AIAA 98-0302
Constant Dynamic Pressure Trajectory Simulation with POST

J. R. Olds
I. A. Budianto
Georgia Institute of Technology
Atlanta, GA

36th Aerospace Sciences Meeting & Exhibit

January 12 -15, 1998 / Reno, NV

Constant Dynamic Pressure Trajectory Simulation with POST

Dr. John R. Olds[†]
Irene A. Budianto[*]
Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332-0150

ABSTRACT

Future space transportation vehicles may well rely on high speed airbreathing propulsion (ramjets and scramjets) to supply much of their motive power. Because of the tradeoff relationship between engine thrust and vehicle airframe weight, ascent trajectories are typically simulated using a constant dynamic pressure phase during airbreathing acceleration. That is, dynamic pressure is increased to benefit vehicle thrust up to some fixed limit imposed by the vehicle structure. The constant dynamic pressure portion of the trajectory typically begins around Mach 2 or 3 and continues to the maximum airbreathing Mach number or until some convective aeroheating limit is reached.

This paper summarizes comparative research on three candidate guidance methods suitable for simulating constant dynamic pressure trajectories. These are generalized acceleration steering, linear feedback control, and cubic polynomial control. All methods were implemented in POST (Program to Optimize Simulated Trajectories) — an industry standard trajectory simulation code. Both quantitative and qualitative comparisons of these methods (i.e. in terms of computer processing time, number of required iterations for convergence, sensitivity to quality of initial values, accuracy and program robustness) are presented. Of the three methods, the linear feedback control approach is found to be the most efficient and robust, with good accuracy.

[†] - *Assistant Professor, School of Aerospace Engineering, Senior member AIAA.*

[*] - *Palace Knight Fellow, School of Aerospace Engineering, Student member AIAA.*

NOMENCLATURE

c_{max}	upper limit for an inequality constraint
C_i	additional trajectory constraints
C_θ	optimization indicator
CPU	central processing unit
GAS	generalized acceleration steering
I_{sp}	specific impulse (sec.)
Kdg	displacement gain in linear feedback guidance equation
Krg	rate gain in linear feedback guidance equation
q	dynamic pressure (psf)
$\dot{q}$	time derivative of dynamic pressure (psf/s)
LFC	linear feedback control steering
LH2	liquid hydrogen
LOX	liquid oxygen
P2	weighted constraint error from POST
POST	Program to Optimize Simulated Trajectories
RBCC	rocket-based combined-cycle propulsion
S_{ref}	reference surface area (ft^2)
T/W	vehicle thrust-to-weight ratio
t_f	final time, constant dynamic pressure phase
t_i	initial time, constant dynamic pressure phase
α	angle of attack (deg.)
β_n	angle of attack polynomial coefficients (n = 0, 1, 2, 3)

INTRODUCTION

Background

Advanced hypersonic airbreathing propulsion is of significant interest for possible application to future, low cost, reusable space transportation vehicles. As first proposed by Goddard in 1932, airbreathing engines can be used to accelerate the vehicle through the atmosphere on its ascent to orbit [1]. The main advantage over conventional rocket

engines is high fuel efficiency; airbreathing engines can provide up to ten times the effective specific impulse of rockets at the lower altitudes mainly because of their utilization of atmospheric oxygen. The disadvantages of airbreathing engines include the added complexity and heavy flowpath requirement — resulting in low thrust-to-engine weight ratios. Furthermore, they can operate only in the atmosphere and must be supplemented by rocket propulsion for in-space applications.

Rocket-based combined-cycle (RBCC) propulsion tries to combine the "best of both worlds." Elements of rocket and airbreathing propulsion are integrated into one engine, eliminating redundant components (i.e., pumps, etc.) that would be found in combination propulsion or rocket + airbreathing propulsion systems. RBCC engines are capable of multi-mode operation (i.e. ejector, ramjet, scramjet, and pure rocket modes) resulting in a high trajectory-averaged fuel efficiency [2].

Motivation

Trajectory simulation and analysis is a key element in conceptual launch vehicle design studies. Performance characteristics are obtained from the analysis to determine the feasibility and viability of a design concept for a particular mission. Trajectory problems are ordinarily formulated in terms of optimizing a performance function (e.g. maximum payload or minimum fuel) with a given set of constraints to be satisfied (e.g. target orbit, limits on angle of attack, etc.).

Dynamic pressure plays a particularly important role in the ascent trajectory of airbreathing vehicles. High dynamic pressure environments can cause excessive structural loading and unacceptable aerodynamic heating. A limit on the maximum dynamic pressure allowable must therefore be imposed for this type of vehicles. On the other hand, ramjet/scramjet thrust performance is directly related to dynamic pressure — that is, higher dynamic pressure yields higher thrust. As a result of this trade-off, it is quite common to simulate the ramjet/scramjet segments of an airbreathing vehicle's ascent trajectory as a constant dynamic pressure path.

The thrust produced by RBCC engines is not constant but varies with altitude, Mach number and dynamic pressure. Coupled with changing vehicle aerodynamics and weight, the vehicle's progress along this constant dynamic pressure boundary path can be complex and difficult to determine.

Problem Formulation

Computer-based trajectory optimization codes are typically employed to determine acceleration, ascent time, propellant consumption, angle of attack, etc. The Program to Optimize Simulated Trajectories (POST) is a widely available tool created by NASA and Lockheed-Martin and has been commonly used to solve various trajectory optimization problems for both atmospheric and orbital vehicles [3]. It is a generalized event-oriented trajectory simulation program capable of optimizing a user-specified performance function subject to certain constraints (dependent variables) by determining the values of the control (independent) variables. The user structures the trajectory by a logical sequence of events and enters the model of the vehicle and the necessary constraints. The user is also required to specify the parameters and control variables.

Employing three-degree-of-freedom POST for the ascent trajectory analysis of an airbreathing vehicle requires some modification to the basic program to simulate a constant dynamic pressure (q) path. Three guidance methods are studied and applied to three different representative vehicle models with varied mission profiles. Furthermore, since control variable initial guesses play an important role in any numerical computations, their influence on the effectiveness of these guidance methods is studied as well. The results of these investigations and their computational requirements are compared.

DESCRIPTION OF TEST CASES

Maglifter

As part of the NASA Highly Reusable Space Transportation (HRST) program, an early feasibility study was conducted for an advanced reusable launch vehicle using RBCC propulsion (LOX/LH2 supercharged ejector ramjet) and a single LOX/LH2

tail rocket. A magnetically-levitated sled and track system (the *Maglifter*) provides an initial velocity of 800 fps. Note that in this paper, this entire sled, track, and orbital vehicle concept will be referred to as the *Maglifter* concept. The orbital vehicle leaves the track horizontally and operates in ejector mode to Mach 2 where it begins to transition to fan-ramjet mode. It intercepts a 2000 psf dynamic pressure boundary at Mach 3 and flies along this constant *q* path until Mach 6. The vehicle then transitions to rocket mode for final acceleration into low earth orbit.

The trajectory simulation problem involves maximizing the vehicle burnout weight. The mission of the *Maglifter* is to deliver 20 klb of payload to 100 nmi circular orbit at 28.5° inclination angle from Cape Canaveral, Florida. The independent variables consist of several relative pitch angles to control vehicle attitude before and after the constant *q* phase. During the constant *q* phase, the vehicle is commanded by angle of attack. The vehicle parameters summarized in Table 1 are obtained from previous work conducted by the author [4].

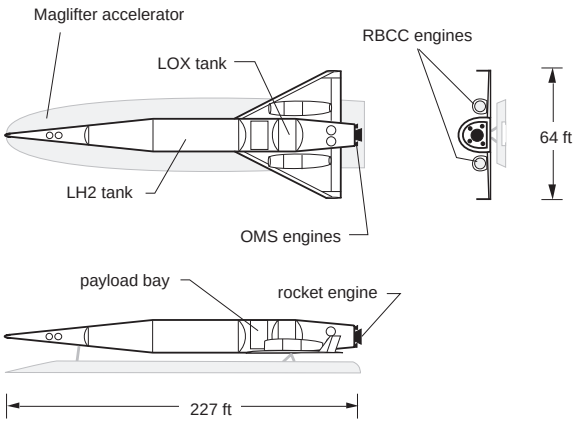

Figure 1. *Maglifter* configuration.

Cerberus I

The *Cerberus* project involves the conceptual design of a low-cost, two-stage-to-orbit (TSTO), horizontal-take-off-and-landing launch vehicle, which utilizes LOX/LH2 RBCC engines as the booster propulsion and liquid rockets to accelerate the waverider upper stage from the point of separation to its final orbit. Graduate students at the Georgia Institute of Technology, School of Aerospace Engineering, undertook the project as an exercise in employing multidisciplinary design methodology in a 'design for cost' environment.

Maximum booster airbreathing Mach number, staging Mach number, and payload delivery mass were identified through Quality Function Deployment [5] to be system-level variables that can significantly affect how the overall vehicle design meets the customer requirements outlined in a fictitious request for proposal. Using Design of Experiments methods, ten different design variable combinations were investigated, resulting in ten different vehicle designs.

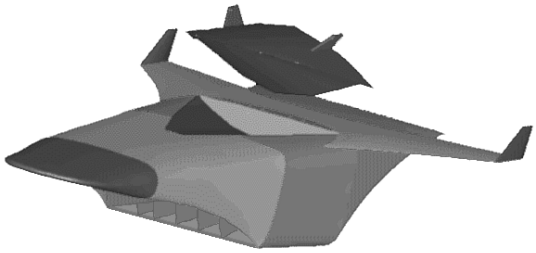

Figure 2. *Cerberus* vehicle at separation.

The vehicle referred herein as *Cerberus I* is one of the cases studied. Listed in Table 1 are its sized characteristics. *Cerberus I* airbreathes up to Mach 6 with a supercharged ejector ramjet and stages at Mach 8 to deliver 10 klb of payload to Space Station orbit (220 x 220 nmi. x 51.6° inclination). The mated vehicle takes off horizontally and intercepts a 1500 psf dynamic pressure boundary at Mach 3. It continues along this dynamic pressure constraint in ramjet mode to Mach 6, gaining both velocity and altitude. The vehicle then performs a pull-up maneuver and briefly transitions to rocket mode. Separation of the two stages occurs at Mach 8. The trajectory simulation continues to follow the upper stage on its ascent to reach 50 x 220 nmi x 51.6° insertion orbit. The booster returns to the launch site in ramjet mode.

Unlike the *Maglifter* case, no optimization variable is specified. The trajectory problem thus only involves meeting trajectory constraints. Pitch angles and angle of attack are the control variables involved in this trajectory problem.

Cerberus II

Another case studied in the *Cerberus* project is a vehicle with maximum airbreathing speed of Mach 9.75. This design uses supercharged ejector scramjet engines (transition from ramjet to scramjet mode of operation occurs around Mach 6). The mated vehicles flies a constant q boundary to Mach 9.75. It then performs a pull-up maneuver as it briefly transitions to rocket-mode to accelerate to a Mach 10.25 staging condition. The upper stage continues its flight to deliver a payload of 10 klb to the International Space Station orbit. This vehicle is referred here as *Cerberus II*. The maximum dynamic pressure constraint (1500 psf), target conditions and control variables are identical to those of *Cerberus I*. Furthermore, the booster external configuration (aerodynamic coefficients) of the two *Cerberus* vehicles are the same. However, this vehicle has been photographically resized to meet the new propellant requirements. The *Cerberus II* waverider upper stage was reconfigured for staging at Mach 10.25.

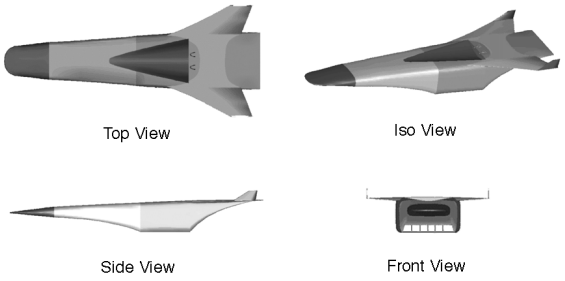

Figure 3. *Cerberus* booster configuration.

Table 1: Summary of test vehicle parameters

Vehicle Parameters	*Maglifter*	*Cerberus I*	*Cerberus II*
Upper Stage:			
Gross Wt (lbs)	----	136,740	107,400
Initial T/W	----	1.1	1.1
Isp_{vac} (s)	----	467	467
Sref (ft^2)	----	1471	1616
Overall:			
Gross Wt (lbs)	997,640	498,900	451,800
Initial T/W	0.55	0.60	0.60
Rocket Mode Isp_{vac} (s)	455	463	463
Sref (ft^2)	3606	3440	4150

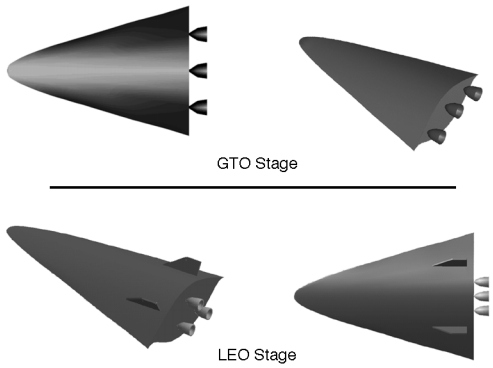

Figure 4. *Cerberus* waverider upper stage.

CANDIDATE GUIDANCE METHODS

Generalized Acceleration Steering (GAS)

An instantaneous rootfinding iterative method was implemented in attempt to hold the dynamic pressure to the specified value by changing angle of attack along the constant q path. A new subroutine was added to be called by the guidance routine in POST to determine the angle of attack, α, that results in the specified $\dot{q}$ (time derivative of dynamic pressure) at each increment of time. For a constant q path, $\dot{q}$ was specified to be zero. That is, at each time step, an α was numerically determined to ensure $\dot{q} = 0$.

Brent's algorithm (bisection) was used to solve the rootfinding problem within the given α interval. This interval also acts as the allowable range for the angle of attack in the constant q phase. A root tolerance of 10^{-6} was used for all trajectories. A rootfinding method, as with any other numerical method, doesn't always guarantee convergence. It often fails and finds no solution. In these cases, the angle of attack is incremented by a fixed $\Delta\alpha$ if the q value is too high or decremented if q is too low. The routine can also create noise when multiple solutions are encountered, causing a relatively large discontinuity in α between time steps.

The rootfinding subroutine further requires a separate function to evaluate the resulting $\dot{q}$ for different trial values of α within the bisection routine. This function is utilizing existing POST subroutines in its computation of atmospheric

parameters, vehicle aerodynamics, equations of motion, etc.

With this option, the only constraints involving the constant q phase are placed at the beginning of this segment to ensure that the vehicle enters it at the desired q and $\dot{q}$:

$$C_1 = q(t_i) - q_D = 0 \quad (1)$$

$$C_2 = \dot{q}(t_i) - \dot{q}_D = 0. \quad (2)$$

q_D is the dynamic pressure limit and $\dot{q}_D$ is specified to be zero in this study. t_i is the initial time at the beginning of the constant q phase.

A significant drawback of the GAS scheme is that the cost in terms of execution time is expected to be expensive. In this case, the rootfinding problem must be solved at each time step during the constant q phase, requiring a high amount of central processor time.

Linear Feedback Control (LFC)

The second guidance method investigated uses an updating approach. At each integration step, the angle of attack is adjusted according to the errors in q and $\dot{q}$. A new subroutine was created, containing the following set of linear feedback steering equations:

$$\alpha_{new} = \alpha_{old} + Kdg(q - q_D) + Krg(\dot{q} - \dot{q}_D) \quad (3)$$

$$\alpha_{new} = \alpha_{old} + 0.5 \times (q - q_D) \quad (4)$$

$$\alpha_{new} = 0.8 \times \alpha_{old} \quad (5)$$

Kdg and *Krg* are displacement and rate feedback control gains, respectively. Equation 3 is used for small errors, equation 4 is used when q is well above q_D, and equation 5 is used when q is well below q_D. Several cases have been observed to be solvable by steering equation 3 alone. Without the equations for conditions 4 and 5, greater gains are needed and the method is observed to reach convergence faster. However, the use of three separate formulations for the different possibly encountered conditions increases program robustness and can avoid large swings in α.

The desired dynamic pressure and its derivative are q_D and $\dot{q}_D$, respectively. To obtain a constant q trajectory, $\dot{q}_D$ is set to zero. α_{old} is always set to the angle of attack *at the last time step*. For users with prior experience with a particular trajectory, the linear feedback gains *Kdg* and *Krg* can be specified in the input file as constants. In this study, however, these constant gains are treated as independent variables, giving POST the control to determine their values.

Since the solution during the constant q segment is non-unique (i.e. several values for *Kdg* and *Krg* are feasible but variations about the desired q may be unacceptably large in some cases), an additional constraint to C_1 and C_2 is needed. A reasonable choice is an inequality constraint C_3 limiting the square error of dynamic pressure, integrated over the entire constant q phase:

$$C_3 = \int_{t_i}^{t_f} (q - q_D)^2 \, dt - c_{\max} \le 0 \quad (6)$$

The duration of the constant q phase and the value of q_D are taken into account in determining a reasonable value for c_{max}. A vehicle flying a higher q boundary for a longer length of time will typically necessitate a greater c_{max} value. Upper and lower side constraints for α are also imposed in the subroutine:

$$\alpha_{\min} \le \alpha_{new} \le \alpha_{\max}. \quad (7)$$

As with the GAS method, α remains constant over each time interval *dt*. Therefore, the linear feedback control (LFC) approach requires small integration interval with small gains to allow the necessary updates in α to hold the vehicle along the constant q boundary. This increases CPU time. However, the execution time requirement per iteration is expected to be significantly less expensive than the GAS method.

Cubic Alpha Polynomial (cubic α)

Several steering options to control the vehicle attitude during the trajectory simulation are already built into POST. One of these commands the aerodynamic angles as a third-order polynomial and allows the user to specify both the coefficient values and the independent argument variables. The cubic alpha polynomial results in a smooth function of α that allows the user to increase the time step during

the constant q segment and thus reduce execution time (CPU time).

Several different possible arguments for the cubic alpha polynomial were investigated. α as a polynomial function of variables such as altitude, velocity, weight, time, Mach number, and their delta's and derivatives were tried and found to be ineffective in flying this trajectory. In each case, the length of the constant q segment made it impossible to determine a single set of coefficients that would result in small steering errors. However, one variable that has been found to work well is the error of dynamic pressure with respect to the desired value. This error is not a regular POST variable and its computation was programmed in the special calculation subroutine.

The resulting cubic alpha polynomial is the following:

$$\alpha = \beta_0 + \beta_1(q - q_D) + \beta_2(q - q_D)^2 + \beta_3(q - q_D)^3 \quad (8)$$

with β_0 set to the value of $\alpha(t_i)$, or angle of attack at the beginning of the phase, which does not change during the constant q segment (unlike the α_{old} in the LFC method above). In this formulation, β_1 is similar to a displacement feedback gain (*Kdg*), and β_2 and β_3 represent feedbacks based on higher order terms. In this study, the coefficients β_1, β_2, and β_3 are declared to be independent variables whose values are to be determined by POST. As in the linear feedback approach, an inequality constraint of the integrated square error of dynamic pressure must be added to the trajectory problem (equation 6).

There is one possible objection to the utility of this α equation. If the error is zero throughout, that is, if the vehicle flies a trajectory that follows the constant q boundary perfectly, then according to this formula, the solution is a constant angle of attack given by the *initial* angle of attack constant, β_0. Since this solution is unlikely for a vehicle with changing thrust, mass, altitude, velocity, etc., the method must work by forcing a slight error to be generated in order to modify β_0 and obtain the required α profile. As a result, it cannot be expected to produce extremely small values of the integrated square of the dynamic pressure error.

RESULTS AND DISCUSSION

Fair 'apples-to-apples' comparisons of the results proves difficult since the application of these guidance methods to the three test vehicle trajectories requires different numbers of constraints and involve different numbers of independent variables. Therefore, within each test case, consistencies are maintained as much as possible. The same integration time step, for example, is used by all three methods. All of the computations are executed on a Silicon Graphics UNIX workstation with operating system IRIX 5.3 running on an R5000/150 MHz IP22 CPU board.

For each vehicle case, the guidance methods are tested using two different sets of starting conditions for the independent variable values. The first set consisted of 'good' initial guesses, that, in particular, produce q close to q_D and $\dot{q}$ close to zero at the start of the constant q segment. This implies good pitch angle control leading up to the constant q phase. The trajectory simulations were observed to be extremely sensitive to these entry conditions at time = t_i.

The second set of independent variables contained 'bad' initial values that were rather inaccurate, requiring more iterations to the solution, or even leading to nonconvergence. A good measure of robustness for any of these methods is how well it can recover from a poor entry conditions into the constant q phase caused by 'bad' initial guesses on early pitch control values. Note that any extra controls (*Kdg* and *Krg* for LFC and β_1, β_2, and β_3 for cubic α) were still given 'good' guesses for initial values for fair comparisons.

Each of the three sample vehicle trajectories was tested against each of the three candidate guidance methods and both 'good' and 'bad' entry conditions into the constant q segment of their trajectories (for a total of 18 cases). The results of these tests are summarized in Table 2 and Table 3.

The constraint errors listed in the tables are the *P2* values computed by POST from the weighted errors in the dependent variables. A *P2* value of less than one indicates that all constraints are satisfied to within their specified tolerances and the trajectory is said to be targeted. The optimization indicator is the

Table 2: Summary of results with 'good' initial values.

	Maglifter			*Cerberus I*			*Cerberus II*		
	GAS	***LFC***	***Cubic α***	***GAS***	***LFC***	***Cubic α***	***GAS***	***LFC***	***Cubic α***
No. of Controls, u	7	9	10	9	11	12	9	11	12
No. of Constraints	6	7	7	7	8	8	7	8	8
Total CPU Time (s)	714.889	293.168	337.572	609.100	482.516	317.286	386.154	151.629	248.461
No. of Iterations	3	3	3	2	3	2	1	1	1
Constraint Error, P2	0.39913	0.50096	0.40071	0.01457	0.41523	0.06006	0.51915	0.48761	0.49673
Constant q Error :									
C_3 (lbs^2-s/ft^4)	18.228	20.312	18.429	2.927	10.045	10.229	195.788	18.338	76.646
Optimization									
Indicator, C_θ	89.958°	89.954°	89.952°	----	----	----	----	----	----
Optimal Value:									
$W_{burnout}$ (lbs)	162,890	162,887	162,890	----	----	----	----	----	----
Initial Values for	----	0.001573 (*Kdg*)	3.613994 (β_1)	----	0.008350 (*Kdg*)	3.688301 (β_1)	----	0.005000 (*Kdg*)	3.200000 (β_1)
Extra Control	----	0.733400 (*Krg*)	0.293002 (β_2)	----	0.200000 (*Krg*)	-0.043143 (β_2)	----	0.150000 (*Krg*)	0.025000 (β_2)
Variables	----	----	0.329083 (β_3)	----	----	0.020121 (β_3)	----	----	0.003000 (β_3)
Final Values for	----	0.001583 (*Kdg*)	3.570396 (β_1)	----	0.008311(*Kdg*)	4.006297 (β_1)	----	0.005000 (*Kdg*)	3.200207 (β_1)
Extra Control	----	0.764399 (*Krg*)	0.292934 (β_2)	----	0.200598 (*Krg*)	-0.043130 (β_2)	----	0.149999 (*Krg*)	0.025000 (β_2)
Variables	----	----	0.329061 (β_3)	----	----	0.020121 (β_3)	----	----	0.003000 (β_3)

Table 3: Summary of results with 'bad' initial values.

	Maglifter			*Cerberus I*			*Cerberus II*		
	GAS	***LFC***	***Cubic α***	***GAS***	***LFC***	***Cubic α***	***GAS***	***LFC***	***Cubic α***
No. of Controls [u]	7	9	10	9	11	12	9	11	12
No. of Constraints	6	7	7	7	8	8	7	8	8
Total CPU Time (s)	1034.617	845.654	1609.752	683.451	755.813	**6592.125**	2106.195	420.745	3774.704
No. of Iterations	4	9	19	2	5	**36**	3	2	18
Constraint Error, P2	0.01961	0.19278	0.29832	0.12319	0.02079	**48.408**	0.34194	0.22485	0.00305
Constant q Error :									
C_3 (lbs^2-s/ft^4)	19.636	34.819	35.005	8.434	4.519	16.957	78.124	106.911	100.276
Optimization									
Indicator, C_θ	**89.800°**	89.958°	90.000°	----	----	----	----	----	----
Optimal Value:									
$W_{burnout}$ (lbs)	162,803	162,862	163,017	----	----	----	----	----	----
Initial Values for	----	0.001073 (*Kdg*)	0.068302 (β_1)	----	0.008350 (*Kdg*)	0.377923 (β_1)	----	0.005000 (*Kdg*)	0.005779 (β_1)
Extra Control	----	0.733400 (*Krg*)	0.000535 (β_2)	----	0.200000 (*Krg*)	-0.000243 (β_2)	----	0.150000 (*Krg*)	-0.000055 (β_2)
Variables	----	----	0.000006 (β_3)	----	----	0.000002 (β_3)	----	----	0.000001 (β_3)
Final Values for	----	-0.000911 (*Kdg*)	2.695586 (β_1)	----	0.008007 (*Kdg*)	5.514056 (β_1)	----	0.005011 (*Kdg*)	3.599330 (β_1)
Extra Control	----	0.787696 (*Krg*)	0.001411 (β_2)	----	0.213421 (*Krg*)	-0.000241 (β_2)	----	0.150041 (*Krg*)	-0.001067 (β_2)
Variables	----	----	0.000010 (β_3)	----	----	0.000002 (β_3)	----	----	0.000003 (β_3)

C_θ angle in POST and is the angle between the gradient of the objective function vector and the plane tangent to the intersection of the linearized active constraints. If C_θ is greater than or equal to 89.9°, the objective variable is considered optimized at the current values for the independent variables. The objective function (maximum burnout weight) is given in the tables for the *Maglifter* case. Recall that neither of the *Cerberus* vehicles involve optimization of the trajectory, just targeting of all constraints.

Results for 'Good' (Near-Solution) Initial Guesses

Good initial guesses on the control variable values allow POST to reach convergence rapidly. All three methods perform well and give similar solutions for each vehicle case (i.e. *P2* always less than one, very few POST iterations required, similar optimum burnout weights for the *Maglifter* trajectory). As shown in Table 2, the linear feedback method generally requires the least runtime per iteration. As expected, the GAS method was found to be the most time consuming.

Samples of the resulting dynamic pressure and altitude profiles along with the angle of attack and relative pitch angle histories are shown in Figure 5 through 10 for the three test cases. The particular cases from which the graphs are taken are identified in the figure caption.

The integrated square error C_3 measures how well the methods can maintain the constant dynamic pressure. The error was constrained to be less than 20 lb^2-s/ft^4 for the *Cerberus I* case, and less than 100 lb^2-s/ft^4 for the *Cerberus II* case. C_3 was calculated, but not used as a constraint in the *Maglifter* case. LFC produces comparable, if not smaller errors in all three vehicle cases. In the *Maglifter* case, all three approaches produce a good optimal value and *P2*

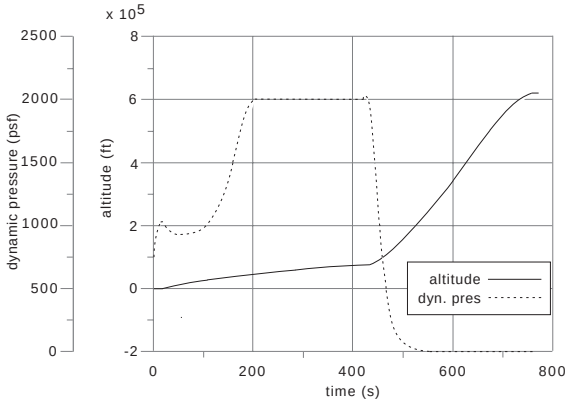

Figure 5. Dynamic pressure and altitude profiles for optimized *Maglifter* trajectory (Cubic α).

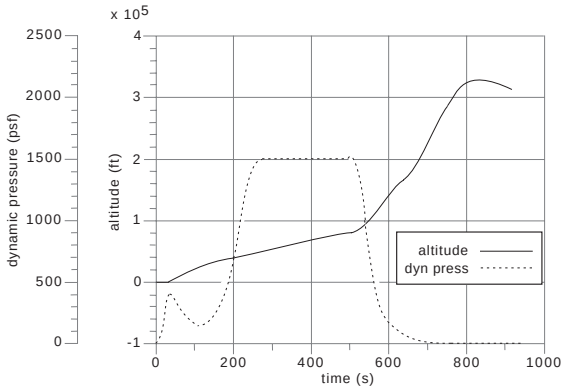

Figure 7. Dynamic pressure and altitude profiles for *Cerberus I* final trajectory (GAS).

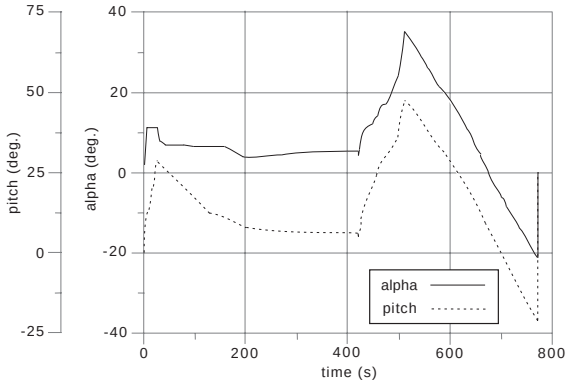

Figure 6. Angle of attack and relative pitch angle histories for optimized *Maglifter* trajectory (Cubic α).

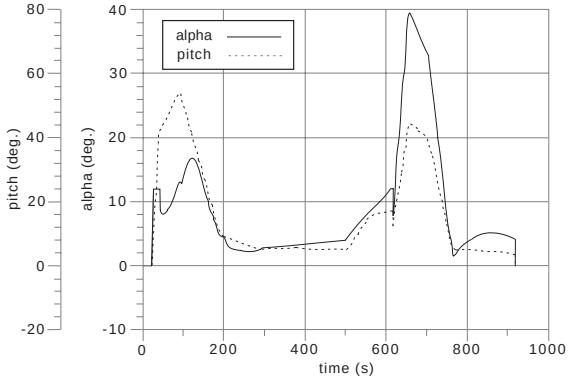

Figure 8. Angle of attack and relative pitch angle histories for *Cerberus I* final trajectory (GAS).

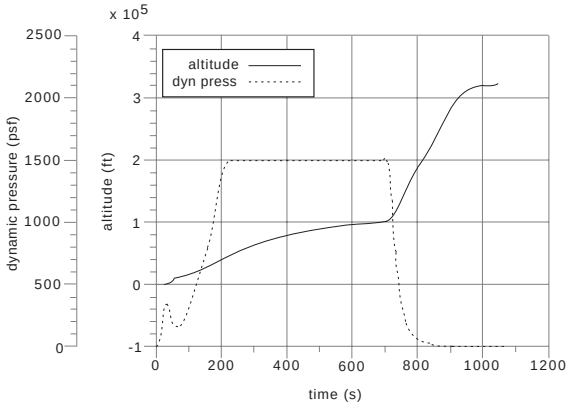

Figure 9. Dynamic pressure and altitude profiles for *Cerberus II* final trajectory (GAS).

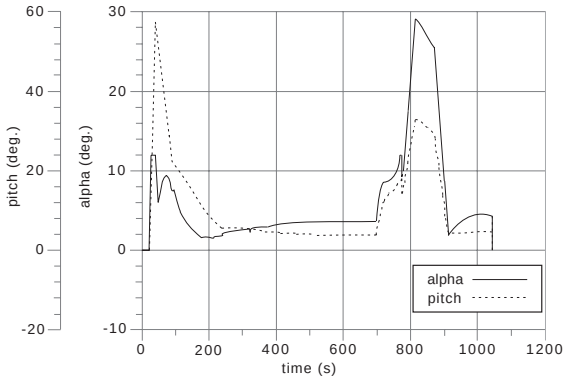

Figure 10. Angle of attack and relative pitch angle histories for *Cerberus II* final trajectory (GAS).

condition. The cubic polynomial control is the fastest among the three methods to converge to the solution for *Cerberus I* case, requiring fewer iterations than LFC and less CPU time per iteration in comparison to GAS. However, in the computation of the *Cerberus II* trajectory, this method requires significantly longer execution time than the linear feedback approach. This may be as a result of the longer duration of the constant *q* phase.

Results for 'Bad' (Off-Solution) Initial Guesses

As previously discussed, constant dynamic pressure trajectory simulations in POST are highly sensitive to the flight conditions entering the constant *q* segment. 'Bad' initial guesses for the pitch angle control variables were used to test the robustness of the three candidate guidance methods — that is, their ability to recover from a poor starting point.

As shown in Table 3 for the *Maglifter* case, an optimal solution is obtained by both the linear feedback and cubic α methods with comparable accuracies. LFC requires significantly fewer iterations and thus less total CPU time than the cubic α method. To obtain a successful nominal trajectory using the cubic α control, the initial values for the polynomial coefficients must be very small to compensate for the large errors. The GAS method, on the other hand, satisfied all the constraints but slightly failed to meet the (somewhat arbitrary) C_θ 'optimized' criteria of 89.9°. Manually restarting the optimization from the previous solution may result in an optimum solution, but the total CPU time for the GAS method is expected to remain high.

Convergence to the solution for *Cerberus I* was not reached using the cubic α scheme. This is indicated by a *P2* constraint error greater than one. The GAS method takes slightly less total CPU time than LFC, but the runtime *per iteration* of GAS is still higher than that of LFC. GAS requires fewer iterations to the solution perhaps because of the fewer number of constraints involved. The trajectory problem for *Cerberus II* was found to be solvable by all three methods, but LFC was found to require the least amount of CPU time.

Figure 11 to Figure 13 plot the angle of attack and dynamic pressure profiles, illustrating the nominal or *iteration 1* trajectories obtained by implementing the guidance methods to the *Cerberus II* test case. Iteration 1 results are initial trajectories derived from the 'bad' initial guesses before the independent variable values are optimized to bring the trajectory into convergence. These figures illustrate the different ways the three methods work to solve the trajectory problem. In each one, the vehicle enters the constant *q* phase at approximately 1300 psf. The generalized acceleration steering method maintains this *q* value by finding the angle of attack that results in zero $\dot{q}$. Subsequent iterations will raise the "entry" *q* to 1500 psf to satisfy the constraints in equations 1 and 2.

The linear feedback control approach, on the other hand, slowly steers the vehicle up to the target *q* path of 1500 psf (even for the iteration 1 trajectory) in order to eliminate the displacement errors in equation 3.

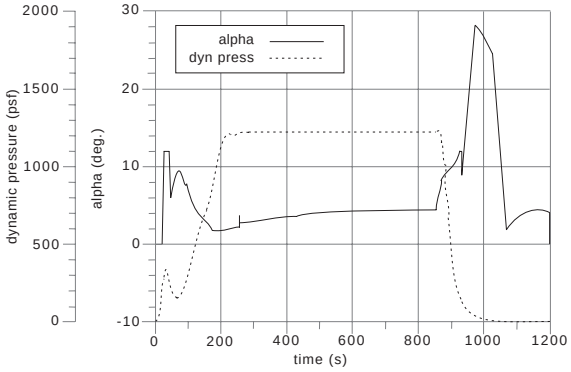

Figure 11. Nominal (Iteration 1) α and q profiles for *Cerberus II* from poor guesses with GAS method.

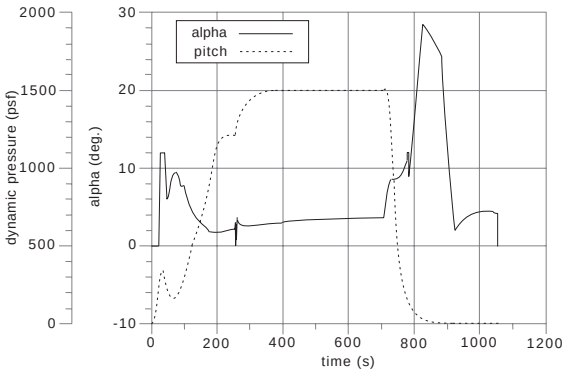

Figure 12. Nominal (Iteration 1) α and q profiles for *Cerberus II* from poor guesses with LFC method.

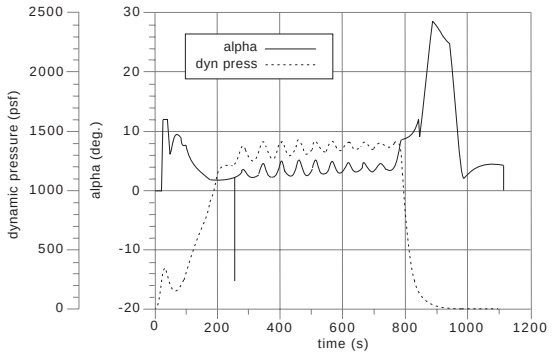

Figure 13. Nominal (Iteration 1) α and q profiles for *Cerberus II* from poor guesses with cubic α method.

The cubic α nominal path is shown to oscillate up to the desired *q* boundary. It has been observed from these test cases that with large errors in entry *q*, small cubic polynomial coefficients are required to obtain usable nominal trajectory. For both *Maglifter* and *Cerberus II*, the coefficients β_1 and β_2 change values quite significantly between the first guess and the final solution as shown in Table 3. However, it was apparent that the trajectory problem is not sensitive to changes in the cubic term. In fact, β_3 need not be defined as a control variable and its value can simply be set to a small constant, even zero.

Based on these examples, the linear feedback control approach proves to be the most robust of the three methods investigated. LFC was also found to be the most efficient, typically requiring fewer iterations than the cubic polynomial approach and less execution time per iteration than the generalized acceleration steering. Its accuracy in maintaining the constant *q* path is comparable to those of the other methods.

CONCLUSION

Summary

This paper presents the study of different guidance methods implemented in POST to simulate constant dynamic pressure ascent trajectories for airbreathing vehicles. Among the three methods investigated, the linear feedback control (LFC) approach gives the best overall performance in terms of execution time, robustness, and accuracy.

The results of both 'good' and 'bad' sets of initial guesses show LFC to be quite robust, solving to convergence efficiently and with good accuracy. Presented are a few cases where generalized acceleration and cubic α steering methods failed, while LFC succeeded in reaching the solution. Furthermore, in all the trajectory problems studied, the required computer processing time per iteration for LFC is consistently the lowest among the three methods.

Future Considerations

In this study, the only guidance control variable investigated for the constant *q* phase was angle of attack. Another possible guidance scheme that can be implemented in POST to achieve the desired trajectory is the 'stick and throttle' command [6]. This involves applying thrust control by throttling

the engine, in addition to commanding α. The GAS or LFC methods might be applied to this guidance scheme with the necessary modifications.

An option for the GAS approach that may improve its effectiveness in solving the constant q trajectory problems is to input a table of desired $\dot{q}_D$ versus current q values. $\dot{q}_D$ would be selected to be negative or positive depending on q to influence the dynamic pressure toward the desired target and prevent numerical drift associated with simply requiring $\dot{q}_D$ to be zero starting at an initial q. This 'scheduling' can be used to achieve the same effect observed with LFC in Figure 12, such that if the vehicle is initially on the wrong q boundary, it is steered toward the desired value, instead of maintaining the current path.

Only horizontal take-off vehicles have been considered in this study. No vertical take-off cases have been included. To better understand the effectiveness and efficiencies of the three guidance methods, they should be tested on this type of vehicles as well.

AKNOWLEDGEMENTS

This work was partially supported by the U. S. Air Force Palace Knight Graduate Fellowship program.

The authors wish to thank the graduate student members of the *Cerberus* development team at Georgia Tech for providing the vehicle weights and performance data for that concept. They are Irene Budianto, Peter Bellini, Anurag Gupta, Mike Lee, Rob LeMoyne, Brian Palmintier, and Saumil Shau.

BIBLIOGRAPHY

1. Goddard, R. H.. "A New Kind of Turbine Rocket Plane for the Upper Atmosphere," *Scientific American*, Vol. 146, March 1932, p. 148-149.

2. William J. D. Escher, ed. *The Synerjet Engine: Airbreathing/Rocket Combined-Cycle Propulsion for Tomorrow's Space Transports*, SAE PT-54, Society of Automotive Engineers, Warrendale, PA, 1997.

3. Brauer, G. L.; Cornick, D. E. and Stevenson, R. *Program to Optimize Simulated Trajectory (POST)*, Final Report for NASA contract NAS1-18147, Martin-Marietta Corp., September 1990.

4. Olds, John R. "Achieving Low Cost Space Transportation," North Carolina State University, HRST Phase I, Final Report, NASA MSFC Letter P.O. H-20753D, July 1995.

5. Hauser, John R. and Clausing, Don. "The House of Quality," *Harvard Business Review*, May-June 1988, pp. 63-73.

6. Kauffman, H. G.; Grandhi, R. V.; Hankey, W. L. and Belcher, P. J. "Improved Airbreathing Launch Vehicle Performance with the Use of Rocket Propulsion," *Journal of Spacecraft*, Vol. 28, No. 2, March-April 1991, pp. 172-178.