MEMORANDUM
RM-5753-PR
OCTOBER 1969

TACTICS: A THREE-BODY THREE-DIMENSIONAL INTERCEPT SIMULATION PROGRAM

J. H. Hutcheson and R. L. Sugden

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA - CALIFORNIA

THE DOCUMENT MAY NOT BE COPIED FOR PRIVATE USE EXCEPT AS AUTHORIZED BY RAND.
MEMORANDUM
RM-5759-PR
OCTOBER 1969

TACTICS: A THREE-BODY,
THREE-DIMENSIONAL
INTERCEPT SIMULATION PROGRAM
J. H. Hutcheson and R. L. Segerblom

This research is supported by the United States Air Force under Project RAND—Con-tract No. F 162947-C-6015—monitored by the Directorate of Operational Requirements
and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF.
Views or conclusions contained in this study should not be interpreted as representing
the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90403
This study is presented as a competent treatment of the subject, worthy of publication. The RAND Corporation vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the authors.

Published by The RAND Corporation
This Memorandum contains descriptive material and reference information necessary to understand and use TACTICS, a computer program which mathematically simulates the flight trajectories of as many as three different vehicles simultaneously. The program has been in use at Rand and elsewhere for about a year and a half, principally in studies of aircraft and missile performance in air-to-air combat and in air-to-surface missile (ASM) and surface-to-air missile (SAM) simulations. Because the simulation model is primarily useful as a research tool for studying interceptor-target guidance and interceptor trajectories in general, emphasis has been placed on providing versatility and flexibility for solving a wide variety of problems.

Magnetic tape copies of the program have been sent upon request to various facilities of the USAF and USN and contractors engaged in research in related fields, as well as to the armed forces (or affiliated institutions) of Japan, Canada, and Germany. The descriptive material should be of interest to those concerned with related fields of effort, while the reference information is necessary for a thorough understanding and effective use of the model.
SUMMARY

TACTICS is a computer program written in FORTRAN IV which mathematically simulates the dynamics of flight in three-dimensional space of as many as three vehicles simultaneously. The purpose of this Memorandum is to acquaint prospective users with the capabilities and basic theory of the program and to serve as a reference manual for those who wish to use the program.

The first part of the Memorandum contains the description and theory of operation and is oriented toward those with a mathematical or technical background. The second part is concerned with how to use the program, i.e., how to provide input data, select options, and develop a flight program. Wherever possible, FORTRAN symbols are relegated to this second part. A number of illustrative examples of a wide variety of problems are given in detail, including data and FORTRAN listings, in order to facilitate the use of TACTICS and the obtaining of results without detailed knowledge of the inner workings of the program. In many cases it should be possible to set up a specialized problem by modifying or combining various features of the examples.
ACKNOWLEDGMENTS

In developing TACTICS, the authors were fortunate in being able to draw upon the many varied experiences of the Rand staff. Since necessity is the mother of invention, credit for the initial concept is due to those who defined the needs so clearly and later contributed many ideas and suggestions leading to improvements. Although there were many, those principally involved were B. Boehm, T. F. Burke, T. B. Garber, T. E. Greene, J. Huntzicker, and D. N. Morris. Others responsible for translating ideas or concepts into computer language, inventing sample problems, cross-checking results, and debugging include J. Bedell, C. Fleming, J. Jolissaint, N. Maguire, and M. Samaniego. L. G. Martin and R. Spicer deserve special credit for many constructive suggestions and for inventing problem runs that were particularly effective in leading to improvements.
CONTENTS

PREFACE .............................................................. iii
SUMMARY ................................................................... v
ACKNOWLEDGMENTS ................................................... vii
LIST OF FIGURES ....................................................... xi
LIST OF SYMBOLS ..................................................... xiii

PART 1
DESCRIPTION AND THEORY OF OPERATION

Section
I. INTRODUCTION .................................................... 3
II. POLICY SUBROUTINE .............................................. 9
III. INPUT-OUTPUT GEOMETRY ..................................... 11
    Absolute and Relative Position Geometry ................. 11
    Absolute and Relative Velocity Geometry ................ 13
    Elevation, Azimuth, and Bearing-Angle Geometry .......... 16
IV. DEFINING THE FORCES AND ACCELERATIONS .......... 20
    Control Laws ....................................................... 20
    Lateral Acceleration and Its Components .................. 21
    Forces and Force Resolution .................................. 23
    Aerodynamic Forces ............................................. 29
    Propulsive Forces ................................................ 30
    Gravitational Forces ............................................. 30
V. AERODYNAMIC, STRUCTURAL, AND TIME-LAG CONSTRAINTS .... 32
    Aerodynamic and Structural Constraints ................. 32
    Time Lag .......................................................... 34
VI. GUIDANCE AND CONTROL LAWS ............................. 36
    Open-Loop Control Laws ...................................... 37
    Closed-Loop Control Laws .................................... 38
VII. CONCLUSIONS .................................................. 40

Part 2
OPERATING THE PROGRAM

VIII. INTRODUCTION .................................................. 43
IX. FORMULATING A POLICY SUBROUTINE ........................................... 44
   Stopping the Program ....................................................... 46
   Integration ........................................................................ 47
   Number of Vehicles Used ..................................................... 47
   Time Lags ........................................................................... 47
   Recall ................................................................................ 48
   Restore ............................................................................... 48
   Round Earth, Rotating or Nonrotating ..................................... 48

X. INPUT FORM AND ORDER OF INPUT .............................................. 50
   Aerodynamic and Propulsion Tables ....................................... 50
   Aerodynamics and Propulsion Without Tables ...................... 54
   Main Set of Data ................................................................... 55
   Running More Than One Case ............................................... 57

XI. OUTPUT ............................................................................. 58

XII. EXAMPLES ........................................................................... 63
    Example 1: Aircraft Maneuvers, No Missile ...................... 63
    Example 2: Launching One Missile ................................. 66
    Example 3: Ground-Launched Missile ............................. 73
    Example 4: Launching Two Missiles, Recall .................... 80
    Example 5: Using Restore Feature to Fire 20-mm Projectiles 86

XIII. NUMERICAL INTEGRATION ...................................................... 94
    Modes of Operation ............................................................. 94
    Integration Accuracy ........................................................... 95
    Fixed-Step Approach to Launch ......................................... 95
    Miss-Distance Computation .............................................. 96
    Initialization ....................................................................... 97

Appendix
    A. AIRCRAFT ATTITUDE AND ORIENTATION ANGLES .............. 99
    B. INTEGRATING THE EQUATIONS OF MOTION ....................... 105
    C. GUIDANCE-AND-CONTROL-LAW DEFINITIONS ................... 116
    D. INSTRUCTIONS FOR CALLING OPTIONAL SUBROUTINES ........ 139
    E. DESCRIPTION OF INPUT DATA ......................................... 142
    F. AERODYNAMIC AND PROPULSION TABLES .......................... 150
    G. PROGRAM SUBROUTINES ................................................ 157
    H. FORTRAN FLAGS ............................................................. 159
    I. MODEL ATMOSPHERE ..................................................... 163

REFERENCES ............................................................................ 165
FIGURES

1. Recall and restore features contrasted ........................................ 6
2. Basic framework for flight simulation (generalized) ......................... 7
3. Absolute and relative position geometry ...................................... 12
4. Absolute geometry ..................................................................... 14
5. Unit vectors normal to velocity vector ......................................... 17
6. Elevation, azimuth, and bearing-angle geometry .............................. 19
7. Coordinated turn in level flight .................................................... 24
8. Bank angle and lateral acceleration components ............................... 26
9. Input form .................................................................................. 51
10. Cards specifying main data .......................................................... 52
11. Order of input ............................................................................. 53
12. Sample initial data values (unlabeled) ......................................... 59
13. Sample initial data values (labeled) ............................................. 60
14. Sample page of output for NPRINT = 3 ........................................ 61
15. Sample POLICY subroutine for aircraft maneuvers, no missiles ......... 65
16. Three-dimensional diagram of initial conditions for Example 1 ......... 67
17. Input data for Example 1 ............................................................. 68
18. Sample POLICY subroutine for launching one missile .................. 70
19. Three-dimensional diagram of initial conditions for Example 2 ......... 72
20. Input data for Example 2 ............................................................. 74
21. Sample POLICY subroutine for ground-launched missile ............... 76
22. Three-dimensional diagram of initial conditions for Example 3 ........ 77
23. Input data for Example 3 ............................................................. 79
24. Sample POLICY subroutine for launching two missiles ................. 82
25. Three-dimensional diagram of initial conditions for Example 4 ......... 85
26. Input data for Example 4 ............................................................. 87
27. Sample POLICY subroutine for firing 20-mm projectiles ................. 89
28. Three-dimensional diagram of initial conditions for Example 5 .......... 91
29. Input data for Example 5 ............................................................. 92
30. Azimuth and elevation coordination transformation ........ 101
31. Roll-angle geometry .................................... 103
32. Geocentric coordinate system .............................. 108
33. Topocentric coordinate system ............................. 112
34. Azimuth-elevation topocentric coordinate system ........ 114
35. F-104 aerodynamic tables .................................. 152
### SYMBOLS

<table>
<thead>
<tr>
<th>FORTRAN Notation</th>
<th>Symbol*</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(I,J)</td>
<td>( \bar{i}_A )</td>
<td>Unit vector normal to velocity vector ( V ) and also in the horizontal plane</td>
</tr>
<tr>
<td>AB( \theta )ST(I)</td>
<td>( a_B )</td>
<td>Boost acceleration, assumed to be an average value ( (\text{ft/sec}^2) )</td>
</tr>
<tr>
<td>AC( \theta )M(I)</td>
<td>( a_C )</td>
<td>Absolute magnitude of vehicle lateral acceleration ( (\text{ft/sec}^2) ), printout in g's</td>
</tr>
<tr>
<td>AC( \theta )MA(I)</td>
<td>( a_{Ch} )</td>
<td>Horizontal component of lateral acceleration (g's)</td>
</tr>
<tr>
<td>AC( \theta )MD(I)</td>
<td>( a_{CV} )</td>
<td>Vertical component of lateral acceleration of vehicle(i) (g's)</td>
</tr>
<tr>
<td>ALPHA(I)</td>
<td>( \alpha )</td>
<td>Angle of attack (deg)</td>
</tr>
<tr>
<td>ALPHAO(I)</td>
<td>( \alpha_o )</td>
<td>Zero-lift angle of attack (deg)</td>
</tr>
<tr>
<td>ALT(I)</td>
<td>( h )</td>
<td>Altitude of the vehicle (ft)</td>
</tr>
<tr>
<td>A( \theta )UT(I)</td>
<td>( a_o )</td>
<td>Absolute magnitude of output lateral acceleration of vehicle (i) (g's)</td>
</tr>
<tr>
<td>A( \theta )UTA(I)</td>
<td>( a_{oh} )</td>
<td>Horizontal component of output lateral acceleration of vehicle (i) (g's)</td>
</tr>
<tr>
<td>A( \theta )UTD(I)</td>
<td>( a_{ov} )</td>
<td>Vertical component of output lateral acceleration of vehicle (i) (g's)</td>
</tr>
<tr>
<td>AREA(I)</td>
<td>( A )</td>
<td>Reference area ( (\text{ft}^2) )</td>
</tr>
<tr>
<td>AS( \theta )MAX(I)</td>
<td>( a_{S\text{max}} )</td>
<td>Structural lateral acceleration limit (g's)</td>
</tr>
<tr>
<td>AZ( \theta )MAX(I)</td>
<td>( n_{\text{max}} )</td>
<td>Maximum azimuth angle (global limit) (deg)</td>
</tr>
<tr>
<td>AZMUTH(I)</td>
<td>( n_{12} )</td>
<td>Azimuth angle in aircraft coordinates, vehicle 2 with respect to vehicle 1 (deg, + for right, - for left)</td>
</tr>
<tr>
<td>A1(I,J)</td>
<td>( \bar{i}_{A1} )</td>
<td>Unit vector normal to the axis of the aircraft and also in the horizontal plane</td>
</tr>
<tr>
<td>BANK(I)</td>
<td>( \psi_B )</td>
<td>Aircraft bank angle defined in relation to wind axes system (see Figs. 7 and 8) (deg)</td>
</tr>
<tr>
<td>BC( \theta )N(I)</td>
<td>( dC_D/d(C_L^2) )</td>
<td>Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient</td>
</tr>
</tbody>
</table>

*Subscripts "i" which indicate vehicle numbers have been omitted throughout the list for notational convenience.*
<table>
<thead>
<tr>
<th>FORTRAN Notation</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEARMG(1)</td>
<td>$B_{12}$</td>
<td>Bearing angle in aircraft coordinates, vehicle 2 with respect to vehicle 1 (deg)</td>
</tr>
<tr>
<td>BEARMG(2)</td>
<td>$B_{13}$</td>
<td>Bearing angle in aircraft coordinates, vehicle 3 with respect to vehicle 1 (deg)</td>
</tr>
<tr>
<td>BEARMG(3)</td>
<td>$B_{23}$</td>
<td>Bearing angle in aircraft coordinates, vehicle 3 with respect to vehicle 2 (deg)</td>
</tr>
<tr>
<td>BEARMG(4)</td>
<td>$B_{21}$</td>
<td>Bearing angle in aircraft coordinates, vehicle 1 with respect to vehicle 2 (deg)</td>
</tr>
<tr>
<td>BEARMG(5)</td>
<td>$B_{31}$</td>
<td>Bearing angle in aircraft coordinates, vehicle 1 with respect to vehicle 3 (deg)</td>
</tr>
<tr>
<td>BEARMG(6)</td>
<td>$B_{32}$</td>
<td>Bearing angle in aircraft coordinates, vehicle 2 with respect to vehicle 3 (deg)</td>
</tr>
<tr>
<td>BETA(I)</td>
<td>$\beta$</td>
<td>Ballistic coefficient (lb/ft$^2$)</td>
</tr>
<tr>
<td>CDOCYN(I)</td>
<td>$C_D$</td>
<td>Zero-lift drag coefficient</td>
</tr>
<tr>
<td>CLMAX(I)</td>
<td>$C_{\text{L}_{\text{max}}}$</td>
<td>Maximum aerodynamic lift coefficient</td>
</tr>
<tr>
<td>CDSTRA(I)</td>
<td>$C_D$</td>
<td>Aerodynamic drag coefficient</td>
</tr>
<tr>
<td>CSLIFT(I)</td>
<td>$C_L$</td>
<td>Aerodynamic lift coefficient</td>
</tr>
<tr>
<td>CWDCT(I)</td>
<td>$\dot{W}$</td>
<td>Time rate of change of weight (lb/sec)</td>
</tr>
<tr>
<td>D(I,J)</td>
<td>$\hat{l}_D$</td>
<td>Unit vector normal to the velocity vector and normal to vector $\hat{l}_A$, forming a right-hand system</td>
</tr>
<tr>
<td>DATA(200)</td>
<td></td>
<td>Initial-condition input data ranging from 1 to 200 (so far, only 143 are used)</td>
</tr>
<tr>
<td>DELV</td>
<td>$\Delta V$</td>
<td>Boost velocity of missile (ft/sec)</td>
</tr>
<tr>
<td>DENS(I)</td>
<td>$\rho$</td>
<td>Air density (slug/ft$^3$)</td>
</tr>
<tr>
<td>DRAG(I)</td>
<td>D</td>
<td>Drag force (lb)</td>
</tr>
<tr>
<td>DTMN</td>
<td></td>
<td>Minimum value for integration step size, used only for determination of miss distance (sec)</td>
</tr>
<tr>
<td>DT$\phi$</td>
<td></td>
<td>Starting value for integration step size (sec)</td>
</tr>
<tr>
<td>DTP$\phi$</td>
<td></td>
<td>Time interval for printing output (sec)</td>
</tr>
<tr>
<td>DVPHI(I)</td>
<td>$\Delta \gamma$</td>
<td>Assumed error in $\gamma$ for aiming (deg)</td>
</tr>
<tr>
<td>DVTTH(I)</td>
<td>$\Delta \theta$</td>
<td>Assumed error in $\theta_v$ for aiming (deg)</td>
</tr>
<tr>
<td>D1(I,J)</td>
<td>$\hat{l}_{D1}$</td>
<td>Unit vector normal to the axis of the aircraft and also in a plane normal to the horizontal plane</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ELEV(1)</td>
<td>$\mathcal{E}_{12}$</td>
<td>Elevation angle in aircraft coordinates, vehicle 2 with respect to vehicle 1 (deg, + for above, - for below)</td>
</tr>
<tr>
<td>ELEV(2)</td>
<td>$\mathcal{E}_{13}$</td>
<td>Elevation angle in aircraft coordinates, vehicle 3 with respect to vehicle 1 (deg, + for above, - for below)</td>
</tr>
<tr>
<td>ELEV(3)</td>
<td>$\mathcal{E}_{23}$</td>
<td>Elevation angle in aircraft coordinates, vehicle 3 with respect to vehicle 2 (deg, + for above, - for below)</td>
</tr>
<tr>
<td>ELEV(4)</td>
<td>$\mathcal{E}_{21}$</td>
<td>Elevation angle in aircraft coordinates, vehicle 1 with respect to vehicle 2 (deg, + for above, - for below)</td>
</tr>
<tr>
<td>ELEV(5)</td>
<td>$\mathcal{E}_{31}$</td>
<td>Elevation angle in aircraft coordinates, vehicle 1 with respect to vehicle 3 (deg, + for above, - for below)</td>
</tr>
<tr>
<td>ELEV(6)</td>
<td>$\mathcal{E}_{32}$</td>
<td>Elevation angle in aircraft coordinates, vehicle 2 with respect to vehicle 3 (deg, + for above, - for below)</td>
</tr>
<tr>
<td>ELEVMAX(1)</td>
<td>$\mathcal{E}_{\text{max}}$</td>
<td>Maximum elevation angle (gimbal limits) (deg)</td>
</tr>
<tr>
<td>EXTR(20)</td>
<td></td>
<td>Extra quantities (real) in COMMON for optional use (the first six are part of standard printout)</td>
</tr>
<tr>
<td>G</td>
<td>$g$</td>
<td>Mass conversion (32.174 ft/sec$^2$); also used as constant gravitational attraction for flat-earth option</td>
</tr>
<tr>
<td>GFORCE</td>
<td></td>
<td>Total lateral acceleration, including gravitational effects specified for a maneuver (g's)</td>
</tr>
<tr>
<td>GFORCE(I)</td>
<td>$F_n$</td>
<td>Total lateral acceleration, including gravitational effects (g's)</td>
</tr>
<tr>
<td>HM</td>
<td></td>
<td>Actual integration step size used by the program (sec)</td>
</tr>
<tr>
<td>HMIN</td>
<td></td>
<td>Minimum specified value for integration step size used in variable Adams-Moulton mode integration (sec)</td>
</tr>
<tr>
<td>HMX</td>
<td></td>
<td>Maximum specified value for integration step size used in variable Adams-Moulton mode integration (sec)</td>
</tr>
<tr>
<td>IAERØ</td>
<td></td>
<td>Flag indicating aerodynamic option</td>
</tr>
<tr>
<td>IEXTRA(10)</td>
<td></td>
<td>Ten extra integer quantities for optional use (COMMON package)</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>ILAUN</td>
<td></td>
<td>Flag indicating sequence of steps involved in missile launch (see Appendix H)</td>
</tr>
<tr>
<td>IMISS</td>
<td></td>
<td>Flag indicating that the program should continue after finding point of closest approach (miss distance)</td>
</tr>
<tr>
<td>IMPLSE(I)</td>
<td>I</td>
<td>Specific impulse of rocket motor (sec)</td>
</tr>
<tr>
<td>INERF, INERT</td>
<td></td>
<td>Flag indicating that a vehicle's velocity is in relation to a rotating or non-rotating (inertial) coordinate system</td>
</tr>
<tr>
<td>EPSLON</td>
<td>$\varepsilon$</td>
<td>Threshold value used in on-off control laws for stability</td>
</tr>
<tr>
<td>ERTEST</td>
<td></td>
<td>Maximum allowable relative truncation error for Adams-Moulton variable step size integration mode</td>
</tr>
<tr>
<td>IPRINT(20)</td>
<td></td>
<td>Flag indicating printout option</td>
</tr>
<tr>
<td>IRÔT8</td>
<td></td>
<td>Flag indicating option for rotating or nonrotating earth</td>
</tr>
<tr>
<td>ISTÔRE</td>
<td></td>
<td>Flag indicating that position and velocity values are to be stored at the time of launch</td>
</tr>
<tr>
<td>ITAU(I)</td>
<td></td>
<td>Flag indicating number of vehicle first-order time lags</td>
</tr>
<tr>
<td>ITHR</td>
<td></td>
<td>Flag indicating thrust option</td>
</tr>
<tr>
<td>JATMÔS</td>
<td></td>
<td>Flag specifying whether initial-condition value of velocity is expressed in Mach number or ft/sec</td>
</tr>
<tr>
<td>JINTEG</td>
<td></td>
<td>Flag specifying integration mode</td>
</tr>
<tr>
<td>JFÔL, KFÔL, LFÔL, NFÔL, NFÔL</td>
<td></td>
<td>Flags used in Policy subroutine</td>
</tr>
<tr>
<td>JVEH(I)</td>
<td></td>
<td>Flag indicating aerodynamic option for tables</td>
</tr>
<tr>
<td>KINTEG</td>
<td></td>
<td>Flag indicating whether a round-earth or flat-earth option is to be used</td>
</tr>
<tr>
<td>KLAUN</td>
<td></td>
<td>Decimal fraction of missile's maximum range at which it is to be launched</td>
</tr>
<tr>
<td>LAMDAO(I)</td>
<td>$\lambda_0$</td>
<td>Navigation constant for closed-loop guidance routines</td>
</tr>
<tr>
<td>LAT(I)</td>
<td>$\phi$</td>
<td>Geocentric latitude of the vehicle (deg)</td>
</tr>
<tr>
<td>LATO</td>
<td>$\phi_0$</td>
<td>Latitude of the local coordinate system origin (deg)</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>LEVEL(I)</td>
<td>L</td>
<td>Flag set in subroutine STRLVL used to communicate to POLICY that the vehicle velocity vector is in a horizontal plane within a tolerance of 0.002 rad</td>
</tr>
<tr>
<td>LIFT(I)</td>
<td>Λ</td>
<td>Aerodynamic lift force (lb)</td>
</tr>
<tr>
<td>LONG(I)</td>
<td>Λ₀</td>
<td>Longitude of the vehicle (deg)</td>
</tr>
<tr>
<td>LONGO</td>
<td></td>
<td>Longitude of the origin of the local coordinate system (deg)</td>
</tr>
<tr>
<td>MACH(I)</td>
<td>M</td>
<td>Mach number</td>
</tr>
<tr>
<td>MACHMX(I)</td>
<td>Mₘₐₓ</td>
<td>Placard limit (maximum Mach number for vehicle)</td>
</tr>
<tr>
<td>MINMR</td>
<td></td>
<td>Missile range to target within which program will automatically initiate process for miss distance computation (ft)</td>
</tr>
<tr>
<td>MODE</td>
<td></td>
<td>Flag indicating captive flight option</td>
</tr>
<tr>
<td>NPRINT</td>
<td></td>
<td>Flag indicating number of sections of output to be printed</td>
</tr>
<tr>
<td>ØMEGA(I,J)</td>
<td>ω</td>
<td>Angular rate vector of vehicle (i) with respect to x, y, z coordinate frame (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAB(1,J)</td>
<td>ωᵦ₁₂</td>
<td>Angular rate bias term used for predictive guidance, vehicle 2 with respect to vehicle 1 (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAB(2,J)</td>
<td>ωᵦ₁₃</td>
<td>Angular rate bias term used for predictive guidance, vehicle 3 with respect to vehicle 1 (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAB(3,J)</td>
<td>ωᵦ₂₃</td>
<td>Angular rate bias term used for predictive guidance, vehicle 3 with respect to vehicle 1 (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAB(4,J)</td>
<td>ωᵦ₂₁</td>
<td>Angular rate bias term used for predictive guidance, vehicle 3 with respect to vehicle 2 (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAB(5,J)</td>
<td>ωᵦ₃₁</td>
<td>Angular rate bias term used for predictive guidance, vehicle 1 with respect to vehicle 3 (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAB(6,J)</td>
<td>ωᵦ₃₂</td>
<td>Angular rate bias term used for predictive guidance, vehicle 2 with respect to vehicle 3 (rad/sec)</td>
</tr>
<tr>
<td>ØMEGAE</td>
<td>ωₑ</td>
<td>Earth's angular rate of rotation (7.29211585 × 10⁻⁵ rad/sec)</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>ÔMEGAR(1,J)</td>
<td>ω₁₂</td>
<td>Relative angular rate vector, vehicle 2 with respect to vehicle 1 (rad/sec)</td>
</tr>
<tr>
<td>ÔMEGAR(2,J)</td>
<td>ω₁₃</td>
<td>Relative angular rate vector, vehicle 3 with respect to vehicle 1 (rad/sec)</td>
</tr>
<tr>
<td>ÔMEGAR(3,J)</td>
<td>ω₂₃</td>
<td>Relative angular rate vector, vehicle 3 with respect to vehicle 2 (rad/sec)</td>
</tr>
<tr>
<td>ÔMEGAR(4,J)</td>
<td>ω₂₁</td>
<td>Relative angular rate vector, vehicle 1 with respect to vehicle 2 (rad/sec)</td>
</tr>
<tr>
<td>ÔMEGAR(5,J)</td>
<td>ω₃₁</td>
<td>Relative angular rate vector, vehicle 1 with respect to vehicle 3 (rad/sec)</td>
</tr>
<tr>
<td>ÔMEGAR(6,J)</td>
<td>ω₃₂</td>
<td>Relative angular rate vector, vehicle 2 with respect to vehicle 3 (rad/sec)</td>
</tr>
<tr>
<td>PHIDOT(1)</td>
<td>ϕ</td>
<td>Time rate of change of ϕ (rad/sec)</td>
</tr>
<tr>
<td>PRES(1)</td>
<td>p</td>
<td>Air pressure (lb/ft²)</td>
</tr>
<tr>
<td>Q(1)</td>
<td>q</td>
<td>Dynamic pressure (lb/ft²)</td>
</tr>
<tr>
<td>R(I,J)</td>
<td>r</td>
<td>Range vector from x, y, z origin to vehicle (i) (ft)</td>
</tr>
<tr>
<td>RAD</td>
<td>π/180°</td>
<td>Factor for converting degrees to radians (rad/deg)</td>
</tr>
<tr>
<td>RDOT(1)</td>
<td>r₁₂</td>
<td>Range rate, vehicle 2 with respect to vehicle 1 (ft/sec)</td>
</tr>
<tr>
<td>RDOT(2)</td>
<td>r₁₃</td>
<td>Range rate, vehicle 3 with respect to vehicle 1 (ft/sec)</td>
</tr>
<tr>
<td>RDOT(3)</td>
<td>r₂₃</td>
<td>Range rate, vehicle 3 with respect to vehicle 2 (ft/sec)</td>
</tr>
<tr>
<td>RDOT(4)</td>
<td>r₂₁</td>
<td>Range rate, vehicle 1 with respect to vehicle 2 (ft/sec)</td>
</tr>
<tr>
<td>RDOT(5)</td>
<td>r₃₁</td>
<td>Range rate, vehicle 1 with respect to vehicle 3 (ft/sec)</td>
</tr>
<tr>
<td>RDOT(6)</td>
<td>r₃₂</td>
<td>Range rate, vehicle 2 with respect to vehicle 3 (ft/sec)</td>
</tr>
<tr>
<td>ELAUN</td>
<td></td>
<td>Range at which missile is to be launched, used in LEADCL subroutine (ft)</td>
</tr>
<tr>
<td>RMTMAX</td>
<td></td>
<td>Maximum range of missile (ft)</td>
</tr>
<tr>
<td>ROLL(1)</td>
<td>ψ</td>
<td>Aircraft roll angle (deg) defined in relation to aircraft axes coordinate system</td>
</tr>
<tr>
<td>ROLLLL</td>
<td></td>
<td>Total angle through which vehicle is to roll (deg)</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>ROLLR8(I)</td>
<td>$\dot{\psi}$</td>
<td>Time rate of change of aircraft roll angle (deg/sec)</td>
</tr>
<tr>
<td>RR(I,J)</td>
<td>$\vec{R}$</td>
<td>Geocentric range vector (ft) directed radially from the earth's center</td>
</tr>
<tr>
<td>RDOT(I,J)</td>
<td>$\dot{\vec{R}}$</td>
<td>Velocity of the topocentric range vector with respect to inertial space (ft/sec)</td>
</tr>
<tr>
<td>RREL(I,J)</td>
<td>$\vec{r}_{12}$</td>
<td>Relative range vector, vehicle 2 relative to vehicle 1 (ft)</td>
</tr>
<tr>
<td>RREL(2,J)</td>
<td>$\vec{r}_{13}$</td>
<td>Relative range vector, vehicle 3 relative to vehicle 1 (ft)</td>
</tr>
<tr>
<td>RREL(3,J)</td>
<td>$\vec{r}_{23}$</td>
<td>Relative range vector, vehicle 3 relative to vehicle 2 (ft)</td>
</tr>
<tr>
<td>RREL(4,J)</td>
<td>$\vec{r}_{21}$</td>
<td>Relative range vector, vehicle 1 relative to vehicle 2 (ft)</td>
</tr>
<tr>
<td>RREL(5,J)</td>
<td>$\vec{r}_{31}$</td>
<td>Relative range vector, vehicle 1 relative to vehicle 3 (ft)</td>
</tr>
<tr>
<td>RREL(6,J)</td>
<td>$\vec{r}_{32}$</td>
<td>Relative range vector, vehicle 2 relative to vehicle 3 (ft)</td>
</tr>
<tr>
<td>RO</td>
<td>$R_0$</td>
<td>Average radius of the earth (20,902,287 ft)</td>
</tr>
<tr>
<td>RO(I)</td>
<td>$\vec{R}_0$</td>
<td>Position of the local x, y, z coordinate system origin with respect to inertial space (ft)</td>
</tr>
<tr>
<td>RDOT(I)</td>
<td>$\dot{\vec{R}}_0$</td>
<td>Velocity of the local x, y, z coordinate system with respect to inertial space (ft/sec)</td>
</tr>
<tr>
<td>SGAMA(I)</td>
<td>$\sigma$</td>
<td>Absolute magnitude of the angle between the velocity vector of the interceptor and the line of sight from interceptor to target (rad)</td>
</tr>
<tr>
<td>SLOPE(I)</td>
<td>$dC_L/ds$</td>
<td>Slope of $C_L$ versus a curve</td>
</tr>
<tr>
<td>SOUND(I)</td>
<td>$s$</td>
<td>Speed of sound (ft/sec)</td>
</tr>
<tr>
<td>TABCON(I)</td>
<td>$T_{ab}$</td>
<td>Constants to be used for afterburner thrust (lb)</td>
</tr>
<tr>
<td>Tau(I,J)</td>
<td>$\tau$</td>
<td>First-order approximation for overall missile response time</td>
</tr>
<tr>
<td>TBURN1</td>
<td></td>
<td>First-stage burning time (sec)</td>
</tr>
<tr>
<td>TBURN2</td>
<td></td>
<td>Second-stage burning time (sec)</td>
</tr>
<tr>
<td>Temp(I)</td>
<td>$T$</td>
<td>Temperature (degrees Fahrenheit)</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>TGUIDE(I)</td>
<td>T_M</td>
<td>Time interval missile is to fly un-guided after launch (sec)</td>
</tr>
<tr>
<td>THCM(I)</td>
<td>T</td>
<td>Constants to be used for military thrust (lb)</td>
</tr>
<tr>
<td>THD(I)</td>
<td>( \hat{\theta} )</td>
<td>Time rate of change of theta (rad/sec)</td>
</tr>
<tr>
<td>THRTRL(I)</td>
<td>K</td>
<td>Constant used to multiply thrust to represent throttle setting</td>
</tr>
<tr>
<td>THRUST(I)</td>
<td>T</td>
<td>Propulsive thrust force (lb)</td>
</tr>
<tr>
<td>TIME</td>
<td>t</td>
<td>Running time (sec)</td>
</tr>
<tr>
<td>TLAUN(I)</td>
<td>t_L</td>
<td>Launch time (sec)</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>Time value at which program is to stop</td>
</tr>
<tr>
<td>UNITL(I,J)</td>
<td>( \bar{l}_{L} )</td>
<td>Unit vector directed along the lift vector</td>
</tr>
<tr>
<td>UNITLL(I,J)</td>
<td>( \bar{l}_{LL} )</td>
<td>Unit vector directed eastward along the local parallel of latitude</td>
</tr>
<tr>
<td>UNITPP(I,J)</td>
<td>( \bar{l}_{P} )</td>
<td>Unit vector directed northward along the local meridian of longitude</td>
</tr>
<tr>
<td>UNITPV(I,J)</td>
<td>( \bar{l}_{P} )</td>
<td>Unit vector normal to velocity vector ( \bar{V} ) and along the net lateral acceleration vector</td>
</tr>
<tr>
<td>UNITR(K,J)</td>
<td>( \bar{r}_{K} )</td>
<td>Unit vector directed along the relative range vectors and using the same subscript notation</td>
</tr>
<tr>
<td>UNITRR(I,J)</td>
<td>( \bar{r}_{R} )</td>
<td>Unit vector directed radially from the earth's center along the vector ( \bar{R} ) (RR(I,J))</td>
</tr>
<tr>
<td>UNITT(I,J)</td>
<td>( \bar{t}_{T} )</td>
<td>Unit vector directed along the longitudinal axis of the vehicle (i) and also assumed to be coincident with the thrust vector ( \bar{T} )</td>
</tr>
<tr>
<td>UNITV(I,J)</td>
<td>( \bar{v}_{V} )</td>
<td>Unit vector corrected along velocity vector ( \bar{V} )</td>
</tr>
<tr>
<td>V(I,J)</td>
<td>( \bar{v} )</td>
<td>Velocity vector (ft/sec)</td>
</tr>
<tr>
<td>VD( \dot{\theta} )(I)</td>
<td>( \dot{\bar{V}} )</td>
<td>Rate of acceleration or deceleration along the trajectory, i.e., changes in speed (ft/sec²)</td>
</tr>
<tr>
<td>VREL(1,J)</td>
<td>( \bar{v}_{12} )</td>
<td>Relative velocity, vehicle 2 with respect to vehicle 1 (ft/sec)</td>
</tr>
<tr>
<td>VREL(2,J)</td>
<td>( \bar{v}_{13} )</td>
<td>Relative velocity, vehicle 3 with respect to vehicle 1 (ft/sec)</td>
</tr>
<tr>
<td>FORTRAN Notation</td>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>VREL(3,J)</td>
<td>$\overrightarrow{V}_{23}$</td>
<td>Relative velocity, vehicle 3 with respect to vehicle 2 (ft/sec)</td>
</tr>
<tr>
<td>VREL(4,J)</td>
<td>$\overrightarrow{V}_{21}$</td>
<td>Relative velocity, vehicle 1 with respect to vehicle 2 (ft/sec)</td>
</tr>
<tr>
<td>VREL(5,J)</td>
<td>$\overrightarrow{V}_{31}$</td>
<td>Relative velocity, vehicle 1 with respect to vehicle 3 (ft/sec)</td>
</tr>
<tr>
<td>VREL(6,J)</td>
<td>$\overrightarrow{V}_{32}$</td>
<td>Relative velocity, vehicle 2 with respect to vehicle 3 (ft/sec)</td>
</tr>
<tr>
<td>WBURN(I)</td>
<td>$W_B$</td>
<td>Weight of missile at burnout (lb)</td>
</tr>
<tr>
<td>WDT(I)</td>
<td>$\dot{W}$</td>
<td>Time rate of change of weight (lb/sec)</td>
</tr>
<tr>
<td>WEIGHT(I)</td>
<td>$W$</td>
<td>Weight (lb)</td>
</tr>
<tr>
<td>W0(I)</td>
<td>$W_0$</td>
<td>Initial weight of vehicle (i) (lb)</td>
</tr>
</tbody>
</table>
Part 1

DESCRIPTION AND THEORY OF OPERATION
I. INTRODUCTION

TACTICS is a computer program written in FORTRAN for use in simulating the kinematics and dynamics of motion of three vehicles in three-dimensional space. It was developed primarily as a research tool for use in detailed explorations of the mechanics, geometry, and vehicle performance characteristics of an interceptor-target engagement. The output is a step-by-step time history of variables relating to each of three vehicles' position, velocity, acceleration, applied forces, attitude or orientation, and aerodynamics. While the program is highly versatile, its most important capabilities relate to interceptor-target guidance and intercept trajectories in general. Since the flights of three vehicles may be represented simultaneously, the program can be used to simulate aerial combat between aircraft (e.g., a two-on-one engagement, or a one-on-one with missile launch). There are also a number of other possibilities, such as (1) using one or more of the vehicles to represent an air-to-surface missile (ASM), surface-to-air missile (SAM), or surface-to-surface missile (SSM), (2) using the target to represent an ICBM reentry vehicle (RV), (3) using vehicles 1 and 2 to represent first- and second-stage boosters, or (4) using one vehicle to represent an orbiting satellite.

So far, the model has been used primarily in simulating fighter-versus-fighter and missile-versus-fighter duels. Because of the number of maneuver routines created for this specialized purpose, the model's present development in this area is farther advanced than in other areas. However, ASM and SAM simulations are currently being performed in connection with other Rand projects, and the program has been used successfully in satellite-intercept problems. The model's usefulness and potential capabilities are expected to grow with each new application.

In the organization of TACTICS, primary emphasis was placed on versatility in order to accommodate a broad spectrum of problems. At the same time, it was thought that the program should be easy to use and easily adaptable to refinements and new features. Accordingly, it was built in modular building-block form, with many subroutines that
could be replaced, modified, or simply disregarded at the option of the user. In fact, new building blocks are welcomed, since they increase the program's potential problem-solving capability. The construction of TACTICS is in many respects comparable to that of the ROCKET\(^{(1)}\) program, although the two have different purposes. Many ideas were borrowed from ROCKET in the formulation of TACTICS, especially its most important distinguishing feature, that of allowing the researcher to obtain results without a detailed knowledge of the inner workings of the program.

The program is able to simulate the flight or trajectory of almost any type of vehicle in almost any mixture of vehicles. Various guidance-law subroutines are available for simulating terminal-homing or command-guided trajectories (e.g., biased proportional and proportional navigation, lead collision, and pursuit and lead pursuit). A variety of open-ended control-law subroutines are also available for simulating aero- dynamic maneuvers such as turning, diving, climbing, and combinations thereof. The library of guidance- and control-law subroutines has been growing and will probably continue to grow as new problems are encountered. In fact, experience has shown that it is convenient to incorporate into one of these control-law subroutines the unique features of a particular vehicle or problem (e.g., Sidewinder or a hypothetical aircraft or missile design).

Options are available for considering the earth either flat or round and either fixed or rotating, but the gravitational field is a simple inverse square law field. Each of the three vehicles has three wind axes associated with its attitude or orientation reference system. The vehicles may be either fixed or in motion with respect to the earth but not below its surface. The miss distance or point of closest approach between two of three vehicles may be calculated and the problem run automatically terminated if so desired.

As mentioned earlier, the step-by-step time history of the engagement is limited to the consideration of the performance of three vehicles at one time. However, two devices have been incorporated for extending this capability to more than three vehicles by means of sequential computations. These devices, designated "recall" and "restore," are
illustrated in Fig. 1. Consider a vehicle 1 which launches a missile 2 at a target 3. At some subsequent time or event (e.g., a miss) it may be desirable to recall vehicle 2 and place it in captive flight on either 1 or 3 for a subsequent quasi-vehicle 4. (It is not necessarily that 4 have the same characteristics as, or even resemble, 2.) To illustrate the restore feature, consider the same example except that at some time or event subsequent to the launching of vehicle 2 (e.g., a hit, miss, or ground impact) we wish to restore the launch-time situation. After restoration has occurred, events may proceed, and a new launch (4, 5, etc.) may take place at a subsequent time or event. Or perhaps branching is desired, i.e., some characteristic or parameter is altered and 4, 5, etc. are to be launched under the same restored conditions of time and geometry.

Consider the factors vital to any program for simulating flight trajectories. As illustrated in Fig. 2, it is necessary to (1) read in data as initial conditions, (2) calculate the geometry, (3) specify the applied forces involved, (4) integrate the equations of motion, and (5) output the answers. This is the basic framework of TACTICS, to which embellishments are added (e.g., model atmosphere, coordinate transformations, and aerodynamic computations). Although several options are provided for modes of integration and output form, the framework may be considered inflexible except for specifying the applied forces. In this case, maximum flexibility is provided by an arrangement comparable to plug-in modules; that is, the applied forces and the time(s), event(s), or situation(s) dictating their application are contained within two subroutines which may be specialized to deal with a particular problem. To illustrate, each particular problem is defined by initial-condition input data and by a number of POLICY statements, which are logical expressions dictating the control laws governing the flight of each vehicle. They are usually conditionally based on time or on geometric and kinematic relationships. (The POLICY subroutine is discussed in more detail in Section II.)

A rudimentary description of the operational principles necessary to simulate an intercept problem will now be useful. As a starting point, imagine a vehicle in three-dimensional space having a position
Fig. 1 — Recall and restore features contrasted
Fig. 2 — Basic framework for flight simulation (generalized)
and a velocity defined by initial-condition input data. If no accelerations or applied forces are involved, the time history or trajectory of the vehicle will obviously be a straight line. However, in the general trajectory case, there will be a net acceleration \((\dot{x}, \dot{y}, \dot{z})\) due to gravitational, aerodynamic, and thrust forces, where the correspondence between the resultant force and \(\dot{x}, \dot{y}, \dot{z}\) is given by \(F = ma\). If it is assumed that all forces can be defined and specified, the trajectory simulation problem is reduced to (1) integrating the net acceleration as a function of time to obtain velocity and (2) integrating velocity to obtain position. If the applied net force were constant or a simple function of time, the trajectory simulation would be straightforward and relatively simple. In the general case, particularly when closed-loop guidance and aerodynamics are involved, the forces may be complicated functions of position, velocity, geometry, and time (and may involve the behavior or predicted behavior of some other vehicle(s)); hence numerical integration techniques must be used to calculate the trajectory stepwise using time increment \(\Delta t\). It should be clear from the preceding discussion that to start a problem run, input data for initial position and velocity must be supplied. The bulk of all other input data will pertain to parameters associated with the definition or calculation of the forces to be applied (e.g., lift, drag, thrust, and gravity). Representing the motion of three different vehicles multiplies the input requirements, of course, but not necessarily by a factor of three. The definition of the intercept problem, in terms of logic, timing, and geometric events, or of any one of these, dictates what types of control laws are to be applied under which circumstances. This definition is contained in a POLICY subroutine that in a sense is also an input to the program. The forces or force functions which are to be called upon by POLICY are contained in the library of control-law subroutines. It is not expected that this library will ever include every conceivable force function or guidance law, especially since TACTICS is a research tool for experimentation. However, experience has shown that new subroutines may usually be conveniently generated by modifying those already on hand.
II. POLICY SUBROUTINE

The POLICY portion of the program, which is supplied by the user, consists of logical statements and expressions dictating the guidance- or control-law subroutines that govern the flight of each vehicle. These subroutines define the various forces that are to be applied to a vehicle to perform a certain maneuver or to guide in accordance with some prescribed guidance law. The result of all forces will be a net acceleration. One need not be familiar with the mathematics involved in guidance, aerodynamics, propulsion, etc. to write a POLICY routine. However, a large number of options are available, and the user should be familiar with them and with the library of available guidance- or control-law routines. To illustrate the use of a POLICY subroutine, consider the problem of simulating the flight of an aircraft from take-off to landing. Clearly, a number of control laws defining the forces and hence the net acceleration would be needed for even the most elementary flight plan. This subject is discussed in further detail in Sections IV and IX. Consider the net accelerations associated with each of the following:

- Climb (vertical plane).
- Straight flight, i.e., no turning component of acceleration.
- Turn (horizontal plane).
- Dive (vertical plane).

Moreover, assume that subroutines are available for describing the above trajectories (CLIMB, STRFLT, RTURN, LTURN, and DIVE, respectively). A POLICY routine is required to define the conditions dictating the transition from one maneuver to another. For example, CLIMB may be called for a given time interval or until some specified altitude or other criterion is reached; then STRFLT is called, and so on.

With respect to the preceding illustration, the most obvious example of choice of option is the selection of the CLIMB, STRFLT, etc. routines. In the argument listings for these routines there are also options pertaining to the following conditions:
-10-

- The vehicle the law is to govern (1, 2, or 3).
- The magnitude (and perhaps direction) of the propulsive or thrust force.
- The number of g's commanded for the climb, dive, or turn maneuvers.
- The use of tabulated values or analytic functions in calculating aerodynamic forces.

Appendix D contains a list of optional subroutines currently available and instructions for their use. Section IX gives examples of problem runs with corresponding POLICY subroutines.

So far this subject has been discussed in the context of choices to be made within the POLICY routine. Numerous other options may be selected by reading in flags or constants as part of initial-condition data. However, it is sometimes desirable to override these initial instructions in POLICY if during a problem run a situation arises that requires, perhaps, a change in printout frequency or integration step size.
III. INPUT-OUTPUT GEOMETRY

In order to construct a problem run, it is necessary to define the problem by reading in initial-condition data and establishing a flight-control program (POLICY subroutine). Before a detailed description of this process is possible, it is essential to have a basic acquaintance with certain fundamentals of the coordinate system and intercept geometry.

ABSOLUTE AND RELATIVE POSITION GEOMETRY

Figure 3 shows two position vectors, \( \vec{r}_1 \) and \( \vec{r}_2 \), in a space referred to a three-dimensional \( x, y, z \) coordinate system of arbitrary origin. The position of point 1 may be defined either in Cartesian coordinates, \( x_1, y_1, z_1 \), or in spherical coordinates, \( r_1, \theta_1, \phi_1 \). Because there are certain advantages to each form of coordinate system, TACTICS can convert one to the other, so that all vectors are expressed in six-element form, as in the following:

\[
\vec{r}_1 = r_1 (x, y, z, \theta, \phi)
\]

Initial-condition position data may be read in either Cartesian or spherical coordinates with respect to the reference frame. Generally, the former is the most convenient for this purpose. For example, it is easy to visualize that one vehicle is initially at an altitude of \( z_1 \) ft and arbitrarily placed at \( x_1 = 0, y_1 = 0 \), while perhaps another vehicle is initially at an altitude of \( z_2 \) ft and displaced horizontally from the first by \( x_2, y_2 \) ft.

Once initial conditions have been established and the vehicles located in a reference coordinate frame, it is generally true that the absolute displacements of the vehicles with respect to this frame are no longer of primary interest. Intercept problems are mainly functions of relative geometry, i.e., relative position and velocity or components thereof.

In Fig. 3, the vector \( \vec{r}_{12} \) represents the range and direction of point 2 relative to point 1. For three vehicles, there is the vector \( \vec{r}_{3} \), not shown. More generally, then, the relative position vectors are
Fig. 3 — Absolute and relative position geometry
\[ \mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i \]
\[ \mathbf{r}_{ik} = \mathbf{r}_k - \mathbf{r}_i \]
\[ \mathbf{r}_{jk} = \mathbf{r}_k - \mathbf{r}_j \]

(1)

These quantities, as well as all other quantities pertaining to relative geometry (usually the entire intercept problem), are calculated and used extensively within the program and form an important portion of the printout. As an example, guided-missile control laws and criteria for missile launch are usually dependent on relative-position and angular-rate data. Because relative-position information is visualized and interpreted more conveniently in spherical coordinates, it is printed out in this form.

In summary:

- Initial-condition position data may be specified in either Cartesian \((x, y, z)\) or spherical \((r, \theta, \varphi)\) coordinates with respect to the reference frame. Units of measurement are feet and degrees.
- All relative geometry (e.g., \(r_{12}\)) is computed within the program and is printed in spherical coordinates.

**Absolute and Relative Velocity Geometry**

Input velocity data and position data are treated almost identically to one another. Figure 4(a) shows a velocity vector \(\mathbf{V}_1\) associated with the position vector \(\mathbf{r}_1\). For initial conditions, \(\mathbf{V}_1\) may be entered in either Cartesian \((V_x, V_y, V_z)\) or spherical \((V, \theta, \gamma)\) form (see Fig. 4(b)). Conversion from one form to the other takes place within the program. The unit of measurement is feet per second.

As in the preceding discussion of position data, the relative velocity vectors are calculated within the program and occupy significant portions of the printout. Similarly, the relative velocities are given by
(a) Absolute position and velocity

(b) Absolute velocity

Fig. 4 — Absolute geometry
\[ \frac{\dot{r}_{ij}}{r_{ij}} = \frac{\dot{r}_j}{r_j} - \frac{\dot{r}_i}{r_i} \quad i = 1, 2, 3 \]
\[ j = 1, 2, 3 \]
\[ i \neq j \] (2)

and so on for \( \frac{\dot{r}_{jk}}{r_{jk}} \), \( \frac{\dot{r}_{ik}}{r_{ik}} \), etc. As with position vectors, coordinate transformations are performed so that these vectors are also expressed six-dimensionally in terms of \( \dot{x}, \dot{y}, \dot{z}, \dot{V}, \dot{\theta}_V, \) and \( \dot{\gamma} \). It is frequently necessary in the computations to call upon both absolute (\( \overline{r}_i \) and \( \dot{\overline{r}}_i \)) and relative (\( \overline{r}_{ij} \) and \( \dot{\overline{r}}_{ij} \)) quantities. The angular rates of rotation of the range vectors, both absolute and relative, are of basic importance to most guidance or trajectory problems. Expressed in vector notation,

\[ \dot{\overline{r}}_i = \overline{r}_i \overline{I}_{r i} + \overline{\omega}_i \times \overline{r}_i \] (3)

where \( \dot{\overline{r}}_i \) = scalar time rate of change of the vector \( \overline{r}_i \)

\( \overline{I}_{r i} \) = a unit vector along \( r_i \left( \overline{I}_{r i} = \overline{r}_i / |r_i| \right) \)

\( \overline{\omega}_i \) = angular-rate vector orthogonal to both \( \overline{r}_i \) and \( \dot{\overline{r}}_i \)

Similarly, for the relative velocity vectors

\[ \dot{\overline{r}}_{ij} = \overline{r}_{ij} \overline{I}_{rij} + \overline{\omega}_{ij} \times \overline{r}_{ij} \] (4)

It is important to note that for approaching vehicles the relative range rate \( \dot{r}_{ij} \) is negative; for vehicles which are separating, it is positive. Also, the point of closest approach or minimum miss distance between two vehicles occurs when the absolute value of this relative range rate is zero.

Changing to spherical notation for convenience, let us consider the vector velocities \( \overline{r}_i \) and \( \overline{r}_j \) or \( \overline{V}_i \) and \( \overline{V}_j \), respectively. The following relationship for the angular-rate vector representing the rotation of the vector \( \overline{r}_{ij} \) is applicable:

\[ \overline{\omega}_{ij} = \frac{\overline{r}_{ij} \left( \overline{V}_j - \overline{V}_i \right)}{r_{ij}^2} \] (5)
Note that with these basic relationships it is possible to resolve vector velocities into two useful components, one along \( \overline{r}_{ij} \) and the other transverse to \( \overline{r}_{ij} \). Also, the angular rotation of the line of sight (LOS) between \( \overline{r}_i \) and \( \overline{r}_j \) is known both in direction and magnitude. This latter quantity is of prime importance in most terminal (or command) missile or aircraft guidance applications. In actual practice, it is usually a quantity determined by rate gyro measurements (in missiles) or by processing \( \theta \), \( \varphi \) angle data measured by a ground radar.

So far, relationships have been given for resolving the velocity vectors into components parallel and perpendicular to the \( \overline{r}_i \) and \( \overline{r}_{ij} \) values and for calculating the angular-rate vectors. In subsequent discussions of guidance laws and acceleration, two useful vectors associated with velocity will become important. Imagine a plane normal to a velocity vector \( \overline{v} \) having spherical coordinates \( V, \theta_V, \gamma \). It is convenient and useful to define a unit vector \( \overline{l}_A \) common to this plane and the horizontal \( x-y \) plane, and a unit vector \( \overline{l}_D \) common to this plane and the vertical plane. Thus, \( \overline{r} \) or \( \overline{V} \) and the unit vector \( \overline{l}_V \) will form a right-hand orthogonal system with \( \overline{l}_A \) and \( \overline{l}_D \), as shown in Fig. 5. The following relationships are applicable for determining the components:

\[
\begin{align*}
1_Ax &= - \sin \theta_V & 1 Dx &= - \sin \gamma \cos \theta_V & 1 Vx &= \cos \gamma \cos \theta_V \\
1_Ay &= \cos \theta_V & 1 Dy &= - \sin \gamma \sin \theta_V & 1 Vy &= \cos \gamma \sin \theta_V \\
1_Az &= 0 & 1 Dz &= \cos \gamma & 1 Vz &= \sin \gamma
\end{align*}
\]  

ELEVATION, AZIMUTH, AND BEARING-ANGLE GEOMETRY

An important consideration in most intercept problems is the direction or orientation of the LOS between one vehicle and another. A simple example of the calculation is a description of relative target position by a pilot in a cockpit in terms of numbers on a clock face (12 o'clock, straight ahead; 3 o'clock, directly to the right; etc.) with elevation designated as high or low. Obviously, this
Fig. 5—Unit vectors normal to velocity vector
description would require alteration if the aircraft were suddenly to make a substantial change in attitude or orientation by pitching, rolling, or yawing. Similarly, TACTICS calculates and prints out azimuth (clock-face position) and elevation (high-low) orientation of the LOS from each vehicle relative to the other two. It also prints out a bearing angle defined here as the angle between the LOS and the longitudinal axis of the vehicle. Azimuth angle is measured in the plane formed by the right wing and the longitudinal axis as a pilot would view the relative geometry. Similarly, elevation angle is measured in the plane formed by the longitudinal axis and the line through the top of the cockpit. Azimuth angle ranges from 0 to ±180 deg, with positive to the right. Elevation angle ranges from 0 to ±90 deg, with positive upward, as shown in Fig. 6. Note that the bearing angle is defined as the total angle between the LOS and longitudinal axis and hence is independent of roll angle (see Appendix A for definition); this is a very useful feature, since it is confusing to visualize azimuth and elevation angles when a vehicle is maneuvering.

For those problems which involve sensors—e.g., infrared, radar, or optical—it is likely that constraints will be imposed on maximum values of elevation and azimuth look angles due to mechanical, electrical, or optical limits. Provisions have accordingly been made to input any such constraints as initial-condition data.

Mathematical details and derivations pertaining to aircraft attitude or orientation angles and elevation, azimuth, and bearing-angle geometry are contained in Appendix A.

In summary:

1. Associated with each vehicle are two LOS oriented toward the other two vehicles.
2. The vehicle's coordinate system may be regarded as a longitudinal axis through the airframe, a line through the right wing, and a line through the top of the cockpit.
3. The orientation of the LOS with respect to this coordinate system is resolved in terms of azimuth, elevation, and bearing angles.
Fig. 6 — Elevation, azimuth, and bearing-angle geometry
IV. DEFINING THE FORCES AND ACCELERATIONS

CONTROL LAWS

TACTICS integrates the equations of motion defined by the three components of net acceleration associated with each of the vehicles. A resultant vector force $\mathbf{F}$ will define an acceleration $\mathbf{a}$. The forces of primary interest which will sum to this resultant force $\mathbf{F}$ may be categorized as (1) gravitational, (2) aerodynamic lift and drag, and (3) propulsive. If we assume that these forces are defined in magnitude and direction, it is a straightforward procedure to add them, resolve them into components, and determine the net acceleration. Each control-law subroutine may be considered as a modular unit where this process or its equivalent is performed, the output being the three components of net acceleration applicable to a particular vehicle, as shown schematically in the block diagram below.

<table>
<thead>
<tr>
<th>Aerodynamic force</th>
<th>Control Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravitational force</td>
<td>Vehicle (i)</td>
</tr>
<tr>
<td>Propulsive force</td>
<td>$\bar{a}_i$</td>
</tr>
</tbody>
</table>

Mass

The important point to note in this diagram is the correspondence of the net sum of forces to the net acceleration. Indeed, there are many control laws in which the net acceleration is by definition the starting point, and all control-law computations are primarily concerned with finding the correspondence between the forces that would be necessary to create such an acceleration.

To illustrate, consider a hypothetical control law in which all components of net acceleration are by definition zero. Assume that propulsive and gravitational forces are determined. The function of
the control law in this simple case is to determine the aerodynamic lift and drag forces necessary to guarantee the postulated output condition, i.e., zero acceleration.

LATERAL ACCELERATION AND ITS COMPONENTS

Many reference texts differ on the definition and usage of the term "lateral acceleration." Throughout this Memorandum, it is defined as a vector quantity in a plane normal to the velocity vector or flight path of the vehicle. Since the velocity vector $\vec{V}$ may be oriented in any direction in the general case, the above definition does not constrain the lateral acceleration vector to any particular direction or to any plane other than the one normal to $\vec{V}$.

Referring to Fig. 5, $\vec{I}_A$ and $\vec{I}_D$ are also by definition in the same plane normal to $\vec{V}$. Accordingly, it is convenient to resolve the lateral acceleration $\vec{a}$ into two components, $a_h$ and $a_v$.

$$a_h = \vec{a} \cdot \vec{I}_A$$  \hspace{1cm} (7)

$$a_v = \vec{a} \cdot \vec{I}_D$$

By definition, the component $a_h$ now represents a turning acceleration in the horizontal plane and $a_v$ represents a climbing or diving acceleration in the vertical plane (caution: $a_v$ is not necessarily in the vertical direction). At this point, it is necessary to distinguish between the specified or commanded values of $\vec{a}$ and its components ($a_h$ and $a_v$) and the modified or output values. In simulating guidance of aircraft or missiles, real-world considerations often necessitate the modification of commanded values, because of constraints such as structural or aerodynamic limitations, time lag, and noise. This subject will be discussed further in Section V.

For the moment, let us assume that commanded values are constrained or modified so that output values result. With subscripts used to denote the difference, a commanded value of lateral acceleration is designated as $\vec{a}_C$ with components $a_{Ch}$ and $a_{Cv}$, and a modified or output
value as \( a_0 \) with components \( a_{oh} \) and \( a_{ov} \). The third component, \( \dot{V} \), is assumed to be a function only of propulsion and aerodynamic drag forces with no constraints or modification. The total resultant acceleration \( \ddot{V} \), describing the motion or trajectory of a vehicle represented as a point mass, is then

\[
\ddot{V} = a_{oh} \mathbf{I}_A + a_{ov} \mathbf{I}_D + \dot{V} \mathbf{I}_V \tag{8}
\]

where \( \mathbf{I}_V \) is the unit vector along \( \mathbf{V} \) and \( \dot{V} \) is the time rate of change in the magnitude of \( \mathbf{V} \). Expressed in inertial Cartesian \((\mathbf{x}, \mathbf{y}, \mathbf{z})\) form, which is more convenient for numerical integration of the equations of motion, the components are

\[
\begin{align*}
\dot{x} &= \dot{V} \cdot \mathbf{i} = a_{oh} \mathbf{1}_{Ax} + a_{ov} \mathbf{1}_{Dx} + \dot{V} \mathbf{1}_{Vx} \\
\dot{y} &= \dot{V} \cdot \mathbf{j} = a_{oh} \mathbf{1}_{Ay} + a_{ov} \mathbf{1}_{Dy} + \dot{V} \mathbf{1}_{Vy} \\
\dot{z} &= \dot{V} \cdot \mathbf{k} = a_{ov} \mathbf{1}_{Dz} + \dot{V} \mathbf{1}_{Vz}
\end{align*}
\tag{9}
\]

where \( \mathbf{i}, \mathbf{j}, \) and \( \mathbf{k} \) are the unit vectors along the \( x, y, z \)-coordinate frame axes, respectively, and \( \mathbf{1}_{Ax}, \mathbf{1}_{Dx} \), etc. are the components of the unit vectors given by Eq. (6).

To represent the flight of three vehicles in motion simultaneously, 18 differential equations must be integrated, 9 involving the accelerations \( \dot{V}(i) \) and 9 the velocities \( V(i) \). TACTICS expresses these differential equations in two different forms: One uses the flat-earth representation (less complex and faster) and the other uses a round rotating or nonrotating earth (essential for space applications, long ranges, or high speeds). The expressions in Eq. (9) are the basic acceleration equations which are integrated for the flat-earth representation. Further details on numerical integration methods, round-earth form of the equations, coordinate transformations, and derivations are included in Appendix B. Next, the correspondence between net lateral acceleration and the gravitational, propulsive, and aerodynamic forces which cause this acceleration will be considered. At
this point it is important to stress that a vehicle maneuver may be defined in either of two ways, depending on the particular problem. A complete force-acceleration correspondence must be established by specifying either (1) the lateral acceleration and sufficient other information about the forces, or (2) the forces and sufficient other information about the lateral acceleration.

**FORCES AND FORCE RESOLUTION**

The various forces assumed to be acting on the center of gravity (c.g.) of each vehicle are shown in Fig. 7. In vector form, the summation of these forces is

$$ \dot{mV} = mg + \overline{L} + \overline{T} + \overline{D} $$  \hspace{1cm} (10)

where $\dot{V}$ = vehicle acceleration

$m$ = mass of the vehicle ($m = W/g$)

$g$ = acceleration due to gravity (unit vector in the -z direction)

$\overline{L}$ = lift force (defined as normal to the velocity $\overline{V}$)

$\overline{T}$ = resultant thrust force

$\overline{D}$ = drag force (unit vector in the $-\overline{V}$ direction)

This is a general vector expression where the positive or negative signs are accounted for by the orientation of the unit vectors associated with each term. For notational convenience, the $i$ subscript which refers to a particular vehicle has been omitted. The above equation may be resolved and separated into two equations, one representing the lateral acceleration normal to $\overline{V}$ and the other representing the $\dot{V}$ acceleration in the direction of $\overline{V}$. Before doing this, certain important definitions or assumptions are necessary:

- The thrust force $\overline{T}$ acting through the c.g. is also coincident with the longitudinal body axis of the vehicle.
- The angle of attack $\alpha$ is taken to be the angle between $\overline{T}$ and the tangent to the flight path $\overline{V}$.
- All maneuvers consist of "coordinated turns," defined by the condition that a plane passed through the longitudinal body axis and including the yaw axis must also include the vector $\overline{V}$. 
Fig. 7 — Coordinated turn in level flight
The situation for a coordinated turn in level flight is shown in Fig. 7. The vectors \( \mathbf{L} \), \( \mathbf{T} \), and \( \mathbf{V} \) are co-planar and \( \mathbf{T} \) is coincident with the longitudinal axis.

For the particular case of a level-flight turn, the bank angle \( \psi_b \) is shown to be the angle between \( \mathbf{L} \) and the vertical, but in the general case it is the angle between \( \mathbf{L} \) and the reference vector \( \mathbf{I}_D \) shown in Fig. 5. (Note that this is a wind axis or velocity reference, since \( \mathbf{I}_D \) is normal to \( \mathbf{V} \).) With these definitions and assumptions in mind, Eq. (10) may now be resolved into two scalar equations by summing forces and accelerations along \( \mathbf{V} \) (direction \( \mathbf{I}_V \)) and in a plane normal to \( \mathbf{V} \), using the symbol \( \mathbf{I}_l \) to denote the direction of this resulting normal or lateral acceleration (see page 22 for definition of lateral acceleration).

\[
\dot{V} = \frac{g}{W} (T \cos \alpha - D - W \sin \gamma) \quad (11)
\]

\[
a_o = \left[ \frac{g}{W} (L + T \sin \alpha) + \mathbf{g} \right] \cdot \mathbf{I}_l \quad (12)
\]

where \( a_o \) is the absolute magnitude of net lateral acceleration \( a_{o_l} \) and \( \alpha \) is the angle of attack, as defined above.

Two examples will illustrate the basic relationships in order to avoid possible misunderstanding of definitions or terminology.

**Case 1: Determination of Forces Required For a Specified Lateral Acceleration**

Assume a simulation of a climbing turn arbitrarily defined by a lateral acceleration component \( a_{o_{lh}} \) in the horizontal plane and a positive \( a_{oV} \) component in the vertical plane. If the maneuver is initiated from level flight (i.e., \( \mathbf{V} \) in the horizontal plane), the \( a_{oV} \) component and \( \mathbf{I}_D \) will initially be oriented toward the vertical direction but not thereafter. Figure 8 shows the vectors and their scalar magnitudes. The total net lateral acceleration will be

\[
a_o = \sqrt{a_{o_{lh}}^2 + a_{oV}^2} \quad (13)
\]
\frac{q}{W}(L + T \sin \alpha) = \sqrt{a_{oh}^2 + (a_{ov} + g \cos \gamma)^2}

Fig. 8—Bank angle and lateral acceleration components
The normal force \( F_n \), defined as the sum of aerodynamic lift and propulsive forces normal to \( \vec{V} \), is

\[
F_n = L + T \sin \alpha = \frac{W}{g} \sqrt{(g \cos \gamma + a_{ov})^2 + a_{oh}^2}
\] (14)

The direction of the vector \( \vec{L} \) is given by

\[
\vec{L} = \frac{W}{F_n}\left[ a_{oh} \vec{A} + (a_{ov} + g \cos \gamma) \vec{D} \right]
\] (15)

Note that the gravitational force component normal to \( \vec{V} \) is associated with the \( g \cos \gamma \) term.

Assuming a small, the initial (\( \gamma = 0 \)) lift force \( L \) required for the maneuver is

\[
L = \frac{W}{g} \sqrt{a_{oh}^2 + (a_{ov} + g)^2}
\] (16)

The direction of \( \vec{a}_o \) is

\[
\vec{a}_o = \frac{(a_{oh} \vec{A} + a_{ov} \vec{D})}{a_o}
\] (17)

and the bank angle is

\[
\psi_B = -\sin^{-1}\left(\frac{a_{oh}}{L + T \sin \alpha} \cdot \frac{W}{g}\right)
\] (18)

The minus sign is arbitrarily assigned to make \( \psi_B \) negative for a left turn, i.e., \( a_{oh} \) is positive for \( \theta_v \) increasing (shown in Fig. 4). Accordingly, in terms of bank angle the components of \( \vec{a}_o \) become

\[
a_{oh} = -\frac{g}{W} (L + T \sin \alpha) \sin \psi_B
\] (19)

\[
a_{ov} = \frac{g}{W} (L + T \sin \alpha) \cos \psi_B - g \cos \gamma
\] (20)

The force \( \vec{F}_n \) is defined as the force acting in the direction of the lift vector \( \vec{L} \). Expressed in g's, it is
\[
\frac{F_n}{W} = \frac{L + T \sin \alpha}{W}
\]  
(21)

From Fig. 5,

\[
\cos \psi_B = \frac{a_{ov} + g \cos \gamma}{F_n} \left(\frac{w}{g}\right)
\]

Hence,

\[
\frac{F_n}{W} = \frac{a_{ov} + g \cos \gamma}{g \cos \psi_B}
\]  
(22)

For a horizontal turn in level flight, as shown in Fig. 7, this reduces to the simple expression

\[
\frac{F_n}{W} = \frac{1}{\cos \psi_B}
\]  
(23)

Case 2: Determination of Acceleration Components Resulting From Specified Forces

Assume that aerodynamic and propulsive forces may be determined utilizing some prescribed control law and that it is necessary to determine the acceleration vector components \(a_{oh}\), \(a_{ov}\), and \(V\) from Eq. (10), which is repeated below:

\[
\dot{V} = \frac{\dot{a}}{g} + \frac{g}{W} (L + T + D)
\]  
(24)

Ignoring for the moment the effects of time lag, aerodynamics, or structural constraints and referring to Eqs. (6), the acceleration components are

\[
\begin{align*}
\dot{V} &= \frac{\dot{V}}{V} \cdot \dot{V} \\
\dot{a}_{oh} &= \frac{\dot{V}}{V} \cdot \dot{A} \\
a_{ov} &= \frac{\dot{V}}{V} \cdot \dot{D}
\end{align*}
\]  
(25)

The orientation of the lateral acceleration vector \(\vec{a}_o\) is given by Eq. (16).
AERODYNAMIC FORCES

Aerodynamic computations may involve a large number of input parameters corresponding to the flight characteristics of a particular airframe. On the other hand, there are trajectory simulation problems where aerodynamic effects are not pertinent and no aerodynamic computations are necessary for a particular vehicle. An example is a point-mass target moving along a path where the motion is arbitrarily postulated, say, a straight-line path at constant speed.

In the general case, expressions for force magnitudes (lift $L$ and drag $D$) are

$$ L = C_L A q $$
$$ D = C_D A q $$
$$ q = \frac{1}{2} \rho V^2 $$

where $C_L$ = lift coefficient
$C_D$ = drag coefficient
$A$ = aerodynamic reference area of the vehicle (ft$^2$)
$q$ = dynamic pressure (lb/ft$^2$)
$\rho$ = air density (slug/ft$^3$) (see Appendix I for a definition of model atmosphere)

In simulating the flights of specific aircraft designs, provisions have been made for incorporating tables to describe the interactions among lift coefficient $C_L$, drag coefficient $C_D$, angle of attack $\alpha$, maximum lift coefficient $C_{L_{\text{max}}}$, and Mach number. Interpolation routines are employed to provide the equivalents of functional relationships. Alternatively, provisions have been made for using the following analytic expressions:

$$ C_L = \frac{dC_L}{d\alpha} (\alpha - \alpha_o) $$
$$ C_D = C_{D_0} + \frac{dC_D}{d(C_L^2)} C_L^2 $$

(27)
where \( \frac{dC_L}{d\alpha} = \) slope of the \( C_L = F(\alpha) \) curve
\( \alpha_0 = \) zero-lift angle of attack (deg)
\( C_{D0} = \) zero-lift drag coefficient
\( \frac{dC_D}{d(C_L^2)} = \) a coefficient used with a parabolic function for drag coefficient, i.e., \( C_D = F(C_L^2) \) (may be treated as a constant or a function of Mach number)

Details on how to provide and use tabulated values and on how to select the various options available are provided in Sections IX through XIII and Appendix D of this Memorandum.

**PROPULSIVE FORCES**

As previously mentioned, propulsive forces are assumed to act on the c.g. of the vehicle and to be coincident with the vehicle’s longitudinal axis. As in aerodynamic computations, tables may be incorporated to describe the interrelationships between military and afterburner thrust, fuel flow, and Mach number. There are also provisions for reading in constant values for thrust as input data. A varying thrust condition may be simulated by the simple expedient of multiplying each thrust value by a "throttle" parameter (normally set to 1.0 unless otherwise specified in POLICY). Additional flexibility is also available by incorporating thrust values or functional relationships in maneuver subroutines specialized to the particular problem or to vehicle characteristics.

**GRAVITATIONAL FORCES**

**TACTICS** is automatically set for the simplest assumption of a gravitational force—a flat earth causing 1 g or 32.174 ft/sec\(^2\) acting downward in the negative z-direction. However, options are provided for considering the more complicated cases of a round earth, rotating or nonrotating, so that centrifugal, Coriolis, and inverse square law effects may be included if pertinent to the problem, e.g., in space applications and hypersonic flight. The basic expressions for the flat-earth representation are given by Eq. (9), and the gravitational
force term appears in a rather straightforward way (which is to be expected, since \( \vec{g} \) is treated as a constant vector, i.e., in magnitude and direction) in Eqs. (11) through (16). All expressions and derivations relating to the round-earth representation are discussed in Appendix B because of the large number of details involving coordinate transformations and reference vectors. The gravitational acceleration in this case is taken to be rotating and varying in magnitude, as

\[
\vec{\ddot{g}} = \frac{\vec{F}_g}{m} = -\frac{\mu \vec{R}}{R^3}
\]

(28)

where \( \vec{F}_g \) = gravitational force

\[
u = 1.407645 \times 10^{16} \text{ ft}^3/\text{sec}^2
\]

\( \vec{R} \) = radius vector from the geocenter to the vehicle

\( R = |\vec{R}| \)
V. AERODYNAMIC, STRUCTURAL, AND TIME-LAG CONSTRAINTS

AERODYNAMIC AND STRUCTURAL CONSTRAINTS

As mentioned in Section IV, in simulating vehicle flight performance it is usually necessary to impose constraints or limits on the forces applied to the vehicle. Two primary limitations are (1) aero-
dynamic, i.e., a limitation on lift coefficient $C_L$ to some specified
value representing a boundary on flight stability, and (2) structural,
i.e., a load limit imposed by possible damage to the airframe or com-
ponents (or possibly to a human being). The aerodynamic constraint
imposes the following condition on the magnitude of the normal force
$F_n$ (see Eq. (14)):

$$F_n \leq C_{L_{\text{max}}} A_q + T \sin \alpha_{\text{max}}$$

(29)

where $C_{L_{\text{max}}}$ is a maximum value for $C_L$ and $\alpha_{\text{max}}$ is a maximum value for
$\alpha$. Since $C_L$ and $\alpha$ are functionally related, either one may be specified
and the other calculated. Similarly, the structural constraint is

$$F_n \leq \left(\frac{W}{g}\right) a_{\text{Smax}}$$

(30)

where $a_{\text{Smax}}$ is the structural (or human) acceleration limit. It should
be noted that by definition the force $F_n$ is normal to the velocity vec-
tor $\vec{V}$. To be precise, the structural limit $a_{\text{Smax}}$ should be considered
as normal to the vehicle's longitudinal axis, which is separated from
$\vec{V}$ by the angle $\alpha$. In the TACTICS program and in the derivations which
follow, this difference is ignored and $a_{\text{Smax}}$ is assumed to be a lateral
acceleration limit, by previous definition also normal to $\vec{V}$. Designat-
ing the specified or computed value of the normal force as $F_{nC}$, the
aerodynamic limit as $F_{na}$, and the structural limit as $F_{ns}$, TACTICS
takes the applied normal force $F_n$ to be

$$F_n = \min(F_{nC}, F_{na}, F_{ns})$$

(31)
where \( \min \) is the minimum magnitude of the three values. When a constraint value (\( F_{na} \) or \( F_{ns} \)) is taken, the initial assumption is that the specified or computed direction of the commanded acceleration, i.e., the unit vector \( \overline{I}_1 \), remains constant but that the magnitudes of the forces and corresponding accelerations should be compatible with the constraining value. If \( a_{\max} \) is designated as

\[
\begin{align*}
  a_{\max} &= \left( \frac{g}{W} \right) \min (F_{na}, F_{ns}) \\
  \text{(32)}
\end{align*}
\]

the problem is to determine the corresponding magnitude of the lateral acceleration magnitude \( a_0 \). The applicable expression is

\[
\begin{align*}
  a_{\max} &= \left| a_0 \overline{I}_1 + g \cos \gamma \overline{I}_D \right| \\
  \text{(33)}
\end{align*}
\]

from which the following equation is derived:

\[
\begin{align*}
  a_0^2 + 2g \cos \gamma (\overline{I}_1 \cdot \overline{I}_D) a_0 + (g \cos \gamma)^2 - a_{\max}^2 &= 0 \\
  \text{(34)}
\end{align*}
\]

If the dot product is denoted as

\[
\begin{align*}
  \cos \delta &= (\overline{I}_1 \cdot \overline{I}_D) \\
  \text{(35)}
\end{align*}
\]

the solution for \( a_0 \) is

\[
\begin{align*}
  a_0 &= -g \cos \gamma \cos \delta \pm \sqrt{a_{\max}^2 - g^2 \cos^2 \gamma \sin^2 \delta} \\
  \text{(36)}
\end{align*}
\]

To solve the quadratic equation, the following conditions must be satisfied:

- The value of \( a_0 \) must by definition be positive, since it represents the magnitude of a vector. (For multiple positive solutions the largest magnitude is taken.)
- The discriminant should not be negative.

Difficulties in maintaining the direction of \( a_0 \) (i.e., \( \overline{I}_1 \)) arise when \( a_{\max} \) becomes small (e.g., a wingless vehicle with small \( C_{L\max} \)). These special conditions are handled as follows:
If a positive solution for $a_o$ is not possible, then the sign of $\cos \delta$ must be made negative, corresponding to a diving condition—i.e., the direction of $a_{oV}$ must be reversed.

If the discriminant becomes negative, then the value of $\sin \delta$ must be adjusted so that

$$\sin \delta = \frac{a_{\text{max}}}{g \cos \gamma}$$  \hspace{1cm} (37)

The net effect of either of the above alterations is to change the direction of the $\overline{I}_1$ vector. The resultant direction is given by

$$\overline{I}_1 = \sin \delta \overline{I}_A + \cos \delta \overline{I}_D$$  \hspace{1cm} (38)

**TIME LAG**

In the diagram below, the block labeled "Lag" represents a transfer function to simulate time lag between an input command acceleration $\overline{a}_c$ and an output response $\overline{a}_o$.

The block labeled "Limit" represents the possible imposition of aerodynamic or structural constraints as discussed previously. Since certain arbitrary assumptions are involved in each of these processes, the purpose of the diagram is to emphasize the modular building-block form; alternate subroutines adapted to a particular problem may be substituted.

The time-lag transfer function describes the performance of the vehicle hardware mechanization in terms of control-system response, gyro prediction, tracking system, filter circuits, etc. Here the main generalization of the simulation occurs, for the question arises as to the pertinence of this aspect of vehicle performance to the problem solution. Unless a specific hardware design is being investigated,
the overall system response is usually represented by a product of simple first-order exponential functions of the form *

$$\frac{K_1}{1 + \tau s}$$

where \( s \) = the Laplace operator
\( \tau \) = a time constant
\( K_1 \) = a constant gain factor

Until a more accurate and elaborate representation is needed, we have elected to use an input-output relationship of the form

$$a'_0 = a_0 \frac{K_1}{1 + \tau_1 s} \cdot \frac{K_2}{1 + \tau_2 s} \cdot \frac{K_3}{1 + \tau_3 s} \quad (39)$$

This notation may be confusing, since \( a_0 \) in the above equation may or may not be the commanded acceleration \( a_c \) depending on whether or not it is constrained. If all time constants are zero and no constraints exist,

$$a_c = a'_0 = a_0$$

*Exponential solution of the form \( K_1 e^{-t/\tau} \).
VI. GUIDANCE AND CONTROL LAWS

Generally, the call for a specified control law is conditional upon the fulfillment of one or more geometric, kinematic, or time conditions. The term "control law" as used here means defining a commanded lateral acceleration vector $\vec{a}_C$ associated with the vector velocity $\vec{V}$. For those laws involving aircraft or missile guidance, such as lead collision or proportional navigation, the calculation of the $\vec{a}_C$ vector quantity is dependent upon the kinematic state of the system (position, velocity, and perhaps acceleration); these may be classified as closed-loop guidance laws. On the other hand, there are open-loop control laws, e.g., turn, dive, climb, etc., in which the kinematic state of the system is not implicit within the law itself. This difference should become more apparent as these laws are summarized later.

Twenty-four different control laws are described in this Memorandum, many of which require lengthy explanations and derivations. For convenience, only a few illustrative examples are given below; a detailed listing of all control laws and their derivations is given in Appendix C.

A brief review of terms and of the basic applicable acceleration equations follows:

1. The commanded lateral acceleration vector $\vec{a}_C$ is by definition in a plane normal to $\vec{V}$ (see page 21) and is subject to possible aerodynamic, structural, and/or time-lag constraints resulting in the formulation of the vector $\vec{a}_o$ (see Section V).

2. The vector $\vec{a}_o$ is resolved into two components by the unit vectors $\vec{T}_A$ and $\vec{T}_D$ so that

$$\vec{a}_{oh} = \vec{a}_o \cdot \vec{T}_A$$

$$\vec{a}_{oV} = \vec{a}_o \cdot \vec{T}_D$$
3. If all forces—gravitational, propulsive, and aerodynamic—are specified, the total net acceleration \( \vec{a} \), consisting of the components \( a_{oh} \), \( a_{ov} \), and \( \vec{V} \), may be determined. On the other hand, if \( a_{ov} \) and \( a_{oh} \) (gravitational and propulsive forces) are specified, the aerodynamic forces may be determined to calculate the corresponding component along \( \vec{V} \) (see Eq. (25)).

4. The basic acceleration equations (flat-earth) are given by Eq. (9) in Cartesian (\( \vec{x}, \vec{y}, \vec{z} \)) coordinates, from which the velocities and positions are determined for each vehicle after numerical integration.

The guidance laws given in the illustrations below are almost all of the type in which the vector \( \vec{a}_C \) and propulsive forces are specified and the corresponding aerodynamic forces are to be determined.

**OPEN-LOOP CONTROL LAWS**

**Straight Flight**

The commanded lateral acceleration \( \vec{a}_C \) is zero. The vehicle will fly a straight-line path (but not necessarily "straight and level"). However, an acceleration or deceleration along this path may occur due to the thrust-drag relationship. The guidance law is

\[
|\vec{a}_C| = 0
\]

\[
\vec{I}_1 = \vec{I}_D
\]

**Captive Flight**

This routine is used as a device to zero out computations and printed values for vehicles which are in captive flight. There are three modes:

- Vehicle 2 locked to vehicle 1.
- Any vehicle (1, 2, or 3) may be locked to the zero origin (i.e., zero position, velocity, and acceleration).
- Vehicle 2 locked to vehicle 3.
Launch

This control law, which simulates the launch-boost phase of a missile flight, may be applied to any of the three vehicles. The call for "launch" is usually based on some criteria stated in POLICY (e.g., range, range rate, geometry, accelerations, time, and—most importantly—combinations thereof). When this routine is called, the boost velocity \( \Delta V \) must be specified as a constant. The commanded lateral acceleration is gravitational only:

\[
\bar{a}_C = -g \cos \gamma \bar{I}_D
\]  

(41)

Left or Right Turn

The commanded lateral acceleration vector \( \bar{a}_C \) is in the horizontal plane and has a constant value as specified when calling the routine(s). The two routines (left and right) are identical except for an algebraic sign corresponding to the direction of the turn (\( \pm \bar{I}_A \)). It was decided to specify the magnitude of the turning acceleration in terms of the resultant normal acceleration, which is expressed in g's or \( \bar{F}_n/\bar{W} \) (see Eq. (21)). Accordingly,

\[
\bar{a}_C = g \sqrt{(\bar{F}_n/\bar{W})^2 - \cos^2 \gamma} \quad (\pm \bar{I}_A)
\]  

(42)

CLOSED-LOOP CONTROL LAWS

Proportional Navigation

The commanded lateral acceleration \( \bar{a}_C \) is proportional to the space rate of rotation of the LOS between missile and target. Expressed in vector notation,

\[
\bar{a}_C = \lambda V \bar{\omega}_r \times \bar{I}_V
\]  

(43)
where $\lambda$ = the "navigation constant" (may be treated as either a constant or a variable)

$V$ = vehicle speed

$\omega_r$ = relative angular-rate vector as defined in Section III

($\omega_r = \omega_{ij}$ in Eq. (5))

$\overline{I}_V$ = unit vector along the missile velocity vector $\overline{V}$, i.e.,

$\overline{I}_V = \overline{V}/V$

The direction of the acceleration is defined by

$$\overline{I}_l = \omega_r \times \frac{\overline{I}_V}{\omega_r}$$

The commanded acceleration $\overline{a}_c$ may be resolved into horizontal and vertical components by

$$\overline{a}_{Ch} = a_c \left( \overline{I}_l \cdot \overline{I}_A \right)$$

$$\overline{a}_{CV} = a_c \left( \overline{I}_l \cdot \overline{I}_D \right)$$

Missile (X)

This routine is provided as an example of how to incorporate all significant aerodynamic, propulsive, and guidance characteristics of a hypothetical guided missile design into a single package. Numerical values which are unique to the configuration and which presumably will not be varied may be listed in the routine rather than supplied as input data for each simulation run (e.g., reference area $A$, initial weight $W_0$, burning rate $\dot{W}$, etc.). Moreover, specialized analytic functions, e.g., linear or polynomial curve fits, may be used for aerodynamic $C_L$, $\alpha$, and $C_D$ relationships as well as for propulsion characteristics. The guidance law for Missile (X) is a modified form of proportional navigation; again, for convenience, all necessary relationships are incorporated within the routine, which makes it unnecessary to call upon the proportional navigation routine. For further details see Appendix C.
VII. CONCLUSIONS

The basic framework, organization, input-output integration, flow, etc. of TACTICS are considered complete. However, in accordance with its purpose as a research tool, it is open-ended and subject to adaptation for each new problem; in this sense, it will never be complete. This adaptive process is simple and flexible because of the available options and because the main variables defining a problem can be treated externally by modular units (e.g., control-law and PÔICY subroutines). Hundreds of simulation problem runs, involving air-to-air combat, SAM, and ASM applications, have been performed with a wide variety of input-data, control-law, and PÔICY options. Experiments have also been performed for space applications using the two-body equation of motion. There is, however, no guarantee that the program will work perfectly for all cases in spite of all the check-out and operational experience. The number of possible configurations and combinations is extremely large, and it is unlikely that all will ever be tried. As with all computer problems, skepticism, intuitive reasoning, and cross-checking are necessary.
Part 2

OPERATING THE PROGRAM
VIII. INTRODUCTION

The sections which follow should be considered an operating manual for the program. The reader’s familiarity with FORTRAN IV is assumed, but detailed knowledge or programming experience is not necessary. Section IX deals with formulating a POLICY subroutine composed of logical statements which dictate the control-law subroutines governing the flight of each vehicle, as explained in Section II. Section X explains how to set up a problem run by reading in initial-condition data and selecting options for integration methods, printout, table values, etc.

After the printed output sections are explained and illustrated in Section XI, several sample problems are described in detail in Section XII. The appendices contain a list of FORTRAN instructions for calling optional subroutines (including an explanation of flags and argument variables), listings of subroutines, and an explanation of aerodynamic tables and formats. Careful study of the illustrative examples in Section XII is recommended, since they may serve as convenient guides for several types of problems.
IX. FORMULATING A POLICY SUBROUTINE

In Section II the purpose and functional operation of the POLICY subroutine were described. An illustration was given of an elementary flight plan calling for maneuvers such as climb, straight flight, and dive. Obviously, policy decisions based on some criteria are necessary to carry out such a flight plan, i.e., to dictate the transition from one maneuver to another. The FORTRAN notation for the elementary maneuver subroutines mentioned above is

- CLIMBl (I, GFORC, IAERO, ITHR)
- STRFLT (I, IAERO, ITHR)
- DIVE1 (I, GFORC, IAERO, ITHR)
- KTRN1 (I, GFORC, IAERO, ITHR)

where I = vehicle to which the law applies (1, 2, or 3)
IAERO = an integer to indicate how aerodynamic computations are to be carried out
ITHR = an integer to indicate how propulsion computations are to be carried out
GFORC = number of g's (F_n/W, as shown in Fig. 8 and given by Eq. (21)) required in the maneuver

The arguments IAERO and ITHR specify whether table values, analytic expressions, or other alternatives are to be used (a complete description is given in Appendix D).

Assume that the flight plan mentioned above is required to simulate an actual takeoff of vehicle 1, an F-104 aircraft, and that the criteria that might be used are the following:

- Time: TIME (straight flight at TIME = 0)
- Speed: V(1,4) (sufficient speed for climb)
- Flight-path angle: V(1,6) (to level off)
- Altitude: R(1,3) (prior to turning)
- Heading: V(1,5) (proper course angle)

The FORTRAN notation for position and velocity, expressed in both Cartesian and spherical coordinate systems, is as follows:
\[ R(1,1) = x_1 \quad R(1,4) = \| \mathbf{r}_1 \| \]
\[ R(1,2) = y_1 \quad R(1,5) = \theta_{R1} \]
\[ R(1,3) = z_1 \quad R(1,6) = \phi_{R1} \]
\[ V(1,1) = \dot{x}_1 \quad V(1,4) = \| \mathbf{v}_1 \| \]
\[ V(1,2) = \dot{y}_1 \quad V(1,5) = \theta_{V1} \]
\[ V(1,3) = \dot{z}_1 \quad V(1,6) = \gamma_1 \]

All FORTRAN notation for position, relative position, velocity, and relative velocity is parallel to the above.

A typical initial POLICY statement for vehicle 1 at time zero might then be

\[ \text{CALL STRFLT (1,2,2)} \]

where \( I = 1 \) (refers to vehicle 1)
\( \text{IAER} = 2 \) (refers to aerodynamic tables\(^\dagger\))
\( \text{ITHR} = 2 \) (refers to military thrust tables\(^\ddagger\))

The successive statements might then be the following:

- Climb at 0.5 \( g \) to 20-deg flight-path angle
  - IF \( (V(1,4) \text{ .GT. } 120.0) \) CALL CLIMB1 (1,0.5,2,2)
  - IF \( (V(1,6) \text{ .GT. } 20.0 \text{*RAD}) \) CALL STRFLT (1,2,2)\(^\ddagger\)

- Begin leveling off at 1000 ft
  - IF \( (R(1,3) \text{ .GT. } 1000.0) \) CALL DIVE1(1,0.5,2,2)

- When within 0.1 deg of level flight, make a 0.5-g turn to a heading of 150.0 deg and resume straight flight
  - IF \( (V(1,6) \text{ .LT. } 0.1 \text{*RAD}) \) CALL RTRN1(1,0.5,2,2)
  - IF \( (V(1,5) \text{ .GE. } 150.0 \text{*RAD}) \) CALL STRFLT (1,2,2)

This elementary illustration defines a policy and flight path for one vehicle. For intercept trajectory problems, policies for two or

\(^\dagger\) F-104 tables will be loaded for vehicle 1.
\(^\ddagger\) The symbol RAD is used to convert degrees to radians.
more vehicles must be similarly defined. Omitting a reference to a vehicle \((i = 1, 2,\) or \(3\)) in \(\text{POLICY}\) results in a zero acceleration definition for that vehicle, which may be appropriate for constant- or zero-velocity (e.g., ground-target) cases. Section XII presents more typical and complex examples of \(\text{POLICY}\) subroutines with sample problems.

So far this subject has been discussed in the context of choices to be made within the \(\text{POLICY}\) routine. Numerous other options may be selected by reading in flags or constants as part of initial-condition data. However, it is sometimes desirable to override these initial instructions in \(\text{POLICY}\) if during a problem run a situation arises that requires, perhaps, a change in frequency of printout or integration step size. The following subsections list possible options in addition to those mentioned above.

**STOPPING THE PROGRAM**

The program automatically terminates after finding the closest miss distance between vehicles 2 and 1 or 2 and 3 unless otherwise specified by the flag IMISS. If no missile is launched, the program stops when running time (TIME) becomes greater than maximum specified time (TOTAL); this value is set in initial-condition data or in \(\text{POLICY}\).

If the program is to continue after finding the miss distance, flag IMISS = 1 must be set in \(\text{POLICY}\). This is used when a second missile is to be launched or if the vehicles are to continue on their trajectories (IMISS must be reset for every launch). For example, if the user wants the program to stop after finding the second missile's miss distance, IMISS must be reset to zero.

In summary:

- IMISS = 0. The program finds the closest miss distance of the missile and then stops.
- IMISS = 1. The program continues after finding the miss distance. IMISS must be reset for each launching.
- If no missile is launched, the program stops when time is greater than TOTAL (DATA 64).
INTEGRATION

Four different types of integration can be used (see Section XIII). The selection is made by setting the JINTEG flag as an initial condition (DATA 122).

If JINTEG = 0, variable-step Adams-Moulton predictor-corrector integration is used. Other values which are necessary when using this type of integration are ERTEST and HMIN. ERTEST (DATA 123) is the truncation error test for variable integration; if not specified by the user, it is automatically set at 1.0E-05. HMIN (DATA 135) is the minimum step size for the integration. The program sets HMIN = DTØ (the initial integration step set in DATA 136) unless otherwise specified.

If JINTEG = 1, fixed-step Runge-Kutta integration is used; if JINTEG = 2, fixed-step Adams-Moulton integration is used. In these cases the step size depends on the value read in for DTØ. If JINTEG = 3, the program integrates on a variable step size, controlled to allow printout exactly at specified intervals.

In summary:

- JINTEG = 0: Variable-step Adams-Moulton integration.

NUMBER OF VEHICLES USED

If the problem does not use one or more of the vehicles, its printout values can be set to zero by calling CAPFLT(I,MODE), where I indicates which vehicle is not to be used and MODE is set at 2. (See Example 3 in Section XII.)

TIMELAGS

Subroutine LAG represents a transfer function simulating the time lag between input command ACØM and output response AØUT. This function
may vary considerably in complexity, from a simple one-to-one correspondence to a highly complicated presentation of a missile guidance and control loop. The degree of realism (and hence complexity) depends, of course, on the specific problem and its significance to the final results. Accordingly, the LAG subroutine should be considered a flexible module which can be modified for a particular problem. A currently available LAG routine represents the time response of a vehicle as the product of as many as three first-order (exponential) time lags. Unless specified by the user, no time lag is used; \( A\theta T = AC\theta M \) unless structurally (ASM) or aerodynamically (CLM) constrained. In order to introduce time delay, flag TAU(I) must be set in initial-conditions data to indicate the number of time lags desired, and TAU(I,J) must be set equal to the value of the time lags. See DATA 65-78 in Appendix E for details on reading the time constants. Example 3 in Section XII is an illustration of the way a time lag is used for a SAM.

**RECALL**

This feature enables the recall of the missile once it has been launched. For details on the use of the recall option, see Fig. 24 and Example 4 in Section XII.

**RESTORE**

The restore option enables the user to restore all numerical values existing at launch time if a hit or miss has occurred. See Fig. 27 and Example 5 in Section XII for details on its use.

**ROUND EARTH, ROTATING OR NONROTATING**

The FORTRAN flags IRT, IRF, IRG, IPRINT(I), INERF, and IERET are used to select the round-earth options. In defining satellite motion, velocity components are usually given with respect to a non-rotating, inertial coordinate frame. However, in defining aircraft or missile motion it is convenient to use velocity components expressed
with respect to a local earth-fixed rotating frame of reference. Of course, if the earth is considered as nonrotating, there is no difference. These options are controlled as follows:

- IR\(\theta\)S = 0: Nonrotating earth.
- IR\(\theta\)S = 1: Rotating earth.
- INEFR = 0: Vehicle 1 (fighter) velocity expressed in relation to local, earth-fixed, rotating frame.
- INEFR = 1: Vehicle 1 velocity expressed in relation to local, earth-fixed nonrotating inertial frame.
- INERT = 0 or 1: Corresponding statements apply to vehicle 3 (target) velocity.

The location of the origin of the local earth-fixed frame is specified by initial-condition data in terms of latitude LATO and longitude LONO. This is essential to properly account for rotational effects. The x-, y-, and z-axes of this coordinate frame are taken to form a right-handed system oriented in the following way: The z-axis is coincident with the local vertical or radius of the earth at the point LATO, LONO; the y-axis is in the local horizontal plane directed eastward; and the x-axis is directed southward or northward. Initial-condition position data for each vehicle must be supplied in terms of this local frame of reference, and there are three coordinate systems which might be used:

- Cartesian (x, y, z) coordinates with respect to the origin defined by LATO, LONO (set IRF or IRT = 0 and use DATA 2-4 or 41-43 respectively).
- Spherical (r, \(\theta\), \(\phi\)) coordinates with respect to the origin defined by LATO, LONO (set IRF or IRT = 1 and use DATA 7-9 or 46-48 respectively).
- Geocentric latitude, longitude, and altitude above a reference spheroid (set IRF or IRT = 2 and use DATA 111-113 or 116-118 respectively).

Mathematical details and derivations pertaining to the coordinate transformations and equations of motion are contained in Appendix B.
X. INPUT FORM AND ORDER OF INPUT

Input data include various problem constants, parameters, control flags (for options), and initial-condition values for position and velocity. There are two sections of data: (1) the aerodynamic tables for specific aircraft or missiles and (2) the main set, which gives initial conditions, flags, and constants. An input form is shown in Fig. 9 for convenience in specifying the main set of data.* A brief explanation is given here, but it is likely that the examples given in Section XII will more clearly illustrate the use of the form. Each line refers to a data card (the examples will show that in general most of the lines and spaces may be left blank).

The first line, i.e., the first data card, specifies whether aerodynamic tables are to be used for a vehicle. It gives values to the flags JVEH(I), I = 1,2,3. If JVEH(I) = 1, tables for vehicle (I) will be used; if JVEH(I) = 0, tables for vehicle (I) will not be used. (If not, the appropriate aerodynamic and propulsion constants may be entered in DATA 80-94.) The spacing for the JVEH card is as shown in Fig. 10. If tables are to be used, they are read immediately following the JVEH card, as shown in Fig. 11.

A title card describing the run (FORMAT 12A6) is placed between the aerodynamic tables and the main set (this card is required). The locations (i.e., data numbers) and descriptions of all input data are given in Appendix E. However, the input form has spaces only for those locations most likely to be used, with spare spaces available at the bottom of the page. (There are provisions for 200 data locations, but only 143 are currently being used.)

AERODYNAMIC AND PROPULSION TABLES

The user is able to approximate the aerodynamics necessary to describe the flight of an aircraft by means of tables. If tables are used, they furnish such information as angle of attack and drag

*Figure 9 shows the input form reduced in size. A full-sized version is included at the end of the Memorandum for the reader's use.
### Vehicle Table Flags (0 or 1)

**Title Card**

<table>
<thead>
<tr>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
<th>65</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Main Set of Initial Conditions**

| 1  | 4  | 8  | 12 | 16 | 20 | 24 | 29 | 32 | 34 | Vehicle #1 | 47 | 48 | 51 | 54 | 57 | 60 | 62 | 71 | 74 |
| 0 | 0 | 0 | 1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**Vehicle #2**

| 1  | 4  | 8  | 12 | 16 | 20 | 24 | 29 | 32 | 34 | Vehicle #2 | 47 | 48 | 51 | 54 | 57 | 60 | 62 | 71 | 74 |
| 0 | 0 | 0 | 1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**Vehicle #3**

| 1  | 4  | 8  | 12 | 16 | 20 | 24 | 29 | 32 | 34 | Vehicle #3 | 47 | 48 | 51 | 54 | 57 | 60 | 62 | 71 | 74 |
| 0 | 0 | 0 | 1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**Print**

| 1  | 4  | 8  | 12 | 16 | 20 | 24 | 29 | 32 | 34 | Vehicle #1 | 47 | 48 | 51 | 54 | 57 | 60 | 62 | 71 | 74 |
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 | 1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0 | 0 | 0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

**Guidance (3 Vehicles)**

<table>
<thead>
<tr>
<th>Number of Time Constants</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Aerodynamics and Propulsion (12 Vehicles)**

<table>
<thead>
<tr>
<th>$C_{L_{max}}$</th>
<th>$C_{D_0}$</th>
<th>$C_{D_{0a}}$</th>
<th>$C_{D_{1}}$</th>
<th>$C_{D_{1a}}$</th>
<th>$C_{D_{2}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Aerodynamic Thrust**

<table>
<thead>
<tr>
<th>Afterburner Thrust</th>
<th>Specific Impulse</th>
<th>$I_{sp}$</th>
<th>$G_{max}$</th>
<th>$G_{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Round-Earth Options**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Miscellaneous Options and Extra Filter**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

*This is the end of the document.*

---

*Leave this section blank if option not used.*

---

*Last line need must have a minus sign in column 1.*
Data for first run

JVEH flag card
Tables (if first card not all zeros)
Title card
Main set of data (DATA 1-143)

Data for second run

JVEH flag card
Tables (if different from first run)
Title card
Main data that differs from that of first run

Data for third run, etc.

Fig. 11 — Order of input
coefficient as a function of lift coefficient and Mach number and thrust as a function of altitude and Mach number.

When calling a control-law subroutine, the flags IAERO and ITHR are used to select the aerodynamic and propulsion operations pertaining to the choice of table values, analytic functions, or constants. Instructions for calling these options are given in Appendix D.

Appendix F describes the form and organization of aerodynamic and propulsion tables. A complete set of tables is also shown as an example in Fig. 35, Appendix F. The model atmosphere is calculated by analytic expressions (from Ref. 4) as given in Appendix I (there are no provisions for using table values for model atmosphere representation).

AERODYNAMICS AND PROPULSION WITHOUT TABLES

There are several possibilities for handling aerodynamic or propulsion calculations for the JVEH(I) = 0 option:

- Constants may be read in the appropriate locations (80-94) and Eq. (27) used for \( C_L \), \( C_D \), and a computations by specifying the argument IAERO = 1 when calling the control-law subroutine. (See Appendix D for instructions.)

- Constants may also be used in appropriate locations (95-109) to specify values of propulsion and fuel flow characteristics, e.g., thrust, specific impulse, burnout weight, or boost acceleration. Thrust is set equal to the data value by using the ITHR flag as explained in Appendix D for ITHR = 3, 4, or 5. Vehicle weights are determined from the expression

\[
W = W_o - \int W \, dt
\]

where \( W \) may be a value determined from fuel flow tables (ITHR = 1, 2) or a constant input value as specified in data locations 144-146 when using the ITHR = 3, 4, 5 option. However, in order to simulate a constant-acceleration boost phase of a missile, the program will automatically calculate vehicle weight from the expression \( W = W_o \exp \frac{a_B (t-t_L)}{gI} \)
where \( W \) = the current weight at time \( t \)
\( W_0 \) = the initial weight
\( t_L \) = the launch time
\( a_B \) = the boost acceleration, considered to be an average value
\( I \) = the specific impulse of the rocket motor

This automatic alternative computation is initiated when an input data value for \( a_B \) is supplied in locations 99, 104, and 109 (as applicable to vehicle 1, 2, or 3). Provisions are also made for supplying missile burnout weights in locations 98, 103, and 108, but no computations are performed on these quantities within the main body of the program itself; the locations are provided merely as a convenience for formulating subroutines requiring this form of input data.

- For those cases where aerodynamic computations for a particular vehicle are not significant to the problem, it is not necessary to read in either tables or constants for that vehicle. The IAERG = 3 argument is used (see Appendix D) and the vehicle is assumed to be a point mass moving at constant speed (but not necessarily at zero lateral acceleration).

- For specialized aerodynamic or propulsion characteristics (e.g., Missile (X), Appendix C) analytic expressions and necessary constants may be incorporated into the control-law subroutine.

**MAIN SET OF DATA**

The main set of data specifies such initial conditions as vehicle position and velocity, aerodynamic constants, structural and attitude limits, time lags, program flags (see Appendix H) and constants. Data numbers with corresponding program variables and descriptions are given in Appendix E. There are 143 separate entry spaces in the main data. If a value is not read into a space, that entry is automatically taken to be zero; therefore, only nonzero data need be specified.

The format for reading in the main data is \((A1, I3, 5E14.8)\). The initial value on the card is the number of the first data entry on the card. Entries which follow this value on the card must be in sequence. A minus sign is placed in the first column of the last data card to indicate that the entries for that case are finished. Any data following
this card are for a new case (see Fig. 10). For sample data, refer to the numerical examples in Section XII.

Initial Position and Velocity Data

As shown on the input form, data spaces are available for those flags that indicate how the position and velocity of the vehicles are to be entered: in spherical or Cartesian coordinates or (in the case of position only) in latitude and longitude. Data entries which follow are for specifying the positions and velocities of vehicles 1 and 3. Vehicle 2 is initially assumed to be a missile attached to vehicle 1; however, data locations 30 through 35 (not shown on the input form) are available for setting initial position and velocity conditions separately for vehicle 2. That is, if data entries are made in any one of these locations (30-35), TACTICS will start simultaneously computing the trajectories of all three vehicles; otherwise, vehicle 2 is initially attached to 1 and computations will be performed for only two vehicles (unless or until there is a POLICY call for launching 2).

Aerodynamic Constants

If analytic functions are to be used for aerodynamic computations, the equation constants, i.e., those applicable to Eqs. (26) and (27), must be entered. Data spaces are assigned for such constants, as shown on the input form.

Other Data

Spaces for structural and aerodynamic constraints are shown on the input form. Maximum elevation and azimuth angle gimbal limits may be specified by DATA 138-143. Up to three consecutive vehicle time lags can be entered (see Section V). Program flags and constants should be set; typically, these would specify minimum integration step, type of integration, and total time of run. If certain necessary values for program operation are not entered, TACTICS will automatically assume "default" values, print an informative message, and continue.
RUNNING MORE THAN ONE CASE

Any number of consecutive cases can be submitted for a single run. As indicated previously, the last card of the data for a case has a minus sign in the first column indicating the end of the case. To submit a second case, only data differing from values in the first case need be entered. For each case a JVEH card and comment card must be entered before the main data. With reference to Fig. 10, note that although the JVEH card contains all zeroes, the same tables are used in the second case as in the first.
XI. OUTPUT

The output from a typical computer run consists of labeled and unlabeled initial-condition data values and optional sections of printout. Initial-condition values will automatically be printed, but the printing of the optional sections must be specified in POLICY. The printing of unlabeled data consists of a listing of all locations, DATA 1-200, as shown in Fig. 12. Labeled initial conditions for these values are as shown in Fig. 13. Table 1 lists sets of available optional output sections, the most important of which is the main set or "standard output" giving position, velocity, acceleration, weight information, and other basic quantities printed at specified time intervals; these intervals are controlled by the variable DTPO (DATA 63). The other sections concern information about aerodynamics, attitude, etc. as indicated in Table 1. In order to specify these optional sections in POLICY, the variable NPRINT is set equal to the required number of output sections.

A typical output specification in POLICY would be the following:

NPRINT = 3: Three sections
IPRINT(1) = 1: Standard output
IPRINT(2) = 2: Aerodynamics
IPRINT(3) = 3: Attitude angles

A sample page of output for the NPRINT = 3 option is shown in Fig. 14. Most of the items shown are self-explanatory except perhaps the following:

1. The integration step size shown at the top of Fig. 14 is the last integration step taken before printout. If the JINTEG = 3 mode of integration is being used, two values will be printed: the step taken to reach printout time (TP0) and the step the routine would have used if there had not been an immediate print requirement. (The step taken to reach printout must necessarily be the smaller of the two.)

2. The units used are feet for distance, degrees for angles, pounds for force and weight, g's for acceleration, and radians per second for angular rates.
### Initial Data Values

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.46190000E 04</td>
<td>0.90000000E 04</td>
<td>0.20000000E 05</td>
<td>0.16990000E 05</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>16</td>
<td>0.10000000E 01</td>
<td>0.91999996E 00</td>
<td>0.90000000E 02</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>21</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>26</td>
<td>0.0</td>
<td>0.0</td>
<td>0.10000000E 01</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>31</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>36</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>41</td>
<td>0.3</td>
<td>0.0</td>
<td>0.22000000E 05</td>
<td>0.33096000E 05</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>46</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>51</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>56</td>
<td>0.86999995E 00</td>
<td>0.90000000E 02</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>61</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>66</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>71</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>76</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>81</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>86</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>91</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>96</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>101</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>111</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>116</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>121</td>
<td>0.0</td>
<td>0.30000000E 01</td>
<td>0.50000000E 01</td>
<td>0.10000000E 04</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>126</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>131</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>136</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>141</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>146</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>151</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>156</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>161</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>166</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>171</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>176</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>181</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>186</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>191</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>196</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Fig. 12 — Sample initial data values (unlabeled)
### Initial Conditions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighter</td>
<td>X(Ft) - 0.46120000E 04</td>
<td>E(Ft) - 0.80000000E 04</td>
</tr>
<tr>
<td>Missile</td>
<td>Y(Ft) - 0.46120000E 04</td>
<td>Z(Ft) - 0.80000000E 04</td>
</tr>
<tr>
<td>Target</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Relative Range

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighter-Missile</td>
<td>X(Ft) - 0.46120000E 04</td>
<td>E(Ft) - 0.80000000E 04</td>
</tr>
<tr>
<td>Fighter-Target</td>
<td>Y(Ft) - 0.46120000E 04</td>
<td>Z(Ft) - 0.80000000E 04</td>
</tr>
<tr>
<td>Missile-Target</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Relative Range Rate of Change (FT/SEC)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighter-Missile</td>
<td>X(FT/SEC) - 0.00</td>
<td>Y(FT/SEC) - 95.55</td>
</tr>
<tr>
<td>Fighter-Target</td>
<td>X(FT/SEC) - 0.00</td>
<td>Y(FT/SEC) - 95.55</td>
</tr>
<tr>
<td>Missile-Target</td>
<td>X(FT/SEC) - 0.00</td>
<td>Y(FT/SEC) - 89.37</td>
</tr>
</tbody>
</table>

### Relative Velocity

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighter-Missile</td>
<td>X(Ft) - 0.00</td>
<td>Y(Ft) - 0.00</td>
</tr>
<tr>
<td>Fighter-Target</td>
<td>X(Ft) - 0.00</td>
<td>Y(Ft) - 59.18</td>
</tr>
<tr>
<td>Missile-Target</td>
<td>X(Ft) - 0.00</td>
<td>Y(Ft) - 59.18</td>
</tr>
</tbody>
</table>

### Flags and Constants

<table>
<thead>
<tr>
<th>Time</th>
<th>DPU</th>
<th>TOTAL</th>
<th>JINTG</th>
<th>OTG</th>
<th>ERTOT</th>
<th>MINT</th>
<th>ERTOT</th>
<th>MINT</th>
<th>ERTOT</th>
<th>MINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

---

Fig. 13 — Sample initial data values (labeled)
<table>
<thead>
<tr>
<th>TIME = 0.5000 SEC</th>
<th>INTEGRATION STEP SIZE = 0.020000</th>
<th>0.160000</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGHTER-MISSILE</td>
<td>HEL. RANGE</td>
<td>HEL. VEL.</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>FIGHTER-TARGET</td>
<td>0.94705391E 04</td>
<td>56.8982</td>
</tr>
<tr>
<td>MISSILE-TARGET</td>
<td>0.44265591E 04</td>
<td>59.5982</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OMEGA</th>
<th>UMEGA</th>
<th>ACM</th>
<th>ADUT</th>
<th>ADUTA</th>
<th>ADUTO</th>
<th>GFORCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.00000</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.00000</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.00000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAMODI</th>
<th>THODIGCUS(GAMO)</th>
<th>VOUT</th>
<th>EXTR</th>
<th>EXTR</th>
<th>STFFLTY</th>
<th>CPFLTY</th>
<th>STFFLTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POSITION</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>ABS. RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.46190000E 04</td>
<td>-0.75215586E 04</td>
<td>0.20000000E 05</td>
<td>0.21861129E 05</td>
</tr>
<tr>
<td>FIGHTER</td>
<td>0.46190000E 04</td>
<td>-0.75215586E 04</td>
<td>0.20000000E 05</td>
<td>0.21861129E 05</td>
</tr>
<tr>
<td>MISSILE</td>
<td>0.14749821E 03</td>
<td>0.44666679E 03</td>
<td>0.22000000E 05</td>
<td>0.22004570E 05</td>
</tr>
<tr>
<td>TARGET</td>
<td>0.14749821E 03</td>
<td>0.44666679E 03</td>
<td>0.22000000E 05</td>
<td>0.22004570E 05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RACH</th>
<th>MACH</th>
<th>THRUST</th>
<th>WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.92754</td>
<td>1.68000</td>
<td>0.52346471E 04</td>
<td>0.16989059E 05</td>
</tr>
<tr>
<td>0.92354</td>
<td>1.70067</td>
<td>0.52346471E 04</td>
<td>0.31094575E 05</td>
</tr>
<tr>
<td>0.91806</td>
<td>1.64900</td>
<td>0.52346471E 04</td>
<td>0.31094575E 05</td>
</tr>
</tbody>
</table>

**************AERODYNAMICS**************

<table>
<thead>
<tr>
<th>CLIMAX</th>
<th>LIFT</th>
<th>CD</th>
<th>DRAG</th>
<th>ALPHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.93970</td>
<td>1.64273</td>
<td>0.02518</td>
<td>2457.08545</td>
<td>2.06763</td>
</tr>
<tr>
<td>0.90155</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.80712</td>
<td>0.11322</td>
<td>3.2648.52344</td>
<td>0.02323</td>
<td>4095.91062</td>
</tr>
</tbody>
</table>

**************ATTITUDE ANGLES**************

<table>
<thead>
<tr>
<th>FIGHTER-MISSILE</th>
<th>ELEV</th>
<th>AZIMUTH</th>
<th>BANK</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.4</td>
<td>-24.9</td>
<td>31.5</td>
<td>0.0</td>
</tr>
<tr>
<td>FIGHTER-TARGET</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MISSILE-TARGET</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MISSILE-FIGHTER</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>TARGET-FIGHTER</td>
<td>-54.4</td>
<td>150.2</td>
<td>148.6</td>
</tr>
<tr>
<td>TARGET-MISSILE</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Fig. 14 — Sample page of output for NPRINT = 3
3. $\mathbf{GAM}\dot{\theta}$ and $\mathbf{THD}\dot{\theta}\mathbf{COS}$ (GAM) are the angular rates of rotation of the vehicles' velocity vectors.

4. $\mathbf{\Omega}$MECAR refers to the angular rates of rotation of the LOS RREL(I,J). $\mathbf{\Omega}$MEGA3 output is provided for guidance laws that use a biased angular-rate term (see Appendix C).

5. An EXTR FORMAT (E16.8) is provided for six extra quantities (middle right of standard output package). This is extremely useful for debugging and printing additional information.

6. A CLMAX or ASMAX print (adjacent to acceleration quantities) will occur whenever these respective limits are exceeded. If no input data value has been given to CLMAX or ASMAX, these values are assumed to be infinite (e.g., $10^6$).

7. PHRREL and THRREL (bottom right) refer to the orientation angles $\phi$ and $\theta$, respectively, of the LOS, i.e., RREL(I,J).

Table 1

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1. Standard output</td>
<td>Position, velocity, acceleration, and other basic quantities</td>
</tr>
<tr>
<td>2</td>
<td>2. Aerodynamics</td>
<td>Lift, maximum lift and drag coefficients, angle of attack, lift, and drag</td>
</tr>
<tr>
<td>3</td>
<td>3. Attitude angles</td>
<td>Roll, bank, elevation, azimuth, and bearing angle</td>
</tr>
<tr>
<td>4</td>
<td>4. Round-earth</td>
<td>Latitude, longitude, and altitude</td>
</tr>
<tr>
<td></td>
<td>coordinates</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5. Initial conditions</td>
<td>Input data</td>
</tr>
<tr>
<td>6</td>
<td>6. Atmospheric</td>
<td>Air pressure, temperature, air density, speed of sound, and Mach number</td>
</tr>
<tr>
<td>9</td>
<td>9. Angular rates</td>
<td>Angular-rate vectors and unit vectors $\mathbf{T}_A$, $\mathbf{T}_D$, $\mathbf{T}<em>V$ and $\mathbf{T}</em>{LV}$</td>
</tr>
</tbody>
</table>

NOTE: Sections 7, 8, and 10 through 18 are blanks available to the user if different types of output are desired.
XII. EXAMPLES

Five examples indicating the type of program which can be run on TACTICS and the way the program can be used are given in this section. Each problem is defined by initial-condition input data and by a POLICY subroutine which dictates the control laws governing the flight of each vehicle. The examples show how a POLICY subroutine is set up and what input data are necessary for a particular problem. The examples selected cover the different options available in TACTICS by employing a number of the special features included in the program (e.g., restore and recall). These examples range from the simple case of vehicles flying maneuvers with no missile to the more complicated one of launching two missiles.

EXAMPLE 1: AIRCRAFT MANEUVERS, NO MISSILE

The flights of two vehicles in different maneuvers are simulated; no missile is involved. In setting up this example, as with all the others, two steps are necessary: (1) developing the POLICY subroutine and (2) entering initial-condition data.

POLICY Subroutine

The following POLICY statements define the problem:

Vehicle 1 (Constant-Speed Aircraft: No Aerodynamic or Propulsion Computations)

- Straight flight.
- After 0.1 sec, pull a 4-g climb.
- When vehicle has climbed 30 deg, level off to horizontal and continue on straight flight.

Vehicle 2

- Not used.
Vehicle 3 (Constant-Speed Aircraft: No Aerodynamic or Propulsion Computations)

- Fly straight flight.
- After 1 sec, perform a barrel roll pulling 4 g's and rolling 60 deg/sec.
- When roll is completed, perform 4-g diving turn with 130-deg roll.

These statements are translated into FORTRAN expressions to formulate the PÔICY subroutine. Figure 15 is a listing of the routine.

The type of output desired is first specified in PÔICY (see Section XI). In this example, only the standard output and attitude angles are to be printed, since no aerodynamic or propulsion computations are being performed.

Commands governing the first vehicle's flight are given. There are different control laws and criteria for changing maneuvers. For each maneuver the type of aerodynamic and thrust computation must be specified in the argument following the maneuver name (see Appendix D). For this example, no aerodynamic or propulsion computations are involved, so IAERØ = 3 and ITHR = 5.

CLIMB1 (simple climb) is called, specifying 4 g's in the argument, until vehicle 1 has climbed 30 deg. The criterion used in this case is V(1,6), the flight-path angle γ₁. STRVL1 is called to level off the vehicle. Flag LEVEL(1) = 2 indicates that the aircraft is horizontal again.

Since the second vehicle is not being used, CAPFLT(2,2) is called, which sets all values pertaining to the vehicle equal to zero.

The motion of the third vehicle is defined by calling STRFLT (IAERØ = 3, ITHR = 5) and then changing to BRLRL1 (barrel roll). In the argument listing for BRLRL1, the number of rolls (1), the g's to be pulled (4), and the roll rate (60 deg/sec) are specified. Flag IRROLL(3) = 2 indicates that the number of rolls required is completed. This is used as the criterion in PÔICY for switching to subroutine RTRNS, a 4-g diving turn with a 130-deg roll. See Appendix D for further instructions on the calling of individual maneuvers.
Sample policy for flying aircraft maneuvers, no missile

C
C Specify output
NPRINT=2
IPRINT(1)=1
IPRINT(2)=3

C
C FIGHTER COMMANDS
GO TO (110,120,130,140),JPOL
110 CONTINUE
IF (TIME .GE. 0.1) GO TO 120
CALL STRFLT(1,3,5) *FIGHTER FLIES STRAIGHT FLIGHT FOR .1 SEC, ZERO THRUST
GO TO 190
120 CONTINUE
IF (V(I,6) *GT. 30.*RAD) GO TO 130
CALL CLIMB(1,4,0.3,5) *FIGHTER PULLS 4 G CLIMB UNTIL IT HAS
JPOL=2 CLIMBED 30 DEG, AT WHICH TIME IT
GO TO 190 BEGINS LEVELING OFF
130 CONTINUE
IF (LEVEL(I) *EQ. 2) GO TO 140
CALL STRLV(1,3,5,LEVEL) *WHEN LEVEL(I)=2, VEHICLE HAS LEVELED
JPOL=3 OFF TO HORIZONTAL (V(I,6)=0) AND
GO TO 190 FLIES STRAIGHT
140 CONTINUE
CALL STRFLT(1,3,5)
JPOL=4
190 CONTINUE

C
C MISSILE COMMANDS
GO TO (210,220,230),KPOL
210 CONTINUE
CALL CAPFLT(2,2) *MISSILE NOT BEING USED SO QUANTITIES
GO TO 290 ZERODED OUT
220 CONTINUE
230 CONTINUE
290 CONTINUE

C
C TARGET COMMANDS
GO TO (310,320,330),LPOL
310 CONTINUE
IF (TIME .GE. 1.0) GO TO 320 *TARGET FLIES STRAIGHT FLIGHT FOR
CALL STRFLT(3,3,5) 1 SEC
GO TO 390
320 CONTINUE
CALL BRLRL(3,1.0,IROLL,4.0,60.0,0,3,5) *TARGET THEN MAKES ONE
IF (TROLL(I) *EQ. 2) GO TO 330 BARREL ROLL (4G ROLL
LPOL=2 60 DEG/SEC), IROLL(I)=2
GO TO 390 INDICATES ROLLS ASKED
330 CONTINUE FOR ARE COMPLETED
CALL RTNR5(3,4.0,130.0,0,3,5) *TARGET PULLS 4 G DIVING TURN
LPOL=3 WITH 130 DEG ROLL
390 CONTINUE

C
RETURN
END

Fig. 15 — Sample POLICY subroutine for aircraft maneuvers, no missiles
Initial Data

After the POLICY subroutine is written, the input data for setting up the initial conditions of the problem must be read in. The initial position and velocity of the vehicles must be specified. In this example, vehicle 1 is situated at the origin at an altitude of 15,000 ft. Vehicle 3 is 8000 ft down the y-axis at an altitude of 20,000 ft. The velocity of vehicle 1 is 1340 ft/sec; it is headed down the -x-axis (V(1,5) = 180.0 deg). Vehicle 2 has a velocity of 1000 ft/sec, and it is headed down the y-axis (V(3,5) = 90 deg). See Fig. 16. Other data used in this example are the following:

Starting value for integration step size (DTΘ) = 0.01.
Structural lateral acceleration limit of vehicles = 7.5, 7.5.
Area of vehicles (AREA) = 196.0, 385.0.
Initial weight of vehicles (W0) = 16,669.0, 33,283.0.
Time interval for printing output (DTPO) = 0.5.
Time value at which program is to stop (TOUT) = 25.0.

Figure 17 shows how the data are entered on the input form.

EXAMPLE 2: LAUNCHING ONE MISSILE

The first vehicle (fighter) is to pursue the third vehicle (target) until a launching position is reached. At this time, the missile is launched from the fighter and the program is to find the closest miss distance of the missile and then terminate. Aircraft tables are to be used to simulate the aerodynamics of vehicles 1 and 3.

POLICY Subroutine

The following POLICY statements define the problem:

Vehicle 1 (F-104 Interceptor Aircraft)

- Fly lead collision navigation course; military thrust.
- If the range to target is less than 6500 ft and the angle off the target's tail is less than 30 deg, launch missile.
Fig. 16 — Three-dimensional diagram of initial conditions for Example 1

15,000 ft
180 deg
8000 ft
90 deg
20,000 ft

(0,0,0)
Fig. 17 — Input data for Example 1
Pull constant-Mach-number, constant-altitude right turn; thrust, afterburner.

Vehicle 2 (Proportional Navigation Missile)

- Fly captive flight until launch criterion is satisfied.
- Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile subroutine.
- When range rate (missile-target) becomes greater than zero, initiate process for finding miss distance and end program.

Vehicle 3 (F-106 Target Aircraft)

- Fly straight and level; thrust, 80-percent throttle setting, military power.
- When missile is launched, perform 5.5-g constant-Mach-number left turn; thrust, afterburner.

See Fig. 18 for the actual P\(\text{S\text{\textsuperscript{\textregistered}}}\)ICY subroutine. The output specified is standard output, aerodynamics, and attitude-angle sections.

LEADCOL (lead collision) is called for the fighter, with \(\text{IAERNO} = 2\), indicating tables for aerodynamic functions, and \(\text{ITHR} = 2\), indicating tables for military thrust. \(\text{ILAUN} = 3\) is a flag indicating that the missile has been launched; it is used in this case as the criterion for the fighter switching to RTRN2 (constant-Mach-number, constant-altitude right turn); \(\text{ITHR} = 1\) indicates tables for afterburner thrust.

The missile is being held in CAPFLT(2,1), captive flight on the fighter, until the range between fighter and target (RREL(2,4)) is less than 6500 ft and the bearing angle between fighter and target is less than 30 deg. The missile is launched with boost (DELV) equal to zero, no aerodynamics (IAERNO = 3), and zero thrust (ITHR = 5). The MISILX routine is called to simulate the aerodynamics and flight of a proportional navigation missile.

The program automatically computes the closest miss distance if the missile comes within MINMR (DATA 124) of the target. This data number should be set at a large value (e.g., 1000 ft). The program will terminate after finding the miss distance unless otherwise specified.
SAMPLE POLICY FOR LAUNCHING ONE MISSILE

**** SPECIFY OUTPUT
NPRINT=3
IPRINT(1)=1
IPRINT(2)=2
IPRINT(3)=3

******** FIGHTER COMMANDS *********
GO TO (110,120,130), JPOL
110 CONTINUE
IF ILAUN .EQ. 3) GO TO 120 *ILAUN=3 AT LAUNCH
CALL LEADCL1(1,0.0,6500.0C,2,2)
GO TO 190 *FIGHTER FLYING LEAD COLLISION NAVIGATION,
120 CONTINUE
CALL TRN2(1,2,1) MILITARY THRUST, UNTIL LAUNCH - THEN
JPOL=2 SWITCHES TO CONSTANT MACH RIGHT TURN,
AFTER-BURNER THRUST
130 CONTINUE
190 CONTINUE

******** MISSILE COMMANDS ********
GO TO (210,220,230), KPOL
210 CONTINUE
IF ( ABS(BEARNG(2)) .LT. 30.0*RAD .AND. RELR(2,4) .LT. 6500.0 )
GO TO 220 *MISSILE LAUNCHED IF RELATIVE RANGE BETWEEN
220 CONTINUE
CALL CAPFLT(2,1) FIGHTER-TARGET IS LESS THAN 6500 FT AND
230 CONTINUE
CALL LAUNCH(2,0.0,3,5) FIGHTER IS WITHIN 30 DEG OFF TARGETS TAIL
290 CONTINUE

******** TARGET COMMANDS *********
GO TO (310,320,330), LPOL
310 CONTINUE
IF IILAUN .EQ. 3) GO TO 320
THROTL(3)=.8 *THROTL PROPORTIONS THRUST
CALL STRTLT(3,2,2)
GO TO 390 *TARGET ON STRAIGHT FLIGHT UNTIL LAUNCH,
320 CONTINUE
CALL TRN3(3,5.5,2,1) THEN PULLS 5.5 G CONSTANT MACH LEFT
LPOL=2 TURN, AFTER-BURNER THRUST
330 CONTINUE
390 CONTINUE
RETURN
END

---70---

Fig. 18 — Sample POLICY subroutine for launching one missile
Target commands are the following: Call STRFLT(IABER = 2, ITHR = 2) until the missile is launched (ILAIN = 3). THRCTL(3) is set equal to 0.8 to give an 80-percent throttle setting. The target then pulls LTRN3 (constant-Mach-number left turn), where the g's are specified as 5.5 in the argument. ITHR = 1 indicates that tables are to be used for obtaining a value for afterburner thrust. See Appendix D for further instructions on the calling of individual maneuvers.

Initial Data

Since table values are used for the aerodynamic and propulsion computations, data decks for the F-104 and F-105 aircraft are entered following the JVEH flag card, which indicates which vehicles have tables.

Figure 19 shows the initial position and velocity of the vehicles as entered in the data. The fighter is at the origin at an altitude of 20,000 ft. The target is 5000 ft down the -x-axis, 9000 ft along the y-axis, and at 25,000 ft altitude. In this case, velocity is read in as Mach number instead of as ft/sec. To do this, JATMOS (DATA 20) is set equal to 1. Only the magnitude of the fighter's velocity (i.e., Mach 1.2) is read in, and the program automatically aims the fighter at the target for a lead collision course. The aiming routine is triggered by setting IVF (DATA 16) equal to 2. (See Appendix E for details on the flags for reading position and velocity.) The target is flying Mach 0.9, heading down the y-axis ($V(3,5) = 90$ deg). Other data used in this example are the following:

Starting value for integration step size ($DT$) = 0.01.
Structural lateral acceleration limit of aircraft ($ASMAX$) = 7.3, 8.0.
Navigation constant for closed-loop guidance routines ($LANDAO$) = 4.0, 4.0.
Area of aircraft ($AREA$) = 196.0, 385.0.
Initial weight of aircraft ($W$) = 16,699.0, 33,287.0.
Time interval for printing output ($DTPO$) = 0.5.
Time value at which program is to stop ($TOTAL$) = 25.0.
Flag specifying that initial-condition value of velocity of aircraft is expressed as Mach number (JATMOS) = 1.
Fig. 19 — Three-dimensional diagram of initial conditions for Example 2
Missile range to target within which program will automatically initiate process for miss-distance computation \( (\text{MINMR}) = 1000.0 \text{ ft.} \)

Since missile parameters such as weight and area are defined within the missile subroutine, their values do not have to be read in. The initial position and velocity of the missile are set equal to that of the fighter.

Figure 20 shows the initial data for this example entered on the input form. Refer to Section X for instructions on preparing such a data package.

**EXAMPLE 3: GROUND-LAUNCHED MISSILE**

A SAM is launched from the ground at a constant-velocity target. Analytic functions are used to compute the missile's aerodynamics. The closest miss distance between missile and target is to be found and the program terminated.

**POLICY Subroutine**

The following POLICY statements define the problem:

**Vehicle 1**

- Not used.

**Vehicle 2 (SAM)**

- Launch missile at time zero with a constant thrust of 5397 lb and an initial velocity of 100 ft/sec.
- Fly in 0-g dive until burnout at 1.0 sec.
- Continue 0-g dive until time to guide; thrust: 0.0.
- Begin guidance 1.0 sec after launch, fly proportional navigation with a time lag of 0.2 sec; thrust: 0.0.

**Vehicle 3 (Target Aircraft)**

- Fly straight and level at constant speed, no aerodynamic or propulsion computations.
Figure 21 shows the POLICY subroutine. The printout is to consist of standard output, aerodynamics, and attitude angles. Because the fighter is not used in this run, its quantities are set to zero by calling CAPFLT(1,2).

The missile is launched at time zero; DIVE1 with 0 g's is called (i.e., a ballistic trajectory). Analytic functions are used to compute aerodynamics, so IAERO = 1; constant thrust is indicated, so ITHR = 4. TLAUN(2) is the time of launch and can be used as a criterion in POLICY. Here it is used both to determine burnout (TBURN(1)) and the time the missile is to begin guiding (TGUIDE(2)). Missile guidance begins by calling PROPNAV (proportional navigation). Thrust now equals zero, so ITHR = 5. A time lag is added to the missile at this point by setting ITAU(2) = 1, which indicates that only one time lag is being used; TAU(2,1) = 0.2, indicating a 0.2-sec lag. STRFLT is called for the target with no aerodynamic or thrust computations, so IAERO = 3 and ITHR = 5. See Appendix D for further instructions on the calling of individual maneuvers.

Initial Data

Figure 22 is a three-dimensional diagram of the initial positions and velocities of the missile and target specified in the input data. The program is capable of reading in values for the position and velocity of the missile, but in this case the initial conditions are set to those of the fighter instead, i.e., the missile's values are read in as the fighter's position and velocity. The fighter values are later zeroed out, since they are not used in this problem. This seemingly needless complexity may be used to advantage for automatic aiming of vehicle 1 against 3 or vice versa using the flags IVF = 2 or IVF = 2. This will automatically call the AIM routine and, in this case, provide for initially aiming 2 against 3 (see the description of subroutine AIM in Appendix C).

The missile is initially placed at the origin, and the altitude component is accordingly zero. The target begins its flight 3000 ft down the -x-axis, 10,000 ft along the y-axis, and at a 500-ft altitude.
SAMPLE POLICY FOR GROUND LAUNCHED MISSILE

*** SPECIFY CUTPLT
APRINT=3
IPRINT(1)=1
IPRINT(2)=2
IPRINT(3)=3

******** FIGHTER COMMANDS *********
CC TC (11C, 12C, 13C), JPCL
11C CONTINUE
CALL CAPF(1, 2)  *FIGHTER ACT BEING USED SC QUANTITIES
CC TC 14C
ZBCEC CLT
12C CONTINUE
13C CONTINUE
19C CONTINUE

******** MISSILE COMMANDS *********
CC TC (21C, 22C, 23C, 24C, 25C), KPCL
21C CONTINUE
CALL LANCH(2, C, C, 3, 5)  *MISSILE LAUNCHED AT TIME ZERC
22C CONTINUE
IF ((TIME-TLAEM(2)) GT. TBLRN1) GO TO 23C
CALL CIVE(2, C, C, 1, 5)  *MISSILE FLIES IN ZERO G DIVE WITH
KPCL=2
THRLST=CONSTANT UNTIL BURNLT(TBLRN1)
CC TC 24C
, AT WHICH TIME THIRST SET TO ZERC
AND MISSILE CONTINUES UNGUIDED
23C CONTINUE
IF ((TIME-TLAEM(?)) GT. TGLICE(2)) GO TO 24C
CALL CIVE(2, C, C, 1, 5)
KPCL=3
CC TC 25C
24C CONTINUE
ITL(2)=1
TAL(2, 1)=.2
25C CONTINUE
CALL FRCNAV(2, 1, 5)
KPCL=5
29C CONTINUE

******** TARGET COMMANDS *********
CC TC (31C, 32C, 33C), LPCL
31C CONTINUE
CALL SHTFLT(3, 3, 5)  *TARGET FLIES STRAIGHT FLIGHT
CC TC 35C
32C CONTINUE
33C CONTINUE
39C CONTINUE

RETURN
END

Fig. 21 — Sample POLICY subroutine for ground-launched missile
Fig. 22 — Three-dimensional diagram of initial conditions for Example 3
The missile's initial velocity is 100 ft/sec, and it is pointed directly down the LOS between missile and target, giving it a heading of 98.9 deg in the horizontal plane (V(1,5)) and 2.55 deg in the vertical plane (V(1,6)). The target has a velocity of 1340 ft/sec headed down the y-axis (V(3,6) = 90 deg).

A significant quantity of aerodynamic data must be entered for this case, since analytic functions are used to compute the missile aerodynamics, and equation constants must therefore be specified for use with Eq. (26). The following data are used in this example:

Starting value for integration step size (DTØ) = 0.01.
Time interval missile is to fly unguided after launch (TGUIDE(2)) = 1.0.
First-stage burning time of missile (TBURN1) = 1.0 sec.
Structural lateral acceleration limit of missile and target (ASMAX) = 15.0, 7.3 g's.
Constants to be used for thrust (THCOS(2)) = 5397.
Navigation constant for closed-loop guidance routines of missile (LAMDAO(2)) = 4.0.
Maximum aerodynamic lift coefficient of missile (CLMAX(2)) = 1.4.
\[ d(C_D)/d(C_L) \], constant of missile (BCOSN(2)) = 0.0041666667.
\[ dC_L/dn \] (assumed to be a constant) SLOPE(2) = 0.1.
Specific impulse of rocket motor (IMPLSE(2))* = 250.0.
Initial weight of missile (WO)* = 50.0.
Area of missile (AREA) = 0.264.
Integration mode (in this case, fixed-step Adams-Moulton) (JINTEG) = 2.
Time interval for printing output (DTFØ) = 0.5.
Time value at which program is to stop (TOTAL) = 25.0.
Missile range to target within which program will automatically initiate process for miss-distance computation (MINMR) = 1000.0.

Figure 23 shows the initial data for this example entered on the input form. Refer to Section X for instructions on preparing such a data package.

*These quantities must be set for computing missile weight during the propulsion interval.
**TACTICS PROGRAM: INPUT FORM**

**VEHICLE TABLE**

<table>
<thead>
<tr>
<th>Flags</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>12</td>
<td>18</td>
</tr>
</tbody>
</table>

**TITLE CARD**

<table>
<thead>
<tr>
<th>CARD</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
<th>65</th>
<th>70</th>
<th>75</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MAIN SET OF INITIAL CONDITIONS**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Initial Weight**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RELATIVE COMPONENTS**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**VEHICLE DATA**

<table>
<thead>
<tr>
<th>VEHICLE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GUIDANCE (3 VEHICLES)**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AERODYNAMICS AND PROPULSION (3 VEHICLES)**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EARTH-ORBIT OPTIONS**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MISCELLANEOUS OPTIONS AND EXTRA INITIALS**

<table>
<thead>
<tr>
<th>CARD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The following units are used: distance (mi), time (sec), velocity (ft/sec or Mach no.), acceleration (ft/sec), weight (lbm), area (sq. ft),

This card required.

*Enter this sign. blank if option not used.

*Last data card must have a minus sign in column 1.

Figure 23 — Input data for Example 3
EXAMPLE 4: LAUNCHING TWO MISSILES, RECALL

The recall features of the program provide the capability of launching more than one missile. After the missile has been launched the first time and the miss distance determined, the program continues instead of terminating, and the missile can be recalled either to vehicle 1 or to vehicle 3. The aerial combat is then continued and the missile is launched again. Therefore, if vehicle 1 misses on the first launching, it can repeat the process by launching another missile, or vehicle 3 can retaliate by launching a missile at vehicle 1.

In this example, vehicle 1 is pursuing vehicle 3 and launches a missile when the launch criteria have been satisfied. After the missile has missed the target it is recalled to vehicle 3, and now vehicle 1 is the target. The missile is now launched against vehicle 1.

POLICY Subroutine

The following POLICY statements define the problem:

Vehicle 1 (F-104 Intercepting Aircraft)

- Fly pursuit-course navigation; thrust, military power.
- If the range to target is less than 7000 ft, launch missile.
- After launch perform constant-Mach-number, constant-altitude left turn; thrust, afterburner.
- When heading angle in horizontal plane is greater than 270 deg, fly straight and level; thrust, military.

Vehicle 2 (First Missile)

- Hold in captive flight by fighter until launch criterion is satisfied.
- Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile routine.
- When range rate (missile-target) becomes greater than zero, initiate process for finding miss distance.
Vehicle 3 (F-105 Target Aircraft)

- Fly straight and level; thrust, military.
- When missile is launched, perform 5-g left turn; thrust, military.
- After fighter-launched missile has missed, switch to pursuit-course navigation tracking vehicle 1; thrust, afterburner.
- If range to fighter is less than 10,000 ft and greater than 7000 ft and the target is within 30 deg off fighter's tail, launch second missile, which has been recalled to vehicle 3.

Vehicle 4 (Second Missile)

- Missile recalled to vehicle 3 after first missile has missed.
- Hold in captive flight by target until launch criterion is satisfied.
- Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile subroutine.
- When range rate (missile-fighter) becomes greater than zero, initiate process for finding miss distance and terminate program.

Figure 24 shows the POLICY subroutine. The printout is to consist of standard output, aerodynamics, and attitude angles.

Fighter flies PRSUIT (pursuit-course navigation) with table values for aerodynamics (IAERO = 2) and table values for military thrust (ITHR = 2). At launch time (ILAUN = 3 indicates that missile has been launched) LTRN2 (constant-Mach-number, constant-altitude left turn) is called with afterburner thrust (ITHR = 1). Fighter flies LTRN2 until its horizontal heading (V(1,5)) is greater than 270 deg. This is used as the criterion for switching to STRFLT.

The first missile is held in CAPFLT(2,1), captive flight by fighter, until the range between fighter and target (RRDEL(2,4)) is less than 7000 ft. The missile is launched, and the MISILL routine is called to simulate the aerodynamics and flight of a proportional navigation missile.

To initiate the recall feature of the program, it is necessary to set the flag IMISS = 1, which causes the program to continue after finding the first missile miss. It is reset to zero for the second
Sample Policy for Launching Two Missiles

**** Specify CLTPLT
KPRINT=3
IPRINT(1)=1
IPRINT(2)=2
IPRINT(3)=3

**** Fighter Commands ******

Call PLTLT(1,2,2) *Fighter flies pursuit course navigation, military
GC TC 15C
Thrust, until missile launched

11C CONTINUE
If (ILALN .EQ. 3) GC TC 12C
*ILALN=3 at launch
Call PLTLT(1,2,2) *Fighter flies pursuit course navigation, military
GC TC 15C
Thrust, until missile launched

12C CONTINUE
If (V(1,5) .GT. 270.0*RACI) GC TC 13C
Call LTRN(1,2,1) *At launch fighter switches to constant Mach,
JPLC=2
constant altitude left turn, after-burner thrust
GC TC 15C

13C CONTINUE
Call SRFNT(1,2,2)
*When heading angle in horizontal plane (Theta)
JPLC=3
is greater than 270 deg, fighter flies straight
flight, military thrust

19C CONTINUE

**** Missile Commands ******

Call PLTLT(1,2,2) *Flag for program to continue after missile miss
GC TC 25C

21C CONTINUE
If (KMISS .EQ. 1) GC TC 25C
*If relative range between fighter-target is less
IPRINT=1
than 700 ft., launch missile. Otherwise miss!
GC TC 25C

22C CONTINUE
Call LAUNCH(2,2,C,3,5)
*Flag for program to continue after
GC TC 25C
Finding minimum miss distance of first missile

23C CONTINUE
If (KMISS .EQ. 2) GC TC 24C
*KMISS=1 indicates second missile is to be used
Call MISSIL(2)
Call MISSIL(2) *If KMISS=2 program is ready to continue after
KPLC=3
Finding minimum miss distance of first missile
GC TC 25C

24C CONTINUE
KMISS=1
KPLC=1

29C CONTINUE

**** Target Commands ******

Continued

Fig. 24 — Sample POLICY subroutine for launching two missiles
CC TC (310,320,330),LPCL
310 CONTINUE
   IF (ILACA .EQ. 3) GC TC 320
   CALL STFLIT(2,2,2) *TARGET FLIES STRAIGHT FLIGHT, MILITARY THRUST
   GC TC 390
320 CONTINUE
   IF (KMISS .EQ. 1) GC TC 320
   CALL LTRA1(3,5,C,2,2) *AFTER LAUNCH, TARGET PULLS 5 G LEFT TURN,
   LPCL=2
   MILITARY THRUST
   GC TC 390
330 CONTINUE
   CALL PSPLIT(3,2,1) *AFTER FIGHTER LAUNCHED MISSILE HAS MISSED, TARG
   LPCL=3
   TARGET Switches TO PSPLIT COURSE NAVIGATION,
390 CONTINUE
   ~AFTER BLANKER THRUST
C
C ******* SECOND MISSLE COMMANDS *******
CC TC (410,420,430,440),HPCL
410 CONTINUE
   IF (KMISS .NE. 1) GC TC 430
   IMISS=0
420 CONTINUE
   *FLAG FOR PROGRAM TO STOP AFTER FINDING MISSED
   IMISS=0
   MISS DISTANCE FOR SECOND MISSILE
   IF (PREL(2,4) .LT. 10000.CC .AND. PREL(2,4) .GT. 7000.CC .AND.
   IABS(HEADNG(2) .LT. 150.CC .AND.) TC 430
   CALL CAFILT(2,3) *MISSILE LAUNCHED IF RELATIVE RANGE BETWEEN TAR
   HPCL=2
   -FIGHTER IS GREATER THAN 7000 FT. AND LESS THE
   GC TC 490
   10000 FT., AND THE TARGET IS WITHIN 30 DEG OF
430 CONTINUE
   FIGHTERS TAIL. OTHERWISE MISSLE IS HELD IN
   CALL LALACH(2,C.C,2,5) CAPTIVE FLIGHT BY TARGET
440 CONTINUE
   CALL MISII(2)
   HPCL=4
490 CONTINUE
C
C  RETURN
ENC

Fig. 24 (continued)
missile, and the program will terminate after the miss-distance calculation. IMISS = 2 is a flag set in the program to indicate that the miss distance for the first missile launch has been computed and that the program is ready to continue. This flag is used as the criterion in POLICY for triggering the flag KM1SS = 1, which indicates that the first missile is no longer in flight and that the second missile's commands should now be followed. (See the first and second missile command sections in POLICY, Fig. 24.)

The second missile is held in captive flight by target CAPFLT(2,3) until the target-fighter range (RREL(2,4)) is greater than 7000 ft but less than 10,000 ft and the fighter-target bearing angle is greater than 150 deg. (See Section XI for the definition of bearing angle.) The missile is launched against the fighter and MISIL1 routine called. Flag IMISS = 0 indicates that the program will stop after determining miss distance.

Target flies STRFLT (IAER = 2, ITHR = 2) until the first missile is launched (ILAUN = 3). It then pulls LTRNL (simple left turn) with 5 g's specified until the first missile has missed and is recalled to the target (this is indicated by KM1SS = 1). Target now flies PRSUIT against the fighter attempting to arrive within firing range at afterburner thrust, ITHR = 1. See Appendix D for instructions on calling the optional subroutines used in POLICY.

**Initial Data**

Since table values are used for computing aerodynamics, data decks for the F-104 and F-105 are entered following the JVEH flag card, as shown in Fig. 11.

Figure 25 is a three-dimensional diagram of the initial positions and velocities used in this example. As can be seen, the fighter is 6000 ft along the y-axis at 15,000 ft altitude. It has a velocity of Mach 0.92 (velocity is given in Mach number instead of ft/sec if JATMOS (DATA 20) is set equal to 1) and a horizontal heading of 210.0 deg (V(1,5)). The target's initial position is 8000 ft down the -x-axis and at 15,000 ft altitude. Initial velocity is Mach 0.87 with 180-deg heading angle (V(3,5)). Other data needed for this example are the following:
Fig. 25 — Three-dimensional diagram of initial conditions for Example 4
Starting value for integration step size (DTΦ) = 0.01.
Structural lateral acceleration limit of aircraft (AMAX) = 6.8, 7.0.
Navigation constants for closed-loop guidance routines (LAMDAO) = 40.0, 40.0.
Area of aircraft (AREA) = 196.0, 385.0.
Initial weight of aircraft (W0) = 16,699.0, 33,283.0.
Time interval for printing output (DTPΦ) = 0.5.
Time value at which program is to stop (TOTAL) = 60.0.
Flag specifying that initial-condition value of velocity of aircraft is expressed as Mach number (JATMΦS) = 1.
Missile range to target within which program will automatically initiate process for miss-distance computation (MNMR) = 1000.0.

Because missile constants and variables are defined within the missile subroutines, their values do not have to be read in. The initial position and velocity of the missile are set equal to those of the fighter.

Figure 26 shows the initial data for this example as entered on the input form. Refer to Section X for instructions for preparing such a data package.

EXAMPLE 5: USING RESTORE FEATURE TO FIRE 20-MM PROJECTILES

After vehicle 2 is launched and a hit or miss has occurred, this option enables the program to restore all numerical values existing at launch time. Now events may proceed, a new launch may take place, and characteristics and parameters may change. In this case the restore feature is used to fire consecutive projectiles at 0.01-sec intervals at a ground target. A projectile is launched first at a given range and the miss distance computed. The numerical values are then restored to those existing at launch time. Integration then occurs, and the next projectile may be launched at a subsequent time.

POLICY Subroutine

The following POLICY statements define the problem:
Fig. 26 — Input data for Example 4
Vehicle 1 (F-104 Pursuing Aircraft)

- Fly pursuit-course navigation against a stationary ground target; thrust, military.
- When a ground range of 3000 ft has been covered, begin launching 20-mm projectiles at 0.01-sec intervals.
- If altitude falls below 5000 ft, perform a 4-g climb; thrust, afterburner.

Vehicle 2 (20-mm Projectile)

- Fly captive flight until launch criterion is satisfied.
- Launch with an incremental velocity of Δv and fly in accordance with characteristics specified in B20MM subroutine, i.e., a ballistic trajectory.
- Find miss distance at point of ground impact, restore program to values of launch time, launch second projectile 0.01 sec later, and continue until five projectiles have been fired.

Vehicle 3 (Stationary Ground Target)

- Remain stationary on ground.

Figure 27 shows the POLICY subroutine. The printout is to consist of standard output, aerodynamics, and attitude angles. To use the restore feature of the program, flag ISTORE = 1 must be set. Flag DMISS = 1 must also be specified so that the program will not stop after computing the missile miss distance. Both these flags are set the first time through POLICY and must be reset for each separate launch.

The fighter is flying PRSUIT (pursuit-course navigation) with table values for aerodynamics (IATR8 = 2) and table values for military thrust (ITHR = 2). If its altitude falls below 5000 ft, the fighter begins a 4-g climb (CLIMB1) with afterburner thrust (ITHR = 1) in order to pull out. The projectiles are held in captive flight by the fighter (CAPFLT(2,1)), until the fighter has flown 3000 ft along the y-axis; i.e., the y-component (R(1,2)) is greater than 3000. Each projectile is launched with a boost velocity (DELY) of 2800 ft/sec, and the B20MM subroutine is called to determine the aerodynamics and flight path.
SAMPLE POLICY USING STORE TO FIRE 20MM CANNON PROJECTILES

C

C**** SPECIFY OUTPUT
NPRINT=3
IPRINT(1)=1
IPRINT(2)=2
IPRINT(3)=3

C

C ************ FIGHTER COMMANDS ************

GO TO (110,120,130), JPOL

110 CONTINUE
IMISS=1
ISTORE=1

120 CONTINUE
IF (R(1,3) <= 5000.0) GO TO 130
CALL PRSUIT(1,2,2)
JPOL=2
GO TO 190

130 CONTINUE
CALL CLIMB1(1,4,0,2,1)

140 CONTINUE
*IF ALTITUDE IS LESS THAN 5000 FT, SWITCH TO 4 G CLIMB TO PULL OUT, AFTER-BURNER THRUST

190 CONTINUE

C ************ MISSILE COMMANDS ************

GO TO (210,220,230,240,250), KPOL

210 CONTINUE
IF (R(1,2) >= 3000.0) GO TO 220
CALL CAPFLT(2,1)
GO TO 290

220 CONTINUE
CALL LAUNCH(2,2800.0,3,5)

230 CONTINUE
IF (ISTORE .EQ. 0) GO TO 240
KPOL=3
GO TO 290

240 CONTINUE
JINTEG=2
IMISS=1
IF (II .EQ. 3) IMISS=0
ISTORE=1
KPOL=5
KK=0

250 CONTINUE
IF (KK .NE. 0) GO TO 220
CALL CAPFLT(2,1)
KK=1
II=II+1
GO TO 290

C ***************TARGET COMMANDS ***************

GO TO (310,320,330,340), LPOL

310 CONTINUE
GO TO 390

390 CONTINUE
RETURN

END

Fig. 27 — Sample POLICY subroutine for firing 20-mm projectiles
At ground impact, the integration routine backs up to obtain the
distance between the impact point and the target.* Since IMISS = 1,
the program then continues instead of ending, and the values are re-
stored to those of launch time. At this point the flag ISTORE = 0
indicates the program has been restored. This is used as the criterion
for switching back to CAPFLI so that the program can integrate ahead
0.01 sec before launching again. This process continues until five
projectiles have been launched. After the fifth launching, IMSS = 0,
and the program ends after computing closest miss. Vehicle 3 is a
stationary ground target and therefore receives no velocity or accel-
eration commands. See Appendix D for further instructions on the
calling of individual maneuvers.

Initial Data

Since table values are used for computing the aerodynamics of the
fighter, the F-104 data deck is entered immediately following the JVEH
flag card, as shown in Fig. 11.

Figure 26 is a three-dimensional diagram describing the initial
positions and velocities of the vehicles in this example. The fighter
is located at the origin at an altitude of 7000 ft. It is headed
directly down the y-axis \((v(1,5) = 90\ \text{deg})\) with a velocity of 1000
ft/sec. The target is located on the ground 6000 ft along the y-axis.
Since it is stationary, it has no initial velocity. Other data needed
for this example are the following:

Starting value for integration step size \((DT\theta) = 0.01.\)
Structural lateral acceleration limit of fighter \((ASMAX) = 7.3.\)
Navigation constant for closed-loop guidance routine
\((LAMDAO) = 40.0.\)
Area of fighter \((AREA) = 196.1.\)
Initial weight of fighter \((W0) = 19,470.0.\)

---

*This is not the same as the point of closest approach or miss
distance as usually defined. (See Section XIII.) TACTICS automat-
ically makes the distinction if the altitude of the target \(R(3,3)\) is
exactly zero.
Fig. 28 — Three-dimensional diagram of initial conditions for Example 5
### Table: Vehicle Table Flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal</td>
</tr>
<tr>
<td>1</td>
<td>Signature</td>
</tr>
<tr>
<td>2</td>
<td>Target</td>
</tr>
<tr>
<td>3</td>
<td>Sensor</td>
</tr>
<tr>
<td>4</td>
<td>Communication</td>
</tr>
</tbody>
</table>

### Title Card

```
<table>
<thead>
<tr>
<th>Title Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>
```

### Main Set of Initial Conditions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle #1</td>
<td></td>
</tr>
<tr>
<td>Vehicle #2</td>
<td></td>
</tr>
<tr>
<td>Vehicle #3</td>
<td></td>
</tr>
<tr>
<td>Mass</td>
<td></td>
</tr>
<tr>
<td>Centre of Gravity</td>
<td></td>
</tr>
<tr>
<td>Initial velocity</td>
<td></td>
</tr>
<tr>
<td>Initial position</td>
<td></td>
</tr>
</tbody>
</table>

### Guidance (3 Vehicles)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>r(t1)</td>
<td></td>
</tr>
<tr>
<td>r(t2)</td>
<td></td>
</tr>
<tr>
<td>r(t3)</td>
<td></td>
</tr>
</tbody>
</table>

### Aerodynamics and Propulsion (3 Vehicles)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drag Coefficient</td>
<td></td>
</tr>
<tr>
<td>Thrust</td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
</tr>
</tbody>
</table>

### Military Threat

<table>
<thead>
<tr>
<th>Threat Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Alternate Thrust

<table>
<thead>
<tr>
<th>Thrust Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Specific Impulse

<table>
<thead>
<tr>
<th>Impulse Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Burnout Weight

<table>
<thead>
<tr>
<th>Weight Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Round-Earth Option

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Miscellaneous Options and Extra Inputs

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

1. The following units are used: distance (m), time (sec), velocity (m/sec, Mach number), acceleration (g/s), angles (deg), weight (lb), area (ft²).
2. This card ignored.
3. Enter this section if option not used.
4. Can only read must have a minus sign in column 1.

Fig. 29 — Input data for Example 5
Aiming error in launching projectiles (DVPHI) = 3.4.
Time interval for printing output (DTPO) = 0.5.
Time value at which program is to stop (TETAL) = 20.0.
Missile range to target within which program will automatically initiate process for ground impact miss computation (MINMR) = 1000.0.

Because ballistic constants and variables are defined within the B20M subroutine, values do not have to be read in. The initial position and velocity of the projectiles are set equal to the fighter values.

Figure 29 shows the initial-condition data for this example as entered on the input form. Refer to Section X for instructions on preparing such a data package.
XI. NUMERICAL INTEGRATION

MODES OF OPERATION

TACTICS uses an integration subroutine that is an updated and modified version of Adams-Moulton, Runge-Kutta SHARE subroutine RW INT. In addition to certain other advantages, this updated version is written in FORTRAN IV source language, whereas RW INT is written in a machine-oriented language. An excellent simplified description of the Adams-Moulton predictor-corrector method of integration is given in Ref. 1, and more detailed information is given in Refs. 2 and 3.

The integration subroutine has four optional modes of operation:

- Adams-Moulton predictor-corrector method using a variable step size.
- Runge-Kutta classical fourth-order method using a fixed step size.
- Adams-Moulton predictor-corrector method using a fixed step size.
- Adams-Moulton predictor-corrector method using a variable step size, controlled to allow printout exactly at specified intervals.

The optional modes may be selected by setting the flag JINTEG equal to 0, 1, 2, or 3, respectively, either in initial-condition data (DATA 122)* or in POLICY. If no value is set, JINTEG will automatically be zero and the variable-step-size mode will be used by the program.

Each process has relative advantages. The Adams-Moulton method (JINTEG = 0) generally requires the least execution time, since the step size is automatically controlled (doubled or halved) to maintain a value consistent with a preset truncation error test limit. A possible disadvantage is that substantial overshoots may occur, so that the printout time only approximates the specified time (e.g., 1.22, 2.34, 3.56 sec, instead of 1.0, 2.0, 3.0 sec, respectively). This disadvantage has been removed with only a slight cost in execution.

*See Appendix E.
time by modifying the variable-step mode of operation so that printout will occur as specified by the user. The fixed-step Adams-Moulton and Runge-Kutta methods generally require the most execution time, but there are certain computational advantages in having a predictable step size. Experience with trial problems indicates that the advantages of both variable- and fixed-step methods may be exploited by changing from one to the other during different phases of the flight simulation, particularly just before and after missile launch. This subject will be discussed in further detail in the following subsections.

INTEGRATION ACCURACY

Double precision procedures are used to reduce round-off errors in accumulating variables. Truncation errors are a function of the step size used. A significant feature of the variable-step Adams-Moulton mode of operation is that the step size is controlled by the integration subroutine so that it will be less than a preset truncation error test limit. This limit, designated ERTEST, is the maximum allowable relative error in any one of the dependent variables in a local region, as distinguished from errors accumulating with each step. Practical values for ERTEST range from $10^{-3}$ to $10^{-8}$; this value is automatically set to $10^{-5}$ unless otherwise specified by input data or by POLICY. With reference to the input form of Fig. 9, the value for ERTEST is set by specifying the number of significant digits of accuracy required in data location 123 (for example, if 7 is entered, ERTEST will be set at $10^{-7}$). It should be emphasized that the truncation error test limit is only applicable when either of the two variable-step modes of operation is used (JINTEG = 0 or JINTEG = 3).

FIXED-STEP APPROACH TO LAUNCH

As previously indicated, the use of the variable-step options will generally reduce execution time, which will result in an appropriate step size for a specified truncation error. Accordingly, these modes should be used whenever the problem permits—e.g., in time-consuming aerial acrobatics. However, a disadvantage exists due to the inability
to predict which step will be used at a particular phase of the problem, particularly when missile-launch criteria are about to be satisfied. A problem arises because the step may be large, allowing an undesirable overshoot of the criteria (similar to the inexactitude of printout in the JINTEG = 0 mode of operation). A remedy has been provided which automatically causes launch criteria (time, range to target, etc.) to be approached with a fixed minimum step size. Actually, the procedure is to approach launch on a variable (or fixed) step and, when overshoot occurs, back up to the previous time step, restore all conditions, and approach again using small fixed steps. Moreover, these fixed integration steps will be maintained during the missile flight time until the point of closest approach to the target or miss distance has been determined. The previous integration mode of operation (whether fixed or variable) will then be restored unless the option is used to terminate the problem run. If the mode of operation was fixed-step in the first place, back-up and fixed-step approaches to launch would not occur.

**MISS-DISTANCE COMPUTATION**

The point of closest approach is taken to be the value of the relative missile range to target when its time derivative equals zero. That is, when approaching the target the range rate will be negative, when closest to the target it will be zero, and after passing the target it will be positive. This criterion is used to determine miss distance by "backing up" the integration process in a manner similar to that described for fixed-step approach to launch. The procedure is to return to the previous time step whenever the missile-to-target range rate becomes positive, restore all conditions, shrink the time step by a factor of ten, and repeat this process until the time step becomes smaller than DTMIN (DATA 134). If this value is not read in by the user, it is automatically set to $10^{-5}$. In order to allow positive range rates to occur without initiating the process for miss-distance computation, an additional condition is imposed: The missile range to target must be less than MLNR (DATA 124). Otherwise, miss distance will not be calculated. For ASM or SSM applications it is usually necessary
to determine the ground range to target at ground impact (i.e., zero missile altitude), which (as mentioned earlier) is generally not the same as the point of closest approach. TACTICS will automatically compute this ground range by using the altitude of vehicle 2 rather than range rate as the criterion for backing up to determine the time and range at ground impact. This feature is triggered if the altitude of vehicle 3 is exactly 0.0.

INITIALIZATION

The Adams-Moulton predictor-corrector modes of integration, both fixed- and variable-step, use previous values of the variables to predict and to correct values for each new step. In other words, past history is used to predict the future. This procedure is not appropriate in many cases because of discontinuities or large step changes (e.g., a large instantaneous reversal in acceleration from right to left turn, or from climb to dive). Large step changes can not only cause a significant waste of execution time (particularly when the variable-step modes are used) but can also cause important errors. This possibility may be eliminated by reinitializing the integration process whenever interfaces occur, e.g., changes in POLICY or propulsion. Initializing may be thought of as simply stopping and starting over again, not using previous values for prediction of new values. An automatic reinitializing feature has been built into the framework of the POLICY subroutine using the integer variables JPOL, KPOL, LPOL, MPOL, and NPOL. A change in any one of these will trigger this process. Accordingly, in formulating a POLICY subroutine, the user should be careful to separate drastic changes in acceleration, e.g., thrust, boost, or maneuver changes, by using these FORTRAN variables. (See the examples in Section XII.)
Appendix A

AIRCRAFT ATTITUDE AND ORIENTATION ANGLES

The relationships given in this appendix, contained in subroutine ATITUD, are applicable to conventional airframes where coordinated turns are assumed, as defined in Section IV (see Fig. 7). Thus the use of this version of ATITUD is optional, depending on a particular vehicle's airframe characteristics. For example, if a cruciform missile airframe is to be represented, the attitude-angle relationships corresponding to those in this appendix may be included in the missile control-law routine without calling upon ATITUD, or a different version of the routine may be formulated.

A right-hand set of orthogonal roll, pitch, and yaw orientation axes is assumed as shown below,

\[ \overline{I}_T \quad \overline{I}_Y \quad \overline{I}_P \]

where \( \overline{I}_P \) is oriented in the direction of the right wing, \( \overline{I}_T \) is oriented in the direction of the thrust vector, and \( \overline{I}_Y \) forms a right-hand orthogonal system with \( \overline{I}_P \) and \( \overline{I}_T \) (approximately in the opposite direction of the lift vector \( \overline{L} \), i.e., neglecting the angle of attack \( \alpha \)).

**YAW AXIS**

With the assumption that the vehicle velocity vector is in the plane formed by \( \overline{I}_T \), \( \overline{I}_Y \) vectors (i.e., no sideslip angle), the unit vector \( \overline{I}_Y \) may be determined by examination of the sketch on the following page:
\[ I_y = I_v \sin \alpha - I_L \cos \alpha \]

where, from Fig. 8,

\[ I_L = \frac{W}{g} \left\{ a_0 A_i + (a_{VY} + g \cos \gamma) \frac{I_D}{P_n} \right\} \quad (45) \]

**ROLL AXIS**

Similarly, the roll axis \( I_T \) is defined by

\[ I_T = I_v \cos \alpha + I_L \sin \alpha \quad (46) \]

**PITCH AXIS**

With \( I_y \) and \( I_T \) determined, the unit vector \( I_p \) may be readily found from the right-hand orthogonal relationship

\[ I_p = I_y \times I_T \quad (47) \]

**AZIMUTH AND ELEVATION ANGLES**

The unit vectors \( I_p, I_y, \) and \( I_T \) define the attitude or orientation of a vehicle. Now we may conveniently perform a coordinate transformation to resolve the relative range vector \( \bar{I}_{ij} \) \((i = 1, 2, 3; j = 1, 2, 3)\) into this frame of reference. The primary objective is to determine
the orientation of \( \overline{r}(i,j) \), the LOS, in terms of an azimuth angle \( \eta \) and an elevation angle \( \varepsilon \), defined as shown in Fig. 30.

\[ \overline{r} = \overline{r}_{ij} / r_{ij} \]

Fig. 30—Azimuth and elevation coordination transformation

The negative sign for \( \varepsilon \) is an unfortunate result of the conventional aerodynamic orientation of the pitch, roll, and yaw vectors. (We wish \( +\varepsilon \) to correspond to the up direction.)

The relative range vector \( \overline{r}_{ij} \) may be transformed into this system by the following equation:

\[ \overline{r}_{ij} = (\overline{r}_{ij} \cdot \overline{I}_T)\overline{I}_T + (\overline{r}_{ij} \cdot \overline{I}_p)\overline{I}_p + (\overline{r}_{ij} \cdot \overline{I}_y)\overline{I}_y \]  \hspace{1cm} (48)

where

\[ \overline{r}_{ij} \cdot \overline{I}_T = r_{ij} \cos \varepsilon \cos \eta \]
\[ \overline{r}_{ij} \cdot \overline{I}_p = r_{ij} \cos \varepsilon \sin \eta \]  \hspace{1cm} (49)
\[ \overline{r}_{ij} \cdot \overline{I}_y = r_{ij} \sin \varepsilon \]

From the above, we may conclude that
\[ n = \tan^{-1} \frac{\vec{I}_r \cdot \vec{I}_p}{\vec{I}_r \cdot \vec{I}_T} \]  

where

\[ \vec{I}_r = \frac{\vec{r}_{ij}}{r_{ij}} \]

**BANK ANGLE**

The bank angle \( \psi_B \) is defined with respect to a wind axis or velocity-vector reference system (see Figs. 7 and 8 and Eq. (19)). TACTICS calculates the angle from the expression

\[ \psi_B = \tan^{-1} \frac{\vec{I}_L \cdot \vec{I}_D}{\vec{I}_L \cdot \vec{I}_A} \]  

where \( \vec{I}_L \) is the unit vector along the lift vector \( \vec{L} \) (normal to \( \vec{V} \)).

**ROLL ANGLE**

Roll angle \( \psi \) (ROLL(I)) is arbitrarily defined as the angle existing between the pitch axis, i.e., the wings of the vehicle, and a line through the c.g. of the vehicle both normal to the longitudinal (i.e., roll) axis and parallel to the horizontal plane. The determination is accomplished as follows. First, the spherical coordinates \( \theta_T \) and \( \varphi_T \) of the unit vector \( \vec{I}_T \) are computed:

\[ \theta_T = \tan^{-1} \frac{1_{Ty}}{1_{Tx}} \]

\[ \varphi_T = \sin^{-1} (1_{Tz}) \]  

(52)
Next, the unit vectors \( \overrightarrow{1_{Al}} \) and \( \overrightarrow{1_{Dl}} \) analogous to the vectors \( \overrightarrow{1_A} \) and \( \overrightarrow{1_D} \) are determined. That is, \( \overrightarrow{1_{Al}} \) is normal to \( \overrightarrow{1_T} \) and in the horizontal plane, and \( \overrightarrow{1_{Dl}} \) is normal to \( \overrightarrow{1_T} \) and in a vertical plane so as to form a right-handed orthogonal system. The components of these vectors are

\[
\begin{align*}
1_{Ax} &= -\sin \theta_T & 1_{Dlx} &= -\sin \varphi_T \cos \theta_T \\
1_{Ay} &= \cos \theta_T & 1_{Dly} &= -\sin \varphi_T \sin \theta_T \\
1_{Az} &= 0 & 1_{Dlz} &= \cos \varphi_T
\end{align*}
\]

(53)

The roll angle \( \psi \) may now be computed from the geometry shown in Fig. 31 as viewed from the tip of unit vector \( \overrightarrow{1_T} \) (directed out of the page).

\[\text{Fig. 31—Roll-angle geometry}\]
\[ \psi = \tan^{-1} \frac{\bar{I}_P \cdot \bar{I}_D}{\bar{I}_Y \cdot \bar{I}_D} \]  

(54)

As defined above, a turn to the right results in a positive roll angle varying from 0 to 180 deg, and a turn to the left results in a negative roll angle varying from 0 to 180 deg.
Appendix B

INTEGRATING THE EQUATIONS OF MOTION

In representing the flight of three vehicles in motion simultaneously, there are 18 differential equations to be numerically integrated, nine involving the accelerations \( \vec{\ddot{V}}(i) \) and the other nine involving the velocities \( \vec{\dot{V}}(i) \). These equations might be expressed in a variety of different forms, each having certain advantages in certain situations. The TACTICS program expresses the differential equations in two different forms: One involves the flat-earth representation (less complex and faster), and the other involves a round rotating or nonrotating earth (essential for space applications, long ranges, or high speeds). Unless specified by the KINTEG option, DATA(110), the flat-earth equations, are used. In order to decrease execution time, only 12 equations are integrated when vehicle 2 is in captive flight (subroutine CAPFLT).

FLAT-EARTH CARTESIAN FORM

The accelerations are determined by resolving the applied forces (e.g., aerodynamic, gravitational, or propulsive) into three components. Two of these components, \( a_{oh} \) and \( a_{ov} \), are normal to the velocity vector \( \vec{V} \), and their vector sum is the net lateral acceleration of the vehicle. The direction of the third component, \( \vec{V} \), is parallel to or along the velocity vector \( \vec{V} \). When integration is to occur, the three components of acceleration \( a_{oh} \), \( a_{ov} \), and \( \vec{\dot{V}} \), as well as the unit vectors \( \vec{I}_A \), \( \vec{I}_D \), and \( \vec{I}_V \), have been determined, so that the expression for the net acceleration vector \( \vec{a}_o \) is

\[
\vec{a}_o = a_{oh} \vec{I}_A + a_{ov} \vec{I}_D + \vec{\dot{V}} \vec{I}_V \quad (55)
\]

Converting to Cartesian coordinates yields the expression

\[
\ddot{x} = a_{oh} \dot{I}_{Ax} + a_{ov} \dot{I}_{Dx} + \dot{\vec{V}} \dot{I}_{Vx} \quad (56)
\]

and similarly for \( \ddot{y}, \ddot{z} \). (See Eqs. (9).)
Integration of the equations for \( \vec{a}_1 \), \( \vec{a}_2 \), and \( \vec{a}_3 \) corresponding to each of the three vehicles will result in \( \vec{v}_1 \), \( \vec{v}_2 \), and \( \vec{v}_3 \), respectively. At this point, the use of relative velocities and positions is introduced to minimize errors in relative position due to differencing large numbers. Accordingly, the velocities of vehicles 1 and 2 relative to vehicle 3 are formed and integrated to obtain the relative positions. The equations are as follows:

\[
\vec{v}_{13} = \vec{v}_3 - \vec{v}_1 \\
\vec{v}_{23} = \vec{v}_3 - \vec{v}_2
\]  

Integration of the above equations yields the relative position vectors \( \vec{r}_{13} \) and \( \vec{r}_{23} \). The velocity of vehicle 3, \( \vec{v}_3 \) (i.e., an absolute velocity relative to the fixed origin of the x, y, z frame), is integrated to yield the absolute position vector \( \vec{r}_3 \). The absolute positions \( \vec{r}_1 \) and \( \vec{r}_3 \) are subsequently determined from

\[
\vec{r}_1 = \vec{r}_3 - \vec{r}_{13} \\
\vec{r}_2 = \vec{r}_3 - \vec{r}_{23}
\]  

This procedure emphasizes the accuracy of the relative rather than absolute position geometry.

**ROUND- EARTH SPHERICAL FORM**

The dynamic equations for a particle whose motion is observed from a rotating reference frame involve centrifugal and Coriolis-force terms. Moreover, since the earth is represented as spherical, a number of coordinate transformations are necessary to relate the various frames of reference. For convenience, the basic equations of motion and certain unit vector notations have been extracted from the ROCKET program\(^1\) in order to provide compatibility. However, since TACTICS is primarily concerned with the relative motion of possibly three
different vehicles, there are significant differences between coordinate systems and computational methods employed.

The basic differential equations of motion are derived using three unit vectors defined as follows:

- \( \overrightarrow{R} \), directed radially from the earth's center to the vehicle, a point mass.
- \( \overrightarrow{L} \), normal to \( \overrightarrow{R} \) and directed eastward along a parallel of latitude.
- \( \overrightarrow{P} \), directed northward along the local meridian so as to form a right-handed orthogonal system with \( \overrightarrow{R} \) and \( \overrightarrow{L} \).

The acceleration of the vehicle with respect to a nonrotating inertial \( X, Y, Z \) coordinate system is

\[
\overrightarrow{a} = \ddot{r} = \dddot{X} \overrightarrow{i} + \dddot{Y} \overrightarrow{j} + \dddot{Z} \overrightarrow{k}
\]  

(59)

This acceleration \( \overrightarrow{a} \) corresponds to a net result of all forces applied, \( F/m \). The following derivation develops expressions for the net acceleration resulting from \( F \) in terms of a rotating-earth fixed-coordinate system, i.e., longitude \( \lambda \), latitude \( \varphi \), radial distance \( r \), and time derivatives thereof.

First, the following simple device may be used to transform coordinates to the \( \overrightarrow{R}, \overrightarrow{L}, \overrightarrow{P} \) system:

\[
\overrightarrow{a} \cdot \overrightarrow{R} = (\overrightarrow{R} \cdot \overrightarrow{i}) \dddot{X} + (\overrightarrow{R} \cdot \overrightarrow{j}) \dddot{Y} + (\overrightarrow{R} \cdot \overrightarrow{k}) \dddot{Z}
\]  

(60)

and similarly for \( \overrightarrow{a} \cdot \overrightarrow{L} \) and \( \overrightarrow{a} \cdot \overrightarrow{P} \).

Referring to Fig. 32, the required dot products are by inspection

\[
\begin{align*}
\overrightarrow{R} \cdot \overrightarrow{i} &= \cos \varphi \cos \theta \\
\overrightarrow{L} \cdot \overrightarrow{i} &= -\sin \theta \\
\overrightarrow{P} \cdot \overrightarrow{i} &= -\sin \varphi \cos \theta \\
\overrightarrow{R} \cdot \overrightarrow{j} &= \cos \varphi \sin \theta \\
\overrightarrow{L} \cdot \overrightarrow{j} &= \cos \theta \\
\overrightarrow{P} \cdot \overrightarrow{j} &= -\sin \varphi \sin \theta \\
\overrightarrow{R} \cdot \overrightarrow{k} &= \sin \varphi \\
\overrightarrow{L} \cdot \overrightarrow{k} &= 0 \\
\overrightarrow{P} \cdot \overrightarrow{k} &= \cos \varphi
\end{align*}
\]  

(61)
Fig. 32 — Geocentric coordinate system
(Note the analogy with the definition of unit vectors $\overline{I}_V$, $\overline{I}_A$, and $\overline{I}_D$; see Eq. (6).)

The desired transformation may now be accomplished using

$$\bar{a} = (a \cdot \overline{I}_R) \overline{I}_R + (a \cdot \overline{I}_L) \overline{I}_L + (a \cdot \overline{I}_p) \overline{I}_p \tag{62}$$

Expanding and substituting the dot products results in

$$\bar{a} = \overline{I}_R (\dddot{x} \cos \varphi \cos \theta + \dddot{y} \cos \varphi \sin \theta + \dddot{z} \sin \varphi)$$

$$+ \overline{I}_L (- \dddot{x} \sin \theta + \dddot{y} \cos \theta + 0)$$

$$+ \overline{I}_p (- \dddot{x} \sin \varphi \cos \theta - \dddot{y} \sin \varphi \sin \theta + \dddot{z} \cos \varphi) \tag{63}$$

The inertial coordinates $X$, $Y$, $Z$ may be expressed in terms of spherical coordinates as

$$X = r \cos \varphi \cos \theta$$

$$Y = r \cos \varphi \sin \theta$$

$$Z = r \sin \varphi \tag{64}$$

Differentiating these expressions twice with respect to time and substituting in Eq. (63) will yield

$$\dddot{a} = \overline{I}_R (\dddot{r} - \dot{r} \dot{\theta}^2 + r \theta^2 \cos^2 \varphi)$$

$$+ \overline{I}_L (\dddot{\theta} \cos \varphi + 2 \dot{r} \dot{\theta} \cos \varphi - 2 \dot{r} \dot{\varphi} \sin \varphi)$$

$$+ \overline{I}_p (\dddot{\varphi} r + r \dot{\theta}^2 \sin \varphi \cos \varphi + 2 \dot{r} \dot{\varphi}) \tag{65}$$

Referring to Fig. 32, we note that $\theta$ is a time variable:

$$\theta = \lambda + \omega_\perp t$$
where
\[ \lambda = \text{an arbitrarily defined angle of longitude on the earth's surface} \]
\[ \omega_e = \text{the earth's angular rate of rotation (} 7.29211585 \cdot 10^{-5} \text{ rad/sec)} \]
\[ t = \text{the time interval of rotation} \]

Accordingly,
\[ \dot{\theta} = \lambda + \omega_e \] (66)

and
\[ \ddot{\theta} = \lambda \] (67)

The resultant force \( \overline{F} \), causing acceleration \( \overline{a} \) with respect to the inertial \( X, Y, Z \) frame, may be resolved into components \( F_R, F_L, F_P \) in the \( \overline{I}_R, \overline{I}_L, \overline{I}_P \) directions, respectively. Substituting \( \overline{a} = \overline{F}/m \) in the preceding equations and equating components results in the desired set of equations of motion:

\[ \ddot{r} = \frac{F_R}{m} + r \dot{\varphi}^2 + r (\dot{\lambda} + \omega)^2 \cos^2 \varphi \]
\[ \ddot{\lambda} = \left[ \frac{F_L}{m} - 2 \dot{\tau} (\dot{\lambda} + \omega) \cos \varphi + 2 \dot{r} \dot{\varphi} (\dot{\lambda} + \omega) \sin \varphi \right] / r \cos \varphi \]
\[ \ddot{\varphi} = \left[ \frac{F_P}{m} - r (\dot{\lambda} + \omega)^2 \sin \varphi \cos \varphi - 2 \dot{r} \dot{\varphi} \right] / r \] (68)

These equations are a set of second-order differential equations in terms of geocentric radial distance \( r \), longitude \( \lambda \), latitude \( \varphi \), and components of resultant force \( \overline{F} \). (The methodology of force definition is discussed in Section IV.) Numerical integration of these second-order equations will yield the first-order velocity components \( \dot{r}, \dot{\lambda}, \) and \( \dot{\varphi} \) at a time \( t + \Delta t \). Theoretically, a second integration will yield the position quantities \( r, \lambda, \) and \( \varphi \), but we wish to avoid this procedure to minimize numerical errors arising from differencing large numbers associated with the geocentric distance \( r \). Accordingly, position components are computed by the following method:
Integrating the second-order equations to determine \( r \), \( \lambda \), and \( \varphi \).

Forming the topocentric velocities \( \vec{\gamma}^{\dagger}\), i.e., the velocities with respect to an arbitrarily chosen reference point (or coordinate system origin) on the earth's surface.

Integrating the topocentric velocities \( \vec{\gamma}^{\dagger}\) to determine the topocentric range position vectors \( \vec{\rho}^{\dagger}\).

The topocentric reference point or local coordinate system origin \( \vec{R}_o \) is defined by input data in terms of latitude \( \varphi_o \) and longitude \( \lambda_o \) so that

\[
\begin{align*}
R_{ox} & = R_o \cos \varphi_o \cos (\lambda_o + \omega_e t) \\
R_{oy} & = R_o \cos \varphi_o \sin (\lambda_o + \omega_e t) \\
R_{oz} & = R_o \sin \varphi_o
\end{align*}
\]

\( R_o \) is taken to be \( 2.0925861 \times 10^7 \) ft.

The velocity \( \vec{R}_o \) with respect to an inertial frame is

\[
\vec{\dot{R}}_o = -R_{oy} \omega_e \vec{i} + R_{ox} \omega_e \vec{j}
\]

Note that in the second-order differential equations the centrifugal and Coriolis accelerations are a function of latitude angle \( \varphi \); hence, numerical solutions will be affected by assumed initial-condition values for vehicle latitude. On the other hand, initial assumptions for vehicle longitude position may be purely arbitrary. Figure 33 shows the topocentric nonrotating coordinate system with the \( X_I \), \( Y_I \), \( Z_I \) axis always remaining parallel to the \( X, Y, Z \) inertial axis. With the \( \vec{R}_o \) origin established, the topocentric range vectors (from the topocentric origin to the vehicles) are

\[
\vec{\rho}_i = \vec{r}_i - \vec{R}_o \quad i = 1, 2, 3
\]

and the velocities are
Fig. 33 — Topocentric coordinate system
\[
\ddot{r}_i = \ddot{r}_i - \ddot{R}_o
\]

(72)

\[
\frac{d\vec{r}}{dt} = \dot{r}_i + \omega_e \times r
\]

local

where

\[
\frac{d\vec{r}}{dt} = \dot{r} \vec{I}_R + r \dot{\lambda} \cos \varphi \vec{I}_L + r \varphi \vec{I}_P
\]

(73)

and

\[
\omega_e \times \vec{r} = r \omega_e \cos \varphi \vec{I}_L
\]

(74)

so that

\[
\dot{r}_i = \dot{r}_R + r(\dot{\lambda} + \omega_e) \cos \varphi \vec{I}_L + r \varphi \vec{I}_P
\]

(75)

The integration of the topocentric velocities \( \dot{r}_i \) will yield the positions \( \vec{s}_i \) in terms of the inertial coordinates \( X_T, Y_T, Z_T \). Although this coordinate system is convenient for astronomical or space applications, it is not desirable for many other problems in which the horizontal plane and the local vertical are more significant. Referring back to the flat-earth \( x, y, z \) system, we would like the \( z \)-component to correspond to altitude and the \( z-y \) plane to correspond approximately to the horizontal plane. Accordingly, all \( \vec{s}_i \) coordinates are transformed into an azimuth-elevation system with the origin at the \( \vec{R}_o \) position, the \( z \)-axis in the direction of \( \vec{R}_o \), and the \( y \)-axis directed eastward, as shown in Fig. 34. Subroutine T0P0CN is used for transforming from either equatorial plane to azimuth-elevation coordinates or vice versa.

The coordinate transformations are as follows. The transformation from topocentric \( X_T, Y_T, Z_T \) coordinates to azimuth-elevation \( x, y, z \) coordinates is given by the following matrix:
Fig. 34 — Azimuth-elevation topocentric coordinate system

- $x_s$ - South
- $y_E$ - East
- $z_h$ - Local vertical

- $x$ - Parallel to $x_s$
- $y$ - Parallel to $y_E$
- $z$ - Parallel to $z_h$
\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} =
\begin{bmatrix}
  \sin \phi_T \cos \theta_T & \sin \phi_T \sin \theta_T & -\cos \phi_T \\
  -\sin \theta_T & \cos \theta_T & 0 \\
  \cos \theta_T \cos \phi_T & \cos \phi_T \sin \theta_T & \sin \phi_T
\end{bmatrix}
\begin{bmatrix}
  X_T \\
  Y_T \\
  Z_T
\end{bmatrix}
\]

(76)

The reverse transformation, from azimuth-elevation \(x, y, z\) coordinates to topocentric \(X_T, Y_T, Z_T\) coordinates, is given by

\[
\begin{bmatrix}
  X_T \\
  Y_T \\
  Z_T
\end{bmatrix} =
\begin{bmatrix}
  \sin \phi_T \cos \theta_T & -\sin \theta_T & \cos \theta_T \cos \phi_T \\
  \sin \phi_T \sin \theta_T & \cos \theta_T & \sin \theta_T \cos \phi_T \\
  -\cos \phi_T & 0 & \sin \phi_T
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
\]

(77)

where \(\phi_T\) and \(\theta_T\) are the latitude and longitude, respectively, of the rotating reference origin at point \(T\), defined by

\[
\phi_T = \phi(t_0) \\
\theta_T = \lambda(t_0) + \omega(t - t_0)
\]

(78)

In addition to the "standard" printout in terms of \(x, y, z\) azimuth-elevation (horizontal-plane) system coordinates, the quantities of latitude \(\phi\), longitude \(\lambda\), and altitude \(h\) for each vehicle are available as an optional printout section. Note that because of the earth curvature and the use of a fixed-earth reference point \(\vec{R}_o\), the magnitude of the \(z\)-position component will not generally correspond exactly to altitude (because of the \(x-y\) displacement from the origin).
Appendix C

GUIDANCE-AND-CONTROL-LAW DEFINITIONS

This appendix contains descriptions and mathematical definitions of most of the significant guidance and control laws developed for use with TACTICS. The list is open-ended and incomplete, since new ideas are continuously being implemented as new subroutines. See Section V for a summary of definitions and the applicable acceleration equations.

OPEN-LOOP CONTROL LAWS

Straight Flight

The commanded lateral acceleration $\bar{a}_C$ is zero. The vehicle will fly a straight-line path (but not necessarily "straight and level"). However, an acceleration or deceleration along this path may occur due to the thrust-drag relationship.

Straight and Level Flight

This routine is used to return a vehicle to level flight in the horizontal plane from a diving or climbing condition. An arbitrary control-law assumption is made that the vehicle will commence the return to level flight at maximum $g$ capability subject to structural or $C_{L_{max}}$ constraints. The magnitude of this acceleration tapers off linearly as the velocity vector reaches an angle of 10 deg from the horizontal plane and becomes zero within 0.1 deg of level flight. The direction of the acceleration is accordingly

$$\bar{I}_1 = \pm \bar{I}_D$$

(positive for climbing and negative for diving). The magnitude of the acceleration follows the arbitrary law
\[ a_C = \min (a_{\text{max}}; |(\gamma/10)a_{\text{max}}|) \]  
\[ a_C = 0 \quad \gamma < 0.1 \]  

**Captive Flight**

This routine is used to zero out computations and printed values for vehicles which are in captive flight. There are three modes:

- Vehicle 2 locked to vehicle 1.
- Vehicle 1, 2, or 3 locked to the zero origin (i.e., all values for position, velocity, and acceleration are zero).
- Vehicle 2 locked to vehicle 3.

Captive flight is not a guidance law in the sense of the preceding discussion but rather a device to eliminate unnecessary computations and improve the appearance of the printed values.

**Launch**

This control law, which simulates the launch-boost phase of a missile flight, may be applied to any of the three vehicles. The call for "launch" is usually based on some criteria stated in POLICY (e.g., range, range rate, geometry, accelerations, time, and—most importantly—combinations thereof). When this routine is called, the boost velocity \( \Delta V \) must be specified as a constant. The commanded lateral acceleration is gravitational only:

\[ \bar{a}_C = -g \cos \gamma \bar{T}_D \]  

(81)

**Aim (1)**

This routine was provided for applications in which it is necessary to simulate the aiming process, e.g., SAM launch. It involves the following problem: Given a target velocity vector \( \bar{V}_T \) and a vehicle speed \( V_M \) (both assumed to remain constant) separated by a range vector \( \bar{R}_{ij} \), find the orientation of \( \bar{V}_M \), i.e., aiming angles, so that an
intercept will occur. Like captive flight, it is a device for defining angles rather than a law for defining $\vec{a}_c$. The solution is outlined below:

\[
\overrightarrow{r}_{ij} + \overrightarrow{v}_T \cdot T = \overrightarrow{v}_M \cdot T
\]  

(82)

where $T$ is the "time-to-go" until impact. The magnitude of $\overrightarrow{v}_M$ is known, but the components $V_{Mx}$, $V_{My}$, and $V_{Mz}$ are to be determined. From the above equation,

\[
|\overrightarrow{r}_{ij} + \overrightarrow{v}_T \cdot T| = \left( r_{ij}^2 + 2\overrightarrow{r}_{ij} \cdot \overrightarrow{v}_T + v_T^2 \right)^{\frac{3}{2}}
\]

\[
T = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad T > 0
\]

(83)

where $a$, $b$, and $c$ are the corresponding coefficients in the quadratic equation. Hence,

\[
\overrightarrow{v}_M = \frac{\overrightarrow{r}_{ij}}{T} + \overrightarrow{v}_T
\]

\[
V_{Mx} = \frac{r_{ij} x}{T} + v_{Tx}
\]

(84)

and similarly for $V_{My}$ and $V_{Mz}$.
The aiming angles \( \Theta_M \) and \( \phi_M \) may be determined from

\[
\Theta_M = \tan^{-1} \left( \frac{V_{My}}{V_{Mz}} \right) \tag{85}
\]

\[
\phi_M = \sin^{-1} \left( \frac{V_{Mz}}{V_M} \right)
\]

Aim (2)

This routine is similar in purpose to the previous item. In certain problems it is necessary to simulate the launching of a missile (or firing of a projectile) along the LOS to the target. The aiming process is more complicated if the launching platform motion is taken into account. In order for the projectile to travel along the LOS (ignoring ballistic drop), the orientation angles of the resultant projectile velocity and platform velocity must be the same as those of the LOS. Hence, if \( \vec{V}_2 \) is the projectile velocity and \( \vec{V}_1 \) is the launching platform velocity, \( \vec{V}_{12} = \vec{V}_2 + \vec{V}_1 \), where \( \vec{V}_{12} \) is the resultant velocity from which

\[
\vec{V}_2 = \vec{V}_{12} - \vec{V}_1 \tag{86}
\]

\[
V^2_2 = V^2_{12} - 2 \vec{V}_1 \cdot \vec{V}_{12} - V^2_1
\]

The magnitude of \( V_{12} \) may now be determined from

\[
V^2_{12} - 2V_1V_{12} \cos(\Theta_{LOS} - \Theta_{V1}) \cos(\phi_{LOS} - \phi_{V1}) - V^2_1 - V^2_2 = 0 \tag{87}
\]

Knowing the magnitude and orientation of \( \vec{V}_{12} \),

\[
V_{2x} = V_{12x} - V_{1x}
\]

\[
V_{2y} = V_{12y} - V_{1y} \tag{88}
\]

\[
V_{2z} = V_{12z} - V_{1z}
\]
The orientation angles of $\overline{V}_2$ are

$$\gamma_{V2} = \frac{(V_{12} \sin q_{\text{LOS}} - V_{1x})}{V_2}$$

$$\theta_{V2} = \frac{(V_{12} \cos q_{\text{LOS}} \sin \theta_{\text{LOS}} - V_{2y})}{(V_{12} \cos q_{\text{LOS}} \cos \theta_{\text{LOS}} - V_{2x})}$$

and the magnitude is the boost velocity

$$|\overline{V}_2| = \Delta \nu$$

The incremental angles which should be added to the launching platform angles for aiming are then

$$\Delta \gamma_{V2} = \gamma_{V2} - \gamma_{V1}$$

$$\Delta \theta_{V2} = \theta_{V2} - \theta_{V1}$$

**Left or Right Turn (1)**

The commanded lateral acceleration vector $\overline{a}_C$ is in the horizontal plane and has a constant value as specified when calling the routine(s). The two routines (left and right) are identical except for an algebraic sign corresponding to the direction of the turn ($\pm \overline{1}_A$). It was decided to specify the magnitude of the turning acceleration in terms of the resultant normal acceleration, expressed in number of g's or $F_n/W$ (see Eq. (21)). Accordingly,

$$\overline{a}_C = \left(\frac{g}{W}\right) \sqrt{F_n^2 - (W \cos \gamma)^2} (\pm \overline{1}_A)$$

**Left or Right Turn (2)**

The commanded lateral acceleration vector $\overline{a}_C$ is in the horizontal plane. The magnitude of $\overline{a}_C$ depends on the excess thrust available to perform a turn with constant Mach number. In other words, these
maneuvers are constant-Mach-number horizontal turns; a necessary aero-
dynamic condition for initiating such a turn is that thrust must exceed
drag, because additional drag will be induced by the maneuver itself.
These routines require iterative processes if aerodynamic tables are
involved, since the tables must be read in reverse (see item 2 below).
A brief outline of the problem is as follows:

1. Since Mach number is constant with altitude, \( \dot{M} = \dot{V} = 0 \), and
the necessary aerodynamic relationship is

\[
D + W \sin \gamma + T \cos \alpha = 0 \tag{92}
\]

from which \( D \) may be calculated. Knowing \( D \),

\[
C_D = \frac{D}{A q} \tag{93}
\]

2. Find \( C_L \) and \( \alpha \) corresponding to \( C_D \) and Mach number. For example,
using the familiar analytic expressions

\[
C_D = C_D + \frac{dC_D}{d(C_L^2)} C_L^2 \tag{94}
\]

\[
C_L = \sqrt{(C_D - C_D) / d(C_L^2)} \frac{dC_D}{d(C_L^2)} \tag{94}
\]

\[
\alpha = \frac{C_L}{(dC_L / d\alpha) + \alpha_o} \tag{94}
\]

\[
L = C_L A q \tag{94}
\]

\[
F_n = L + T \sin \alpha \tag{94}
\]

The equivalent of the preceding operations must be performed if tabular
values are used. In effect, values for \( dC_L / d\alpha \) and \( dC_D / d(C_L^2) \) must be
numerically determined.
3. Find the magnitude and direction of \( \overline{a}_C \) corresponding to the available force \( \overline{F}_n \).

\[
\overline{a}_C = \left( \frac{g}{\omega} \right) \sqrt{\frac{F_n^2}{n} - (u \cos \gamma)^2} = \left( \pm \left( \frac{n}{A} \right) \right) \quad (95)
\]

where (+) is used for a left turn and (-) for a right turn. Note that the magnitude of \( \overline{a}_C \) is not explicitly specified but is dependent on \( F_n \) and \( \gamma \), i.e., it is not known a priori.

**Left or Right Turn (3)**

These maneuvers are similar to the Left or Right Turn (2) maneuvers described above except for the constant-altitude requirement. By means of a dive, gravity may be utilized to hold Mach number constant during the turn. If there is insufficient thrust to meet the conditions for constant Mach number and a specified number of g's in the turn, the vehicle will dive in order to satisfy these conditions. On the other hand, if excess thrust is available (or after a sustained dive), the vehicle will climb. If a maximum lift coefficient \( (C_{L_{max}}) \) limit is reached, the vehicle will turn at the corresponding allowable acceleration for \( C_{L_{max}} \), and Mach number will equal a constant. Since Mach number varies with altitude, the following relationship must be satisfied:

\[
\dot{M} = \frac{\dot{V}}{s} - \frac{V \dot{s}}{s^2} = 0 \quad (96)
\]

Referring to the relationships given in Appendix I and performing the necessary arithmetical operations,

\[
\dot{\phi} = -6.9697419 \times 10^{-8} (1 - 6.8865741)^{3.256} \quad (97)
\]

\[
\dot{\phi} = -32.207899 \rho \dot{z} \quad (97)
\]

\[
\dot{s} = 0.5929 \left( \frac{\dot{\rho}}{\sqrt{\rho \rho}} - \frac{\sqrt{\rho \rho}}{\rho^{3/2}} \right)
\]
From the preceding expression for \( M = 0 \),

\[
\dot{V} - \frac{V_s}{s} = 0
\]  

(98)

The necessary aerodynamic relationship is

\[
D + W \sin \gamma + T \cos \alpha - \frac{V_s}{s} \left( \frac{W}{g} \right) = 0
\]  

(99)

from which the drag \( D \) may be determined by means of the applicable expressions as given by Eq. (26). The components of commanded acceleration are

\[
as_{Ch} = \pm \left( \frac{g}{W} \right) \min (F'_n, F'_n) \]

\[
as_{CV} = \sqrt{(g/W) F'_n - a_{Ch}^2 - g \cos \gamma}
\]  

(100)

where \( F'_n \) (or rather \( F'_n/W \) in terms of number of \( g \)'s) is to be specified when the maneuver is called and \( F'_n \) corresponds to the \( L \), \( C_D \), \( C_L \), and \( \alpha \) relationships of Eq. (27). The component \( a_{Ch} \) is positive for left turns and negative for right turns. Special cases arise if the specified acceleration \( F'_n/W \) exceeds structural or aerodynamic constraints. If \( F'_n \) should exceed the structural constraints, the value is altered to correspond to the maximum value \( a_{Smax} \). If \( F'_n \) exceeds the aerodynamic value

\[
F_{nmax} = C_{Lmax} Aq + T \sin \alpha_{max}
\]  

(101)

then \( F'_n \) is altered to correspond to \( F_{nmax} \). Since \( C_{Lmax} \) varies with Mach number and by definition the Mach number should not change, \( F'_n \) will remain equal to \( F_{nmax} \) under this constraint.

**Left or Right Turn (4)**

These maneuvers are identical to the Left or Right Turn (3) maneuvers except under the \( F_{nmax} \) constraint discussed above. Since \( C_{Lmax} \)
varies with Mach number, a correspondence must be established between the specified Mach number and the specified number of g's in the turn. The Left or Right Turn (3) maneuvers limit the g's to the corresponding Mach number and $C_{L_{\text{max}}}$ condition. The Left or Right Turn (4) maneuvers allow the vehicle to dive and increase Mach number (and hence $C_{L_{\text{max}}}$) in order to obtain the specified number of g's in the turn. That is, if the specified $F'_n$ exceeds $F_{n_{\text{max}}}$, the constant-Mach-number maneuver requirement is initially abandoned to increase speed and $F_{n_{\text{max}}}$ so that

$$F_{n_{\text{max}}} > F'_n$$

Under the constraint conditions, the initial dive is defined arbitrarily by

$$a_{Ch} = 0$$

$$A_{CV} = -2.5 \ g \ cos \ \gamma$$

(102)

Left or Right Turn (5)

These are climbing or diving turns as specified by arguments of the number of g's required, i.e., $F_n/W$ and the bank angle $\psi_B$. As shown in Figs. 7 and 8, the commanded acceleration components are

$$a_{Ch} = -g \left(\frac{F_n}{W}\right) \sin \psi_B$$

$$a_{CV} = g \left(\frac{F_n}{W}\right) \cos \psi_B - g \ cos \ \gamma$$

(103)

A right turn results for positive values from 0 to 180 deg and a left turn results for negative $\psi_B$ values from 0 to -180 deg. Diving or climbing occurs for absolute-value magnitudes greater or less than 90 deg, respectively.
Climb or Dive (1)

These maneuvers are identical to the Left or Right Turn (1) described above except that the acceleration vector $\vec{a}_C$ is in the vertical plane. That is,

$$\vec{a}_C = \pm g \left( \frac{F_n}{W} + \cos \gamma \right) \vec{I}_D$$

(104)

The plus sign corresponds to the climb maneuver and the minus sign to the dive maneuver.

Climb (2)

This maneuver is a constant-Mach-number climb depending on excess thrust available at the time it is initiated. The procedure is the same as that for Left or Right Turn (2) described above except that the vector $\vec{a}_C$ is in the vertical plane. That is,

$$\vec{a}_C = g \left( \frac{F_n}{W} - \cos \gamma \right) \vec{I}_D$$

(105)

where $F_n$ is not specified but rather calculated from the relationships described in Left or Right Turn (2) and (3) (see Eq. (94)).

Barrel Roll (1)

In attempting to describe the trajectory of a barrel roll within the framework of coordinated turn definition (as described in Section IV), there does not seem to be universal agreement among pilots and engineers as to precise mathematical definition. The following equations were extracted from a Target Generator Program received from Eglin Air Force Base:

$$a_{Ch} = -g \left( \frac{F_n}{W} \right) \sin \psi_B$$

$$a_{CV} = g \left( \frac{F_n}{W} \right) \cos \psi_B - g \cos \gamma$$

(106)

*Private communication.
where the bank angle $\psi_B$ is obtained from a specified banking rate $\dot{\psi}_B$ so that

$$\psi_B = \int \dot{\psi}_B \, dt \quad (107)$$

Referring to Fig. 8, we note that the force $F_n$ rotates about the velocity vector $\vec{V}$. The acceleration component $a_{ch}$ varies sinusoidally, whereas the component $a_{cv}$ has a varying gravitational term $g \cos \gamma$. In calling for Barrel Roll (1), it is necessary to specify (1) the number of 360-deg rolls required, (2) the number of g's for $F_n/W$, and (3) the banking rate $\dot{\psi}_B$ in degrees per second.

**Barrel Roll (2)**

In order to decrease the altitude loss due to gravitational effects, this routine assumes that the net acceleration vector $\vec{a}_C$ rotates about the velocity vector $\vec{V}$ at a constant specified rate $\psi'$. The net accelerations in the horizontal and vertical planes are

$$a_{ch} = -a_C \sin \psi' \quad (108)$$

$$a_{cv} = a_C \cos \psi'$$

where $\psi'$ is given by

$$\psi' = \int \dot{\psi}' \, dt \quad (109)$$

Referring to Fig. 8, we see that the acceleration vector $\vec{a}_C$ will rotate uniformly about the velocity vector $\vec{V}$, but the force $F_n$ will vary with the rotation. $F_n$ is given by

$$F_n = \left( \frac{W}{g} \right) \left[ a_{ch}^2 + (a_{cv} + g \cos \gamma)^2 \right]^{1/2} \quad (110)$$

In calling for Barrel Roll (2), it is necessary to specify (1) the number of 360-deg rolls required, (2) the number of g's for $\vec{a}_C$ (not $F_n/W$), and (3) the angular rate $\dot{\psi}_B$ in degrees per second.
Constant-Roll-Angle Climbing or Diving Turn (GREYLL)

The term roll angle, as used here, is an angle referenced to the vehicle's pitch, roll, and yaw coordinate system, as distinguished from bank angle \( \psi_b \), which is referenced to a wind or velocity system. In TACTICS, the roll angle \( \psi \) is taken to be the angle between the vehicle pitch axis, i.e., the wings of the vehicle, and a line through the c.g. of the vehicle both normal to the longitudinal (i.e., roll) axis and parallel to the horizontal plane. The constant-roll-angle maneuver requires this angle to be held constant. The normal force in g's, \( F_n/N \), is also specified. For values of \( \psi \) between -90 deg to -180 deg and between +90 deg to +180 deg, a diving turn to the left or right, respectively, will result.

Since \( F_n \) and \( \psi \) (see Appendix A for definition) are specified, the subroutine solves the problem of finding the components \( a_{oh} \) and \( a_{ov} \) to force the following two conditions:

\[
a_{oh}^2 + \left( a_{ov} + g \cos \gamma_v \right)^2 = \left( \frac{g}{U} F_n \right)^2 \quad \text{constant} \tag{111}
\]

and

\[
\psi = \psi(a_{oh}, a_{ov}) = \text{constant} \tag{112}
\]

The functional relationship \( \psi(a_{oh}, a_{ov}) \) is not amenable to solving these two equations for \( a_{oh} \) and \( a_{ov} \) directly, so numerical methods are used.

The technique employed is to rotate the lift vector \( \mathbf{L} \) about \( \mathbf{V} \) by an amount \( \Delta \phi \) (the absolute value \( |\mathbf{L}| \) is known but the direction is not). With each incremental rotation \( \Delta \phi \), the angle \( \psi \) is computed until the required conditions are fulfilled within an error \( \Delta \psi \) less than 1 mrad. Note that by holding a roll angle of absolute magnitude greater than zero, the vehicle will continuously be climbing or diving until the longitudinal roll axis becomes vertical. However, the velocity vector will be

*Apparently, there is no universal agreement in texts or among engineers on terminology or exact definition.
lagging by approximately the angle of attack $\alpha$. When this condition is reached, roll angle becomes singular and a warning message is printed. Thereafter, the vehicle will continue its climb (or dive) until the velocity vector becomes vertical; then straight flight up (or down) is assumed.

**Ballistic 20-mm Projectile**

This routine simulates the trajectory of a 20-mm type M56A1 projectile. The drag coefficient $C_D$ versus Mach number characteristics are represented by several linear functions over various Mach-number regimes. The commanded accelerations correspond to a 0-g ballistic trajectory:

$$a_h = 0.0$$

$$a_v = -g \cos \gamma$$

$$F_n = 0.0$$

(113)

The projectile weight is 0.22 lb, and the reference area is 0.0033842 ft$^2$. The muzzle velocity ranges from 3350 to 3450 ft/sec and should be set as $\Delta V$ in the launch subroutine (see Launch above).

**CLOSED-LOOP CONTROL LAWS**

**Proportional Navigation**

The commanded lateral acceleration $\ddot{a}_C$ is proportional to the space rate of rotation of the LOS between missile and target. Expressed in vector notation,

$$\ddot{a}_C = \lambda V \omega_r \times \bar{I}_v$$

(114)

where

$\lambda$ = the "navigation constant" (it may be treated as either a constant or a variable)
\[ V = \text{missile speed} \]
\[ \vec{\omega}_r = \text{relative angular-rate vector as defined by Eq. (5) in Section III.} \]
\[ \vec{I}_V = \text{unit vector along the missile velocity } \vec{V}, \text{ i.e., } \vec{I}_V = \vec{V}/V \]

The direction of the acceleration may be defined by

\[ \vec{I}_1 = \vec{\omega}_r \times \vec{I}_V/\omega_r \]  \hspace{1cm} (115)

The commanded acceleration \( \vec{a}_C \) may be resolved into horizontal and vertical components by

\[ \vec{a}_{\text{Ch}} = a_C \left( \vec{I}_1 \cdot \vec{I}_A \right) \]
\[ \vec{a}_{\text{CV}} = a_C \left( \vec{I}_1 \cdot \vec{I}_D \right) \]  \hspace{1cm} (116)

**Biased Proportional Navigation**

The commanded lateral acceleration \( \vec{a}_C \) is proportional to a biased (or "modified") space rate of rotation of the LOS between interceptor and target. The bias term may be either a constant or a time variable; and its general purpose is (1) to provide for accelerations involved in the problem and/or (2) to shape (i.e., straighten or curve) the resultant trajectory. The calculations to determine this quantity may range in complexity from the estimation of a constant to the solution of complicated second-order prediction equations. The present version of this routine uses a bias term to account for an average boost velocity increment, such as would occur for a high-specific-impulse missile launch. It may be used, for example, in simulating a lead collision course where the interceptor aircraft is to steer so as to account for the \( \Delta V \) occurring at missile launch. The equations used are

\[ \vec{a}_C = \lambda V \left( \vec{\omega}_r - \vec{\omega}_B \right) \times \vec{I}_V \]  \hspace{1cm} (117)
where

$$\omega_B = \frac{(\Delta V \, \bar{r}_{ij} \times \bar{I}_V)}{r_{ij}^2}$$  \hspace{1cm} (118)$$

$$\Delta V = \text{terminal boost velocity}$$

$$\bar{r}_{ij} = \text{relative interceptor-target range vector}$$

$$\bar{I}_V = \text{unit vector along the interceptor velocity } \bar{V} \left( \bar{I}_V = \bar{V}/|\bar{V}| \right)$$

Note that the bias-term evaluation described above requires an a priori knowledge of only $\Delta V$ and LOS orientation and hence requires minimal hardware on board the vehicle.

**Lead Collision**

This guidance law may be considered as a more sophisticated version of biased proportional navigation. Since range and range-rate information are assumed available, it is feasible to make more accurate predictions in providing for the missile launch. Moreover, the effective gain, $\lambda$, may be varied as a function of a computed time-to-go, $T$. The miss vector (lead collision) form of guidance is briefly described in the following paragraphs. Consider the following miss vector diagram (the vectors are not necessarily co-planar), in which

$$T = \text{time-to-go, e.g., from "now" to the missile launching point}$$

$$\bar{M} = \text{a miss vector--to be driven to zero.}$$

$$\bar{F} = \text{vector representing the distance and direction a missile will travel after launch in a flight time } t_f$$

$$\bar{V}_F = \text{launching aircraft velocity vector}$$

$$\bar{V}_T = \text{target velocity vector}$$

$$\bar{r} = \bar{r}_{ij}, \text{ the relative range vector (LOS)}$$
With reference to the vector diagram,

\[ \overline{V}_T T + \overline{r} = \overline{V}_F \left( T - T_f \right) + \overline{F} + \overline{M} \]  \hspace{1cm} (119)

from which

\[ \overline{M} = \overline{r} - \overline{F} + \overline{V}_F T_f + \left( \overline{V}_T - \overline{V}_F \right) T \]  \hspace{1cm} (120)

Next, calculate the vector \( \overline{F} \) and the flight time \( T_f \). Designate \( r_o \) as some preset value of interceptor-target range at which the missile is to be launched. When \( r = r_o \) at time of launch,

\[ \overline{r}_o = \overline{r} \]

\[ T = T_f \]

\[ \left( V_F + \Delta V \right)^2 T_f^2 = r_o^2 + 2 \overline{r} \cdot \overline{V}_T T_f + V_T^2 T_f^2 \]  \hspace{1cm} (121)

From this quadratic equation \( T_f \) may be determined. Then

\[ \overline{F} = \Delta V T_f \overline{I}_V \]  \hspace{1cm} (122)
The vector $\vec{M}$ may now be resolved into components parallel and perpendicular to the LOS vector $\vec{r}$. First, consider the parallel component, designated $M_{||}$:

$$M_{||} = \vec{M} \cdot \vec{I}_r = r - \vec{F} \cdot \vec{I}_r + \vec{V}_F \cdot \vec{I}_r + r T$$  \hspace{1cm} (123)

since

$$\dot{r} = \left(\vec{V}_r - \vec{V}_F\right) \cdot \vec{I}_r$$  \hspace{1cm} (124)

Next arbitrarily set $M_{||} = 0$ and solve for $T$, the time-to-go.

$$T = \frac{\left(\vec{F} - \vec{V}_F\right) \cdot \vec{I}_r - r}{\dot{r}} = \frac{t_f \Delta \vec{V} \cdot \vec{I}_r - r}{\dot{r}}$$ \hspace{1cm} (125)

since

$$F = (\vec{V}_F + \Delta \vec{V}) t_f$$ \hspace{1cm} (126)

For notational convenience we will define

$$\vec{F}^* = \Delta V t_f \vec{I}_v$$ \hspace{1cm} (127)

Because of the arbitrary definition of $T$ in Eq. (125), the parallel component of miss $M_{||}$ is zero at $T$. The problem is reduced to finding an expression for interceptor lateral acceleration so that $M_\perp$ at time $T$ is driven to zero as shown below:
Note that the lateral acceleration is by definition normal to \( \overline{V}_F \). The lateral acceleration command may (arbitrarily) be made proportional to \( M_1 \)

\[
a_C \sim \delta
\]  

(128)

where

\[
\delta = \frac{M_1}{\overline{V}_F T + F^*}
\]

\[
a_C = \lambda \overline{V} \delta
\]

where \( \lambda \) represents the guidance-loop gain. The magnitude and orientation of \( \overline{a}_C \) is determined as follows:

\[
M_1 = |\overline{M} \times \overline{I}_r| = |-\overline{F^*} \times \overline{I}_r + T(\overline{V}_T - \overline{V}_F) \times \overline{I}_r| = |-\overline{F^*} \times \overline{I}_r + \overline{\omega} r T| \]  

(129)

The orientation of the acceleration \( \overline{a}_C \) is given by

\[
\overline{I}_l = \frac{\overline{a}_C/a_C}{M_1} = \frac{(\overline{M} \times \overline{I}_r) \times \overline{I}_V}{M_1}
\]  

(130)

The complete expression for \( \overline{a}_C \) may now be written as

\[
\overline{a}_C = \frac{\lambda \overline{V}_F}{\overline{V}_F T + F^*} \left[ -\left( \overline{F^*} \times \overline{I}_r \right) + \frac{\overline{\omega}}{\overline{r}} r T \right] \times \overline{I}_V
\]

(131)

This relationship may be expressed in a different form to show the analogy with biased proportional navigation:

\[
\overline{a}_C = \lambda' \overline{V} (\overline{\omega} - \overline{\omega}_B) \times \overline{I}_V
\]

(132)

where
\[
\lambda' = \frac{\lambda r_T}{V_r T + \bar{F}^*} \\
\omega_B = \frac{\bar{F}^* \times \bar{I}_r}{r T}
\]

(133)  
(134)

Note that \(\lambda'\) and \(\omega_B\) are time-varying functions.

**Pure Pursuit Course (1)**

The commanded lateral acceleration \(\bar{a}_C\) is proportional to the angular difference \(\Delta\gamma\) between the interceptor's velocity vector \(\bar{V}\) and the interceptor-target range vector \(\bar{r}_{ij}\). This may be expressed as follows:

\[
\Delta\gamma = \sin^{-1}|\bar{I}_V \times \bar{I}_r|
\]

\[
\bar{I}_l = (\bar{I}_V \times \bar{I}_r) \times \bar{I}_V/\Delta\gamma
\]

\[
\bar{a}_C = \lambda V \Delta\gamma \bar{I}_l
\]

(135)

\[
a_{Ch} = a_C(\bar{I}_l \cdot \bar{I}_A)
\]

\[
a_{CV} = a_C(\bar{I}_l \cdot \bar{I}_D)
\]

It is significant to mention that a pursuit-course navigation law is not usually applied for missile terminal homing guidance, since \(|\bar{a}_C|\) may become infinitely large as the relative range approaches zero. However, this control law is very useful for describing fighter aircraft "gunsight aiming" flight paths. The gain constant \(\lambda\) is arbitrarily selected, since it represents the pilot's reaction and skill in keeping the target in the crosshairs (practical values range from about 4 to 40).

**Pursuit (2)**

Pure pursuit-course navigation is defined in terms of maintaining the velocity vector of the pursuing vehicle along the LOS to the target.
In practice, a pilot would only be able to approximate a pure pursuit course without angle of attack or velocity information. Rather than the velocity vector, the most convenient reference is the airframe itself, i.e., the longitudinal axis. With modern jet aircraft in high-g maneuvers, angle-of-attack differences may range from 15 to 20 deg. The Pursuit (2) routine may be termed a deviated pursuit course where the bearing angle $B$ between the longitudinal axis of the pursuing vehicle and the LOS is maintained at a near-zero value. A necessary condition for beginning the deviated pursuit guidance is that the magnitude of the bearing angle be less than $\alpha_{max}$ which corresponds to the $C_{Lmax}$ condition. The routine calls for pure pursuit guidance until

$$|B| < \alpha_{max}$$

The trajectory we wish to simulate is described mathematically as follows: Find an $a_h$ and an $a_v$ resulting in a net acceleration vector $\overline{a}$ and a normal force vector $\overline{F_n}$. This normal force will require a lift vector $\overline{L}$ with a corresponding angle of attack $\alpha$. The problem requires solving for the three unknown components of $\overline{F_n}$, all functions of $C_L$ and $\alpha$. There are two functions which should be minimized:

$$f_1 (a_h, a_v) = \alpha - \gamma$$

(136)

where $\gamma$ is the angle between $\overline{V}$ and the LOS and $\alpha$ is the angle between $\overline{V}$ and the longitudinal axis $\overline{l}$, and

$$f_2 (a_h, a_v) = |B|$$

(137)

In other words, we would like $\alpha = \gamma$ and $|B| = 0$ for an idealized trajectory. Since nonlinear relationships (i.e., table values) and transcendental relationships are involved, numerical iterative techniques are used to determine the acceleration $\overline{a}$ and its components $a_h$ and $a_v$. As a first approximation, the proportional navigation law is used to obtain a first guess for $\overline{l}$. Next, the $C_L - \alpha$ correspondence is established for $\alpha = \gamma$, thereby determining $L$ and $\overline{F_n}$. Computations
are then performed to obtain \( a_h \), \( a_v \), and ultimately \( B \). The vector \( \overline{F_n} \) is then rotated by a small angle \( \overline{\epsilon} \) to minimize \(|B|\).

**On-Off (1)**

This guidance law (sometimes called bang-bang) is analogous to proportional navigation. The direction of the lateral acceleration unit vector \( \overline{I}_l \) is determined identically. However, the magnitude of the acceleration is not proportional to the angular rotation rate \(|\overline{\omega}_r|\) of the LOS, but dual-valued. That is,

\[
a = 0.0 \quad \text{when } 0 < |\overline{\omega}_r| < \overline{\epsilon}
\]

\[
a = a_{\text{Smax}} \quad \text{when } |\overline{\omega}_r| > \overline{\epsilon}
\]

where \( \overline{\epsilon} \) is some small specified threshold value of angular rate (e.g., 1 mrad/sec) required for stability.

**On-Off (2)**

This guidance law is analogous to pure pursuit-course navigation, and the direction of \( \overline{I}_l \) is determined identically. As in Eq. (138),

\[
a = 0.0 \quad \text{when } 0 < |\gamma| < \overline{\epsilon}
\]

\[
a = a_{\text{Smax}} \quad \text{when } |\gamma| > \overline{\epsilon}
\]

where \( \overline{\epsilon} \) is some small specified threshold value of angle (e.g., 1 mrad) required for stability.

**Missile (X)**

The commanded lateral acceleration \( \overline{a}_c \) is proportional to the space rate of rotation of the LOS between interceptor and target, i.e., proportional navigation. Many of the significant guidance and aerodynamic parameters vary with Mach number and are unique to this hypothetical missile design, as are the boost/burn/guide-time intervals. Accordingly,
these detailed characteristics are packaged into a specialized subroutine. The pertinent characteristics associated with this missile performance are listed below.

**Constants**

\[ A = 0.13635 \text{ ft}^2 \]
\[ W_o \text{ (initial weight)} = 187 \text{ lb} \]
\[ W_B \text{ (burnout weight)} = 125 \text{ lb} \]
\[ t_B \text{ (burn time)} = 4.75 \text{ sec} \]
\[ \dot{W} = (125 - 187)/4.75 = -13.06 \text{ lb/sec} \]
Thrust = 3053 lb

**Time Constants**

\[ \tau_1 = 0.1 \]
\[ \tau_2 = 0.15 \]
\[ \tau_3 = 0.15 \left( \frac{p_o}{p_0} \right) \]

**Parameters**

\[ \lambda = 0.35 + 12.22/M^{3/2} \]
\[ a_{S_{\text{max}}} = \min \left( \frac{15.0M}{\frac{p_o}{p_0}} - 0.95 \right) \]
\[ C_D = C_{D_0} + dC_D/d(C_L^2) \]

The values of \( dC_L/d\alpha \) and \( dC_D/d(C_L^2) \) with respect to Mach number are given in the tables on the following page:
<table>
<thead>
<tr>
<th>$\frac{dC_L}{d\alpha}$ (per rad)</th>
<th>Mach No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$23.52 + 109.9/M^2$</td>
<td>$\geq 1.85$</td>
</tr>
<tr>
<td>55.5</td>
<td>$\geq 1.5$, $&lt; 1.85$</td>
</tr>
<tr>
<td>$12 + 45M$</td>
<td>$\geq 1.1$, $&lt; 1.5$</td>
</tr>
<tr>
<td>37.5</td>
<td>$&lt; 1.1$</td>
</tr>
<tr>
<td>$0.51 + 2.52/M^2$</td>
<td>$\geq 1.5$</td>
</tr>
<tr>
<td>2.19</td>
<td>$\geq 1.0$, $&lt; 1.5$</td>
</tr>
<tr>
<td>$0.515 + 1.675M$</td>
<td>$\geq 0.6$, $&lt; 1.0$</td>
</tr>
<tr>
<td>1.5</td>
<td>$&lt; 0.6$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\frac{dC_D}{d(C_L^2)}$ (per rad)</th>
<th>Mach No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10 + 93/M$</td>
<td>$\geq 1.5$</td>
</tr>
<tr>
<td>72</td>
<td>$&lt; 1.5$</td>
</tr>
<tr>
<td>$24.33 + 43.33M$</td>
<td>$\geq 0.5$, $&lt; 1.1$</td>
</tr>
<tr>
<td>46</td>
<td>$&lt; 0.5$</td>
</tr>
</tbody>
</table>
Appendix D

INSTRUCTIONS FOR CALLING OPTIONAL SUBROUTINES

Many optional subroutines are available with TACTICS, including maneuver and guidance routines. (Launch and captive-flight routines are considered special cases in this category, as shown in Table 2.) When a maneuver or guidance subroutine is called, the following three arguments must always be specified (unless the routines are formulated to represent specific vehicles such as MISILX):

I: Vehicle to be used (1, 2, or 3).
IAERØ: Type of aerodynamics to be used.
   IAERØ = 1, analytic functions.
   IAERØ = 2, aerodynamic tables.
   IAERØ = 3, VDØT = 0.0.
ITHR: Thrust to be used (1b).
   ITHR = 1, afterburner thrust obtained from tables.
   ITHR = 2, military thrust obtained from tables.
   ITHR = 3, afterburner thrust = constant (DATA 96, 101, 106).
   ITHR = 4, military thrust = constant (DATA 95, 100, 105).
   ITHR = 5, thrust = 0.0.

In addition, the following arguments may be required:
DELV: Boost velocity of missile (ft/sec).
MØDE: Flag indicating captive-flight option.
   MØDE = 1, holds missile in captive flight on fighter.
   MØDE = 2, sets all quantities related to designated vehicle equal to zero.
   MØDE = 3, holds missile in captive flight on target.
EPSLØN: Threshold value used in on-off control laws for stability.
LEVEL(I): Flag set in subroutine STRVL used to communicate to POLICY that the vehicle velocity vector is in a horizontal plane within a tolerance of 0.002 rad.
GFØRC: Total lateral acceleration, including gravitational effects, for a maneuver (g's).
Table 2  
MANEUVER AND GUIDANCE ROUTINES

<table>
<thead>
<tr>
<th>Routine and Argument Listing</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIASPN(I, DELV, IAERØ, ITHR)</td>
<td>Biased proportional navigation guidance law</td>
</tr>
<tr>
<td>BRLRL1(I, RØLLS, IRØLL, GFØRC, RØLLRT, IAERØ, ITHR)</td>
<td>Constant-(g) barrel roll where the force (F_n) rotates about velocity vector</td>
</tr>
<tr>
<td>BRLRL2(I, RØLLS, IRØLL, GFØRC, RØLLRT, IAERØ, ITHR)</td>
<td>Constant-(g) barrel roll where the lateral acceleration (a_o) rotates about the velocity vector</td>
</tr>
<tr>
<td>B20MM(I)</td>
<td>Simulating ballistics for 20-mm cannon projectiles</td>
</tr>
<tr>
<td>CAPFLT(I, MODE)</td>
<td>Captive flight</td>
</tr>
<tr>
<td>CLIMB1(I, GFØRC, IAERØ, ITHR)</td>
<td>Simple constant-(g) climb</td>
</tr>
<tr>
<td>CLIMB2(^a)(I, IAERØ, ITHR)</td>
<td>Constant-Mach climb depending on available excess thrust</td>
</tr>
<tr>
<td>DIVE1(I, GFØRC, IAERØ, ITHR)</td>
<td>Simple constant-(g) dive</td>
</tr>
<tr>
<td>GFRØLL(I, GFØRC, RØLL, IAERØ, ITHR)</td>
<td>Desired GFØRC AND RØLL (GFØRC and RØLL in argument listing) are specified and corresponding accelerations are computed</td>
</tr>
<tr>
<td>LAUNCH(I, DELV, IAERØ, ITHR)</td>
<td>Launches missile</td>
</tr>
<tr>
<td>LEADC1(I, DELV, IAERØ, ITHR)</td>
<td>Lead collision guidance law</td>
</tr>
<tr>
<td>LTRNL(I, GFØRC, IAERØ, ITHR)</td>
<td>Simple constant-(g) left turn</td>
</tr>
<tr>
<td>LTRN2(^a)(I, IAERØ, ITHR)</td>
<td>Constant-altitude, constant-Mach left turn depending on available excess thrust</td>
</tr>
<tr>
<td>LTRN3(^a)(I, GFØRC, IAERØ, ITHR)</td>
<td>Constant-Mach left turn where GFØRC is specified</td>
</tr>
<tr>
<td>LTRN4(^a)(I, GFØRC, IAERØ, ITHR)</td>
<td>Constant-Mach left turn unless aerodynamic flight conditions (CLMAY) prohibit obtaining specified value, in which case aircraft will maneuver at maximum acceleration in conformity with CLMAY limitations, keeping Mach number constant</td>
</tr>
</tbody>
</table>

\(^a\)These subroutines require the routine MACHRØ.
Table 2 (continued)

<table>
<thead>
<tr>
<th>Routine and Argument Listing</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTRNS&lt;sup&gt;a&lt;/sup&gt;(I, GFØRC, RØLLL, IAERØ, ITHR)</td>
<td>Constant-g climbing or diving left turn as defined by bank angle RØLLL (see Appendix C)</td>
</tr>
<tr>
<td>MISILX(I)</td>
<td>Proportional navigation missile</td>
</tr>
<tr>
<td>MISIL2(I)</td>
<td>Rearward-launched MISILX(I)</td>
</tr>
<tr>
<td>ØNØFF(I, EPLØN, IAERØ, ITHR)</td>
<td>Bang-bang (on-off) control law, analogous to PRØNAV</td>
</tr>
<tr>
<td>ØNØFF2(I, EPLØN, IAERØ, ITHR)</td>
<td>Bang-bang (on-off) control law, analogous to PRSUIT</td>
</tr>
<tr>
<td>PRØNAV(I, IAERØ, ITHR)</td>
<td>Proportional-navigation guidance law</td>
</tr>
<tr>
<td>PRSUIT(I, IAERØ, ITHR)</td>
<td>Pursuit-course navigation, velocity vector pointed down LOS</td>
</tr>
<tr>
<td>PRSUIT2(I, IAERØ, ITHR)</td>
<td>Pursuit-course navigation, thrust vector pointed down LOS; must be used with subroutine FUNKIN(I) (See Appendix C)</td>
</tr>
<tr>
<td>RTRN1(I, GFØRC, IAERØ, ITHR)</td>
<td>Simple constant-g right turn</td>
</tr>
<tr>
<td>RTRN2&lt;sup&gt;a&lt;/sup&gt;(I, IAERØ, ITHR)</td>
<td>Constant-altitude, constant-Mach right turn depending on available excess thrust</td>
</tr>
<tr>
<td>RTRN3&lt;sup&gt;a&lt;/sup&gt;(I, GFØRC, IAERØ, ITHR)</td>
<td>Constant-Mach right turn where GFØRC is specified</td>
</tr>
<tr>
<td>RTRN4&lt;sup&gt;a&lt;/sup&gt;(I, GFØRC, IAERØ, ITHR)</td>
<td>Constant-Mach right turn unless aerodynamic flight conditions (CLMAX) prohibit obtaining specified value, in which case aircraft will maneuver at maximum acceleration in conformity with CLMAX limitations, keeping Mach number constant</td>
</tr>
<tr>
<td>RTRNS&lt;sup&gt;a&lt;/sup&gt;(I, GFØRC, RØLLL, IAERØ, ITHR)</td>
<td>Constant-g climbing or diving right turn as defined by bank angle RØLLL (see Appendix C)</td>
</tr>
<tr>
<td>STRFLT(I, IAERØ, ITHR)</td>
<td>Straight flight</td>
</tr>
<tr>
<td>STRLVL(I, IAERØ, ITHR, LEVEL)</td>
<td>Level off to horizontal position</td>
</tr>
</tbody>
</table>

<sup>a</sup>These subroutines require the routine MACHR8.
Appendix E
DESCRIPTION OF INPUT DATA

<table>
<thead>
<tr>
<th>Data No.</th>
<th>Program Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IRF</td>
<td></td>
<td>Position flag, vehicle 1 (Cartesian coordinates); IRF = 0</td>
</tr>
<tr>
<td>2</td>
<td>R(1,1)</td>
<td>x₁</td>
<td>x-coordinate, vehicle 1</td>
</tr>
<tr>
<td>3</td>
<td>R(1,2)</td>
<td>y₁</td>
<td>y-coordinate, vehicle 1</td>
</tr>
<tr>
<td>4</td>
<td>R(1,3)</td>
<td>z₁</td>
<td>z-coordinate, vehicle 1</td>
</tr>
<tr>
<td>5</td>
<td>W₀(1)</td>
<td>W₀₁</td>
<td>Initial weight, vehicle 1</td>
</tr>
<tr>
<td>6</td>
<td>IRF</td>
<td></td>
<td>Position flag, vehicle 1 (spherical coordinates); IRF = 1</td>
</tr>
<tr>
<td>7</td>
<td>R(1,4)</td>
<td>r₁</td>
<td>Range vector magnitude, vehicle 1</td>
</tr>
<tr>
<td>8</td>
<td>R(1,5)</td>
<td>θ₁</td>
<td>Range angle measured in horizontal plane, vehicle 1 (see Fig. 3)</td>
</tr>
<tr>
<td>9</td>
<td>R(1,6)</td>
<td>φ₁</td>
<td>Range angle measured in vertical plane, vehicle 1 (see Fig. 3)</td>
</tr>
<tr>
<td>10</td>
<td>AREA(1)</td>
<td>A₁</td>
<td>Reference area, vehicle 1</td>
</tr>
<tr>
<td>11</td>
<td>IVF</td>
<td></td>
<td>Velocity flag, vehicle 1 (Cartesian coordinates); IVF = 0</td>
</tr>
<tr>
<td>12</td>
<td>V(1,1)</td>
<td>.x₁</td>
<td>x-component of velocity, vehicle 1</td>
</tr>
<tr>
<td>13</td>
<td>V(1,2)</td>
<td>.y₁</td>
<td>y-component of velocity, vehicle 1</td>
</tr>
<tr>
<td>14</td>
<td>V(1,3)</td>
<td>.z₁</td>
<td>z-component of velocity, vehicle 1</td>
</tr>
<tr>
<td>15</td>
<td>ASMAX(1)</td>
<td>aₘₚₓₙ₁</td>
<td>Maximum lateral acceleration limit (structural), vehicle 1</td>
</tr>
<tr>
<td>16</td>
<td>IVF</td>
<td></td>
<td>Velocity flag, vehicle 1 (spherical coordinates); IVF = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Magnitude only (calls AIM); IVF = 2</td>
</tr>
<tr>
<td>17</td>
<td>V(1,4)</td>
<td>V₁</td>
<td>Velocity vector magnitude, vehicle 1</td>
</tr>
<tr>
<td>18</td>
<td>V(1,5)</td>
<td>θᵥ₁</td>
<td>Velocity angle measured in horizontal plane, vehicle 1 (see Fig. 4)</td>
</tr>
<tr>
<td>19</td>
<td>V(1,6)</td>
<td>Y₁</td>
<td>Flight-path angle measured in vertical plane, vehicle 1 (see Fig. 4)</td>
</tr>
</tbody>
</table>

*Units are distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft²).*
<table>
<thead>
<tr>
<th>Data No.</th>
<th>Program Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>JATMOS</td>
<td></td>
<td>Flag for reading in velocity magnitudes (DATA 17, 33, and 36) JATMOS = 0 for ft/sec JATMOS = 1 for Mach number (applies to all vehicles)</td>
</tr>
<tr>
<td>21</td>
<td>TGUIDE(1)</td>
<td>$t_{g1}$</td>
<td>Time interval that vehicle 1 (if a missile) is to fly unguided after launch</td>
</tr>
<tr>
<td>25</td>
<td>W0(2)</td>
<td>$W_{02}$</td>
<td>Initial weight, vehicle 2</td>
</tr>
<tr>
<td>26</td>
<td>AREA(2)</td>
<td>$A_{2}$</td>
<td>Reference area, vehicle 2</td>
</tr>
<tr>
<td>27</td>
<td>ASMAX(2)</td>
<td>$a_{s max2}$</td>
<td>Maximum lateral acceleration limit (structural), vehicle 2</td>
</tr>
<tr>
<td>28</td>
<td>ICAP</td>
<td></td>
<td>Flag to indicate initial-condition flight status of vehicle 2 (see Appendix H)</td>
</tr>
<tr>
<td>29</td>
<td>RMTMAX</td>
<td></td>
<td>Maximum flight range, vehicle 2</td>
</tr>
<tr>
<td>30</td>
<td>R(2,1)</td>
<td>$x_{2}$</td>
<td>x-coordinate, vehicle 2</td>
</tr>
<tr>
<td>31</td>
<td>R(2,2)</td>
<td>$y_{2}$</td>
<td>y-coordinate, vehicle 2</td>
</tr>
<tr>
<td>32</td>
<td>R(2,3)</td>
<td>$z_{2}$</td>
<td>z-coordinate, vehicle 2</td>
</tr>
<tr>
<td>33</td>
<td>V(2,4)</td>
<td>$v_{2}$</td>
<td>Velocity vector magnitude, vehicle 2</td>
</tr>
<tr>
<td>34</td>
<td>V(2,5)</td>
<td>$e_{v2}$</td>
<td>Angle measured in horizontal plane, vehicle 2 (see Fig. 4)</td>
</tr>
<tr>
<td>35</td>
<td>V(2,6)</td>
<td>$\gamma_{2}$</td>
<td>Flight path angle measured in vertical plane, vehicle 2 (see Fig. 4)</td>
</tr>
<tr>
<td>36</td>
<td>TBURN1</td>
<td>$t_{B1}$</td>
<td>First stage rocket motor burning time (may be used for vehicle 1, 2, or 3)</td>
</tr>
<tr>
<td>37</td>
<td>TBURN2</td>
<td>$t_{B2}$</td>
<td>Second stage rocket motor burning time (may be used for vehicle 1, 2, or 3)</td>
</tr>
<tr>
<td>38</td>
<td>KLAUN</td>
<td></td>
<td>Decimal fraction of vehicle's maximum range at which it is to be launched (may be used for vehicle 1, 2, or 3)</td>
</tr>
<tr>
<td>Data No.</td>
<td>Program Variable</td>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>39</td>
<td>TGUIDE(2)</td>
<td>(t_{g2})</td>
<td>Time interval that vehicle 2 (if a missile) is to fly unguided after launch</td>
</tr>
<tr>
<td>40</td>
<td>IRT</td>
<td></td>
<td>Position flag, vehicle 3 (Cartesian coordinates); IRT = 0</td>
</tr>
<tr>
<td>41</td>
<td>R(3,1)</td>
<td>(x_3)</td>
<td>x-coordinate, vehicle 3</td>
</tr>
<tr>
<td>42</td>
<td>R(3,2)</td>
<td>(y_3)</td>
<td>y-coordinate, vehicle 3</td>
</tr>
<tr>
<td>43</td>
<td>R(3,3)</td>
<td>(z_3)</td>
<td>z-coordinate, vehicle 3</td>
</tr>
<tr>
<td>44</td>
<td>W0(3)</td>
<td>(W_{03})</td>
<td>Initial weight, vehicle 3</td>
</tr>
<tr>
<td>45</td>
<td>IRT</td>
<td></td>
<td>Position flag, vehicle 3 (spherical coordinates); IRT = 1</td>
</tr>
<tr>
<td>46</td>
<td>R(3,4)</td>
<td>(r_3)</td>
<td>Range vector magnitude, vehicle 3</td>
</tr>
<tr>
<td>47</td>
<td>R(3,5)</td>
<td>(\theta_3)</td>
<td>Range angle measured in horizontal plane, vehicle 3 (see Fig. 3)</td>
</tr>
<tr>
<td>48</td>
<td>R(3,6)</td>
<td>(\varphi_3)</td>
<td>Range angle measured in vertical plane, vehicle 3 (see Fig. 3)</td>
</tr>
<tr>
<td>49</td>
<td>AREA(3)</td>
<td>(A_3)</td>
<td>Reference area, vehicle 3</td>
</tr>
<tr>
<td>50</td>
<td>IVT</td>
<td></td>
<td>Velocity flag, vehicle 3 (Cartesian coordinates); IVT = 0</td>
</tr>
<tr>
<td>51</td>
<td>V(3,1)</td>
<td>(x_3)</td>
<td>x-component of velocity, vehicle 3</td>
</tr>
<tr>
<td>52</td>
<td>V(3,2)</td>
<td>(y_3)</td>
<td>y-component of velocity, vehicle 3</td>
</tr>
<tr>
<td>53</td>
<td>V(3,3)</td>
<td>(z_3)</td>
<td>z-component of velocity, vehicle 3</td>
</tr>
<tr>
<td>54</td>
<td>ASMAX(3)</td>
<td>(a_{\text{Smax3}})</td>
<td>Maximum lateral acceleration limit (structural), vehicle 3</td>
</tr>
<tr>
<td>55</td>
<td>IVT</td>
<td></td>
<td>Velocity flag, vehicle 3 (spherical coordinates); IVT = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Magnitude only (calls AIM); IRF = 2</td>
</tr>
<tr>
<td>56</td>
<td>V(3,4)</td>
<td>(V_3)</td>
<td>Velocity-vector magnitude, vehicle 3</td>
</tr>
<tr>
<td>57</td>
<td>V(3,5)</td>
<td>(\theta_{V3})</td>
<td>Velocity angle measured in horizontal plane, vehicle 3 (see Fig. 4)</td>
</tr>
<tr>
<td>58</td>
<td>V(3,6)</td>
<td>(\gamma_3)</td>
<td>Flight-path angle measured in vertical plane, vehicle 3 (see Fig. 4)</td>
</tr>
<tr>
<td>59</td>
<td>TGUIDE(3)</td>
<td>(t_{g3})</td>
<td>Time interval that vehicle 3 (if a missile) is to fly unguided after launch</td>
</tr>
<tr>
<td>Data No.</td>
<td>Program Variable</td>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td>(Not used)</td>
</tr>
<tr>
<td>61</td>
<td></td>
<td></td>
<td>(Not used)</td>
</tr>
<tr>
<td>62</td>
<td>TIME</td>
<td>( t_0 )</td>
<td>Running time (when entered as input data, it is equivalent to initial time, i.e., time at which problem is to start)</td>
</tr>
<tr>
<td>63</td>
<td>DTP( \phi )</td>
<td></td>
<td>Print interval, i.e., time increment for printout</td>
</tr>
<tr>
<td>64</td>
<td>T( \Theta )( T )( A )AL</td>
<td></td>
<td>Time limit placed on internal program running time if it is not to be terminated after miss calculation</td>
</tr>
<tr>
<td>65</td>
<td>ITAU(1)</td>
<td></td>
<td>Flag indicating number of first-order time lags, vehicle 1 (see Section V)</td>
</tr>
<tr>
<td>66</td>
<td>TAU(1,1)</td>
<td>( \tau_{11} )</td>
<td>First time lag, vehicle 1</td>
</tr>
<tr>
<td>67</td>
<td>TAU(1,2)</td>
<td>( \tau_{12} )</td>
<td>Second time lag, vehicle 1</td>
</tr>
<tr>
<td>68</td>
<td>TAU(1,3)</td>
<td>( \tau_{13} )</td>
<td>Third time lag, vehicle 1</td>
</tr>
<tr>
<td>69</td>
<td>LAMDAO(1)</td>
<td>( \lambda_{01} )</td>
<td>Navigation constant for guidance, vehicle 1</td>
</tr>
<tr>
<td>70</td>
<td>ITAU(2)</td>
<td></td>
<td>Flag indicating number of first-order time lags, vehicle 2 (see Section V)</td>
</tr>
<tr>
<td>71</td>
<td>TAU(2,1)</td>
<td>( \tau_{21} )</td>
<td>First time lag, vehicle 2</td>
</tr>
<tr>
<td>72</td>
<td>TAU(2,2)</td>
<td>( \tau_{22} )</td>
<td>Second time lag, vehicle 2</td>
</tr>
<tr>
<td>73</td>
<td>TAU(2,3)</td>
<td>( \tau_{23} )</td>
<td>Third time lag, vehicle 2</td>
</tr>
<tr>
<td>74</td>
<td>LAMDAO(2)</td>
<td>( \lambda_{02} )</td>
<td>Navigation constant for guidance, vehicle 2</td>
</tr>
<tr>
<td>75</td>
<td>ITAU(3)</td>
<td></td>
<td>Flag indicating number of first-order time lags, vehicle 3 (see Section V)</td>
</tr>
<tr>
<td>76</td>
<td>TAU(3,1)</td>
<td>( \tau_{31} )</td>
<td>First time lag, vehicle 3</td>
</tr>
<tr>
<td>77</td>
<td>TAU(3,2)</td>
<td>( \tau_{32} )</td>
<td>Second time lag, vehicle 3</td>
</tr>
<tr>
<td>78</td>
<td>TAU(3,3)</td>
<td>( \tau_{33} )</td>
<td>Third time lag, vehicle 3</td>
</tr>
<tr>
<td>79</td>
<td>LAMDAO(3)</td>
<td>( \lambda_{03} )</td>
<td>Navigation constant for guidance, vehicle 3</td>
</tr>
<tr>
<td>Data No.</td>
<td>Problem Variable</td>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>80</td>
<td>CLMAX(1)</td>
<td>$C_{L_{\text{max}1}}$</td>
<td>Maximum aerodynamic lift coefficient</td>
</tr>
<tr>
<td>81</td>
<td>CD0C0N(1)</td>
<td>$C_{D_{01}}$</td>
<td>Zero or profile drag coefficient: $C_D$, vehicle 1</td>
</tr>
<tr>
<td>82</td>
<td>BC0N(1)</td>
<td>$dC_D/dC_L^2$</td>
<td>Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient, vehicle 1</td>
</tr>
<tr>
<td>83</td>
<td>SL0PE(1)</td>
<td>$dC_L/d\alpha_1$</td>
<td>Slope of $C_L$ vs $\alpha$ curve, assumed to be a constant (analytic functions), vehicle 1</td>
</tr>
<tr>
<td>84</td>
<td>ALPHAO(1)</td>
<td>$\alpha_{01}$</td>
<td>Total zero-lift angle of attack, vehicle 1</td>
</tr>
<tr>
<td>85</td>
<td>CLMAX(2)</td>
<td>$C_{L_{\text{max}2}}$</td>
<td>Maximum aerodynamic lift coefficient, vehicle 2</td>
</tr>
<tr>
<td>86</td>
<td>CD0C0N(2)</td>
<td>$C_{D_{02}}$</td>
<td>Zero or profile drag coefficient: $C_D$, vehicle 2</td>
</tr>
<tr>
<td>87</td>
<td>BC0N(2)</td>
<td>$dC_D/dC_L^2$</td>
<td>Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient, vehicle 2</td>
</tr>
<tr>
<td>88</td>
<td>SL0PE(2)</td>
<td>$dC_L/d\alpha_2$</td>
<td>Slope of $C_L$ vs $\alpha$ curve, assumed to be a constant, vehicle 2</td>
</tr>
<tr>
<td>89</td>
<td>ALPHAO(2)</td>
<td>$\alpha_{02}$</td>
<td>Total zero-lift angle of attack, vehicle 2</td>
</tr>
<tr>
<td>90</td>
<td>CLMAX(3)</td>
<td>$C_{L_{\text{max}3}}$</td>
<td>Maximum aerodynamic lift coefficient, vehicle 3</td>
</tr>
<tr>
<td>91</td>
<td>CD0C0N(3)</td>
<td>$C_{D_{03}}$</td>
<td>Zero or profile drag coefficient, vehicle 3</td>
</tr>
<tr>
<td>92</td>
<td>BC0N(3)</td>
<td>$dC_D/dC_L^2$</td>
<td>Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient, vehicle 3</td>
</tr>
<tr>
<td>93</td>
<td>SL0PE(3)</td>
<td>$dC_L/d\alpha_3$</td>
<td>Slope of $C_L$ vs $\alpha$ curve, assumed to be a constant, vehicle 3</td>
</tr>
<tr>
<td>94</td>
<td>ALPHAO(3)</td>
<td>$\alpha_{03}$</td>
<td>Zero-lift angle of attack, vehicle 3</td>
</tr>
</tbody>
</table>

*Note that DATA 80-109 are required only when analytic functions are used instead of table values.*
<table>
<thead>
<tr>
<th>Data No.*</th>
<th>Program Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>THCØN(1)</td>
<td>T(_{M1})</td>
<td>Constant for military thrust, vehicle 1</td>
</tr>
<tr>
<td>96</td>
<td>TABCØN(1)</td>
<td>T(_{ab1})</td>
<td>Constant for afterburner thrust, vehicle 1</td>
</tr>
<tr>
<td>97</td>
<td>IMPLSE(1)</td>
<td>I(_1)</td>
<td>Specific impulse of rocket motor, vehicle 1</td>
</tr>
<tr>
<td>98</td>
<td>WBCRN(1)</td>
<td>W(_{B1})</td>
<td>Weight at rocket motor burnout, vehicle 1</td>
</tr>
<tr>
<td>99</td>
<td>ABØØST(1)</td>
<td>a(_{b1})</td>
<td>Boost acceleration (assumed to be a constant), vehicle 1</td>
</tr>
<tr>
<td>100</td>
<td>THCØN(2)</td>
<td>T(_{M2})</td>
<td>Constant for military thrust, vehicle 2</td>
</tr>
<tr>
<td>101</td>
<td>TABCØN(2)</td>
<td>T(_{ab2})</td>
<td>Constant for afterburner thrust, vehicle 2</td>
</tr>
<tr>
<td>102</td>
<td>IMPLSE(2)</td>
<td>I(_2)</td>
<td>Specific impulse of rocket motor, vehicle 2</td>
</tr>
<tr>
<td>103</td>
<td>WBCRN(2)</td>
<td>W(_{B2})</td>
<td>Weight at rocket motor burnout, vehicle 2</td>
</tr>
<tr>
<td>104</td>
<td>ABØØST(2)</td>
<td>a(_{b2})</td>
<td>Boost acceleration (assumed to be a constant), vehicle 2</td>
</tr>
<tr>
<td>105</td>
<td>THCØN(3)</td>
<td>T(_{M3})</td>
<td>Constant for military thrust, vehicle 3</td>
</tr>
<tr>
<td>106</td>
<td>TABCØN(3)</td>
<td>T(_{ab3})</td>
<td>Constant for afterburner thrust, vehicle 3</td>
</tr>
<tr>
<td>107</td>
<td>IMPLSE(3)</td>
<td>I(_3)</td>
<td>Specific impulse of rocket motor, vehicle 3</td>
</tr>
<tr>
<td>108</td>
<td>WBCRN(3)</td>
<td>W(_{B3})</td>
<td>Weight at rocket motor burnout, vehicle 3</td>
</tr>
<tr>
<td>109</td>
<td>ABØØST(3)</td>
<td>a(_{b3})</td>
<td>Boost acceleration (assumed to be a constant), vehicle 3</td>
</tr>
<tr>
<td>110</td>
<td>KINTEGR</td>
<td></td>
<td>Flat- or round-earth coordinate system flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flat-earth: KINTEGR = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Round-earth: KINTEGR = 1</td>
</tr>
<tr>
<td>111</td>
<td>ALT(1)</td>
<td>h(_1)</td>
<td>Altitude, vehicle 1 (geocentric coordinates)</td>
</tr>
</tbody>
</table>

*Note that DATA 80-109 are required only when analytic functions are used instead of table values.*
<table>
<thead>
<tr>
<th>Data No.</th>
<th>Program Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>LONG(1)</td>
<td>$\lambda_1$</td>
<td>Longitude, vehicle 1 (geocentric coordinates)</td>
</tr>
<tr>
<td>113</td>
<td>LAT(1)</td>
<td>$\phi_1$</td>
<td>Latitude, vehicle 1 (geocentric coordinates)</td>
</tr>
</tbody>
</table>
| 114      | INERF            |         | Flag for reading vehicle 1 velocity in local or inertial system 
Local: INERF = 0 
Inertial: INERF = 1 |
| 115      | IRÔTS            |         | Flag for earth's rotation 
Nonrotating: IRÔTS = 0 
Rotating: IRÔTS = 1 |
| 116      | ALT(3)           | $h_3$   | Altitude, vehicle 3 (geocentric coordinates) |
| 117      | LONG(3)          | $\lambda_3$ | Longitude, vehicle 3 (geocentric coordinates) |
| 118      | LAT(3)           | $\phi_3$  | Latitude, vehicle 3 (geocentric coordinates) |
| 119      | INERT            |         | Flag for reading vehicle 3 velocity in local or inertial system 
Local: INERT = 0 
Inertial: INERT = 1 |
| 120      | LONGO            | $\lambda_0$ | Longitude of the local coordinate system origin |
| 121      | LATO             | $\phi_0$  | Latitude of the local coordinate system origin |
| 122      | JINTEG           |         | Integration flag 
Variable-step Adams-Moulton: JINTEG = 0 
Fixed-step Runge-Kutta: JINTEG = 1 
Fixed-step Adams-Moulton: JINTEG = 2 
Variable step with exact print-out: JINTEG = 3 (see Section XIII) |
<p>| 123      | ERTEST           |         | Number of significant digits of accuracy required (see Section XIII) of numerical integration |
| 124      | MINMR            | $R_{MIN}$ | Vehicle 2 range to target within which program will automatically initiate process for miss-distance computation |
| 125      | DVTH(1)          | $\Delta \theta_1$ | Assumed error in $\theta$, for aiming vehicle 1 |</p>
<table>
<thead>
<tr>
<th>Data No.</th>
<th>Program Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
<td>DVTH(2)</td>
<td>$\Delta \theta_2$</td>
<td>Assumed error in $\theta_y$ for aiming vehicle 2</td>
</tr>
<tr>
<td>127</td>
<td>DVTH(3)</td>
<td>$\Delta \theta_3$</td>
<td>Assumed error in $\theta_y$ for aiming vehicle 3</td>
</tr>
<tr>
<td>128</td>
<td>DVPHI(1)</td>
<td>$\Delta \gamma_1$</td>
<td>Assumed error in $\gamma_1$ for aiming vehicle 1</td>
</tr>
<tr>
<td>129</td>
<td>DVPHI(2)</td>
<td>$\Delta \gamma_2$</td>
<td>Assumed error in $\gamma_2$ for aiming vehicle 2</td>
</tr>
<tr>
<td>130</td>
<td>DVPHI(3)</td>
<td>$\Delta \gamma_3$</td>
<td>Assumed error in $\gamma_3$ for aiming vehicle 3</td>
</tr>
<tr>
<td>131</td>
<td>BETA(1)</td>
<td>$\beta_1$</td>
<td>Ballistic coefficient, vehicle 1</td>
</tr>
<tr>
<td>132</td>
<td>BETA(2)</td>
<td>$\beta_2$</td>
<td>Ballistic coefficient, vehicle 2</td>
</tr>
<tr>
<td>133</td>
<td>BETA(3)</td>
<td>$\beta_3$</td>
<td>Ballistic coefficient, vehicle 3</td>
</tr>
<tr>
<td>134</td>
<td>DTMN</td>
<td></td>
<td>Minimum integration step size for backup to compute miss distance</td>
</tr>
<tr>
<td>135</td>
<td>IMIN</td>
<td></td>
<td>Minimum allowable size of integration step used in variable Adams-Moulton mode integration (excluding miss distance computation)</td>
</tr>
<tr>
<td>136</td>
<td>DTØ</td>
<td></td>
<td>Initial (starting) value of integration step size</td>
</tr>
<tr>
<td>137</td>
<td>IRMX</td>
<td></td>
<td>Maximum specified value for integration step size used in variable Adams-Moulton mode integration</td>
</tr>
<tr>
<td>138</td>
<td>ELEVMX(1)</td>
<td>$\epsilon_{\text{max}1}$</td>
<td>Maximum elevation angle limit, vehicle 1</td>
</tr>
<tr>
<td>139</td>
<td>ELEVMX(2)</td>
<td>$\epsilon_{\text{max}2}$</td>
<td>Maximum elevation angle limit, vehicle 2</td>
</tr>
<tr>
<td>140</td>
<td>ELEVMX(3)</td>
<td>$\epsilon_{\text{max}3}$</td>
<td>Maximum elevation angle limit, vehicle 3</td>
</tr>
<tr>
<td>141</td>
<td>AZMAX(1)</td>
<td>$\eta_{\text{max}1}$</td>
<td>Maximum azimuth angle limit, vehicle 1</td>
</tr>
<tr>
<td>142</td>
<td>AZMAX(2)</td>
<td>$\eta_{\text{max}2}$</td>
<td>Maximum azimuth angle limit, vehicle 2</td>
</tr>
<tr>
<td>143</td>
<td>AZMAX(3)</td>
<td>$\eta_{\text{max}3}$</td>
<td>Maximum azimuth angle limit, vehicle 3</td>
</tr>
</tbody>
</table>
Appendix F

AERODYNAMIC AND PROPULSION TABLES

Tables furnishing variables for computing the aerodynamic functions and the propulsion characteristics of specific aircraft, including the F-104 and F-105, are available. These are to be used when IAERDN = 2 (when the table section of AERODN is being used).

An example set of tables for an aircraft is given below; sets of tables for other aircraft may be constructed following a similar format.

- Maximum lift coefficient as a function of Mach number.
- Drag coefficient as a function of Mach number and lift coefficient.
- $\Delta C_D$, change in $C_D$ when external stores, e.g., fuel tanks, are jettisoned (the program does not now have the feature for using $\Delta C_D$, but cards must be included in data deck).
- Afterburner thrust as a function of Mach number and altitude.
- Fuel flow (afterburner thrust) as a function of Mach number and altitude.
- Military thrust as a function of Mach number and altitude.
- Fuel flow (military thrust) as a function of Mach number and altitude.
- Placard limit (maximum permissible Mach number) as a function of altitude.
- Angle of attack as a function of Mach number and lift coefficient.

See Table 3 for the format of an aerodynamic table using the above-mentioned functions. Figure 35 shows the actual data deck for the F-104 aircraft.
Table 3

FORMATS FOR AERODYNAMIC TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>No. of Cards</th>
<th>Card No.</th>
<th>Quantity</th>
<th>Format</th>
<th>No. of Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{L\text{max}} )</td>
<td>3</td>
<td>1-3</td>
<td>Mach number</td>
<td>18F4.3</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4-6</td>
<td>( C_{L\text{max}} )</td>
<td>18F4.3</td>
<td>42</td>
</tr>
<tr>
<td>( C_D )</td>
<td>1</td>
<td>7</td>
<td>Mach number</td>
<td>18F4.3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8-9</td>
<td>( C_L )</td>
<td>18F4.3</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>10-41</td>
<td>( C_D )</td>
<td>18F4.4</td>
<td>32×16</td>
</tr>
<tr>
<td>( \Delta C_D )</td>
<td>9</td>
<td>42-50</td>
<td>( \Delta C_D )</td>
<td>16F4.4</td>
<td>9×16</td>
</tr>
<tr>
<td>ABT(^a) and fuel flow</td>
<td>1</td>
<td>51</td>
<td>Mach number</td>
<td>18F4.3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>52-53</td>
<td>Altitude</td>
<td>14F5.0</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>54-85</td>
<td>ABT</td>
<td>14F5.0</td>
<td>16×16</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>86-117</td>
<td>Fuel Flow</td>
<td>14F5.0</td>
<td>16×16</td>
</tr>
<tr>
<td>Military thrust and fuel flow</td>
<td>1</td>
<td>118</td>
<td>Mach number</td>
<td>18F4.3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>119-120</td>
<td>Altitude</td>
<td>14F5.0</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>121-152</td>
<td>Thrust (military)</td>
<td>14F5.0</td>
<td>16×16</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>153-184</td>
<td>Fuel Flow</td>
<td>14F5.0</td>
<td>16×16</td>
</tr>
<tr>
<td>Placard limit</td>
<td>1</td>
<td>185</td>
<td>Altitude</td>
<td>14F5.4</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>186</td>
<td>Mach number</td>
<td>14F5.4</td>
<td>14</td>
</tr>
<tr>
<td>Angle of attack</td>
<td>2</td>
<td>187-188</td>
<td>Mach number</td>
<td>12F6.2</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>189-190</td>
<td>( C_L )</td>
<td>12F6.2</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>191-234</td>
<td>Alpha</td>
<td>12F6.2</td>
<td>13×22</td>
</tr>
<tr>
<td></td>
<td>211</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^a\)Afterburner thrust.
### F-104 TABLES FOR AERODYNAMICS

<table>
<thead>
<tr>
<th>MACH</th>
<th>COLIFTMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>500 600 700 800 900 980 990 990 970 1000 1050 1100 1150 1200 1250 1300 1330 1400 1450 1500 1550 1600</td>
</tr>
<tr>
<td>1700</td>
<td>1800 1900 2000</td>
</tr>
<tr>
<td>735</td>
<td>737 741 748 766 794 882 1010 1051 1068 938 911 902 886 840 786 738 697</td>
</tr>
<tr>
<td>694</td>
<td>639 612 591 574</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MACH, COLIFT, CODRAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 800 850 900 925 950 975 1000 1050 1100 1150 1200 1250 1300 1330 1400 1450 1500 1550 1600 1700 1800 1900 2000</td>
</tr>
<tr>
<td>000 050 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000 1050 1100</td>
</tr>
<tr>
<td>161 166 175 197 228 256 310 372 406 460 569 941 1315 1726 2082 2424 2628 2866</td>
</tr>
<tr>
<td>161 166 175 197 228 256 310 372 406 460 569 941 1315 1726 2082 2424 2628 2866</td>
</tr>
<tr>
<td>165 170 179 199 229 257 314 382 473 684 958 1337 1757 2122 2513 3293</td>
</tr>
<tr>
<td>170 176 183 205 234 278 336 405 497 719 989 1375 1814 2204 2628 2830 3087</td>
</tr>
<tr>
<td>191 193 199 218 251 297 363 429 517 751 1022 1421 1859 2258 2693 3164</td>
</tr>
<tr>
<td>230 231 243 268 306 354 417 490 576 800 1081 1482 1938 2383 2783 3258</td>
</tr>
<tr>
<td>309 322 338 357 382 428 489 563 656 881 1169 1576 2035 2400 2878 3350</td>
</tr>
<tr>
<td>379 391 403 420 452 508 576 651 735 981 1272 1692 1712 1725 1993 2673</td>
</tr>
<tr>
<td>469 472 485 506 554 603 672 757 868 1117 1403 1833 2343 2809 3326 3895</td>
</tr>
<tr>
<td>500 504 516 550 598 657 731 821 919 1172 1481 1937 2469 2950 3481 4081</td>
</tr>
<tr>
<td>500 508 522 552 599 662 737 830 941 1206 1542 2029 2591 3093 3645 4244</td>
</tr>
<tr>
<td>491 500 518 552 602 665 752 848 958 1236 1581 2050 2693 3238 3800 4441</td>
</tr>
<tr>
<td>449 457 478 511 572 644 741 852 982 1287 1645 2178 2807 3379 4011 4705</td>
</tr>
<tr>
<td>407 415 440 481 543 627 739 857 998 1327 1693 2273 2988 3684 4415 5525</td>
</tr>
<tr>
<td>380 395 419 470 537 630 747 874 1028 1388 1787 2421 3204 3934 4769 5695</td>
</tr>
<tr>
<td>369 370 394 450 552 678 800 980 1140 1600 2100 3063 4400 5740 7320 9140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DELTA CODRAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 27 29 32 35 39 45 51 51 46 43 30 30 30 30 30 30 30 30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MACH, ALTITUDE, THRUST, (AFTER-BURNER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 35 — F-104 aerodynamic tables</td>
</tr>
</tbody>
</table>

Continued
### Table

<table>
<thead>
<tr>
<th>Mach</th>
<th>Altitude</th>
<th>Fuel Flow (Afterburner)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>3000</td>
<td>5200 800 9001100011001200130014001500160017001800190020002100</td>
</tr>
<tr>
<td>0.60</td>
<td>1200</td>
<td>6250 7250 8250 9250 10250</td>
</tr>
<tr>
<td>0.70</td>
<td>1500</td>
<td>1800 1000 1800 1000 1800 1000</td>
</tr>
<tr>
<td>0.80</td>
<td>2100</td>
<td>1500 2200 1900 2100 1800 2000</td>
</tr>
<tr>
<td>0.90</td>
<td>2700</td>
<td>1300 2000 1700 1900 1600 1800</td>
</tr>
<tr>
<td>1.00</td>
<td>3300</td>
<td>1100 1800 1500 1700 1400 1600</td>
</tr>
</tbody>
</table>

---

**Fig. 35 (continued)**
<table>
<thead>
<tr>
<th>Mach</th>
<th>Altitude</th>
<th>Thrust (Military)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>250.00</td>
<td>400.00</td>
</tr>
<tr>
<td>0.20</td>
<td>400.00</td>
<td>500.00</td>
</tr>
<tr>
<td>0.25</td>
<td>500.00</td>
<td>600.00</td>
</tr>
<tr>
<td>0.30</td>
<td>600.00</td>
<td>700.00</td>
</tr>
<tr>
<td>0.35</td>
<td>700.00</td>
<td>800.00</td>
</tr>
<tr>
<td>0.40</td>
<td>800.00</td>
<td>900.00</td>
</tr>
</tbody>
</table>

Continued
C ALTITUDE, MACHMAX (PLACARE LIMIT)

C Mach, CLlift, Alpha

0.0 0.0 1.2 1.3 1.4
0.4 0.8 0.9 1.0
1.1
9.0 2.0 3.8 5.6 7.4 9.2 11.0 13.0 15.3 17.9
9.0 2.0 3.8 5.6 7.4 9.2 11.0 13.0 15.3 17.9
9.0 2.0 3.7 5.4 7.2 9.0 10.8 12.9 15.3 18.0
8.2 1.8 3.5 5.0 6.7 8.5 10.3 12.6 15.2 17.1
7.4 1.6 3.0 4.4 5.8 7.4 9.3 11.6 15.0 19.5
6.6 1.4 2.8 4.0 5.2 6.4 7.8 9.6 11.9 14.7 18.0
5.8 1.7 2.8 4.1 5.4 6.7 8.0 9.4 10.9 12.5 14.2
4.4 2.1 3.6 4.9 6.3 7.7 9.2 10.7 12.5 14.3 16.1
4.9 1.6 3.1 4.8 6.3 7.8 9.3 10.8 12.7 14.5 16.2
6.9 1.1 2.7 4.2 6.0 7.4 9.2 10.9 12.8 14.7 16.6
7.0 1.0 2.8 4.5 6.2 8.0 9.7 11.6 13.5 15.5 17.6
7.8 1.2 3.0 4.8 6.6 8.5 10.4 12.4 14.5 16.6
8.6 1.4 3.3 5.3 7.2 9.2 11.2 13.3 15.6 17.8

Continued

Fig. 35 (continued)
- 1.9 - 0.9 1.9 3.6 5.7 7.8 9.9 12.1 14.4 16.7
- 9.2 - 0.4 1.8 4.0 6.2 8.5 10.7 13.0 15.4 17.8
- 9.5 - 0.3 2.0 4.3 6.8 9.1 11.6 13.9 16.6 19.7
- 9.8 - 0.2 2.2 4.7 7.3 9.8 12.5 14.9 17.8 21.2
- 10.0 0 2.9 5.0 7.9 10.5 13.4 16.0 19.0 22.4
- 10.7 0.1 2.8 5.4 8.4 11.3 14.4 17.0 19.1
- 11.0 0.2 3.0 5.6 9.0 12.0 15.4 18.1 20.1
- 11.6 0.4 3.4 6.2 9.6 12.8 16.4 19.2 21.2

Fig. 35 (continued)
Appendix G

PROGRAM SUBROUTINES

The subroutines in Table 4, which constitute the main body of TACTICS, must always be used when running the program; all other subroutines are optional.

Table 4
TACTICS SUBROUTINES

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>AERÖDN</td>
<td>Computes aerodynamic variables and outputs change in velocity</td>
</tr>
<tr>
<td>AIM</td>
<td>Aims fighter to obtain initial-condition values for V(I,5) and V(I,6) angles</td>
</tr>
<tr>
<td>ATMÖS</td>
<td>Computes model atmosphere</td>
</tr>
<tr>
<td>ATTITUD</td>
<td>Computes attitude angles for the vehicles</td>
</tr>
<tr>
<td>AUXCÖM</td>
<td>Calls output at specified times</td>
</tr>
<tr>
<td>CÖORD</td>
<td>Makes rectangular and spherical coordinate transformations</td>
</tr>
<tr>
<td>CRÖSS</td>
<td>Computes vector cross product</td>
</tr>
<tr>
<td>DAUX</td>
<td>Computes derivatives for position and velocity</td>
</tr>
<tr>
<td>DECRD</td>
<td>Reads initial-condition data</td>
</tr>
<tr>
<td>DÖT</td>
<td>Computes vector dot product</td>
</tr>
<tr>
<td>GEÖFRö</td>
<td>Computes force for geocentric integration</td>
</tr>
<tr>
<td>GEÖFCENö</td>
<td>Determines earth rotation rate and finds unit geometric vectors</td>
</tr>
<tr>
<td>INCÖND</td>
<td>Reads, computes, and prints initial conditions</td>
</tr>
<tr>
<td>INITS</td>
<td>Initializes conditions for integration</td>
</tr>
<tr>
<td>INTGRT</td>
<td>Calls integration and computes new velocities and ranges</td>
</tr>
<tr>
<td>LAG</td>
<td>Imposes time lag (see Section V)</td>
</tr>
</tbody>
</table>

*a If storage is a problem, dummy decks may be substituted for these subroutines, since they are only used when special options are selected. (Since the routines in the main body refer to the special options, however, some type of deck must always be used to represent them.)
Table 4 (continued)

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMIT</td>
<td>Imposes aerodynamic and structural (lateral) acceleration constraints when applicable (see Section V)</td>
</tr>
<tr>
<td>MAG</td>
<td>Computes magnitude of a vector</td>
</tr>
<tr>
<td>MAIN</td>
<td>Orders operations</td>
</tr>
<tr>
<td>ØUTPUT</td>
<td>Prints output</td>
</tr>
<tr>
<td>PLACRD</td>
<td>Reduces throttle setting if there is a danger of exceeding maximum Mach number (placard limit)</td>
</tr>
<tr>
<td>RATES</td>
<td>Computes angular rates</td>
</tr>
<tr>
<td>STORE(^a)</td>
<td>Stores position and velocity when using restore option</td>
</tr>
<tr>
<td>TABINT</td>
<td>Interpolates aerodynamic tables</td>
</tr>
<tr>
<td>TABLER</td>
<td>Reads aerodynamic tables if used</td>
</tr>
<tr>
<td>THRST</td>
<td>Computes thrust of velocities</td>
</tr>
<tr>
<td>TOPCON(^a)</td>
<td>Makes local and inertial coordinate transformations</td>
</tr>
<tr>
<td>WEIGH</td>
<td>Computes weight of vehicles</td>
</tr>
</tbody>
</table>

\(^a\)If storage is a problem, dummy decks may be substituted for these subroutines, since they are only used when special options are selected. (Since the routines in the main body refer to the special options, however, some type of deck must always be used to represent them.)
Appendix H

FORTRAN FLAGS

As is usually the case with large computer programs, TACTICS uses many FORTRAN flags for internal control and operation. Although in most cases the user need not be concerned with their definition or functional purpose, certain key flags require explanation. Some of those flags may be set in POLICY or by an initial-condition data value to select a program option; others are important because they control or are indicators of various phases of program operation. The following is an explanatory list of those FORTRAN flags considered to be of primary importance:

IMISS: Whether program is to continue after finding closest missile approach.
IMISS = 0, the program stops after finding miss distance.
IMISS = 1, the program continues after finding missile miss; flag is set in POLICY.
IMISS = 2, the program has determined the miss distance and is ready to continue; flag is set in AUXCOM, and is normally used in POLICY as a criterion.

ISTORE: Whether position and velocity values are to be restored to those existing at launch time.
ISTORE = 0, the positions and velocities are not to be stored.*
ISTORE = 1, the store option is to be used. The values of position and velocity are stored at launch, and the program returns to these values after computing the closest point of approach; flag is set in POLICY and used in MAIN to trigger STORE routine.

*ISTORE is set at zero in STORE routine to indicate that values have been restored, and is normally used in POLICY as a criterion.
JPÔL, KPÔL, LPÔL, MPÔL, NPÔL

To be used in PÔLICY; initially set equal to 1 in INCÔND.

JINTEG: The type of integration to be used; set by DATA 122. If
integration is to be changed during a run, JINTEG may be
reset in PÔLICY.
JINTEG = 0, variable-step Adams-Moulton.
JINTEG = 1, fixed-step Runge-Kutta.
JINTEG = 2, fixed-step Adams-Moulton.
JINTEG = 3, variable-step Adams-Moulton with exact
printout.

ISTART: Used within program to determine specific events.
ISTART = 0, the start of the program (time = 0.0).
ISTART = 1, value is set after the first integration
step, and remains at this value until closest
missile approach or until the end of the problem
run; flag is set in MAIN.
ISTART = 2, the program is backing up to compute the
closest missile approach; flag is set in INTGRT.
If the program is to continue after computing
miss distance, ISTART is automatically reset to
1 in AUXCOM.

ILAUN: Used within program to indicate status of missile launching.
ILAUN = 1, the missile has not been launched; flag is
set initially in INCÔND and is reset in CAPFLT
if recall or restore options are used.
ILAUN = 2, launch criteria have been overshot because of
an excessive step size, and program is to back
up to approach launch on a smaller fixed step;
flag is set in LAUNCH (see Section XI).
ILAUN = 3, the missile has been launched; flag is set
in LAUNCH and is usually used as a criterion in
PÔLICY.
ILAUN = 4, the program has backed up to the step before launch and will approach on fixed step size integration; flag is set in INITS. This is to keep CAPFLT from resetting ILAUN to 1, which would prevent the missile derivatives from being integrated.

JATMOS: Whether the velocities of the fighter and target are to be read in ft/sec or Mach number for initial data.
JATMOS = 0 (ft/sec)
JATMOS = 1 (Mach number)

KINTEG: Whether a flat or spherical earth gravitational force representation is to be used (see Section IV and Appendix B).
KINTEG = 0, the flat-earth option is to be used.
KINTEG = 1, the round-earth option is to be used.

ICAP: This flag may serve several different functions, all related to the flight status of vehicle 2. Its primary purpose is to avoid needless computations and integration of the equations of motion for vehicle 2 when it is not being used, i.e., captive flight or undefined motion. The ICAP value should be read in as 0, 1, 2, or 3 (DATA 28).

ICAP = 0, computations and integration will occur for vehicle 2. If a nonzero value for V(2,4) (DATA 33) has been entered as an initial-condition value, the ILAUN = 3 flight status is assumed to exist. If DATA 33 is an exact zero value, the ILAUN = 1 status is assumed and computation will occur, presumably in anticipation of the ground launch of vehicle 2 at some subsequent time. If an intercept problem is involved, the target vehicle will be vehicle 3 for miss-distance computation.

ICAP = 1, vehicle 2 is in captive flight on vehicle 1; if launch occurs, it will intercept against
vehicle 3. No computations for vehicle 2 are performed prior to reaching criteria for launch.

ICAP = 2, vehicle 2 is not being used; it will automatically be located at the zero origin with zero velocity and acceleration. No computations or integration will occur for vehicle 2.

ICAP = 3, vehicle 2 is in captive flight on vehicle 3; if launch occurs, it will intercept against vehicle 1. No computations for vehicle 2 are performed prior to reaching criteria for launch.
Appendix I

MODEL ATMOSPHERE

A model atmosphere may be postulated in either tabular or analytic form depending on the degree of realism required by the problem. For most intercept problems, an exponential analytic form is sufficient, and no provision is made in TACTICS for incorporating tabular values. The air density $\rho$, pressure $p$, and absolute temperature $T$ may be approximated as a function of altitude $z$ from the following expressions, which are consistent with those given for the ICAO standard atmosphere (although constants are not represented to the same degree of significance). (4)

For the troposphere ($0 < z < z_s$ (ft))

$$t = 59 - 0.00357 \, z \, ^{(°F)}$$

$$p = p_s \left(1 - \frac{0.00357 \, z}{518.4}\right)^{5.256}$$ \hspace{1cm} (140)

$$\rho = 0.002378 \left(1 - \frac{0.00357 \, z}{518.4}\right)^{4.756} \text{slug/ft}^3$$

For the stratosphere ($z_s < z < 86,000$ ft)

$$t = -67 °F$$

$$p = 489.456 \, e^{-\frac{z-z_s}{h_s}} \text{lb/ft}^2$$

$$h_s = h_a \frac{T_s}{T_a} \text{ ft}$$ \hspace{1cm} (141)

$$\rho = \frac{p}{gh_s} \text{ slug/ft}^3$$
where

\[ p = \text{pressure at } z \text{ ft (lb/ft}^2) \]
\[ T = \text{absolute temperature (}^\circ\text{R}) = t + 459.4 \text{ (}^\circ\text{F}) \]
\[ t = \text{temperature at altitude } z \text{ ft (} z < z_s \text{) (}^\circ\text{F}) \]
\[ \rho = \text{air density at } z \text{ ft} \]
\[ h_a = \text{pressure head at sea level (ft)} \]
\[ p_a = \text{pressure at sea level (lb/ft}^2) \]
\[ T_a = \text{absolute temperature at sea level (}^\circ\text{R)} \]
\[ h_s = \text{pressure head at } z_s \text{ (ft)} \]
\[ p_s = \text{pressure at } z_s \text{ (calculated from Eqs. (141)) (lb/ft}^2) \]
\[ T_s = \text{absolute temperature at altitude } z_s \text{ (}^\circ\text{R)} \]
\[ z_s = \text{an altitude defining the beginning of the stratosphere and isothermal conditions (ft)} \]

The above relationships are applicable for the so-called standard atmosphere, with a temperature gradient of \(-3.57^\circ\text{F}/1000 \text{ ft}\) in the troposphere. Assuming the atmosphere can be characterized by isothermal conditions above an altitude of 35,300 ft at a temperature of \(-67^\circ\text{F}\) yields the following:

\[
\begin{align*}
  h_a &= 27,650 \text{ ft} & h_s &= 20,916 \text{ ft} \\
  p_a &= 2,116 \text{ lb/ft}^2 & p_s &= 489.456 \text{ lb/ft}^2 \\
  T_a &= 518.4 ^\circ\text{R} & T_s &= 392.4 ^\circ\text{R} \\
  t_a &= 59^\circ\text{F} & t_s &= -67^\circ\text{F} \\
  z_s &= 35,300 \text{ ft} 
\end{align*}
\]

The speed of sound, \(s\), is equal to \(\sqrt{1.4 \cdot \frac{p}{\rho}} \text{ ft/sec}\). Mach number is related to vehicle speed \(V\) by \(M = \frac{V}{s}\).
REFERENCES


