

01 Jan 1988

Aircraft Vertical Profile Implementation using Directed-Graph Methods

L. T. Brewster

Paul D. Stigall

Missouri University of Science and Technology, tigall@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

L. T. Brewster and P. D. Stigall, "Aircraft Vertical Profile Implementation using Directed-Graph Methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 6, pp. 682 - 692, Institute of Electrical and Electronics Engineers, Jan 1988.

The definitive version is available at <https://doi.org/10.1109/7.18635>

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Aircraft Vertical Profile Implementation Using Directed-Graph Methods

L.T. BREWSTER, Member, IEEE

P.D. STIGALL, Senior Member, IEEE
University of Missouri-Rolla

Aircraft vertical profile simulation is realized using a demand-driven minimal-calculation directed graph structure to reduce calculation time and to force synchronization of the performance measurement functions with the system state variables. Performance-directed model adaptation makes dynamic vertical profile path corrections, in the presence of fixed drag variations, possible.

Drag variations ranging from +10 percent to -10 percent yielded fuel consumption improvements of less than 1 percent in the majority of the cases.

Calculation time improvement for path simulation ranges from a factor of 1.19 in the worst case to 1.5 in the best case.

Manuscript received January 12, 1988.

IEEE Log No. 24680.

Authors' address: Dept. of Electrical Engineering and Computer Science, 123 Electrical Engineering Bldg., University of Missouri-Rolla, Rolla, Mo., 65401-0249.

0018-9251/88/1100-0682 \$1.00 © 1988 IEEE

I. INTRODUCTION

A. Optimal Profiles

Optimal control problems pervade all areas of science and engineering. All physical problems that transform a system from one state to another consume energy resources in some form or other. Associated with this transformation is the cost of the energy resources used to perform the transformation. It is the concern of these costs that this paper addresses.

Mathematical theories have evolved over many years to handle such problems. Probably the most widely known is the use of calculus to locate simple maximum and minimum conditions over one or more independent variables. This led to the development of a more robust method called the calculus of variations, which is concerned with the maximization and/or minimization of a class of functions called functionals [12, p. 199]. Yet another advancement was introduced by Pontryagin [9] called the maximum principle. Of these methods, the most powerful is the maximum principle.

The reason for this hierarchy of added capability is twofold. First, the calculus has as its result an optimum condition at a given point in the state space of the system under consideration. The calculus of variations approach selects from a class of functions the "best" function (by some metric) that transforms the system from one state to another. Indeed, this description also fits the result obtained by using Pontryagin's method. This leads us into the second property used to classify the three approaches. The ordinary calculus approach views a point optimum without regard for the controls necessary to transform the system to some future state. The calculus of variations approach does not allow independence of the control and state variable vectors [12, p. 219]. Pontryagin's method allows for the constraint of any or all state variables. This property makes Pontryagin's method the one of choice in dealing with physical problems, as virtually all physical problems have constraints on their state variables.

However, physical problem solutions rely on prior system performance knowledge to optimize the state variable paths. Without this information, system optimization is not possible. How critical is the path deviation? What gains could be realized if a real-time correction to the state variable path were performed? What tools would be necessary to achieve these goals? These are the questions that are addressed here.

Physical process control problems amenable to Pontryagin's optimum control theory are two-point boundary control problems. Inherent with this class of problems is the need to have at one's disposal the prior knowledge of the initial as well as the final state of the process before the path (or time history) of the control variables can be generated. In many cases, simulation of such problems is necessary to determine this information. Inherent with simulation is the introduction of errors,

such as model approximations, that lead the simulation process to false conclusions about the optimal path.

A method that uses real-time simulation and model adaptation was developed to improve on minimal cost state variable transitions in the presence of errors. The simulation process incorporates a demand-driven minimal-calculation directed graph structure to minimize calculation time and to synchronize the performance measurement values with the state variables of the system.

The problem chosen for investigation was the minimum fuel consumption vertical profile of a turbojet-powered transport category aircraft. The goal was to determine optimality in the presence of fixed drag variations. An aircraft design was created using typical wing and structure drag estimations, engine modeling, and specific fuel consumption estimates. The aircraft model was then simulated on a microprocessor-based computer and performance data extracted. The optimum profiles were generated and "flown" with nonstandard versions of the aircraft (i.e., variations in drag). Dynamic performance measurements of the nonstandard versions were used to calculate real-time corrections to the flight path to determine the amount of savings that could be realized.

B. Software Methods

A demand-driven minimal-calculation directed graph method was developed to reduce the number of calculations performed during profile optimization and to synchronize the performance measurements with the state variables. This method is achieved by creating a directed graph structure that contains function values as nodes. The links of the directed graph are pointers to the successor functions that use the current function value as an argument. Retention of the last returned value of the function is a central property of the function structure. In addition, the validity of the current value is maintained. If no predecessor functions have been altered since the last calculation, then the function node retains its valid status, and the function value is returned without further calculation. If a predecessor is altered, its successors are marked as invalid, and the function is evaluated to reflect the current state of the system.

It should be noted that this method requires two classes of variables. The first class is referred to as **system** variables and the second as **ordinary** variables.

System variables are those variables that are internal to the system under simulation and must be kept current to maintain system consistency. An example would be the drag error which has a system effect should it be changed. Ordinary variables are those that keep external or historical information about the past or future desired state of the system. An example would be the optimum mach and altitude that is the goal state of the system during climb and/or descent.

C. Development System

The development system used during this research consists of an IBM PC-XT with a Definicon DSI-020 accelerator board installed. The Definicon board has 1 Mbyte of memory, a Motorola 68020 microprocessor, and a Motorola 68881 math coprocessor that run at a clock speed of 12.5 MHz. The programs were coded in PASCAL as compiled with a Silicon Valley Software certified compiler. Timing comparison data was obtained on two programs with the use of a stop watch. One of the programs used conventional functions and variables, and the other used the demand-driven minimal-calculation directed graph data structure.

II. AIRCRAFT MODEL

A. Airframe

1) *Basic Structure*: The basic airframe structure used in the case study is a medium-sized twin engine turbojet-powered transport category aircraft. This model was chosen because of the number of airframes in this category that are currently being used and the number of proposed designs that are similar. The structure is a widebody design of 330,000 lb maximum gross takeoff weight. The wings, horizontal stabilizer and vertical stabilizer have areas of 3,000 ft², 900 ft² and 600 ft², respectively, and they are moderately swept. The engines are mounted on pods below the wing. A commercially available design aid marketed by Kern International [10] was used to create the efficiency factors and drag tables for the aircraft model.

2) *Specific Design Criteria*: Seven categories of information are needed by the Kern design aid to specify the aircraft structure. The categories are basic data, wing, horizontal tail, vertical tail, fuselage, nacelles, and miscellaneous. Each of these areas has several items of information that define the parameters necessary for calculating the drag coefficients and wing span efficiencies (see Table I).

Other terms used in Table I are as follows. The zero fuel weight plus the fuel weight is the aircraft total gross weight. " $C_{l(max)}$ " is the maximum value of the coefficient of lift. Theoretical wing area is the reference area that is used in the equations to calculate total drag. The chord of a structure, such as wing, horizontal tail, etc., is the measurement taken perpendicular to its length (or span). Aspect ratio [2, p. 149] is the square of the span divided by the area. Taper ratio is the chord length at the tip of the surface divided by the chord length at the root. Defining a relative position along the chord of a surface (Fig. 1) can be done by specifying a fraction of the distance from the leading edge of that surface. This fraction distance is called x in Table I and defines a span line some fraction of the chord aft of the leading edge. The entry "sweep of $x/c = 0.25$ " means take the span line positioned 25 percent aft of the leading edge, and

TABLE I
Airframe Data

Basic Data		Vertical Tail	
zero fuel weight	22000	single fin area	600
fuel weight	110000	aspect ratio	1.5
engines	2	taper ratio	0.25
thrust multi	1	sweep of x/c	35
fuel flow multi	1	x/c	0.25
c _l (max)	1.5	(t/c)max	0.08
		loc of (t/c)max	0.4
Wing		Fuselage	
theoretical area	3000	wetted area	8500
aspect ratio	8.7	length	160
taper ratio	0.3	width	18
sweep of x/c	34	height	18
x/c	0.25		
(t/c)max	0.09		
loc of (t/c)max	0.6		
airfoil m/c	0.005875		
Horizontal Tail		Nacelles	
exposed area	900	wetted area	600
aspect ratio	4	length	16
taper ratio	0.3	diameter	10
sweep of x/c	38		
x/c	0.25		
(t/c)max	0.08	Miscellaneous	
loc of (t/c)max	0.4	misc factor	0.05

Note: Roman symbols here correspond to italic symbol in text.

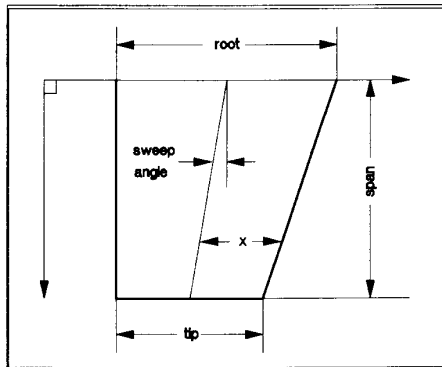


Fig. 1. Surface terms.

measure the angle at which this line intersects the longitudinal axis. The term "airfoil m/c " is the wing leading edge radius of curvature [7, section 4.1]. "Wetted area" [2, p. 201] is the total surface area in contact with the air.

A series of figures (Figs. 2–5) illustrate the data generated by the Kern design aid. Fig. 2 is a plot of the zero lift coefficient of drag for the airplane at 37,000 ft and is typical of that which the aircraft experiences during cruise flight. The sharp rise that occurs at approximately 0.84 mach is due to compressible air flow as the air flowing over the top surface of the wing (and around other surfaces) reaches supersonic speeds. At higher altitudes and airspeeds, this contribution to drag is

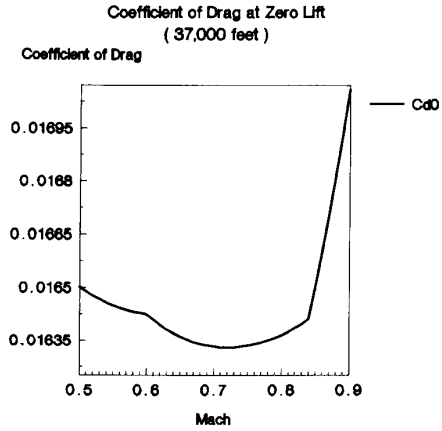


Fig. 2. Coefficient of drag.

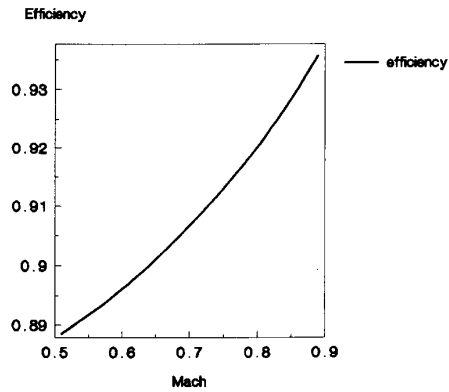


Fig. 3. Efficiency function.

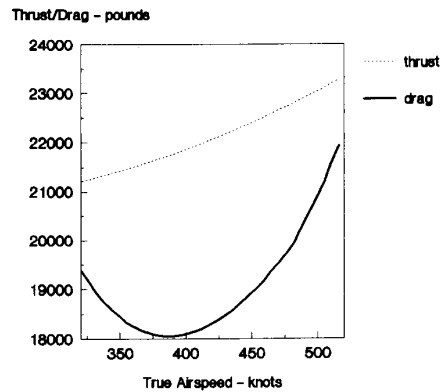


Fig. 4. Thrust and drag.

significant. Span efficiency factor (Fig. 3) is a function of mach number only and approaches a value of one as mach increases. This is one of the factors used to calculate the total drag of the aircraft.

A typical plot of maximum thrust available and total drag is seen in Fig. 4. Fig. 5 displays the altitude/velocity capability of the aircraft at a weight of 270,000 lb. The limit to the left of the plot is determined by the stall

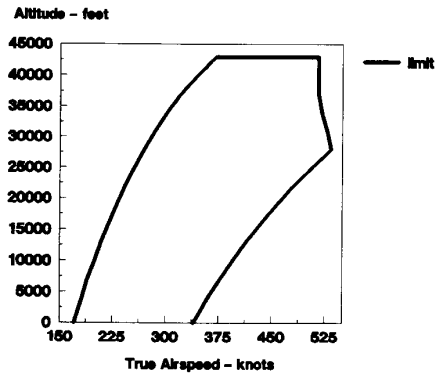


Fig. 5. Operating envelope.

speed of the aircraft. The upper limit is due to the climb capability and is a function of thrust. The right limit is determined by one of two quantities: 1) at the lower altitudes, the limiting factor is the maximum velocity which is in part determined by the structural strength; and 2) at the upper altitudes, the limiting factor is the mach number which is determined by strength, control, and stability considerations.

The structurally dependent information covered in this section that is necessary to determine the total drag of the structure is: the coefficient of drag at zero lift (Fig. 2), the wing span efficiency factor (Fig. 3), the coefficient of lift (described in a later section), the wing reference area, and wing aspect ratio. It should be noted that a scaling factor of 1.3 is applied to the total drag generated by the Kern program.

B. Engine

The parameters that define the engine model are the thrust and specific fuel consumption (SFC). The engine model is a turbofan¹ jet engine and conforms to the model used by Stengel and Marcus [15, pp. 197–198] as far as thrust curve shape is concerned. Since a standard atmosphere is assumed throughout the case study, all references to nonstandard temperature have been eliminated in the engine model equation. The resulting equation (Fig. 6) was further modified by adding a constant multiplying factor of 5.5 to adjust the thrust model to the structural model. The standard-day-sea-level-rated thrust of the engine model is approximately 72,000 lb.

The equation for SFC is a function of throttle position, mach number, and temperature ratio (Fig. 7) as suggested in part by Miele [8, pp. 100–112]. The SFC value is nominally 0.55 lb of thrust per pound of fuel burned.

¹A turbofan engine is an axial flow turbojet engine that develops a portion of its thrust by incorporating a ducted fan that processes air that bypasses the turbine section of the engine.

```

normalized altitude hn := (altitude - 25500.0) / 29500.0
normalized mach nm := (mach - 0.55) / 0.25

constants  a := 5.500    e := 731.9590
           b := 5763.499  f := 1609.4457
           c := 5590.819  g := 805.6959
           d := 77.995    h := 385.8750

max_thrust := a * (b - c * hn + d * nm2 + e * nm * hn
               + f * hn2 - g * nm * hn2 + h * hn3)

```

Fig. 6. Thrust equation.

```

function sfc : double;
const  a := 5.93541458430015E+0;
       b := -6.99118425982775E-1;
       c := 5.51294828131334E-1;
       d := 6.57477405255063E-1;
       e := -3.02057045923884E-1;
       f := -1.07647006300008E+1;
       g := 5.18015495222687E+0;

var    t : double;

begin
  t := tropopause_temperature / temperature;
  sfc := ( a + b * throttle + c * sqrt( throttle )
          + d * mach + e * sqrt( mach )
          + f * sqrt( t ) + g * t ) * sfcfactor;
end {sfc};

```

Fig. 7. SFC equation.

III. THEORETICAL BASIS

The theoretical basis for the computation of an optimal (in the sense of fuel consumption) vertical profile for the aircraft is the mathematics presented by Pontryagin [9, p. 1] in his development of the **maximum principle**.

The physical processes which take place in technology are, as a rule, controllable, i.e., they can be realized by the various means depending on the will of man. In this connection, there arises the question of finding the very best (in one sense or another) or, as is said, the *optimal* control of the process. . . . Mathematically formulated, these are problems in the calculus of variations, which in fact owes its origin to these problems. However, the solution of a whole range of variational problems, which are important in contemporary technology, is outside the classical calculus of variations. . . . In its essential features, this solution is unified in one general mathematical method, which we call the maximum principle.

Since there are several references that treat Pontryagin's maximum principle in detail [9, 12], only a general overview is presented. Also, Erzberger [3] and Sorensen [14] developed the necessary specifics to apply Pontryagin's maximum principle to the optimal vertical profile problem. These references describe the principles which are the basis for the construction of the profiles used in this research.

The two essential aspects of the maximum principle are to determine the location of the optimum state and to determine the progression from an arbitrary initial state to the optimal state. The optimum state for the vertical profile problem occurs at the minimum value of the time rate of change in cost divided by the horizontal velocity. This reference or terminal condition (H_c) is calculated by

$$H_c = \frac{C_f \dot{w} + C_t}{V \cos \gamma} \quad (1)$$

The equation that is used to determine the state transition path is

$$-\lambda = \frac{(C_f \dot{w} + C_t - H_c V \cos \gamma) mg}{(T - D)V} \quad (2)$$

and is minimized at each energy level along the path.

The symbol C_f is the cost per pound of fuel, C_t is the time cost, \dot{w} is the fuel-burn rate, V is the true airspeed, T is thrust, D is drag, m is mass, g is gravity, and γ is the angle of climb with respect to the horizon [14, eqn. (10) and (11)].

IV. SIMULATION MODEL

A. Model Assumptions

The forces acting on an aircraft in flight (Fig. 8) are thrust, drag, lift, and gravity. The equations that describe a point mass of the aircraft model are

$$m \dot{V}_a = T \cos \alpha - D - mg \sin \gamma$$

$$m V_a \dot{\gamma} = T \sin \alpha + L - mg \cos \gamma$$

$$\dot{h} = V_a \sin \gamma$$

$$\dot{x} = V_a \cos \gamma$$

$$\dot{m} = -\dot{w}$$

For general trajectory shaping, the flight path angle dynamics can be neglected, and it can be assumed that the angle-of-attack α is small. Then, the equations of motion of the aircraft in the longitudinal plane are

$$\dot{V}_a = (T - D)/m - g \sin \gamma$$

$$\dot{h} = V_a \sin \gamma$$

$$\dot{x} = V_a \cos \gamma + V_w$$

where the longitudinal component of the wind is included.²

The symbol V_a is true airspeed, m is mass, T is thrust, D is drag, g is the gravitational constant, γ is the climb angle, \dot{h} is rate of climb, and \dot{x} is the horizontal component of true airspeed. In this presentation the wind velocity V_w is assumed to be zero; hence, V without the subscript a refers to true airspeed.

Standard atmosphere is assumed, which makes

²See [14].

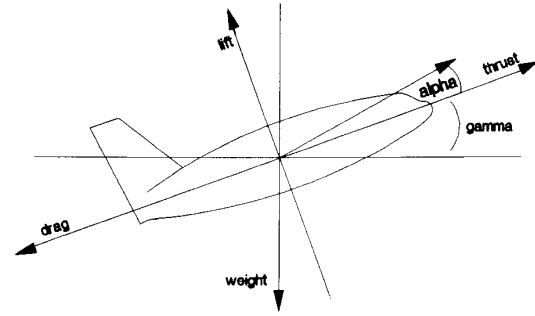


Fig. 8. Aircraft force vectors.

temperature, pressure, and density all functions of altitude only.

Limits have been placed on weight, altitude, mach, and the drag error function. Initial weight is bounded between 180,000 and 330,000 lbs; however, once the simulation is in progress it is possible to burn sufficient fuel to reduce the weight below 180,000 lbs. Altitude limits are from sea level to 43,000 ft and are determined by the drag data available. In the same manner, mach number is restricted to the values of 0.21 to 0.90 of the speed of sound. Drag error is bounded between 0.8 and 1.2 of standard drag.

The upper limit of throttle position is 1.0, and the lower limit is dependent on density ratio. At sea level the lower limit is 0.063 and increases as the density ratio decreases.

Since only quasi-steady-state conditions of the point mass model are allowed, angle-of-attack (α) is assumed to be a function of the coefficient of lift. This means that α is adjusted to a value that balances the vertical component of lift with the weight of the aircraft. With respect to the coefficient of lift in all flight conditions, α is linear.

B. Simulation Implementation

A flight can be segmented into three parts: climb, cruise, and descent. There can be several climb segments if altitude change is allowed as weight is reduced through fuel consumption. This is referred to as step climb. Additionally, there are situations where there is no cruise segment. This occurs when the distance needed to climb and descend is equal to the flight length.

During climb, the airspeed is determined by minimizing lambda (2). A search, starting at the maximum airspeed, is performed until an airspeed that minimizes lambda is located.

After convergence of the climb fuel burn, there are two actions possible depending on whether or not a drag error corrected profile is desired. If correction is not preferred, then the profile is flown as simulated, and the time, distance, and burn values are logged. If correction is in effect, a measurement of the initial climb rate is performed and compared with the calculated value. Correction is then made to the drag function to account

for the performance difference,³ and the climb process is repeated from the beginning.

There are two means of terminating the climb segment: 1) reaching the optimum altitude, or 2) reaching the fixed cruise altitude specified for the profile. When the fixed cruise altitude specified for the profile is an altitude above the current optimum, then the optimum supersedes the fixed altitude.

It is necessary to know the top-of-climb weight for the aircraft in order to determine the optimal cruise point. An estimate of climb fuel burn is made, and the weight of the aircraft adjusted. The climb segment then begins with a search over altitude and airspeed for the point at which the fuel-burn rate divided by the horizontal velocity is a minimum. This point is the terminal boundary condition of the climb segment. The initial boundary conditions are defined as sea level and initial climb airspeed. Since the optimum point is a global minimum [4], a linear search is conducted from the maximum allowable altitude and airspeed until the optimal point is located. An iteration over the climb segment with burn estimate correction using rms convergence is performed until the simulated burn is within tolerance of the estimated burn. This is accomplished by using the theoretical drag values.

In fact, the cruise segment may be followed by another climb segment. As fuel is burned, the optimum altitude increases, and the optimal speed decreases due to weight reduction. Every 50 nmi, airspeed is adjusted and altitude checked to see if the next higher altitude is more economical. If this is the case, a step climb is performed. However, if the distance to destination is 200 nmi or less, the step climb is inhibited. As always, the lower of the optimum or the specified altitude is the limit.

When the distance remaining reaches 200 nmi or less, a trial descent profile is performed using sea level as the target altitude and 1.2 times the stall speed of the aircraft as the airspeed target. The distance and fuel burn calculated by the trial descent is used to determine the termination point of the cruise segment.

The optimum profile is determined by running an unrestricted profile and then performing repeated fixed cruise profiles at lower altitudes until the optimum cruise altitude is determined.

C. Directed Graph Method

1) *General Overview:* A demand-driven minimal-calculation directed graph method for execution time minimization through the reduction of unnecessary reevaluation of functions is defined and implemented here. This concept requires a knowledge of the structure of all functions and variables in the program. The problem presented here is a state variable system that requires synchronization or consistency of all state variables, internal variables and performance functions. In

³This simulation assumes that only drag variations occur.

addition, there are requirements for external or goal variables that must retain their values independent of any system state transitions. These two classes of variables are referred to as **system** variables and **ordinary** variables. System variables must retain system consistency, such as the drag factor that is used to correct for drag errors in the system. Whereas, the ordinary variables, such as optimum mach and optimum altitude used in climb and descent, must retain their values as the system goes through various state transitions.

Since all nodes in the structure are function values, system variables that appear as predecessors (i.e., to the right of the assignment symbol) must be converted to functions since all nodes in the structure are function values. This function structure or hierarchy is necessary to determine the links in a directed graph. Links in the directed graph represent the hierarchy of dependence of the functions. Nodes of the graph contain the last function evaluation results. This structure consists of units that contain three pieces of information: a status, a value, and a successor link. The meaning of these units is: 1) **status** indicates whether or not a value needs to be recalculated, 2) **value** is the last calculated value of function, and 3) **link** connects this function unit to all dependent functions.

2) *Constructing the Graph:* In a case study, a subset of function units serves as an example for the method of constructing a linkage structure. The graph subsection of the function “**attitude**” depicts the general method of generating the system-directed graph. Sample functions are

Attitude	() which is independent
Gamma	(attitude, alfa)
\dot{V}	($T - D$, mass, gamma)
\dot{X}	(TAS, gamma)
\dot{H}	(\dot{E} , weight, TAS, \dot{V})
Psi	(\dot{C} , \dot{X})
Lambda	($T - D$, TAS, weight, \dot{C} , \dot{X}).

The following method (see Fig. 9) is used to determine the successors. Select the first function (attitude) and locate all functions that depend on it. The only one is gamma, so Attitude has only one successor. Next select Gamma which appears as a predecessor of \dot{V} and \dot{X} . This makes \dot{V} and \dot{X} successors of Gamma. This process is continued until all function successors are determined. At this time, the linked list is constructed (Fig. 9).

3) *How a Directed-Graph Program Works:* Following a chain of pointers to maintain proper status markings on all node values creates some overhead when assigning the unit values during execution.

All value inquiries are performed by a **demand** function (Fig. 10) that returns a Boolean value which indicates whether or not the unit should be recalculated.

Assignment of the values is executed by the **assign** function (Fig. 11) which alters the value of the unit and

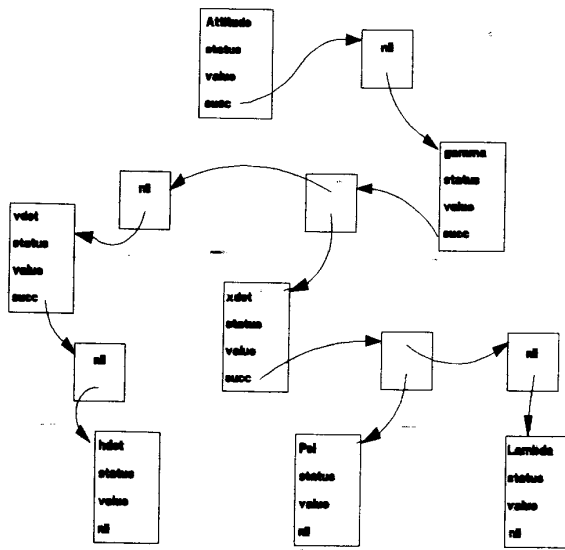


Fig. 9. Link structure. Note: vdot, hdot, xdot correspond to \dot{v} , \dot{h} , \dot{x} in text.

```
function demand( element : itemptr;
                var number : double )
    { returns } : boolean;
begin
    with element^ do begin
        demand := status;
        number := value;
    end {with};
end {demand};
```

Fig. 10. Demand function.

```
procedure assign ( element : itemptr;
                 data : double);
begin
    with element^ do begin
        if value <> data then begin
            value := data;
            if succ <> nil then mark_list( succ );
            status := true;
        end {if};
    end {with};
end {assign};
```

Fig. 11. Assign function.

invokes a function called **mark-list** (Fig. 12) which informs all successors that the unit value has changed. The mark and mark-list procedures work together to denote all successors of a unit as invalid. These procedures are employed only if an assignment to a node value results in a change.

```
procedure mark( element : itemptr ) : forward;
procedure mark_list( ptr : chainptr );
begin
    with ptr^ do begin
        mark( item );
        if link <> nil then mark_list( link );
    end {with};
end {mark_list};

procedure mark; {see forward declaration above}
begin
    with element^ do begin
        status := false;
        if succ <> nil then mark_list( succ );
    end {with};
end {mark};
```

Fig. 12. Mark and mark-list functions.

System variables must be handled as functions to maintain system consistency. One problem does arise as conventional variables can have left values (i.e., can appear on the left of an assignment operator). This problem is resolved by substituting the **assign** function for each occurrence of a left-valued variable.

4) *Converting a Conventional Program:* Converting a program using the directed-graph method involves determining which specific functions and variables must be internal and which ones must be external to the system. To determine the class in which a variable should be placed, the following question must be answered: "Is it necessary for this function or variable to be consistent with the system, or must it retain its value as the system makes state transitions?" If consistency is not desired, then it is listed in the class of ordinary variables; otherwise, it is placed in the class of system variables.

The necessary modification of a conventional program only concerns the functions and system variables. Converting a function (Fig. 13) is attained by adding an inquiry function as the entry statement of each function and an assignment statement just after the evaluation statements. Depending on the compiler, it may also be necessary to add a dummy variable to transport values to and from the nodes of the graph. Since system variables are nodes in the directed graph, they must be converted to functions. All nodes must be functions. This is effected by using the demand function as seen in Fig. 14. No alteration of ordinary variables is necessary.

5) *Directed-Graph Example:* Program initialization marks the **status** of each node (Fig. 9) invalid (false) to force initial calculation of requested values. If the function \dot{V} (Fig. 13) is invoked, the **demand** function (Fig. 10) is also invoked and returns false which forces the calculation of \dot{V} . Since \dot{V} invokes γ , this scenario is repeated for γ . This sequence continues until all pertinent functions are evaluated. The independent functions, such


```

Standard program function :

function vdot : real;
begin
  vdot := ( t_d ) / mass - gravity * sin( gamma );
end {vdot};

```

```

Converted to directed graph function :

function vdot : real;
var x : real;
begin
  if not demand( vdotp, x ) then begin
    x := ( t_d ) / mass - gravity * sin( gamma );
    assign( vdotp, x );
  end {if};
  vdot := x;
end {vdot};

```

Fig. 13. Program modification. Note: vdot corresponds to \dot{v} in text.

```

Converting the variable 'dragfactor' to a function

function dragfactor : double;
var dummy : boolean;
    anything : double;
begin
  dummy := demand( dragfactorp, anything );
  dragfactor := anything;
end {dragfactor};

```

Fig. 14. Converting a variable.

as attitude, always return their current value. As each dependent function is evaluated, its node is marked valid, and the calculated value is returned until the entire chain is traversed (in this case, resulting in a valid value for \dot{V}). In this state, an invocation of \dot{H} (Fig. 9) only requires the calculation of \dot{H} since \dot{V} (Fig. 13) will return its current valid value.

The counter example is illustrated by assigning a new value to attitude. The `assign` procedure (Fig. 11) is used to alter the node values. If this assignment does not alter the node value, no further action is taken; however, if the value is altered, the `mark-list` procedure (Fig. 12) traverses the directed graph (Fig. 9) marking all nodes that depend on attitude as invalid.

V. RESULTS

A. Optimal Profiles

For the aircraft model selected, research indicates that a modest reduction (usually less than one percent) in fuel consumption (Table II) can be realized by using real-time adaptive techniques for optimal path correction.

TABLE II
Fuel Consumption Table

Drag Change	Distance	Uncorrected Burn	Corrected Burn	Pounds Saved	Percent Saved
+10	300	8480.6	8476.2	4.4	0.05
+10	1000	24004.7	23903.4	101.3	0.42
+10	3000	63552.3	63232.4	319.9	0.50
+5	300	8093.1	8082.0	11.1	0.14
+5	1000	22666.8	22641.7	25.1	0.11
+5	3000	60005.1	59949.6	55.5	0.09
-5	300	7358.9	7350.7	8.2	0.11
-5	1000	20218.6	20173.4	45.2	0.22
-5	3000	53437.3	53279.1	158.2	0.30
-10	300	7012.6	6994.6	18.0	0.26
-10	1000	19099.3	18956.5	142.8	0.75
-10	3000	50425.9	49872.2	553.7	1.10

Several data plots, Figs. 15–19, pictorially describe the data generated by the simulator. The example shown is the 300 nmi limited-climb minimum-fuel profile.

The altitude capability graph, Fig. 15, shows that the optimum altitude (minimum fuel per distance traveled) occurs very near the maximum altitude capability of the aircraft. Fig. 16 shows the altitude and true airspeed path taken during two profiles. The first is labeled full climb and is the path that reaches the minimum fuel per distance altitude. The second is labeled limited climb and

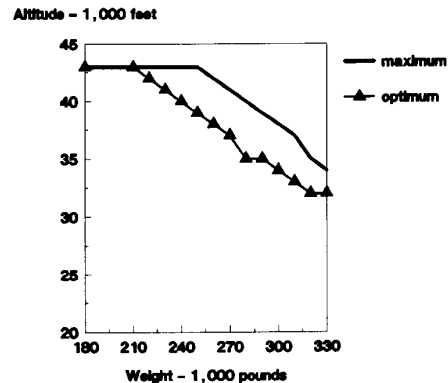


Fig. 15. Altitude capability.

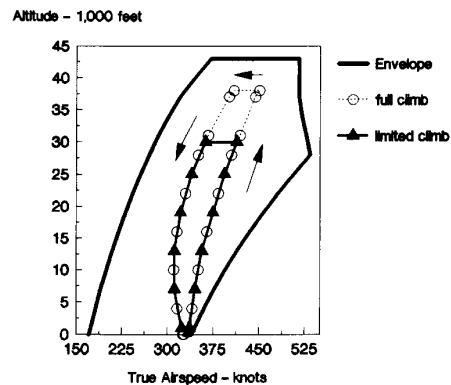


Fig. 16. True airspeed path.

is the path taken that gives the minimum fuel consumed for the entire flight.

It should be noted that the maximum altitude actually reached during a flight may be less than the minimum fuel per distance altitude. Fig. 17 shows the altitude-distance relationship for a 300 nmi flight. The minimum-fuel profile occurs at a cruise altitude of 30,000 ft. A limited altitude profile is illustrated in Fig. 18 which shows that leveling off at a lower altitude results in less fuel per distance.

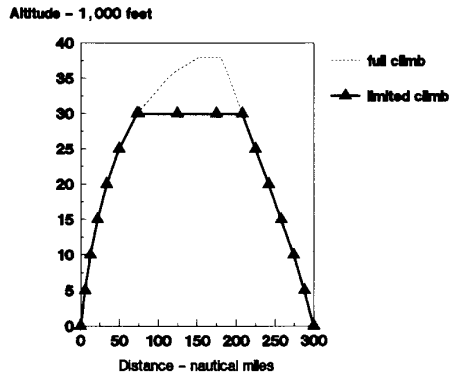


Fig. 17. Profiles.

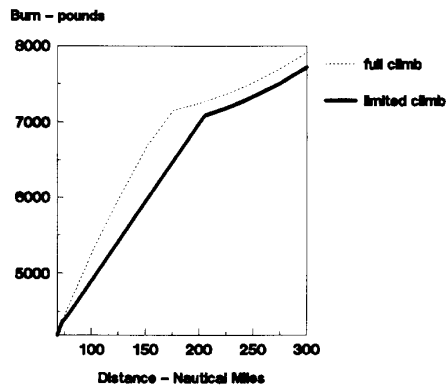


Fig. 18. Burn comparison.

The optimum profile is determined by generating profiles with varying altitude limits and selecting the minimum fuel burn. Flight length is critical in formulating the optimum altitude. Over a limited distance, climbing from one altitude to another consumes more fuel than cruising at the original altitude. However, if the distance to cruise is sufficiently long, the lower fuel per distance at the higher altitude will recover the cost necessary to perform the climb.

Prior to climbing to a higher altitude, two factors must be considered. First, is the instantaneous economy better, and second, is there sufficient distance at this better economy to recover the added cost of climbing? It should be pointed out that this property is due to the fact that a fixed throttle climb and descent simulation (Fig. 19) is used [14, pp. 94-95].

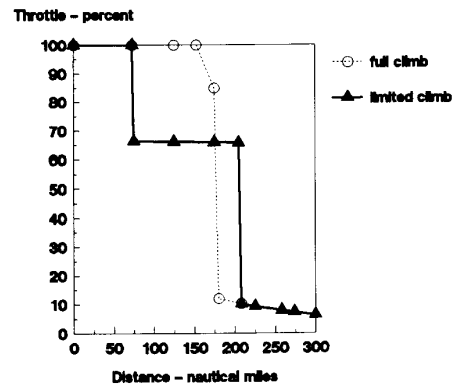


Fig. 19. Throttle path.

If thrust is allowed to vary during climb and descent, the profiles are parabolic in shape for flight lengths where the maximum altitude of the aircraft is not reached. Obviously, flight lengths that are sufficiently long for the aircraft to reach the maximum altitude of the aircraft will once again contain a level cruise section. Profiles of this nature were not studied during this research.

B. Directed-Graph Method

The demand-driven minimal-calculation directed-graph structure developed to reduce calculation time (Table III) and to force synchronization of the performance

TABLE III
Directed-Graph Timing Table

Distance	Profile	Digraph	Dtime	Factor
300	2:07.91	1:25.28	42.63	1.50
1000	2:28.96	1:48.85	40.11	1.37
3000	3:25.80	2:52.55	33.25	1.19
5000	3:37.23	3:02.80	34.43	1.19

measurement values with the state variables of the system, yielded significant performance improvement. Profile simulations that are predominantly climb and descent calculations, demonstrated improvements by a factor of 1.5; whereas, profiles that are mostly cruise calculations showed improvement factors of 1.19. Program size was increased by approximately 20 percent in this implementation. However, it should be pointed out that there is a sizeable amount of code used to generate the graph structure.

VI. CONCLUSIONS AND FUTURE INVESTIGATIONS

A. Optimal Profiles

The proposed method confirms that improvements in fuel consumption are possible using adaptive techniques.

However, the magnitudes of these gains are less than encouraging.

The fuel savings made possible by using an adaptive method, as described in this presentation, is dependent on the shape of the drag and thrust curves. Engines with thrust curves that are less linear than the turbojet and less conventional wing designs may, in fact, be more critical in the selection of altitude and/or airspeed. Investigations using various power plant technologies and wing designs should be conducted to determine their susceptibility to adaptive modeling.

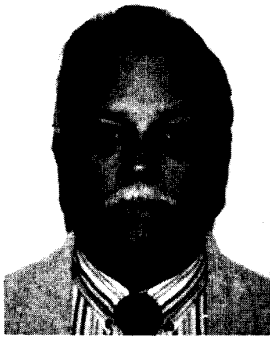
Since such large quantities of fuel are consumed by the aviation industry, any improvement, however small, must be thoroughly investigated before being discarded. Simulation of an actual aircraft should be undertaken using the drag, thrust, and specific fuel consumption of the physical aircraft.

B. Directed-Graph Method

It is the opinion of the author that, if the directed-graph method were made a part of the compiler generating the code, the program size increase would be substantially less than the 20 percent observed in this example. The only modification to normal programming methods necessary to use the directed-graph approach would be the addition of the system variable as a new data type. The programmer would have to distinguish between the system variables and ordinary variables. Additionally, various types of programs should be used as examples to determine the relationship of time improvement to program type.

REFERENCES

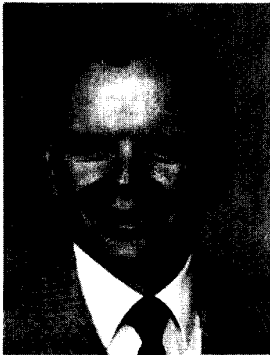
- [1] Burrows, J.W. (1982)
Fuel optimal trajectory computation.
AIAA Journal of Aircraft, 19 (Apr. 1982), 324–29.
- [2] Dommasch, D.O., et al. (1961)
Airplane Aerodynamics (3rd ed.).
New York: Pitman, 1961.
- [3] Erzberger, H., and Homer, L. (1980)
Constrained optimal trajectories with specific range.
AIAA Journal of Guidance and Control, 3, (Jan.–Feb. 1980).
- [4] Erzberger, H., and Homer, L. (1980)
Algorithm for fixed-range optimal trajectories.
Technical paper 1565, NASA, July 1980.
- [5] Farmer, R.H. (1981)
Flight management systems for modern jet aircraft.
In *Proceedings of the National Aerospace and Electronics Conference*, 2, May 1981, pp. 823–29.
- [6] Gracey, C., and Price, D.B. (1983)
Altitude/path-angle transitions in fuel-optimal problems for transport aircraft.
In *IEEE Proceedings of the 1983 American Control Conference*, 2, Sept. 1983, pp. 519–25.
- [7] Hoak, D.E., and Finck, R.D. (1975)
United States Air Force contract report 33(616)-6460 revised F33615-74-C-3021.
United States Air Force Stability and Control DATCOM (N76-73204), Oct. 1960, revised Jan. 1975.
- [8] Miele, A. (1962)
Flight Mechanics, Theory of Flight Paths, Vol. 1.
Reading, Mass.: Addison-Wesley, 1962.
- [9] Pontryagin, L.S., et al. (1962)
The Mathematical Theory of Optimal Processes. Edited by L.W. Neustadt. Translated by K.N. Trivogoff.
New York: Wiley, 1962.
- [10] Powers, S.A. (1984)
Basic Aircraft Performance.
Duxbury, Mass.: Kern Publications, 1984.
- [11] Rutowski, E.S. (1954)
Energy approach to the general aircraft performance problem.
Journal of the Aeronautical Sciences (Mar. 1954), 187–95.
- [12] Schultz, D.G., and Melsa, J.L. (1967)
State Functions and Linear Control Systems.
New York: McGraw-Hill, 1967.
- [13] Sorensen, J.A. (1983)
Generation and evaluation of near-optimal vertical flight profiles.
In *IEEE Proceedings of the 1983 American Control Conference*, 2, Sept. 1983, pp. 513–18.
- [14] Sorensen, J.A., and Waters, M.H. (1981)
Generation of optimal vertical profiles for an advanced flight management system.
NASA contract report 165674, Mar. 1981.
- [15] Stengel, R.F., and Marcus, F.J. (1976)
Energy management techniques for fuel conservation in military transport aircraft.
Air Force Flight Dynamics Laboratory report AFFDL-TR-75-156 and National Technical Information Service AD-A023 527, 1976.
- [16] Stengel, R.F., and Marcus, F.J. (1976)
Energy management techniques for fuel conservation in military transport aircraft.
IEEE Transactions on Aerospace and Electronic Systems, AES-12, 4 (July 1976), 464–71.
- [17] Wauer, J.C., et al. (1981)
Aircraft sensitivities to lateral and vertical profiles.
AIAA Journal of Guidance and Control, 4 (Nov.–Dec. 1981).



Larry Thomas Brewster (S'70—M'71) was born in Brownwood, Tex., on September 1, 1941. He received the B.S.E.E. degree from New Mexico State University, Las Cruces, in 1966; the M.S.E.E. degree from the University of Missouri-Columbia in 1978; and the Ph.D. degree in computer science from the University of Missouri-Rolla in 1987.

He worked in telemetry research and development at White Sands Missile Range, New Mex. and as a staff engineer for Hewlett Packard in Las Cruces, New Mex. He is currently employed as a pilot for Trans World Airlines and an Adjunct Assistant Professor at the University of Missouri-Rolla.

Dr. Brewster is a member of the Association for Computing Machinery, Eta Kappa Nu, Upsilon Pi Epsilon, and Sigma Xi.



Paul D. Stigall (S'60—M'62—SM'78) received the B.S. degree in electrical engineering from the University of Missouri-Rolla in 1962 and the M.S. and Ph.D. in electrical engineering from the University of Wyoming in 1965 and 1968.

Paul D. Stigall is Professor of electrical engineering and computer science at the University of Missouri-Rolla, which he joined in 1970. Before joining the university, Stigall worked for McDonnell Douglas, the Navy Electronics Laboratory, and the Collins Radio Company. He also was an instructor at the University of Wyoming. His research interests include computer architecture and systems, digital signal processing, microprocessor and minicomputer application, digital circuits, and fault-tolerant computing. He is the author of several technical publications and the principal investigator on several research projects.

A registered Professional Engineer in Missouri, he is a member of the Association for Computing Machinery, Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.