NASA Technical Memoradum 81334

(NASA-TM-81334) ALGORITHM FOR FUEL CONSERVATIVE HORIZONTAL CAPTURE TRAJECTORIES (NASA) 31 p HC A03/MF A01 CSCL 01B

N82-11018

Unclas
G3/01 08309

Algorithm for Fuel Conservative Horizontal Capture Trajectories

Frank Neuman and Heinz Erzberger

October 1981

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 81334

Algorithm for Fuel Conservative Horizontal Capture Trajectories

Frank Neuman
Heinz Erzberger, Ames Research Center, Moffett Field, California

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ALGORITHM FOR FUEL CONSERVATIVE HORIZONTAL CAPTURE TRAJECTORIES

Frank Neuman
and
Heinz Erzberger

SUMMARY

This paper describes a real-time algorithm for computing constant altitude fuel-conservative approach trajectories for aircraft. It is a refinement in the horizontal plane of the fuel conservative capture algorithm previously flight tested at the Ames Research Center. The trajectories may start at any initial position, heading and speed and end at any other final position heading and speed. They are geometrically simple, as they consist only of straight lines and a series of circular arcs of varying radius. Throttle control is also simple as it consists of maximum thrust in the acceleration segments of flight, nominal thrust to balance drag in the constant speed segments, and zero thrust in the deceleration segments. The bank-angle control is either zero or approximately 30°. The algorithm is designed to approximate constant bank-angle decelerating turns by combining a series of short circular arcs of different radii into a single turn. The bank angle varies slightly due to flying the circular arc segments with varying speed.

To study possible further fuel reduction, a more sophisticated algorithm, which is suited for long-range flightpaths but not short-range paths, was developed that uses constant bank-angle turns (rather than a sequence of circular arcs to approximate a constant bank angle). The more sophisticated algorithm is too slow for onboard implementation and was only investigated to serve as a basis of comparison with the real-time algorithm.

The characteristics of the trajectory computed by the two algorithms were chosen to approximate the extremal trajectories obtained from the optimal control solution to the problem. The fuel consumption of the trajectories computed with the two algorithms was compared with the corresponding of the extremal. For the most frequently occuring trajectories in the terminal area the comparison with the extremals showed a fuel difference of only 0.5 - 2% for the real-time algorithm in favor of the extremals. Zero-thrust extremals were the most difficult to approximate by the algorithm and the synthesized trajectories took up to several hundred pounds more fuel than the corresponding extremal.

INTRODUCTION

The escalating costs of fuel have increasingly focused attention on the problem of minimizing fuel consumption of commercial aircraft. Recent work in aircraft guidance has demonstrated that onboard optimization of aircraft trajectories offers an efficient method for fuel conservation.

Aircraft trajectory optimization problems can be divided into two classes, namely enroute problems with trajectory lengths of 50 n.m. and longer, and terminal area problems with lengths of from three to 50 n.m. For the enroute problem, an onboard algorithm for fuel/cost optimum climb-cruise-descent has been developed by application of optimal control theory (ref. 1). The solution was developed for the vertical plane only, since horizontal maneuvers are not a significant feature of enroute flight.

Terminal area trajectory problems are more difficult to solve since vertical and horizontal maneuvers involving speed, altitude and heading changes occur simultaneously, and are of comparable significance in influencing fuel consumption. The implementation of the exact solution to the optimal control problem is too complicated for real-time mechanization in a flight computer. As a result it is necessary to develop approximate solutions to the problem. An algorithm for onboard calculation of fuel conservative terminal area trajectories was recently described by Erzberger and McLean (ref. (2)). Their algorithm computes fuel conservative approach trajectories, which they refer to as capture trajectories, starting from an arbitrary initial position, heading airspeed and altitude and ending at a specified final position, heading, airspeed and altitude. The final position is chosen as close to the touchdown point as possible.

The algorithm of Erzberger and McLean separates the problem into two parts in order to achieve computational simplicity: first, the synthesis of a horizontal trajectory consisting of constant radius turns (up to three) and a straight line; and second, the synthesis of fuel conservative airspeed and altitude profiles along the previously computed horizontal trajectory. The synthesized horizontal trajectory minimizes the distance to fly.

It is known from several studies of fuel and time optimum trajectories (ref.(3), (4), and (5)) that horizontal maneuvers flown during simultaneous accelerations or decelerations often require turns at maximum bank angle rather than turns at constant radius and variable bank angle as in the algorithm of ref. (2). Constant radius turns simplify the computational problem. The present study improves the fuel efficiency of the algorithm in ref. (2) for certain types of capture trajectories by generating the final turn of the capture trajectory at near constant bank angle. The non-circular turns are computed by a real-time algorithm which uses the algorithm of ref.(2)in an iterative loop. In addition, significant characteristics of the optimum airspeed profile during turns, determined in ref. (5), are incorporated in the synthesis of the airspeed profile. These refinements increase the computational complexity of the capture algorithm to a modest degree.

In addition, a more sophisticated algorithm is described, which provides a much closer match to the optimal solution for long-range flightpaths. Although this algorithm is too time consuming for implementation in an onboard computer and is not capable of handling short-range flightpaths, it is thought to represent the limit to which an on-board algorithm may be improved for the long-range problem.

Finally, the fuel consumption used in flying trajectories computed by the real-time algorithm and the more sophisticated algorithm are compared with the fuel consumption that would occur if the aircraft were to fly the trajectories defined by the extremal solutions to the optimal control problem.

In this initial study the effects of altitude and winds on the trajectories were ignored. The impact of these assumptions on the algorithm is currently under study.

REAL-TIME ALGORITHM FOR THE CONSTRUCTION OF FUEL CONSERVATIVE CAPTURE FLIGHTPATHS

Overview

Two major steps are involved in deriving the real-time algorithm. First it is necessary to identify and briefly review the salient features of certain extremal fuel optimal trajectories determined in ref. (5). Then it is necessary to approximate these characteristics for use in the algorithm. Since relatively long approach flightpaths (over 10 miles) are the rule, rather than the exception, the characteristics of long flightpaths generated, as in ref. (5), were examined. Such paths contain a beginning and a final turn, connected by a line segment of extremely small curvature, which is almost a straight-line. The speed profile for such a path is as follows: To achieve a tighter initial turn there is often a minor speed reduction followed by an acceleration, which continues in the almost straight-line segment. In the straight-line segment the speed of minimum fuel per unit distance is approached but never achieved. Deceleration at zero thrust toward the landing speed usually takes place in the final turn, a considerable portion of which is flown at the maximum bank angle limit. If the final turn is large the deceleration is often followed by a brief acceleration.

It is shown in ref. (5) that the controls, the bank angle and the thrust, are continuous in time. To generate and store such control commands would take a large amount of computer time and storage. Therefore, in the real-time algorithm developed here, we shall be satisfied with step commands in the controls. Throttle control will consist of maximum thrust in the acceleration segments of flight, thrust to equal drag in the constant speed segments, and zero thrust in the deceleration segments. Also, we shall be satisfied in this algorithm with flightpaths that are constructed of a combination of straight-lines and sequential circular arcs, which approximate constant bank-angle turns. For simplicity, acceleration segments will occur only on the straight-line segment of the path. That means that the initial turn is a constant radius, constant speed turn. The bank angle at the start of each segment will be either zero or 30°, which is the maximum allowable bank angle for a commercial jet liner. This bank angle gives the tightest possible turn for the given speed; indeed in the optimal trajectories (ref. (5)) the turns quite frequently do occur at the bank-angle limit. The speed profile and the shape of the flightpath are interconnected, since the turn radii depend on the speed of the turns.

Flightpath Development

Turning Segments. - To construct a flyable flightpath which includes a sequence of circular arcs to approximate a constant bank-angle turn, the equations for the circular arc segments of the flightpath are first developed. The use of these segments to approximate the constant bank-angle turn is then discussed.

During constant radius decelerating turns the bank angle, ϕ, is related to the instantaneous velocity by

$$R = v^2/(g \tan\phi) \qquad (1)$$

where R is the radius of turn. R is chosen so that, the maximum bank angle of 30° occurs at the maximum speed v_1, From Eq.(1),(7) and (8) of Appendix I with T = 0 we have

$$dv/dt = -(k_3 v^2 + k_4/v^2) \qquad (2)$$

where k_3 and k_4 are

$$k_3 = g(k_1 + k_2/(g^2R^2)/W \qquad (3)$$

$$k_4 = g\ k_2/W \qquad (4)$$

Eq (2) includes the effect of the varying bank angle while decelerating on a constant radius turn.

To calculate the end velocity v_2 from the start velocity v_1 on a circular arc of angle 0 or vice versa, dt in Eq.(2) is replaced by its equivalent ds/v. Integration gives

$$v_2 = ((v_1 + v_m)/e^{4K_3 R\theta} - v_m)^{1/4} \qquad (5)$$

where v_m is the minimum drag velocity

$$v_m = (k_4/k_3)^{1/4} \qquad (6)$$

In specific applications the aircraft model may be more complex, in which case numerical integration of the equations will be called for as in the algorithm of ref. (2).

When the final speed and the angle of the circular arc, 0, are given, and the radius of turn and initial speed are the unknowns (ref.(5)) has to be solved by successive approximations, since the unknown v_1 cannot be isolated.

The constant bank-angle turn is approximated by a sequence of circular arcs of angle 0 with varying radii. If 0 is too large the aircraft will fly a decelerating turn that is less tight than it theoretically could, and the resulting flightpath would normally be less fuel efficient. In addition, upon transfer from one circular arc to the next a large transient in bank angle would occur. On the other hand, if we choose the angle θ, too small, computation time and computer storage increases in the airborne computer. As a compromise a $\theta = 30^{\circ}$ segment was chosen. Usually, the final speed and heading were chosen to be 180 knots and 0 degrees respectively. Starting from this end condition, by using the relationships of equations (1) and (5), the decelerating spiral turn can be constructed as shown in figure 1.

The Horizontal Trajectory. - The construction of the horizontal trajectory is carried out with the help of the horizontal capture trajectory algorithm described in ref. (2) and (6). This algorithm was originally designed to provide minimum time, (and near minimum fuel), constant speed trajectories. It will construct a trajectory from a given initial position and heading to a final position and heading by means of a turn along a circular arc followed by a straight line (or, less frequently, by a second circular arc) and a subsequent circular turn. Depending on the end conditions, the algorithm delivers between a minimum of two to a maximum of six different paths, of which the shortest path is chosen unless a specific type of flightpath is specified (e.g. the final turn must be a right turn). As input, the algorithm must be given the starting and the final turn radius, and the user must make sure that the desired speed profile can be accommodated by this flightpath. As shall be shown, this algorithm can be used sequentially to generate the approximate

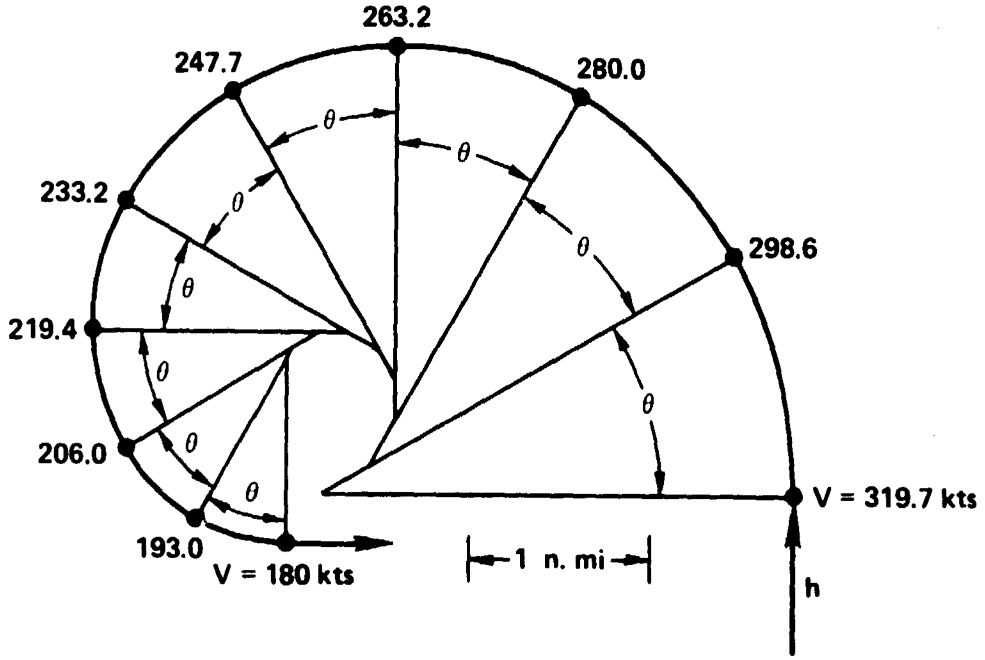


Figure 1. - Construction of a Decelerating Turn Flight Path were waypoint h_2 defined by position (x_h, y_h), heading ψ , nominal speed v_n at the waypoint, radius of turn R_n, nominal control settings T_n, ϕ_n.

constant bank-angle deceleration discussed in the previous section.

For simplicity, the initial turn radius is kept constant. Assume that the starting and final position, heading, and speed are given as shown in figure 2a. For the moment also assume that deceleration is required in the entire last turn. From Eq. (5) we solve for v_{n-1} the initial velocity of the nth segment where $v_n = v_f$, and using v_1 and v_{n-1}, we solve for R_1 and R_{n-1} from Eq. (1). Application of the algorithm (ref. (6)) generates the path of figure 2b. This path could not be flown, however, since the turn from waypoint n - 1 to waypoint 3 is tighter than the speed permits. So we use the conditions at point (n-1) as the final position speed and heading and repeat the above process. This results in the flightpath of figure 2c. We repeat the process until the final circular segment is 30^o, as shown in figure 3d. The resulting speed profile is shown as solid line in figure 2e where the deceleration starts in the straight-line segment. Additional fuel savings can be had if changes in speed are allowed such as is shown in a dashed line in figure 2e. This is further discussed in the next section.

Occasionally, for short flightpaths, application of the algorithm in ref. (6) to approximate a constant bank-angle turn in the manner described above may not result in the circular segments all going in the same direction. Instead the algorithm may determine that a turn to point n-i in the opposite direction is more optimal. Because of this, the real-time algorithm developed here was modified to compute both the path and the fuel that would be used if the final turn were to be constrained to a single direction and to compute the path and the fuel that would be used without this constraint. The path using the least fuel is then selected.

In general, the final deceleration in the flightpath may begin in the initial turn, the straight-line segment or the final turn. The algorithm, which is described in greater detail in the appendix, provides for these different cases. If the flightpath is too short to accommodate the required speed change, the flightpath is rejected and the next longer flightpath is chosen from the available set.

Speed Profile for the Straight-Line Segment. - The flight path developed in the last section has a straight constant speed segment. If the speed is allowed to exceed the initial speed v_1 (e.g., 250 knots for the terminal area) further fuel reduction may be obtained by varying the speed in the straight- line segment. For a design criterion we turn to the minimum fuel extremals for straight-line flight, an example of which is shown as solid line in figure 3. The constant speed which is approached but never reached is the speed for minimum fuel per unit distance v_c*, and the thrust is continuous but is zero in the latter part of the deceleration segment.

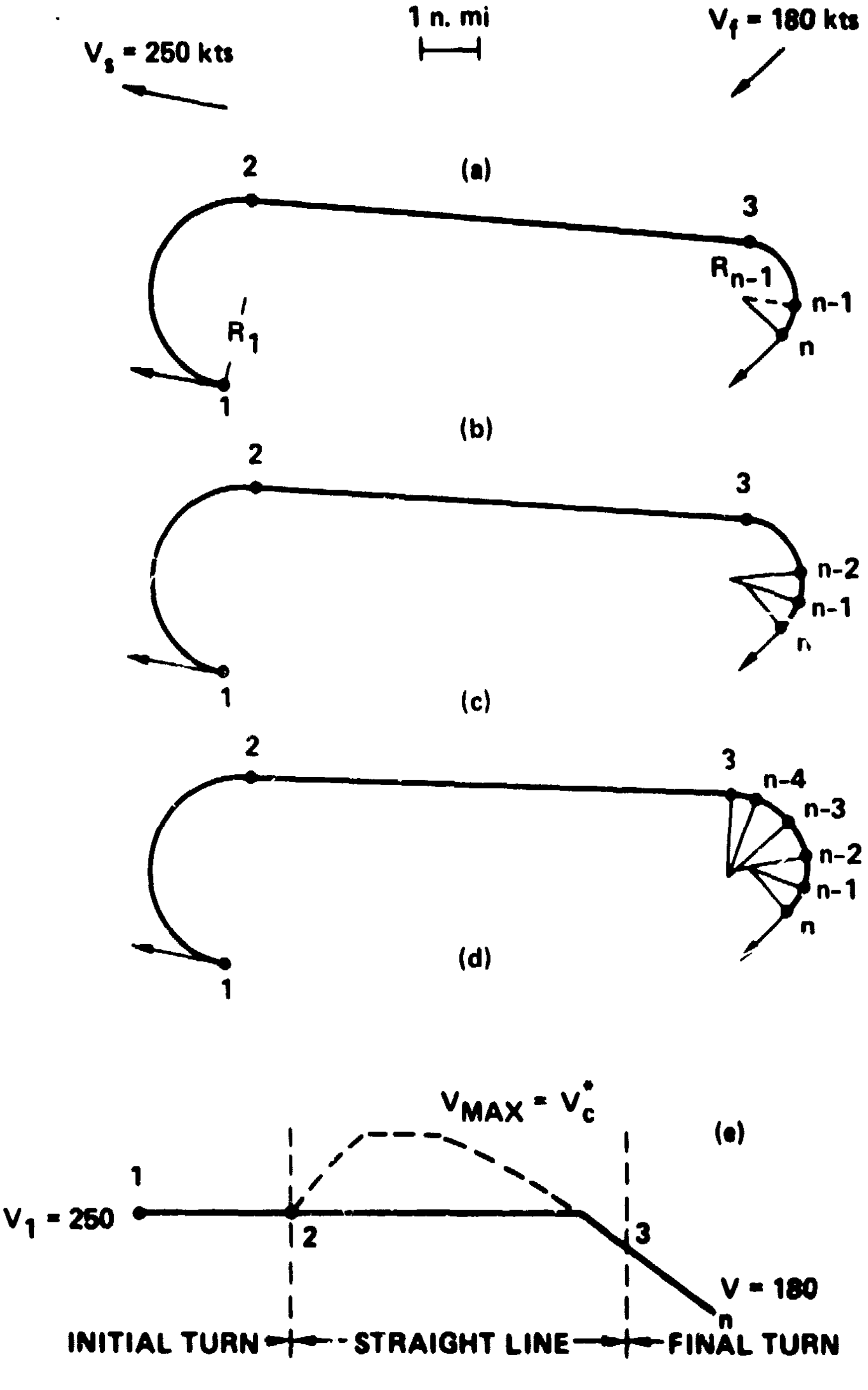


Figure 2. - Construction of a Fuel Conservative Capture Flightpath .

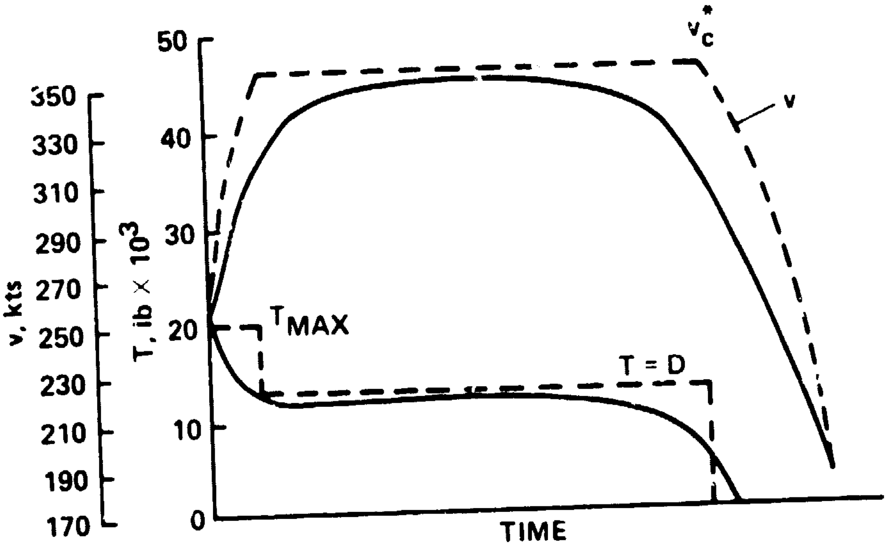


Figure 3. - Optimal and Suboptimal Speed and Thrust Profiles for a Straight Line Flightpath.

Equations (2) and (5) apply to deceleration on a straight-line path by setting the second term in Eq. (3) to zero since $R = \infty$. We approximate the optimal profiles by the dashed line of figure 3. This approximation is, in fact, the optimal solution when $c_2 = 0$ in Eq. (9) of Appendix I. It is identical to the one used in ref. (2). Such a solution is simple to implement, and, as will be shown in the results section, comes close to the true optimal solution.

A MORE SOPHISTCATED ALGORITHM FOR LONG RANGE FLIGHTPATHS.

The previously described flightpaths consist of only straight-line and circular elements. Therefore, they are completely definable by a series of waypoints. Also, an airborne digital guidance system exists at the Ames Research Center which provides guidance signals along such paths. Hence the guidance problems for such paths have already been solved completely and selected flightpaths could be flight-tested quickly if desired. In contrast, the turning segments of a flightpath with true constant bank-angle turns cannot be represented in such a manner. New equations for a cockpit path display to the pilot and path error nulling guidance equations would have to be developed. Still it is of interest to determine how much improvement a constant bank-angle turn can provide when coupled with a speed profile that more closely simulates that of the optimal solution. It is to be noted that, to keep programming simple, this algorithm was intended for long flightpaths only, (with a straight segment of at least 10 nautical miles).

The speed profile chosen is that given in figure 4. If the initial turn is greater than 100° then we decelerate initially without going below v_{min}. Until the turn remaining is 100°. We then accelerate toward v_c*. If the initial turn is less than 100° we immediately accelerate. The final turn, which is generated in the backward direction, the angle of the decelerating portion of the turn, ψ_D , is a function of the final speed according to

$$\psi_D = 6.732 - 0.01179\ v_f \qquad (7)$$

Where ψ_D is in radians and v_f in ft/sec.

This expression was obtained as an approximation of certain extremal trajectory characteristics. This makes $\psi_D = 100$ degrees for a final speed, v_f, of 250 knots, and $\psi_D = 180$ degrees for a final speed of 180 knots. The remaining turn is flown at constant speed or maximum acceleration so that v_f is met at the end point. For final turns less than the above numbers there occurs deceleration only.

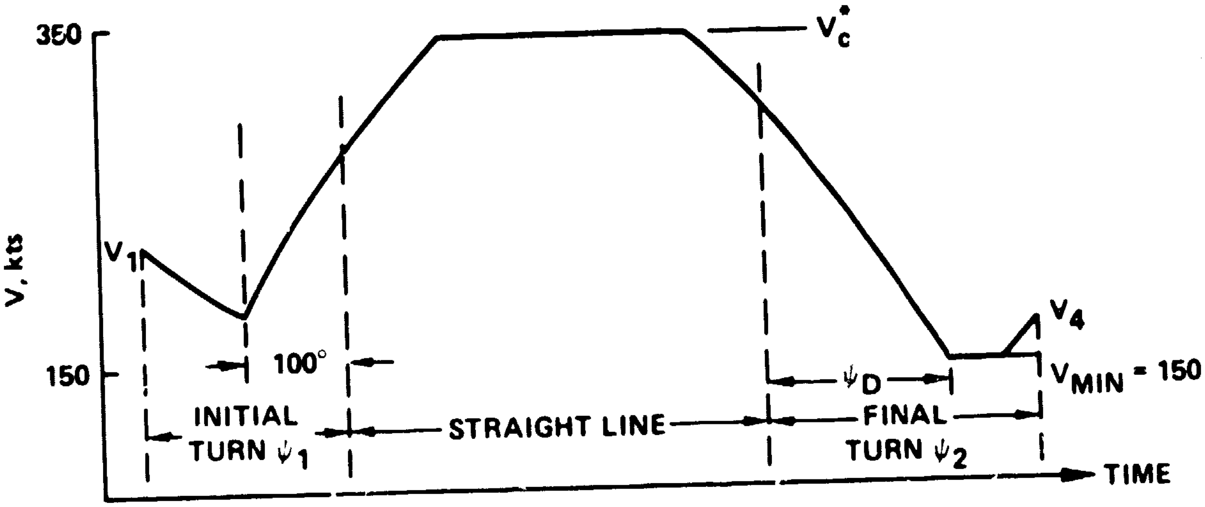


Figure 4. - Speed Profile for Continous Turn Algorithm.

Having defined the speed profile along the path, we turn to the path construction. Again the capture algorithm of ref. (6) is called to determine turn directions and approximate subtended angles of the initial and final turns ψ_1 and ψ_2 of figure 4. It must be remembered that the algorithm of ref. (6) delivers circular turns, while the desired turns are constant bank-angle turns. We select the larger of the two turns ψ_1 or ψ_2 to calculate the path of a true constant bank-angle turn by numerical integration of the equations of motion either in the forward or backward direction while following the speed profile. Since the ψ's from the algorithm in ref. (6) are only approximate, a path corresponding to only one-half of the indicated turn angle is calculated. This process results in a new end point for the remaining path to be developed. Then the algorithm of ref. (6) is called again from the new end point. The above process is repeated until the remaining initial and final turns are less than one degree. At this time the less than one degree circular turns determined by the algorithm in ref. (6) are flown. The development of the speed profile for the remaining straight-line path is identical with that of the algorithm. In addition to the technical difficulties of this algorithm described in the introductory paragraph we note that it is also computationally lengthier. It is therefore not suggested for onboard mechanization but is presented here as a limit that may be approached by the algorithm described earlier if one would include the more complex speed profile and make the circular turn segment smaller. (e.g. 15° instead of 30°).

EVALUATION PROCEDURE

The real-time algorithm always constructs a flyable flightpath given any set of initial and end conditions. The more sophisticated algorithm constructs a path provided the end conditions are separated by 10 miles or more. In contrast, when extremals are generated from the optimal control solution described in Kreindler, and Neuman (ref. 5) only the end or initial condition can be specified in advance, not both. Therefore many extremals had to be generated to obtain a set of flightpaths which were judged to be reasonable test cases for terminal area approaches. Further, the extremals were divided into two classes. Relatively long, gently turning flightpaths such as would normally be expected on approaches to an airport, and short tightly turning flightpaths, that may represent emergency maneuvers. The fuel used on the extremal was compared with the fuel used on the equivalent path generated by the two algorithms. These comparisons are not intended as a complete statistical analysis, which would be very complex and is not warranted at this time.

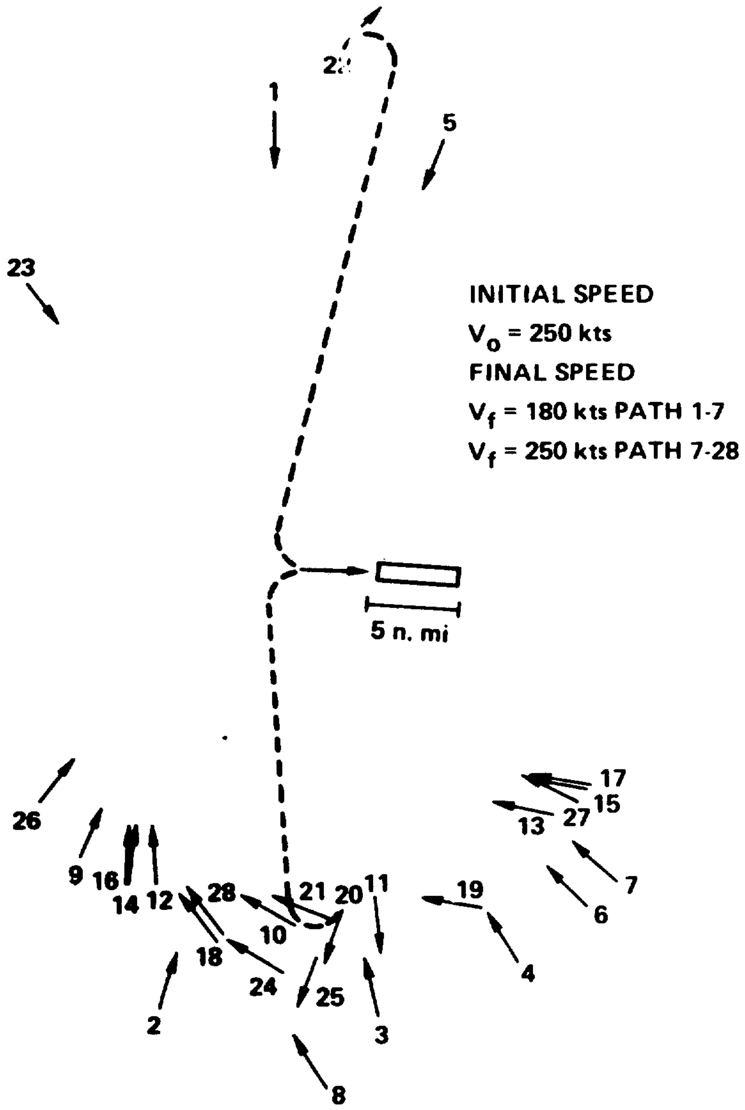
At present, there is no general method to prove the (global) optimality of a given extremal. Therefore, some of the extremals that were picked for comparison were shown to be not globally optimal by constructing a more fuel efficient path via the algorithm in conjunction with a "tree search". A more complete discussion of this problem is covered in ref.(5).

RESULTS AND DISCUSSION

Evaluation of the Algorithm. - Figure 5 shows the initial states for 28 relatively long approach paths that resulted from the computation of extremals (see ref.(5)). Two flightpaths are drawn in for illustration. Also shown are comparisons of the fuel used when flying an extremal trajectory with the fuel used when flying the trajectory generated by the real-time algorithm and the more spohisticated algorithm. The real-time algorithm provided paths that used on the average only 1.53 percent or 10.2 lbs more fuel than the optimal solution. The more sophisticated algorithm further decreased this to an average of 0.57 percent or 3.7 lbs. For long, speed-constrained flightpaths (v_{max} = 250 knots) (not shown) the suboptimal paths perform even better in comparison with the extremals, because the extremals have a long, almost straight-line segment with constant speed $v(t) = v_{max}$, which, of course, is easy to approximate.

To get an idea of the effectiveness of the real-time algorithm during the straight-line segment, we study long straight paths with an initial speed of 250 knots and a final speed of 180 knots. Comparing two possible suboptimal speed profiles a flight with a constant speed of 250 knots followed by a final deceleration to 180 knots, with a flight that initially accelerates toward v_c^* (the optimum cruise speed) we find that the latter is 0.9 percent (or 1.6 lbs) better for an 8 nautical mile path and 5.18 percent (or 21.8 lbs) better for a 16 nautical mile path. Hence, the initial acceleration pays off if the path is sufficiently long. When the suboptimal paths which contain the initial acceleration are compared with the extremals we find that they are 1.19 percent (or 5.61 lbs) off the optimal for a 12 nautical mile path and only 0.18 percent (or 3.4 lbs) off the optimal for a 71 nautical mile flightpath. This agrees with our general observation: the longer the flightpath, the closer the real time algorithm solution will compare with the optimal flightpath.

Figure 6 shows the end states and fuel consumption for 33 short approach paths. Three extremals are drawn as dashed lines for illustration. The initial states were obtained by backward integration of extremals from a final state. The data of figure 6 are ordered with respect to the initial speeds, and again, within each category, with respect to the fuel used on the flightpath. The flightpaths that are marked as coasting trajectories had zero thrust for the entire flight. Note that some fuel is consumed in these cases, because the engines are not shut off even though thrust is

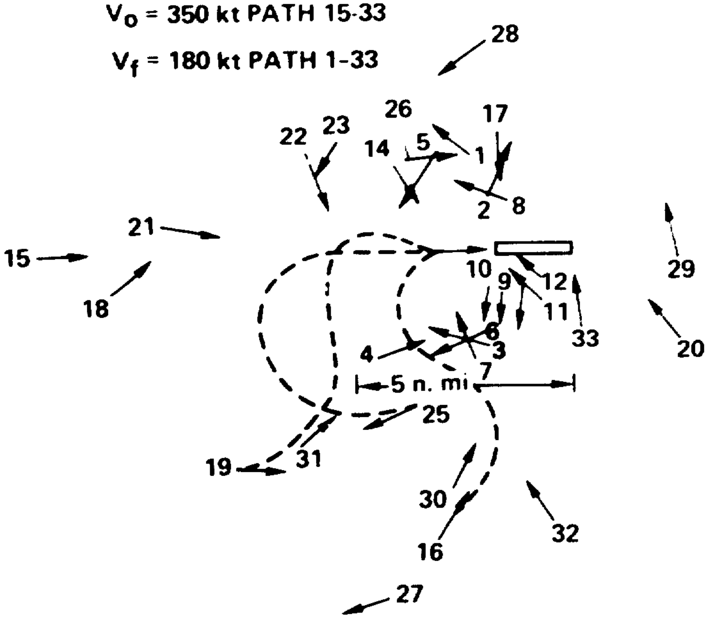


PATH =	FUEL USED ON EXTREMAL lbs	INCREASE IN FUEL FOR REAL TIME ALGORITHM		INCREASE IN FUEL FOR MORE SOPHISTICATED ALGORITHM	
		%	lbs	%	lbs
1	668.4	0.6	4.0	0.43	2.9
2	668.8	0.5	3.3	0.43	2.9
3	667.5	0.8	5.3	0.55	3.7
4	666.6	0.8	5.3	0.44	2.9
5	667.2	0.7	4.7	0.44	2.9
6	666.0	0.9	6.0	0.44	2.9
7	665.7	1.0	6.6	0.45	3.0
8	805.7	1.6	12.9	0.46	3.7
9	586.1	0.9	5.3	0.59	3.5
10	595.9	1.8	10.7	0.57	3.4
11	713.5	2.6	18.6	0.75	5.4
12	786.8	1.2	9.4	0.60	4.7
13	653.8	2.3	15.0	0.76	5.0
14	576.6	1.2	6.9	0.64	3.7
15	652.3	2.4	15.6	0.86	5.6
16	576.6	2.0	11.5	0.62	3.6
17	652.0	2.9	18.9	0.83	5.4
18	622.0	1.5	9.3	0.56	3.5
19	659.1	2.3	15.1	0.58	3.8
20	603.9	2.4	14.5	0.37	2.2
21	682.9	1.2	8.2	0.56	3.8
22	940.2	1.4	13.6	0.41	3.9
23	618.5	0.7	4.4	0.51	3.2
24	666.8	2.0	13.3	0.95	3.3
25	755.3	1.8	13.6	0.54	4.0
26	556.3	0.8	4.5	0.61	3.4
27	606.0	2.95	17.9	0.60	3.6
28	617.9	1.6	9.9	0.55	3.4
			$\bar{x}$ = 10.15		3.7
		$\bar{x}$ = 1.53%		$\bar{x}$ = 0.57%	

Figure 5. - Initial Conditions and Test Results for 28 Long Approaches.

INITIAL AND FINAL SPEEDS

V_o = 180 kt PATH 1-2
V_o = 250 kt PATH 3-14
V_o = 350 kt PATH 15-33
V_f = 180 kt PATH 1-33



PATH =	FUEL USED ON EXTREMAL		INCREASE IN FUEL			
			REAL TIME ALGORITHM		TREE SEARCH	
			%	lbs	%	lbs
1	178.0		10.45	18.6	10.45	18.6
2	233.8		8.77	20.5	8.77	20.5
3	44.5	COASTING	391.01	174.2	146.07	64.9
4	44.6		630.04	280.9	561.88	250.6
5	44.9		683.52	306.9	302.23	135.6
6	76.0		158.82	121.3	124.49	95.0
7	94.3		197.56	86.3	41.04	38.7
8	100.7		120.06	120.9	31.98	32.2
9	121.4		75.62	91.2	10.63	12.9
10	130.0		69.54	90.4	10.31	13.4
11	142.1		64.32	91.4	22.45	31.9
12	169.0		28.11	47.5	22.43	37.9
13	220.3		31.32	69.0	-5.36*	-11.8
14	278.7		8.43	23.5	-2.91*	-8.1
15	105.3	COASTING	238.37	251.0	2.69	8.1
16	105.3		155.08	163.3	15.00	15.8
17	105.3		312.73	329.3	17.09	18.0
18	105.3		292.02	307.5	19.56	20.6
19	105.3		210.35	221.5	40.59	42.7
20	105.3		41.03	43.2	41.03	43.2
21	105.3		250.90	264.2	111.97	117.9
22	105.5		667.87	703.3	75.35	79.3
23	105.8		246.22	259.3	42.63	44.9
24	109.2		194.78	212.7	15.84	17.3
25	110.1		150.77	166.0	15.62	17.2
26	136.8		206.94	283.1	76.02	103.9
27	155.1		103.16	160.0	-25.15*	-30.0
28	182.4		294.13	536.5	-6.36*	-11.6
29	190.8		-38.42*	-73.3	-38.47*	-73.4
30	203.0		19.41	39.4	-0.84*	-1.7
31	229.8		6.31	14.5	-44.21*	-101.6
32	281.2		0.89	2.5	-9.21*	-25.9
33	289.9		-60.57*	-175.6	-68.39*	-198.3

* PROOF THAT EXTREMAL IS NON-OPTIMAL

Figure 6. - Short Turning Trajectories Initial Conditions and Test Results.

retarded to zero. (This is reflected by c_o in Eq. (9) of Appendix I). The others had speed reductions below the final speed and a buildup of thrust towards the end of the flight.

Figure 7 shows a horizontal track of some of the extremals numbered to correspond to the path identification number of column 1 in figure 6. Figure 7 is merely intended to give an appreciation of the complex patterns of turns of these extremals. Many of the highly turning extremals can be thought of as path streching maneuvers, to dissipate the kinetic energy while decelerating to the landing speed. It is clear that a simple algorithm that provides for two, or, at most, three turns cannot come close to generating flightpaths with similar fuel consumption.

The real-time algorithm often delivered paths that were shorter than the extremal, but they were too short to complete the deceleration that is required. Hence the next longer flightpath developed by the real-time algorithm was chosen, which then required substantially more fuel (see the column labeled "Real-Time Algorithm" in figure 6). While the real-time algorithm shows a fuel consumption up to several times that of the extremal, the algorithm always delivers a realizable flightpath. Also, we note that in two cases the algorithm delivered a path that was more fuel efficient than the extremal. This brings us to the necessity of an attempt to check extremals for global optimality before using them as test cases for a suboptimal high speed algorithm. This was done by a "tree search" which found seven additional non-optimal extremals. As previously mentioned, this problem is discussed in ref. 5.

SUMMARY AND CONCLUSIONS

The real-time algorithm always delivers a realizable flightpath. It is fuel conservative and near fuel optimum for most flightpaths occuring in the terminal area. However for the few short emergency maneuvers, which primarily consist of highly turning paths flown with zero thrust the algorithm will deliver a flightpath that will be far from fuel optimum.

The more sophisticated algorithm presents an upper limit to the performance for the onboard algorithm during long range flightpaths. This is, for long flightpaths, the algorithm may be improved from being within 1.5% of the optimal solution to 0.6% of the optimal solution.

The variable turn radius feature of the real-time algorithm saves fuel but adds computation time when compared to a previously developed algorithm. Before this algorithm can be incorporated in an onboard guidance system it must be expanded to include the effects of altitude, winds and navigation errors.

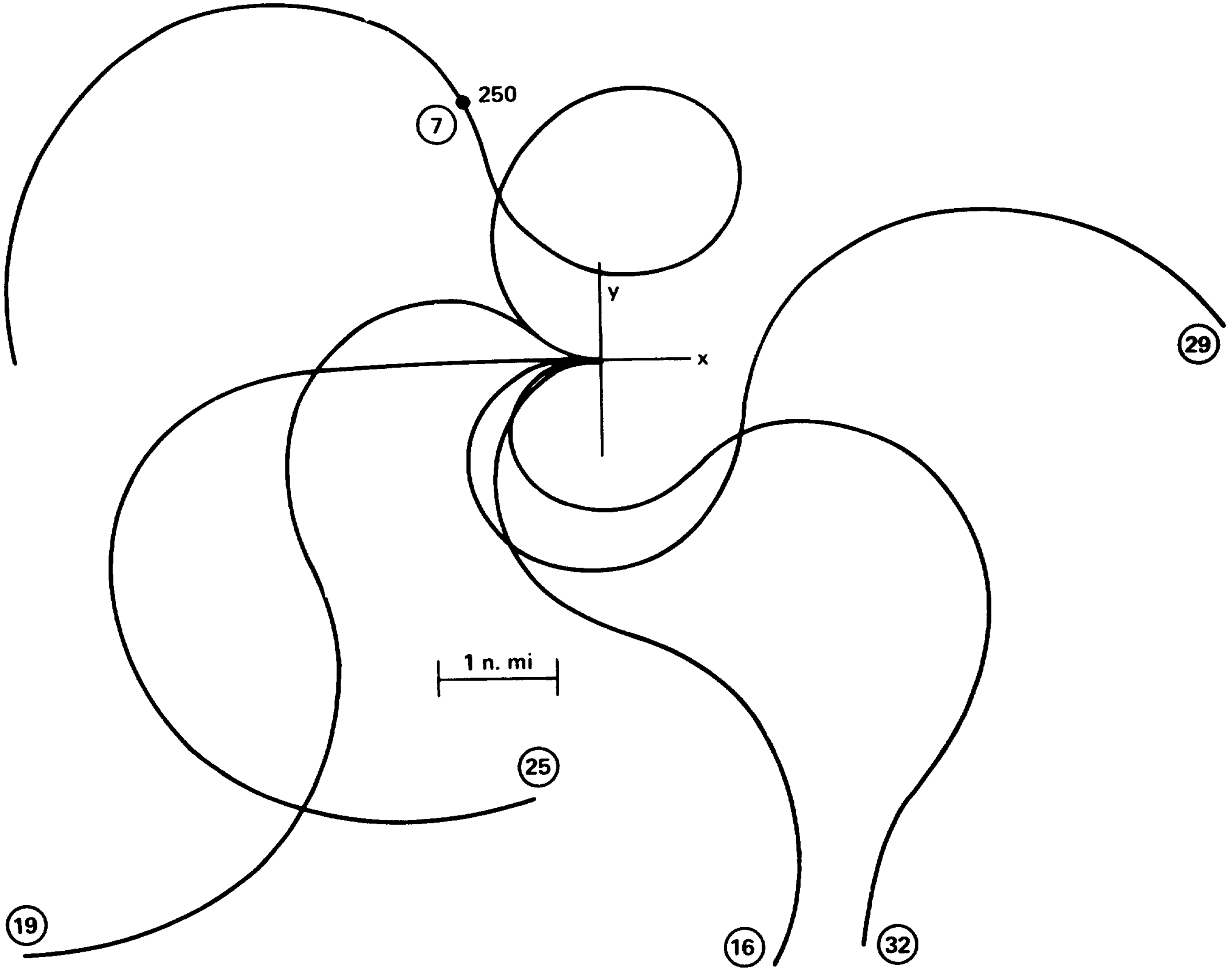


Figure 7. - Examples of Short Turning Extremals.

APPENDIX I

AIRCRAFT MODEL

The aircraft model used is an approximation of that of a Boeing 727 aircraft. Since it is identical with that in ref. (5), it is restated here without detailed development. The bank angle is limited to

$$- 30^{\circ} < \phi < 30^{\circ} \qquad (1)$$

The thrust T is limited to

$$0 \leq T \leq 30000 \text{ lbs} \qquad (2)$$

For horizontal flight vertical lift equals weight (fig. 8a)

$$L \cos \phi = W \qquad (3)$$

The point mass equations of motions are as follows:

The ground velocity components are (fig. 8b)

$$x = v \cos \psi \qquad \text{ft/sec} \qquad (4)$$

$$y = v \sin \psi \qquad \text{ft/sec} \qquad (5)$$

The heading rate is

$$x = -\frac{g}{V} \tan \phi \qquad \text{rad/sec} \qquad (6)$$

The acceleration along the velocity vector is

$$v = g \quad (T - D)/W \text{ ft/sec} \qquad (7)$$

The drag is

$$D = k_1 v^2 + \frac{k_2}{v^2} (1 + \tan^2 \phi) \text{ lbs} \qquad (8)$$

The fuel flow is

$$f = c_o + c_1 T + c_2 T^2 \text{ lbs/sec} \qquad (9)$$

The constants for a typical commercial jet aircraft in the velocity range of 150 - 350 knots are:

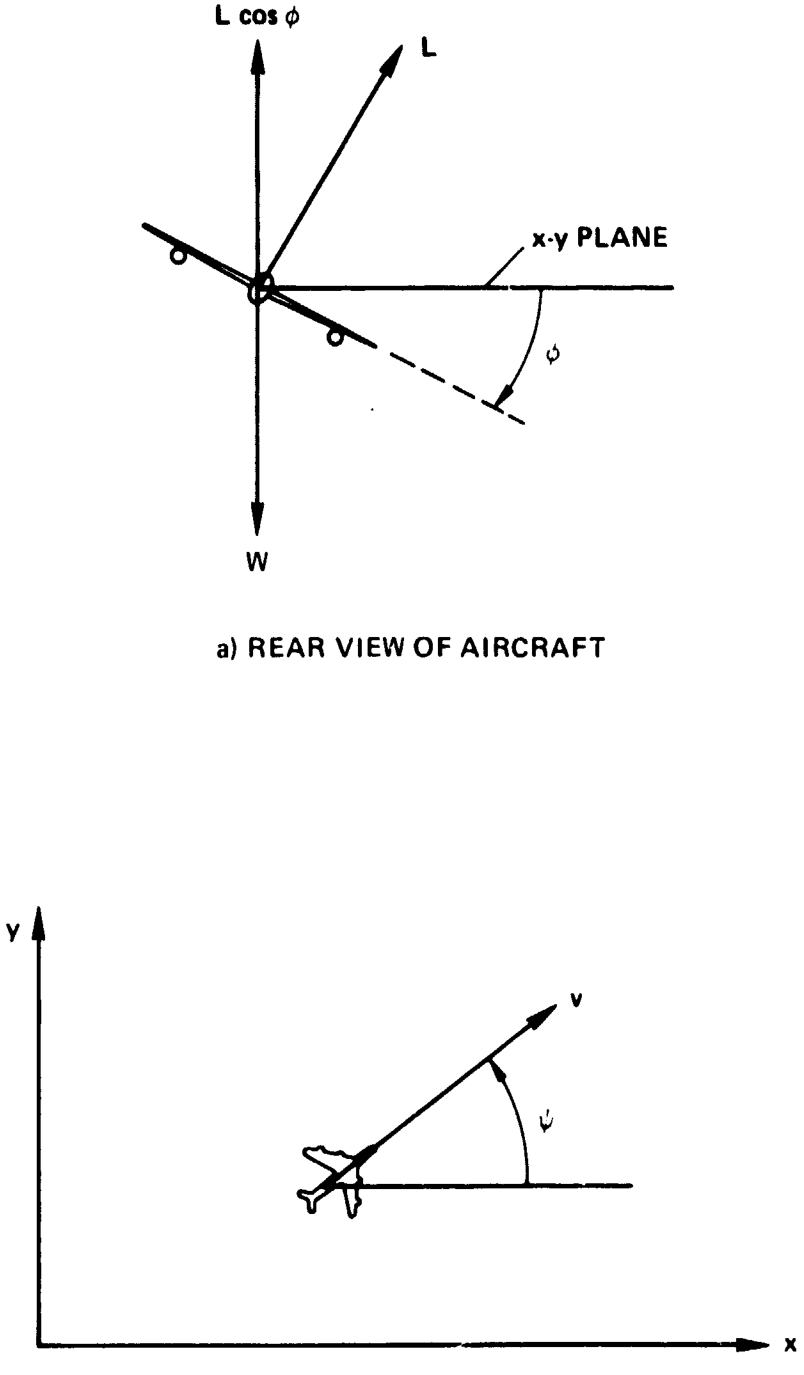
$k_1 = 0.02808 \text{ lbs sec}^2 \text{ ft}^{-2}$ $\qquad g = 32.2 \text{ ft/sec}^2$

$k_2 = 606{,}055{,}000. \text{ lbs ft}^2 \text{ sec}^{-2}$ $\qquad W = 150000. \text{ lbs}$

$c_c = 0.80833 \text{ lbs sec}^{-1}$

$c_1 = 0.000150694 \text{ sec}^{-1}$

$c_2 = 5.4 \times 10 \text{ sec}^{-1} \text{ lbs}^{-1}$



a) REAR VIEW OF AIRCRAFT

b) TOP VIEW OF AIRCRAFT

Figure 8. - Problem Coordinate System.

APPENDIX II

SUBROUTINE DESCRIPTION FOR THE REAL-TIME ALGORITHM

Subroutine Linkage -

	SUBROUTINES CALLED									
Calling Program	LEVEL 3	FDECEL	VSTART	MAXD	NEWWP	PRNT	FUELL	CAPT	SLFP	MAXDEC
MAIN	X	X								
LEVEL 3			X	X	X	X	X	X	X	
FDECEL				X		X				
SLFP										X

SUBROUTINE LEVEL 3

Given: VS = starting velocity, V4 = final velocity, starting position and heading, and final position and heading.

Desired: Construct a flyable trajectory in essentially real time, for any two end conditions (VS ≥ V4) which is a reasonable approximation to a minimum fuel flight path.

Such approximate paths usually have an initial turn, possibly with variable radius, a straight-line flight segment, and a final variable radius turn. For fuel efficiency, the required deceleration is performed as late on the flightpath as possible, and turns are made at the maximum bank angle allowable for the given speed. Depending on the length of the path and the speed differential between starting and final velocity, deceleration may occur in all three flight segments. The final variable radius turn is approximated by a series of constant radius turns, each with a heading change of 30 degrees or less.

1. Calculate the turn radius R1 of the initial turn from VS the starting speed. Check if final speed is greater than initial speed. If yes go to 25.

2. Call VSTART. Calculate the turn radius R2 and the initial speed V1 for a 30 degree circular arc portion of the final turn from the final speed V2, assuming maximum deceleration (thrust = 0) and maximum bank angle at the beginning of the segment.

3. If V1 calculated in step 2 is greater than the initial speed VS, because the actual remaining portion of a circular arc is smaller than 30 degrees or because deceleration begins in the final turn after an initial constant speed turn, the turn radius calculated would be larger than required and will be set to the maximum value R2=R1, (R1 determined in step 1).

4. Call CAPT(6) which constructs the shortest flightpath using R1 and R2 as initial and final turn radii from the specified initial to the specified final point and heading, using only circular turns and a straight line. (If desired CAPT can be made to consider only a final right or a final left turn).

5. Check if final turn is less than 30 degrees (turn completed). If yes go to step 10.

6. Check if this 30 degree segment is V1 greater than VS, that is could we decelerate from VS to V2 in less than a 30 degree turn. If yes go to 19.

7. Call MAXD. From the radius R2, and the initial and final speeds of the 30 degree turn segment V1 and V2 calculate the time to fly, fuel, and distance.

8. Call NEWWP. Calculate the position of the beginning of the turn segment, which defines a waypoint and forms the new endpoint for the continued backward development of the flightpath

9. Go back to step 2.

10. Call MAXD. First segment of the final turn. Using the remaining turn angle and R2 and V2 calculated earlier, find the speed at the beginning of the final turn and the length of the first circular arc.

11. If V1 calculated in step 10 is larger than the maximum speed VS the deceleration begins in the first segment of the final turn, go to 19.

12. Call MAXD. The deceleration starts before the last turn. Depending on whether the trajectory found last by CAPT(6) was a turn-straight-turn or a turn-turn-turn one the distance S is calculated that it would take on the middle segment of flightpath to decelerate from the maximum velocity VS to the speed at the beginning of the final turn.

13. If the length of the middle segment, D, calculated the last time step 4 was executed is longer than S from step 12 the acceleration begins in the middle segment, else go to step 15.

14. From the ratio S/D and the location of the endpoints of the middle segment the position of the beginning of deceleration is calculated. (This calculation is different for a straight-line and a turning-center segment). Go to 2?.

15. The deceleration begins in the first turn. Calculate the speed at the end of the first turn, V1, (which is also the beginning of the middle segment) and the fuel and time for the middle segment, which is completely flown at zero thrust.

16. Check if the first turn is long enough for the required deceleration, by calculating the distance SD it would take to decelerate from VS to V1 with a turn radius R1 and comparing it to the actually available distance calculated in step 4. If okay go to 18.

17. Path required is longer than available. Print that path stretching is required. END.

18. Path is long enough to have a constant speed segment. Calculate where constant speed segment begins and to to 23.

19. Deceleration begins in the last turn. Calculate the time and fuel for the deceleration (MAXD) and position (NEWWP) of the beginning of the deceleration turn.

20. Calculate the remaining constant speed flightpath CAPT(6) and calculate fuel and time on the initial part of the last turn.

21. Calculate fuel and time for the middle segment (straight or turning).

22. Check if middle segment is straight or turning. If straight go to 24.

23. Call FUELC. Calculate fuel and time for the initial constant speed turn. END.

24. The center segment is a straight line. Calculate how much fuel can be saved by accelerating toward the optimum cruising velocity at the beginning of the segment and decelerating back to the nominal velocity at the end. If this maneuver is permitted (no speed limiting) then the possible savings is subtracted from the total fuel for the flightpath under consideration. Go to 23.

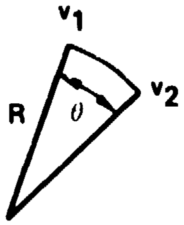
25. When VS ≤ V4 first and last turn are flown at at constant speed with minimum radii for the given speeds. Fly the straight line with initial acceleraton towards the speed for maximum distance per unit of fuel final deceleration towards V4 (call SLFP) END.

FDECEL (X1, Y1, H1, VS, V4, J, L)

Calculates for an initial right (L = 1) or left (L = -1) turn successive coordinates for a 0 thrust decelerating turn, including new radius and new starting speed, total time and fuel used, and prints out the data, where J is the number of 30 degree increments.

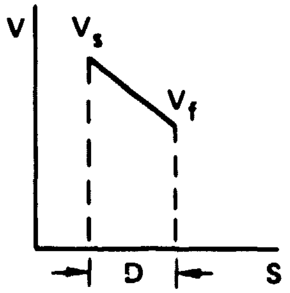
Purpose: To permit an initial decelerating turn to the left or right before developing the remainder of the flightpath in the LEVEL 3 subroutine.

VSTART



Given the end velocity V2 and the turn angle θ the subroutine calculates the starting velocity V1 and the radius of turn R given that the initial bank angle is $\phi = 30^{\circ}$ and a power off decelerated turn is performed.

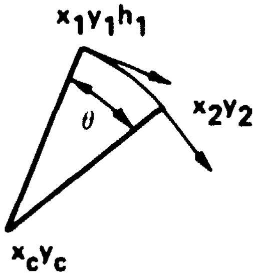
MAXDEC



VS = Starting Vel.
VF = Final Vel.
D = Distance
R = Radius of turn (R = 0 for straight line).

Given two of the three quantities VS, VF, D and R the subroutine calculates the third quantity via Numerical integraton for a power off deceleration of the aircraft. In addition it calculates the time and fuel for the maneuver.
MAXD same as MAXDEC but uses analytical expressions for the calculations.

NEWWP



Calculates heading and position (X, Y) of the next waypoint from the position and heading of the old waypoint as a function of direction of turn, turn angle θ, direction of evaluation (get x_2 y_2 from x_1, y_1, or x_1, y_1, from x_2, y_2).

PRNT

This subroutine converts waypoint variables to desired dimensions for printout, prints out the variables and transfers data to calculate next waypoint coordinates.

CAPT(MX)

COMMON/FUEL/FUELT(20), JT

The subroutine capt will generate from a given position (x_1, y_1, ψ_1) to a final position and heading (x_4, y_4, ψ_4) and initial and final turn radius R_1 and R_4 the minimum length flightpath from the six possible candidates shown on the next page. For the purpose of finding an approximation of a minimum fuel path various types of candidates may be excluded from the search for the minimum length path.

MX = 4 exclude the Turn-turn-turn trajectories from the search (fig. 9.5 and 9.6 next page)

MX = 6 include all types of trajectories

From the types of trajectories not excluded by MX in addition <u>exclude</u>

JT = 1	none others
JT = 2	all that have a <u>left</u> turn last (RSL, LSL, LRL)
JT 1, JT 2	all that have a <u>right</u> turn last (RSR, LSR, RLR)

Types of flightpaths considered by the algorithm for generation of horizontal capture trajectories from a given position and heading to a final position and heading. At least two and at most 6 solutions exist when R_1 and R_2 are given. The shortest path from the existing solutions is chosen.

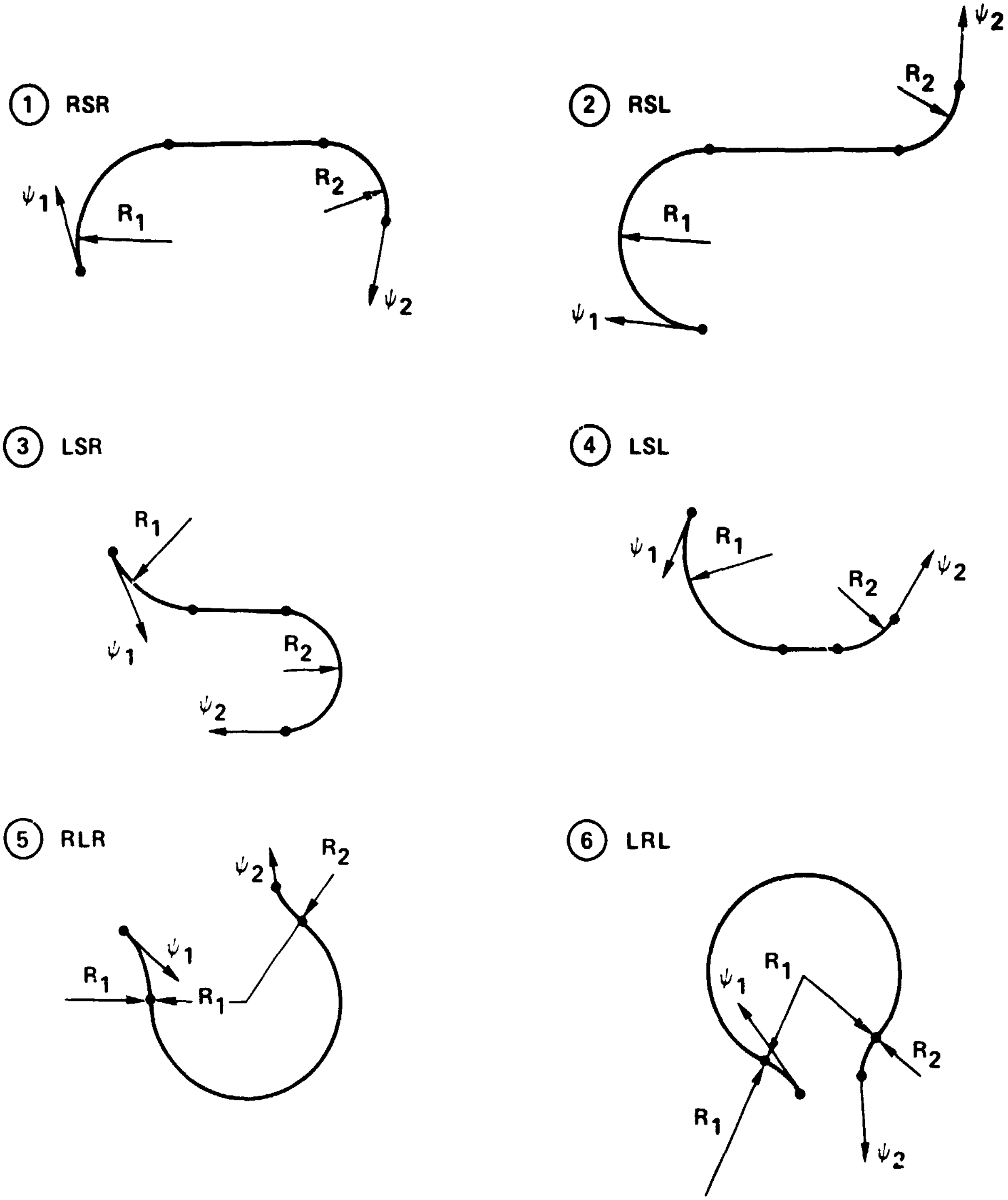
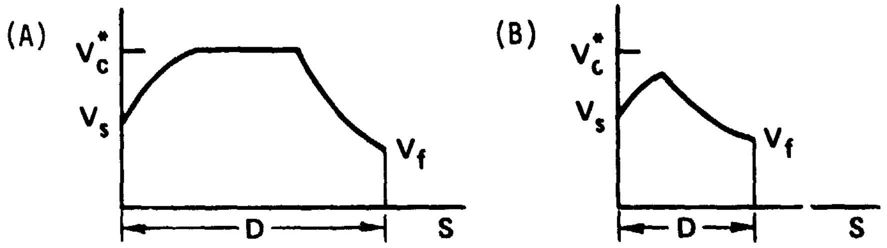


Figure 9. - Types of Capture Trajectories.

SLFP

This subroutine calculates time and fuel it takes for a straight-line flightpath of distance D, starting speed VS and final speed VF. The strategy that is used would be fuel optimum if fuel flow rate were linearly related to thrust (f = k T).

Strategy: Accelerate with max thrust towards the speed at which min. Whether it is reached or not, start decelerating with 0 thrust at the proper point in time so that the final speed is reached at the distance D.



This is accomplished as follows. A distance D, to accelerate or decelerate from V_s to V_f is calculated and compared with D_a, the available distance. If D_a is smaller than D no flightpath exists to meet the end conditions. If D_a is larger than D, the remaining path is generated via numerical integration of the accelerating and decelerating segments simultaneously until either the loitering speed is reached (A) or until the path length D has been achieved before reaching the loitering speed (B). In case of A a constant speed segment is added to achieve the complete distance D_a, and in case of B the calculations are finished.

References

1. Heinz Erzberger, John D. McLean and John F. Barman; "Fixed-Range Optimum Trajectories for Short Haul Aircraft", NASA TN-D 8115, Dec. 1975.

2. Heinz Erzberger and John D. McLean; "Fuel-Conservative Guidance System for Powered-Lift Aircraft", Paper 79-1709, AIAA Guidance and Control Conference, Aug. 6-8, 1979, Boulder, Colorado, also Journal of Guidance and Control, May 1981.

3. W. C. Hoffman, Arthur E. Bryson, Jr.; "A Study of Techniques for Real Time, On-Line Optimum Flight Path Control" Air Force Office of Scientific Research, Contract No. F44620-71-C-0016, Sept. 1971.

4. M. C. Parsons, A. E. Bryson, Jr., W.C. Hoffman; "Long Range Energy-State Maneuvers for Minimum Time to Specified Terminal Conditions", Journal of Optimization Theory and Applications, Vol. 17, Nos. 5-6, Dec. 1975.

5. Eliezer Kreindler and Frank Neuman, "Darboux Points in Minimum-Fuel Aircraft Landing Problems", JACC 1981.

6. John D. McLean; "A New Algorithm for Horizontal Capture Trajectories", NASA TM 81186, March 1980.