQUATION OF MOTION TAKEN FROM
      C         DOMASH  AIRPLANE AERODYNAMICS .
      C
      C         ANGLES
      C
      C         PHI = ROLL ANGLE (BANK)
      C         THETA = PITCH ANGLE
      C         ALPHA = ANGLE OF ATTACK
      C         GAMMA = FLIGHT PATH ANGLE
      C         PSI = HEADING ANGLE
      C
      C         FORCES
      C
      C         THRUST - LBS
      C         DRAG - LBS
      C         LIFT - LBS
      C         WEIGHT - LBS
      C         MASS - SLUGS
      C         G = 32.2 FT/SEC2
      C
      C         RATE
      C
      C         VELDT - RATE OF CHANGE OF VELOCITY
      C         PSIDT - RATE OF CHANGE OF PSI
      C         GAMDT - RATE OF CHANGE OF GAMMA
      C         DELX - RATE OF CHANGE OF X POSITION
      C         DELY - RATE OF CHANGE OF Y POSITION
      C         DELZ - RATE OF CHANGE OF Z POSITION
      C
      C         POSITION
      C
      C         X - PLANAR POSITION
      C         Y - PLANAR POSITION
      C         Z - ALTITUDE
      C
      C         ATMOSPHERE
      C
      C         TEMP - AIR TEMPERATURE
      C         PRES - AIR PRESSURE
      C         DENS - AIR DENSITY
```

```
C      SOUND - SP OF SOUND (FT/SEC)
C      MACH - OF AIRCRAFT
C      Q - DYNAMIC PRESSURE
C
C      AERO
C
C      CL = LIFT COEFFICIENT
C      CD = DRAG COEFFICIENT
C
C    BANKN  = NEW BANK ANGLE
C    BANKNR = NEW BANK ANGLE IN RADIANS
C    BANKO  = OLD BANK ANGLE
C    CALD   =
C    CALV   =
C    FRTIME = FRAME TIME
C    HCTRX  = HORIZON CENTER X
C    HCTRY  = HORIZON CENTER Y
C    HEADCR = HEADING CHANGE IN RADIANS
C    HEADC  = HEADING CHANGE
C    HDRATE = HEADING RATE
C    ISPEED = SPEED OF AIRCRAFT
C    NAVG   = AREA NAVIGATIONAL DATA IS STORED
C    INCT   = NUMBER OF NAVIGATIONAL POINTS              (TFOR1)
C    INCNT  = NUMBER OF NAVIGATIONAL POINTS TO DISPLAY   (TFOR2)
C    ITCT   = NUMBER OF TARGETS                          (TFOR1)
C    ITCNT  = NUMBER OF TARGETS TO DISPLAY               (TFOR2)
C    STHEAD = START HEADING OF AIRCRAFT
C    SIZET  = SIZE OF TEWS DISPLAY
C    SIZEN  = SIZE OF NAVIGATIONAL DISPLAY
C    TBANKN = TANGENT OF NEW BANK ANGLE IN RADIANS
C    TARG   = AREA TARGET DATA IS STORED
C    TRGHDN = NEW TRIG HEADING
C    XAC    = X OF AIRCRAFT
C    XT     = X OF TARGET
C    YAC    = Y OF AIRCRAFT
C    YT     = Y OF TARGET
C
C
      SUBROUTINE  ADCOMP
      REAL MAXG,MING
      COMMON INCT, ITCT, INALIN(6), SPEED,SPEEDK,ISPD
      COMMON ICTLP, ITCNT, INCNT
      COMMON IDTX(100),IDTY(100),IDTYPE(100),IDTCHA(100)
      COMMON IDNX(20),IDNY(20),INCHAR(20),INCTYP(20)
      COMMON XRA,YRA,XLA,YLA,ITIMER,LMPDX,LMPDY,RMPDX,RMPDY
      COMMON RPMY1, RPMY2, RPM1, RPM2
      COMMON  CX(6)
      COMMON COMPX, COMPY
```

```
      COMMON RPMY3, RPMY4
      COMMON     XX(13), YY(13)
      COMMON  IAS(6),  ISIAS(5)
      COMMON   KALT, KVV, KGAM, KMACH, KGLOD
      COMMON  KALF, KXAC, KYAC
      COMMON MACHXY(6),MACHTX(5), IVRAST, IVDUM
      COMMON  DIGIN1(10)
      COMMON  ISWT8
      COMMON IALHXY(6),  IALTXH(5), IALTXY(6),  IALTXT(5)
      COMMON IACCY(6), IACCTX(5), IADAY(6), IADATX(5)
      COMMON  ISWTC2, ISWTC3, ISWTC4, ISWTC5, ISWTC6, ISWTC7
      COMMON ALPHA, BETA, THETA, TRGHDN, VELDT, DELZ, GLOAD
      COMMON ALT, GDSPD, IAFBFN
      COMMON IPRNTA(50)
      COMMON  IHUTSW, NCSW, ISTART, IRESET, ICNCAR
      COMMON  ICNTAB(552), IMSEQ(100)
      COMMON TX(100),TY(100),TON(100),TOF(100),ITYPE(100),ICHAR(100)
      COMMON NX(20), NY(20), SLOP(80), NCTY(20)
      COMMON / CARD/ISPEED, FRTIME, XAC, YAC, STHEAD, SIZET, SIZEN,
     1 CALV, CALD, BETRMX, ALFARX, SCALEM
      COMMON / BANK / BANK0, BANKN
      COMMON /RXUYV/ X21, U21, Y21, V21, XN21,UN21, YN21, VN21
      COMMON /RASTXY/ X2, X1, U2, U1, Y2, Y1, V2, V1
      COMMON /RASNXY/ XN2, XN1, UN2, UN1, YN2, YN1, VN2, VN1
C
      COMMON / LYONS / PHI, GAMMA, PSI, G,
     1 SINPHI, COSPHI, SINGAM, COSGAM, APX, APY, Z
C
      COMMON /ARTHOR/ R, PI, PI2, CONST, CONST2
C
C
      REAL NX, NY, NON, NOF, KALF
      REAL LIFT /17000./
C
C
      INTEGER * 2  IMSEQ
      INTEGER * 2 XRA, YRA, XLA, YLA,INCTYP,NCTY
      INTEGER * 2  IAS, IVRAST, IVDUM
      INTEGER * 2 MACHXY
      INTEGER * 2 DIGIN1
      INTEGER * 2  LMPDX, LMPDY, RMPDX, RMPDY
      INTEGER * 2  RPMY1, RPMY2, RPM1, RPM2,RPMY3, RPMY4
      INTEGER * 2 IDTX, IDTY, IDTYPE, IDTCHA, IDNX, IDNY, INCHAR,INALIN
      INTEGER * 2  COMPX, COMPY, CX
      INTEGER * 2 IALTXY, IALHXY
      INTEGER * 2  IACCY, IADAY
      INTEGER  * 2  ISWTC2, ISWTC3, ISWTC4, ISWTC5, ISWTC6, ISWTC7
      INTEGER * 4 HNDX, GNDX, ANDX
```

```
      INTEGER  HI/1450/, LO/300/
      INTEGER  HIUR/3500/, LOUR/2350/
      INTEGER * 2  IHAF(2), DGP
      INTEGER * 2  CHAR(2), BLANK/'  '/
      EQUIVALENCE  (ICNCAR, CHAR(1))
      DATA  IBC1/0/, IBC2/0/
      DATA  ICARFG/0/
      LOGICAL * 1  L1(2)
      EQUIVALENCE  (IFULL, IHAF(1)), (IHAF(2), L1(1))

      DIMENSION  CDEL(5), SDEL(5), DELTA(5)

      DIMENSION  IMACH(21)
      DIMENSION  ISPDS(21)
      DIMENSION  IALT1(136)
      DIMENSION  IALT(23)
    DIMENSION STATEMENT FOR ACCELEROMETER DATA

      DIMENSION  IACC(22)

    DIMENSION STATEMENT FOR ANGLE OF ATTACK

      DIMENSION  IAOA(15)

      DATA DRAG/0./, THRUST/0./
      DATA  ISPDS /'  0 ',' 100',' 200',' 300',' 400',' 500',' 600',
     1      ' 700',' 800',' 900','1000','1100','1200','1300','1400',
     2      '1500','1600','1700','1800','1900','2000'/

      DATA IMACH/' 0.0',' 0.1',' 0.2',' 0.3',' 0.4',' 0.5',' 0.6',
     1      ' 0.7',' 0.8',' 0.9',' 1.0',' 1.1',' 1.2',' 1.3',
     2      ' 1.4',' 1.5',' 1.6',' 1.7',' 1.8',' 1.9',' 2.0'/
      DATA  KRPM/0/, NRPM/0/, IFLIP/1/, INCR/1/
      DATA  RADIUS/44.0/
      DATA  KOUNT/1/
      DATA  CLO/0.1/,CLSLO/2.5/, AREA/202./, WEIGHT/17000./
      DATA  VEL/1000./
      DATA IBET/1/


      DATA MAXG/15.0/
      DATA MING/-5.0/

    FOLLOWING DATA IS FOR GENERATING A TAPE
    TYPE DISPLAY REPRESENTING AN ALTIMETER
    WITH A SCALE IN INCREMENTS OF 100

      DATA  IALT/' 200',' 100','  0 ',
```

```
     *         ' 900',' 800',' 700',' 600',' 500',' 400',' 300',
     1      ' 200',' 100','  0 ',' 900',' 800',' 700',' 600',' 500',
     2      ' 400',' 300',' 200',' 100','  0 '/
C
C     FOLLOWING DATA IS FOR GENERATING A TAPE
C     TYPE DISPLAY REPRESENTING AN ALTIMETER
C     WITH A SCALE IN INCREMENTS OF 1000
C
      DATA  IALT1/' 120',' 119',' 118',' 117',' 116',' 115','114',
     *    ' 113',' 112',' 111',' 110',' 109',' 108',' 107',' 106',
     1    ' 105',' 104',' 103',' 102',' 101',' 100','  99','  98',
     2    '  97','  96','  95','  94','  93','  92','  91','  90',
     3    '  89','  88','  87','  86','  85','  84','  83','  82',
     4    '  81','  80','  79','  78','  77','  76','  75','  74',
     5    '  73','  72','  71','  70','  69','  68','  67','  66',
     6    '  65','  64','  63','  62','  61','  60','  59','  58',
     7    '  57','  56','  55','  54','  53','  52','  51','  50',
     8    '  49','  48','  47','  46','  45','  44','  43','  42',
     9    '  41','  40','  39','  38','  37','  36','  35','  34',
     A    '  33','  32','  31','  30','  29','  28','  27','  26',
     B    '  25','  24','  23','  22','  21','  20','  19','  18',
     C    '  17','  16','  15','  14','  13','  12','  11','  10',
     D    '  9 ','  8 ','  7 ','  6 ','  5 ','  4 ','  3 ','  2 ',
     E    '  1 ','  0 ',' -1 ',' -2 ',' -3 ',' -4 ',' -5 ',' -6 ',
     F    ' -7 ',' -8 ',' -9 ','-10 ','-11 ','-12 ','-13 ','-14 ',
     G    '-15 '/
C
C     ACCELEROMETER DATA STATEMENT
C
      DATA  IACC/' 15 ',' 14 ',' 13 ',' 12 ',' 11 ',' 10 ','  9 ',
     1          '  8 ','  7 ','  6 ','  5 ','  4 ','  3 ','  2 ','  1 ',
     2          '  0 ',' -1 ',' -2 ',' -3 ',' -4 ',' -5 ',' -6 '/
C
C     DATA STATEMENT FOR ANGLE OF ATTACK
C
      DATA  IAOA/' 45 ',' 40 ',' 35 ',' 30 ',' 25 ',' 20 ',' 15 ',
     1          ' 10 ','  5 ','  0 ',' -5 ','-10 ','-15 ','-20 ','-25 '/
C
C ** COMPUTE VOLTS FROM ANALOG INPUT **
C
C
C      FOLLOWING DATA STATEMENT IS FOR USE IN SCALING
C      THE XY COORDINATES OF ARTIFICIAL HORIZON
       DATA AHM6 /-6.0/
C
       VOLTS = INALIN(1)
       VOLTS = VOLTS * 5.0 / 32768.
C
```

```
C ** COMPUTE ROLL RATE  **
C
CC     ROLRAT = ( VOLTS - CALV ) * CALD
CC     BANKN  = BANKO  + ( ROLRAT * FRTIME )
CC     IF ( BANKN .GT. 180.0) GO TO 40
CC     IF ( BANKN .GT. 75.0 ) GO TO 50
CC     GO TO 60
CC 40 IF ( BANKN  .LT. 285. ) GO TO 45
CC     GO TO 60
CC 45 BANKN = 285.
CC     GO TO 60
CC 50 BANKN  = 75.0
CC 60 CONTINUE
C
C*****  COMPUTE  *****
C
CC     BANKN = AMOD( BANKN,360.)
C
CC     IF ( BANKN ) 61,70,70
CC 61 BANKN = BANKN + 360.
CC 70 CONTINUE
CC     BANKO = BANKN
C
C**********  COMPUTE ARTIFICIAL HORIZON  **********
C
       GO TO ( 80, 82 ), KOUNT
C
   80 KOUNT = 2
C
       DO 81 I = 1,5
       DEL = I
       DELTA(I) = DEL * CONST2
       CDEL(I) = COS(DELTA(I))
   81 SDEL(I) = SIN(DELTA(I))
C
   82 CONTINUE
C      COMPUTE ALPHA
       DT = FRTIME
       ALFAR8 = INALIN(4)
       IF ( ABS(ALFAR8)  . LT.100.) ALFAR8   =0.0
       ALFAR8 = (ALFAR8/16384.0) *ALFARX
       GLOAD = LIFT/WEIGHT
       IF ( GLOAD.LT.1.0) GO TO 1015
       ALFAR8 =ALFAR8- (GLOAD-1.0) *ALFARX/MAXG
       GO TO 1016
 1015 ALFAR8 =ALFAR8+ (GLOAD-1.0) *ALFARX/(MING*5.0)
 1016 CONTINUE
       ALPHA = ALPHA + ALFAR8 * DT
```

```
C
      IF (ALPHA .LT. -0.2) ALPHA=-0.2
      RHO = GAMMA
      FBET=0.
      IF ( IBET.LT. 0) FBET=1.
      BETAR8 = INALIN(1)
      IF ( ABS(BETAR8)  . LT.100.) BETAR8   =0.0
      BETAR8 = ( BETAR8/32768.0)*BETRMX
      BETA = BETA + BETAR8 * DT + FBET * PI
C
C
      D = RHO * CONST
      A = SIN(BETA)
      B = COS(BETA)
      X = D * A
      Y = -D * B
      XX(1) = X - 18.0 * B
      YY(1) = Y - 18.0 * A
      XX(2) = X + 18.0 * B
      YY(2) = Y + 18.0 * A
C
C
      XX(3) = X
      YY(3) = Y
C
      K = 3
      XX(4) = X -6.0 * ( B * CDEL(K) - A * SDEL(K))
      YY(4) = Y -6.0 * ( A * CDEL(K) + B * SDEL(K))
C
      DO 99 I = 1,4
C
      XX(I) = ((XX(I) - AHM6) * 1400. / 12. ) + 1375.
C
      YY(I) = ((YY(I) - AHM6) * 1400. / 12.) + 2350.
C
   99 CONTINUE
C
C
C ** COMPUTE HEAD RATE **
C
      MASS = WEIGHT / G
      PHI = -BETA
      SINPHI = SIN (PHI)
      COSPHI = COS (PHI)
      THETA = RHO
C     COMPUTE ATMOSPHERE
      ALT = Z
      IF (Z .GE. 35300) GO TO 1010
```

```
      TEMP = 59.0 - 0.00357 * ALT
      DUM1 = 1.0 - 0.00357 / 518.4 * ALT
      PRES = 2116.0 * DUM1 ** 5.256
      DENS = 0.002378 * DUM1 ** 4.256
      GO TO 1020
 1010 TEMP = -67.0
      DUM2 = (ALT - 35300.0) / 20930.
      PRES = 489.456 * EXP (-DUM2)
      DENS = PRES / 673946.0
 1020 SOUND = SQRT (PRES * 1.406 / DENS)
      MACH = VEL / SOUND *100.
      Q = 0.5 * DENS * VEL ** 2
C     COMPUTE FORCES
      CL = CL0 + CLSLO * ALPHA
      CD=0.03+0.27 * CL**2
      DRAG=Q*CD*AREA
      IF(ALPHA.LT.0.4) GO TO 1038
      CL=1.0 -(ALPHA-0.4)*2.
      IF(ALPHA.LT.0.6) GO TO 1038
      CL = 0.
      ALPHA = 0.6
      IF(VEL.LT.100.0) ALPHA=0
 1038 CONTINUE
      IF(VEL.LT.1.0) VEL=1.0
      FALT = 11500. - 0.25 * ALT
      FMACH= 2327.0 + 0.172 * ALT - 0.0000031*ALT*ALT
      THROT = INALIN(2)
      THROT = ( THROT + CALV ) / 32768. * CALD
      TMAX = FMACH* FLOAT(MACH)/100. + FALT
      TMAX = TMAX*2.0
      IF(ISWT8.EQ.0) TMAX=TMAX/4.0
      THRUST = THROT* TMAX
      LIFT = Q * CL * AREA
      LIFT = LIFT + THRUST * SIN (ALPHA)
      THRUST = THRUST * COS (ALPHA)
C     DYNAMICS
      MV = MASS * VEL
      VELDT = (THRUST - DRAG - WEIGHT * SINGAM) / MASS
      PSIDT = (LIFT * SINPHI) / (MV * COSGAM)
      GAMDT = (LIFT * COSPHI - WEIGHT * COSGAM) / MV
      VEL = VELDT * DT + VEL
      PSI = PSIDT * DT + PSI
      GAMMA = GAMDT * DT + GAMMA
      IF (IBET.LT.0) IBET=1
      IF ( ABS(GAMMA) . LT . PI2 ) GO TO 1040
      IBET = -IBET
      PSI = PSI+ PI
      IF ( GAMMA . LT . 0. ) GO TO 1035
```

```
      GAMMA = PI - GAMMA
      GO TO 1040
 1035 GAMMA = -PI - GAMMA
 1040 CONTINUE
      SINGAM = SIN (GAMMA)
      COSGAM = COS (GAMMA)
      DELX = VEL * COSGAM * COS (PSI) * DT
      DELY = VEL * COSGAM * SIN (PSI) * DT
      DELZ = VEL *SINGAM * DT
      APX = APX + DELX
      APY = APY + DELY
      Z = Z + DELZ
      ALT = Z
      VVEL = DELZ / DT
      XVEL = DELX / DT
      YVEL = DELY / DT
      GDSPD=VEL *0.592
      KGLOD = LIFT / WEIGHT * 100.
      TRGHDN = PSI
      GAMMAD = GAMMA * 57.296
C
C ** COMPUTE COSINE OF TRGHDN **
C
      CTRGHD = COS ( TRGHDN )
C
C ** COMPUTE SINE OF TRGHDN   **
C
      STRGHD = SIN ( TRGHDN )
C
C  COMPUTE COMPASS POINT ***
C
      ZZ = 90. / 57.2956
      TRIG2 = 2. * ZZ - TRGHDN
      COMPX = (((39.9 * COS(TRIG2 - 100. / 57.2956)) - UN1) * XN21 /
     1   UN21) + XN1
      COMPY = (((39.9 * SIN (TRIG2 - 100. / 57.2956)) - VN1) * YN21 /
     1   VN21) + YN1
C
      DO 90  LPP = 1,7,2
      CX(LPP) = (((RADIUS * COS(TRIG2)) - UN1) * XN21 / UN21) + XN1 -56.
      CX(LPP+1) = (((RADIUS * SIN(TRIG2)) - VN1) * YN21 / VN21) + YN1
   90 TRIG2 = AMOD((TRIG2 - ZZ), 6.2832)
C
C
C
C ** GET NEW POSITION OF XAC AND YAC
C
      XAC = APX / 6080.
```

```
      YAC = APY / 6080.
C
C
      K = 1
      ITCNT = 0
C
      CTLOOP = ICTLP
      IF (ITCT .EQ. 0) GO TO 475
      DO 300 I= 1,ITCT
      IF ( CTLOOP .GT. TOF(I)) GO TO 300
      IF ( CTLOOP .LT. TON(I)) GO TO 475
C
C** COMPUTE A UT AND VT AND CHECK IF   **
C** THEY ARE WITHIN SIZET LIMITS       **
      UT = TX(I) - XAC
      VT = TY(I) - YAC
C
      IF (ABS(UT) - SIZET) 100, 100, 300
C
  100 IF (ABS(VT) - SIZET) 110, 110, 300
  110 CONTINUE
C** GENERATE A DX AND DY AND CONVERT  **
C** TO RASTER UNITS                   **
C
      XS = (-1.) * VT * CTRGHD + UT * STRGHD
      YS =         VT * STRGHD + UT * CTRGHD
C
      IF (ABS(XS) - SIZET)125,125,300
C
  125 IF (ABS(YS) - SIZET)135,135,300
  135 CONTINUE
C
C** CONVERT TO RASTER UNITS
C
      IDTX(K) = (( XS - U1) * X21 / U21 ) + X1
      IDTY(K) = (( YS - V1) * Y21 / V21 ) + Y1
C
      IDTYPE(K) = ITYPE(I)
      IDTCHA(K) = ICHAR(I)
      K = K + 1
      ITCNT = ITCNT + 1
  300 CONTINUE
C
C** COMPUTE X AND Y NAVIGATIONAL POSITIONS **
C
  475 CONTINUE
      N = 1
      L = 1
```

```
      INCNT = 0
      IF (INCT .EQ. 0) GO TO 560
C
  500 CONTINUE
C
C** COMPUTE A UN AND VN AND CHECK IF **
C** THEY ARE WITHIN SIZEN LIMITS     **
C
      UN = NX(N) - XAC
      VN = NY(N) - YAC
C
      IF (ABS(UN) - SIZEN) 520, 520, 550
C
  520 IF (ABS(VN) - SIZEN) 530, 530, 550
C
  530 CONTINUE
C** GENERATE A DNX AND DNY AND CONVERT **
C** TO RASTER UNITS                    **
C
      XNS = (-1.) * VN * CTRGHD + UN * STRGHD
      YNS =         VN * STRGHD + UN * CTRGHD
      IF (ABS(XNS) - SIZEN ) 535, 535, 550
C
  535 IF (ABS(YNS) - SIZEN ) 540, 540, 550
C
  540 CONTINUE
C
C** CONVERT TO RASTER UNITS  **
C
      IDNX(L) = (( XNS - UN1)* XN21 / UN21 ) + XN1
      IDNY(L) = (( YNS - VN1)* YN21 / VN21 ) + YN1
      INCTYP(L) = NCTY(N)
      L = L + 1
      INCNT = INCNT + 1
  550 CONTINUE
      N = N + 1
C
C** CHECK IF (N) IS LESS THAN INCT ( INCT  = NO. OF NAVG. PTS ) **
C
      IF ( N .LE. INCT ) GO TO 500
C
C***************************************************************
C* SET UP THE REACTION TESTS IF THERE ARE ANY TO SET UP       *
C***************************************************************
  560 CONTINUE
      IF (IBC1 .EQ. 1) GO TO 660
      IF (IHOTSW .EQ. 0) GO TO 800
      NEWTIM = ICTLP
```

```
      IBC1 = 1
      IBC2 = 0
C
C     DECODE THE MANEUVER NO.
C
      DGP = DIGIN1(3) - 1024
      M = 0
      IF (DGP .EQ. 0) GO TO 580
      M = 1
      J = 2
C
      DO 570  L = 1,9
      IF (DGP .EQ. J) GO TO 580
      J = J * 2
      M = M + 1
  570 CONTINUE
      M = 0
      GO TO 620
C
  580 M = M * 10
      DGP = DIGIN1(4)
      IF (DGP .GE. 1024) DGP = DGP - 1024
      N = 0
      IF (DGP .EQ. 0) GO TO 600
      N = 1
      J = 2
C
      DO 590  L = 1,9
      IF (DGP .EQ. J) GO TO 600
      J = J * 2
      N = N + 1
  590 CONTINUE
      M = 0
      GO TO 620
C
  600 M = M + N
C
  620 IF (M .NE. 0) GO TO 630
C
C  FALL THRU HERE FOR INVALID MANEUVER NO. AND IGNORE THIS MANEUVER
C
  625 IBC1 = 0
      GO TO 800
C
  630 CONTINUE
      IF (IMSEQ(M) .EQ. 0) GO TO 625
      ISEQNC = IMSEQ(M)
      ISEQCT = ICNTAB(ISEQNC)
```

```
      IPTR = ISEQNC + 1
C
C
  640 IF (IBC2 .EQ. 1) GO TO 800
C
C  FALL THRU HERE TO CHECK FOR A NEW CHARACTER TO BE DISPLAYED.
C
      IFULL = ICNTAB(IPTR)
      IGOTIM = IHAF(1) + NEWTIM
      IF (ICTLP .LT. IGOTIM) GO TO 800
      NCSW = 1
      ICARFG = 1
      IEND = ICTLP + 50
      CHAR(1) = IHAF(2)
      GO TO 800
C
C
C
  660 IF (IHOTSW .EQ. 1) GO TO 670
      IBC1 = 0
      IRESET = 1
      CHAR(1) = BLANK
      ISTART = 0
      ICARFG = 0
      GO TO 800
C
  670 IF (ICARFG .EQ. 0) GO TO 640
      IF (ISTART .EQ. 0) GO TO 680
      IF (ICTLP .LT. IEND) GO TO 800
C
C  FALL THRU HERE IF SUBJECT FAILED TO RESPOND TTO A CHARACTER.
C
      CHAR(1) = BLANK
      ISTART = 0
C
C
C  SET UP NEXT CHARACTER IN THE MEMORY SET
C
  680 ICARFG = 0
      IRESET = 1
      ISEQCT = ISEQCT - 1
      IF (ISEQCT .GT. 0) GO TO 690
      IBC2 = 1
      GO TO 800
C
  690 IPTR = IPTR + 1
      IBC2 = 0
      GO TO 640
```

```
C
C
  800 CONTINUE
C       CHECK IF CATEGORY 3 SWITCH IS ON .
C       IF SWITCH IS ON THAN COMPUTE DATA
C       NECESSARY TO GENERATE DISPLAY
C
        IF ( ISWTC3 .EQ. 0 .AND. ISWTC2 .EQ. 0 ) GO TO 850
C
        ISWTC3 = 0
        ISWTC2 = 0
        RPM1=  THROT*100.
        RPM2= RPM1
        RPMY1 = RPM1 * 650 / 100 + 850
        RPMY2 = RPM2 * 650 / 100 + 850
C
        RPMY3 = RPM1 * 700 / 100 + 450
        RPMY4 = RPM2 * 700 / 100 + 450
C
C     RESCALE MACH VALUES TO RASTER UNITS   **
C
C
C
  850 CONTINUE
        MACHXY(1) = 0
        M = 1
        RASM = 3830.0 * VEL / SOUND + 2925.
        IRASM = RASM
C
        DO  870  N = 1,21
C
        IF ( IRASM .GT. HIUR .OR. IRASM .LT. LOUR ) GO TO 860
C
        MACHTX(M) = IMACH(N)
        MACHXY(M+1) = IRASM
        MACHXY(1) = MACHXY(1) + 1
        M = M + 1
C
  860 IRASM = IRASM - 383
C
  870 CONTINUE
C
  900 CONTINUE
        IAS(1) = 0
        K = 1
        SPEEDK = GDSPD
        RAST = SPEEDK * 2.87 + 2925.
        IRAS = RAST
```

```
C
      DO  910  I = 1,21
C
      IF (IRAS .GT. HIUR .OR. IRAS .LT. LOUR) GO TO 905
C
      ISIAS(K) = ISPDS(I)
      IAS(K+1) = IRAS
      IAS(1) = IAS(1) + 1
      K = K + 1
C
  905 IRAS = IRAS - 287
C
  910 CONTINUE
C
C  *****************************************************************
C    COMPUTE Y FOR VERTICAL VELOCITY AND CONVERT TO RASTER UNITS   **
C  *****************************************************************
C
      VRAST = VVEL * 13.8 + 2925.
      IVDUM = VVEL * .6
      IVRAST = VRAST
C
      IF (IVRAST .GT. 3385 ) GO TO 925
C
      IF (IVRAST .LT. 2465 ) GO TO 930
C
      GO TO 950
  925 IVRAST = 3385
      GO TO 950
C
  930 IVRAST = 2465
C
  950 CONTINUE
      KMACH = TMAX/10.
      KALT = ALT
      KALF = DALF
      KVV = ALPHA*1800/PI
      KGAM = GAMMA * 57.3
      KGAM = (THRUST-DRAG)/10.
      ISPD = GDSPD
      KXAC = XAC
      KYAC = YAC
C
C
C
C    RESCALE ATLIMETER Y VALUES TO RASTER UNITS FOR 1000 SCALE
C
      IALTXY(1) = 0
```

```
      IF ( ABS(ALT) .LT. 1. ) ALT = 1.
C
C
C   27255 = 230 * 121 - 575
C   121 REPRESENTS THE NUMBER OF ENTRIES
C   TO ENTRY NUMBER 0 . 230
C   REPRESENTS 230 RASTER UNITS BETWEEN
C   EACH ENTRY ON THE DISPLAY .
C   575 IS THE NUMBER OF RASTER UNITS TO THE
C   CENTER OF THE DISPLAY . THE DISPLAY
C   IS 1150 RASTER UNITS LONG IN THE Y COORD.
C
      IF (ALT .GT. 120000.) ALT = 120000.
      IF (ALT .LT. -15000.) ALT = -15000.
      RALT = -(ALT / 1000. ) * 230. + 27255.
      IALTA = RALT / 230.0
C
C     3500. REPRESENTS THE HIGHEST RASTER
C     UNIT USED IN THE DISPLAY . THE Y COORD.
C     GOES FROM 2350 RU TO 3500 RU ( 1150 RU,S )
C
      DIFF = RALT - ( IALTA * 230 ) + 3500.
      JNDX = IALTA + 1
      DIFF = DIFF - 230.
C
      DO 1150 J = 1,5
C
      IALTXY(J) = IALT1(JNDX)
      IALTXY(J+1) = DIFF
      IF (IALTXY(J+1) .LT. LCUR ) GO TO 1151
      DIFF = DIFF - 230.
      IALTXY(1) = IALTXY(1) + 1
      JNDX = JNDX + 1
      IF ( JNDX .GT. 136 ) GO TO 1151
 1150 CONTINUE
 1151 CONTINUE
C
C
C
C   RESCALE ALTIMETER Y VALUES TO RASTER UNITS FOR 100 SCALE
C
      IALHXY(1) = 0
      IF ( ABS(ALT) .LT. 1.) ALT = 1.
C
      IDUM1 = ALT / 1000.
      ALT100 = ALT -( 1000. * FLOAT ( IDUM1 ))
      ALT100 = ALT100 / 100.
C
```

```
C
C     2415 = 230 * 13 - 575
C     13 REPRESENTS THE NUMBER OF ENTRIES
C     TO ENTRY CONTAINING THE 2ND ZERO .
C     230 REPRESENTS 230 RASTER UNITS BETWEEN
C     EACH ENTRY ON THE DISPLAY .
C    575 IS THE NUMBER OF RASTER UNITS TO THE
C    CENTER OF THE DISPLAY . THE DISPLAY
C    IS 1150 RASTER UNITS LONG IN THE Y COORD.
C
      RALTH = -ALT100 * 230. + 2415.
      IALTH = RALTH / 230.0
C
C     3500. REPRESENTS THE HIGHEST RASTER
C     UNIT USED IN THE DISPLAY . THE Y COORD.
C     GOES FROM 2350 RU TO 3500 RU ( 1150 RU,S )
C
      DIFFH = RALTH - ( IALTH * 230 ) + 3500.
      HNDX = IALTH + 1
      DIFFH = DIFFH - 230.0
C
      DO 1250 J = 1,5
C
      IALTXH(J) = IALT(HNDX)
      IALHXY(J+1) = DIFFH
C
      IF (IALHXY(J+1) .LT. LCUR ) GO TO 1251
C
      DIFFH = DIFFH - 230.
      IALHXY(1) = IALHXY(1) + 1
      HNDX = HNDX + 1
 1250 CONTINUE
 1251 CONTINUE
C
C
C
C    COMPUTE Y FOR ACCELEROMETER ( G-LOAD )
C    AND CONVERT TO RASTER UNITS
C
      RGLOAD = KGLOD
      IACCY(1) = 0
      IF ( ABS( RGLOAD ) .LT. 0.01) RGLOAD = .01
C
C
C     3105 = 230 * 16 - 575
C     16 REPRESENTS THE NUMBER OF ENTRIES
C     TO ENTRY CONTAINING THE ZERO .
C     230 REPRESENTS 230 RASTER UNITS BETWEEN
```

```
C      EACH ENTRY ON THE DISPLAY
C     575 IS THE NUMBER OF RASTER UNITS TO THE
C     CENTER OF THE DISPLAY . THE DISPLAY
C     IS 1150 RASTER UNITS LONG IN THE Y COORD.
C
       RGLOD = -( RGLOAD / 100. ) * 230. + 3105
       IACCL = RGLOD / 230.0
C
C      3500. REPRESENTS THE HIGHEST RASTER
C      UNIT USED IN THE DISPLAY . THE Y COORD.
C      GOES FROM 2350 RU TO 3500 RU ( 1150 RU,S )
C
       DIFFG = RGLOD - ( IACCL * 230 ) + 3500.
       GNDX = IACCL + 1
       DIFFG = DIFFG - 230.0
C
       DO  1350 J = 1,5
C
       IACCTX(J) = IACC(GNDX)
       IACCY(J+1) = DIFFG
C
       IF ( IACCY(J+1) .LT. LOUR ) GO TO 1351
C
       DIFFG = DIFFG - 230.0
       IACCY(1) = IACCY(1) + 1
       GNDX = GNDX + 1
 1350 CONTINUE
 1351 CONTINUE
C
C     COMPUTE Y FOR ANGLE OF ATTACK
C     AND CONVERT TO RASTER UNITS
C
       IAOAY(1) = 0
C
       DALF = ALPHA * 36. / PI
       IF ( ABS(DALF) .LT. .01) DALF = .01
C
C
C      1725 = 230 * 10 - 575
C      10 REPRESENTS THE NUMBER OF ENTRIES
C      TO ENTRY CONTAINING ZERO .
C      230 REPRESENTS 230 RASTER UNITS BETWEEN
C      EACH ENTRY ON THE DISPLAY .
C     575 IS THE NUMBER OF RASTER UNITS TO THE
C     CENTER OF THE DISPLAY . THE DISPLAY
C     IS 1150 RASTER UNITS LONG IN THE Y COORD.
C
       RAOA = -DALF * 230. + 1725.
```

```
      IRADA = RADA / 230.0
C
C     3500. REPRESENTS THE HIGHEST RASTER
C     GOES FROM 2350 RU TO 3500 RU ( 1150 RU,S )
C     UNIT USED IN THE DISPLAY . THE Y COORD.
C
      DIFFA = RADA - ( IRADA * 230 ) + 3500.
      ANDX = IRADA + 1
      DIFFA = DIFFA - 230.0
C
      DO  1450 J = 1,5
C
      IADATX(J) = IADA(ANDX)
      IADAY(J+1) = DIFFA
C
      IF ( IADAY(J+1) .LT. LOUR ) GO TO 1451
C
      DIFFA = DIFFA - 230.0
      IADAY(1) = IADAY(1) + 1
      ANDX = ANDX + 1
 1450 CONTINUE
 1451 CONTINUE
      RETURN
      END
```

APPENDIX B

DESCRIPTION OF COMPUTER PROGRAM

This appendix documents the FORTRAN computer program written to demonstrate the CPM technique. The details of the computer program are given and a listing is included. Figure B-1 shows a functional block flow chart for the program.

The initial setup portion of the program contains the declaration of all matrices and of variables with abnormal type attributes. Certain arrays are initialized to zero. The initial time is set to zero and the numerical integration step size is set to 0.5 seconds.

The next section of coding contains read statements which input from cards as follows: the initial state vector X(0) and the X-position of the aircraft with respect to the earth X_e; the reference state vector X_R; the reference control vector U_R; the diagonal elements of the Q and R matrices; the initial control settings (U_1, U_2, and U_3); the initial setting for the autopilot's normalized percent throttle; a program control switch named ISWT; and the number of iterations desired for program execution, a parameter called ITIME. Table B-1 presents the input data card structure for the eight input cards needed for the program. Note that if ISWT is greater than zero the autopilot controls the aircraft, but if ISWT is set less than zero, the aircraft motion is governed by the approximate optimal control. In this section of coding certain off-diagonal elements of the Q matrix were set equal to non-zero values.

Next the program prints the initial state vector, reference state vector, reference control and diagonal elements of the Q and R matrices.

The next step is the main loop of the program in which values of the elements of the F and G matrices are computed from the present value of the state vector. The four intermediate variables CL1, CL2, L and AL1 are computed in this process, according to the equations of Section 5.

The next section of coding implements an iterative technique to obtain a numerical solution for the elements of the matrix K(X), which is the positive definite symmetric solution to the algebraic Riccati equation. Appendix C contains a detailed discussion of the numerical method used to compute K(X). This is the longest section of coding in the computer program and consumes the major portion of the execution time.

The next section of coding contains write statements to print some intermediate results. The G matrix; F matrix; the matrix F_o

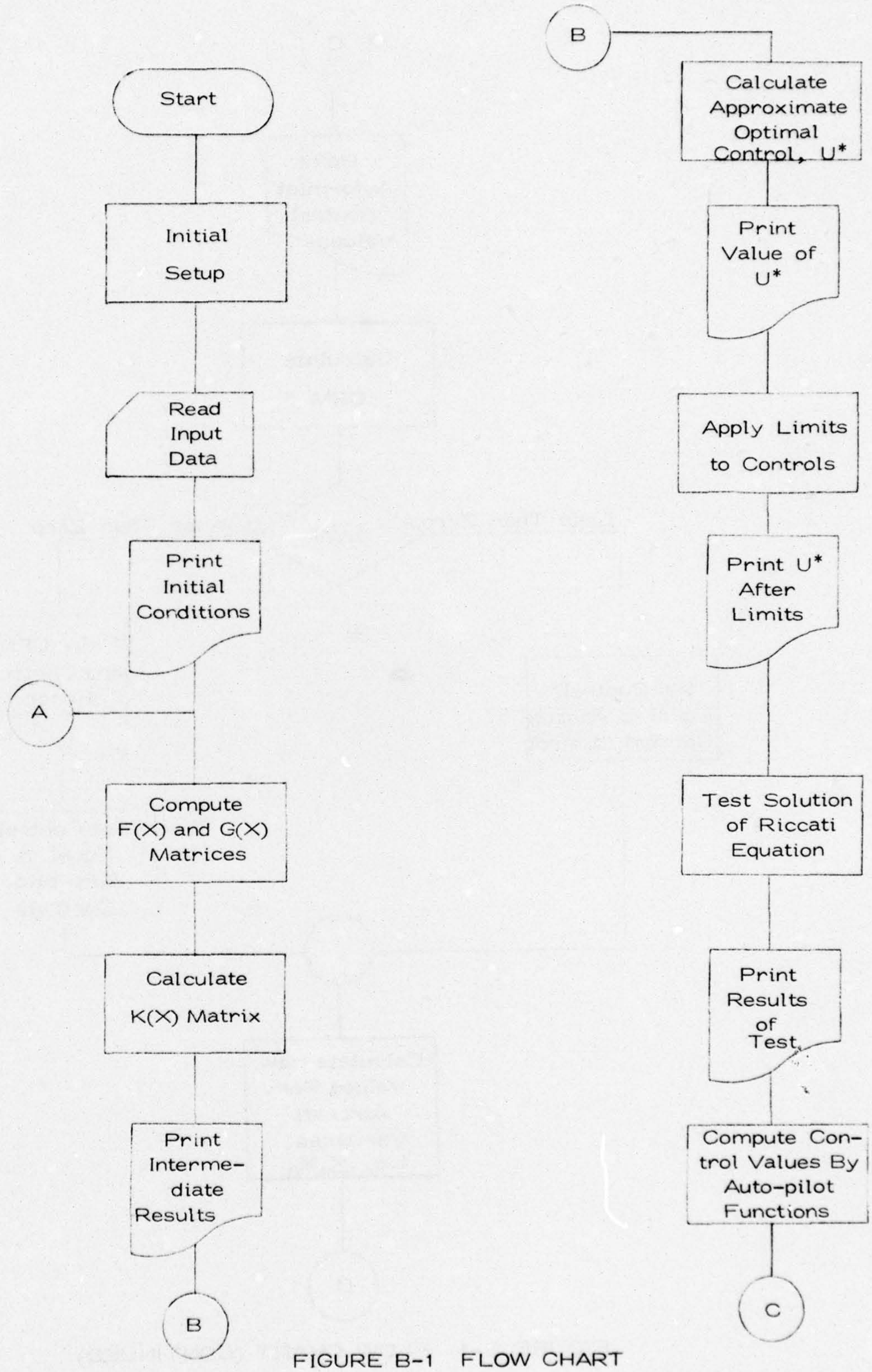


FIGURE B-1 FLOW CHART

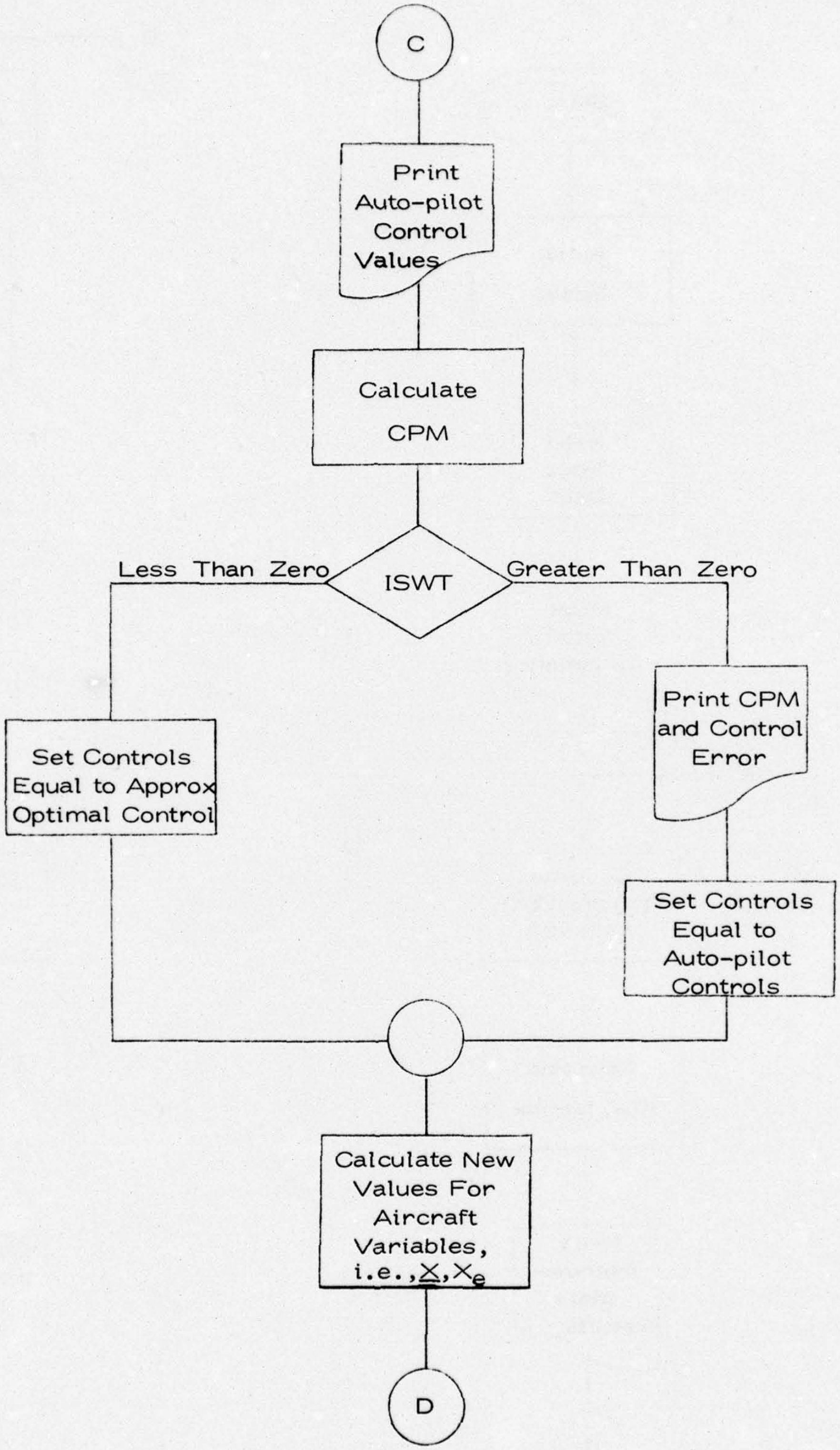


FIGURE B-1 FLOW CHART (CONTINUED)

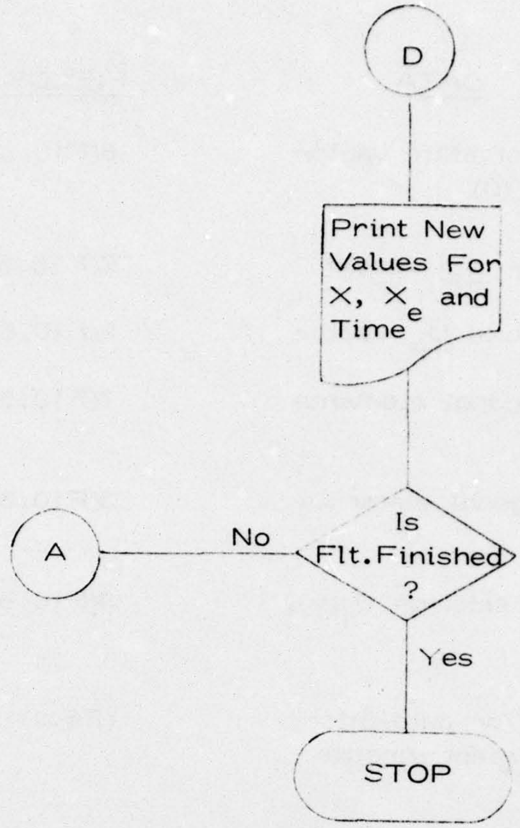


FIGURE B-1 FLOW CHART (CONCLUDED)

CARD	DATA	FIELDS	TYPE
1	Initial values of state vector $X(0)$ and of $X_e(0)$	8(F10.5)	Real
2	Reference state $\underline{X}_R$ vector	7(F10.5)	Real
3	Reference control $\underline{U}_R$ vector	3(F10.5)	Real
4	The seven diagonal elements of Q matrix	7(F10.5)	Real
5	The three diagonal elements of R matrix	3(F10.5)	Real
6	Initial control settings $U_1(0)$, $U_2(0)$, $U_3(0)$	3(F10.5)	Real
7	Initial setting for auto-pilot normalized percent throttle UPTL(3)	(F6.2)	Real
8	Program control parameters ISWT and ITIME where ISWT = $>$ 0 for auto-pilot $<$ 0 for optimal ITIME = number of interations 2 ITIME = total number of seconds of run	(I2, I6)	Integer

TABLE B-1 INPUT DATA CARD STRUCTURE

used to start the iterative method of obtaining matrix K; the parameters CL1, CL2, L and AL1; and the matrix K are printed.

The next several sections of coding calculate the approximate optimal control in the following manner. First, the approximate optimal control vector is computed and the control values printed. Next, appropriate limits, as discussed in Section 5, are applied to the approximate optimal control to conform to realistic stick and throttle positions. Finally, the approximate optimal controls after the limits have been applied are printed.

The approximate solution to the Riccati equation (K) is evaluated. This is done by computing the left hand side to Equation 5.23 which should equal zero. The result of this evaluation is then printed.

Next, the value of the control variables are computed using the autopilot functions described in Section 5.4. This solves the "non-optimal" autopilot aircraft control equations. These autopilot control values are then printed.

The following section of coding calculates the CPM when the autopilot equations are used. The control error, (the approximate optimal control subtracted from the autopilot generated control), is computed followed by calculation of the CPM.

If ISWT is less than zero, the approximate optimal controls are used and the program goes to the section of coding that calculates the new state vector. If, however, ISWT is greater than zero, the values for the CPM and the control error are printed and the autopilot generated controls are used. The aircraft state vector will then reflect the impact of these non-optimal autopilot inputs when the program updates the location of the aircraft.

Next, new values of the aircraft variables, are obtained by numerical integration. Rectangular integration is used, with a fixed step size of 0.5 seconds. The values of the new aircraft variables and time are then printed.

Finally, the program determines if the flight calculation is complete by comparing the number of main loop iterations to the input parameter ITIME. If not, the program returns to the start of the main loop (Point A in Figure B-1) for a new iteration. If the number of iterations exceeds ITIME, the program terminates.

Extensive use is made by the main program of three subroutines documented in the IBM Scientific Subroutine Package: (1) a matrix inversion subroutine called MINV which replaces the square input matrix by its inverse; (2) a general matrix product subroutine called GMPRD that multiplies any two conformable matrices and stores the result in a different matrix; and (3) a matrix addition subroutine called MADD that stores the linear combination of two matrices of the same dimensions in a different matrix.

The following is a listing of the FORTRAN program just described.

```
      DOUBLE PRECISION DPW(7,7),DPRINV(3,3),DPDUM2(7,7),WDET,RDET,RPDET,
     - DETCPL
      REAL K,L,N,KXMN,LP
      DIMENSION X(7),G(7,3),F(7,7),XDOT(7),FX(7),GU(7),U(3),T(3,7),
     -    GT(7,7),FO(7,7),TT(7,3),GTRAN(3,7),DUM(7,7),TR(7), COF(8),
     -   RDOT(14),LP(7,7),RP(7,7),WORK1(7),WORK2(7),BUF(7,7),DUM1(7,7),
     -   DUM2(7,7),CTRL(7,21),MWORK(7),LWORK(7),Q(7,7),R(3,3),FT(7,7),
     -   XRP(7,7),K(7,7),RINV(3,3),UCP(3),XR(7),N(7),XNORM(7),
     -   UPLT(3),CPM1(7),CPM2(3),CPM5(7),KXMN(7),C(7,7),CUTX(7),
     -   BUFF3(7,3),W(7,7),RINT(7,7),BUFF4(7,3),BUFF2(3,7),BUFF1(3,7),
     -   ROOT1(8),ROOT2(8),WRK9(8),IWORK1(7),IWORK2(7),IWORK3(8),
     -   IWORK4(8),IWORK5(3),IWORK6(3),WRK3(7),WRK4(3),MWORK1(21),
     -   H1(7,7),H2(7,7),E1(7,7),EE(7,7),AA(7,7),XXD(7),EX(7),UP(3),
     -   ZZ(3),UE(3),BUF88(7,7)
      DATA Q,R,XDOT,XEDOT/49*0.0,9*0.0,7*0.0,0.0/
      DT=0.5
      PHIL=40.0*0.017453
      TTIME=0.0
      LN=40
C
C
C        INPUT INITIAL CONDITIONS
C
C
      READ(5,7702) ((X(I)),I=1,7),XE
 7702 FORMAT(8(F10.5))
      READ(5,7700) ((XR(I)),I=1,7)
      READ(5,7703) ((UR(I)),I=1,3)
 7703 FORMAT(3F10.5)
      READ(5,7700) ((Q(I,I)),I=1,7)
      READ(5,7701) ((R(I,I)),I=1,3)
      READ(5,7701) ((U(I)),I=1,3)
 7701 FORMAT(3(F10.5))
      UOP(3)=U(3)
      Q(2,3)=0.001
      Q(2,6)=0.001
      Q(3,2)=0.001
      Q(6,2)=0.001
      READ(5,7710) UPLT(2)
      READ(5,7720) ISWT,ITIME
 7700 FORMAT(7 (F10.5))
 7710 FORMAT(F6.2)
 7720 FORMAT(I2,I6)
C
C
C        OUTPUT INITIAL CONDITIONS
C
C
      WRITE(6,2117)
 2117 FORMAT('1','INITIAL CONDITIONS FOLLOWS')
      WRITE(6,1072)
 1072 FORMAT('-','   THE INITIAL STATE VECTOR FOLLOWS: ')
      WRITE(6,1070) ((X(I)),I=1,7),XE
 1070 FORMAT(' ',8(3X,F10.5))
      WRITE(6,2119) ((XR(I)),I=1,7)
 2119 FORMAT(2X,'REFFRANCE X FOLLOWS'/2X,7(2X,F13.6))
      WRITE(6,1066) ((UP(I)),I=1,3)
 1066 FORMAT(' ',10X,'REFFRANCE  CONTROL IS:'/' ',3(5X,F10.5))
      WRITE(6,2122) ((Q(I,I)),I=1,7)
 2122 FORMAT(2X,'Q DIAGIONAL ASSIGNMENTS FOLLOW'/2X,7(2X,F13.6))
      WRITE(6,2123) ((R(I,I)),I=1,3)
```

```
 2123 FORMAT(2X,'R DIAGINIAL DEFINITIONS FOLLOW'/2X,3(2X,F13.6))
      DO 999 INDEX=1,ITIME
      IF(ISWT.GT.0) U3=UPLT(3)
      IF(ISWT.LT.0) U3=UCP(3)
C
C
C  THIS SEGMENT CALCULATES THE VALUES OF THE F AND
C  G MATRICES IN THE STATE EQUATIONS
C
C
      ALPH=X(1)
      PHI  =X(2)
      PSI  =X(3)
      GAM  =X(4)
      V    =X(5)
      YE   =X(6)
      ZE   =X(7)
      V2   =V*V
      ALPH2=ALPH*ALPH
C
C
C THE FOLLOWING VALUES ARE PARAMETERS OF F AND G
C
C
      IF (ALPH .GE. 0.6) GO TO 110
      IF (ALPH .GE. 0.4) GO TO 120
      CL1=0.1
      CL2=2.5
      GO TO 150
  110 CL1=0.0
      CL2=0.0
      GO TO 150
  120 CL1=1.8
      CL2=-2.0
  150 L=(0.13673)*CL1*V2+(0.13673)*CL2*ALPH*V2+
     -     ((2.1103)*V+3500.0)*SIN(ALPH)*U3
      AL1=00.0666
      IF (L .LT.17000.) AL1=00.04
C
C
C  THE FOLLOWING SETS G EQUAL TO ZERO AND THEN DEFINES ALL
C THE NONZERO VALUES
C
C
      DO 200 I=1,7
      DO 200 J=1,3
  200 G(I,J)=0.0
      G(1,1)=1.0
      G(1,3)=((1.241E-4)*V+(0.205930))*SIN(ALPH)*(-AL1)
      G(2,2)=1.0
      G(3,3)=((3.997E-3)+(6.62880)/V)*SIN(ALPH)*SIN(PHI)
     -          /COS(GAM)
      G(4,3)=((3.997E-3)+6.62880/V)*SIN(ALPH)*
     -           COS(PHI)
      G(5,3)=((3.997E-3)*V+6.6288)*COS(ALPH)
C
C
C        TRANSPOSE G TO GTRAN
C
C
      DO 211 IG=1,7
```

```
      DO 211 JG=1,3
      GTRAN(JG,IG)=G(IG,JG)
  211 CONTINUE
C
C
C  THE FOLLOWING DEFINES F IN A SIMILAR WAY
C
C
      DO 300 I=1,7
      DO 300 J=1,7
  300 F(I,J)=0.0
      F(6,5)=COS(GAM)*SIN(PSI)
      F(7,5)=SIN(GAM)
      F(1,5)=(-(8.0E-6)*AL1*CL1*V)
      F(1,1)=(-(8.0E-6)*AL1*CL2*V2)
      F(1,7)=AL1/ZE
      F(3,1)=(2.589E-4)*CL2*V*SIN(PHI)/COS(GAM)
      F(3,5)=(2.589E-4)*CL1*SIN(PHI)/COS(GAM)
      F(4,7)= -(32.2)*COS(GAM)/(V*ZE)
      F(4,1)=(2.589E-4)*CL2*V*COS(PHI)
      F(4,5)=(2.589E-4)*CL1*COS(PHI)
      F(5,7)=-(32.2)*SIN(GAM)/ZE
      F(5,1)=(-3.4952E-5)*V2
      F(5,5)=(-4.3689E-4)*ALPH2*V-(8.466E-6)*V
C
C
C     THIS SEGMENT CALCULATES A STARTING FO WHICH WILL
C     BE USED IN NEWTON'S METHOD TO GENERATE K
C
C
 3011  DO 50 I=1,3
      DO 50 J=1,7
   50 T(I,J)=00.0
C
C
C        FIND THE SEED
C
C
      DT1=0.1
      TIME=1.5
      DO 3010 ID=1,7
      DO 3010 JD=1,7
      W(ID,JD)=0.0
 3010 CONTINUE
      TOTOT=DT1
 3020 DO 3015 ID=1,7
      DO 3015 JD=1,7
      RINT(ID,JD)=0.0
      DUM(ID,JD)=0.0
      IF(ID.EQ.JD) DUM(ID,JD)=1.0
 3015 CONTINUE
C
C
C       RINT=EXP(-P*TOTOT)
C
C
      PROD=1.0
      DO 3030 ID=1,11
      RI=ID
      CALL MADE(RINT,DUM,RINT,7,7,1.0,PROD)
      PROD=PROD*(-TOTOT)/RI
```

```
      CALL GMPRD(DUM,F,BUF,7,7,7)
      CALL MADD(BUF,DUM,DUM,7,7,1.0,0.0)
 3030 CONTINUE
C
C
C       RINT=(RINT*G)*TRANSPOSE(RINT*G)
C
C
      CALL GMPRD(RINT,G,BUFF4,7,7,3)
      DO 3035 ID=1,7
      DO 3035 JD=1,3
      BUFF2(JD,ID)=BUFF4(ID,JD)
 3035 CONTINUE
      CALL GMPRD(BUFF4,BUFF2,PINT,7,3,7)
C
C
C      INTEGRATION IS NOW DONE
C
C
      CALL MADD(W,RINT,W,7,7,1.0,DT1)
      TOTDT=TOTDT+DT1
      IF(TOTDT.LE.TIME) GO TO 3020
      DO 917 I=1,7
      DO 917 J=1,7
      DPW(I,J)=W(I,J)
  917 CONTINUE
      CALL MINV(DPW,7,WDET,IWORK1,IWORK2)
      DO 918 I=1,7
      DO 918 J=1,7
      W(I,J)=DPW(I,J)
  918 CONTINUE
      CALL GMPRD(GTRAN,W,T,3,7,7)
      CALL GMPRD(G,T,GT,7,3,7)
      CALL MADD(F,GT,FD,7,7,1.0,-1.0)
C
C
C
C THIS SEGMENT IMPLMENTS KLEINMAN'S METHOD IN SOLVING FOR THE
C OPTIMAL CONTROL
C
C ALL THE MATRICES NEEDED FOR THE METHOD ARE FIRST
C OBTAINED
C
C THE TRANSPOSE OF T IS DEFINED AS TT AND THAT OF G AS GTRAN
C
C
C
C
      ICT=0
    2 DO 10 I=1,3
      DO 10 J=1,7
      TT(J,I)=T(I,J)
   10 CONTINUE
C
C
C THE INVERSE OF R IS CALLED RINV
C
C
      DO 919 I=1,3
      DO 919 J=1,3
      DPRINV(I,J)=R(I,J)
  919 CONTINUE
```

```
      CALL MINV(DPRINV,3,RDET,IWORK5,IWORK6)
      DO 920 I=1,3
      DO 920 J=1,3
      RINV(I,J)=DPRINV(I,J)
  920 CONTINUE
C
C
C  THE TRANSPOSE OF FO IS DEFINED AS FT
C
C
      DO 30 I=1,7
      DO 30 J=1,7
      FT(J,I)=FO(I,J)
   30 CONTINUE
C
C
C    SET C=Q+TT*R*T
C
C
      CALL GMPRD(R,T,BUFF2,3,3,7)
      CALL GMPRD(TT,BUFF2,DUM1,7,3,7)
      CALL MADD(Q,DUM1,C,7,7,1.0,1.0)
C
C
C  THE SERIES SOLUTION TO THE EQUATION IS NOW CALCULATED
C
C
      E=20.0
C
C
C  THE FOLLOWING ARE TWO INTERMEDIATE MATRICES NEEDED FOR
C  THE METHOD
C
C
      DO 35 I=1,7
      DO 35 J=1,7
      LP(I,J)=FT(I,J)
      RP(I,J)=-FO(I,J)
      IF (I .NE. J) GO TO 35
      LP(I,J)=LP(I,J)+E
      RP(I,J)=RP(I,J)+E
   35 CONTINUE
      DO 36 I=1,7
      DO 36 J=1,7
      DPDUM2(I,J)=RP(I,J)
   36 CONTINUE
      CALL MINV(DPDUM2,7,RPDET,IWORK1,IWORK2)
      DO 37 I=1,7
      DO 37 J=1,7
      RP(I,J)=DPDUM2(I,J)
   37 CONTINUE
C
C
C  THE FIRST TERM IS NOW OBTAINED
C
C
      CALL GMPRD(C,RP,BUF,7,7,7)
      CALL MADD(DUM1,BUF,K,7,7,0.0,1.0)
C
C
C  THE NEXT TERM IN THE SERIES IS ADDED, K=LP*K*RP+BUF
```

```
C
C
      DO 3017 IG=1,20
      CALL GMPRD(K,PP,DUM1,7,7,7)
      CALL GMPRD(LP,DUM1,DUM2,7,7,7)
      CALL MADD(DUM2,BUF,K,7,7,1.0,1.0)
 3017 CONTINUE
C
C
C  THE NEXT T IS OBTAINED, T=RINV*GTRAN*K
C
C
  100 CALL GMPRD(GTRAN,K,BUFF1,3,7,7)
      CALL GMPRD(RINV,BUFF1,T,3,3,7)
C
C
C  THE NEW FO IS OBTAINED, FO=F-G*T
C
C
      CALL GMPRD(G,T,DUM1,7,3,7)
      CALL MADD(F,DUM1,FO,7,7,1.0,-1.0)
C
C
C  A TEST ON THE NUMBER OF ITERATIONS IS MADE
C
C
      ICT=ICT+1
      IF(ICT .LE. 20) GO TO 2
C
C
C  THE RESULTS ARE NOW PRINTED
C
C
      WRITE (6,1000)
 1000 FORMAT('1','   THE G MATRIX IS:'/)
      WRITE (6,1100) ((G(I,J),J=1,3),I=1,7)
 1100 FORMAT (3(1X,E20.9),/)
      WRITE (6,2000)
 2000 FORMAT (' THE F MATRIX IS :',/)
      WRITE (6,2100)  ((F(I,J),J=1,7),I=1,7)
 2100 FORMAT(7(2X,E13.6)/)
      WRITE (6,2200)
 2200 FORMAT('1','  THE FO MATRIX IS:'/)
      WRITE (6,5999) ((FO(I,J),J=1,7),I=1,7)
      WRITE (6,3000) CL1
 3000 FORMAT (' CL1= ',E20.9,/)
      WRITE (6,3100) CL2
 3100 FORMAT (' CL2= ',E20.9,/)
      WRITE (6,3200) L
 3200 FORMAT (' L  = ',E20.9,/)
      WRITE (6,3300) AL1
 3300 FORMAT (' AL1= ',E20.9,/)
      WRITE(6,45)
   45 FORMAT('1','THE K MATRIX:'//)
      WRITE(6,5999)(((K(I,J)),J=1,7),I=1,7)
C
C
C  THE OPTIMUM FEEDBACK CONTROL IS NOW OBTAINED
C
C
      CALL GMPRD(RINV,GTRAN,BUFF1,3,3,7)
```

```
      CALL GMPRD(BUFF1,K,DUM1,3,7,7)
      CALL MADD(X,XR,EX,7,1,1.0,-1.0)
      CALL GMPRD(DUM1,EX,ZZ,3,7,1)
      CALL MADD(UR,ZZ,UOP,3,1,1.0,-1.0)
      WRITE(6,5131) ((UOP(I)),I=1,3)
 5131 FORMAT('1','    UOP BEFORE LIMITS:'/' ',3(5X,F13.6))
      IF(UOP(3).LT.0.0) UOP(3)=0.0
       IF(UOP(3).GT.1.0) UOP(3)=1.0
      IF((X(2).GT.PHIL).AND.(UOP(2).GT.0.0)) UOP(2)=0.0
      IF((X(2).LT.(-PHIL)).AND.(UOP(2).LT.0.0)) UOP(2)=0.0
      IF((X(1).GT.0.61).AND.(UOP(1).GT.0.0)) UOP(1)=C.
      IF((X(1).LT.(-0.2)).AND.(UOP(1).LT.0.0)) UOP(1)=C.C
      WRITE(6,2234)  ((UOP(I)),I=1,3)
 2234 FORMAT('-','   UOP AFTER LIMITS:'/3(2X,F13.6))
 5999 FORMAT('0',7 (3X,F13.6))
C
C
C
        CALL GMPRD(FT,K,DUM1,7,7,7)
      CALL GMPRD(K,F,DUM2,7,7,7)
      CALL GMPRD(K,G,BUFF4,7,7,3)
      CALL GMPRD(BUFF4,RINV,BUFF3,7,3,3)
      CALL GMPRD(BUFF3,GTRAN,BUF,7,3,7)
      CALL GMPRD(BUF,K,BUF88,7,7,7)
      CALL MADD(Q,DUM1,DUM,7,7,1.0,1.0)
      CALL MADD(DUM,DUM2,DUM1,7,7,1.0,1.0)
      CALL MADD(DUM1,BUF88,DUM,7,7,1.0,-1.0)
      WRITE(6,7777)
 7777 FORMAT('1',' TEST OF GOODNESS OF K'/
     -      ' ','THE FOLLOWING MATRIX SHOULD BE APPROX. ZERO')
      WRITE(6,5999) (((DUM(I,J)),J=1,7),I=1,7)
C
C
C  COMPUTE U BY AUTO-PILOT FUNCTIONS
C
C
      ZERROR=XR(7)-X(7)
      IF(ZERROR.GT.100.0) ZERROR=100.0
      IF(ZERROR.LT.-100.0) ZERROR=-100.0
      UPLT(1)=0.001*ZERROR-0.0010*XDOT(7)-4.0*XDOT(4)
      IF((X(1).GT.0.2).AND.(UPLT(1).GT.0.0)) UPLT(1)=0.0
      IF((X(1).LT.-0.2).AND.(UPLT(1).LT.0.0)) UPLT(1)=0.0
      PSICK=X(3)
      IF(PSICK.GT.4.7123) PSICK=PSICK-6.2831
      PERROR=XR(3)-PSICK
      UPLT(2)=-0.1*PERROR+2.2*XDOT(3)
      UPLT(2)=(-UPLT(2))
      IF((X(2).GT.PHIL).AND.(UPLT(2).GT.0.0)) UPLT(2)=0.0
      IF((X(2).LT.(-PHIL)).AND.(UPLT(2).LT.0.0)) UPLT(2)=0.C
      VERROR=X(5)-XR(5)
      UPLT3=(-C.001)*(VERROR+120.0*XDOT(5))
      UPLT(3)=UPLT(3)+UPLT3*DT
      IF(UPLT(3).LT.0.0) UPLT(3)=0.0
      IF(UPLT(3).GT.1.0) UPLT(3)=1.0
      WRITE(6,3007) UPLT(1),UPLT(2),UPLT(3)
 3007 FORMAT('- ',5X,'THE NON-OPTIMAL CONTROL IS:',3(10X,F13.6))
C
C
C
C            COMPUTATION OF CPM
C
C
```

```
      DO 5122 IU=1,3
      UE(IU)=UPLT(IU)-UDP(IU)
 5122 CONTINUE
      CALL GMPRD(R,UE,ZZ,3,3,1)
      CPM=0.5*(UE(1)*ZZ(1)+UE(2)*ZZ(2)+UE(3)*ZZ(3))
      IF(ISWT.LT.0) GO TO 5211
      WRITE(6,1007) CPM
 1007 FORMAT('-',10X,'CONT. PERFORMANCE MEASURE = ',F13.6)
      WRITE(6,1067) UE
 1067 FORMAT('-',5X,'THE CONTROL ERROR IS:'/'0',5X,3(5X,F13.6))
 5211 CONTINUE
      DO 17 M3=1,3
      IF(ISWT.LT.0) U(M3)=UDP(M3)
      IF(ISWT.GT.00) U(M3)=UPLT(M3)
   17 CONTINUE
C
C
C  THIS SEGMENT CALCULATES THE NEW STATE VECTOR
C  FROM THE STATE EQUATION
C
C
      XEDOT=X(5)*COS(X(4))*COS(X(3))
      XE=XE+XEDOT*DT
      CALL GMPRD(F,X,FX,7,7,1)
      CALL GMPRD(G,U,GU,7,3,1)
      CALL MADD(FX,GU,XDOT,7,1,1.0,1.0)
      CALL MADD(X,XDOT,X,7,1,1.0,DT)
C
C
C         OUTPUT DELTA STATE AND NEW STATE
C
C
      WRITE(6,1666) ((XDOT(I)),I=1,7),XEDOT
 1666 FORMAT('-',5X,'DELTA STATE VECTOR IS:'/'0',8(5X,F13.6))
      TTIME=TTIME+DT
   40 WRITE(6,1071)
 1071 FORMAT('0',5X,'TIME',13X,'X1',13X,'X2',13X,'X3',
     -       13X,'X4',13X,'X5',13X,'X6',13X,'X7',13X,'XE'///)
      WRITE(6,1079)TTIME,((X(I)),I=1,7),XE
 1079 FORMAT('0',3X,F8.1,7 (2X,E13.6),2X,E13.6)
  999 CONTINUE
  888 CONTINUE
      STOP
      END
      SUBROUTINE GMPRD(A,B,R,N,M,L)
      DIMENSION A(1),B(1),R(1)
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

```
      SUBROUTINE MADD(A,B,C,NR,NC,FCTR1,FCTR2)
      DIMENSION A(NR,NC),B(NR,NC),C(NR,NC)
      DO 10 I=1,NR
      DO 10 J=1,NC
      C(I,J)=FCTR1*A(I,J)+FCTR2*B(I,J)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MINV(A,N,D,L,M)
      DOUBLE PRECISION A,D,BIGA,HOLD
      DIMENSION A(1),L(1),M(1)
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C       INTERCHANGE ROWS
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI)=HOLD
C       INTERCHANGE COLUMNS
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI)=HOLD
C       DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C       CONTAINED IN BIGA)
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C       REDUCE MATRIX
      DO 65 I=1,N
      IK=NK+I
```

```
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C       DIVIDE ROW BY PIVOT
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C       PRODUCT OF PIVOTS
      D=D*BIGA
C       REPLACE PIVOT BY RECIPROCAL
      A(KK)=1.0/BIGA
   80 CONTINUE
C       FINAL ROW AND COLUMN INTERCHANGE
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI)=HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI)=HOLD
      GO TO 100
  150 RETURN
      END
```

APPENDIX C

AN ITERATIVE TECHNIQUE FOR SOLUTION OF ALGEBRAIC RICCATI EQUATIONS

An iterative technique for computing an approximate solution to the algebraic matrix Riccati equation:

$$Q + F^T K + KF - K GR^{-1} G^T K = 0 \qquad \text{(C-1)}$$

is described in this section.

This equation arose in the development of the approximate optimal control law for Segment 4 of the mission, as presented in Section 5.2. The technique presented is an extension of Newton's method first developed by Kleinman (1968 and 1970) and extended by Sandell (1974) for algebraic Riccati equations. The technique calculates an approximate solution to the feedback gain matrix K which was implemented in the computer program as described in Appendix B.

1.0 CONDITIONS FOR A UNIQUE POSITIVE DEFINITE SOLUTION TO THE ALGEBRAIC RICCATI EQUATION

As in Section 5, it is assumed that the constant matrices R and Q are chosen such that they are both positive definite symmetric matrices. The matrix K is defined to be a (7 x 7) symmetric matrix. It has been shown (Bryson and Ho, 1967, Athans and Falb, 1966, and Anderson and Moore, 1971) that a unique positive definite solution exists for Equation C-1 if the matrix pair F, G are controllable, i.e., if for the system of Section 5.1 the (7 x 21) matrix

$$M = \left[G \;\vdots\; FG \;\vdots\; \ldots \;\vdots\; F^6 G \right] ,$$

has rank 7. For the state equations of Section 5.1, it was assumed that this condition could be met for every value of X in the range of interest.

2.0 NEWTON'S METHOD

Kleinman (1968) has applied Newton's method in function space to develop an iterative solution to the algebraic matrix Riccati equation. A summary of the results is presented. Kleinman (1968) contains the formal proofs for this technique.

Let K_i, $i = 0, 1, 2, \ldots$ be the sequence of unique positive definite symmetric matrix solutions to the matrix equation:

$$0 = F_i^T K_i + K_i F_i + Q + T_i^T R T_i \tag{C-2}$$

where T_i is the matrix computed from

$$T_i = R^{-1} G^T K_{i-1} \tag{C-3}$$

and F_i is the matrix computed from

$$F_i = F - G T_i \tag{C-4}$$

Choose a starting matrix T_0 such that the matrix F_0, i.e.,

$$F_0 = F - G T_0 \tag{C-5}$$

has all of its eigenvalues with negative real parts. Then the sequence of solutions K_i will approach the unique positive definite symmetric solution K to Equation C-1 in the limit as the index i tends toward infinity (Kleinman, 1968), that is

$$\lim_{i \to \infty} K_i = K \tag{C-6}$$

Thus an approximation to the matrix K can be found by iterating through the above procedure a finite number of times.

A starting value for the matrix T_0 can be generated in the following manner (Kleinman, 1970, and Sandell, 1974). Let T_0 be given by

$$T_0 = G^T B^{-1} \tag{C-7}$$

where the matrix B is given by

$$B = \int_0^{t_1} e^{-Ft} GG^T e^{-F^T t} dt \tag{C-8}$$

The upper limit of integration, time t_1, is arbitrary. The matrix B can be computed by using the series expansion for the exponential term in Equation (C-8), i.e.,

$$e^{-Ft} = \sum_{k=0}^{\infty} \frac{1}{k!} (-F)^k t^k \tag{C-9}$$

Equation C-9 is truncated at some appropriate number of terms and Equation C-8 is numerically integrated. Since the system is assumed controllable, the matrix B will have an inverse and the value of T_0 will yield an F_0 which has all of its eigenvalues with negative real parts.

In order to use this technique, Equation C-2 must be solved for K_i for each iteration. A series solution to Equation C-2 was developed for this contract based on a technique proposed by Smith (1968) to solve the matrix equation XA + BX = C. Let

$$C_i = Q + T_i^T R T_i \tag{C-10}$$

Then Equation C-2 becomes:

$$F_i^T K_i + K_i F_i = -C_i \tag{C-11}$$

Adding and subtracting from the lefthand side of Equation C-11 a factor ϵ times the matrix K_i, where ϵ is a positive scalar parameter, Equation C-11 can be put into the form of

$$[F_i^T + \epsilon I] K_i + K_i [F_i - \epsilon I] = -C_i \tag{C-12}$$

where I is the identity matrix. Multiply both sides of Equation C-12 by the inverse of $[F_i - \epsilon I]$. Since F_i has negative eigenvalues (Kleinman, 1968), ϵ can be chosen such that the indicated inverse exists. Then Equation C-12 can be put into the form

$$K_i - [\epsilon I + F_i^T] K_i [\epsilon I - F_i]^{-1} = C_i [\epsilon I - F_i]^{-1} \tag{C-13}$$

A necessary and sufficient condition that Equation C-13 has a unique solution for each C_i is that each F_i be a Hurwitzian matrix, i.e., all eigenvalues have negative real parts (Smith, 1968). Since F_i will have all its eigenvalues with negative real parts, this condition is met. A series solution for K_i which is convergent is given by

$$K_i = \sum_{j-1}^{\infty} [\epsilon I + F_i^T]^{(j-1)} C_i[(\epsilon I - F_i)^{-1}]^j$$

3.0 COMMENTS ON THE ITERATIVE SOLUTION TECHNIQUE

In order to implement the computation technique on a digital computer, several approximations are required. In order to obtain a starting value T_0 for the method, the integral in Equation C-8 must be obtained. A series solution given by Equation C-9 is used for the exponential. The number of terms in this series must be selected. For the computer program discussed in Appendix B, the series used 11 terms. An integration scheme must be used to calculate matrix B. For the computer program described in Appendix B, the integration time interval was from 0 to 1.5 seconds and rectangular integration with a step size of 0.1 second was used. In order to solve Equation C-2 for K_i, the series solution given by Equation C-14 is used. Kleinman recommends ten iterations for a good approximation. For this method, as a first cut, we have used 20 iterations through the total method, i.e., $i = 20$.

The sequence of K_i's, that is, K_0, K_1, K_2, ... is monotonically convergent from above, it is also quadratically convergent. Once an F_0 matrix has been found which has negative real parts for its eigenvalues, then so does the F_1 matrix, the F_2 matrix, the F_3 matrix, etc. In order to use Equation C-14 as an approximate solution for the K_i matrix, a value of ϵ must be chosen. For the computer program described in Appendix B, the value of ϵ is chosen as 20. With only limited experience, it was found that ϵ had to be chosen greater than one in order for the computer subroutine that calculates the inverse to be numerically well behaved. Further investigation into the choice of parameters, the number of terms that should be taken in each series, the value of the starting matrix B is required to decrease computation time. This technique, however, can be used on line. After further investigation it should be possible to simplify this method to generate a good approximation to the matrix K while requiring shorter computation time. Figure C-8 shows a flow chart for the iterative solution to the algebraic Riccati equation.

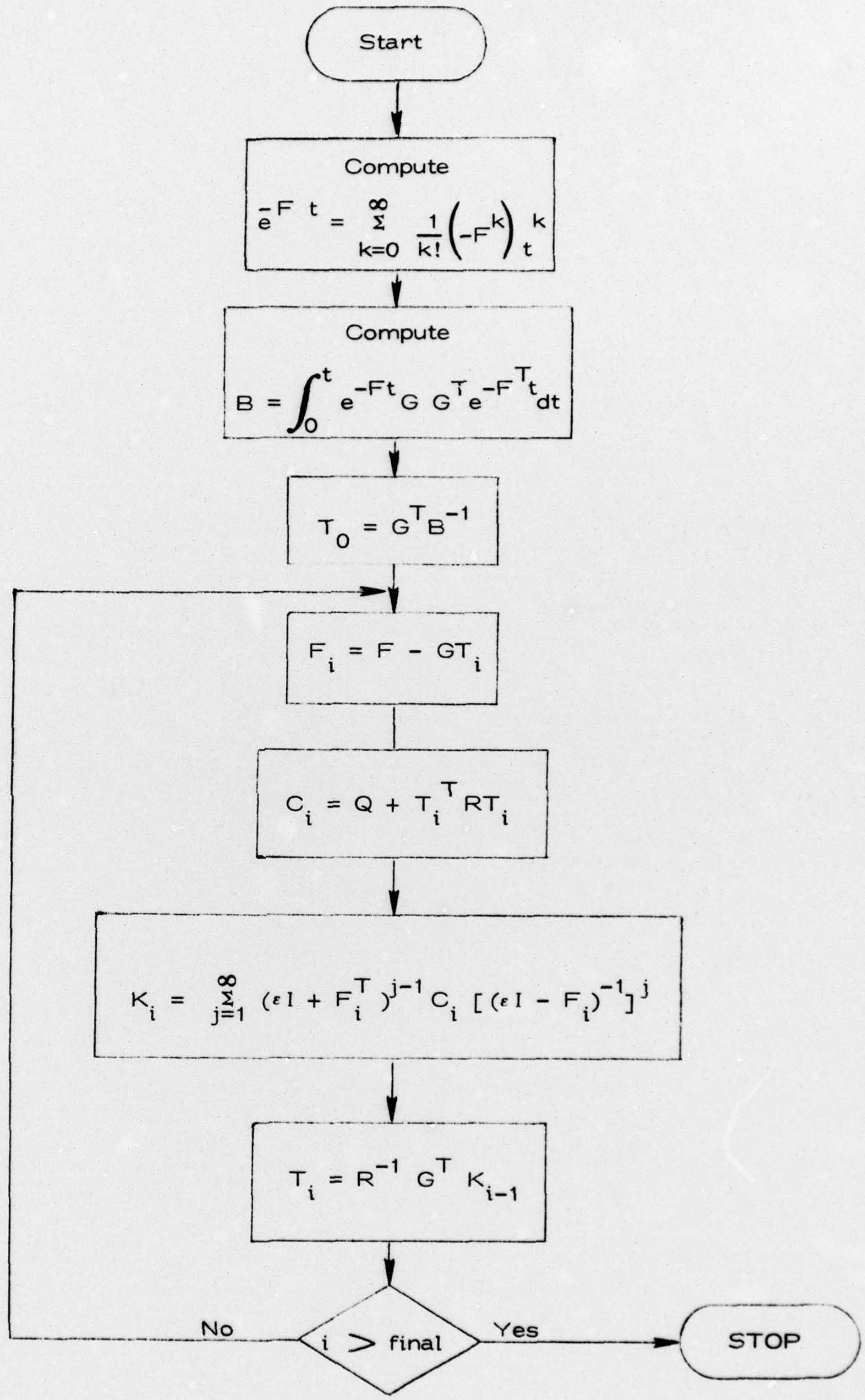


FIGURE C-1 ALGORITHM FLOW CHART

☆ U. S. GOVERNMENT PRINTING OFFICE: 1977 — 757-001/461