NASA TECHNICAL NOTE **NASA TN D-7829**

NASA TN D-7829

(NASA-TN-D-7829) FOUR-DIMENSIONAL GUIDANCE ALGORITHMS FOR AIRCRAFT IN AN AIR TRAFFIC CONTROL ENVIRONMENT (NASA) 140 p HC $5.75 CSCL 01B
N75-23549
Unclas
H1/04 22432

FOUR-DIMENSIONAL GUIDANCE ALGORITHMS FOR AIRCRAFT IN AN AIR TRAFFIC CONTROL ENVIRONMENT

Thomas Pecsvaradi

Ames Research Center
Moffett Field, Calif. 94035

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1975

1. Report No. NASA TN D-7829	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FOUR-DIMENSIONAL GUIDANCE ALGORITHMS FOR AIRCRAFT IN AN AIR TRAFFIC CONTROL ENVIRONMENT		5. Report Date March 1975
		6. Performing Organization Code
7. Author(s) Thomas Pecsvaradi		8. Performing Organization Report No. A-5530
9. Performing Organization Name and Address Ames Research Center Moffett Field, Calif. 94035		10. Work Unit No. 501-03-11
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Technical Note
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Detailed theoretical development and computer implementation of three guidance algorithms are presented. From a small set of input parameters the algorithms generate the ground track, altitude profile, and speed profile required to implement an experimental 4-D guidance system.

Given a sequence of waypoints that define a nominal flight path, the first algorithm generates a realistic, flyable ground track consisting of a sequence of straight line segments and circular arcs. Each circular turn is constrained by the minimum turning radius of the aircraft. The ground track and the specified waypoint altitudes are used as inputs to the second algorithm which generates the altitude profile. The altitude profile consists of piecewise constant flight path angle segments, each segment lying within specified upper and lower bounds. The third algorithm generates a feasible speed profile subject to constraints on the rate of change in speed, permissible speed ranges, and effects of wind. The flight path parameters are then combined into a chronological sequence to form the 4-D guidance vectors. These vectors can be used to drive the autopilot/autothrottle of the aircraft so that a 4-D flight path could be tracked completely automatically; or these vectors may be used to drive the flight director and other cockpit displays, thereby enabling the pilot to track a 4-D flight path manually. The algorithms have a number of unique features that aid in achieving precise arrival times: the ability to capture any one of the waypoints along the flight path, the simplicity of engaging and disengaging the tracking mode (both automatic and manual), and the provision for manual path stretching as part of the overall 4-D guidance scheme.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
4-D guidance of aircraft Automated guidance of aircraft Terminal area navigation Air traffic control	Unclassified – Unlimited STAR Category 04

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	140	$5.75

*For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

SYMBOLS

a	rate of speed change
a_{min}	minimum admissible rate of speed change (maximum deceleration)
a_{max}	maximum admissible rate of speed change (maximum acceleration)
a_k	rate of speed change during kth guidance interval
C_i	distance between points $P_i(Q_i)$ and WP_i where WP_i is an ordinary waypoint and $P_i(Q_i)$ is the beginning (end) of the turn associated with WP_i
$\underline{c}$	fraction of zero-flap stall speed indicating minimum admissible cruising airspeed in terminal area
$\bar{c}$	fraction of zero-flap stall speed indicating maximum admissible cruising airspeed in terminal area
c_1	ground speed for any given airspeed when aircraft heading is parallel to wind heading
c_2	ground speed for any given airspeed when aircraft heading is perpendicular to wind heading
CWP	index of capture waypoint ($1 \leq CWP \leq NWP$)
DB	length of straight-line segment in capture ground track
D_i	length of straight-line segment from point Q_{i-1} to point P_i
$ENRTIM_i$	flight time from point Q_i to point Q_{NWP} along 4-D flight path
f_k	kth guidance vector
g	acceleration due to gravity
GAMB	constant flight-path angle of capture flight path
GAM_i	constant flight-path angle from point Q_{i-1} to point Q_i along ground track
$INDEX_i$	waypoint type indicator, $INDEX_i = 0 \Rightarrow WP_i$ is ordinary waypoint; $INDEX_i = 1 \Rightarrow WP_i$ is final heading waypoint

MAXSEG	maximum number of nondegenerate guidance vectors needed to generate the 4-D flight path from first to last waypoint
NCI	number of nondegenerate guidance vectors needed to generate the 4-D capture flight path
NCI_i	1 + number of nondegenerate guidance vectors needed to generate the 4-D flight path from $WP_1 = Q_1$ to Q_i
O	center of circular turn
PA(PB)	beginning of first (second) turn of capture flight; (PA is initial aircraft position)
P_i	beginning of turn associated with WP_i
QA(QB)	end of first (second) turn of capture flight path
Q_i	end of turn associated with WP_i
r	generic term for turning radius
RA(RB)	turning radius of first (second) turn of capture flight path
R_i	turning radius of turn associated with WP_i
$RMIN_i$	minimum turning radius of turn associated with WP_i
SAB	distance required to change ground speed from VQA to VPB at maximum (or minimum) rate
$S11_i(\rho)$, $S12_i(\rho)$, $S22_i(\rho)$	straight-line distances flown during the three subsegments of the speed profile from point Q_{i-1} to point P_i corresponding to a speed level ρ
t	generic term for time
TABS	current value of absolute time
TCAP	flight time along 4-D capture flight path
$TMIN_i$ ($TMAX_i$)	minimum (maximum) attainable flight time from point Q_i to point Q_{NWP} along 3-D flight path
T1, T2, T3, T4	time duration of the four guidance intervals for the 4-D capture flight path
$T11_i(\rho)$, $T12_i(\rho)$, $T22_i(\rho)$, $T23_i(\rho)$	time duration of the four guidance intervals from point Q_{i-1} to point Q_i along the 4-D flight path corresponding to a speed level ρ

$T_i(\rho)$	flight time from point Q_{i-1} to point Q_i along the 4-D flight path corresponding to a speed level ρ
$T(\rho)$	total flight time from point Q_{CWP} to point Q_{NWP} along the 4-D flight path corresponding to a speed level ρ
u	inverse of turning radius, (= 1/r)
u_k	inverse of turning radius during kth guidance interval
v	generic term for speed
$\dot{v}$	time derivative of speed
$\underline{v}$	minimum admissible cruising airspeed in terminal area
$\bar{v}$	maximum admissible cruising airspeed in terminal area
VA(VB)	constant airspeed during first (second) turn of capture flight path
$VA_i, VA_i(\rho)$	constant airspeed during turn associated with WP_i
V_{final}	desired airspeed at last waypoint
V_{AC}	initial airspeed of aircraft
VG_i	ground speed at WP_i from which a ground speed $VGMAX_{i+1}$ at point P_{i+1} can be achieved by flying from WP_i directly to P_{i+1} and using a deceleration a_{min}
$VGMAX_i$	maximum possible ground speed of aircraft at WP_i
$vmin_i(vmax_i)$	airspeed at Q_i from which an airspeed $VMIN_{i+1}$ $(VMAX_{i+1})$ at point P_{i+1} can be achieved by flying from Q_i directly to P_{i+1} and using a deceleration a_{min}
$v_{min}(v_{max})$	generic term for minimum (maximum) admissible speed
$VMIN_i(VMAX_i)$	minimum (maximum) admissible airspeed at WP_i
VPB(VQA)	ground speed at beginning (end) of second (first) turn of capture flight path
$VP_i(\rho),(VQ_i(\rho))$	ground speed at beginning (end) of turn associated with WP_i corresponding to a speed level ρ
$V_p(\delta)$	flap placard speed with flaps set at $\delta°$
$V_s(\delta)$	stall speed with flaps set at $\delta°$

V_w	magnitude of wind
WP_i	waypoint i
x,y,z	generic terms for Cartesian coordinates of aircraft
$\dot{x},\dot{y},\dot{z}$	time derivatives of x,y,z
X_{AC},Y_{AC},Z_{AC}	Cartesian coordinates of initial position of aircraft
XO,YO	Cartesian coordinates of center of turn
XP_i,YP_i,ZP_i	Cartesian coordinates of beginning of turn associated with WP_i
XQ_i,YQ_i,ZQ_i	Cartesian coordinates of end of turn associated with WP_i
XPA,YPA,ZPA	Cartesian coordinates of beginning of first turn in capture flight path (same as X_{AC},Y_{AC},Z_{AC})
XQA,YQA,ZQA	Cartesian coordinates of end of first turn in capture flight path
XPB,YPB,ZPB	Cartesian coordinates of beginning of second turn in capture flight path
XQB,YQB,ZQB	Cartesian coordinates of end of second turn in capture flight path (same as $XQ_{CWP},YQ_{CWP},ZQ_{CWP}$)
XWP_i,YWP_i,ZWP_i	Cartesian coordinates of WP_i
z_r	reference altitude used to compute lead time for finite pitch-rate compensation
$\ddot{z}_{max}$	maximum admissible vertical acceleration
$\alpha_1,\alpha_2,\alpha_3$	angles used to compute a turn by subroutine NEWPSI
γ	generic term for flight-path angle
$\dot{\gamma}_{max}$	maximum admissible pitch rate
$\gamma_{min}(\gamma_{max})$	minimum (maximum) admissible flight-path angle
γ_k	flight-path angle during kth guidance interval
δ	flap setting
$\Delta DA(\Delta DB)$	arclength of first (second) turn in capture flight path
ΔD_i	arclength of turn associated with WP_i

Δt_k	time duration of kth guidance interval
ΔTOA	desired change in time of arrival
$\Delta \Psi A (\Delta \Psi B)$	angular extent of first (second) turn in capture flight path
$\Delta \psi_i$	angular extent of turn associated with WP_i
ε	arbitrarily small positive number used to stop an iterative procedure or to determine the nondegenerate guidance vectors
ϕ	generic term for bank angle
ϕ_{max}	maximum admissible bank angle
$\dot{\phi}$	time derivative of ϕ
$\dot{\phi}_{max}$	maximum admissible roll rate
ψ	generic term for heading
$\dot{\psi}$	time derivative of ψ
ψ_i	heading of straight line from point Q_{i-1} to point P_i
ψ_{final}	desired heading at final waypoint
ψ_w	wind direction
ψ_{AC}	initial heading of aircraft
$\Psi A (\Psi B)$	heading at beginning of first (second) turn of capture flight path
ΨC	heading at end of second turn of capture flight path
ψ_r	reference heading used to compute lead time for finite roll-rate compensation
ρ	fraction indicating speed profile deviation from maximum admissible speed profile
τ_ϕ	lead time for finite roll-rate compensation
τ_γ	lead time for finite pitch-rate compensation

FOUR-DIMENSIONAL GUIDANCE ALGORITHMS FOR AIRCRAFT IN AN AIR TRAFFIC CONTROL ENVIRONMENT

Thomas Pecsvaradi

Ames Research Center

SUMMARY

This report describes the detailed theoretical development of three guidance algorithms that form a part of an experimental 4-D guidance system. The algorithms generate the ground track, altitude profile, and speed profile required to implement the 4-D guidance concept.

The ground track algorithm computes a realistic, flyable ground track consisting of a sequence of straight line segments and circular arcs. Each circular turn is constrained to have a radius at least as large as the minimum turning radius of the aircraft. The altitude profile algorithm generates an altitude profile along the previously computed ground track. The altitude profile consists of a sequence of piecewise constant flight-path angle segments, each segment lying within specified upper and lower bounds. Finally, to determine the motion of the aircraft as a function of time along the 3-D flight path, the speed profile algorithm generates a feasible speed profile subject to constraints on the rate of speed change, permissible speed ranges, and the effects of wind. The various flight path parameters are then combined into a chronological sequence to form the 4-D guidance vectors. These vectors can be used to drive the autopilot/autothrottle of the aircraft, so that a 4-D flight-path can be tracked completely automatically; or these vectors may be used to drive the flight director and other cockpit displays, thereby enabling the pilot to track a 4-D flight path manually. The system has a number of unique features that aid in achieving precise arrival times: the ability to capture any one of the waypoints along the flight path, the ease with which the tracking mode (both automatic and manual) can be engaged and disengaged, and the provision for manual path stretching as part of the overall 4-D guidance scheme.

The three guidance algorithms have been implemented in the form of a set of computer programs and these programs have been tested extensively in a static environment on the IBM 360/70 computer at Ames Research Center.

INTRODUCTION

Accurate terminal area guidance and control can be achieved with a system concept referred to as 4-D guidance. During the last several years at Ames Research Center, effort has been devoted to the design, development, and implementation of an experimental 4-D guidance system. The system uses a technique to guide and control an aircraft not only in the three spatial dimensions, but in time as well. The aim of the technique is to provide a flexible guidance scheme that can be used to predict and control accurately the motion of an aircraft as a function of time. It must be emphasized that the technique by itself is not sufficient to solve the complete multi-aircraft terminal area guidance problem, for which many aircraft must be scheduled and sequenced into common paths while minimum separation standards are assured. The problem of aircraft scheduling has been discussed by McLean and Tobias (ref. 1), who developed an interactive real time system for scheduling and monitoring many aircraft in the terminal area. An overall terminal area guidance system that uses both the 4-D algorithms presented here and the scheduling algorithms described in reference 1 appears to have a great potential for increasing terminal area effectiveness.

Research in 4-D guidance is currently under way by the Department of Transportation, universities, and NASA. However, publications on alternate approaches are not generally available. One possible alternate approach that has been published is path stretching in a defined terminal area corridor (ref. 2).

This report describes the detailed theoretical development and computer implementation of those algorithms that generate the 4-D reference flight path. Specifically, three main guidance algorithms are developed: the first generates the ground track of the reference flight path, the second calculates the altitude profile, and the third computes the desired speed profile. A feasible method of combining the three algorithms into an on-line 4-D guidance system is also described. Certain general aspects of some of these algorithms are extensions of earlier work done at Ames; the particular techniques used to generate the ground track and speed profile, however, represent new contributions. The design of the algorithms is modular so that many of the sub-algorithms can be used independently or can be replaced with a minimum effect on the overall system.

The algorithms are implemented in the form of a set of computer programs. These programs were written in FORTRAN IV and were tested on the IBM 360/70 at Ames. Detailed programming descriptions of all subroutines, including flow charts and source listings, are given in the appendix.

PROBLEM DESCRIPTION AND BASIC APPROACH

The fundamental problem facing controllers and pilots in the terminal area is the safe and efficient guidance and control of aircraft from various points in the terminal area to the runway. (The problem of safe and efficient guidance of departing aircraft is just as important; however, since the basic guidance tasks in the two cases are quite similar, it is sufficient to consider the guidance tasks associated with approaching aircraft in developing a theoretical framework.) This task consists of two separate subtasks:
(i) selecting a 3-D flight path the aircraft is to follow and a time schedule to be maintained along the flight path, and (ii) controlling the aircraft along

the 4-D flight profile determined in (i). In the current air traffic control (ATC) system, subtask (i) is generally considered to be the responsibility of the controllers. Selecting terminal area flight paths is made considerably simpler by the existence of Standard Terminal Arrival Routes (STARS) and Standard Instrument Departure (SID) routes. Portions of these routes, combined with certain less well-defined paths determined at the time of flight and transmitted to the pilot in the form of a series of verbal instructions (vectoring), comprise the majority of terminal area flight paths. Thus, selecting three-dimensional terminal area flight paths is reasonably straightforward. The determination, in advance, of a desired time schedule along such paths, on the other hand, is much more difficult since it requires knowledge of the ground speed of the aircraft as a function of time, which, in turn, must be integrated to give position information. Clearly, accomplishing such a task manually is very difficult, especially in a multi-aircraft environment.

Tracking a predetermined four-dimensional flight profile is, of course, the responsibility of the pilot. This task can be carried out either manually or automatically, using the autopilot/autothrottle of the aircraft. One of the major shortcomings of current autopilots is, however, their limited capability of tracking complex, four-dimensional flight profiles. Although most autopilots can perform such single functions as altitude hold, heading hold, flight-path-angle hold, etc., they are not designed to perform a sequence of such operations automatically. Similar comments apply to the autothrottle. This report describes the development of a guidance technique that is sufficiently flexible to permit manual or automatic tracking of complex four-dimensional flight paths.

Most flight paths of aircraft can be modeled as sequences of straight-line flights and coordinated turns, executed while flying at piecewise constant flight-path angles. Speed changes occur at relatively few points along the flight path and are usually accomplished at a nearly linear rate. The principal control signals used to maneuver aircraft along such typical flight paths are bank angle, flight-path angle, and acceleration of the aircraft. It is assumed throughout the report that the flight-path angle is small, the centrifugal force due to the turn is balanced by the horizontal component of the lift, and the weight of the aircraft is balanced by the vertical component of the lift. It is further assumed that the speed of the aircraft is directly controllable by an acceleration input. Under these assumptions, no sideslip occurs and the point-mass equations of motion of an aircraft can be written as

$$\dot{x} = v \cos \psi \tag{1}$$

$$\dot{y} = v \sin \psi \tag{2}$$

$$\dot{z} = v\gamma \tag{3}$$

$$\dot{\psi} = \frac{g}{v} \tan \phi \tag{4}$$

$$\dot{v} = a \tag{5}$$

where x, y, and z are the Cartesian coordinates of the aircraft position in the usual z-downward-positive coordinate system, ψ is the heading angle measured clockwise from the positive x-axis, v is the speed, ϕ is the bank angle, γ is the flight-path angle, a is the acceleration of the aircraft, and g is the acceleration due to gravity. Positive ϕ, by convention, means right wing down, which corresponds to a right turn; similarly, positive γ means ascending flight. The quantities x, y, z, ψ, v are considered to be the state of the aircraft, while ϕ, γ, and a are the controls.

In order to assure passenger comfort and to remain within the safe operational characteristics of the aircraft, it is necessary to impose constraints on the control variables. Accordingly, ϕ, γ, and a are assumed to satisfy the following constraints:

$$|\phi(t)| \leq \phi_{max} \tag{6}$$

$$\gamma_{min} \leq \gamma(t) \leq \gamma_{max} \tag{7}$$

$$a_{min} \leq a(t) \leq a_{max} \tag{8}$$

These bounds, of course, depend a great deal on the type of aircraft considered. For a typical jet transport aircraft, extreme values of these bounds may be $\phi_{max} = 30°$, $\gamma_{min} = -3°$, $\gamma_{max} = 8°$, $a_{min} = -0.6$ m/sec^2, $a_{max} = 0.6$ m/sec^2; for STOL aircraft, γ_{min} may be as low as $-8°$.

In addition, bounds must be placed on the admissible speeds of the aircraft. The upper bound is generally dictated by either Federal Aviation Regulations or by structural considerations, while the lower bound depends basically on the stall characteristics of the airplane. A more detailed discussion of the speed constraints is given under Speed Profile. For the present general discussion, it is simply assumed that the speed of the aircraft satisfies constraints of the form

$$v_{min} \leq v(t) \leq v_{max} \tag{9}$$

It is convenient at this point to introduce a change of variables in equation (4). When the ground track of an aircraft is considered, the instantaneous radius of curvature, r, of the flight path is defined by (see ref. 3)

$$r = \frac{v}{\dot{\psi}} = \frac{v^2}{g \tan \phi} \tag{10}$$

Thus, the radius of curvature is closely related to the bank angle. For purposes of analysis, it is convenient to consider r, rather than ϕ, as the principal control variable used for horizontal maneuvers. Actually, to avoid infinite values during the control process, a more appropriate control variable, u, is defined as follows

$$u = \frac{1}{r} = \frac{g \tan \phi}{v^2} \tag{11}$$

In terms of the control variable u, equation (4) can be rewritten as

$$\dot{\psi} = vu \tag{4'}$$

Since ϕ and v are constrained by inequalities (6) and (9), respectively, the admissible values of u are bounded from above and below. Thus, given any instantaneous speed, say $v(t)$, along the flight path, the instantaneous value of $u(t)$ must satisfy the inequality

$$|u(t)| \le \frac{g \tan \phi_{max}}{v^2(t)} \tag{12}$$

It can now be seen why u is a particularly convenient control variable for accurate aircraft guidance: a constant nonzero u yields a ground track with constant radius of turn, that is, a circular flight segment. Furthermore, if $u \equiv 0$, then the ground track of the resulting flight is a straight line. Hence, if piecewise constant values are used for u, simple yet realistic ground tracks can be generated. Piecewise constant values of u can be obtained by appropriately varying $\phi(t)$ according to equation (11).

While inequalities (6) through (8) constrain the admissible magnitudes of the control variables ϕ, γ, and a, the rates at which these variables are allowed to change are not limited. It is recognized that, in reality, these controls can change only at finite rates. It is felt, however, that the assumption of infinite control rates is not at all unreasonable for a guidance technique whose purpose is to generate four-dimensional terminal-area flight paths for periods of time that can be several orders of magnitude greater than the time constants of the controls. Furthermore, the inclusion of rate constraints on the controls would result in a much more complicated problem and would make the implementation of the results much more difficult. In view of these remarks, the control rates are assumed to be unlimited. Under Four-Dimensional Guidance Commands, a simple but effective way of accounting for finite roll and pitch rates is described.

Before stating the precise problem to which this report is addressed, it is instructive to look at the states and controls that correspond to a typical terminal area flight path. Figure 1 shows the ground track of an aircraft from an arbitrary point A to the touchdown point K. The ground track consists of a series of connected straight lines and circular arcs. The altitude, speed, and heading profiles corresponding to this ground track are shown in figure 2; figure 3 illustrates the controls required for the flight path depicted in figures 1 and 2. At point A, the aircraft is flying straight and level but begins to decelerate to a lower speed. This lower speed is achieved at point B, so that the flight from B to C is straight and level at constant speed. At point C, the aircraft begins to descend to a lower altitude at some constant, negative flight-path angle. While continuing the descent, at D it begins a left turn, capturing the new required heading at point E. From E to F, the flight is straight at constant speed but is still descending. At

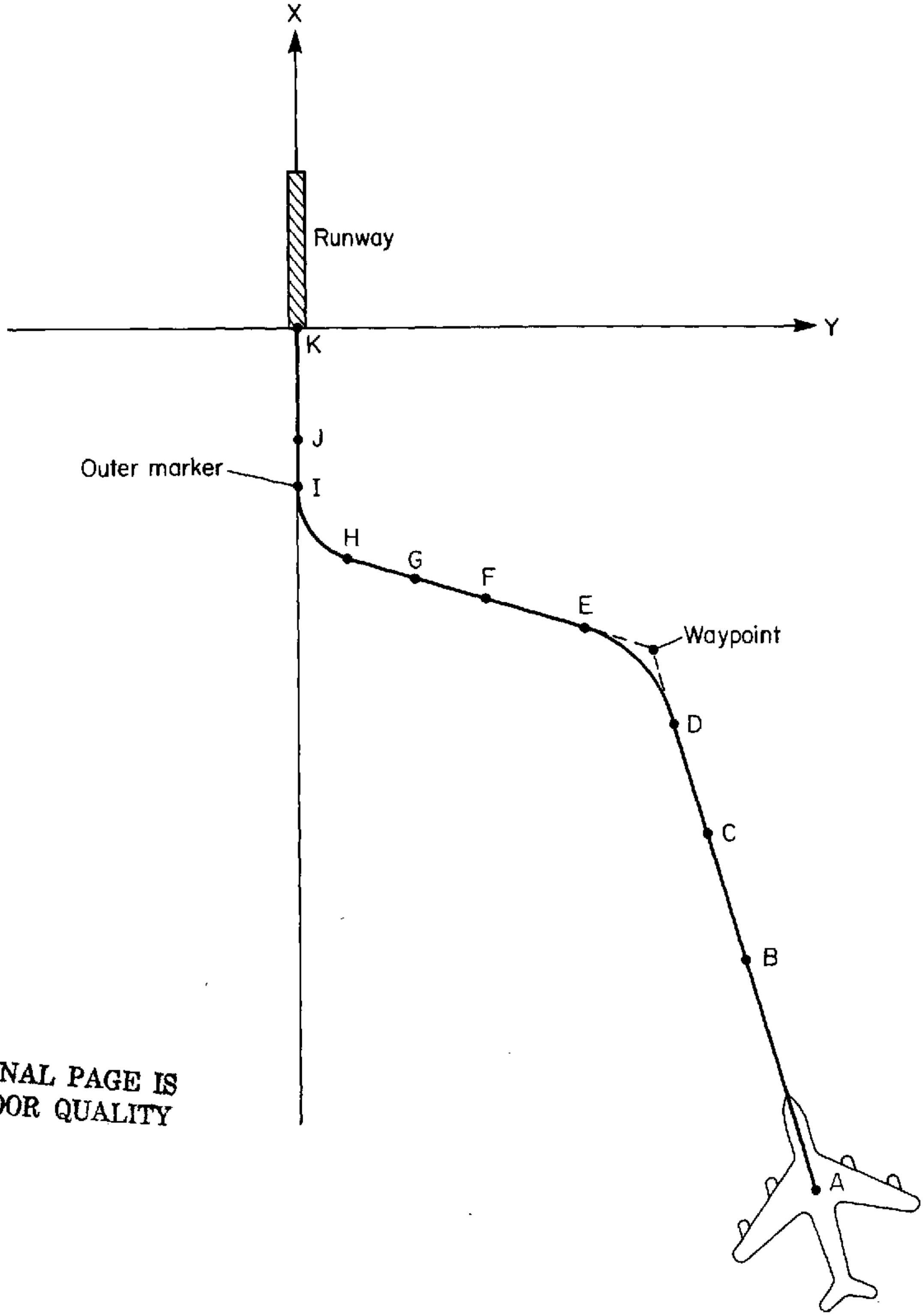


Figure 1.– Ground track of aircraft in terminal area.

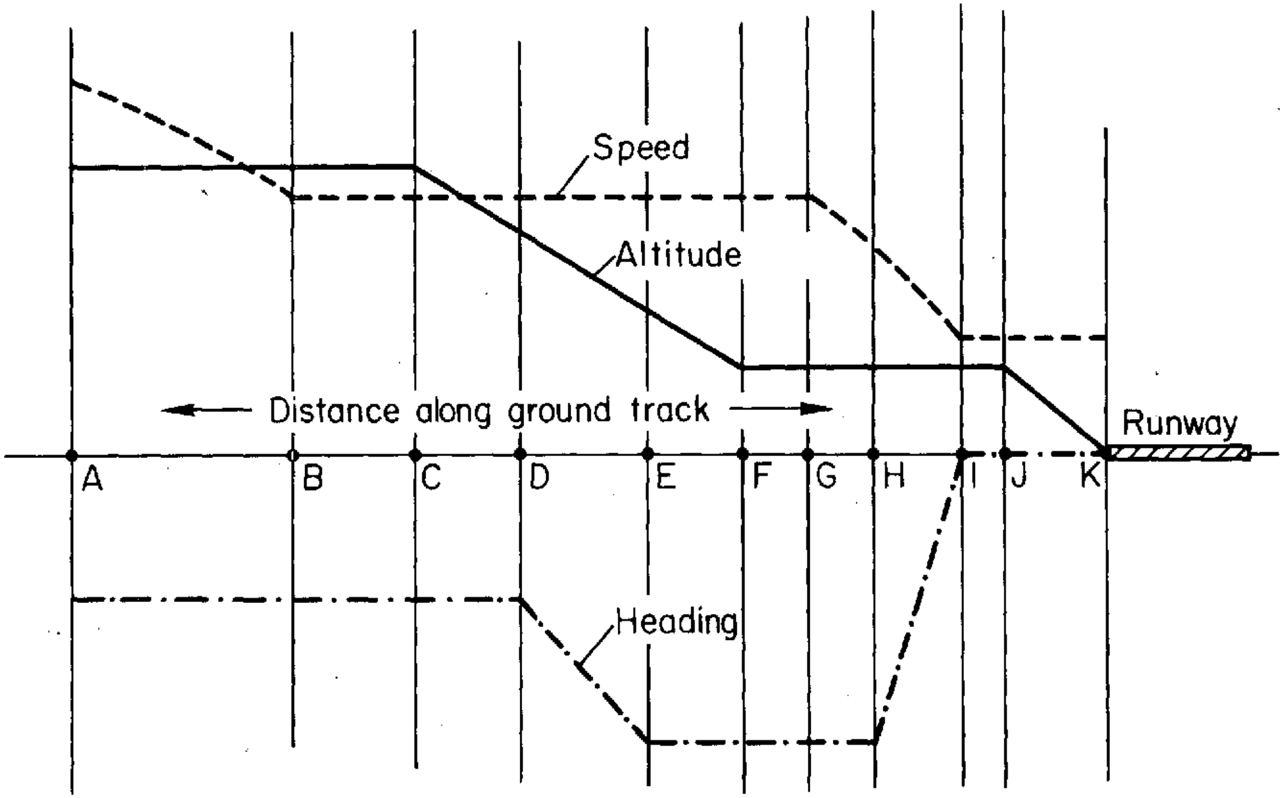


Figure 2.— Altitude, speed, and heading of aircraft in terminal area.

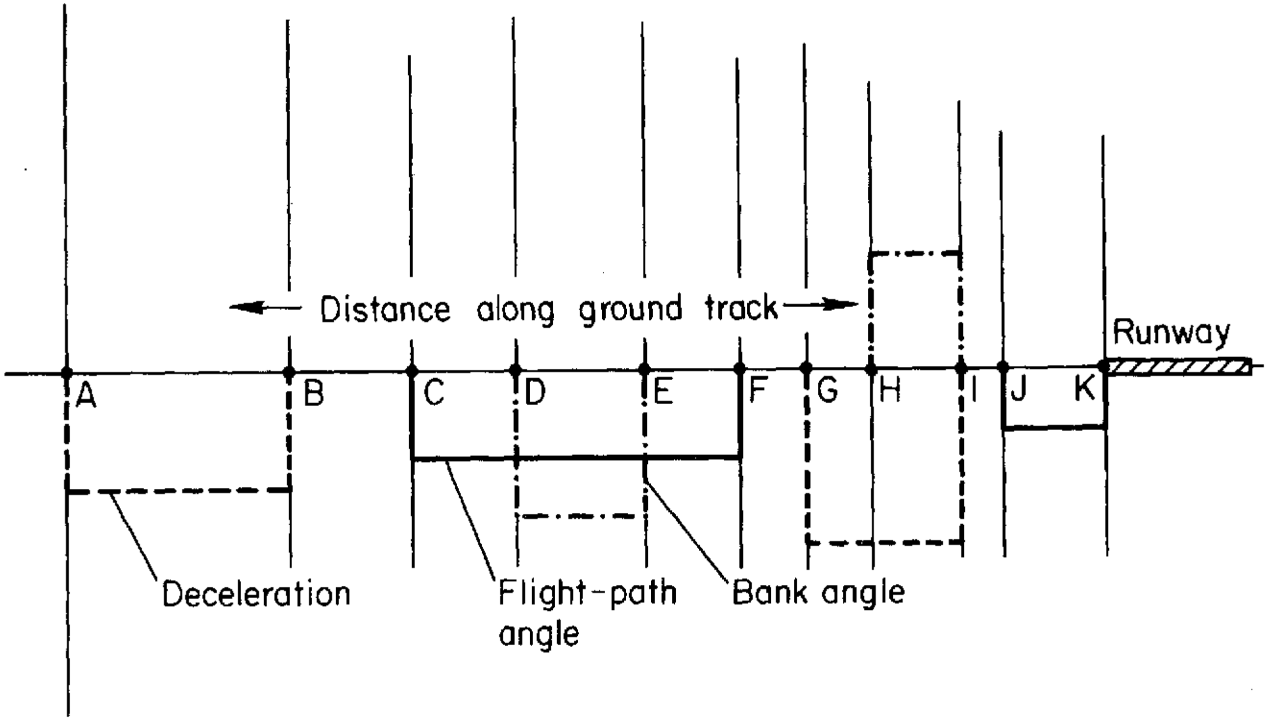


Figure 3.— Control signals for flight profile of figures 1 and 2.

F, the required altitude is achieved, so the aircraft levels off, maintaining a straight and level flight at constant speed between F and G. At point G, the aircraft begins to decelerate to the final approach speed, while at point H it begins a right turn, all at level flight. At point I, the runway centerline is intercepted and the final approach speed is achieved simultaneously. At this point, the aircraft begins its straight-in approach. At point J, transitioning occurs from level flight to capture the glide slope, and the approach is continued until touchdown at point K.

In the current ATC system, the above guidance process is basically a manual one: the controller transmits the commands to the pilot via radiotelephone, and the pilot then executes these commands by appropriately maneuvering the aircraft. Two major limiting factors in such a manual guidance-control loop are the great deal of communications required and the lack of accuracy with which arrival times to the runway can be predicted. However, both limitations can be reduced substantially by use of airborne digital computers, sophisticated cockpit displays, and a digital data-link between ground and aircraft. These considerations are incorporated in the development of the guidance scheme presented here.

The point of view adopted in this research was that the fundamental problem of aircraft guidance is to generate a time sequence of guidance commands corresponding to a desired four-dimensional flight path. That is, if a set of realistic guidance commands could be identified for any terminal area flight path, then the commands could be either coupled to the autopilot/autothrottle of the aircraft for automatic tracking, or they could be presented on appropriate displays to the pilot for manual tracking. Since the problems associated with tracking a sequence of guidance commands have been treated elsewhere (see, e.g., refs. 4 and 5), they are not discussed here.

The first question to be resolved is: what is a minimum set of simple inputs from which realistic guidance commands could be synthesized by an efficient algorithm? To answer this question it is necessary to determine those attributes of a 4-D flight path which are essential to describe it uniquely. Consider the three spatial dimensions, that is, the 3-D path. As far as the horizontal projection of typical flight paths is concerned, aircraft generally navigate from point to point (referred to as waypoints) in straight-line segments (fig. 1). At or near these waypoints, coordinated turns are executed to capture the new heading of the next straight flight segment. Thus, the ground track of an arbitrary flight path could be simply characterized by the x,y coordinates of these waypoints in some coordinate system. Since aircraft cannot change heading instantaneously, feasible turning radii should be specified for the turns at the waypoints. A sequence of x,y coordinates and turning radii alone, however, does not yield a unique ground track and therefore additional logic is required. As shown in the next section, two simple classes of waypoints can be defined so that a unique ground track is obtained by specifying the following four input quantities for each waypoint: x,y coordinates, turning radius, and a binary variable that denotes the class to which the waypoint belongs.

As discussed previously, the motion of aircraft in the vertical dimension can be considered a sequence of constant altitude and constant flight-path-angle segments (fig. 2). Therefore, arbitrary altitude profiles can be uniquely determined by specifying the altitudes of those points along the flight path at which the flight-path angle changes from one constant value to another. To synthesize altitude profiles, it is assumed that the flight-path angle can change from one constant value to another instantaneously. A simple method of accounting for finite pitch rates is discussed under Four-Dimensional Guidance Commands.

Having specified a complete 3-D flight path in the manner described above, it now remains to characterize the motion of an aircraft as a function of time along the prespecified 3-D path. This is equivalent to characterizing the velocity profile along the flight path. Due to operational constraints, aircraft performance limitations, and the desire to control accurately the arrival times of aircraft, the problem of specifying feasible velocity profiles is difficult. It is further complicated when velocity profiles are to be specified in the presence of wind. It is shown, however, under Speed Profile, that judicious approximations can be made and realistic speed profiles can be synthesized from a small number of input parameters.

The flow of computation in the 4-D flight-path synthesis is sequential and is illustrated by the block diagram in fig. 4. The first step is to generate the ground track from the given inputs. The ground track is then used together with additional input parameters to compute a feasible altitude profile. At this point, a 3-D path and the corresponding 3-D guidance commands are determined. The last step is to generate a feasible speed profile that satisfies the arrival time requirements. A detailed description of the ground track synthesis is given in the following section, the altitude profile generation is described under Altitude Profile, and the speed profile generating algorithm is discussed under Speed Profile.

GROUND TRACK SYNTHESIS

The ground track of a typical terminal area flight path consists of an alternating sequence of straight lines and circular arcs. The transitions from one straight-line flight to another with different heading occur at or near points called waypoints. For example, a typical flight segment might be a straight flight toward a VOR station along an inbound VOR radial, followed by a straight flight on a different outbound radial. The transition from one heading and the capture of the new heading occurs near the VOR station. In this example, the VOR station can be considered a waypoint. Note, in this case (as in many situations), there is no compelling reason for the aircraft to fly directly over the waypoint. Since area navigation procedures make it possible to offset VOR stations in a nearly arbitrary fashion, this type of waypoint could be defined abundantly in the terminal area.

A different situation exists when an aircraft, after a period of straight flight, is required to fly over a waypoint and capture a new heading

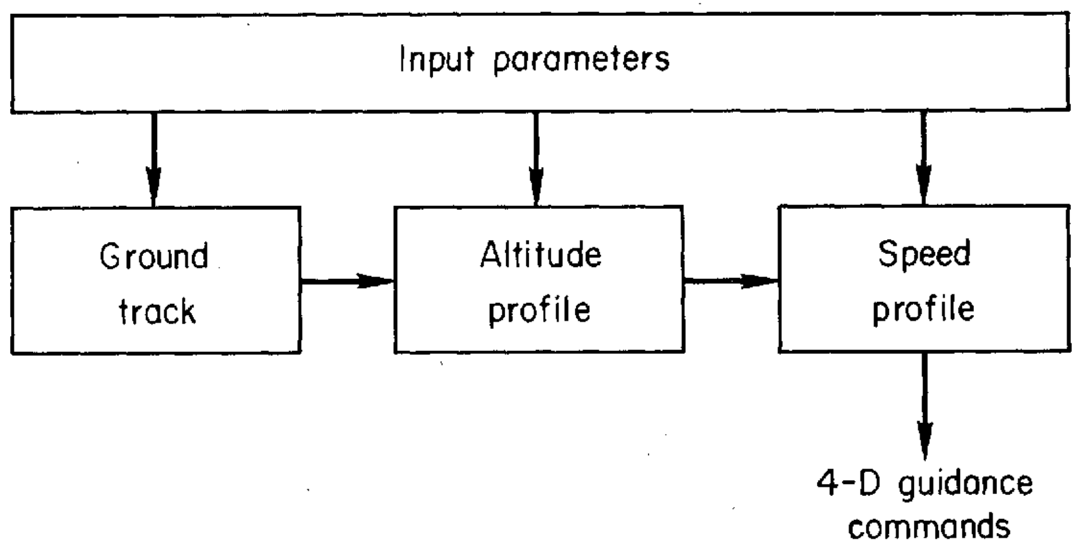


Figure 4.— Four-Dimensional Guidance command synthesis.

simultaneously. Such a waypoint might be the outer marker or the approach gate. It may also be any waypoint at which two or more flight paths merge and continue along a common path.

It is observed that the ground tracks corresponding to many terminal area flight paths, and indeed most of the enroute flight paths as well, can be characterized by these two types of waypoints. Consequently, the following waypoint definitions are made.

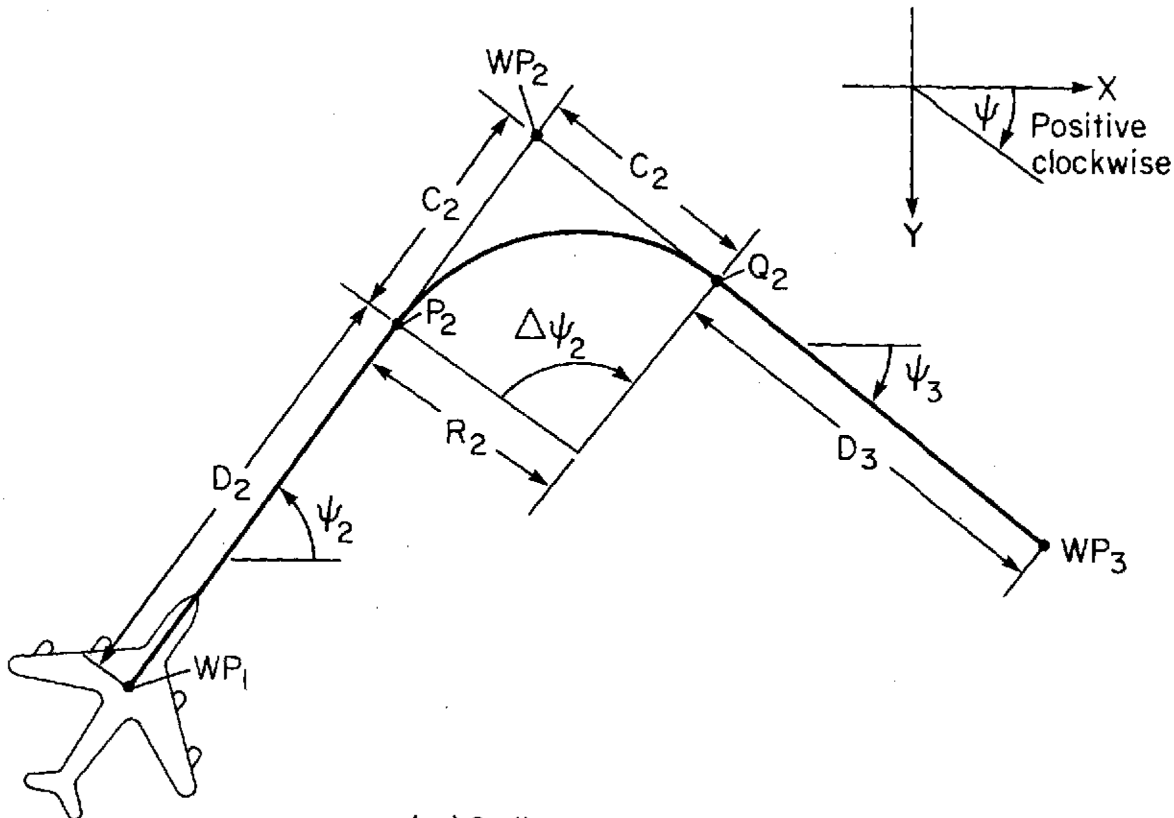
Ordinary Waypoint

Figure 5(a) is a graphical illustration of an ordinary waypoint where the ground track of a flight-path segment corresponds to a straight-turn-straight sequence from point WP_1, via point WP_2 to point WP_3. The aircraft at point WP_1 is flying at a heading ψ_2 directly toward point WP_2. Prior to reaching WP_2, at point P_2, a circular turn of radius R_2 is begun and is continued until, at point Q_2, the new desired heading, ψ_3, is achieved. If such a ground track between points WP_1 and WP_3 is to be obtained by specifying the x,y coordinates of WP_2 and the desired turning radius, then WP_2 is defined as an ordinary waypoint. (This definition does not imply that points WP_1 and/or WP_3 are ordinary waypoints.) Clearly, any straight-turn-straight sequence can be specified by an ordinary waypoint as long as the angular extent of the turn is less than 180°. Stated in a different form, if the intersection of two straight directed line segments is considered to be an ordinary waypoint, then the ground track of the resulting straight-turn-straight sequence is uniquely determined (assuming, of course, that the desired turn radius, R_2, is also specified).

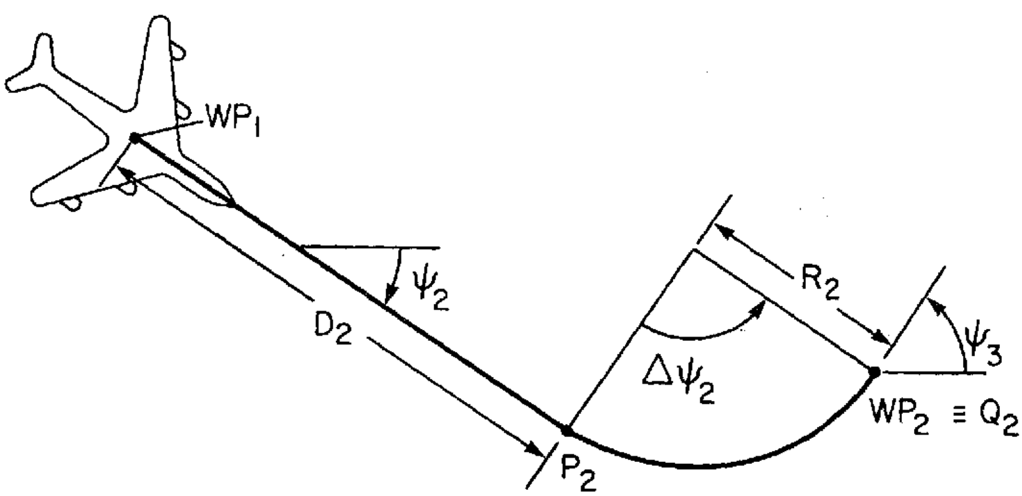
Final Heading Waypoint

A final heading waypoint is one over which the aircraft just completes a turn and captures the desired heading of the next straight-line segment. This situation is illustrated in figure 5(b), which shows an aircraft at point WP_1 flying toward point P_2 at a heading ψ_2. At P_2, the appropriate turn of radius R_2 is begun and is continued until, precisely at point WP_2, the new desired heading ψ_3 is achieved. If the ground track between points WP_1 and WP_2 is generated in this manner, then WP_2 is defined as a final heading waypoint. (This does not imply that point WP_1 is a final heading waypoint.) The essential quantities to be specified for a final heading waypoint are the x,y coordinates of WP_2, the heading ψ_3 from WP_2 to the next point, say WP_3, and the radius of turn, R_2, at WP_2. It follows from the above definition that, if the ground track of a straight-turn-straight sequence is specified by a final heading waypoint, then the initial heading, ψ_2, and the point P_2 where the aircraft initiates the turn must be computed from geometric considerations. Note that it is possible to generate a turn greater than 180° by using a final heading waypoint.

It is clear from the above definitions that any ground track consisting of a sequence of straingt-line segments and circular arcs can be uniquely



(a) Ordinary waypoint



(b) Final heading waypoint

Figure 5.– Waypoint parameters.

generated by merely specifying the appropriate sequence of waypoint parameters. What is required is an algorithm to calculate those parameters of the ground track that are not specified explicitly. Thus, for an ordinary waypoint, such as WP_2 in figure 5(a), only the x,y coordinates of the points WP_1, WP_2, and WP_3, along with the turn radius R_2, would be specified explicitly; all other parameters shown in figure 5(a) need to be computed. For a final heading waypoint, such as WP_2 in figure 5(b), only the x,y coordinates of points WP_1 and WP_2, along with the turn radius R_2 and the desired waypoint heading ψ_3, would be specified explicitly; the other parameters shown in figure 5(b) must be computed.

It is convenient to partition the algorithm into two parts, one for each of the two types of waypoints. Before the overall algorithm is presented, the two separate subalgorithms are described in detail.

First, consider an ordinary waypoint. Referring to figure 5(a), assume that the x,y coordinates of the points WP_1, WP_2, WP_3 are given by (XWP_1, YWP_1), (XWP_2, YWP_2), and (XWP_3, YWP_3), respectively. Then the headings ψ_2 and ψ_3 are given by

$$\psi_2 = \tan^{-1}\left(\frac{YWP_2 - YWP_1}{XWP_2 - XWP_1}\right), \qquad -\pi < \psi_2 \leq \pi \tag{13}$$

$$\psi_3 = \tan^{-1}\left(\frac{YWP_3 - YWP_2}{XWP_3 - XWP_2}\right), \qquad -\pi < \psi_3 \leq \pi \tag{14}$$

If ψ_2 and ψ_3 are known, the direction and angular extent of the turn $\Delta\psi_2$ can be expressed as

$$\Delta\psi_2 = \left\{\begin{array}{lll} \psi_3 - \psi_2 + 2\pi \;, & \text{if SIGN} > 0 & \text{and} \quad \psi_3 < \psi_2 \\ \psi_3 - \psi_2 - 2\pi \;, & \text{if SIGN} < 0 & \text{and} \quad \psi_3 > \psi_2 \\ \psi_3 - \psi_2 \;, & \text{otherwise} & \end{array}\right\} \tag{15}$$

where SIGN is given by

$$\text{SIGN} = \text{SGN}[(YWP_3 - YWP_2) \cos \psi_2 - (XWP_3 - XWP_2) \sin \psi_2] \tag{16}$$

and the function SGN(·) is defined as

$$\text{SGN}(\cdot) = \left\{\begin{array}{ll} 1, & \text{if} \quad (\cdot) > 0 \\ 0, & \text{if} \quad (\cdot) = 0 \\ -1, & \text{if} \quad (\cdot) < 0 \end{array}\right\} \tag{17}$$

By convention, the direction of the turn is to the right if $\Delta\psi_2$ is positive and to the left if negative. Since $-\pi < \psi_2, \psi_3 \leq \pi$, $\Delta\psi_2$ as given by equation (15) lies within the range $-\pi < \Delta\psi_2 < \pi$.

Once $\Delta\psi_2$ is calculated, the expression for C_2 (see fig. 5(a)) is obtained in terms of $\Delta\psi_2$ and the radius R_2:

$$C_2 = R_2 \cdot \tan\left(\frac{|\Delta\psi_2|}{2}\right) \tag{18}$$

The lengths of the two straight segments, D_2 and D_3, are then given by

$$D_2 = [(XWP_2 - XWP_1)^2 + (YWP_2 - YWP_1)^2]^{1/2} - C_2 \tag{19}$$

$$D_3 = [(XWP_3 - XWP_2)^2 + (YWP_3 - YWP_2)^2]^{1/2} - C_2 \tag{20}$$

It is now a simple matter to determine (XP_2, YP_2) and (XQ_2, YQ_2), the x,y coordinates of the beginning and end of the turn, respectively:

$$\left.\begin{aligned} XP_2 &= XWP_2 - C_2 \cdot \cos\psi_2 \\ YP_2 &= YWP_2 - C_2 \cdot \sin\psi_2 \end{aligned}\right\} \tag{21}$$

$$\left.\begin{aligned} XQ_2 &= XWP_2 + C_2 \cdot \cos\psi_3 \\ YQ_2 &= YWP_2 + C_2 \cdot \sin\psi_3 \end{aligned}\right\} \tag{22}$$

It is important to recognize that, since the locations of points WP_1, WP_2, WP_3 may be completely arbitrary, it is possible to specify them in such a way that D_2 or D_3 or both (as given by eqs. (19) and (20)), are negative. Negative D_2 and/or D_3 implies that WP_1 and/or WP_3 is too close to the waypoint WP_2 for the given turning radius. The implications of this situation, and possible remedies for it, are discussed later in this section.

The above computational procedure is well suited for computer implementation as a separate subroutine. Accordingly, a subroutine called ROUND has been written and is described in more detail in the appendix.

Consider a final heading waypoint such as shown in figure 5(b). It is assumed that the following quantities are specified: (XWP_1, YWP_1) and (XWP_2, YWP_2), the x,y coordinates of the points WP_1 and WP_2; ψ_3, the desired heading at WP_2; and R_2, the radius of turn. The parameters to be determined are the heading ψ_2 and length D_2 of the straight flight segment, the x,y coordinates (XP_2, YP_2) of point P_2, and the direction and magnitude of turn $\Delta\psi_2$.

Before proceeding with the solution, it is informative to consider this problem from a different point of view. Consider an aircraft located at point WP_2 with a heading ψ_3' (see fig. 6) given by

$$\psi_3' = MOD_\pi(\psi_3 + \pi) \tag{23}$$

(for an arbitrary angle ψ expressed as the sum $\psi = \psi_m + n \cdot \pi$ such that $-\pi < \psi_m \leq \pi$ and n is an integer, $MOD_\pi(\psi) = \psi_m$). Assume that the aircraft is to fly directly to point WP_1 in such a manner that the path length of the corresponding ground track is minimized subject to the minimum turning radius constraint R_2. The resulting ground track is seen to be precisely the solution to the original problem, except that the aircraft will actually fly it in the reverse direction, that is, from WP_1 to WP_2. This type of minimum path-length guidance problem was first proposed in reference 6, where the appropriate switching functions for the control law were calculated. A modified solution to the problem is given here, one particularly suitable for computer implementation.

Define ψ_2' and $\Delta\psi_2'$ as

$$\psi_2' = MOD_\pi(\psi_2 + \pi) \tag{24}$$

$$\Delta\psi_2' = -\Delta\psi_2 \tag{25}$$

and let (XO,YO) denote the x,y coordinates of the center of the turn, point O. Using certain geometrical arguments, the expressions for (XO,YO) are

$$XO = XWP_2 - R_2 \cdot SIGN \cdot \sin\psi_3' \tag{26}$$

$$YO = YWP_2 + R_2 \cdot SIGN \cdot \cos\psi_3' \tag{27}$$

where SIGN is defined as

$$SIGN = SGN[(YWP_1 - YWP_2)\cos\psi_3' - (XWP_1 - XWP_2)\sin\psi_3'] \tag{28}$$

The length, D_2, of the straight-line segment is then

$$D_2 = [(XWP_1 - XO)^2 + (YWP_1 - YO)^2 - R_2{}^2]^{1/2} \tag{29}$$

and the expressions for angles α_1, α_2, and α_3 (see fig. 6) are

$$\alpha_1 = MOD_\pi(\psi_3' - SIGN \cdot \frac{\pi}{2}) \tag{30}$$

$$\alpha_2 = \tan^{-1}\left(\frac{YWP_1 - YO}{XWP_1 - XO}\right) \tag{31}$$

$$\alpha_3 = \tan^{-1}\left(\frac{D_2}{R_2}\right) \tag{32}$$

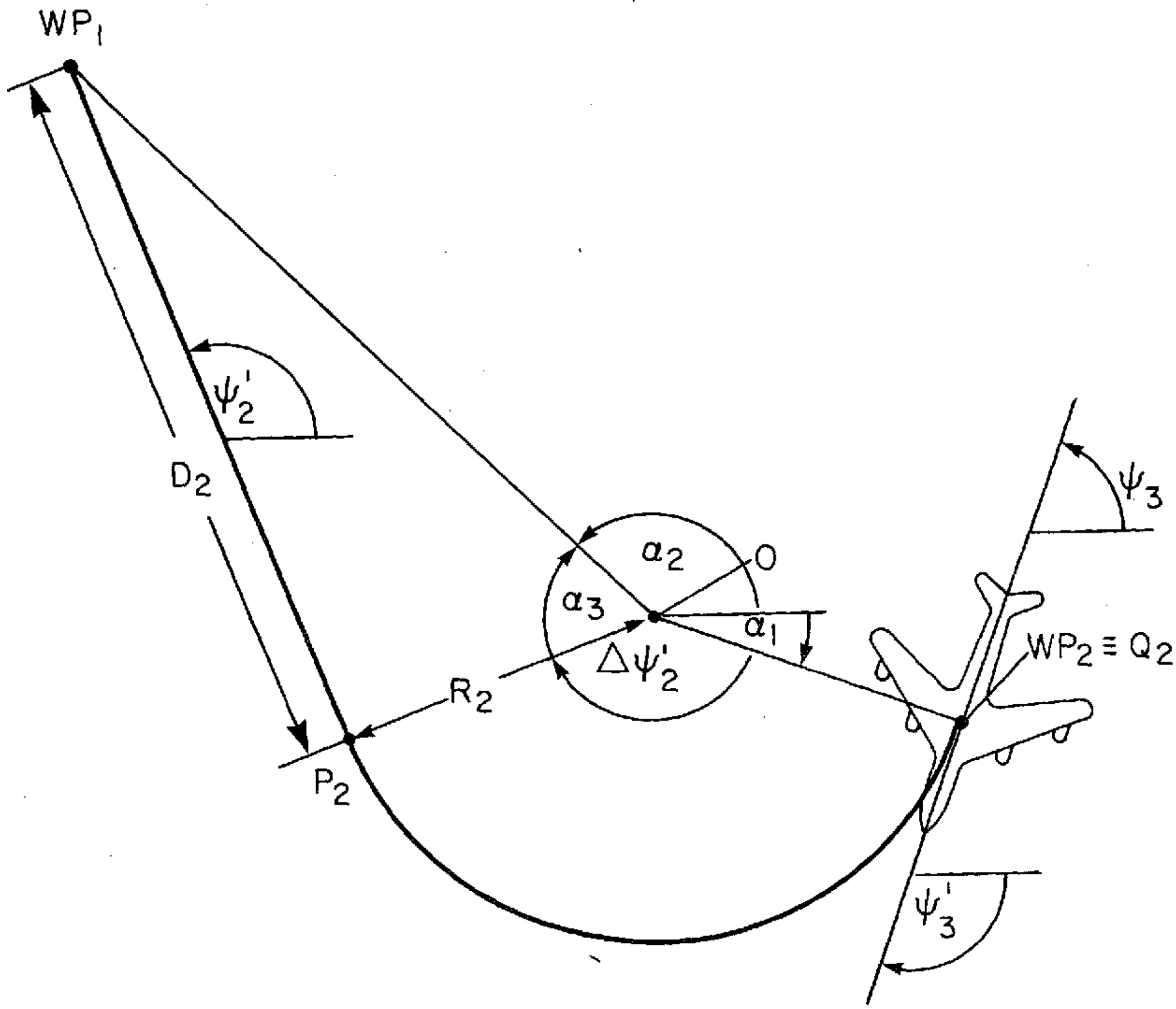


Figure 6.– Inverse final heading waypoint problem.

Once α_1, α_2, and α_3 are known, the direction and angular extent of the turn $\Delta\psi_2'$ can be expressed as

$$\Delta\psi_2' = \begin{cases} \alpha_2 - \alpha_1 - \mathrm{SIGN} \cdot \alpha_3 + 2\pi, & \text{if} \quad \mathrm{SIGN} > 0 \quad \text{and} \quad \alpha_2 < \alpha_1 \\ \alpha_2 - \alpha_1 - \mathrm{SIGN} \cdot \alpha_3 - 2\pi, & \text{if} \quad \mathrm{SIGN} < 0 \quad \text{and} \quad \alpha_2 > \alpha_1 \\ \alpha_2 - \alpha_1 - \mathrm{SIGN} \cdot \alpha_3 & \text{otherwise} \end{cases} \tag{33}$$

where the expression for SIGN is given by equation (28). The heading, ψ_2', of the straight-line segment and the x,y coordinates (XP_2,YP_2) of point P_2 are obtained in a straightforward fashion:

$$\psi_2' = \mathrm{MOD}_\pi(\psi_3' + \Delta\psi_2') \tag{34}$$

$$\left.\begin{aligned} XP_2 &= XO + R_2 \cdot \mathrm{SIGN} \cdot \sin \psi_2' \\ YP_2 &= YO - R_2 \cdot \mathrm{SIGN} \cdot \cos \psi_2' \end{aligned}\right\} \tag{35}$$

The desired parameters of the original final heading waypoint problem are now easily derived. In fact, the solution to the modified problem yields the length of the straight flight segment and the x,y coordinates of the beginning of the turn, point P_2. The heading ψ_2 is obtained from ψ_2' by

$$\psi_2 = \mathrm{MOD}_\pi(\psi_2' + \pi) \tag{36}$$

and the direction and magnitude of the turn $\Delta\psi_2$ are given by equation (25), that is,

$$\Delta\psi_2 = -\Delta\psi_2' \tag{37}$$

As in the case of ordinary waypoint, it is possible to specify a final heading waypoint in such a way that the radicand in equation (29) is negative. This condition implies that, for the given waypoint location, desired final heading, and minimum turning radius, point WP_1 is too close to waypoint WP_2. Action to correct this condition is discussed later.

A subroutine called NEWPSI has been written to compute the various parameters associated with a final heading waypoint problem. This subroutine is described in more detail in the appendix.

Ground Track Computation

Using the definitions of ordinary and final heading waypoints, along with the computational subalgorithms described previously, an algorithm is now described that generates a complete two-dimensional ground track from a sequence of specified waypoints. A crucial idea in the algorithm is that it generates the ground track by processing the waypoints in *reverse order*, beginning with the last waypoint and proceeding sequentially until the first waypoint is processed. This procedure makes it possible to generate the required headings at final heading waypoints automatically, thereby eliminating the need to specify these headings explicitly as separate input data. The only exception to this is the last waypoint (always assumed to be a final heading waypoint), which requires that the value of the desired final heading, denoted by ψ_{final}, be specified explicitly. This assumption is consistent with usual terminal area operational procedures. Placing the last waypoint on the extension of the runway centerline, declaring it to be a final heading waypoint, and specifying the runway heading as the desired heading assures that the aircraft is guided to the localizer and is properly aligned with the runway at some point prior to landing. Such a final waypoint could be the approach gate, the outer marker, or any other point on the runway centerline beyond which 4-D control of the aircraft is either inappropriate or not desirable. For example, based on the nominal landing speed of the aircraft, the last waypoint may be 1 minute of flight from touchdown.

Consider a sequence of NWP waypoints, denoted by WP_i, i = 1,2,...,NWP, and let the following parameters be given for each waypoint:

XWP_i, YWP_i	x,y coordinates
R_i	turning radius
$INDEX_i$	type of waypoint

The x,y coordinates are given in a runway-centered coordinate system, in which the x-axis coincides with the runway centerline and is positive in the direction of landing, the y-axis is perpendicular to the runway and positive to the right, and the origin is at the touchdown point. What might be a sequence of five waypoints for the ground track of an arriving aircraft is shown in figure 7. The dashed lines represent the idealized ground track the aircraft would follow if it could change heading instantaneously at the waypoints.

The turning radius R_i may or may not be explicitly specified as an input. If it is not specified, then the minimum turning radius is computed based on the maximum admissible bank angle and maximum possible ground speed at waypoint i, and R_i is set equal to this minimum turning radius. If R_i is specified as an input, then the minimum turning radius is computed as described above and is compared with R_i: if R_i is greater than or equal to the minimum feasible turning radius, then processing continues; otherwise, an error message is generated and processing stops. For the remainder of this section, R_i is assumed to be greater than or equal to the respective minimum feasible turning radius for all i = 2,3,...,NWP. (The computation of the minimum turning radius is discussed at the end of this section.

The input parameter $INDEX_i$ is a binary variable whose value is 0 or 1. $INDEX_i = 0$ implies that waypoint i is an ordinary waypoint, while $INDEX_i = 1$ means that waypoint i is a final heading waypoint. As mentioned earlier, the last waypoint is always a final heading waypoint, that is,

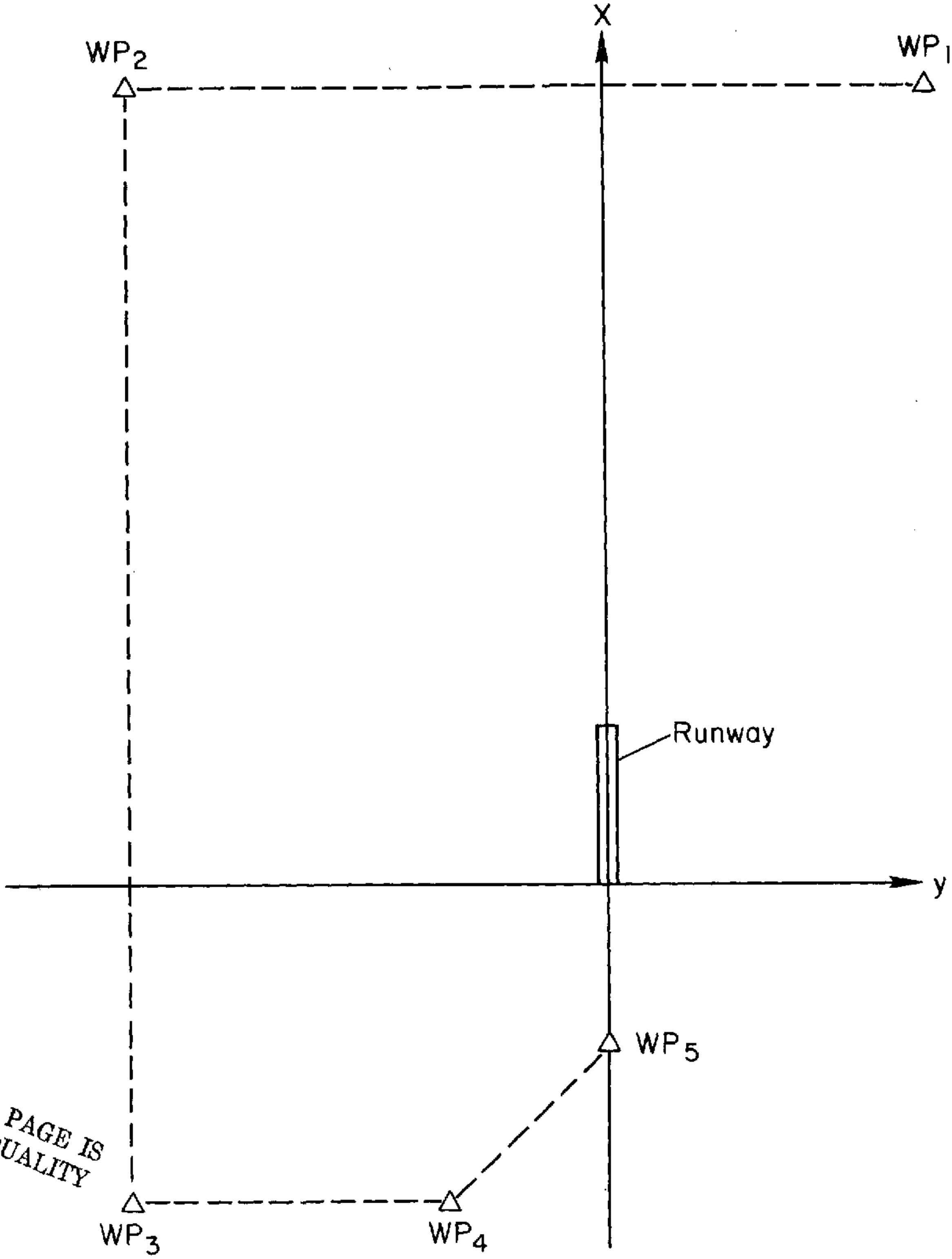


Figure 7.— A typical sequence of waypoints.

$INDEX_{NWP} = 1$. The first waypoint is also treated as a final heading waypoint, although as will be shown, this is not necessary and is merely a convention. All other waypoints may be chosen as ordinary or final heading type, so that $\{INDEX_i\}$, $i = 2,3,\ldots,NWP-1$, is a completely arbitrary sequence of 0's and 1's.

Assume that the algorithm has processed waypoints $NWP, NWP-1, \ldots, i+1$, and is currently to process waypoint i. At this stage of the algorithm, the following quantities are available for all $j = i+1, i+2, \ldots, NWP$:

ψ_j	heading of straight flight segment from WP_{j-1} to point P_j
(XP_j, YP_j)	x,y coordinates of P_j, the point where the turn associated with waypoint j begins
$\Delta\psi_j$	angular extent of turn associated with waypoint j
(XQ_j, YQ_j)	x,y coordinates of Q_j, the point where the turn associated with waypoint j ends

Now assume that waypoint i is an ordinary waypoint, that is, $INDEX_i = 0$ (see fig. 8). Then, from figure 5(a), point WP_1 is replaced by WP_{i-1}, WP_2 by WP_i, and WP_3 by P_{i+1}, and the subroutine ROUND is called to compute the above parameters for waypoint i. The expressions for these parameters are essentially the same as equations (13) through (22), but are repeated below for completeness:

$$\psi_i = \tan^{-1}\left(\frac{YWP_i - YWP_{i-1}}{XWP_i - XWP_{i-1}}\right), \quad -\pi < \psi_i \le \pi \tag{38}$$

$$SIGN = SGN[(YP_{i+1} - YWP_i)\cos\psi_i - (XP_{i+1} - XWP_i)\sin\psi_i] \tag{39}$$

$$\Delta\psi_i = \begin{cases} \psi_{i+1} - \psi_i + 2\pi, & \text{if } SIGN > 0 \text{ and } \psi_{i+1} < \psi_i \\ \psi_{i+1} - \psi_i - 2\pi, & \text{if } SIGN < 0 \text{ and } \psi_{i+1} > \psi_i \\ \psi_{i+1} - \psi_i, & \text{otherwise} \end{cases} \tag{40}$$

$$C_i = R_i \cdot \tan\left(\frac{|\Delta\psi_i|}{2}\right) \tag{41}$$

$$\left.\begin{aligned} XP_i &= XWP_i - C_i \cos\psi_i \\ YP_i &= YWP_i - C_i \sin\psi_i \end{aligned}\right\} \tag{42}$$

$$\left.\begin{aligned} XQ_i &= XWP_i + C_i \cos\psi_i \\ YQ_i &= YWP_i + C_i \sin\psi_i \end{aligned}\right\} \tag{43}$$

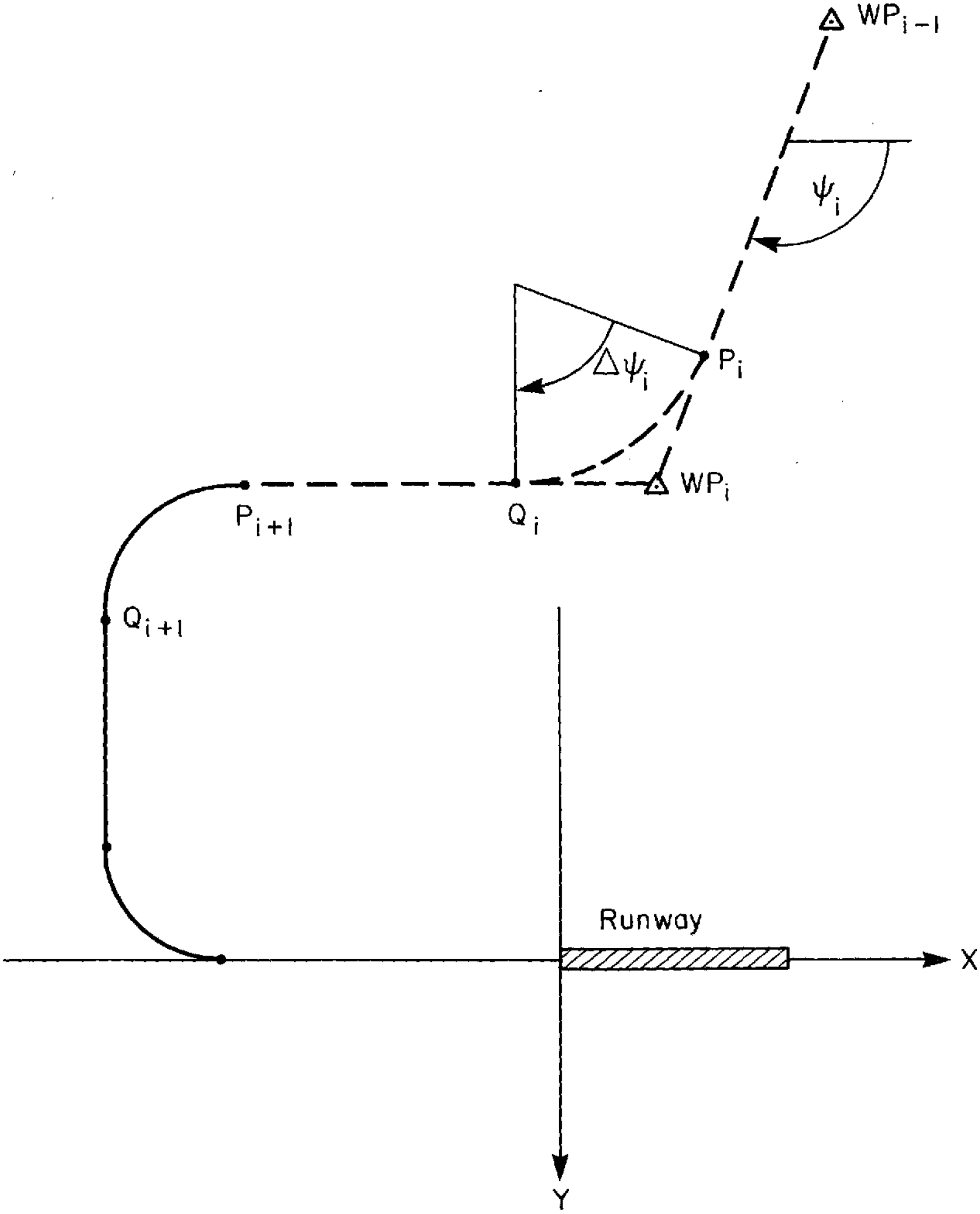


Figure 8.— Ground track synthesis with ordinary waypoint.

If waypoint i is a final heading waypoint, then $INDEX_i = 1$ (fig. 9). In this case, from figure 5(b), point WP_1 is replaced by WP_{i-1}, WP_2 by WP_i, P_2 by P_i, and the subroutine NEWPSI is called to generate the following ground track parameters:

$$\psi'_{i+1} = MOD_\pi(\psi_{i+1} + \pi) \tag{44}$$

$$SIGN = SGN[(YWP_{i-1} - YWP_i)\cos\psi'_{i+1} - (XWP_{i-1} - XWP_i)\sin\psi'_{i+1}] \tag{45}$$

$$\left.\begin{aligned} XO &= XWP_i - R_i \cdot SIGN \cdot \sin\psi'_{i+1} \\ YO &= YWP_i + R_i \cdot SIGN \cdot \cos\psi'_{i+1} \end{aligned}\right\} \tag{46}$$

$$D_i = [(XWP_{i-1} - XO)^2 + (YWP_{i-1} - YO)^2 - R_i{}^2]^{1/2} \tag{47}$$

$$\alpha_1 = MOD_\pi\left(\psi'_{i+1} - SIGN \cdot \frac{\pi}{2}\right) \tag{48}$$

$$\alpha_2 = \tan^{-1}\left(\frac{YWP_{i-1} - YO}{XWP_{i-1} - XO}\right), \qquad -\pi < \alpha_2 \leq \pi \tag{49}$$

$$\alpha_3 = \tan^{-1}\left(\frac{D_i}{R_i}\right) \tag{50}$$

$$\Delta\psi'_i = \begin{cases} \alpha_2 - \alpha_1 - SIGN \cdot \alpha_3 + 2\pi, & \text{if } SIGN > 0 \text{ and } \alpha_2 < \alpha_1 \\ \alpha_2 - \alpha_1 - SIGN \cdot \alpha_3 - 2\pi, & \text{if } SIGN < 0 \text{ and } \alpha_2 > \alpha_1 \\ \alpha_2 - \alpha_1 - SIGN \cdot \alpha_3, & \text{otherwise} \end{cases} \tag{51}$$

$$\Delta\psi_i = -\Delta\psi'_i \tag{52}$$

$$\psi'_i = MOD_\pi(\psi'_{i+1} + \Delta\psi'_i) \tag{53}$$

$$\psi_i = MOD_\pi(\psi'_i + \pi) \tag{54}$$

$$\left.\begin{aligned} XP_i &= XO + R_i \cdot SIGN \cdot \sin\psi'_i \\ YP_i &= YO - R_i \cdot SIGN \cdot \cos\psi'_i \end{aligned}\right\} \tag{55}$$

$$\left.\begin{aligned} XQ_i &= XWP_i \\ YQ_i &= YWP_i \end{aligned}\right\} \tag{56}$$

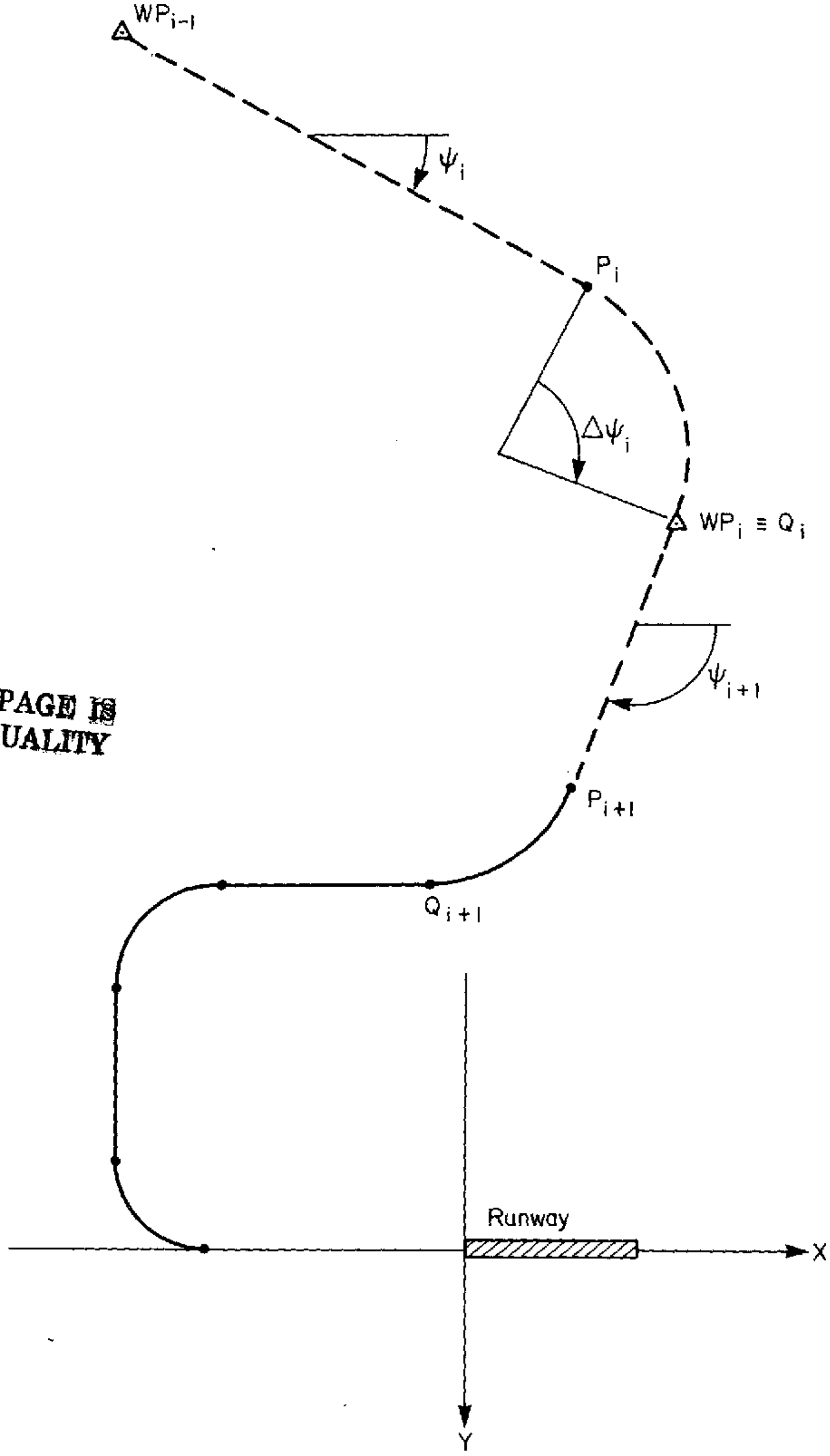


Figure 9.— Ground track synthesis with final heading waypoint.

In order to describe the two-dimensional ground track completely, it remains to determine the length of the straight flight segments and the arclength of the turns. Let D_i denote the length of straight flight from point Q_{i-1} to point P_i, and let ΔD_i represent the arclength of the turn $\Delta\psi_i$, $i = 2,3,\ldots,NWP$. Then the expressions for D_i and ΔD_i are

$$D_i = [(XP_i - XQ_{i-1})^2 + (YP_i - YQ_{i-1})^2]^{1/2} \tag{57}$$

$$\Delta D_i = R_i \cdot |\Delta\psi_i| \tag{58}$$

Minimum Turning Radius

In order to compute the minimum turning radius at each waypoint, it is necessary to know the maximum ground speed of the aircraft at the waypoint and the maximum permissible bank angle. The speed profile of the aircraft, however, is not known when the ground track is generated. Consequently, the maximum ground speed must be approximated to compute the minimum turning radius. In order to assure that at no waypoint will the guidance commands require the aircraft to violate the maximum bank-angle constraint, the maximum ground speed at each waypoint is taken to be the scalar sum of the maximum possible airspeed at the waypoints and the wind magnitude. The maximum possible airspeed, in turn, is a function of the maximum permissible airspeed at the last waypoint and the distance along the flight path to the last waypoint.

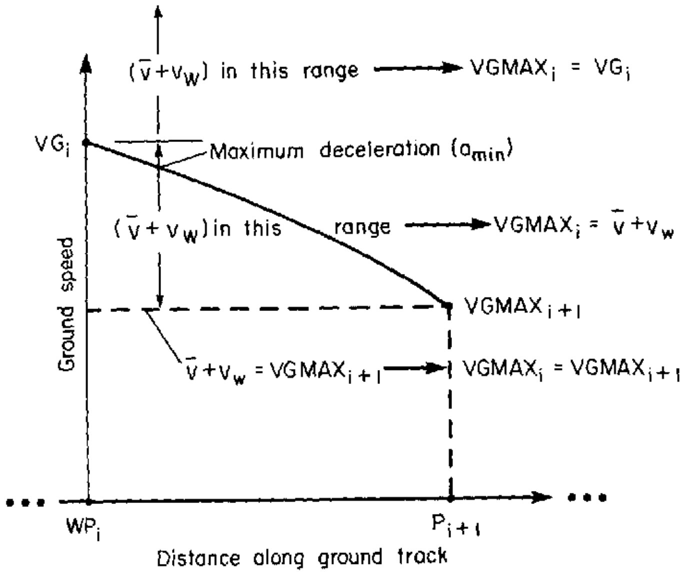
Let $\bar{v}$ denote the maximum admissible airspeed in the terminal area, v_w the magnitude of the wind, and $VGMAX_i$ the maximum possible ground speed of the aircraft at waypoint i. Since the airspeed to be achieved at the last waypoint is specified as v_{final}, $VGMAX_{NWP} = v_{final} + v_w$. For $i < NWP$, $VGMAX_i$ is approximated by the following (see sketch (a)).

$$VGMAX_i = \begin{cases} VGMAX_{i+1}, & \text{if } VGMAX_{i+1} = \bar{v} + v_w \\ \bar{v} + v_w, & \text{if } VG_i \geq \bar{v} + v_w \\ VG_i, & \text{otherwise} \end{cases} \tag{59}$$

where the expression for VG_i is

$$VG_i = \left\{VGMAX_{i+1}^2 - 2 \cdot a_{min}[(XP_{i+1} - XWP_i)^2 + (YP_{i+1} - YWP_i)^2]^{1/2}\right\}^{1/2} \tag{60}$$

The physical interpretation of VG_i is as follows: if the aircraft were to fly directly from waypoint i to point P_{i+1} and apply the maximum deceleration a_{min}, then it could just achieve a ground speed change from VG_i at point WP_i to $VGMAX_{i+1}$ at point P_{i+1}. Note how equations (59) and (60) reflect the assumption that the upper envelope of the admissible speeds



Sketch (a).

decreases monotonically as the aircraft approaches the last waypoint. Specifically, if for some value of i, say $i = j$, $VGMAX_j = \bar{v} + v_w$, then $VGMAX_{i-1} = VGMAX_i = \bar{v} + v_w$ for all $i \leq j$.

Once $VGMAX_i$ is known, the minimum turning radius, $RMIN_i$, at waypoint i can be computed from

$$RMIN_i = \frac{VGMAX_i{}^2}{g \cdot \tan \phi_{max}} \tag{61}$$

ALTITUDE PROFILE

Except for the very last phase of flight, namely flaring and decrabbing, the motion of an aircraft in the vertical plane consists of constant flight-path-angle segments connected by some sort of smooth transitioning between adjacent flight-path angles. To generate an altitude profile, it is assumed that transitioning from one flight-path angle to another is instantaneous. Whatever inaccuracy results from this assumption is minimized by introducing appropriate lead times in the actual control law. This method of compensating for finite pitch rate is discussed under Four-Dimensional Guidance Commands.

The basic assumption in generating an altitude profile is that the flight-path angle between adjacent waypoints remains constant. Thus, in order to generate the constant flight-path-angle segments, the desired waypoint altitudes and the horizontal distance between each pair of adjacent waypoints must be known.

In using waypoint altitudes to generate a sequence of constant flight-path angles, the following ambiguity must be resolved: ordinary waypoints do not lie on the actual flight path generated by the techniques described previously; consequently, the meaning of waypoint altitude for ordinary waypoints must be defined. Therefore, it is assumed, somewhat arbitrarily, that the waypoint altitude should be achieved precisely at the end of the turn associated with the waypoint in question. Note that this is entirely consistent with the definition of final heading waypoints. Thus, if ZWP_i, i = 1, 2, ..., NWP, are the desired waypoint altitudes, then ZWP_i is actually the altitude of the aircraft at the point when the turn associated with WP_i is completed. Since the end of a turn at WP_i is denoted by point Q_i, the altitude, ZQ_i, at point Q_i is therefore given by

$$ZQ_i = ZWP_i \ , \quad i = 1,2,\ldots,NWP \tag{62}$$

The horizontal distance between points Q_{i-1} and Q_i has already been computed and is simply the sum of the straight flight distance D_i and the arclength of the turn at WP_i, that is, ΔD_i. If the constant flight-path angle between points Q_{i-1} and Q_i is denoted by GAM_i, then the expression for GAM_i is (fig. 10)

$$GAM_i = \tan^{-1}\left(\frac{ZQ_{i-1} - ZQ_i}{D_i + \Delta D_i}\right), \quad i = 2,3,\ldots,NWP \tag{63}$$

Note that since the vertical axis is positive downward by convention, $GAM_i > 0$ implies ascending flight, while $GAM_i < 0$ implies descending flight.

For sufficiently small values of its argument, the function $\tan^{-1}(\cdot)$ can be approximated to a high degree of accuracy by its argument. Since the argument of $\tan^{-1}(\cdot)$ in equation (63) is generally quite small, and since evaluating $\tan^{-1}(\cdot)$ requires a relatively long execution time, for the computer implementation of the guidance scheme, the value of GAM_i is computed from the approximate expression

$$GAM_i = \frac{ZQ_{i-1} - ZQ_i}{D_i + \Delta D_i} \tag{64}$$

Computing the sequence of flight-path angles according to equation (64) is straightforward. It is conceivable that GAM_i (as given by eq. (64)) may be either infeasible from an aircraft performance point of view or undesirable due to other operational constraints. Therefore, GAM_i must be checked whether it satisfies the constraints

$$\gamma_{min} \le GAM_i \le \gamma_{max} \ , \quad i = 2,3,\ldots,NWP \tag{65}$$

where γ_{min} and γ_{max} are input parameters. If GAM_i satisfies inequalities (65), then computation continues; otherwise, a message is generated stating that GAM_i lies outside the admissable range and computation stops.

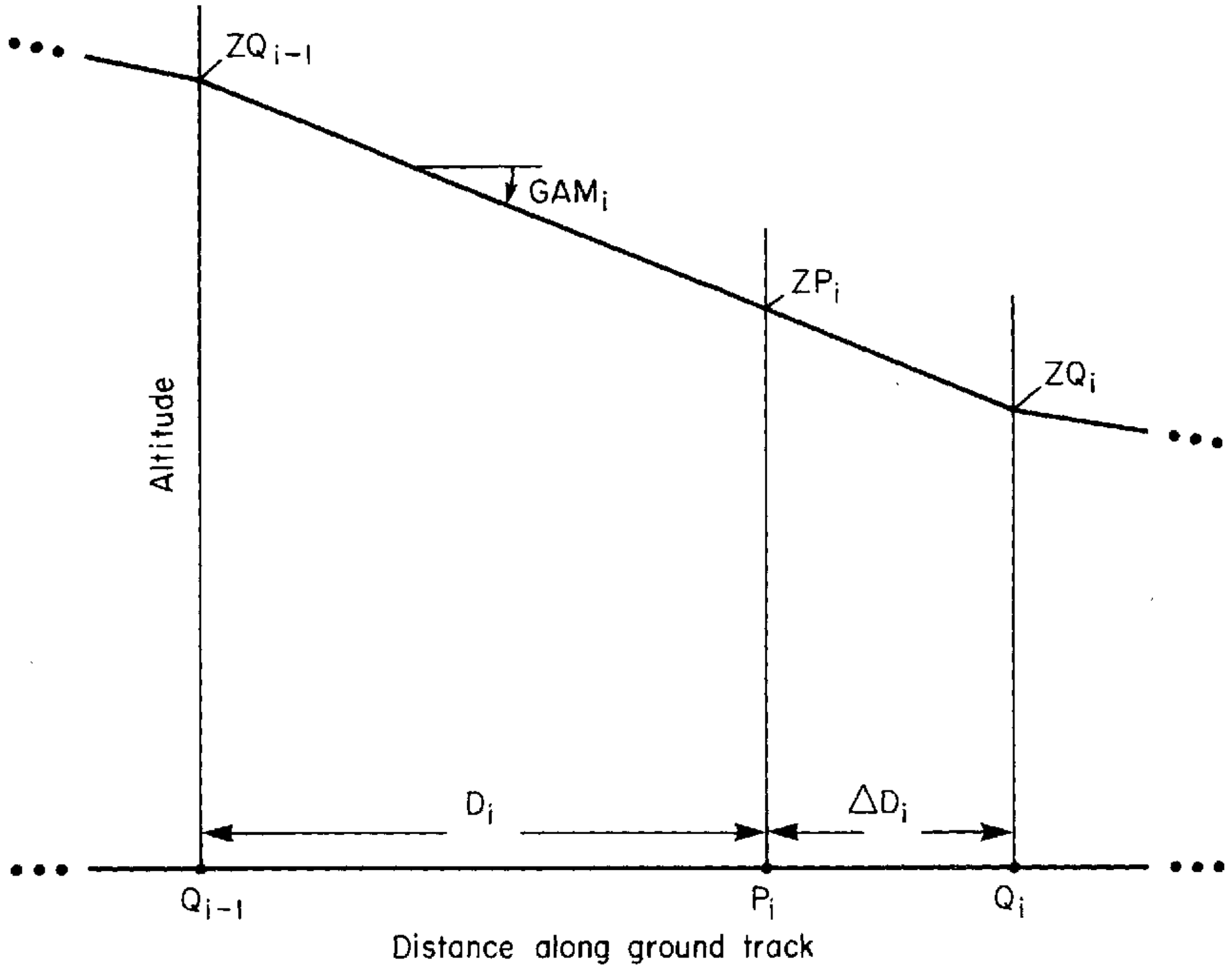


Figure 10.– Altitude profile synthesis.

For a complete altitude profile, it remains merely to determine what the altitude should be at the beginning of each turn, that is, at the points P_i. Let ZP_i denote the altitude at point P_i, then ZP_i is given by

$$ZP_i = ZQ_{i-1} - D_i \cdot GAM_i \tag{66}$$

A typical altitude profile between two arbitrary points Q_{i-1} and Q_i is shown in figure 10.

SPEED PROFILE

The primary objective of the 4-D guidance system described here is to guide the aircraft accurately not only in the three spatial dimensions, but also to control its motion as a function of time along the desired flight path, thereby controlling the time of its arrival at the end of the flight path. Thus, after the desired 3-D flight path is computed, it is necessary to determine a time schedule the aircraft is to maintain along the flight path. This, in turn, requires the determination of an appropriate speed profile.

Clearly, if the speed profile of the aircraft were specified rigidly, then so would its motion as a function of time along the flight path, in which case control of its arrival time would not be possible. While the admissible speed is not entirely arbitrary, at almost every point during the flight there exists a definite range of speeds within which the aircraft may operate.

Admissible Speed Ranges

As mentioned briefly under Problem Description and Basic Approach, the maximum admissible speed of an aircraft is dictated by either Federal Aviation Regulations or by structural constraints. FAR 91.70a limits all aircraft to an indicated airspeed of 250 knots below 3048 m (10,000 ft) mean sea level. Since most terminal area operations take place below 3048 m, the 250-knot speed may be considered an absolute upper bound. When the aircraft descends to lower altitudes and approaches the runway, the speed is reduced and flaps are deployed to maintain sufficient stall margin. Since at any flap setting it is unsafe to fly at speeds greater than the flap placard speed, the upper bound on the admissible speeds must be reduced so that it is consistent with the flap placard speed.

The lower bound on the admissible speeds is basically a function of the stall speed of the aircraft at any given flap setting. The generally accepted lower bound on terminal area speeds is 30 percent above stall speed, which itself is a function of the flap setting. This lower bound is often increased to account for wind and gust effects.

In addition to the above considerations concerning the minimum and maximum admissible speeds, it is generally required that, for any flap setting, the aircraft be capable of flying at a maximum speed 80 percent above the stall speed (for the given flap setting) without the need for configuration change (ref. 7). Thus, if the flap setting of an aircraft is denoted by δ and the stall speed at that flap setting is represented by $V_s(\delta)$, then the speed range available to the aircraft at any flap setting without configuration change is given by

$$1.3V_s(\delta) \leq v \leq 1.8V_s(\delta) \tag{67}$$

From the point of view of pilot workload, it is clearly desirable to have a speed profile that requires a minimum number of configuration changes. Furthermore, operational and economic considerations dictate that aircraft land as quickly as possible with a minimum of delay. This, in turn, means that aircraft should maintain as high a speed as is consistent with the current air traffic situation for as long as possible. The question therefore arises: what is the simplest way to specify the admissible speed ranges for various types of aircraft so that the above ideas are reflected in the resulting speed profile? One alternative could be to specify the minimum and maximum speeds at every, or nearly every, point along the flight path. Such an approach would be not only prohibitive from the point of view of computer storage requirements, but also totally unnecessary. Instead, the basic philosophy is adopted that if the aircraft is sufficiently far away from the final waypoint where generally a prespecified speed must be achieved, then the minimum admissible speed should be set to $1.3\ V_s(o)$, while the maximum admissible speed should be given by the smallest of the three quantities: (a) 250 knots, (b) the flap placard speed $V_p(o)$ corresponding to zero flaps, and (c) $1.8\ V_s(o)$, where $V_s(o)$ is the stall speed of the aircraft with zero flap setting. As the aircraft proceeds along the flight path and approaches the final waypoint, both the minimum and maximum speeds must be lowered gradually so that the desired speed at the final waypoint can be achieved without excessive rates of speed change (it is assumed that the proper flap setting is maintained during the entire flight).

At this point, the objective is to select a small number of parameters which, when specified as input, could be used to generate the minimum and maximum speeds consistent with the concepts described above. It will be shown in the sequel that the stall speed and the flap placard speed of the aircraft at zero flap setting, the maximum rate of speed change, the desired air speed at the final waypoint, and the magnitude and direction of the wind, along with the knowledge of the ground track, are sufficient to determine the admissible speed ranges.

Let the desired air speed at the final waypoint be denoted by v_{final}, and let v_w and ψ_w denote the magnitude and direction of the constant wind field, respectively. Furthermore, let $VMIN_i$, $VMAX_i$, $i = 1,2,\ldots,NWP$, represent the minimum and maximum air speeds at each waypoint. More precisely, it is assumed that if there is a turn associated with waypoint i, then it is to be flown at a constant airspeed, say VA_i, that satisfies the constraints:

$$VMIN_i \leq VA_i \leq VMAX_i \tag{68}$$

Now let $\underline{c}$ and $\bar{c}$ be two prespecified numbers satisfying the inequalities

$$1.3 \leq \underline{c} < \bar{c} \leq 1.8 \tag{69}$$

and define the minimum and maximum admissible cruising airspeeds, $\underline{v}$ and $\bar{v}$ in the terminal area as follows:

$$\left.\begin{aligned} \underline{v} &= \underline{c} \cdot V_s(o) \\ \bar{v} &= \min\,[\bar{c} \cdot V_s(o),\ V_p(o),\ 250 \text{ knots}] \end{aligned}\right\} \tag{70}$$

It is now possible to determine the speeds $VMIN_i$ and $VMAX_i$, $i = 1,2,\ldots,NWP$. The procedure is sequential and starts with the last waypoint. Since the air speed at the last waypoint is specified, the following equalities are obvious:

$$VMIN_{NWP} = VMAX_{NWP} = v_{final} \tag{71}$$

It will be assumed throughout this analysis that

$$v_{final} < \underline{v} \tag{72}$$

This assumption is entirely consistent with current operational procedures since v_{final}, in most cases, will be the final approach speed of an aircraft. For $i < NWP$, $VMIN_i$ and $VMAX_i$ are computed as

$$VMIN_i = \begin{cases} \underline{v} & \text{if } VMIN_{i+1} = \underline{v}\text{; or if } VMIN_{i+1} < \underline{v} \text{ and } vmin_i \geq \underline{v} \\ vmin_i & \text{otherwise} \end{cases} \tag{73}$$

$$VMAX_i = \begin{cases} \bar{v} & \text{if } VMAX_{i+1} = \bar{v}\text{; or if } VMAX_{i+1} < \bar{v} \text{ and } vmax_i \geq \bar{v} \\ vmax_i & \text{otherwise} \end{cases} \tag{74}$$

where the expressions for $vmin_i$ and $vmax_i$ are

$$vmin_i = \left\{[VMIN_{i+1} + v_w \cos(\psi_{i+1} - \psi_w)]^2 - 2 \cdot a_{min} \cdot D_{i+1}\right\}^{1/2} - v_w \cos(\psi_{i+1} - \psi_w) \tag{75}$$

$$vmax_i = \left\{[VMAX_{i+1} + v_w \cos(\psi_{i+1} - \psi_w)]^2 - 2 \cdot a_{min} \cdot D_{i+1}\right\}^{1/2} - v_w \cos(\psi_{i+1} - \psi_w) \tag{76}$$

Both $vmin_i$ and $vmax_i$ can be given straightforward physical interpretation (fig. 11). Assume that the aircraft has just completed the turn at waypoint i and is located at the corresponding point Q_i. Then $vmin_i$ is the precise air speed from which the aircraft could decelerate, using the maximum

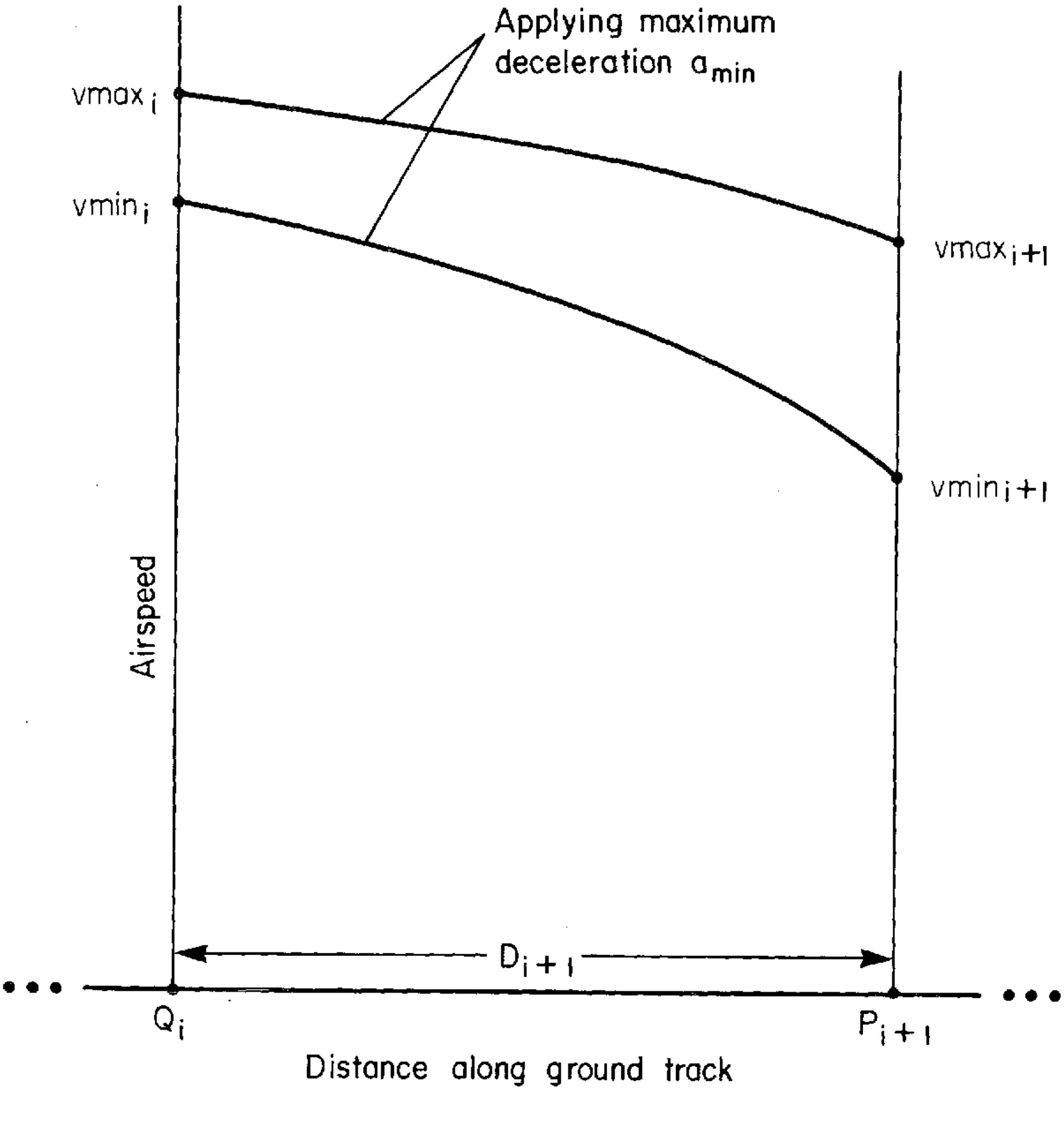


Figure 11.— Meaning of vmin and vmax.

deceleration a_{min}, such that the resulting air speed at the beginning of the next turn, that is, at point P_{i+1}, is exactly $VMIN_{i+1}$; similarly, $vmax_i$ is the precise air speed from which the aircraft could decelerate, using the maximum deceleration a_{min}, such that the resulting air speed at point P_{i+1} is exactly $VMAX_{i+1}$. The subroutine used to compute the speeds $VMIN_i$ and $VMAX_i$ is called VRANGE and is described in the appendix.

In order to define completely the minimum and maximum admissible speeds along the entire flight path, and not just at the waypoints, it is assumed that between any two adjacent waypoints the airspeed is constrained to the union of the admissible speed ranges at the two waypoints. Thus, for aircraft in flight from point Q_i to point P_{i+1}, the air speed is limited to the range $[\min(VMIN_i, VMIN_{i+1}), \max(VMAX_i, VMAX_{i+1})]$. This point is discussed further when the computation of the actual speed profile is described.

Attainable Arrival Times

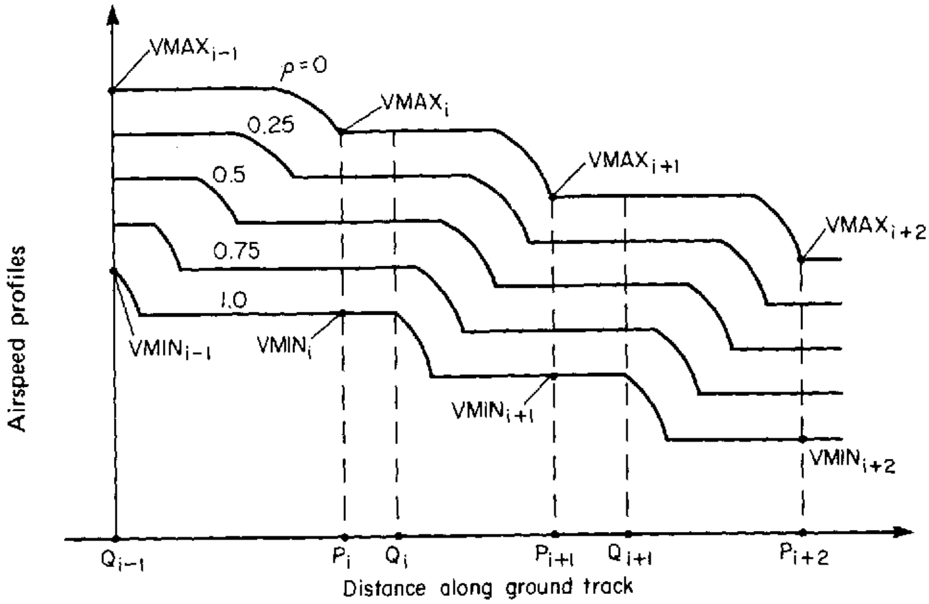
Once the minimum and maximum admissible speeds are determined, the computation of the corresponding range in the attainable times of arrival to the last waypoint becomes relatively straightforward. In order to compute the range of arrival times, it is necessary to determine the minimum and maximum flight times from the first to last waypoint. With no additional effort, the range of flight times from every waypoint to the last can be calculated. These numbers are useful when implementing the overall guidance scheme.

Clearly, a flight profile at the maximum admissible speeds results in the minimum flight time; similarly, flight at the minimum speeds yields the maximum flight time (it is understood that the flight is along the previously generated 3-D flight path). The subroutine called TRANGE is used to compute the minimum and maximum flight times.

Let $TMIN_i$, $TMAX_i$, $i = 1,2,\ldots,NWP$ denote the minimum and maximum flight times, respectively, from waypoint i to the last waypoint. $TMIN_i$ and $TMAX_i$ are computed via the subroutine called TRANS. The basic inputs required for TRANS are the specification of a waypoint number i, the 3-D flight-path parameters (actually, only the 2-D ground track parameters are needed) from point Q_{i-1} to point Q_i, the minimum and maximum admissible speeds at the two waypoints, and a positive number ρ satisfying the inequalities

$$0 \leq \rho \leq 1 \tag{77}$$

The parameter ρ plays an important role in the computation of flight times. It not only determines what the air speeds should be at the two waypoints involved, but it also specifies the point at which the required speed change should occur. In effect, ρ completely determines the speed profile between waypoints (i-1) and i once the minimum and maximum admissible speeds at the two waypoints are given (sketch (b)).



Sketch (b).

Let $VA_{i-1}(\rho)$ and $VA_i(\rho)$ denote the desired air speeds at waypoints (i-1) and i, respectively. Then $VA_{i-1}(\rho)$ and $VA_i(\rho)$ are defined in terms of ρ as

$$\left.\begin{aligned} VA_{i-1}(\rho) &= VMAX_{i-1} - \rho(VMAX_{i-1} - VMIN_{i-1}) \\ VA_i(\rho) &= VMAX_i - \rho(VMAX_i - VMIN_i) \end{aligned}\right\} \tag{78}$$

Note that, since $VMIN_i \le VMIN_{i-1}$ and $VMAX_i \le VMAX_{i-1}$, it follows that $VA_i(\rho) \le VA_{i-1}(\rho)$. The ground speeds at points Q_{i-1} and P_i are then given in terms of the above air speeds and the appropriate wind component. Thus, if $VQ_{i-1}(\rho)$ and $VP_i(\rho)$ denote the ground speeds at points Q_{i-1} and P_i, then

$$VQ_{i-1}(\rho) = VA_{i-1}(\rho) + v_W \cos(\psi_i - \psi_W) \tag{79}$$

$$VP_i(\rho) = VA_i(\rho) + v_W \cos(\psi_i - \psi_W) \tag{80}$$

If $VMIN_i < VMIN_{i-1}$ and/or $VMAX_i < VMAX_{i-1}$, then $VA_i(\rho) < VA_{i-1}(\rho)$ and therefore $VP_i(\rho) < VQ_{i-1}(\rho)$. This implies that a speed reduction must occur somewhere between waypoints i-1 and i. Where this speed change should occur is again determined by the parameter ρ. Let $S11_i(\rho)$ be the distance flown at the ground speed $VQ_{i-1}(\rho)$, $S12_i(\rho)$ the distance required to effect the speed change from $VQ_{i-1}(\rho)$ to $VP_i(\rho)$, and $S22_i(\rho)$ the distance flown at the ground speed $VP_i(\rho)$ (fig. 12). Since any speed change is assumed to occur during straight flight, the following equality must hold

$$S11_i(\rho) + S12_i(\rho) + S22_i(\rho) = D_i \tag{81}$$

The distance $S12_i(\rho)$ is merely a function of the two ground speeds involved and is given by

$$S12_i(\rho) = \frac{VP_i{}^2(\rho) - VQ_{i-1}^2(\rho)}{2 \cdot a_{min}} \tag{82}$$

The distance $S22_i(\rho)$ is then defined in terms of ρ as

$$S22_i(\rho) = \rho \cdot [D_i - S12_i(\rho)] \tag{83}$$

The flight times associated with each of the above three distances can be calculated as

$$T11_i(\rho) = \frac{S11_i(\rho)}{VQ_{i-1}(\rho)} = \frac{D_i - S12_i(\rho) - S22_i(\rho)}{VQ_{i-1}(\rho)} \tag{84}$$

$$T12_i(\rho) = \frac{VP_i(\rho) - VQ_{i-1}(\rho)}{a_{min}} \tag{85}$$

$$T22_i(\rho) = \frac{S22_i(\rho)}{VP_i(\rho)} \tag{86}$$

In order to compute the total flight time from point Q_{i-1} to point Q_i, it remains to determine the time $T23_i(\rho)$ spent in the turn from point P_i to point Q_i. A detailed discussion of the general problem of calculating the flight time during curved flight is quite involved (see ref. 8). The problem considered here is simplified somewhat by the assumption that the air speed is held constant during the turn. It is shown in reference 8 that if the desired constant air speed during the turn $\Delta\psi_i$ (see ground track generation) is $VA_i(\rho)$, then the duration of the turning flight, $T23_i(\rho)$, is given by

$$T23_i(\rho) = \frac{2 \cdot R_i \cdot SGN(\Delta\psi_i)}{c_2} \left\{\tan^{-1}\left[\frac{c_2}{c_1} \tan\left(\frac{\psi_i - \psi_w + \Delta\psi_i}{2}\right)\right] - \tan^{-1}\left[\frac{c_2}{c_1} \tan\left(\frac{\psi_i - \psi_w}{2}\right)\right]\right\} \tag{87}$$

where

$$c_1 = VA_i(\rho) + v_w \tag{88}$$

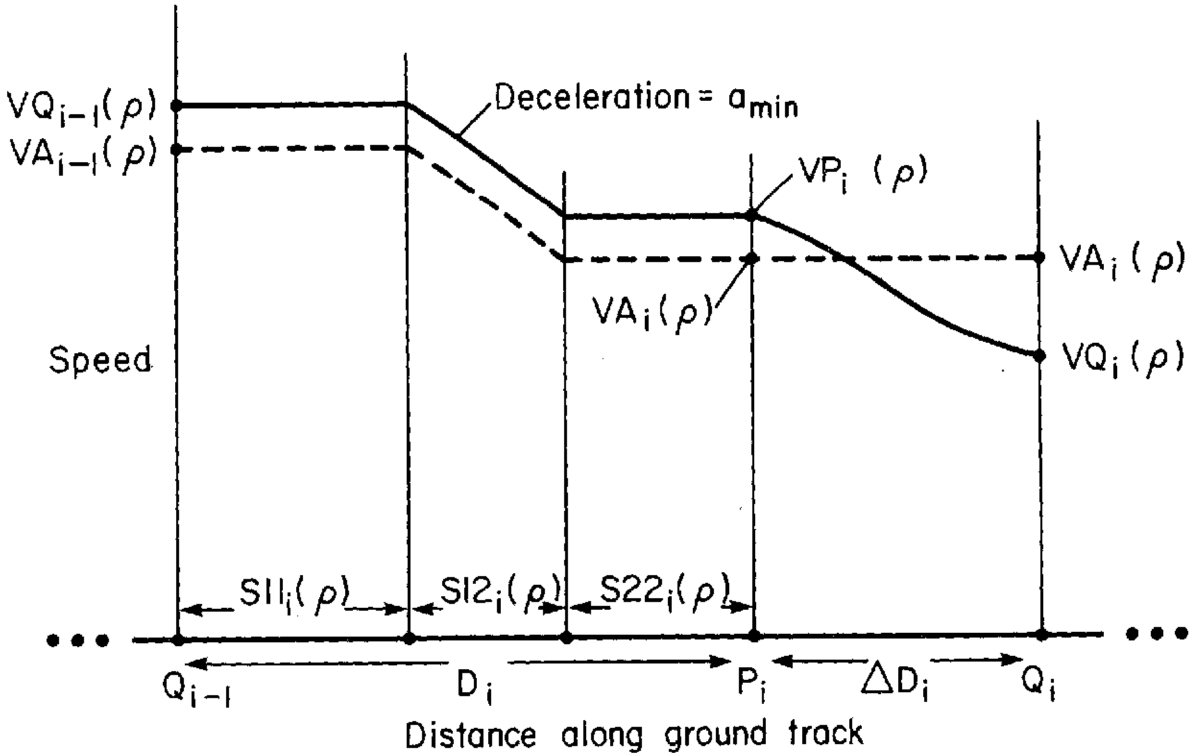


Figure 12.— Speed profile synthesis.

$$c_2 = [VA_i{}^2(\rho) - v_w{}^2]^{1/2} \tag{89}$$

Note that the ground speed is generally not constant during a turn, so that $VP_i(\rho)$ and $VQ_i(\rho)$ are different. For the sake of flexibility and modularity, a separate routine, TTTURN, was written to calculate $T23_i(\rho)$ (see the appendix).

The minimum and maximum flight times can now be generated from any waypoint to the last waypoint. Given any waypoint i, $i = 2, 3, \ldots, NWP$, and a ρ satisfying inequalities (77), the corresponding flight time from point Q_{i-1} to point Q_i is denoted by $T_i(\rho)$. From the above discussion, $T_i(\rho)$ is given by

$$T_i(\rho) = T11_i(\rho) + T12_i(\rho) + T22_i(\rho) + T23_i(\rho) \tag{90}$$

and is an output of the subroutine TRANS. Clearly, $\rho = 0$ results in the maximum speed profile between the two adjacent waypoints involved, while $\rho = 1$ yields the minimum speed profile between the waypoints. Thus, $T_i(0)$ is the minimum flight time and $T_i(1)$ is the maximum flight time between points Q_{i-1} and Q_i.

The definitions of $TMIN_i$ and $TMAX_i$ require that

$$TMIN_{NWP} = TMAX_{NWP} = 0 \tag{91}$$

For $i < NWP$, the flight times $TMIN_i$ and $TMAX_i$ are computed sequentially, but in reverse order. Thus, subroutine TRANGE starts with $i = NWP$, sets $\rho = 0$ or 1, calls the subroutine TRANS to compute $T_i(0)$ or $T_i(1)$, and then generates $TMIN_{i-1}$ and $TMAX_{i-1}$ according to

$$TMIN_{i-1} = TMIN_i + T_i(0) \tag{92}$$

$$TMAX_{i-1} = TMAX_i + T_i(1) \tag{93}$$

Desired Speed Profile

Once the minimum and maximum flight times from the first to the last waypoint are known, that is, $TMIN_1$ and $TMAX_1$, a desired flight time, denoted by $ENRTIM_1$, from the first to the last waypoint is selected. If the aircraft is to achieve the desired flight time while flying along the previously generated 3-D flight path without violating the speed constraints, $ENRTIM_1$ must satisfy the following inequalities:

$$TMIN_1 \leq ENRTIM_1 \leq TMAX_1 \tag{94}$$

Assuming that $ENRTIM_1$ is chosen such that inequalities (94) are satisfied, the next step is to generate a speed profile which results in a flight time equal to $ENRTIM_1$. Let such a speed profile be called the desired speed profile, with the following characteristics:

1. The airspeed remains between the minimum and maximum admissible air speeds along the flight path.

2. The rate of speed change does not exceed the acceleration/deceleration capabilities of the aircraft.

3. The ground speed yields the desired flight time between the first and last waypoints.

From earlier developments in this section, generating the desired speed profile is equivalent to finding the appropriate value for the single parameter ρ. This particular parameterization scheme of the speed profile by a single parameter was chosen to simplify the computations, thereby reducing computer storage and time requirements. In certain situations, this scheme may not be appropriate. Thus, it may be necessary to use two parameters, one to reflect the speed level between the minimum and maximum admissible speeds, the other to determine the points along the ground track at which speed changes occur.

Conceptually, the problem of determining the appropriate ρ is straightforward. In practice, however, an iterative procedure must be used because the functional relationship between ρ and the corresponding flight time is not invertible. Thus, if the flight time from the first waypoint to the last one corresponding to any ρ satisfying equation (77) is denoted by $T(\rho)$, then the explicit expression for $T(\rho)$ is

$$T(\rho) = \sum_{i=2}^{NWP} T_i(\rho) \tag{95}$$

where $T_i(\rho)$ are the output of the subroutine TRANS and are given by equation (90). Consequently, the problem of generating the desired speed profile is reduced to finding the value of ρ, say $\hat{\rho}$, that satisfies the equation

$$\sum_{i=2}^{NWP} T_i(\hat{\rho}) = ENRTIM_1 \tag{96}$$

Iterative Procedure to Find $\hat{\rho}$

Step 1— Let $j = 1$ and select the initial value of ρ_j to be

$$\rho_j = \frac{ENRTIM_1 - TMIN_1}{TMAX_1 - TMIN_1} \tag{97}$$

Step 2— Compute $T(\rho_j)$ according to equation (95), namely,

$$T(\rho_j) = \sum_{i=2}^{NWP} T_i(\rho_j) \tag{98}$$

Define $ERROR_j$ as

$$ERROR_j = ENRTIM_1 - T(\rho_j) \; , \tag{99}$$

and let ε be an arbitrary positive number specified in advance.

Step 3— If $ERROR_j$ satisfies the stopping criterion

$$|ERROR_j| < \varepsilon \tag{100}$$

then $\hat{\rho} = \rho_j$ and the iterative procedure stops. If equation (100) is not satisfied, then ρ_{j+1} is set equal to

$$\rho_{j+1} = \frac{1}{2}\left\{SAT\left[2\left(\rho_j + \frac{ERROR_j}{TMAX_1 - TMIN_1} - \frac{1}{2}\right)\right] + 1\right\} \tag{101}$$

and steps 2 and 3 are repeated. The function $SAT(\cdot)$ used above is defined as

$$SAT(\cdot) = \begin{cases} 1 & \text{if} \quad (\cdot) \geq 1 \\ (\cdot) & \text{if} \quad -1 < (\cdot) < 1 \\ -1 & \text{if} \quad (\cdot) \leq -1 \end{cases} \tag{102}$$

Defining ρ_{j+1} recursively by equation (101) assures that, if $T(\rho_j) < ENRTIM_1$, then $\rho_{j+1} > \rho_j$; if $T(\rho_j) > ENRTIM_1$, then $\rho_{j+1} < \rho_j$; and if $T(\rho_j) = ENRTIM_1$, then $\rho_{j+1} = \rho_j$. Furthermore, equation (101) assures that $0 \leq \rho_{j+1} \leq 1$ for all $j = 1, 2, \ldots$. Although it has not been proved, it is strongly suspected that $T(\rho)$ as given by equation (95) is a monotonic function of ρ. Consequently, the iterative procedure is expected to converge in a small number of iterations. Computational experience indicates that this is indeed the case.

Having thus found the value of $\hat{\rho}$ that satisfies equation (96), it is a simple matter to determine the desired speed profile. The required air speeds at the various waypoints are given by equations (78), the corresponding ground speeds are generated by equations (79) and (80), and the points at which speed changes occur are derived from equation (83); in all cases ρ is replaced by $\hat{\rho}$. The desired speed profile is therefore completely determined.

It will be useful to generate a sequence of NWP numbers that correspond to the times required to fly from the various waypoints to the last waypoint while tracking the 4-D flight path. More precisely, let $ENRTIM_i$, $i = 1, 2, \ldots, NWP$ denote the time required to fly from the end of the turn associated with waypoint i, namely, Q_i, to the end of the turn associated with waypoint NWP, namely, Q_{NWP}, while tracking the desired speed profile computed previously. From the definitions of $T_i(\rho)$ and $\hat{\rho}$, it follows that $ENRTIM_i$ can be computed by the recursive relationship:

$$ENRTIM_i = ENRTIM_{i-1} - T_i(\hat{\rho}) , \qquad i = 2, 3, \ldots, NWP \tag{103}$$

where $ENRTIM_1$ is specified as an external input. Clearly, $ENRTIM_i$ satisfies the following inequalities:

$$TMIN_i \le ENRTIM_i \le TMAX_i , \qquad i = 1, 2, \ldots, NWP \tag{104}$$

All computations described in this section have been implemented in the form of a subroutine called SPEED. A detailed programming description of SPEED is given in the appendix.

4-D GUIDANCE COMMANDS

In order to generate the sequence of 4-D guidance commands corresponding to the desired ground track, altitude profile, and speed profile computed in the three previous sections, it is necessary to interleave the individual command components in chronological order. This interleaving process is accomplished conveniently in the subroutine SPEED immediately following the speed profile computations of the previous section.

Before giving a detailed description of generating the chronological sequence of guidance commands, it is useful to recall the basic structure of each of the three components — ground track, altitude profile, and speed profile. The ground track of any flight path (as computed under Ground Track Synthesis) consists of a sequence of alternating straight lines and circular arcs (some of the turns, of course, may be degenerate). Thus, a straight-line flight followed by a circular turn can be considered as the basic element or "building-block" of the ground track. This basic element is completely defined by the following parameters: the x,y coordinates of the beginning of the straight line, the heading and length of the straight line, and the direction, angular extent, and radius of the turn. Given these parameters, the x,y coordinates of the beginning and end of the turn are uniquely determined and can be computed easily. From the definition of the basic element, it is clear that the number of basic elements in a complete ground track is equal to the number of waypoints.

The altitude profile (as described under Altitude Profile) consists of piecewise constant flight-path-angle segments between the beginnings of straight-line flights. Each segment can be considered as the basic element of the altitude profile. This is particularly appropriate since, in this case, the basic element of the altitude profile coincides with that of the ground track, resulting in a simple, well-defined basic element for the 3-D flight path. The parameters that define each element of the altitude profile are the z coordinate of the beginning of the segment and the sense and magnitude of the constant flight-path angle. Since the ground track is already known, the z coordinate at the end of the segment, as well as any other point, can be computed. The expression for the z coordinate of the end of the straight flight segment is given under Altitude Profile.

Finally, recall from the previous section that the most general speed profile consists of at most four separate subsegments for each basic element of the 3-D flight path, that is, from the beginning of one straight line flight to the beginning of the next one. The first three subsegments occur in straight flight: the first subsegment is a constant speed flight, during the second subsegment the speed is changed at a constant rate, and the third subsegment is again a constant speed flight at the new speed. The fourth subsegment occurs during the curved flight, and the speed profile is such that a constant air speed is maintained during the turn. These four subsegments can be considered as the basic unit of the speed profile. A typical unit, such as the one just described, is shown in figure 12. Due to the particular technique of generating the speed profile, the only parameters needed to compute the four subsegments are the minimum and maximum admissible air speeds at the waypoints, the direction and magnitude of the wind, and the parameter ρ that reflects how close the actual speed level is to the maximum admissible level.

Each basic element of the 3-D flight path contains one to four basic elements of the speed profile. Consequently, the entire sequence of 4-D guidance commands is decomposed into subsequences, each of which describes the desired motion of the aircraft from one waypoint to the next (actually, by convention, a subsequence starts at the beginning of a straight flight segment and ends at the beginning of the next straight flight segment).

Guidance Vectors

It is now evident that the guidance commands are piecewise constant during individual guidance intervals. Let Δt_k denote the duration of the kth guidance interval, and let the corresponding constant guidance commands during the kth interval be represented by a_k, u_k, and γ_k, where a_k is the rate of change of air speed, u_k is the inverse of the turning radius (see equation (11)), and γ_k is the flight-path angle. Thus, the 4-D commands for the kth guidance interval can be represented by a constant 4-vector, f_k, of the form:

$$f_k = [\Delta t_k, a_k, u_k, \gamma_k] \tag{105}$$

In order to assemble the chronological sequence of guidance commands, it is necessary to determine the sequence of constant 4-vectors given by equation (105). This is accomplished in the subroutine SPEED using various parameters of the ground track, altitude profile, and speed profile.

Assume that the command vectors f_k have been determined from waypoint 1 to waypoint $i-1$ or, more precisely, from the beginning of the first straight-line segment to the end of the turn associated with waypoint $i-1$, that is, point Q_{i-1}. Furthermore, assume that the number of nondegenerate command vectors for this portion of the flight path is $(NCI_{i-1} - 1)$, where a command vector f_k is said to be nondegenerate if the command interval Δt_k is nonzero. (Actually, from a computational point of view, f_k is considered nondegenerate if $\Delta t_k \geq \varepsilon$, where ε is an arbitrarily small positive number. In the computer implementation of the guidance system, ε is taken to be 0.1 second.) The numbers NCI_j, $j = 1, 2, \ldots, NWP$ play an important role in the overall guidance scheme and will be discussed later.

Letting $k = NCI_{i-1}$, the nondegenerate guidance vectors from point Q_{i-1} to point Q_i can now be easily determined by the following sequence of operations:

$$\left.\begin{array}{lll}
\text{Step 1} - \text{If } T11_i(\hat{\rho}) \geq \varepsilon, \text{ set} & f_k = [T11_i(\hat{\rho}), 0, 0, GAM_i], & \\
 & k = k + 1, \text{ and} & \text{go to step 2;} \\
\qquad \text{otherwise,} & & \text{go to step 2.} \\[1ex]
\text{Step 2} - \text{If } T12_i(\hat{\rho}) \geq \varepsilon, \text{ set} & f_k = [T12_i(\hat{\rho}), a_{min}, 0, GAM_i], & \\
 & k = k + 1, \text{ and} & \text{go to step 3;} \\
\qquad \text{otherwise,} & & \text{go to step 3.} \\[1ex]
\text{Step 3} - \text{If } T22_i(\hat{\rho}) \geq \varepsilon, \text{ set} & f_k = [T22_i(\hat{\rho}), 0, 0, GAM_i], & \\
 & k = k + 1, \text{ and} & \text{go to step 4;} \\
\qquad \text{otherwise,} & & \text{go to step 4.} \\[1ex]
\text{Step 4} - \text{If } T23_i(\hat{\rho}) \geq \varepsilon, \text{ set} & f_k = \left[T23_i(\hat{\rho}), 0, \dfrac{SGN(\Delta\psi_i)}{R_i}, GAM_i\right], & \\
 & k = k + 1, \text{ and} & \text{go to step 5;} \\
\qquad \text{otherwise,} & & \text{go to step 5.} \\[1ex]
\text{Step 5} - NCI_i = k. & &
\end{array}\right\} \tag{106}$$

From (106), it is clear that $NCI_i - NCI_{i-1} \leq 4$ and therefore the 4-D flight path between any two adjacent points Q_{i-1} and Q_i can be described by at most four nondegenerate guidance vectors of the form of equation (105). This in turn, implies that the maximum number, MAXSEG, of nondegenerate guidance vectors needed to generate a complete 4-D flight path satisfies the inequality

$$\text{MAXSEG} \leq 4 \cdot (\text{NWP} - 1) \tag{107}$$

In view of (106), the number NCI_i can be interpreted as a pointer in the following sense. Suppose that all nondegenerate guidance vectors for a particular 4-D flight path have been determined via the operations defined by (106). Then the first guidance vector immediately following waypoint i (actually, point Q_i) is f_{NCI_i}. Thus, if for some reason the only 4-D guidance vectors of interest are those starting at waypoint i, the pointer NCI_i makes it possible to identify immediately the correct starting vector. For reasons which will become evident in the next section, the sequence $\{NCI_i\}$ is initialized by setting $NCI_1 = 5$.

Once the guidance vectors are known, they can be applied in two different ways for tracking the 4-D flight path. If the control surfaces of the aircraft are coupled to the autopilot/autothrottle, then the guidance vectors could be used as input to the autopilot/autothrottle, which, in turn, would generate the appropriate signals to drive the control surfaces and the throttle. This would allow one to track a 4-D flight path completely automactically. On the other hand, the guidance vectors could be used to drive the flight director and other displays in the cockpit, thereby enabling the pilot to track the 4-D flight path manually. Various mixed modes of operation would also be feasible, for example, the throttle could be driven automatically, while the remaining tracking functions could be performed by the pilot.

The two main tracking options described previously have been incorporated in the design of an automated guidance and control system for STOL aircraft (ref. 5). The system has been tested extensively in the STOLAND flight simulator at Ames Research Center, and the manual tracking option has been tested in flight.

Compensation for Finite Roll and Pitch Rates

Throughout the present analysis, it was assumed that aircraft can make changes in bank angle and flight-path angle instantaneously. This is a reasonable assumption for the purposes of generating 4-D flight paths that may be orders of magnitude longer (in time) than the time constants of the controls involved. However, in order to achieve very precise tracking via the guidance vectors described earlier, it is necessary to account, in some fashion, for the fact that both the roll and pitch rates of aircraft are finite. A simple yet effective means of compensating for these finite rates is by introducing appropriate lead times in the guidance vectors. Thus, if the rolling and pitching maneuvers are initiated slightly in advance of the times dictated by the guidance vectors defined by (106), then the tracking errors can be minimized. It will be shown in the sequel that very simple expressions can be derived for the proper lead times, and these lead times depend only on the maximum control signal (bank angle and flight-path angle) and the maximum control rates (roll and pitch rates).

Roll-Rate Compensation

Assume that the aircraft is flying along a straight line until at some instant of time, say t_1, a new guidance vector is applied that requires the aircraft to roll to its maximum permissible bank angle ϕ_{max}. In generating the guidance vectors, it was assumed that the aircraft is capable of rolling to ϕ_{max} instantaneously at t_1. This is clearly not true since the maximum roll rate of aircraft in generally limited. Denote this maximum roll rate by $\dot{\phi}_{max}$ and assume that all banking maneuvers are performed at this maximum rate. Thus, if the roll begins at time t_o, then the bank angle, as a function of time, is given by

$$\phi(t) = \dot{\phi}_{max} \cdot (t - t_o) \tag{108}$$

Let t_2 be the earliest time the aircraft achieves ϕ_{max}. Then the minimum time required to effect a bank-angle change from 0 to ϕ_{max} is

$$(t_2 - t_o) = \frac{\phi_{max}}{\dot{\phi}_{max}} \tag{109}$$

The specific question therefore is: at what time, t_o, should a banking maneuver of the form of equation (108) begin if the sequence of idealized 4-D guidance vectors requires an instantaneous bank-angle change from 0 to ϕ_{max} at time t_1? In order to transform the question into a well-defined problem, the following boundary condition is imposed: at the instant, t_2, when $\phi(t)$ (eq. (108)) achieves ϕ_{max}, the heading should be the same as if ϕ_{max} had been achieved instantaneously at t_1 and maintained until t_2. These two bank-angle histories are shown in figure 13(a).

Using equation (108) in equation (4) yields the following expression for the heading at time t_2:

$$\psi(t_2) = \psi(t_o) - \frac{g}{V\dot{\phi}_{max}} \ln\left\{\cos[\dot{\phi}_{max} \cdot (t_2 - t_o)]\right\} \tag{110}$$

Similarly, if the instantaneous bank-angle change of ϕ_{max} were applied at t_1 and maintained until t_2, then the reference heading, $\psi_r(t_2)$, at time t_2 would be

$$\psi_r(t_2) = \psi(t_o) + \frac{g}{V} (\tan \phi_{max})(t_2 - t_1) \tag{111}$$

Let τ_ϕ be the lead time, that is, $\tau_\phi = (t_1 - t_o)$. Then equating $\psi(t_2)$ to $\psi_r(t_2)$ and using equation (109) yields the following for τ_ϕ:

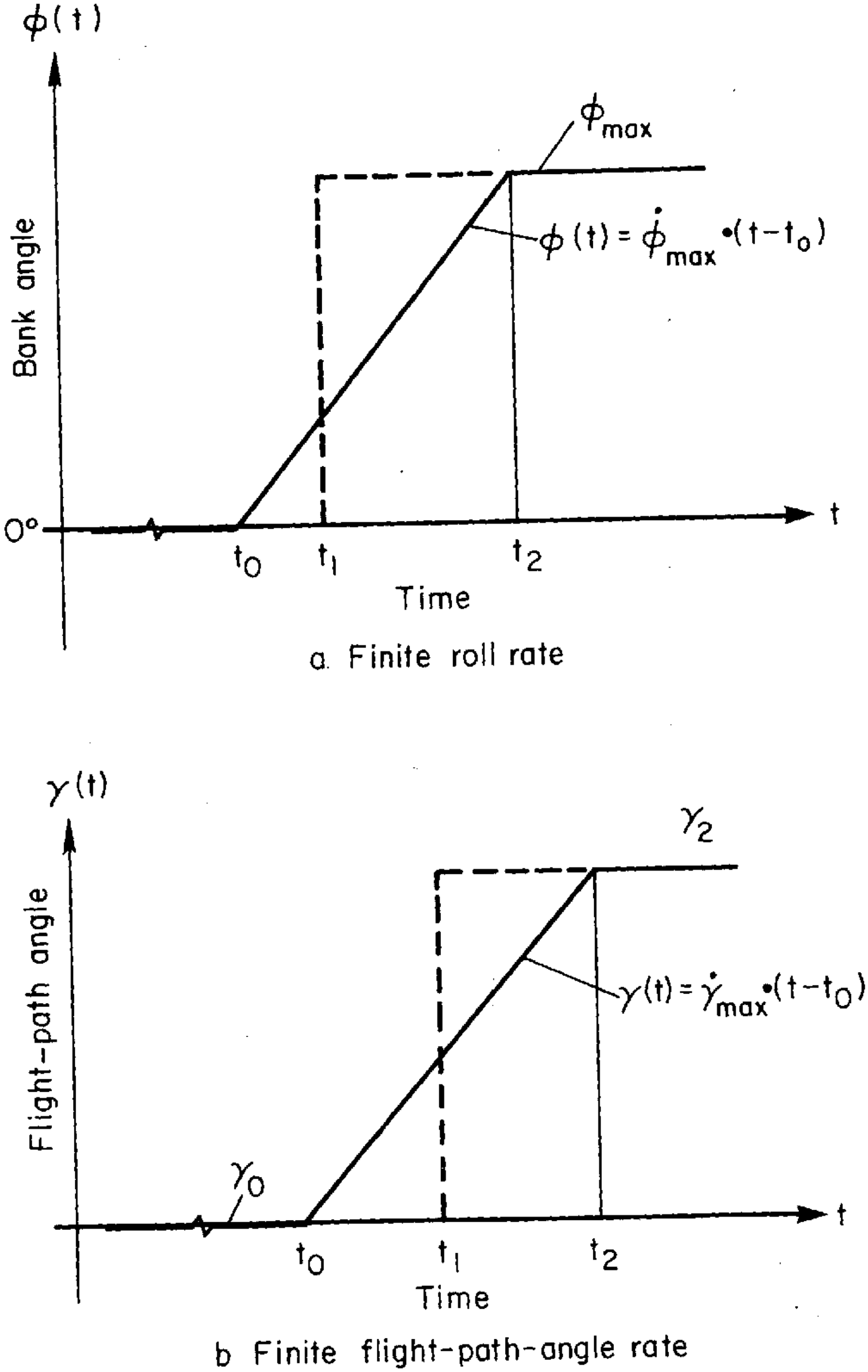


Figure 13.— Compensating for finite roll and pitch rates.

$$\tau_\phi = \frac{\phi_{max}}{\dot{\phi}_{max}} + \frac{\ln(\cos\phi_{max})}{\dot{\phi}_{max}\tan\phi_{max}} \tag{112}$$

Since, for the most terminal area operations, $\phi_{max} \le 30°$, the following approximations are valid:

$$\ln(\cos\phi_{max}) \approx -\frac{\phi_{max}^2}{2} \tag{113}$$

$$\tan\phi_{max} \approx \phi_{max} \tag{114}$$

and the resulting expression for τ_ϕ is

$$\tau_\phi = \frac{\phi_{max}}{2\dot{\phi}_{max}} \tag{115}$$

(Note that the lead time given by equation (115) is equally applicable when the aircraft is to roll out of a turn, that is, when the bank angle changes from ϕ_{max} to 0.) A typical value of $\dot{\phi}_{max}$ may be 5 degrees/second, and therefore the lead times required for banking maneuvers may be of the order of 2 to 3 seconds.

Pitch-Rate Compensation

The derivation of the appropriate lead time to compensate for finite pitch rate is analogous to the derivation of τ_ϕ. The essential difference is that the boundary condition imposed on the problem is in terms of the altitude of the aircraft.

Referring to figure 13(b), let the flight-path angle $\gamma(t)$ be given by

$$\gamma(t) = \gamma_0 + \dot{\gamma}_{max}(t - t_o) \tag{116}$$

where $\dot{\gamma}_{max}$ is the maximum permissible pitch rate. Furthermore, let τ_γ be the lead time defined by $\tau_\gamma = t_1 - t_o$. The problem is to determine τ_γ subject to the following boundary condition: at the instant t_2 when $\gamma(t)$ (as given by eq. (116)) becomes γ_2, the altitude should be the same as if γ_2 had been achieved instantaneously at t_1 and maintained until t_2 (fig. 13(b)).

Using equation (116) in (3) yields the following expression for the altitude at time t_2 (assuming constant speed):

$$z(t_2) = z(t_o) + v\gamma_o \cdot (t_2 - t_o) + \frac{1}{2} v\dot{\gamma}_{max} \cdot (t_2 - t_o)^2 \tag{117}$$

If the flight-path angle were changed instantaneously from γ_o to γ_2 at t_1 and maintained until t_2, then the reference altitude, $z_r(t_2)$, at time t_2 would be

$$z_r(t_2) = z(t_o) + v\gamma_o \cdot (t_1 - t_o) + v\gamma_2 \cdot (t_2 - t_1) \tag{118}$$

Equating (117) to (118) and using the fact that $(t_2 - t_o) = (\gamma_2 - \gamma_o)/\dot{\gamma}_{max}$, yields the following for τ_γ

$$\tau_\gamma = \frac{\gamma_2 - \gamma_o}{2\dot{\gamma}_{max}} \tag{119}$$

In practice, the admissible maneuvers in the vertical plane are limited not so much by the maximum pitch rate of the aircraft, but by the vertical acceleration due to the pitching motion. Assuming that flight-path angle changes take place at constant speed, the maximum vertical acceleration, $\ddot{z}_{max}$, is obtained from equation (3) by simple differentiation:

$$\ddot{z}_{max} = v\dot{\gamma}_{max} \tag{120}$$

Combining equations (119) and (120) yields

$$\tau_\gamma = \frac{v \cdot (\gamma_2 - \gamma_o)}{2\ddot{z}_{max}}$$

An accepted value of $\ddot{z}_{max}$ is 0.68 m/sec^2 (2.25 ft/sec^2), so that, for typical terminal area speeds and flight-path angles, τ_γ is of the same order of magnitude as τ_ϕ.

Having thus obtained the lead times τ_ϕ and τ_γ, the timing of any 4-D guidance vector that requires a change in bank angle and/or flight-path angle is merely advanced by the appropriate lead time. Extensive simulations on the STOLAND flight simulator indicate that compensating for finite roll and pitch rates in the manner described above is more than adequate for accurate tracking of typical terminal area flight paths.

A particularly convenient place to compute the lead times τ_ϕ and τ_γ is in the subroutine SPEED. Details of these computations are discussed in the appendix.

CAPTURE MODE

The aim of any 4-D guidance scheme is to guide the aircraft accurately in space and in time. Specifically, the aircraft is to be guided along a precise 3-D flight path and a precise time schedule so that its arrival time to the last waypoint, and therefore to the runway threshold, can be predicted accurately. In order to accomplish this objective, not only must the aircraft be guided accurately from the first waypoint to the last waypoint, but also from its initial position in the terminal area to the first waypoint. Essentially, a 4-D flight path must be generated from the current state of the aircraft to the desired final state. The major difficulty in computing such a flight path is the fact that the current state of the aircraft, which would serve as the reference point for the computation, is continuously changing. Thus, rather than consider the current position of the aircraft as the first waypoint and contend with a nonstationary reference point, the segment from the current aircraft position to the first waypoint is treated separately. This part of the flight path is referred to as the capture flight path since it represents the maneuvers required to capture the first waypoint.

Two important advantages are derived from separating the capture flight path from the rest. First, the very complex and time-consuming computations needed to generate the 4-D flight path from the first to the last waypoint are performed usually once or, at most, a few times during the flight (the reasons for computing the 4-D flight path more than once are discussed later). Thus, valuable computer time is available for other guidance, control, and navigation functions. Secondly, since the capture flight path is relatively simple and is generated rapidly, manual path stretching and time control maneuvers can be performed by the pilot. Such maneuvers will be shown to be important elements of the overall 4-D guidance scheme.

Assume that the aircraft is at an arbitrary point in the terminal area, say point PA, with heading ΨA and airspeed VA (fig. 14). Furthermore, assume that the 4-D flight path from waypoint 1 to waypoint NWP has already been computed. Since the capability of capturing any one of the waypoints is a very useful feature, the point to be captured is denoted by QB rather than by Q_1. Note that this convention in notation anticipates the likelihood that the capture flight path, in general, will begin and end with a turn. Thus, QB may be the endpoint of the turn associated with any one of the waypoints (from now on, it is called the capture waypoint). The heading to be achieved at the capture waypoint is simply the heading of the straight-line segment following point QB, and is denoted by Ψc. Finally, the airspeed at the capture waypoint is denoted by VB. Thus, the objective is to generate a 4-D flight path that starts at PA with a heading ΨA and speed VA and ends at point QB with heading ΨC and speed VB.

The capture flight path satisfying the above end conditions is generated using the same basic logic used in previous sections. First, the ground track is computed using a straight flight segment and circular turns; then the altitude profile is determined and, finally, an appropriate speed profile is generated.

Ground Track Computation

Since minimum flight time is generally a desirable property of a flight path, the ground track is to have a minimum path length. The problem of computing the minimum path length ground track for arbitrary end conditions is

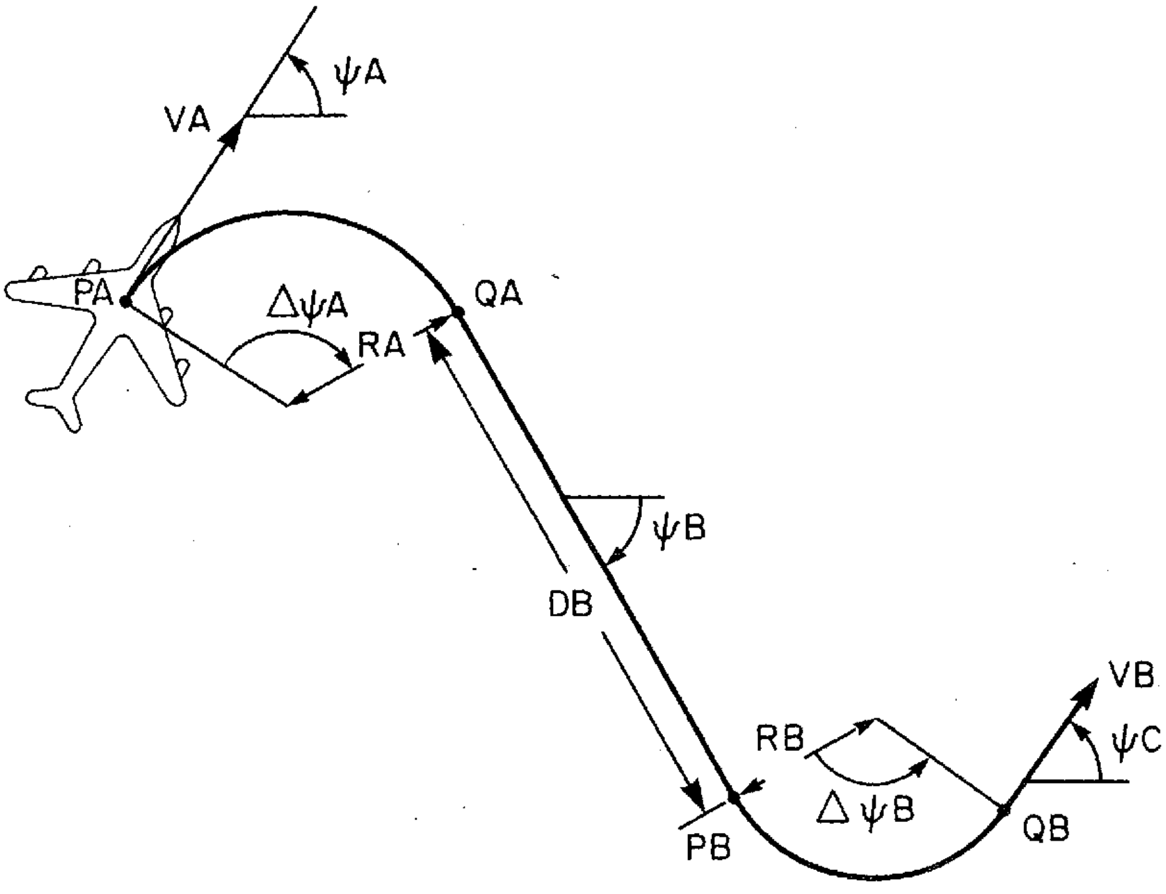


Figure 14.— Ground track of capture flight path.

quite complex (ref. 9). It was shown in reference 9 that the most general ground track of minimum length subject to a turning radius constraint consists of either three alternating circular turns or two circular turns separated by a straight-line flight. Rather than the complete algorithm developed in reference 9, only a simplified version is used here, according to which ground tracks with three alternating circular turns are ruled out. This simplification has two significant advantages. First, the substantial computational requirements to check for the existence of, and the actual determination of, a three-turn minimum length flight path for arbitrary end conditions are eliminated. It is shown in reference 9 that the regions of end conditions for which the minimum length ground track consists of three consecutive opposite turns are quite small and they usually occur when the two end points are very close. Eliminating this type of ground track therefore reduces the computational complexity without significantly affecting the minimum length property.

The second advantage of not considering three-turn ground tracks is related to pilot workload. When the 4-D flight path is to be flown completely manually, the workload involved in tracking accurately a 4-D flight path consisting of three consecutive, opposite turns appears to be excessive.

Thus the problem is reduced to finding a ground track that consists of a turn, followed by a straight flight, followed by another turn, such that the specified end conditions are met and the resulting ground track has minimum path length among all such ground tracks. Such a ground track for two arbitrary end conditions (shown in fig. 14) is generated by the following iterative procedure.

Iterative procedure— Let RA be the radius of the turn beginning at point PA (assuming a turn at PA is required), RB the radius of the turn ending at point QB (again assuming that a turn at QB is required), and define $\Psi C'$ as

$$\Psi C' = \mathrm{MOD}_{\pi}(\Psi C + \pi) \tag{121}$$

Now initialize the iterative procedure by setting $QA_0 = PA$ and $j = 1$.

Step 1. With subroutine NEWPSI, generate the minimum path length ground track starting at point QB with heading $\Psi C'$ and ending at point QA_{j-1} subject to the minimum turning radius RB. Under Problem Description and Basic Approach, it was shown that such a ground track consists of a circular turn of radius RB, followed by a straight-line segment to point QA_{j-1}. Denote the angular extent of the resulting turn by $\Delta\Psi B_j$, the end point of the turn by PB_j, and the heading of the straight line by $\Psi B_j'$.

Step 2. With subroutine NEWPSI, generate the minimum path-length ground track starting at point PA with heading ΨA and ending at point PB_j subject to the minimum turning radius RA. Again, such a ground track consists of a circular turn of radius RA, followed by a straight-line segment to point PB_j. Denote the angular extent of the resulting turn by $\Delta\Psi A_j$, the end point of the turn by QA_j, and the heading of the straight line by ΨB_j.

Step 3. Define the quantity TEST by

$$\text{TEST} = \pi - |\Psi B_j - \Psi B_j'| \tag{122}$$

and let ε be an arbitrarily small positive number. Then, if $|\text{TEST}| \geq \varepsilon$, set $j = j + 1$ and repeat steps 1, 2, and 3; if $|\text{TEST}| < \varepsilon$, the iterative procedure is considered to have converged, yielding

$$\left.\begin{aligned} \Delta\Psi A &= \Delta\Psi A_j \\ QA &= QA_j \\ \Psi B &= \Psi B_j \\ PB &= PB_j \\ \Delta\Psi B &= -\Delta\Psi B_j \end{aligned}\right\} \tag{123}$$

Results of the first few iterations are shown graphically in figure 15 for two arbitrary end conditions. Extensive computational experience indicates that the iterative procedure converges very quickly, usually after two or three iterations.

Since at the time the iterative procedure is applied, the two end conditions can be arbitrary, it is conceivable that the type of turn-straight-turn ground track described above may not be feasible because of the close proximity of the end points. If this is the case, then at some stage of the iterative procedure the minimum length turn-straight ground track from either point QB to point QA_{j-1} or from point PA to point PB_j is not feasible. If this occurs, then an error message is generated by subroutine NEWPSI and computation stops.

Altitude Profile

The altitude profile for the capture flight path is assumed to consist of a single, constant flight-path-angle segment from the current aircraft altitude to the desired altitude at the capture waypoint. Assuming that PA is the current aircraft position and QB, the capture waypoint (fig. 14), the flight-path angle GAMB is given by

$$\text{GAMB} = \frac{\text{ZPA} - \text{ZQB}}{\Delta\text{DA} + \text{DB} + \Delta\text{DB}} \tag{124}$$

where ZPA and ZQB are the z coordinates of points PA and QB, respectively; ΔDA and ΔDB are the arclengths of turns $\Delta\Psi$A and $\Delta\Psi$B, respectively; and DB is the length of the straight flight from point QA to point PB (fig. 14). Note that equation (124) is the approximate expression for GAMB analogous to equation (64), rather than the exact expression involving the $\tan^{-1}(\cdot)$ function. Before GAMB (as given by eq. (124)) is considered acceptable, it must be checked to see whether it lies within the minimum and maximum

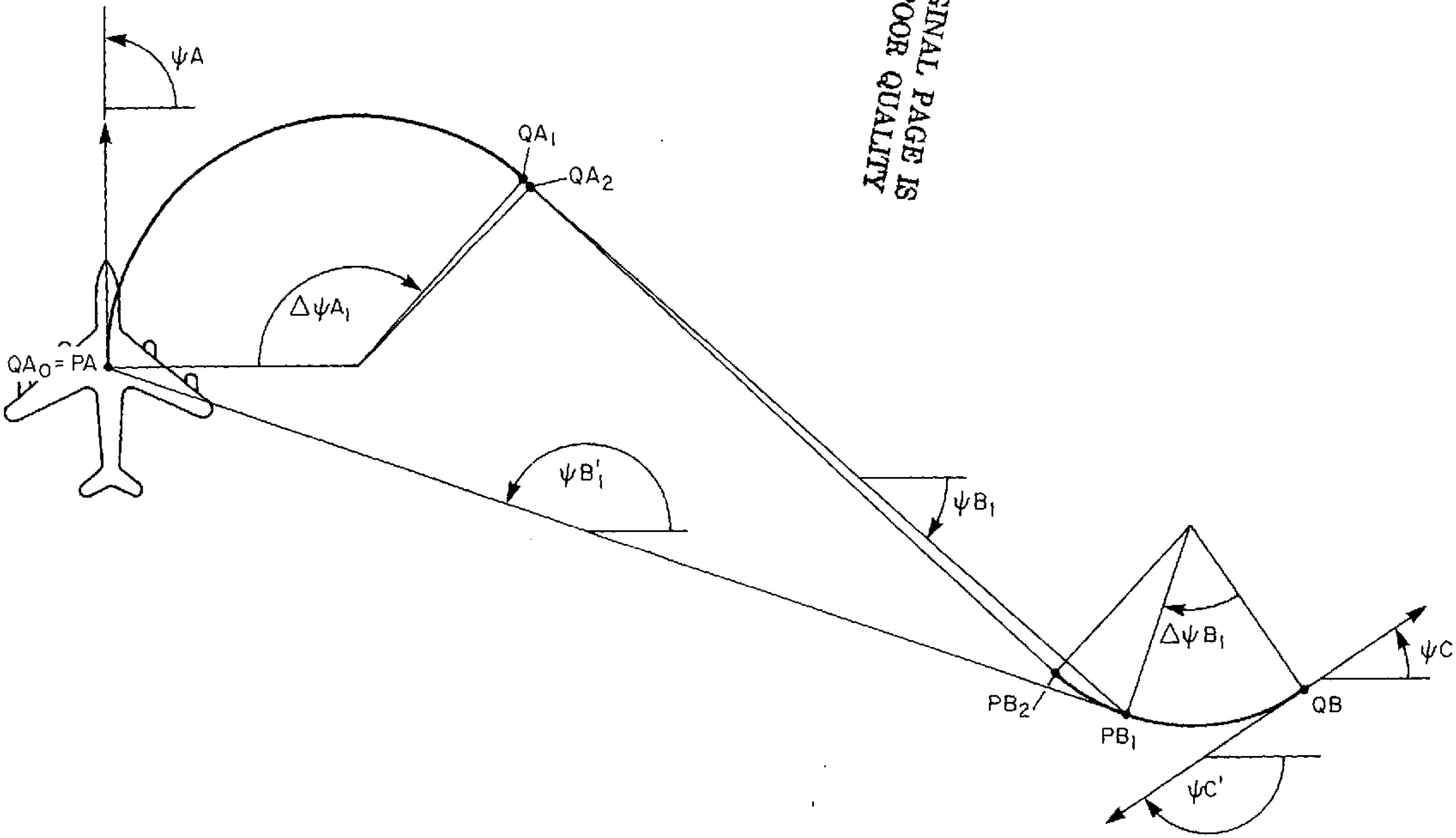


Figure 15.— Ground track synthesis for capture flight path.

admissible flight-path angles. If it lies outside the allowed range, then an error message is generated and computation stops.

Finally, the z coordinates of points QA and PB are determined using GAMB and some of the ground track parameters:

$$\left.\begin{aligned} ZQA &= ZPA - \Delta DA \cdot GAMB \\ ZPB &= ZQA - DB \cdot GAMB \end{aligned}\right\} \tag{125}$$

Figure 16 shows the altitude profile corresponding to the ground track of figure 14.

Speed Profile

For computational simplicity, the speed profile is assumed to consist of at most four segments. The initial turn from point PA to QA is flown at constant air speed VA; the second and third segments occur during the straight-line flight, the second one being a constant speed segment at speed VA and the third, a speed change segment from VA to VB at a constant rate. Finally, the fourth segment is a turn at constant air speed VB. Such a typical speed profile is depicted in figure 17, which shows both the air speed and ground speed profiles. The expression for the ground speeds at points QA and PB, namely, VQA and VPB, are

$$\left.\begin{aligned} VQA &= VA + v_w \cdot \cos(\Psi B - \psi_w) \\ VPB &= VB + v_w \cdot \cos(\Psi B - \psi_w) \end{aligned}\right\} \tag{126}$$

The distance SAB required to achieve the ground speed change from VQA to VPB is

$$SAB = \begin{cases} \dfrac{VPB^2 - VQA^2}{2 \cdot a_{min}}, & \text{if } VPB \le VQA \\ \dfrac{VPB^2 - VQA^2}{2 \cdot a_{max}}, & \text{otherwise} \end{cases} \tag{127}$$

Clearly, a speed profile of the structure described above is feasible only if

$$SAB \le DB \tag{128}$$

If equation (128) is not satisfied, then an error message is generated and computation stops.

Finally, let T1, T2, T3, and T4 denote the time duration of each of the four speed segments. T1 and T4, which correspond to the two turns, are computed by the use of subroutine TTTURN. Times T2 and T3 are given by

ORIGINAL PAGE IS
OF POOR QUALITY

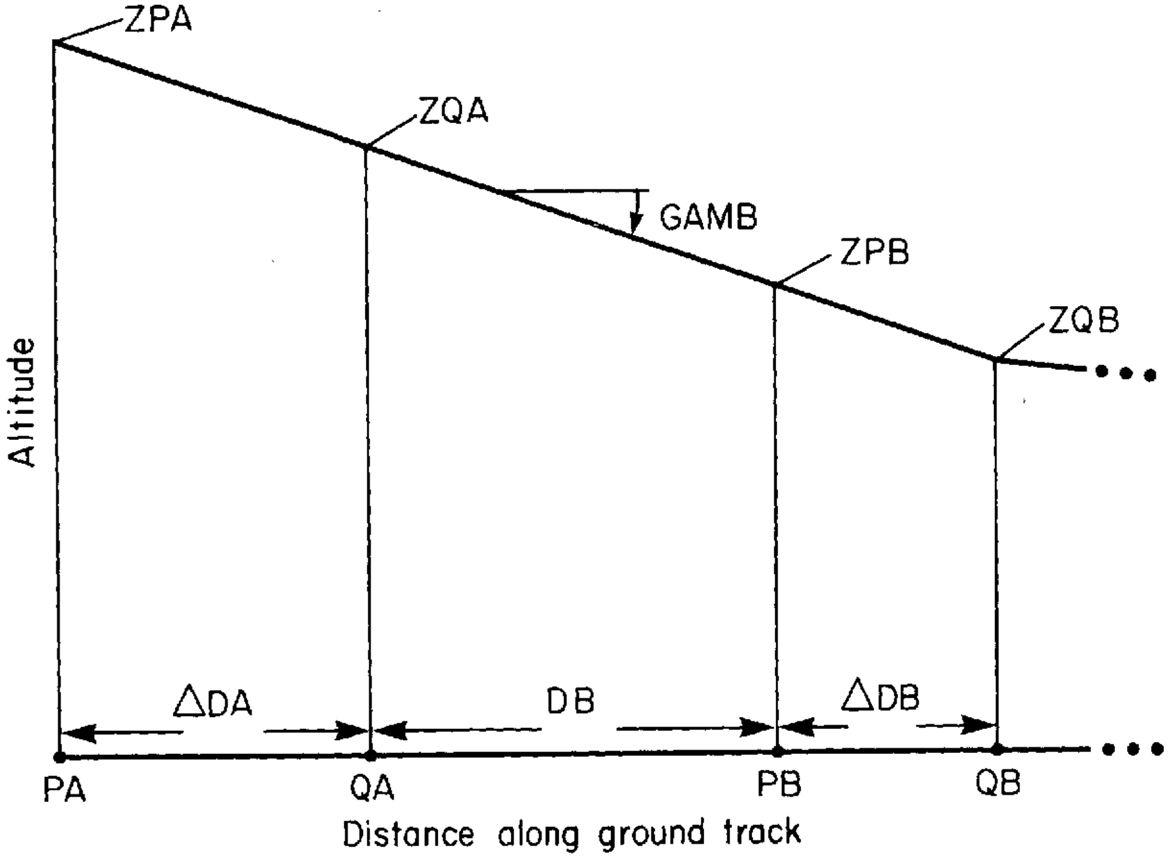


Figure 16.– Altitude profile synthesis for capture flight path.

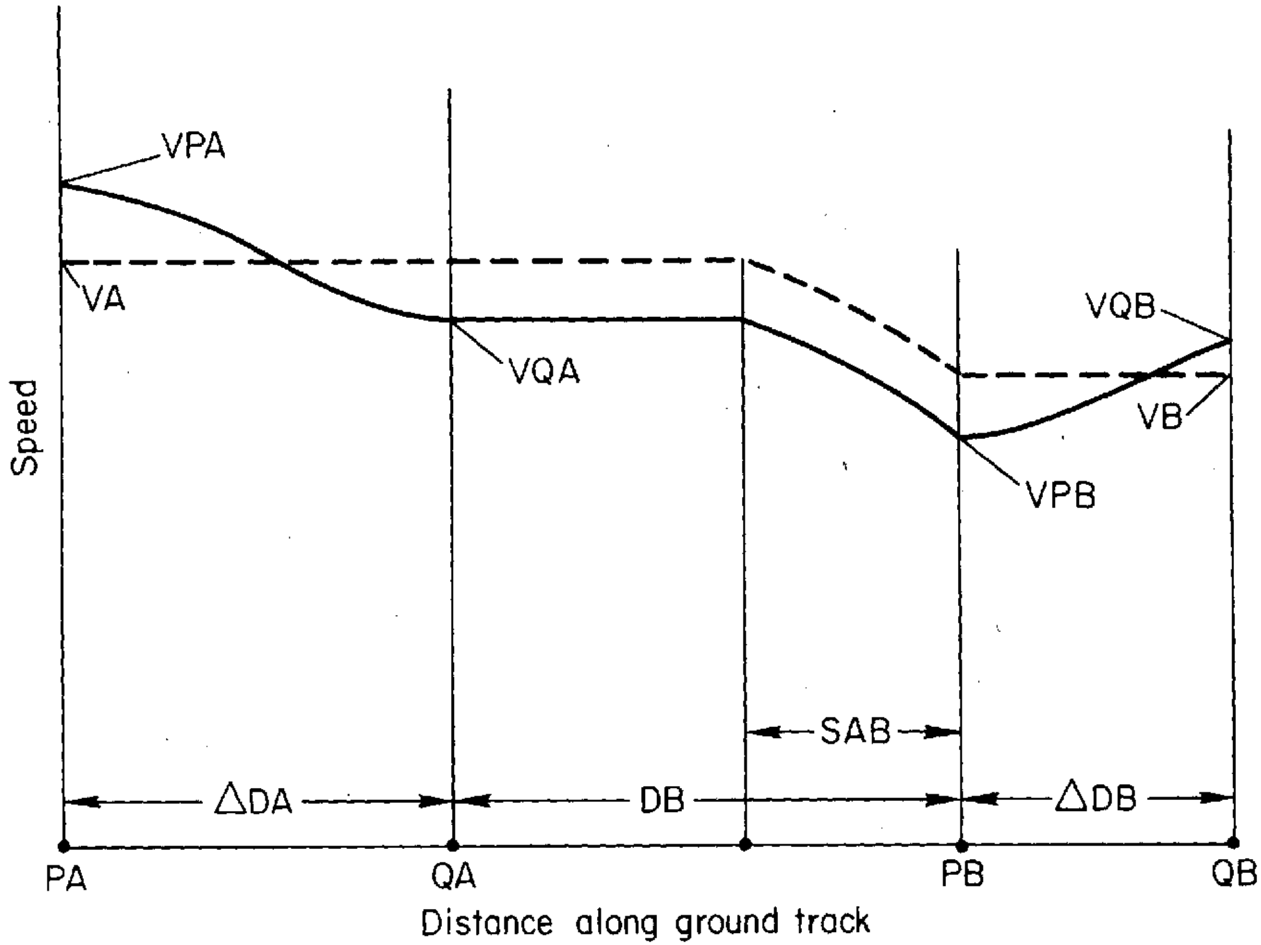


Figure 17.— Speed profile synthesis for capture flight path.

$$T2 = \frac{DB - SAB}{VQA} \tag{129}$$

$$T3 = \begin{cases} \dfrac{VPB - VQA}{a_{min}}, & \text{if } VPB \le VQA \\ \dfrac{VPB - VQA}{a_{max}}, & \text{otherwise} \end{cases} \tag{130}$$

4-D Commands

The 4-D commands for the capture flight path can be assembled in a manner similar to those for the flight path from the first to last waypoint described under Four-Dimensional Guidance Commands. At most four constant guidance vectors are needed, one for each of the speed segments described above. Let the four guidance vectors be f_k, k = 1, 2, 3, and 4. Letting k = 1 and using the same logic as in equation (106), the nondegenerate vectors f_k are determined by the following sequence of operation:

$$\left.\begin{array}{l}
\text{Step 1 — If } T1 \gtrsim \varepsilon, \text{ set } f_k = \left[T1,\ 0,\ \dfrac{SGN(\Delta\Psi A)}{RA},\ GAMB\right], \\
\qquad\qquad k = k + 1, \text{ and go to step 2;} \\
\qquad \text{otherwise, go to step 2.} \\
\text{Step 2 — If } T2 \gtrsim \varepsilon, \text{ set } f_k = [T2,\ 0,\ 0,\ GAMB], \\
\qquad\qquad k = k + 1, \text{ and go to step 3;} \\
\qquad \text{otherwise, go to step 3.} \\
\text{Step 3 — If } T3 \gtrsim \varepsilon, \text{ set } f_k = \left[T3,\ \dfrac{VPB^2 - VQA^2}{2 \cdot SAB},\ 0,\ GAMB\right], \\
\qquad\qquad k = k + 1, \text{ and go to step 4;} \\
\qquad \text{otherwise, go to step 4.} \\
\text{Step 4 — If } T4 \gtrsim \varepsilon, \text{ set } f_k = \left[T4,\ 0,\ \dfrac{SGN(\Delta\Psi B)}{RB},\ GAMB\right], \\
\qquad\qquad k = k + 1, \text{ and go to step 5;} \\
\qquad \text{otherwise, go to step 5.}
\end{array}\right\} \tag{131}$$

Step 5 — NCI = k - 1.

In (131), ε is the same small positive number as in (106), and NCI is the number of nondegenerate command intervals for the capture flight path. Clearly, NCI $\le$ 4. If TCAP denotes the time duration of the capture flight path, then TCAP is given by

$$TCAP = \sum_{k=1}^{NCI} \Delta t_k \tag{132}$$

The reason for starting the indexing of the guidance vectors f_k in the previous section with $k = 5$ should now be obvious. Letting the first four vectors in the sequence $\{f_k\}$ refer to the capture flight path (actually, there are only $NCI \leq 4$ nondegenerate vectors for the capture flight path) and the remaining vectors f_k, $k = 5, 6, \ldots, MAXSEG + 4$ refer to the flight path from the first to the last waypoint, the entire 4-D flight path can be characterized by the single sequence of guidance vectors $\{f_k\}$, $k = 1, 2, \ldots, NCI, 5, 6, \ldots, MAXSEG + 4$. This convention eliminates long and cumbersome recomputations of the guidance vectors that would otherwise be required during on-line use of the guidance system.

The computation of the capture flight path is implemented in the form of a subroutine called TST whose programming details are described in the appendix.

ON-LINE 4-D GUIDANCE

The two separate parts of the 4-D flight path can now be combined to yield a continuous, on-line guidance scheme from an arbitrary initial aircraft position, velocity, and heading to any desired final position, velocity, and heading. Assume that the 4-D flight path from the first to the last waypoint, as well as the 4-D capture flight path from the current aircraft position to the capture waypoint, have been computed. Denote the index of the capture waypoint by CWP, let TABS be the current value of absolute time and TOA the absolute time of arrival to the last waypoint. Then, assuming that the aircraft begins immediate tracking of the 4-D flight path generated from its current state through the capture waypoint to the final waypoint, the expression for arrival time to the last waypoint is

$$TOA = TABS + TCAP + ENRTIM_{CWP} \tag{133}$$

Equation (133) plays a crucial role in the precise time control of aircraft. First, the arrival time to any specified point, such as the outer marker or approach gate, can be predicted by the use of equation (133) long before the actual arrival. Thus, if the flight paths of several aircraft merge at a common point, the arrival time of each aircraft to the merge point is accurately predicted by equation (133). Consequently, potential conflicts can be detected early, long before last minute collision avoidance maneuvers would be required. Secondly, not only can the possible arrival time be predicted by use of equation (133), but it can also be changed in a highly controlled fashion. Once the aircraft is under 4-D control, a desired arrival time can be achieved in three ways:

1. Change the initial state of the aircraft by performing arbitrary path-stretching maneuvers and speed changes, thereby changing TCAP.

2. Generate a new speed profile along the 3-D flight path from the capture waypoint to the last waypoint, thereby changing $ENRTIM_{CWP}$.

3. Select a different capture waypoint, that is, a new value for CWP, thereby affecting both TCAP and $ENRTIM_{CWP}$.

The first of these is the principle method of making gross changes in the time of arrival. It is used primarily to delay arrival time since the 4-D capture flight path is already very nearly the minimum time flight path from the current aircraft position to the capture waypoint (unless the possibility of increasing the speed is considered). Thus, the arrival time can be delayed arbitrarily by appropriate path-stretching maneuvers. If the desired delay is sufficiently long, then these maneuvers generally take the form of holding patterns. By continuously recomputing the 4-D capture flight path from the current updated aircraft state to the capture waypoint, the value of TCAP is also updated. The attainable time of arrival from any point of the path stretching or holding maneuver to the last waypoint is then found simply by using the updated values of TABS and TCAP in equation (133). Assuming that TCAP, and therefore TOA, change in a sufficiently continuous manner as the aircraft performs the various maneuvers, any desired arrival time could be achieved by initiating the tracking mode precisely at the instant TOA becomes the desired time of arrival. (Although questions concerning the continuity of TCAP have not been rigorously investigated, computational experience indicates that TCAP is a sufficiently continuous function along the usual path-stretching and holding maneuvers; for further discussion, see ref. 9.)

The second method of changing the arrival time is useful when minor adjustments are desired. Recall that during the process of generating the desired speed profile it was necessary to compute the two arrays $TMIN_i$, $TMAX_i$, $i = 1, 2, \ldots, NWP$, which are the minimum and maximum feasible flight times from waypoint i to the last waypoint along the 3-D flight path. Furthermore, the actual flight time from each waypoint to the last one corresponding to the desired speed profile is also available, and is denoted by $ENRTIM_i$. Since inequalities (104) hold for all $i = 1, 2, \ldots, NWP$, it is clear that $ENRTIM_{CWP}$, and therefore TOA, could be increased or decreased by simply recomputing a new speed profile from the capture waypoint to the last waypoint. Denote the desired change in TOA by ΔTOA. Then the new desired time of arrival can be achieved using this second method if and only if ΔTOA satisfies the inequalities:

$$TMIN_{CWP} - ENRTIM_{CWP} \leq \Delta TOA \leq TMAX_{CWP} - ENRTIM_{CWP} \qquad (134)$$

If ΔTOA satisfies inequalities (134), then the new value of $ENRTIM_{CWP}$ is set equal to $ENRTIM_{CWP} + \Delta TOA$ and subroutine SPEED is called to generate the new desired speed profile.

Note that the application of this method to change the arrival time makes sense only if the aircraft is already tracking the capture flight path. This is necessary so that the precise effect on TOA of changing $ENRTIM_{CWP}$ can be predicted. Otherwise, it would be difficult to separate the effects on TOA of changing $ENRTIM_{CWP}$ and changing TCAP.

The third technique of changing TOA is to select an entirely new capture waypoint. This is perhaps the most general method of achieving a new arrival time. Clearly, once a new capture waypoint is selected, the first two methods are again applicable. The significance of the third technique lies in the fact that, in most cases, arrival time to the last waypoint can be advanced substantially by capturing waypoints that are closer and closer to the last waypoint (closeness, in this case, is measured by distance along the flight path). In fact, the absolute earliest possible arrival time could be achieved by setting CWP = NWP, that is, selecting the last waypoint as the capture waypoint. This follows from the near minimum time property of the capture flight path.

At this point, it seems appropriate to describe briefly the sequence of events that would occur as a result of using the 4-D guidance scheme presented here. Assume that an aircraft has just entered the terminal area and is about to be placed under 4-D control. The first step is to specify the necessary input parameters so that the 4-D flight path from the first to the last waypoint can be generated. This can be achieved in several different ways. One possibility is for the pilot to enter the input parameters into the onboard computer, which would then generate the 4-D flight path — more precisely, the 4-D guidance vectors — from the first to the last waypoint. Since there is no compelling reason to carry out this computation onboard the aircraft, another possibility is to compute the 4-D flight path from the first to the last waypoint in advance, perhaps even prior to takeoff, and store it in the onboard computer. (Actually, several alternative 4-D flight paths may be generated and stored; in each case, however, the last waypoint is assumed to be the outer marker or the approach gate or some other point along the final approach.) Whichever method is used, it is assumed that the 4-D flight path from the first to the last waypoint is available prior to engaging the online guidance system.

The next step is to engage the online system, which immediately generates the capture flight path from the current aircraft position to the first waypoint (assuming a feasible capture flight path exists). The two flight paths are then combined to form a complete 4-D flight path from current aircraft position to the last waypoint. Simultaneously, the would-be arrival time is computed and displayed to the pilot. As the aircraft continues its flight in the terminal area, the capture flight path and the projected arrival time are continuously recomputed to reflect the current initial state of the aircraft. At some point during the flight, Air Traffic Control specifies the desired arrival time for the aircraft. The pilot then initiates one of the three techniques described earlier to achieve the desired arrival time. When the would-be arrival time displayed in the cockpit coincides with that requested by ATC, he engages the tracking mode. Barring unforeseen circumstances, the aircraft then follows the 4-D flight path and reaches the last waypoint precisely at the requested arrival time.

It is conceivable that, at some point during the tracking mode, the aircraft may have to abandon the original arrival time, disengage the tracking mode, and attempt to meet a new arrival time. This change may be due to various reasons such as another aircraft given unshceduled priority to land, sudden change in weather over the airport, unforeseen runway congestion, collision avoidance maneuver, etc. If the request is by ATC, then the pilot would disengage the tracking mode, perform the necessary maneuvers to achieve the new arrival time, and then re-engage the tracking mode. If the change is due to a collision avoidance maneuver, then the tracking mode is again disengaged (this time automatically). After the conflict has been resolved, a new desired arrival time is specified and the pilot attempts to meet it using one of the three techniques presented earlier.

The 4-D guidance system presented here has been implemented, along with the necessary control law and executive functions, using the STOLAND avionics hardware-software package (ref. 10) at Ames Research Center. The system has undergone extensive testing in the STOLAND simulator, and it is currently awaiting actual flight tests in the CV-340 aircraft. The system is described further in reference 5.

AN ILLUSTRATIVE EXAMPLE

A simple example illustrates the ideas described here. The aircraft under consideration is a STOL-type aircraft subject to the following performance and operational constraints:

$$\left.\begin{aligned}
\phi_{max} &= 30^\circ\text{, maximum bank angle}\\
\gamma_{max} &= 15^\circ\text{, maximum flight-path angle}\\
\gamma_{min} &= -7.5^\circ\text{, minimum flight-path angle}\\
a_{max} &= 0.305\text{ m/sec}^2\text{, maximum acceleration}\\
a_{min} &= -0.305\text{ m/sec}^2\text{, minimum acceleration}\\
V_s(0) &= 45.72\text{ m/sec, stall speed with } \delta = 0^\circ \text{ flaps}
\end{aligned}\right\} \tag{135}$$

Further, it was assumed that during flight, when the aircraft is sufficiently far away from the outer marker, the admissible air speed may lie anywhere between $1.3V_s(0)$ and $1.7V_s(0)$, that is,

$$\underline{c} = 1.3 \ , \qquad \bar{c} = 1.7 \tag{136}$$

The geometry of the 3-D flight path is defined by six waypoints whose input parameters are (for this example, distances are in m, speeds in m/sec, and times in sec):

i	1	2	3	4	5	6
$INDEX_i$	1	0	0	0	1	1
XWP_i	2286	7010	7010	-5182	-5334	-2438
YWP_i	2438	2438	-2591	-2591	0	0
ZWP_i	-988	-988	-988	-988	-549	-244
R_i	---	1219	1219	---	1295	---

The desired final heading and final speed of the aircraft at the last waypoint are given by

$$\left.\begin{aligned} \psi_{final} &= 0° \\ v_{final} &= 41.15 \text{ m/sec} \end{aligned}\right\} \tag{137}$$

Finally, it is assumed that, in this particular case, the magnitude of the wind is negligible so that

$$v_w = 0 \text{ m/sec} \tag{138}$$

The first step is to generate the 3-D flight path by computing the ground track and then the altitude profile. This was accomplished using subroutine THREED. The resulting ground track is shown in figure 18, and the altitude profile is illustrated in figure 19.

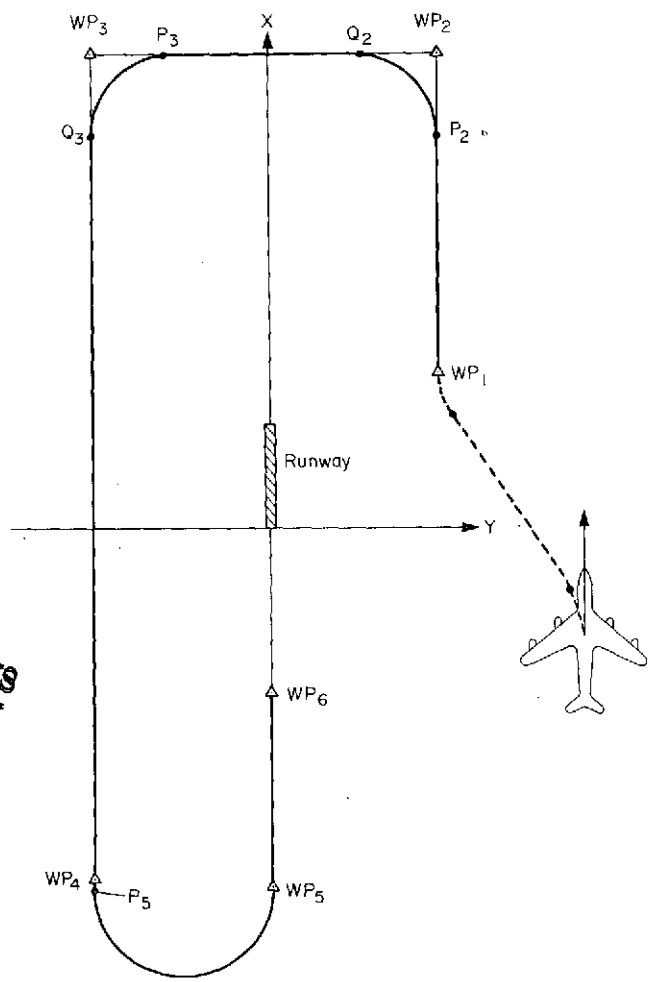


Figure 18.– Ground track for example problem.

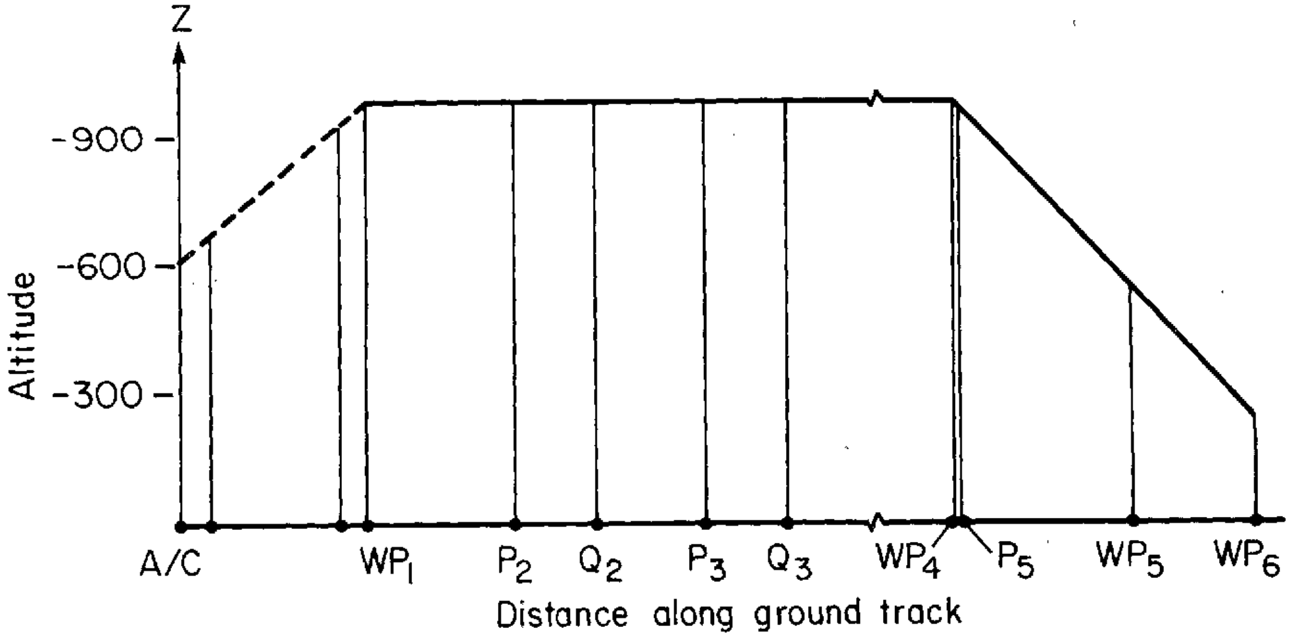


Figure 19.— Altitude profile for example problem.

i	2	3	4	5	6
ψ_i	0	-90°	180°	180°	0
GAM_i	0	0	0	-6.0°	-6.0°
D_i	3505	2591	10973	152	2896
XP_i	5791	7010	-5182	-5334	-2438
YP_i	2438	-1372	-2591	-2591	0
ZP_i	-988	-988	-988	-972	-244
$\Delta\psi_i$	-90°	-90°	0	-180°	0
R_i	1219	1219	627	1295	299
ΔD_i	1915	1915	0	4070	0
XQ_i	7010	5791	-5182	-5334	-2438
YQ_i	1219	-2591	-2591	0	0
ZQ_i	-988	-988	-988	-549	-299

After the 3-D flight path is completely determined, the next step is to compute the speed profile. This involves the computation of the admissible speed ranges at the waypoints with subroutine VRANGE, the attainable minimum and maximum flight times with subroutine TRANGE, and, finally, the computation of the actual speed profile corresponding to a desired time of arrival. The minimum and maximum admissible airspeeds at the waypoints are

i	1	2	3	4	5	6
$VMIN_i$	59.4	59.4	59.4	59.1	58.5	41.1
$VMAX_i$	77.7	77.7	77.7	59.1	58.5	41.1

The corresponding minimum and maximum flight times from each waypoint to the last waypoint are

i	1	2	3	4	5	6
$TMIN_i$	406	336	278	130	58	0
$TMAX_i$	483	392	316	130	58	0

At this point, a specific desired flight time must be selected from the first waypoint to the last waypoint. Although any number in the range ($TMIN_1$, $TMAX_1$) can be selected, it was decided to generate the minimum time flight path. Thus, the desired flight time from the first to last waypoint was chosen to be $TMIN_1$, namely,

$$ENRTIM_1 = TMIN_1 = 406 \text{ sec} \tag{139}$$

Having thus specified $ENRTIM_1$, subroutine SPEED is called to generate the speed profile (fig. 20). As expected, the speed profile coincides with the maximum admissible speed at every point since the minimum flight time was desired. The minimum admissible speed profile is also shown in figure 20 for comparison.

The next step in the guidance algorithm is to assemble the various guidance commands in chronological order and to form the 4-D guidance vectors. This task is accomplished by subroutine SPEED, resulting in the following chronological sequence of nondegenerate 4-D guidance vectors:

k	Δt_k	a_k	u_k	γ_k
5	45.1	0	0	0
6	24.6	0	-1/1219	0
7	33.3	0	0	0
8	24.6	0	-1/1219	0
9	87.5	0	0	0
10	61.0	-.305	0	0
11	0.6	0	0	-6.0
12	2.0	-.305	0	-6.0
13	69.5	0	-1/1295	-6.0
14	0.9	0	0	-6.0
15	57.0	-.305	0	-6.0

The corresponding values of NCI_i are

i	1	2	3	4	5	6
NCI_i	5	7	9	11	14	16

The lead times τ_ϕ and τ_γ are also computed in subroutine SPEED. Since the ground speed, and therefore the bank angle, may be different at the beginning of a turn from that at the end of the turn, each turn requires the computation of two lead times. For the specific example under consideration,

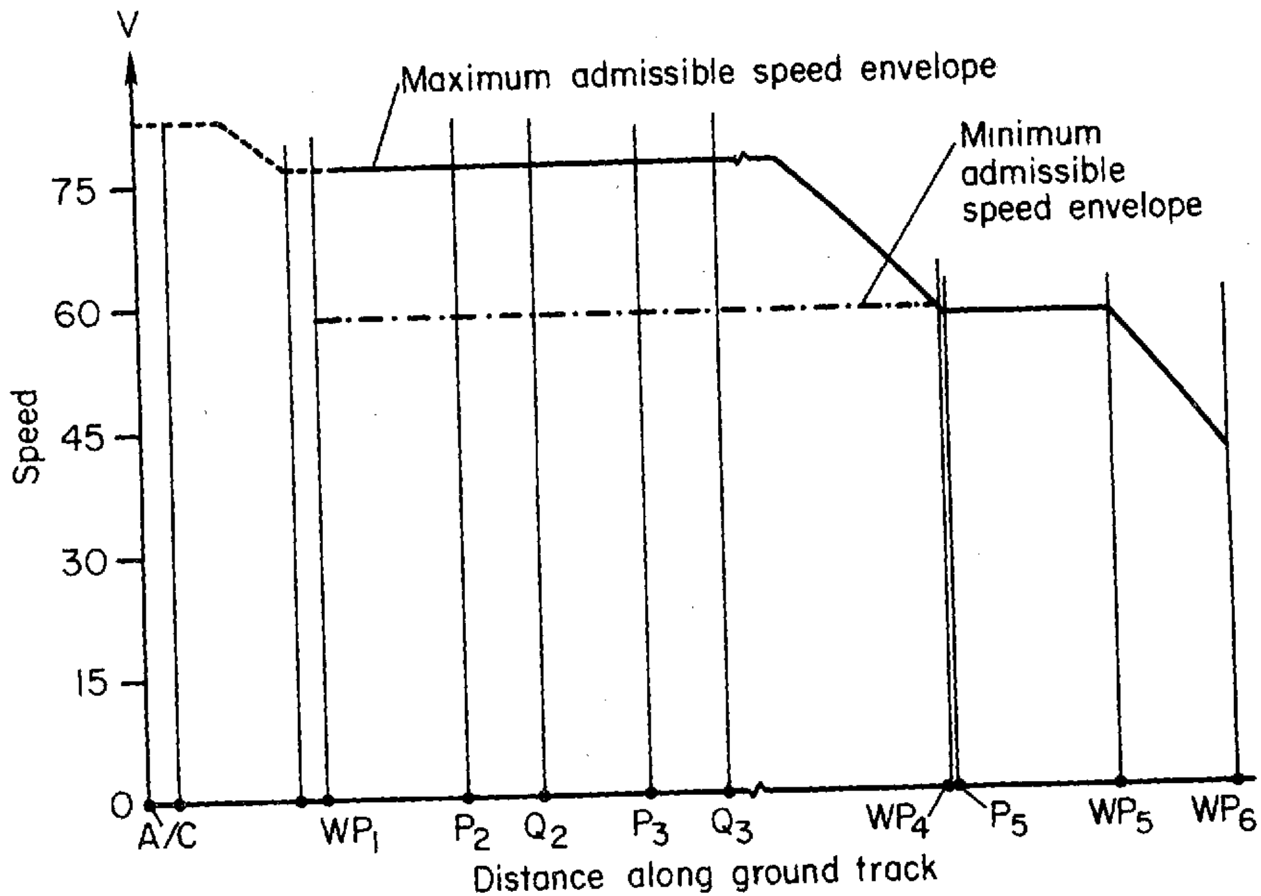


Figure 20.— Speed profile for example problem.

however, the wind was assumed to be negligible, so the ground speed, and therefore the bank angle, remain constant during a turn. Consequently, the same lead time applies to both the beginning and end of a turn. The values of the lead times for this example are

i	2	3	4	5	6
τ_{ϕ_i}	2.7	2.7	3.0	1.5	3.0
τ_{γ_i}	0	0	4.5	0	3.2

This completes the computation of the 4-D flight path from the first to the last waypoint.

In order to illustrate the computation of the 4-D capture flight path, the following initial conditions are assumed for the aircraft:

$$\left.\begin{array}{ll} \left.\begin{array}{l} X_{AC} = -1524 \\ Y_{AC} = 4572 \\ Z_{AC} = -610 \end{array}\right\} & \text{aircraft position} \\ \psi_{AC} = 0^\circ & \text{aircraft heading} \\ V_{AC} = 83.8 \text{ m/sec} & \text{aircraft airspeed} \end{array}\right\} \quad (140)$$

Assuming that the first waypoint is the capture waypoint, that is, CWP = 1, subroutine TST is called to generate the capture flight path. Using the notation of figure 14, the 3-D parameters of the capture flight path are

$$\left.\begin{array}{lll} XPA = -1524 & YQA = 4352 & \Delta\Psi B = 34.6^\circ \\ YPA = 4572 & ZQA = -674 & RB = 1067 \\ ZPA = -610 & GAMB = 4.9^\circ & \Delta DB = 645 \\ \Psi A = 0 & \Psi B = -34.6^\circ & XQB = 2286 \\ \Delta\Psi A = -34.6^\circ & DB = 3036 & YQB = 2438 \\ RA = 1241 & XPB = 1680 & ZQB = -988 \\ \Delta DA = 750 & YPB = 2627 & \\ XQA = -819 & ZPB = -933 & \end{array}\right\} \quad (141)$$

The speed profile along the capture flight path was then computed by TST, and the 4-D guidance commands were assembled into the following nondegenerate 4-D guidance vectors:

k	Δt_k	a_k	u_k	γ_k
1	8.9	0	-1/1241	4.9°
2	17.0	0	0	4.9°
3	20.0	-0.305	0	4.9°
4	8.3	0	1/1067	4.9°

Assuming that no lead time is introduced for the first turn of the capture flight path, the lead times for the capture waypoint – in this case, the first waypoint – were found to be $\tau_{\phi_1} = 3.0$ sec and $\tau_{\gamma_1} = 4.8$ sec.

CONCLUSIONS

This report presented the detailed theoretical development and computer implementation of three main guidance algorithms for an experimental 4-D guidance system. Using a small number of input parameters, the three algorithms generate the ground track, altitude profile, and speed profile of the 4-D reference flight path, respectively. At every stage of the computations, special care is exercised to assure that the final 4-D flight path is feasible from the point of view of aircraft maneuverability and structural limitations, terminal area operational constraints, and passenger comfort.

The flexibility of the algorithms is reflected in the various options that can be provided to the pilot for purposes of controlling the time of arrival: any combination of manual path stretching, speed profile recomputation, and the arbitrary selection of the capture waypoint may be used to control the arrival time of the aircraft. Furthermore, the method of representing the 4-D flight paths, namely, by sequences of guidance vectors, is particularly well suited for automatic or manual tracking.

Ames Research Center
National Aeronatuics and Space Administration
Moffett Field, Calif., 94035, October 8, 1974

APPENDIX

The Appendix describes the programming details of all routines used in the computer implementation of the 4-D guidance system. These routines are:

Main executive routine: FOURD

Subroutines: THREED, ROUND, NEWPSI,
VRANGE, TRANGE, SPEED,
TTTURN, TRANS, and TST

External functions: PIMOD and SGN

For each program, a statement of the required function to be performed is given, followed by a brief description of the computational method. All input, output, and temporary variables are defined. A list of numerical constants used in each program is also included. Finally, the subroutines called by each program, as well as the name of the calling program, are listed.

In order to make the appendix a self-contained programming reference for the 4-D guidance system, a flow chart and source listing for each program are also included. Finally, the printed output of a typical computer run is given.

FOURD

Required function:

FOURD is the main executive routine used to compute a complete 4-D flight path.

Method:

Numerical values for the required input variables are read from punched cards and are printed out for visual inspection. Constants and often-used variables are defined and polar variables are converted from degrees to radians. Then the subroutine THREED is called, which generates the 3-D flight path between the first and last waypoints. Next the routine VRANGE is called to compute the admissible speed ranges along the 3-D flight path; then TRANGE is executed to determine the corresponding minimum and maximum feasible time ranges from each waypoint to the last one (in order to enable the program to be used on-line in an interactive manner, the index of the first waypoint to be captured is defined as a variable and is denoted by CURWPT; given a set of waypoints WP_i, i = 1, 2, . . . , NWP and an integer value for CURWPT, 1 ≤ CURWPT ≤ NWP, the program will generate a 4-D flight path from the waypoint whose index is CURWPT to the last one, i.e., whose index is NWP).

At this point, a desired flight time, denoted in the program by $ENRTIM_{CURWPT}$, is required from the capture waypoint to the last waypoint. This time can be any number in the range ($TMIN_{CURWPT}$, $TMAX_{CURWPT}$). In the present version of the program, $TMIN_{CURWPT}$ was chosen arbitrarily. Calling the subroutine SPEED generates the corresponding feasible speed profile. Thus, the 4-D flight path is computed. Polar variables are converted from radians to degrees and the flight-path parameters are printed out.

The remaining portion of FOURD (starting with internal statement number 9500) is used as an executive routine to generate the capture flight path via subroutine TST. The input set needed for this part of the program consists of the position, heading, and speed of the aircraft, as well as the index of the waypoint to be captured (CURWPT). This input is read from cards, printed out, and then subroutine TST is called. The parameters of the resulting 4-D capture flight path are printed.

Input data:

MAXPHI	maximum admissible bank angle, deg
MAXGAM	maximum admissible flight-path angle, deg
MINGAM	minimum admissible flight-path angle, deg
MAXACC	maximum admissible acceleration (positive), ft/sec^2
MAXDEC	maximum admissible deceleration (negative), ft/sec^2
PHIDOT	maximum admissible roll rate, deg/sec
VSNF	stall speed of aircraft with zero flaps, ft/sec
VSFF	stall speed of aircraft with full flaps, ft/sec
CUPPER	ratio of maximum admissible cruising airspeed in terminal area to zero-flap stall speed (same as $\bar{c}$ in eqs. (69) and (70))
CLOWER	ratio of minimum admissible cruising speed in terminal area to zero-flap stall speed (same as $\underline{c}$ in eqs. (69) and (70))
NWP	number of waypoints used to define flight path
HFINAL	desired heading at the last waypoint, deg
GFINAL	desired flight-path angle at the last waypoint, deg
VFINAL	desired airspeed at the last waypoint, ft/sec
HW	estimated wind direction, deg
VW	estimated wind magnitude, ft/sec

$INDEX_i$ waypoint-type indicator; $INDEX_i = 0 \Rightarrow$ waypoint i is ordinary waypoint; $INDEX_i = 1 \Rightarrow$ waypoint i is final heading waypoint

XWP_i, YWP_i, ZWP_i Cartesian coordinates of waypoint i, ft

R_i desired turning radius at waypoint i, ft; if R_i is unspecified (i.e., $R_i = 0.$), then R_i is set equal to the minimum feasible radius

Output data:

See the output from subroutines THREED, VRANGE, TRANGE, and SPEED.

Temporary variables:

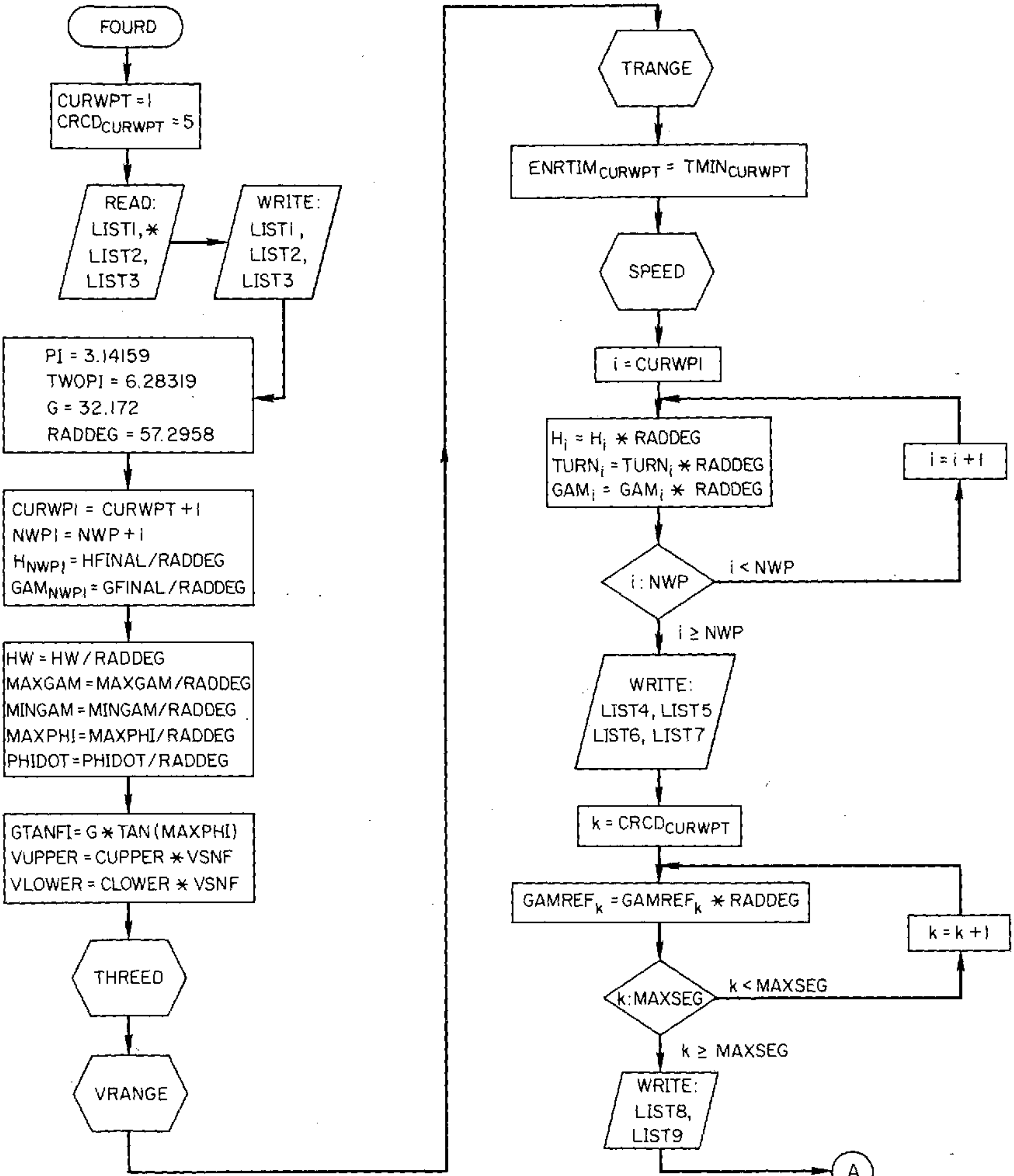
CURWP1	index of waypoint following the capture waypoint (=CURWPT+1)
NWP1	=NWP+1
GTANFI	=G*TAN(MAXPHI)
L	$=CRCD_{CURWPT}$

Constants:

PI	= 3.14159
TWOPI	= 6.28319
G	acceleration due to gravity, 32.172 ft/sec^2
RADDEG	radians-degrees conversion factor, 57.2958 deg/rad

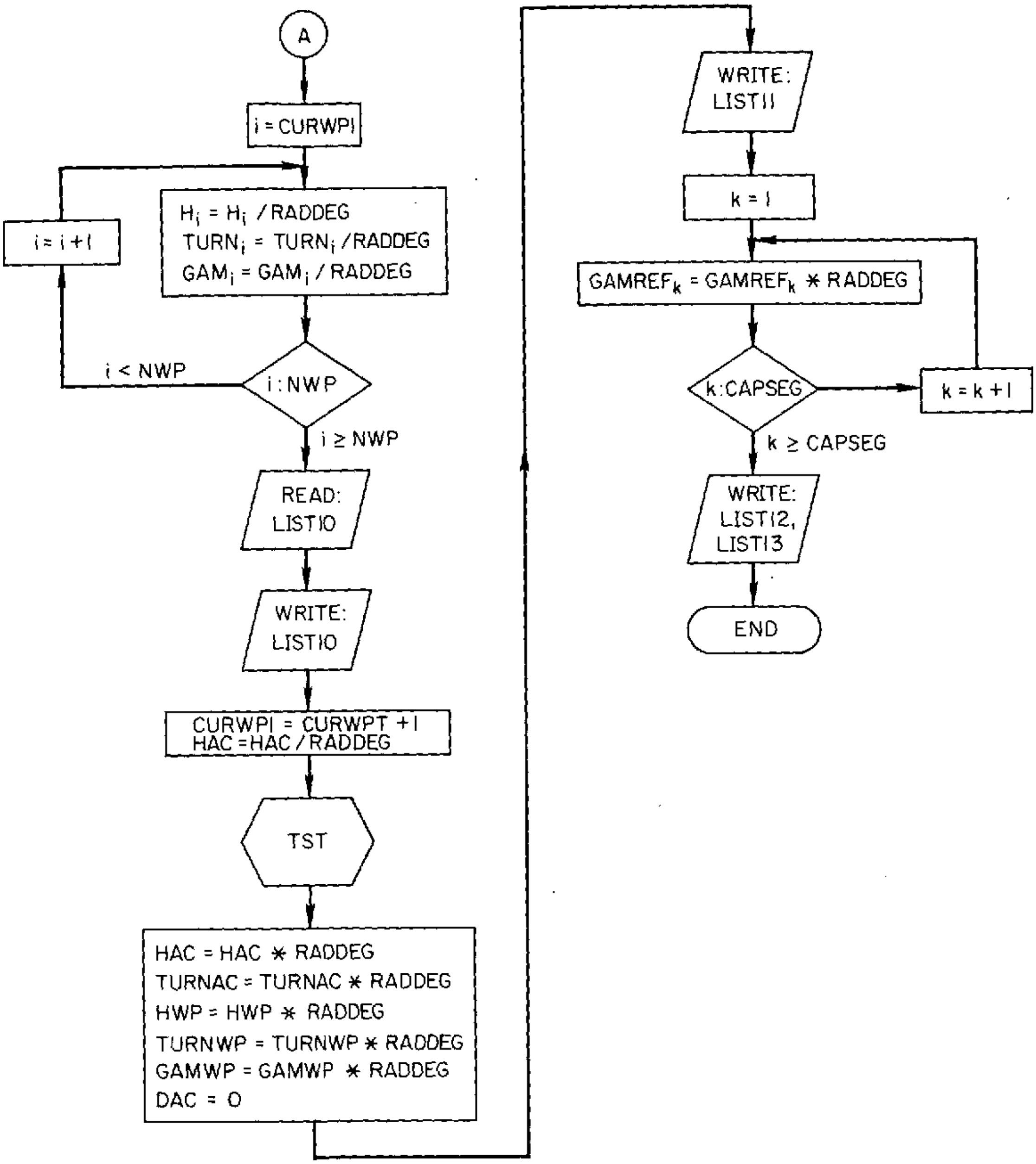
Subroutines required:

THREED	generates 3-D flight-path parameters
VRANGE	computes minimum and maximum feasible airspeeds at each waypoint
TRANGE	computes minimum and maximum attainable flight times from each waypoint to the last waypoint
SPEED	generates the sequence of 4-D command sequence
TST	generates the 4-D capture flight path from initial aircraft state to the capture waypoint



* For variables contained in LIST1 . . . LIST13, see page following this flow diagram

ORIGINAL PAGE IS OF POOR QUALITY



A
i = CURWPI
Hi = Hi / RADDEG
TURNi = TURNi / RADDEG
GAMi = GAMi / RADDEG
i = i + 1
i < NWP
i : NWP
i ≥ NWP
READ: LIST10
WRITE: LIST10
CURWPI = CURWPT + 1
HAC = HAC / RADDEG
TST
HAC = HAC * RADDEG
TURNAC = TURNAC * RADDEG
HWP = HWP * RADDEG
TURNWP = TURNWP * RADDEG
GAMWP = GAMWP * RADDEG
DAC = 0
WRITE: LIST11
k = 1
GAMREFk = GAMREFk * RADDEG
k:CAPSEG
k = k + 1
k ≥ CAPSEG
WRITE: LIST12, LIST13
END

LIST1,LIST2, . . . ,LIST13 contain the following variables:

LIST1: MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,CUPPER, CLOWER

LIST2: NWP,HFINAL,GFINAL,VFINAL,VW,HW

LIST3: $INDEX_i, XWP_i, YWP_i, ZWP_i, R_i$, i=CURWPT,CURWPT+1, . . . ,NWP

LIST4: $H_i, GAM_i, D_i, XP_i, YP_i, ZP_i, TURN_i, R_i, DELD_i, XQ_i, YQ_i, ZQ_i$; i=CURWPT+1,CURWPT+2, . . . ,NWP

LIST5: $VMIN_i, VMAX_i$; i=CURWPT,CURWPT+1, . . . ,NWP

LIST6: $TMIN_i, TMAX_i$; i=CURWPT,CURWPT+1, . . . ,NWP

LIST7: VA_i, $ENRTIM_i, CRCD_i$; i=CURWPT,CURWPT+1, . . . ,NWP

LIST8: $TIMREF_k, TRNREF_k, GAMREF_k, ACCREF_k$; $k=CRCD_{CURWPT}$, $CRCD_{CURWPT}+1$, . . . ,MAXSEG

LIST9: $TLEADT_i, TLEADS_i, TLEADP_i$; i=CURWPT+1,CURWPT+2, . . . ,NWP

LIST10: XAC,YAC,ZAC,HAC,VAC,CURWPT

LIST11: HAC,GAMWP,DAC,XAC,YAC,ZAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC, HWP,GAMWP,DWP,XPWP,YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP

LIST12: $TIMREF_k, TRNREF_k, GAMREF_k, ACCREF_k$; k=1,2, . . . ,CAPSEG

LIST13: $TLEADT_1, TLEADS_1, TLEADP_1$

```
      COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
     1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
     2VLOWER,CURWPT,CURWP1,NWP1,I,RU,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
     3CAPSEG,DAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC,HWP,GAMWP,DWP,XPWP,
     4YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP,INDEX(20),XWP(20),
     5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
     6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
     7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
     8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
     9TRNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
      REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
      INTEGER CURWPT,CURWP1,CAPSEG,CRCD
      CURWPT=1
      CRCD(CURWPT)=5
      READ(5,1) MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,
     1VSNF,VSFF,CUPPER,CLOWER
    1 FORMAT(6F10.1/4F10.1)
      READ(5,2) NWP,HFINAL,GFINAL,VFINAL,HW,VW
    2 FORMAT(I10,5F10.1)
      READ(5,3) (INDEX(I),XWP(I),YWP(I),ZWP(I),R(I),I=CURWPT,NWP)
    3 FORMAT(I10,4F10.1)
      WRITE(6,4) MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,
     1VSNF,VSFF,CUPPER,CLOWER
    4 FORMAT(///2X,'MAXPHI  =',F6.1/2X,'MAXGAM  =',F6.1/2X,'MINGAM  =',
     1F6.1/2X,'MAXACC  =',F6.1/2X,'MAXDEC  =',F6.1/2X,'PHIDOT  =',F6.1/
     22X,'VSNF  =',F6.1/2X,'VSFF  =',F6.1/2X,'CUPPER  =',F6.1/2X,
     3'CLOWER  =',F6.1)
      WRITE(6,5) NWP,HFINAL,GFINAL,VFINAL,HW,VW
    5 FORMAT(///2X,'NWP  =',I4,5X,'HFINAL  =',F8.1,5X,'GFINAL  =',F8.1,
     15X,'VFINAL  =',F8.1,5X,'HW  =',F8.1,5X,'VW  =',F8.1)
      WRITE(6,6)
    6 FORMAT(///2X,'INDEX(I)',4X,'XWP(I)',4X,'YWP(I)',4X,'ZWP(I)',6X,
     1'R(I)'/)
      WRITE(6,3) (INDEX(I),XWP(I),YWP(I),ZWP(I),R(I),I=CURWPT,NWP)
      PI=3.14159
      TWOPI=6.28319
      G=32.172
      RADDEG=57.2958
      CURWP1=CURWPT+1
      NWP1=NWP+1
      HW=HW/RADDEG
      H(NWP1)=HFINAL/RADDEG
      GAM(NWP1)=GFINAL/RADDEG
      MAXGAM=MAXGAM/RADDEG
      MINGAM=MINGAM/RADDEG
```

```
      MAXPHI=MAXPHI/RADDEG
      GTANFI=G*TAN(MAXPHI)
      PHIDOT=PHIDOT/RADDEG
      VUPPER=CUPPER*VSNF
      VLOWER=CLOWER*VSNF
      CALL THREED(&100)
      CALL VRANGE
      CALL TRANGE
      ENRTIM(CURWPT)=TMIN(CURWPT)
      CALL SPEED
      DO 11 I=CURWP1,NWP
         H(I)=H(I)*RADDEG
         TURN(I)=TURN(I)*RADDEG
   11    GAM(I)=GAM(I)*RADDEG
      WRITE(6,12)
   12 FORMAT(///8X,'H(I)',4X,'GAM(I)',6X,'D(I)',5X,'XP(I)',5X,'YP(I)',
     15X,'ZP(I)',3X,'TURN(I)',6X,'R(I)',3X,'DELD(I)',5X,'XQ(I)',5X,
     2'YQ(I)',5X,'ZQ(I)'/)
      WRITE(6,13) (H(I),GAM(I),D(I),XP(I),YP(I),ZP(I),TURN(I),R(I),
     1DELD(I),XQ(I),YQ(I),ZQ(I),I=CURWP1,NWP)
   13 FORMAT(2X,2F10.2,10F10.1)
      WRITE(6,14)
   14 FORMAT(///2X,'VMIN(I)',10X,'VMAX(I)'/)
      WRITE(6,15) (VMIN(I),VMAX(I),I=CURWPT,NWP)
   15 FORMAT(3X,F5.1,12X,F5.1)
      WRITE(6,16)
   16 FORMAT(///2X,'TMIN(I)',10X,'TMAX(I)'/)
      WRITE(6,17) (TMIN(I),TMAX(I),I=CURWPT,NWP)
   17 FORMAT(2X,F6.1,11X,F6.1)
      WRITE(6,18)
   18 FORMAT(///2X,'VA(I)',10X,'ENRTIM(I)',10X,'CRCD(I)'/)
      WRITE(6,19) (VA(I),ENRTIM(I),CRCD(I),I=CURWPT,NWP)
   19 FORMAT(2X,F5.1,10X,F7.1,15X,I2)
      L=CRCD(CURWPT)
      DO 21 K=L,MAXSEG
   21    GAMREF(K)=GAMREF(K)*RADDEG
      WRITE(6,22)
   22 FORMAT(///9X,'TIMREF(K)',4X,'TRNREF(K)',7X,'GAMREF(K)',5X,
     1'ACCREF(K)'/)
      WRITE(6,23) (TIMREF(K),TRNREF(K),GAMREF(K),ACCREF(K),K=L,MAXSEG)
   23 FORMAT(2X,4F14.1)
      WRITE(6,31)
   31 FORMAT(//10X,'TLEADT(I)',5X,'TLEADS(I)',5X,'TLEADP(I)'/)
      WRITE(6,32) (TLEADT(I),TLEADS(I),TLEADP(I),I=CURWP1,NWP)
   32 FORMAT(2X,3F14.1)
```

```
      DO 24 I=CURWP1,NWP
         H(I)=H(I)/RADDEG
         TURN(I)=TURN(I)/RADDEG
 24      GAM(I)=GAM(I)/RADDEG
      READ(5,25) XAC,YAC,ZAC,HAC,VAC,CURWPT
 25   FORMAT(5F10.1,I10)
      WRITE(6,26)
 26   FORMAT(///7X,'XAC',7X,'YAC',7X,'ZAC',7X,'HAC',7X,'VAC',6X,'CURWPT'
     1/)
      WRITE(6,25) XAC,YAC,ZAC,HAC,VAC,CURWPT
      CURWP1=CURWPT+1
      HAC=HAC/RADDEG
      CALL TST(&100)
      HAC=HAC*RADDEG
      TURNAC=TURNAC*RADDEG
      HWP=HWP*RADDEG
      TURNWP=TURNWP*RADDEG
      GAMWP=GAMWP*RADDEG
      DAC=0.
      WRITE(6,27)
 27   FORMAT(///11X,'H',7X,'GAM',9X,'D',8X,'XP',8X,'YP',8X,'ZP',6X,
     1'TURN',9X,'R',6X,'DELD',8X,'XQ',8X,'YQ',8X,'ZQ'/)
      WRITE(6,28) HAC,GAMWP,DAC,XAC,YAC,ZAC,TURNAC,RAC,DELDAC,XQAC,YQAC,
     1ZQAC,HWP,GAMWP,DWP,XPWP,YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP
 28   FORMAT(2X,12F10.1)
      DO 29 K=1,CAPSEG
 29      GAMREF(K)=GAMREF(K)*RADDEG
      WRITE(6,22)
      WRITE(6,23) (TIMREF(K),TRNREF(K),GAMREF(K),ACCREF(K),K=1,CAPSEG)
      WRITE(6,31)
      WRITE(6,32) TLEADT(1),TLEADS(1),TLEADP(1)
100   CONTINUE
      STOP
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

Required function:

Using waypoint input parameters and aircraft performance constraints, THREED computes the 3-D flight-path parameters.

Method:

Beginning with the last waypoint, the minimum turning radius is computed and compared to the specified turning radius (if the radius is not specified, it is set equal to the minimum turning radius). If the specified radius is smaller than the minimum feasible one, a diagnostic message is printed and control is returned to the main executive routine FOURD.

Next, the type of waypoint under consideration is determined by checking the value of $INDEX_i$. For ordinary waypoints ($INDEX_i$=0), subroutine ROUND is called; for final heading waypoints ($INDEX_i$=1), NEWPSI is called. This step yields the ground track between the waypoint under consideration and the preceding one. Then an upper bound on the ground speed at the previous waypoint is computed, and the above process is repeated sequentially for all but the capture waypoint.

When the complete ground track is known, the altitude at the end of the turn associated with each waypoint is set equal to the prespecified waypoint altitude and the constant flight-path-angle segments between adjacent waypoints are determined. Each flight-path-angle segment is checked to assure that the minimum and maximum flight-path-angle constraints are not violated. If any one of the flight-path-angles is outside the admissible range, a diagnostic message is printed and control is returned to FOURD.

Input data:

NWP	number of waypoints used to define flight path
VFINAL	desired airspeed at the last waypoint, ft/sec
VW	estimated wind magnitude, ft/sec
$INDEX_i$	waypoint type indicator (see input data for FOURD)
XWP_i,YWP_i ZWP_i	Cartesian waypoint coordinates, ft
GTANFI	=G*TAN(MAXPHI)
CURWPT	index of capture waypoint
CURWP1	=CURWPT+1

MAXGAM	maximum admissible flight-path angle, radian
MINGAM	minimum admissible fligh-path angle, radian
MAXDEC	maximum admissible deceleration (negative), ft/sec^2
VUPPER	maximum admissible cruising airspeed in terminal area, ft/sec

Output data:

H_i	heading of straight flight from point Q_{i-1} to point P_i, radian
GAM_i	flight-path angle from point Q_{i-1} to point Q_i, radian
D_i	length of straight flight from point Q_{i-1} to point P_i, ft
XP_i, YP_i, ZP_i	x,y,z-coordinates of beginning of turn at waypoint i, ft
$TURN_i$	angular extent of turn at waypoint i; $TURN_i > 0 \Rightarrow$ right turn, $TURN_i < 0 \Rightarrow$ left turn; radian
R_i	turning radius at waypoint i, ft
$DELD_i$	arclength of $TURN_i$, ft
XQ_i, YQ_i, ZQ_i	x,y,z-coordinates of end of turn at waypoint i, ft

Temporary variables:

VGMAX	maximum possible ground speed at waypoint to be processed, ft/sec
RMIN	minimum admissible turning radius at waypoint to be processed, ft
HI1	reverse of heading H_{i+1} at final heading waypoints; $HI1 = mod_{\pi}(H_{i+1}+\pi)$, radian
HI	reverse of heading H_i, radian

Constants:

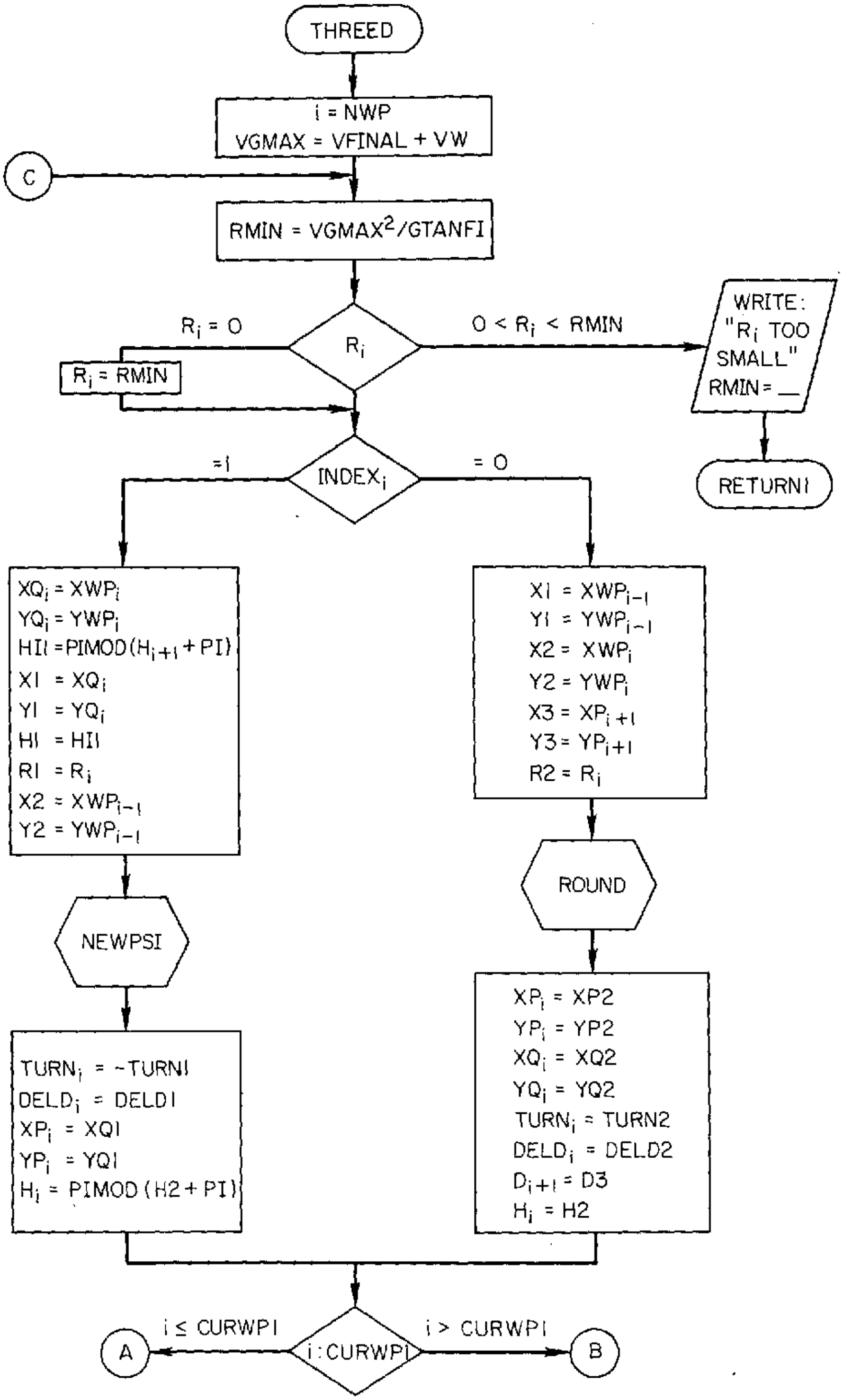
GTANFI	=G*TAN(MAXPHI)
PI	=3.14159

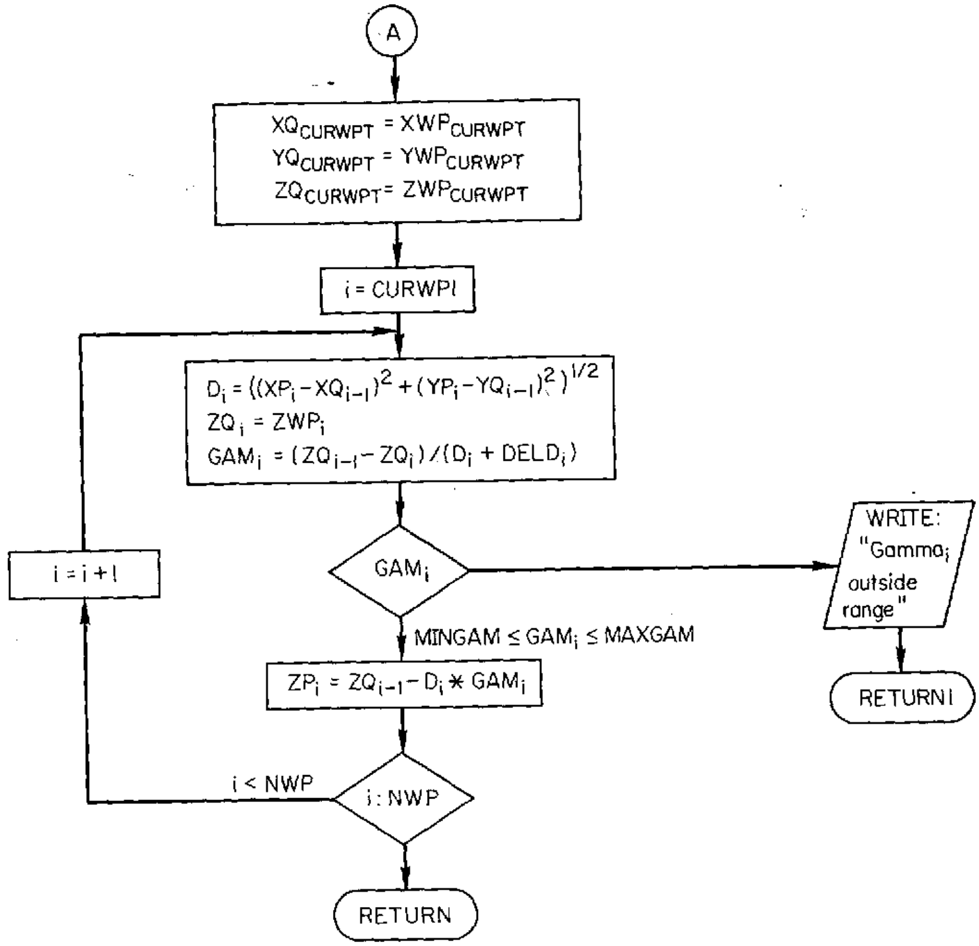
Subroutine required:

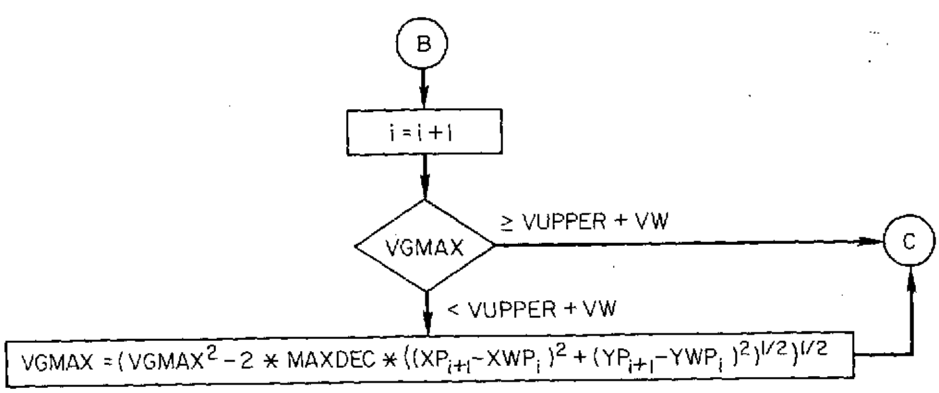
NEWPSI computes ground track between waypoints (i-1) and i when waypoint i is a final heading type (i.e., $INDEX_i=1$)

ROUND computes ground track between waypoints (i-1), i, and (i+1) when waypoint i is an ordinary type (i.e., $INDEX_i=0$)

Called by: FOURD


THREED
i = NWP
VGMAX = VFINAL + VW
C
RMIN = VGMAX²/GTANFI
Rᵢ = 0
Rᵢ
0 < Rᵢ < RMIN
WRITE: "Rᵢ TOO SMALL" RMIN = __
Rᵢ = RMIN
RETURN1
=1
INDEXᵢ
= 0
XQᵢ = XWPᵢ
YQᵢ = YWPᵢ
H11 = PIMOD (Hᵢ₊₁ + PI)
X1 = XQᵢ
Y1 = YQᵢ
H1 = H11
R1 = Rᵢ
X2 = XWPᵢ₋₁
Y2 = YWPᵢ₋₁
X1 = XWPᵢ₋₁
Y1 = YWPᵢ₋₁
X2 = XWPᵢ
Y2 = YWPᵢ
X3 = XPᵢ₊₁
Y3 = YPᵢ₊₁
R2 = Rᵢ
NEWPSI
ROUND
TURNᵢ = -TURN1
DELDᵢ = DELD1
XPᵢ = XQ1
YPᵢ = YQ1
Hᵢ = PIMOD (H2 + PI)
XPᵢ = XP2
YPᵢ = YP2
XQᵢ = XQ2
YQᵢ = YQ2
TURNᵢ = TURN2
DELDᵢ = DELD2
Dᵢ₊₁ = D3
Hᵢ = H2
i ≤ CURWP1
i : CURWP1
i > CURWP1
A
B


A
XQCURWPT = XWPCURWPT
YQCURWPT = YWPCURWPT
ZQCURWPT = ZWPCURWPT
i = CURWPI
Di = ((XPi − XQi−1)2 + (YPi − YQi−1)2)1/2
ZQi = ZWPi
GAMi = (ZQi−1 − ZQi)/(Di + DELDi)
GAMi
WRITE: "Gammai outside range"
RETURNI
MINGAM ≤ GAMi ≤ MAXGAM
ZPi = ZQi−1 − Di * GAMi
i = i + 1
i < NWP
i : NWP
RETURN


B
i = i + 1
VGMAX
≥ VUPPER + VW
C
< VUPPER + VW
VGMAX = (VGMAX2 − 2 * MAXDEC * ((XPi+1 − XWPi)2 + (YPi+1 − YWPi)2)1/2)1/2

```
      SUBROUTINE THREED(*)
      COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
     1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
     2VLOWER,CURWPT,CURWP1,NWP1,I,RO,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
     3CAPSEG,DAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC,HWP,GAMWP,DWP,XPWP,
     4YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP,INDEX(20),XWP(20),
     5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
     6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
     7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
     8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
     9THNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
      REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
      INTEGER CURWPT,CURWP1,CAPSEG,CRCD
      I=NWP
      VGMAX=VFINAL+VW
10    RMIN=VGMAX*VGMAX/GTANFI
      IF(R(I).GT.0.) GO TO 20
         R(I)=RMIN
         GO TO 30
20    IF(R(I).GE.RMIN) GO TO 30
         WRITE(6,21) I,RMIN
21       FORMAT(///2X,'R(',I2,') TOO SMALL, RMIN = ',F8.1)
         GO TO 100
30    IF(INDEX(I).EQ.0) GO TO 40
         XQ(I)=XWP(I)
         YQ(I)=YWP(I)
         HI1=PIMOD(H(I+1)+PI)
         CALL NEWPSI(XQ(I),YQ(I),HI1,R(I),XWP(I-1),YWP(I-1),TURN(I),
     1                   DELD(I),XP(I),YP(I),HI,&100)
         TURN(I)=-TURN(I)
         H(I)=PIMOD(HI+PI)
         GO TO 50
40    CALL ROUND(XWP(I-1),YWP(I-1),XWP(I),YWP(I),XP(I+1),YP(I+1),R(I),
     1XP(I),YP(I),XQ(I),YQ(I),TURN(I),DELD(I),D(I+1),H(I),&100)
50    IF(I.LE.CURWP1) GO TO 60
         I=I-1
         IF(VGMAX.GE.VUPPER+VW) GO TO 10
         VGMAX=SQRT(VGMAX*VGMAX-2.*MAXDEC*SQRT((XP(I+1)-XWP(I))**2+
     1   (YP(I+1)-YWP(I))**2))
         IF(VGMAX.GE.VUPPER+VW) VGMAX=VUPPER+VW
         GO TO 10
60    XQ(CURWPT)=XWP(CURWPT)
      YQ(CURWPT)=YWP(CURWPT)
      ZQ(CURWPT)=ZWP(CURWPT)
      DO 70 I=CURWP1,NWP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      D(I)=SQRT((XP(I)-XQ(I-1))**2+(YP(I)-YQ(I-1))**2)
      ZQ(I)=ZWP(I)
      GAM(I)=(ZQ(I-1)-ZQ(I))/(D(I)+DELD(I))
      IF(GAM(I).LE.MAXGAM.AND.GAM(I).GE.MINGAM) GO TO 70
         WRITE(6,61) I
 61      FORMAT(///2X,'GAMMA OUTSIDE RANGE',I10)
         GO TO 100
 70   ZP(I)=ZQ(I-1)-D(I)*GAM(I)
      RETURN
100   RETURN 1
      END
```

ROUND

Required function:

This routine is used to compute the ground track parameters of a flight path defined by three points in the following manner: the ground track begins at one of the points with a straight line directed toward the second point; near the second point, a circular turn of a given radius is made such that the resulting new heading is directly toward the third point.

Method:

The headings of the straight lines from the first point to the second point and from the second point to the third point are first computed. From their relationship, the magnitude and direction of the required turn are determined. It is then checked whether the three points are sufficiently far apart so that the turn computed above is in fact feasible. If either end point is too close to the middle point, a diagnostic message is printed and control is returned to the calling program (in this case, THREED). If the turn is feasible, then the arclength and the x,y-coordinates of the beginning and end of the turn are computed.

Input data:

X1,Y1	x,y-coordinates of first point, ft
X2,Y2	x,y-coordinates of second point, ft
X3,Y3	x,y-coordinates of third point, ft
R2	turning radius at second point, ft

Output data:

H2	heading of straight line from first point to second point, radian
H3	heading of straight line from second point to third point, radian
TURN2	angular extent of turn near second point, radian (note: $\lvert TURN2 \rvert < \pi$)
D2	length of straight line from first point to point where turn begins, ft
D3	length of straight line from end of turn to third point, ft
DELD2	arclength of TURN2, ft

XP2,YP2   x,y-coordinates of beginning of turn, ft

XQ2,YQ2   x,y-coordinates of end of turn, ft

Temporary variables:

SIGN   reflects direction of TURN2; SIGN=+1 $\Rightarrow$ TURN2$\geq$0, SIGN=-1 $\Rightarrow$ TURN2<0

SIDE   distance between beginning (end) of turn and second point, ft

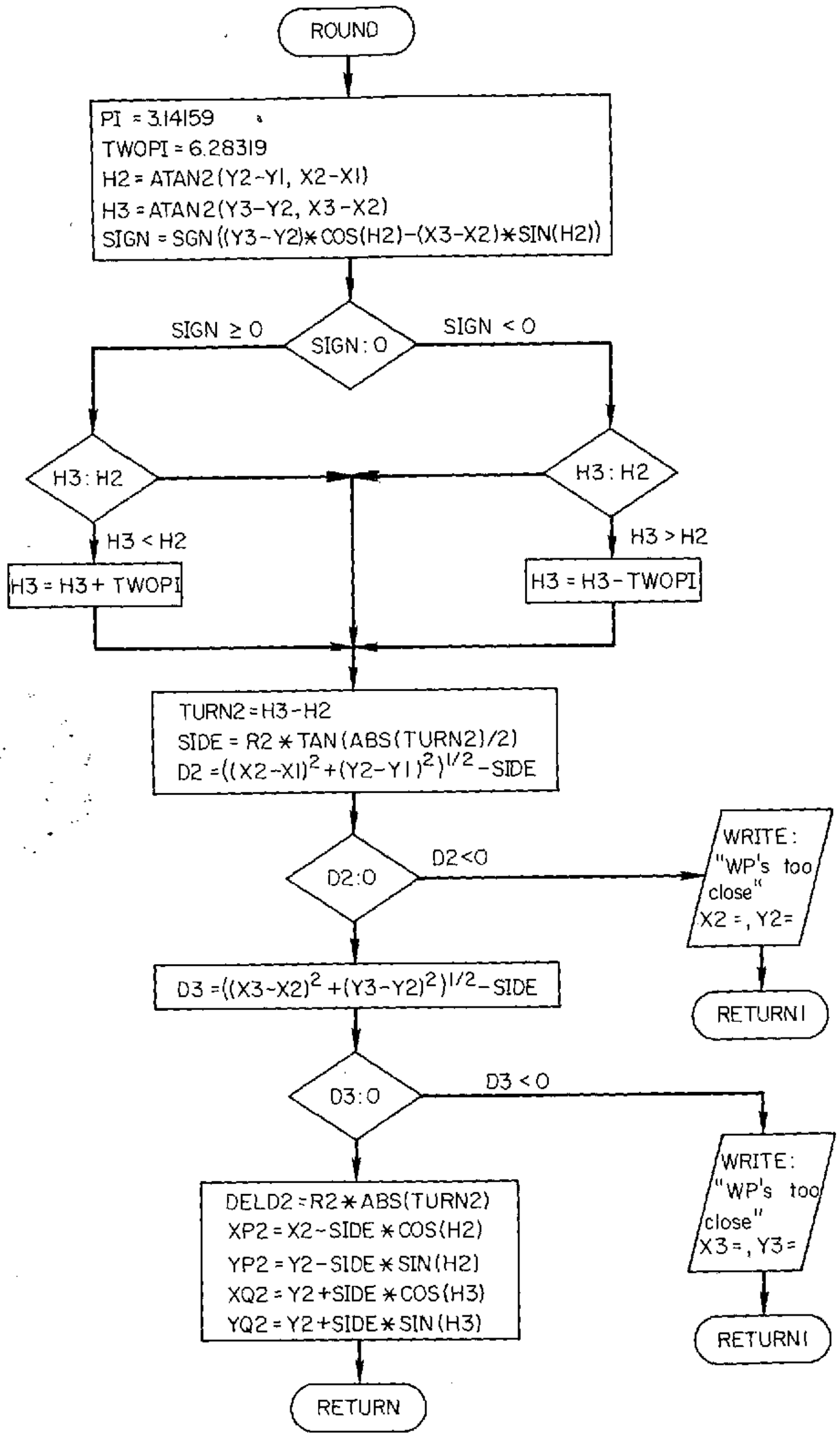
Constants:

PI   =3.14159

TWOPI   =6.28319

Subroutine required: None

Called by: THREED


ROUND
PI = 3.14159
TWOPI = 6.28319
H2 = ATAN2(Y2-Y1, X2-X1)
H3 = ATAN2(Y3-Y2, X3-X2)
SIGN = SGN((Y3-Y2)*COS(H2)-(X3-X2)*SIN(H2))
SIGN ≥ 0
SIGN : 0
SIGN < 0
H3 : H2
H3 : H2
H3 < H2
H3 > H2
H3 = H3 + TWOPI
H3 = H3 - TWOPI
TURN2 = H3-H2
SIDE = R2 * TAN(ABS(TURN2)/2)
D2 = ((X2-X1)^2 + (Y2-Y1)^2)^1/2 - SIDE
D2:0
D2<0
WRITE: "WP's too close" X2 =, Y2=
RETURN1
D3 = ((X3-X2)^2 + (Y3-Y2)^2)^1/2 - SIDE
D3:0
D3 < 0
WRITE: "WP's too close" X3 =, Y3 =
RETURN1
DELD2 = R2*ABS(TURN2)
XP2 = X2-SIDE * COS(H2)
YP2 = Y2-SIDE * SIN(H2)
XQ2 = Y2+SIDE * COS(H3)
YQ2 = Y2+SIDE * SIN(H3)
RETURN

```
      SUBROUTINE ROUND(X1,Y1,X2,Y2,X3,Y3,R2,XP2,YP2,XQ2,YQ2,TURN2,
     1DELD2,D3,H2,*)
      PI=3.14159
      TWOPI=6.28319
      H2=ATAN2(Y2-Y1,X2-X1)
      H3=ATAN2(Y3-Y2,X3-X2)
      SIGN=SGN((Y3-Y2)*COS(H2)-(X3-X2)*SIN(H2))
      IF(SIGN.LT.0.) GO TO 1
      IF(H3.LT.H2) H3=H3+TWOPI
      GO TO 2
    1 IF(H3.GT.H2) H3=H3-TWOPI
    2 TURN2=H3-H2
      SIDE=R2*TAN(ABS(TURN2)/2.)
      D2=SQRT((X2-X1)*(X2-X1)+(Y2-Y1)*(Y2-Y1))-SIDE
      IF(D2.GE.0.) GO TO 3
         WRITE(6,10) X2,Y2
   10    FORMAT(///2X,'WAYPOINTS TOO CLOSE'//F20.1/F20.1)
         GO TO 5
    3 D3=SQRT((X3-X2)*(X3-X2)+(Y3-Y2)*(Y3-Y2))-SIDE
      IF(D3.GE.0.) GO TO 4
         WRITE(6,10) X3,Y3
         GO TO 5
    4 DELD2=R2*ABS(TURN2)
      XP2=X2-SIDE*COS(H2)
      YP2=Y2-SIDE*SIN(H2)
      XQ2=X2+SIDE*COS(H3)
      YQ2=Y2+SIDE*SIN(H3)
      RETURN
    5 RETURN 1
      END
```

NEWPSI

Required function:

This routine is used to compute the ground-track parameters of the minimum length flight path between two given points; the flight path and therefore the ground track is subject to a given initial heading and a minimum turning radius constraint.

Method:

The desired ground track consists of a circular turn followed by a straight-line segment. It is first determined whether the two points are distinct. In the pathological case when the two points coincide, the turn is set equal to zero and the intermediate computation is omitted. If the two points are distinct, then the direction of the turn is determined by noting whether the second point lies to the right or left of the directed line that passes through the first point and has the given initial heading. When the direction of the turn is known, its center is computed, and it is checked whether the second point lies inside the resulting circle. If the second point lies inside the circle, a diagnostic message is printed and control is returned to the calling program. If the second point is not inside the circle, then the three angles ANGLE1, ANGLE2, and ANGLE3 are computed and are used to determine the angular extent of the turn.

The last step is to compute the arclength, the straight-line heading toward the second point, and the x,y-coordinates of the end of the turn.

Input data:

X1,Y1	x,y-coordinates of the first point, ft
H1	initial heading at the first point, radian
R1	radius of turn at the first point, ft
X2,Y2	x,y-coordinates of the second point, ft

Output data:

TURN1	angular extent of turn, radian ($-2\pi <$ TURN1 $< 2\pi$)
DELD1	arclength, ft
XQ1,YQ1	x,y-coordinates of end of turn, ft
H2	heading of straight line, radian

Temporary variables:

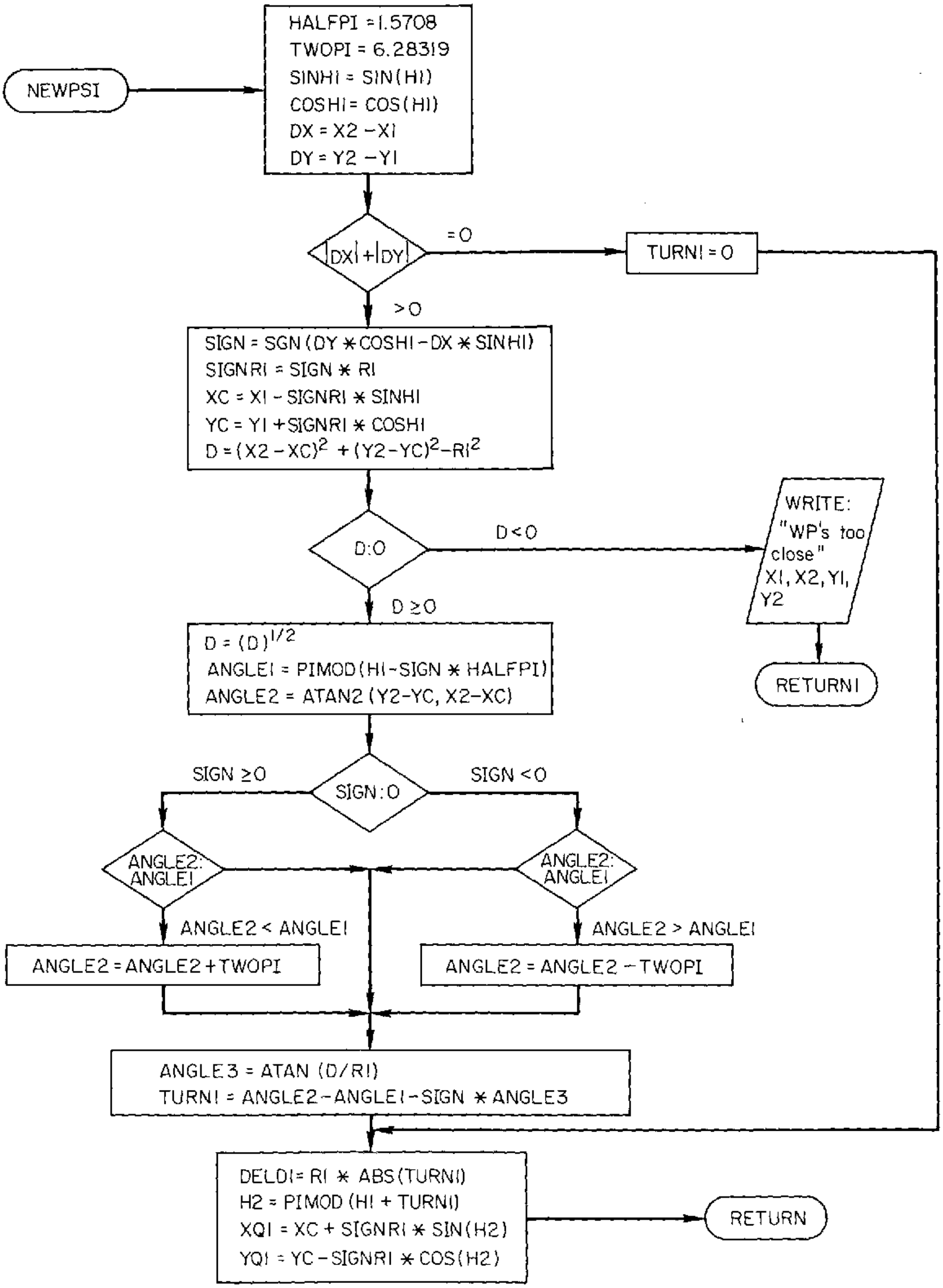
SINH1	=sin(H1)
COSH1	=cos(H1)
DX	=X2-X1, ft
DY	=Y2-Y1, ft
TEMP	$=\|DX\|+\|DY\|$, ft
XC,YC	x,y-coordinates of the center of turn, ft
SIGN	=+1 if TURN1≥0; =-1 if TURN1<0
SIGNR1	=SIGN*R1, ft
D	length of straight-line segment (or its square), ft
ANGLE1	heading of line from center of turn to first point, radian
ANGLE2	heading of line from center of turn to second point, radian
ANGLE3	$=\tan^{-1}$ (D/R1), radian

Constants:

HALFPI	=1.5708
TWOPI	=6.28319

Subroutines required: None

Called by: THREED and TST


NEWPSI
HALFPI = 1.5708
TWOPI = 6.28319
SINH1 = SIN(H1)
COSH1 = COS(H1)
DX = X2 - X1
DY = Y2 - Y1
|DX| + |DY|
= 0
TURN1 = 0
> 0
SIGN = SGN (DY * COSH1 - DX * SINH1)
SIGNR1 = SIGN * R1
XC = X1 - SIGNR1 * SINH1
YC = Y1 + SIGNR1 * COSH1
D = (X2 - XC)^2 + (Y2 - YC)^2 - R1^2
D:0
D < 0
WRITE: "WP's too close" X1, X2, Y1, Y2
RETURN1
D ≥ 0
D = (D)^1/2
ANGLE1 = PIMOD(H1 - SIGN * HALFPI)
ANGLE2 = ATAN2 (Y2-YC, X2-XC)
SIGN ≥ 0
SIGN:0
SIGN < 0
ANGLE2: ANGLE1
ANGLE2 < ANGLE1
ANGLE2 = ANGLE2 + TWOPI
ANGLE2: ANGLE1
ANGLE2 > ANGLE1
ANGLE2 = ANGLE2 - TWOPI
ANGLE3 = ATAN (D/R1)
TURN1 = ANGLE2 - ANGLE1 - SIGN * ANGLE3
DELD1 = R1 * ABS(TURN1)
H2 = PIMOD (H1 + TURN1)
XQ1 = XC + SIGNR1 * SIN(H2)
YQ1 = YC - SIGNR1 * COS(H2)
RETURN

C2

```
      SUBROUTINE NEWPSI(X1,Y1,H1,R1,X2,Y2,TURN1,DELD1,XQ1,YQ1,H2,*)
      HALFPI=1.5708
      TWOPI=6.28319
      SINH1=SIN(H1)
      COSH1=COS(H1)
      DY=Y2-Y1
      DX=X2-X1
      TEMP=ABS(DX)+ABS(DY)
      IF(TEMP.GT.0.) GO TO 1
         TURN1=0.
         GO TO 5
 1    SIGN=SGN(DY*COSH1-DX*SINH1)
      SIGNR1=SIGN*R1
      XC=X1-SIGNR1*SINH1
      YC=Y1+SIGNR1*COSH1
      D=(X2-XC)*(X2-XC)+(Y2-YC)*(Y2-YC)-R1*R1
      IF(D.GE.0.) GO TO 2
         WRITE(6,10) X1,X2,Y1,Y2
10       FORMAT(///2X,'WAYPOINTS TOO CLOSE'/2F20.1/2F20.1)
         GO TO 6
 2    D=SQRT(D)
      ANGLE1=PIMOD(H1-SIGN*HALFPI)
      ANGLE2=ATAN2(Y2-YC,X2-XC)
      IF(SIGN.LT.0.) GO TO 3
      IF(ANGLE2.LT.ANGLE1) ANGLE2=ANGLE2+TWOPI
      GO TO 4
 3    IF(ANGLE2.GT.ANGLE1) ANGLE2=ANGLE2-TWOPI
 4    ANGLE3=ATAN(D/R1)
      TURN1=ANGLE2-ANGLE1-SIGN*ANGLE3
 5    DELD1=R1*ABS(TURN1)
      H2=PIMOD(H1+TURN1)
      XQ1=XC+SIGNR1*SIN(H2)
      YQ1=YC-SIGNR1*COS(H2)
      RETURN
 6    RETURN 1
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

VRANGE

Required function:

This routine computes the minimum and maximum admissible airspeeds at each specified waypoint.

Method:

First the wind component along the ground track at the beginning and end of each turn is computed. Then, starting with the last waypoint and proceeding sequentially backward, a minimum and maximum airspeed at the previous waypoint are generated under the following assumptions: the desired minimum and maximum cruising speeds (VLOWER and VUPPER) are maintained as long as possible, at the last waypoint the desired speed VFINAL is achieved, and speed changes occur at the maximum rate.

Input data:

CURWP1	index of waypoint immediately following the capture waypoint (=CURWPT+1)
NWP	index of last waypoint
NWP1	=NWP+1
VW	estimated wind magnitude, ft/sec
HW	estimated wind direction, radian
H_i	headings of straight-line segments in ground track, radian (i=CURWP1,CURWP1+1, . . . ,NWP1)
VFINAL	desired airspeed at last waypoint, ft/sec
VLOWER	desired minimum cruising speed, ft/sec
VUPPER	desired maximum cruising speed, ft/sec
MAXDEC	maximum admissible deceleration, ft/sec^2 (negative)

Output data:

$VMIN_i$, $VMAX_i$	minimum and maximum airspeeds at waypoint i,i=1,2, . . . ,NWP; ft/sec
DVW_i	component of the wind in the direction H_i, i=CURWP1, CURWP1+1, . . . ,NWP1; ft/sec

Temporary variables:

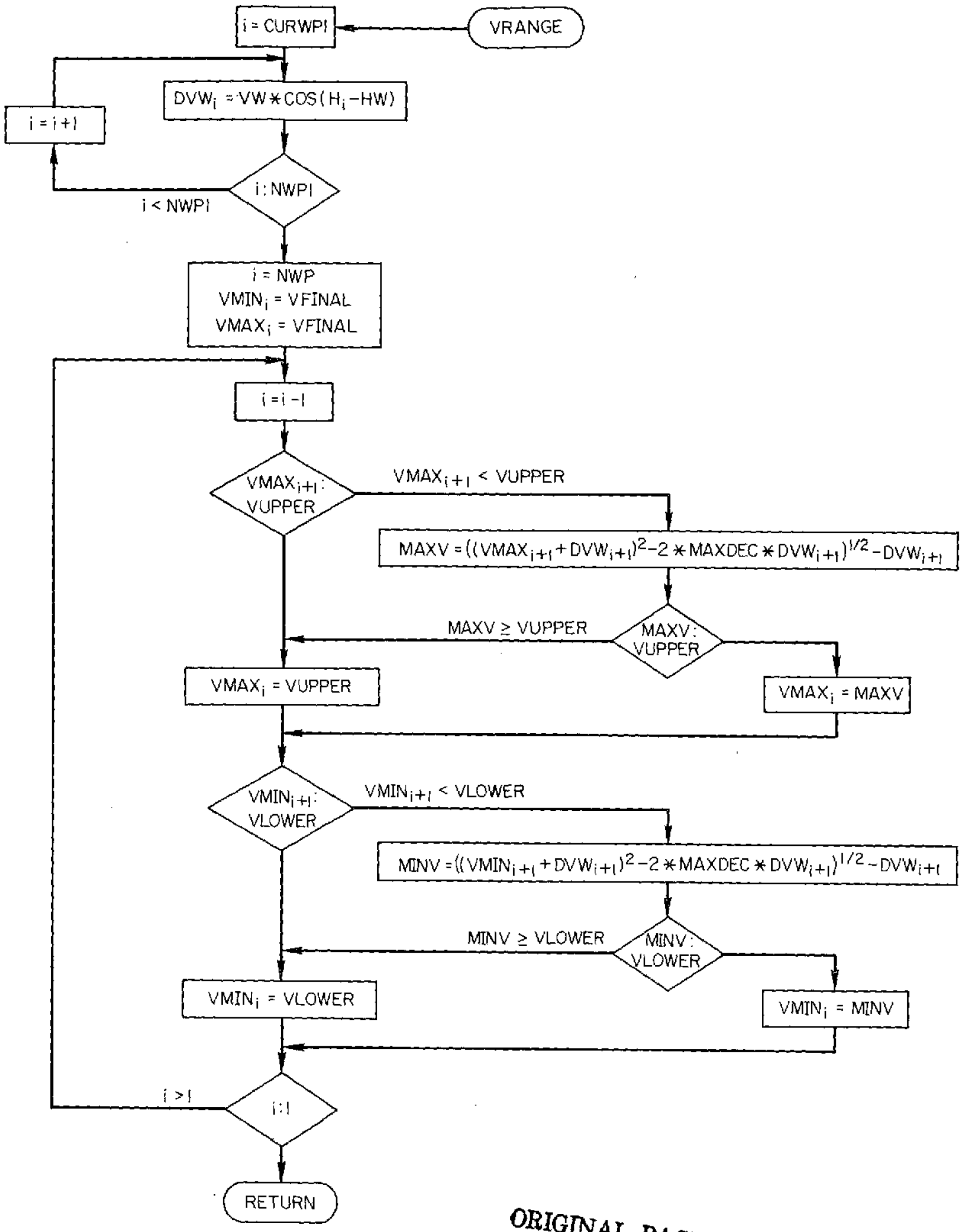
MAXV maximum airspeed at point Q_i from which aircraft could decelerate at a rate MAXDEC and achieve $VMAX_{i+1}$ at point P_{i+1}, ft/sec

MINV maximum airspeed at point Q_i from which aircraft could decelerate at a rate MAXDEC and achieve $VMIN_{i+1}$ at point P_{i+1}, ft/sec

Constants: None

Subroutines required: None

Called by: FOURD



ORIGINAL PAGE IS OF POOR QUALITY

```
      SUBROUTINE VRANGE
      COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
     1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
     2VLOWER,CURWPT,CURWP1,NWP1,I,RO,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
     3CAPSEG,DAC,TURNAC,RAC,DELDAC,XOAC,YOAC,ZOAC,HWP,GAMWP,DWP,XPWP,
     4YPWP,ZPWP,TURNWP,RWP,DELDWP,XOWP,YOWP,ZOWP,INDEX(20),XWP(20),
     5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
     6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
     7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
     8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
     9TRNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
      REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
      INTEGER CURWPT,CURWP1,CAPSEG,CRCD
      DO 20 I=CURWP1,NWP1
20       DVW(I)=VW*COS(H(I)-HW)
      I=NWP
      VMIN(I)=VFINAL
      VMAX(I)=VFINAL
30    I=I-1
      IF(VMAX(I+1).GE.VUPPER) GO TO 40
         MAXV=SQRT((VMAX(I+1)+DVW(I+1))**2-2.*MAXDEC*D(I+1))-DVW(I+1)
         IF(MAXV.GE.VUPPER) GO TO 40
            VMAX(I)=MAXV
            GO TO 50
40    VMAX(I)=VUPPER
50    IF(VMIN(I+1).GE.VLOWER) GO TO 60
         MINV=SQRT((VMIN(I+1)+DVW(I+1))**2-2.*MAXDEC*D(I+1))-DVW(I+1)
         IF(MINV.GE.VLOWER) GO TO 60
            VMIN(I)=MINV
            GO TO 70
60    VMIN(I)=VLOWER
70    IF(I.GT.1) GO TO 30
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

TRANGE

Required function:

The routine computes the minimum and maximum feasible flight times from each waypoint to the last one.

Method:

The minimum and maximum flight times are computed by use of subroutine TRANS. Starting with the last waypoint and proceeding sequentially backward, RO is first set to zero. This implies the maximum feasible speed profile; calling TRANS generates the corresponding minimum flight time between adjacent waypoints. RO is then set to 1 and the process is repeated for the maximum flight time between adjacent waypoints.

Input data:

CURWP1 index of waypoint immediately following the capture waypoint

NWP index of last waypoint

Also, all inputs required for subroutine TRANS.

Output data:

$TMIN_i$ minimum feasible flight time from waypoint i (actually, from point Q_i) to last waypoint (to point Q_{NWP}), sec

$TMAX_i$ maximum feasible flight time from waypoint i (actually, from point Q_i) to last waypoint (to point Q_{NWP}), sec

Temporary variables:

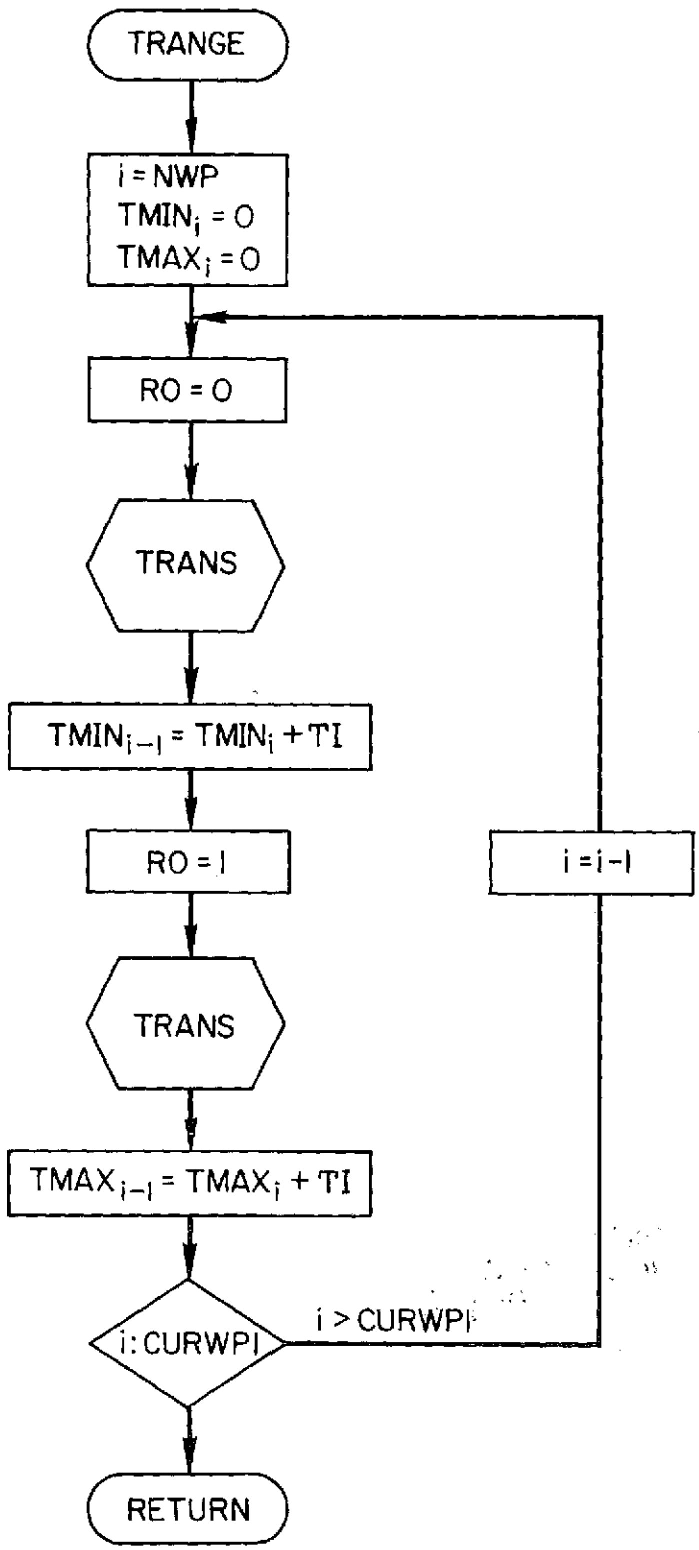
RO (see input data for TRANS)

TI flight time from point Q_{i-1} to point Q_i corresponding to a particular value of RO (TI is the output of TRANS), sec

Constants: None

Subroutines required: TRANS

Called by: FOURD


TRANGE
i = NWP
TMIN$_i$ = 0
TMAX$_i$ = 0
RO = 0
TRANS
TMIN$_{i-1}$ = TMIN$_i$ + TI
RO = 1
TRANS
TMAX$_{i-1}$ = TMAX$_i$ + TI
i : CURWPI
i > CURWPI
i = i - 1
RETURN

```
      SUBROUTINE TRANGE
      COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
     1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
     2VLOWER,CURWPT,CURWP1,NWP1,I,RO,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
     3CAPSEG,DAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC,HWP,GAMWP,DWP,XPWP,
     4YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP,INDEX(20),XWP(20),
     5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
     6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
     7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
     8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
     9TRNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
      REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
      INTEGER CURWPT,CURWP1,CAPSEG,CRCD
      I=NWP
      TMIN(I)=0.
      TMAX(I)=0.
10    RO=0.
      CALL TRANS
      TMIN(I-1)=TMIN(I)+TI
      RO=1.
      CALL TRANS
      TMAX(I-1)=TMAX(I)+TI
      IF(I.LE.CURWP1) RETURN
        I=I-1
        GO TO 10
      END
```

SPEED

Required function:

The purpose of this routine is threefold. First, the value of RO that yields the desired flight time from the capture waypoint to the last waypoint is determined. This is accomplished by an iterative process. Once the proper RO is found, the 4-D reference commands are assembled in a chronological sequence. Finally, the appropriate lead times are computed to compensate for finite roll and pitch rates.

Method:

The iterative procedure to find RO starts with an initial value and computes the corresponding flight time from the capture waypoint to the last waypoint. If the resulting flight time differs from the desired value by more than 1 sec (arbitrary), then RO is updated and the flight-time computation is repeated.

After the correct value of RO is found, the 4-D reference commands are assembled between each pair of adjacent waypoints, starting with the capture waypoint and proceeding to the last waypoint. This operation results in four arrays that contain the sequences of piecewise constant 4-D guidance commands for the entire flight path. Note that any command whose time duration is less than 0.1 sec was omitted from the final sequence.

The last block of computations for each waypoint involves the determination of the appropriate lead times for finite roll and pitch rate compensation. Since the lead time for roll-rate compensation depends on ground speed, and since the ground speeds at the beginning and end of a turn are generally different, two lead times are computed for each turn — one applies to the roll into the turn, the other, to the roll out of the turn.

Input data:

CURWPT	index of capture waypoint
CURWP1	=CURWPT+1
NWP	index of last waypoint
$TMIN_{CURWPT}$	minimum feasible flight time from the capture waypoint to the last waypoint, sec
$TMAX_{CURWPT}$	maximum feasible flight time from the capture waypoint to the last waypoint, sec

$ENRTIM_{CURWPT}$	desired flight time from the capture waypoint to the last waypoint, sec ($TMIN_{CURWPT} \leq ENRTIM_{CURWPT} \leq TMAX_{CURWPT}$)
$CRCD_{CURWPT}$	index of the reference command applied at the capture waypoint; this variable initializes a pointer for the sequence of 4-D reference commands
MAXACC(MAXDEC)	maximum admissible acceleration (deceleration), ft/sec^2
R_i	turning radius at waypoint i, ft
$TURN_i$	direction and extent of turn at waypoint i, radian
GAM_i	flight-path angle from point Q_{i-1} to point Q_i, radian
$VP_i(VQ_i)$	ground speed at beginning (end) of the turn at waypoint i, ft/sec
PHIDOT	maximum admissible roll rate, radian/sec
VA_{NWP}	desired airspeed at last waypoint, ft/sec (same as VFINAL)
DVW_{NWP1}	(see output data of subroutine VRANGE)

Also, all other inputs required for the subroutine TRANS.

Output data:

$ENRTIM_i$	flight time from waypoint i (from point Q_i) to the last waypoint, sec
VQ_{NWP}	ground speed at last waypoint, ft/sec
$ACCREF_k$	constant value of the reference rate of change of airspeed during the kth command interval, ft/sec^2
$TRNREF_k$	constant value of the signed reference turning radius (or its inverse) during the kth command interval, ft
$GAMREF_k$	constant value of the reference flight-path angle during the kth command interval, radian
$TIMREF_k$	time duration of the kth command interval, sec
$TLEADT_i$	lead-time compensation for finite roll rate prior to rolling into the turn at waypoint i, sec
$TLEADS_i$	lead-time compensation for finite roll rate prior to rolling out of the turn at waypoint i, sec

$TLEADP_i$	lead-time compensation for finite pitch rate prior to changing flight-path angle from GAM_i to GAM_{i+1} at waypoint i (actually at point Q_i), sec
$CRCD_i$	index of the reference command applied at waypoint i (actually at point Q_i)
MAXSEG	total number of nondegenerate guidance commands needed to generate the complete 4-D flight path

Temporary variables:

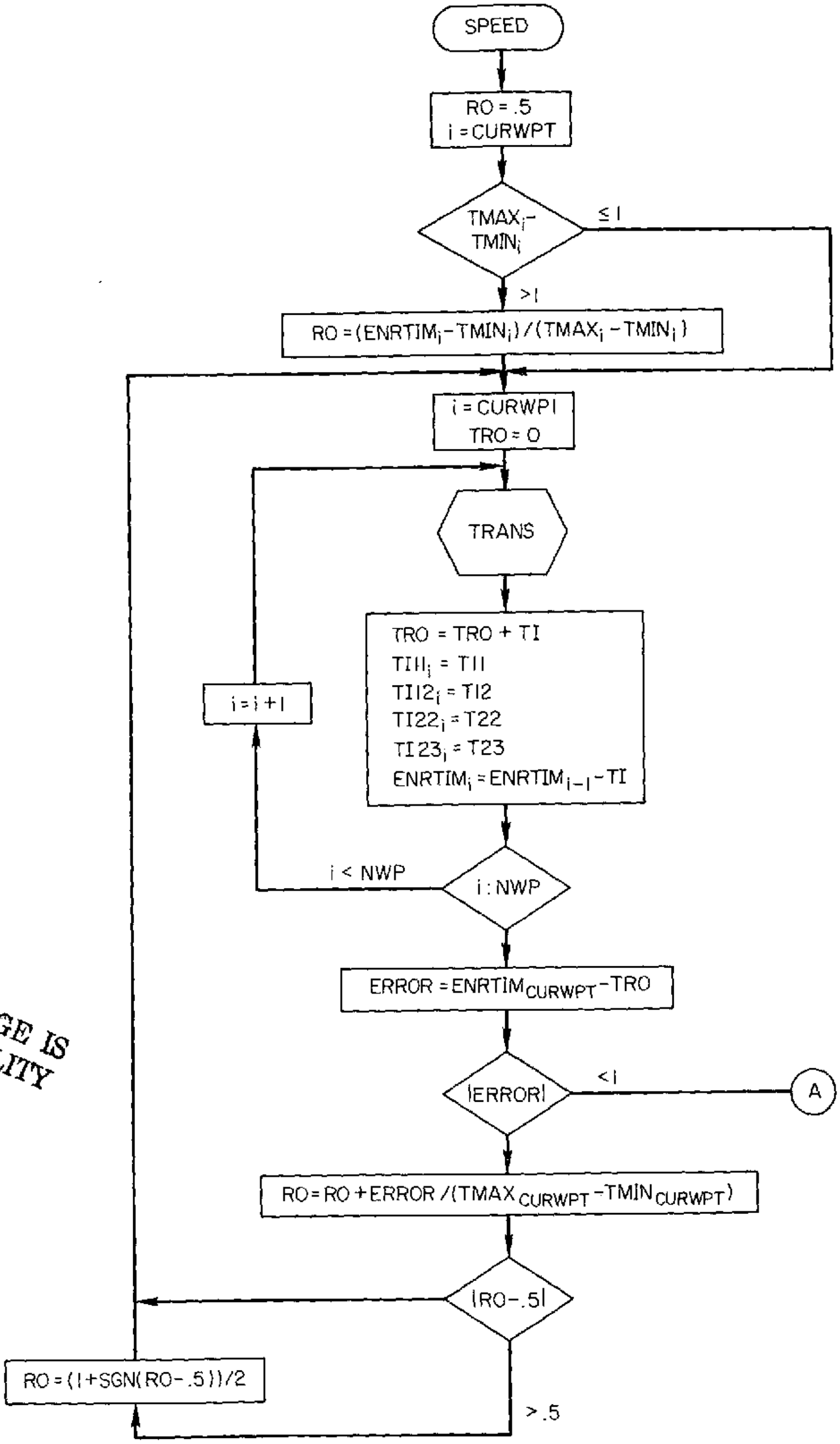
RO	(see input data for TRANS)
TRO	flight time from the capture waypoint to the last waypoint corresponding to any given value of RO, sec
TI11,TI12, TI22,TI23,	time duration of the four guidance intervals from point Q_{i-1} to point Q_i corresponding to any given value of RO, sec
ERROR	difference between desired flight time from the capture waypoint to the last waypoint and that corresponding to any given value of RO, sec

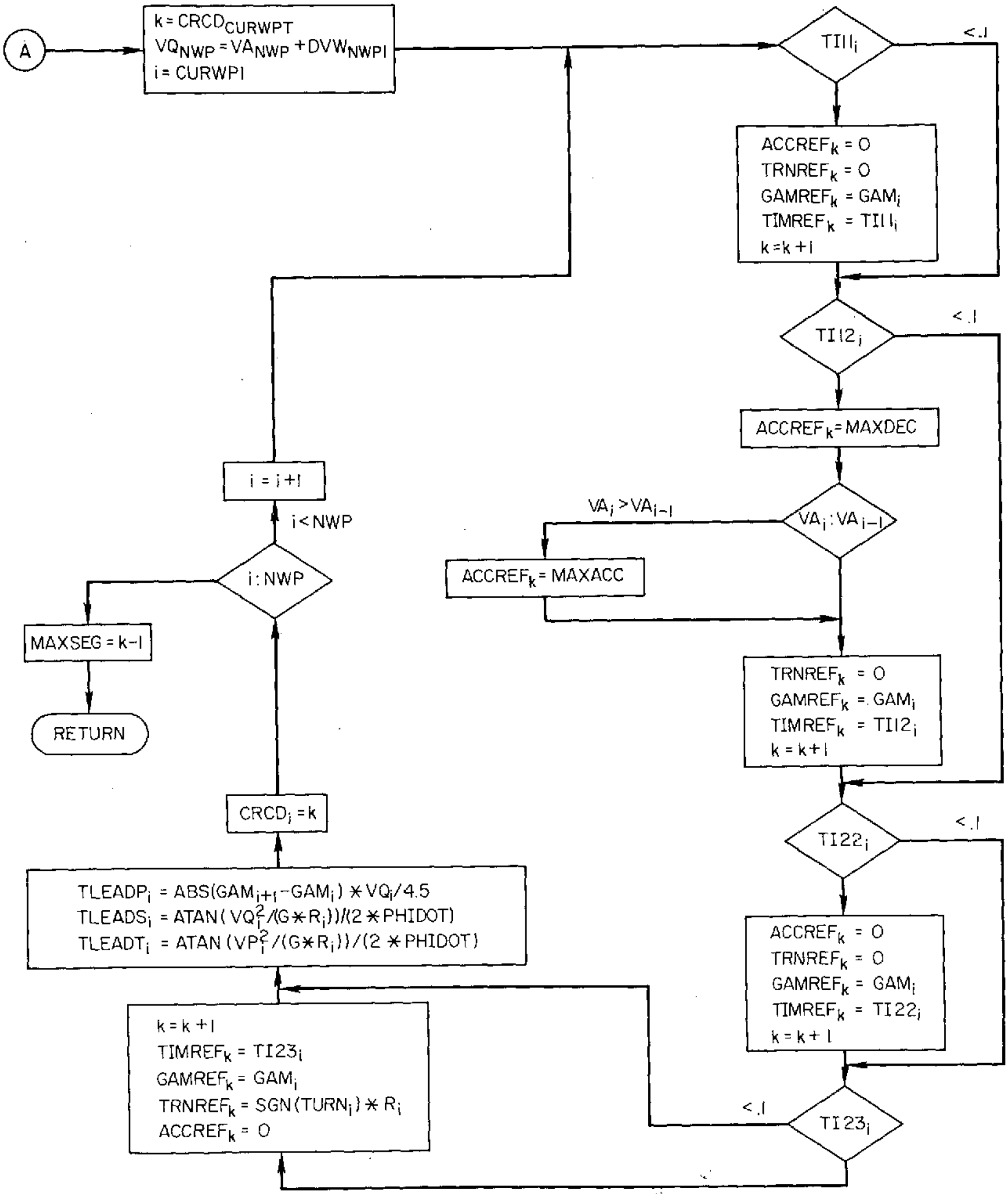
Constants:

G	acceleration due to gravity, 32.172 ft/sec^2

Subroutines required: TRANS

Called by: FOURD


SPEED
RO = .5
i = CURWPT
TMAX_i - TMIN_i
≤ I
> I
RO = (ENRTIM_i - TMIN_i) / (TMAX_i - TMIN_i)
i = CURWPI
TRO = O
TRANS
TRO = TRO + TI
TI11_i = T11
TI12_i = T12
TI22_i = T22
TI23_i = T23
ENRTIM_i = ENRTIM_i-1 - TI
i = i + 1
i < NWP
i : NWP
ERROR = ENRTIM_CURWPT - TRO
|ERROR|
< I
A
RO = RO + ERROR / (TMAX_CURWPT - TMIN_CURWPT)
|RO - .5|
RO = (I + SGN(RO - .5)) / 2
> .5



ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE SPEED
      COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
     1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
     2VLOWER,CURWPT,CURWP1,NWP1,I,RO,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
     3CAPSEG,DAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC,HWP,GAMWP,DWP,XPWP,
     4YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP,INDEX(20),XWP(20),
     5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
     6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
     7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
     8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
     9TRNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
      REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
      INTEGER CURWPT,CURWP1,CAPSEG,CRCD
      RO=0.5
      IF(TMAX(CURWPT)-TMIN(CURWPT).GT.1.)
     1RO=(ENRTIM(CURWPT)-TMIN(CURWPT))/(TMAX(CURWPT)-TMIN(CURWPT))
   10 I=CURWP1
      TRO=0.
   20 CALL TRANS
      TRO=TRO+TI
      TI11(I)=T11
      TI12(I)=T12
      TI22(I)=T22
      TI23(I)=T23
      ENRTIM(I)=ENRTIM(I-1)-TI
      IF(I.GE.NWP) GO TO 30
         I=I+1
         GO TO 20
   30 ERROR=ENRTIM(CURWPT)-TRO
      IF(ABS(ERROR).LT.1.) GO TO 40
         RO=RO+ERROR/(TMAX(CURWPT)-TMIN(CURWPT))
         IF(ABS(RO-0.5).GT.0.5) RO=(1.+SGN(RO-0.5))/2.
         GO TO 10
   40 K=CRCD(CURWPT)
      VQ(NWP)=VA(NWP)+DVW(NWP1)
      DO 90 I=CURWP1,NWP
         IF(TI11(I).LT.0.1) GO TO 50
            ACCREF(K)=0.
            TRNREF(K)=0.
            GAMREF(K)=GAM(I)
            TIMREF(K)=TI11(I)
            K=K+1
   50    IF(TI12(I).LT.0.1) GO TO 60
            ACCREF(K)=MAXDEC
            IF(VA(I).GT.VA(I-1)) ACCREF(K)=MAXACC
```

```
         TRNREF(K)=0.
         GAMREF(K)=GAM(I)
         TIMREF(K)=TI12(I)
         K=K+1
60    IF(TI22(I).LT.0.1) GO TO 70
         ACCREF(K)=0.
         TRNREF(K)=0.
         GAMREF(K)=GAM(I)
         TIMREF(K)=TI22(I)
         K=K+1
70    IF(TI23(I).LT.0.1) GO TO 80
         ACCREF(K)=0.
         TRNREF(K)=SGN(TURN(I))*R(I)
         GAMREF(K)=GAM(I)
         TIMREF(K)=TI23(I)
         K=K+1
80    TLEADT(I)=ATAN((VP(I)*VP(I))/(G*R(I)))/(2.*PHIDOT)
      TLEADS(I)=ATAN((VQ(I)*VQ(I))/(G*R(I)))/(2.*PHIDOT)
      TLEADP(I)=ABS(GAM(I+1)-GAM(I))*VQ(I)/4.5
90  CRCD(I)=K
    MAXSEG=K-1
    RETURN
    END
```

TTTURN

Required function:

Given the parameters of a circular turn (i.e., initial and final headings and turning radius), the magnitude and direction of the estimated wind, and a desired constant airspeed during the turn, this routine computes the time required to complete the turn.

Method:

The time spent during the turn is given by expression (87). Since the $\tan^{-1}(\cdot)$ functions in (87) are one-to-many, additional logic must be incorporated to arrive at the correct result. The final values of BETA2 and BETA1 represent the first and second $\tan^{-1}(\cdot)$ functions on the right side of equation (87), respectively.

Input data:

HSTART	initial heading at beginning of the turn, radian
TRN	direction and angular extent of the turn, radian
RTRN	radius of the turn, ft
VWIND	magnitude of wind, ft/sec
HWIND	direction of wind, radian
VAIR	desired constant airspeed during turn, ft/sec

Output data:

TRNTIM	time spent during the turn, sec

Temporary variables:

C1	ground speed corresponding to the given airspeed and wind when the aircraft heading is parallel to the wind heading, ft/sec
C2	ground speed corresponding to the given airspeed and wind when the aircraft heading is perpendicular to the wind heading, ft/sec
ALPHA1	difference between initial heading and wind direction, radian
ALPHA2	=ALPHA1+TRN, radian

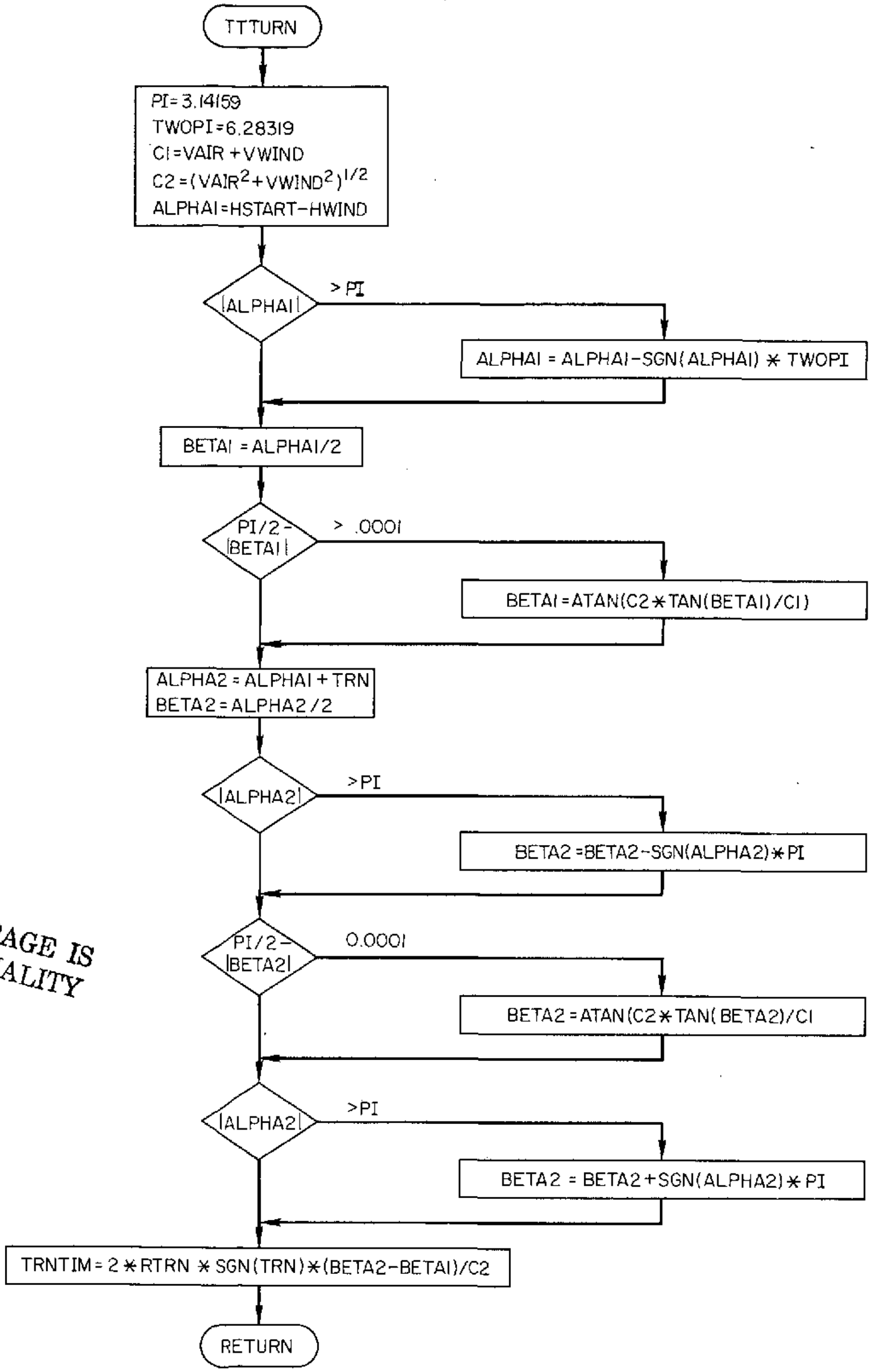
BETA1	value of the second $\tan^{-1}(\cdot)$ function in equation (87), radian
BETA2	value of the first $\tan^{-1}(\cdot)$ function in equation (87), radian

Constants:

PI	=3.14159
TWOPI	=6.28319

Subroutines required: None

Called by: TRANS,TST


TTTURN
PI=3.14159
TWOPI=6.28319
C1=VAIR+VWIND
C2=(VAIR^2+VWIND^2)^1/2
ALPHA1=HSTART−HWIND
|ALPHA1|
>PI
ALPHA1 = ALPHA1−SGN(ALPHA1) * TWOPI
BETA1 = ALPHA1/2
PI/2−|BETA1|
> .0001
BETA1=ATAN(C2*TAN(BETA1)/C1)
ALPHA2 = ALPHA1 + TRN
BETA2 = ALPHA2/2
|ALPHA2|
>PI
BETA2 =BETA2−SGN(ALPHA2)*PI
PI/2−|BETA2|
0.0001
BETA2 = ATAN(C2*TAN(BETA2)/C1
|ALPHA2|
>PI
BETA2 = BETA2+SGN(ALPHA2)*PI
TRNTIM = 2*RTRN*SGN(TRN)*(BETA2−BETA1)/C2
RETURN

```
SUBROUTINE TTTURN(HSTART,TRN,RTRN,VWIND,HWIND,VAIR,TRNTIM)
PI=3.14159
TWOPI=6.28319
C1=VAIR+VWIND
C2=SQRT(VAIR*VAIR+VWIND*VWIND)
ALPHA1=HSTART-HWIND
IF(ABS(ALPHA1).GT.PI) ALPHA1=ALPHA1-SGN(ALPHA1)*TWOPI
BETA1=ALPHA1/2.
IF(PI/2.-ABS(BETA1).GT.0.0001) BETA1=ATAN(C2*TAN(BETA1)/C1)
ALPHA2=ALPHA1+TRN
BETA2=ALPHA2/2.
IF(ABS(ALPHA2).GT.PI) BETA2=BETA2-SGN(ALPHA2)*PI
IF(PI/2.-ABS(BETA2).GT.0.0001) BETA2=ATAN(C2*TAN(BETA2)/C1)
IF(ABS(ALPHA2).GT.PI) BETA2=BETA2+SGN(ALPHA2)*PI
TRNTIM=2.*RTRN*SGN(TRN)*(BETA2-BETA1)/C2
RETURN
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

Required function:

Given a value of $i, 2 \le i \le NWP$, certain parameters of the 3-D flight path between waypoints (i-1) and i, the minimum and maximum airspeeds $VMIN_{i-1}$, $VMIN_i$, $VMAX_{i-1}$, $VMAX_i$, and a specific value of RO, this routine computes the transfer time from point Q_{i-1} to point Q_i.

Method:

First the airspeeds at waypoints i-1 and i are computed using the minimum and maximum admissible airspeeds and the speed level parameter RO. Then the corresponding ground speeds at the beginning and end of the straight flight segment are determined. Next, the parameter RO is used to determine the point in the straight flight segment at which the necessary speed change should begin. Once the lengths of the constant speed and constant speed-change segments are known, the corresponding time durations are calculated. In order to determine the flight time during the turn, subroutine TTTURN is called.

Input data:

i	waypoint number indicating the second of a pair of adjacent waypoints between which the transfer time is to be computed, $2 \le i \le NWP$
$VMIN_{i-1}$ ($VMAX_{i-1}$)	minimum (maximum) admissible airspeed at waypoint i-1, ft/sec
$VMIN_i$, $VMAX_i$	minimum (maximum) admissible airspeed at waypoint i, ft/sec
RO	fraction indicating speed profile deviation from maximum admissible speed profile, $0 \le RO \le 1$
DVW_i	component of wind along the heading H_i, ft/sec
MAXACC(MAXDEC)	maximum admissible acceleration (deceleration), ft/sec^2
D_i	length of straight-line segment from point Q_{i-1} to point P_i, ft
H_i	heading of straight-line segment from point Q_{i-1} to point P_i, radian
$TURN_i$	direction and magnitude of turn at waypoint i, radian

R_i radius of turn at waypoint i, radian

VW wind magnitude, ft/sec

HW wind direction, radian

Output data:

VA_{i-1}, VA_i airspeeds at waypoints $i-1$ and i, ft/sec

VQ_{i-1}, VP_i ground speeds at points Q_{i-1} and P_i, ft/sec

T11 durations of straight flight flown at constant ground speed VQ_{i-1}, sec

T12 duration of straight flight during which speed is changed from VQ_{i-1} to VP_i, sec

T22 duration of straight flight flown at constant ground speed VP_i, sec

T23 duration of curved flight flown at constant airspeed VA_i, sec

TI duration of flight from point Q_{i-1} to point Q_i, sec

Temporary variables:

A12 rate of change of speed occurring in straight flight, ft/sec^2

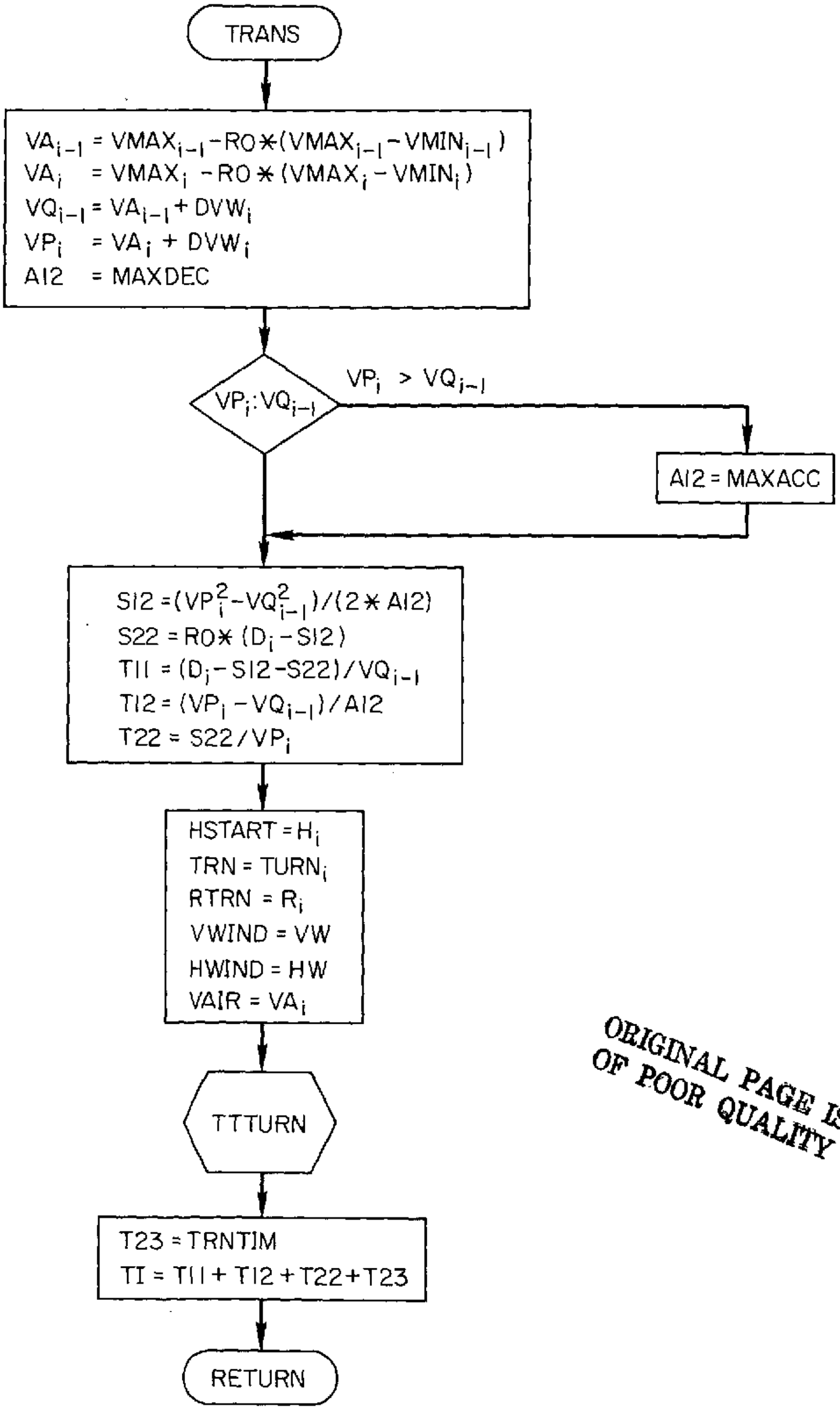
S12 distance required to change speed from VQ_{i-1} to VP_i at a rate given by A12, ft

S22 length of straight flight flown at constant ground speed VP_i, ft

Constants: None

Subroutines required: TTTURN

Called by: TRANGE and SPEED


TRANS
VA_{i-1} = VMAX_{i-1} - RO*(VMAX_{i-1} - VMIN_{i-1})
VA_i = VMAX_i - RO*(VMAX_i - VMIN_i)
VQ_{i-1} = VA_{i-1} + DVW_i
VP_i = VA_i + DVW_i
AI2 = MAXDEC
VP_i : VQ_{i-1}
VP_i > VQ_{i-1}
AI2 = MAXACC
SI2 = (VP_i^2 - VQ_{i-1}^2)/(2*AI2)
S22 = RO*(D_i - SI2)
TI1 = (D_i - SI2 - S22)/VQ_{i-1}
TI2 = (VP_i - VQ_{i-1})/AI2
T22 = S22/VP_i
HSTART = H_i
TRN = TURN_i
RTRN = R_i
VWIND = VW
HWIND = HW
VAIR = VA_i
TTTURN
T23 = TRNTIM
TI = TI1 + TI2 + T22 + T23
RETURN

```
 SUBROUTINE TRANS
 COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
2VLOWER,CURWPT,CURWP1,NWP1,I,RO,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
3CAPSEG,DAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC,HWP,GAMWP,DWP,XPWP,
4YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP,INDEX(20),XWP(20),
5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
9TRNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
 REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
 INTEGER CURWPT,CURWP1,CAPSEG,CRCD
 VA(I-1)=VMAX(I-1)-RO*(VMAX(I-1)-VMIN(I-1))
 VA(I)=VMAX(I)-RO*(VMAX(I)-VMIN(I))
 VQ(I-1)=VA(I-1)+DVW(I)
 VP(I)=VA(I)+DVW(I)
 A12=MAXDEC
 IF(VP(I).GT.VQ(I-1)) A12=MAXACC
 S12=(VP(I)*VP(I)-VQ(I-1)*VQ(I-1))/(2.*A12)
 S22=RO*(D(I)-S12)
 T11=(D(I)-S12-S22)/VQ(I-1)
 T12=(VP(I)-VQ(I-1))/A12
 T22=S22/VP(I)
 CALL TTTURN(H(I),TURN(I),R(I),VW,HW,VA(I),T23)
 TI=T11+T12+T22+T23
 RETURN
 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

Required function:

This routine is used to generate a 4-D capture flight path from the current aircraft state to one of the waypoints (capture waypoint) on the precomputed flight path.

Method:

The capture flight path is computed in several distinct stages. After proper initialization, subroutine NEWPSI is used in an iterative fashion to generate the ground track. If at any stage of the iterative process the ground track is not feasible (because the aircraft is located too near the capture waypoint), a diagnostic message is printed and control is returned to the calling program.

After the ground-track parameters have been computed, the constant flight-path angle is determined. If this angle falls outside the permissible range, a diagnostic message is printed and control is returned to the calling program.

Next it is checked whether the ground track (its straight segment) is long enough to allow the required speed change. If not, a diagnostic message is printed and control is returned to the calling program.

The last stage of the process generates the sequence of 4-D reference commands for the capture flight path and the lead times associated with the capture waypoint. This stage is analogous to the last stage of subroutine SPEED.

Input data:

XAC,YAC,ZAC	x,y,z-coordinates of aircraft position, ft
HAC	aircraft heading, radian
VAC	airspeed of aircraft, ft/sec
CURWPT	index of the capture waypoint, 1≤CURWPT≤NWP
VW	wind magnitude, ft/sec
HW	wind direction, radian
XQ_{CURWPT}, YQ_{CURWPT}, ZQ_{CURWPT}	x,y,z-coordinates of capture waypoint (actually of the end of the turn associated with the capture waypoint), ft

VA_{CURWPT}	airspeed at capture waypoint, ft/sec
H_{CURWP1}	heading of straight-flight segment immediately following the capture waypoint, radian
MINGAM(MAXGAM)	minimum (maximum) admissible flight-path angle, radian
DVW_{CURWP1}	wind component along the heading H_{CURWP1}, ft/sec
MAXACC(MAXDEC)	maximum admissible acceleration (deceleration), ft/sec^2
PHIDOT	maximum admissible roll rate, radian/sec
VQ_{CURWPT}	ground speed at the end of the turn associated with th capture waypoint, ft/sec
GAM_{CURWP1}	flight-path angle immediately following the capture waypoint, radian

Output data:

TURNAC	direction and magnitude of first turn in the capture flight path, radian
RAC	turning radius of first turn in capture flight path, ft
DELDAC	arclength of TURNAC, ft
XQAC,YQAC,ZQAC	x,y,z-coordinates of end of first turn in capture flight path, ft
VQAC	ground speed at end of first turn in capture flight path, ft/sec
HWP	heading of straight segment in capture flight path, radian
GAMWP	constant flight-path angle of the capture flight path, radian
DWP	length of straight segment in capture flight path, ft
XPWP,YPWP,ZPWP	x,y,z-coordinates of beginning of second turn in capture flight path, ft
TURNWP	direction and magnitude of second turn in capture flight path, radian
RWP	turning radius of second turn in capture flight path, ft

DELDWP	arclength of TURNWP, ft
XQWP,YQWP,ZQWP	x,y,z-coordinates of end of second turn in capture flight path (same as XQ_{CURWPT},YQ_{CURWPT},ZQ_{CURWPT}), ft
VPWP	ground speed at beginning of second turn in capture flight path, ft/sec
$ACCREF_k$	constant value of the reference rate of change of airspeed during the kth command interval, ft/sec^2
$TRNREF_k$	constant value of the signed reference turning radius (or its inverse) during the kth command interval, ft
$GAMREF_k$	constant value of reference flight-path angle during the kth command interval, radian
$TIMREF_k$	time duration of the kth command interval, sec
$TLEADT_1$	lead-time compensation for finite roll rate prior to rolling into the turn at the capture waypoint, sec
$TLEADS_1$	lead-time compensation for finite roll rate prior to rolling out of the turn at the capture waypoint, sec
$TLEADP_1$	lead-time compensation for finite pitch rate prior to changing flight-path angle from GAMWP to GAM_{CURWP1} at the capture waypoint, sec
CAPSEG	total number of nondegenerate guidance commands needed to generate the 4-D capture flight path, $1 \leq CAPSEG \leq 4$

Temporary variables:

VWP	airspeed of capture waypoint, ft/sec
HH2	reverse of heading H_{CURWP1}, radian
HH1	heading of straight line generated by NEWPSI the first time it is called in the iterative process, radian
TEST	variable that measures the convergence of the iterative process generating the ground track of the capture flight path, radian
DUMMY	wind component along the heading HWP, ft/sec
A12	rate of change of speed occurring in the straight-flight segment of the capture flight path, ft/sec^2

S12 — distance required to change speed from VQAC to VPWP at a rate given by A12, ft

TA — time duration of any one of the command intervals in capture flight path, sec

Constants:

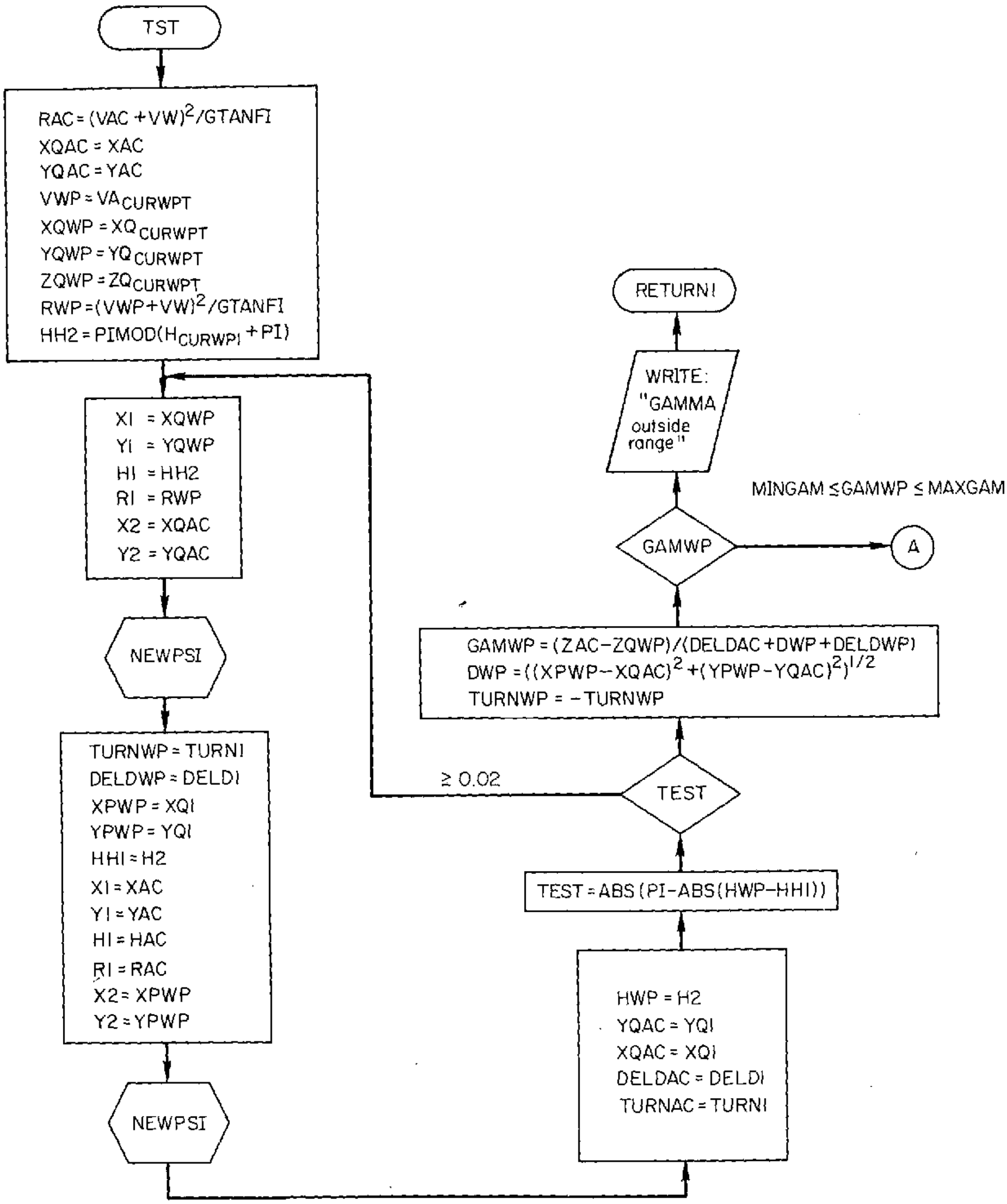
PI — =3.14159

G — acceleration due to gravity, 32.172 ft/sec^2

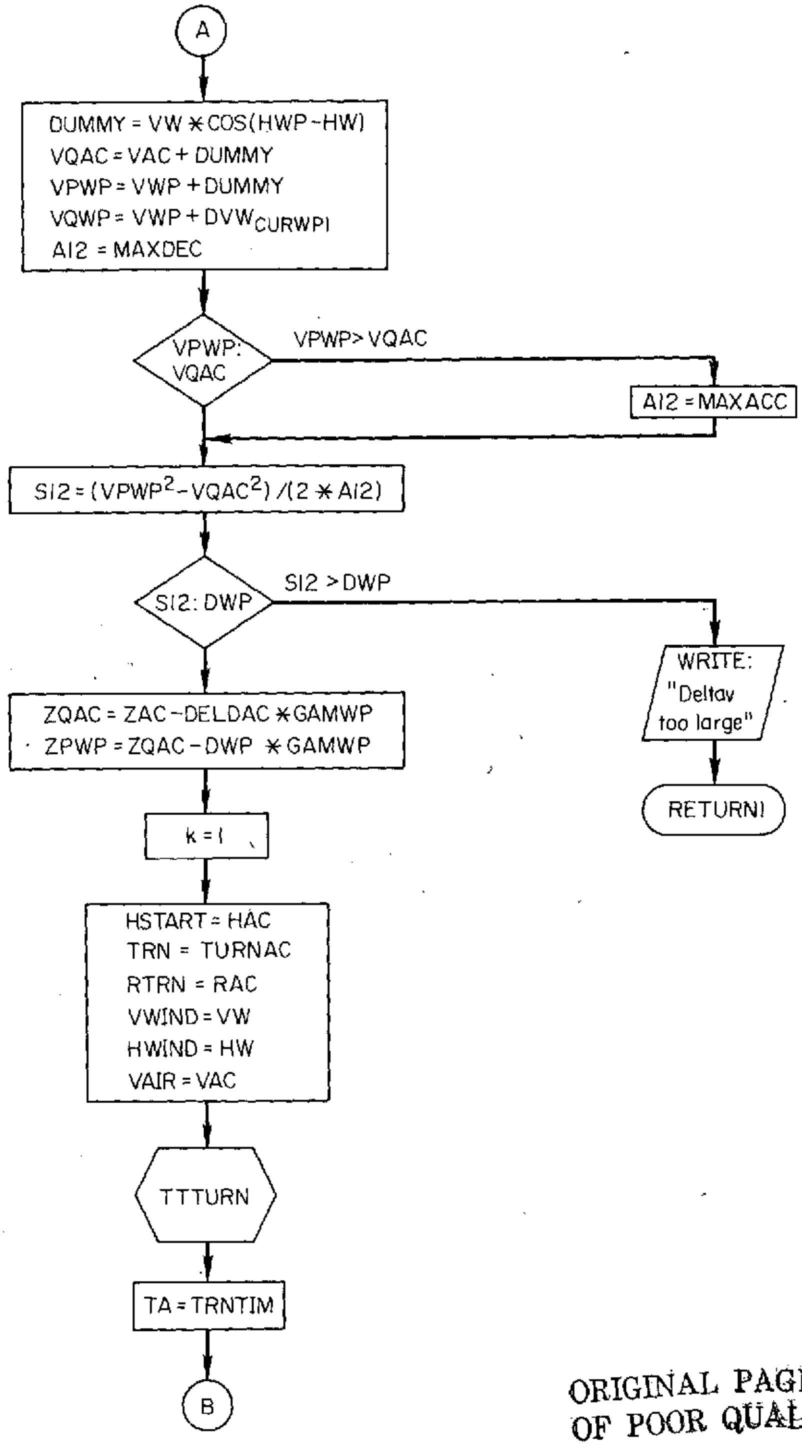
GTANFI — =G*TAN(MAXPHI)

Subroutines required: NEWPSI and TTTURN

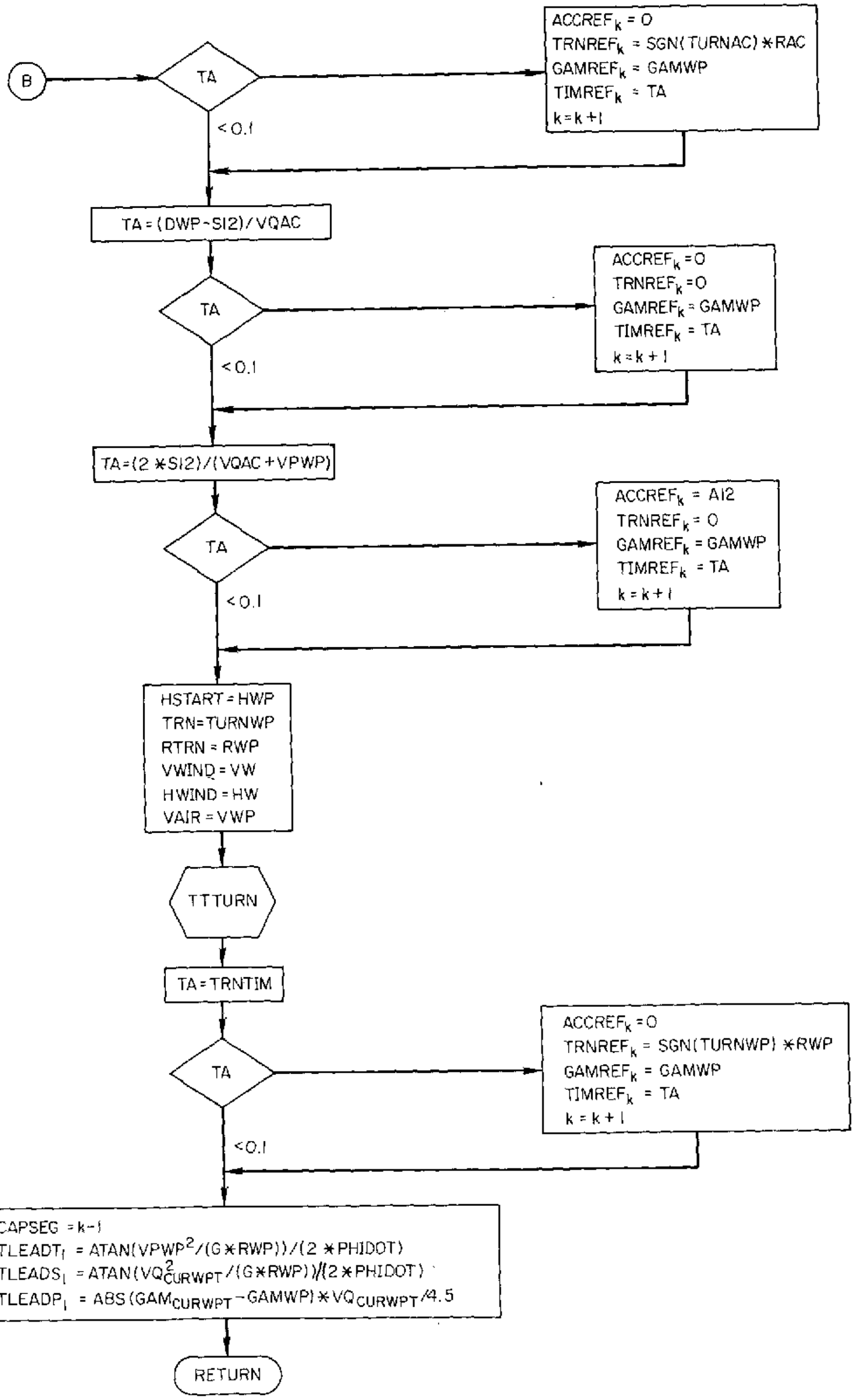
Called by: FOURD



ORIGINAL PAGE IS
OF POOR QUALITY



ORIGINAL PAGE IS
OF POOR QUALITY



ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE TST(*)
      COMMON PI,TWOPI,G,RADDEG,GTANFI,NWP,HFINAL,GFINAL,VFINAL,HW,VW,
     1MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC,PHIDOT,VSNF,VSFF,VUPPER,
     2VLOWER,CURWPT,CURWP1,NWP1,I,RO,TI,XAC,YAC,ZAC,HAC,VAC,MAXSEG,
     3CAPSEG,DAC,TURNAC,RAC,DELDAC,XQAC,YQAC,ZQAC,HWP,GAMWP,DWP,XPWP,
     4YPWP,ZPWP,TURNWP,RWP,DELDWP,XQWP,YQWP,ZQWP,INDEX(20),XWP(20),
     5YWP(20),ZWP(20),R(20),H(21),GAM(21),D(20),XP(20),YP(20),ZP(20),
     6TURN(20),DELD(20),XQ(20),YQ(20),ZQ(20),VMIN(20),VMAX(20),VA(20),
     7VP(20),VQ(20),DVW(21),TMIN(20),TMAX(20),ENRTIM(20),CRCD(20),
     8TI11(20),TI12(20),TI22(20),TI23(20),T11,T12,T22,T23,TIMREF(80),
     9TRNREF(80),GAMREF(80),ACCREF(80),TLEADT(20),TLEADS(20),TLEADP(20)
      REAL MAXPHI,MAXGAM,MINGAM,MAXACC,MAXDEC
      INTEGER CURWPT,CURWP1,CAPSEG,CRCD
      RAC=(VAC+VW)*(VAC+VW)/GTANFI
      XQAC=XAC
      YQAC=YAC
      VWP=VA(CURWPT)
      XQWP=XQ(CURWPT)
      YQWP=YQ(CURWPT)
      ZQWP=ZQ(CURWPT)
      RWP=(VWP+VW)*(VWP+VW)/GTANFI
      HH2=PIMOD(H(CURWP1)+PI)
10    CALL NEWPSI(XQWP,YQWP,HH2,RWP,XQAC,YQAC,TURNWP,DELDWP,XPWP,YPWP,
     1HH1,&100)
      CALL NEWPSI(XAC,YAC,HAC,RAC,XPWP,YPWP,TURNAC,DELDAC,XQAC,YQAC,
     1HWP,&100)
      TEST=ABS(PI-ABS(HWP-HH1))
      IF(TEST.GE.0.02) GO TO 10
      TURNWP=-TURNWP
      DWP=SQRT((XPWP-XQAC)**2+(YPWP-YQAC)**2)
      GAMWP=(ZAC-ZQWP)/(DELDAC+DWP+DELDWP)
      IF(GAMWP.LE.MAXGAM.AND.GAMWP.GE.MINGAM) GO TO 20
         WRITE(6,15)
15       FORMAT(///2X,'GAMMA OUTSIDE RANGE')
         GO TO 100
20    DUMMY=VW*COS(HWP-HW)
      VQAC=VAC+DUMMY
      VPWP=VWP+DUMMY
      VQWP=VWP+DVW(CURWP1)
      A12=MAXDEC
      IF(VPWP.GT.VQAC) A12=MAXACC
      S12=(VPWP*VPWP-VQAC*VQAC)/(2.*A12)
      IF(S12.LE.DWP) GO TO 30
         WRITE(6,25)
25       FORMAT(///2X,'DELTAV TOO LARGE')
```

```
      GO TO 100
 30   ZQAC=ZAC-DELQAC*GAMAP
      ZPWP=ZQAC-DWP*GAMWP
      K=1
 40   CALL ITTURN(HAC,TURNAC,RAC,VW,HW,VAC,TA)
      IF(TA.LT.0.1) GO TO 50
         ACCREF(K)=0.
         TRNREF(K)=SGN(TURNAC)*RAC
         GAMREF(K)=GAMWP
         TIMREF(K)=TA
         K=K+1
 50   TA=(DWP-S12)/VQAC
      IF(TA.LT.0.1) GO TO 60
         ACCREF(K)=0.
         TRNREF(K)=0.
         GAMREF(K)=GAMWP
         TIMREF(K)=TA
         K=K+1
 60   TA=(2.*S12)/(VQAC+VPWP)
      IF(TA.LT.0.1) GO TO 70
         ACCREF(K)=A12
         TRNREF(K)=0.
         GAMREF(K)=GAMWP
         TIMREF(K)=TA
         K=K+1
 70   CALL ITTURN(HWP,TURNWP,RWP,VW,HW,VWP,TA)
      IF(TA.LT.0.1) GO TO 80
         ACCREF(K)=0.
         TRNREF(K)=SGN(TURNWP)*RWP
         GAMREF(K)=GAMWP
         TIMREF(K)=TA
         K=K+1
 80   CAPSEG=K-1
      TLEADT(1)=ATAN((VPWP*VPWP)/(G*RWP))/(2.*PHIDOT)
      TLEADS(1)=ATAN((VQ(CURWPT)*VQ(CURWPT))/(G*RWP))/(2.*PHIDOT)
      TLEADP(1)=ABS(GAM(CURWP1)-GAMWP)*VQ(CURWPT)/4.5
      RETURN
100   RETURN 1
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

PIMOD(XYZ)

Required function:

This function converts an arbitrary angle XYZ to the equivalent angle within the range $-\pi < XYZ \leq \pi$.

Method:

If $XYZ > \pi$, it is decreased by 2π decrements until it falls within the desired range. If $XYZ \leq -\pi$, it is increased by 2π increments until it falls within the desired range.

Input data (argument):

XYZ	any angle, radian

Output data (value of function):

PIMOD(XYZ)	$-\pi < PIMOD(XYZ) \leq \pi$

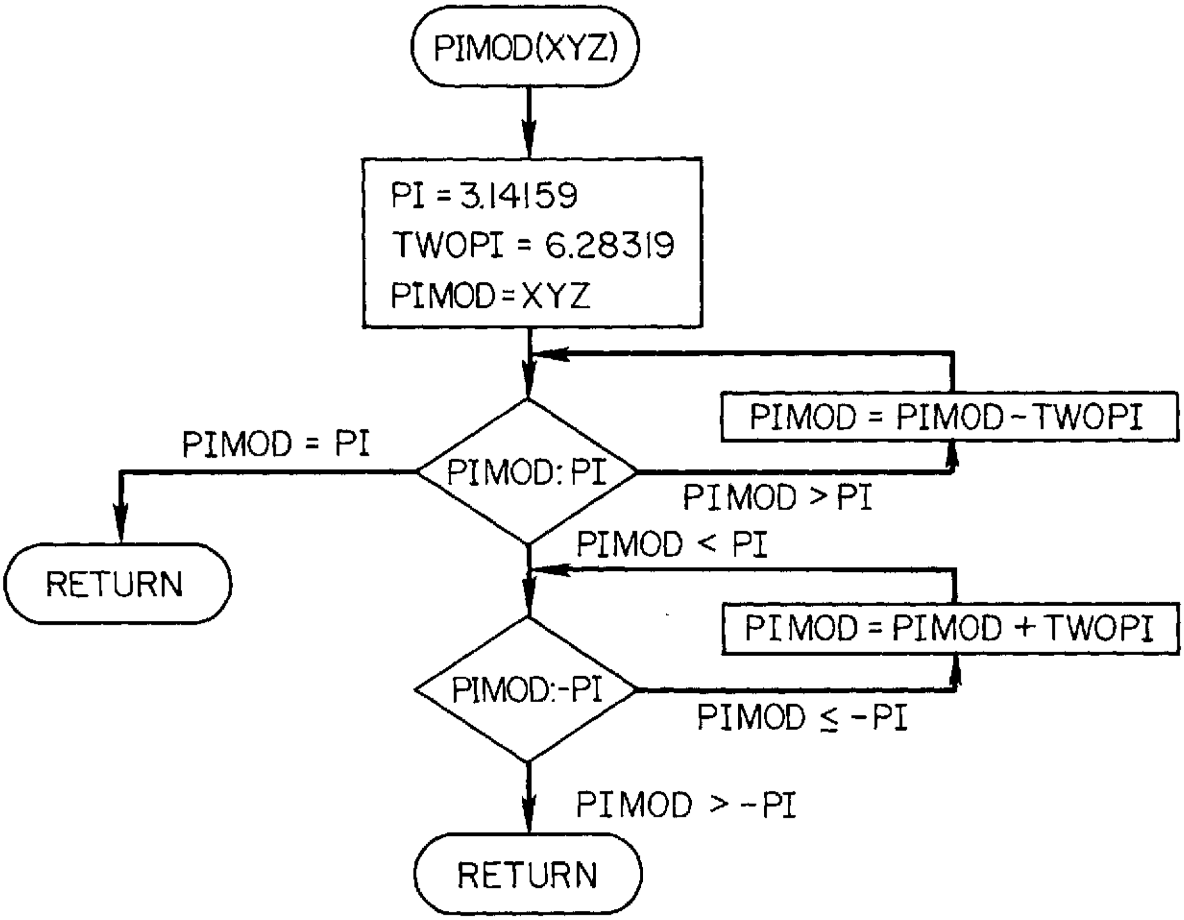
Temporary variables: None

Constants:

PI	=3.14159
TWOPI	=6.28319

Subroutines required: None

Called by: THREED, NEWPSI, and TST


PIMOD(XYZ)
PI = 3.14159
TWOPI = 6.28319
PIMOD = XYZ
PIMOD = PIMOD - TWOPI
PIMOD = PI
PIMOD: PI
PIMOD > PI
RETURN
PIMOD < PI
PIMOD = PIMOD + TWOPI
PIMOD:-PI
PIMOD ≤ -PI
PIMOD > -PI
RETURN

```
      FUNCTION PIMOD(XYZ)
      PI=3.14159
      TWOPI=6.28319
      PIMOD=XYZ
   10 IF(PIMOD.EQ.PI) RETURN
      IF(PIMOD.LT.PI) GO TO 20
         PIMOD=PIMOD-TWOPI
         GO TO 10
   20 IF(PIMOD.GT.-PI) RETURN
         PIMOD=PIMOD+TWOPI
         GO TO 20
      END
```

SGN(XYZ)

Required function:

This function generates the sign of its argument.

Method:

If $XYZ \geq 0$, the value of the function is set to $+1$; if $XYZ < 0$, it is set to -1.

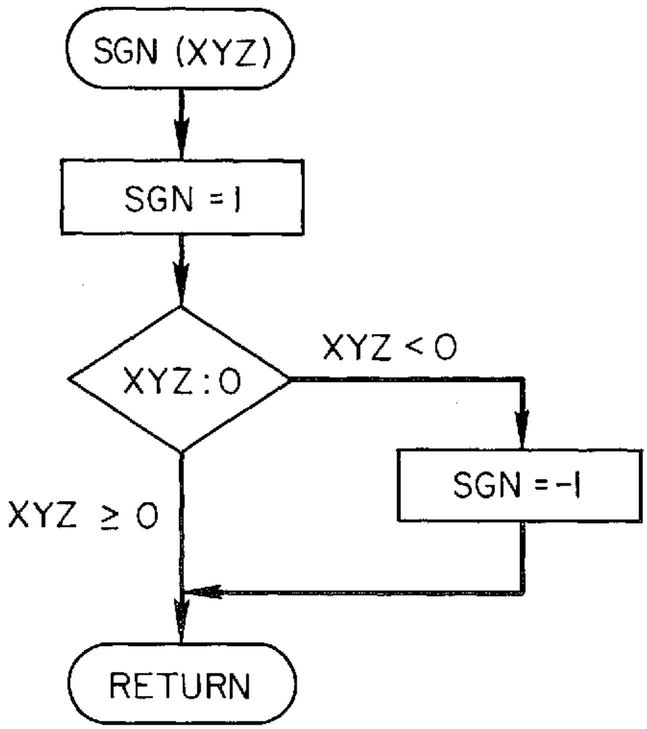
Input data (argument):

XYZ	any number

Output data (value of function):

$$SGN(XYZ) = \begin{cases} +1 & \text{if } XYZ \geq 0 \\ -1 & \text{if } XYZ < 0 \end{cases}$$

Temporary variables:	None
Constants:	None
Subroutines required:	None
Called by:	ROUND, NEWPSI, SPEED, TTTURN, and TST


SGN (XYZ)
SGN = 1
XYZ : 0
XYZ < 0
SGN = -1
XYZ ≥ 0
RETURN

```
FUNCTION SGN(XYZ)
SGN=1.
IF(XYZ.LT.0.) SGN=-1.
RETURN
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS OF POOR QUALITY

CALL FOURD

```
MAXPHI  =  30.0
MAXGAM  =  15.0
MINGAM  =  -7.5
MAXACC  =   1.0
MAXDEC  =  -1.0
PHIDOT  =   5.0
VSMF  = 150.0
VSFF  = 120.0
CUPPER  =   1.7
CLOWER  =   1.3
```

NWP = 6   HFINAL = 0.0   GFINAL = 0.0   VFINAL = 135.0   HW = 180.0   VW = 0.0

INDEX(I)	XWP(I)	YWP(I)	ZWP(I)	R(I)
1	7500.0	8000.0	-3240.0	0.0
0	23000.0	8000.0	-3240.0	4000.0
0	23000.0	-8500.0	-3240.0	4000.0
0	-17000.0	-8500.0	-3240.0	0.0
1	-17500.0	0.0	-1800.0	4250.0
1	-8000.0	0.0	-800.0	0.0

H(I)	GAM(I)	D(I)	XP(I)	YP(I)	ZP(I)	TURN(I)	R(I)	DELD(I)	XO(I)	YO(I)	ZO(I)
0.00	0.00	11500.0	19000.0	8000.0	-3240.0	-90.0	4000.0	6283.2	23000.0	4000.0	-3240.0
-90.00	0.00	8500.0	23000.0	-4500.0	-3240.0	-90.0	4000.0	6283.2	19000.0	-8500.0	-3240.0
180.00	0.00	36000.0	-17000.0	-8500.0	-3240.0	0.0	2057.9	0.0	-17000.0	-8500.0	-3240.0
180.00	-5.96	500.0	-17500.0	-8500.0	-3188.0	-180.0	4250.0	13351.8	-17500.0	0.0	-1800.0
-0.00	-6.03	9500.0	-8000.0	-0.0	-800.0	-0.0	981.2	0.0	-8000.0	0.0	-800.0

[illegible](I)	[illegible](I)
195.0	255.0

195.0	255.0
195.0	255.0
194.0	194.0
192.0	192.0
135.0	135.0

TMIN(I)	TMAX(I)
406.3	483.0
336.5	391.8
278.5	316.0
130.1	130.5
57.9	58.3
0.0	0.0

VA(I)	ENRTIM(I)	CRCD(I)
255.0	406.3	5
255.0	336.5	7
255.0	278.5	9
194.0	130.1	11
192.0	57.9	14
135.0	-0.0	16

TIMREF(K)	TRNREF(K)	GAMREF(K)	ACCREF(K)
45.1	0.0	0.0	0.0
24.6	-4000.0	0.0	0.0
33.3	0.0	0.0	0.0
24.6	-4000.0	0.0	0.0
87.5	0.0	0.0	0.0
61.0	0.0	0.0	-1.0
0.6	0.0	-6.0	0.0
2.0	0.0	-6.0	-1.0
69.5	-4250.0	-6.0	0.0
0.9	0.0	-6.0	0.0
57.0	0.0	-6.0	-1.0

TLEADT(I)	TLEADS(I)	TLEADP(I)
2.7	2.7	0.0
2.7	2.7	0.0
3.0	3.0	4.5
1.5	1.5	0.1
3.0	3.0	3.2

XAC	YAC	ZAC	HAC	VAC	CURWPT
-5000.0	15000.0	-2000.0	0.0	275.0	1

H	GAM	D	XP	YP	ZP	TURN	R	DELD	XQ	YQ	ZQ
0.0	4.9	0.0	-5000.0	15000.0	-2000.0	-34.6	4071.4	2460.0	-2687.0	14279.2	-2209.8
-34.6	4.9	9961.9	5511.2	8619.8	-3059.6	34.6	3500.8	2115.2	7500.0	8000.0	-3240.0

TIMREF(K)	TRNREF(K)	GAMREF(K)	ACCREF(K)
8.9	-4071.4	4.9	0.0
17.0	0.0	4.9	0.0
20.0	0.0	4.9	-1.0
8.3	3500.8	4.9	0.0

TLEADT(I)	TLEADS(I)	TLEADP(I)
3.0	3.0	4.8

REFERENCES

1. McLean, J. D.; and Tobias, L.: An interactive real time simulation for scheduling and monitoring of STOL aircraft in the terminal area, Journal of Aircraft, vol. 10, no. 6, June 1973, pp. 372-378.

2. Hynes, R. J.; Stevenson, L. E.; and Capen, E. B.: 4-D Guidance of STOL Aircraft, AIAA Paper No. 71-770, July 12-14, 1971.

3. Miele, A.: Flight Mechanics, vol. 1 - Theory of Flight Paths. Addison-Wesley (Reading, Mass.), 1962.

4. Lee, H. Q.; and McLean, J. D.: Guidance Technique for Automated Air Traffic Control, AIAA Paper No. 72-121, Jan. 17-19, 1972.

5. Lee, H. Q.; Neuman, F.; and Hardy, G.: 4D Area Navigation Operating System Description and Flight Test Results. NASA TN D-7874, 1975.

6. Erzberger, H.; and Lee, H. Q.: Terminal-Area Guidance Algorithms for Automated Air-Traffic Control. NASA TN D-6773, Apr. 1972.

7. Duning, K. E.; Hemesath, N. B.; Hickok, C. W.; Lammer, D. G.; and Goemaat, M. L.: Curved Approach Path Study. Final Report, Collins Radio Company, Cedar Rapids, Iowa, Oct. 1972.

8. Erzberger, H.; and Pecsvaradi, T.: 4-D Guidance System Design With Application to STOL Air Traffic Control. Stanford University, Stanford, Calif., Aug. 16-18, 1972, Joint Automatic Control Conference, 13th, 1972.

9. Pecsvaradi, T.: Optimal Horizontal Guidance Law for Aircraft in the Terminal Area. IEEE Trans. Automatic Control, vol. AC-16, no. 6, Dec. 1972, pp. 763-772.

10. Hansen, Q. M; Young, L. S.; Rouse, W. E.; and Osder, S. S.: Development of STOLAND, a Versatile Navigation, Guidance and Control System. AIAA Paper No. 72789, Aug. 7-9, 1972.

NASA-Langley, 1975 A-5530