JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 34, No. 3, May–June 1997

Six-Degree-of-Freedom Trajectory Targeting and Optimization for Titan Launch Vehicles

Prabhakara P. Rao,* Brian M. Sutter,† and Philip E. Hong‡
Lockheed Martin Astronautics, Denver, Colorado 80201

An important part of launch vehicle mission design is the trajectory optimization to maximize vehicle performance for a specified payload weight. For a Titan IV mission to park orbit or beyond, the optimization is usually performed using three-degree-of-freedom models. Additional propellant is reserved to account for unmodeled rotational vehicle dynamic characteristics. With recent advances in computer technology and optimization techniques, it is possible to use higher-fidelity models to minimize the differences between three and six degrees of freedom and simplify launch-day operations. A six-degree-of-freedom trajectory optimization program is developed for a typical boost flight from liftoff to park orbit inject. Flight software guidance and autopilot logic are simplified to achieve a fast, yet accurate trajectory simulation. The optimization logic uses numerical integration of analytically derived partial derivatives to improve accuracy, stability, and computational speed. Results compare well with other high-fidelity trajectory simulations.

Nomenclature

$\boldsymbol{A}_{AB}, \boldsymbol{A}_{TB}$	= aerodynamic and thrust accelerations in the body frame, ft/s^2
$\boldsymbol{c}(c_1, c_2, \ldots, c_m)$	= vector of m equality and inequality constraints
$\boldsymbol{e}(e_0, e_1, e_2, e_3)$	= quaternion
F	= scalar objective function
$\boldsymbol{f}(f_1, f_2, \ldots, f_k)$	= dynamic functions
$\boldsymbol{G}$	= control variational vector
$\boldsymbol{G}_I$	= gravitational acceleration in the Earth-centered inertial frame, ft/s^2
h_x, h_y, h_z	= angular momentum vector components, ft^2/s
$\boldsymbol{M}_{AB}, \boldsymbol{M}_{TB}$	= external moments acting on the vehicle center of gravity due to aerodynamic and thrust forces, lb · ft
R	= radius vector magnitude, ft
$\boldsymbol{r}(x, y, z)$	= position vector, ft
t_a	= control activation time, s
t_d	= control deactivation time, s
$\boldsymbol{u}(u_1, u_2, \ldots, u_n)$	= vector of n optimization or control variables
V	= velocity vector magnitude, ft/s
$\boldsymbol{v}(v_x, v_y, v_z)$	= velocity vector, ft/s
W_G	= gross vehicle weight, lb
$\dot{W}_{Pi}$	= propellant flow rate for the ith engine, lb/s
$\dot{W}_{Sj}$	= rate of change of structural weight for the jth step, lb/s
$\boldsymbol{x}(x_1, x_2, \ldots, x_k)$	= k-dimensional time-dependent state vector
α	= angle of attack, deg
β	= sideslip angle, deg
γ	= flight-path angle, deg
η_i	= throttle setting for the ith engine
$\boldsymbol{\theta}$	= control transition vector
λ_i	= Lagrangian multipliers associated with constraints c_i
τ	= time, s
Ω	= longitude of ascending node, deg
ω	= argument of perigee, deg
$\boldsymbol{\omega}_B(\omega_{\Xi x}, \omega_{\Xi y}, \omega_{\Xi z})$	= inertial angular velocity expressed in body frame, rad/s

Matrices

$[BI]$	= 3 × 3 body-to-inertial transformation matrix
$[E]$	= 4 × 3 quaternion matrix, $\begin{bmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix}$
$[F]$	= state variational matrix
$[I]$	= 3 × 3 inertia matrix
$[II]$	= 3 × 3 inverse inertia matrix
$[\dot{I}]$	= 3 × 3 inertia rate matrix
$[J]$	= Jacobian matrix
$[\Phi]$	= state transition matrix

Introduction

THE optimal design of flight trajectories plays an important role in reducing operational launch support cost and maximizing payload injection capability to a desired final orbit. Traditionally, the trajectory is shaped using an open-loop three-degree-of-freedom (DOF) vehicle dynamic model to achieve the optimum result within a reasonable computation time.[1] With recent advancements in computer technology and optimization techniques, it is possible to use higher-fidelity models for optimization. In recent years, a number of attempts[2] have been made to incorporate six-DOF effects into three-DOF trajectory simulation models to allow for the dynamic response of a vehicle to guidance steering commands.

The open-loop three-DOF ascent trajectory is shaped to maximize the propellant margin for a specified payload weight into park orbit. The actual flight can be made to mimic the simplified three-DOF trajectory by appropriate selection of guidance parameters. Although the launch vehicle follows the simplified optimal path, it is still off-optimal because of vehicle inertias, center-of-gravity offsets, and flight control responses to guidance commands. Traditionally, a high-fidelity six-DOF trajectory simulation is used prior to launch to validate the optimal three-DOF solution and to evaluate aerodynamic loading and aerothermal heating. The differences between the three-DOF and six-DOF simulations are sometimes significant, particularly when launch-site winds are taken into account.

Received July 10, 1996; presented as Paper 96-3776 at the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, July 29–31, 1996; revision received Feb. 24, 1997; accepted for publication Feb. 28, 1997.

*Manager, 6D Optimization Program. Associate Fellow AIAA.

†Senior Engineer, Scientific Software.

‡Senior Staff Engineer, Scientific Software.

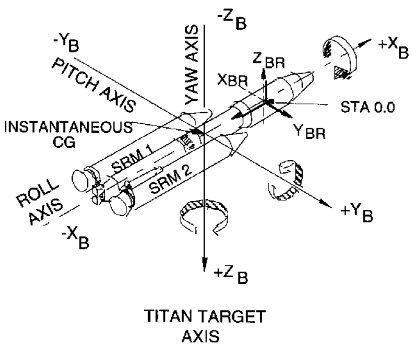

Fig. 1 Titan IV configuration and coordinate system.

Propellant margin is reserved to allow for three-DOF-to-six-DOF differences at the cost of payload weight.

It is a logical extension to use higher-fidelity trajectory optimization to minimize three-DOF-to-six-DOF differences and simplify launch-day operations. To accomplish these objectives, the trajectory simulation should include both translational and rotational dynamics, representative flight software guidance, and flight control logic. The optimization process should be computationally fast to meet day-of-launch timelines and numerically stable when subjected to launch-site wind profiles. The purpose of this paper is to provide an overview of the six-DOF trajectory targeting and optimization algorithm development, called six-DOF TOP. This nonlinear programming solution is applied to a typical Titan IV mission from liftoff to park orbit. Six-DOF TOP is compatible with the POST family of programs, which include generalized three-DOF simulation and optimization and detailed six-DOF simulation. The POST tool set is used for Titan IV trajectory and mission design.

The Titan IV trajectory is divided into three segments. The first segment, stage 0, is that part of the trajectory flown under the power of the solid rocket motors (SRMs). This stage is flown with open-loop guidance. The guidance is loaded with a body rate schedule that determines the attitude of the vehicle at any given point during stage 0. An autopilot steers toward these commanded attitudes using closed-loop feedback. The stage 0 segment of flight follows a standard sequence of events in which the vehicle rolls to the flight azimuth, followed by a constant pitch rate. Between 25 and 85 s, the vehicle flies a load relief profile. During this period, winds may have a profound effect on the trajectory. Load limits, such as maximum dynamic pressure, must be strictly observed.

As the stage 0 engines lose thrust, stage I engines take over and the SRMs are jettisoned. Stages I and II fly in roughly equivalent manners, utilizing closed-loop guidance steering. The guidance model generates steering commands that allow the vehicle to satisfy aim points. During stage I, the payload fairing is jettisoned, since atmospheric effects are no longer harmful to the payload. At stage II shutdown, the desired park orbit has been achieved. Figure 1 shows the Titan IV configuration and coordinate system.

The Titan IV optimization problem is to maximize payload weight, or its near-equivalent, stage II propellant margin. Independent, or control, parameters include attitude rates during stage 0 and guidance aimpoints for stages I and II. Dependent, or constraint, parameters include load relief limits and desired conditions in the parking orbit.

Mathematical Analysis

Formulation

The trajectory optimization problem can be stated in general terms as follows. Minimize

$$F[\boldsymbol{x}(t_f), t_f, \boldsymbol{u}] \tag{1}$$

subject to constraints

$$\boldsymbol{c}[\boldsymbol{x}(t), t, \boldsymbol{u}] \geq 0 \tag{2}$$

and dynamic equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}[\boldsymbol{x}(t), t, \boldsymbol{u}] \tag{3}$$

with initial conditions

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{4}$$

Table 1 Six-DOF optimization variables

Item	Variable
1. Objective function	Stage II propellant margin
2. Constraint or dependent variables	
a. Max. dynamic pressure in stage 0 flight	$q_{\max}$
b. Product of dynamic pressure and angle of attack during load relief in stage 0	q_α
c. Product of dynamic pressure and sideslip angle during load relief in stage 0	q_β
d. Angle of attack at stage 0–I separation	$\alpha_{0/I}$
e. Radius vector magnitude at park orbit inject (POI)	R
f. Velocity vector magnitude at POI	V
g. Cosine of flight-path angle at POI	$\cos\gamma$
h. Cosine of inclination angle at POI	$\cos i$
i. Cosine of longitude of ascending node at POI	$\cos\Omega$
j. Cosine of argument of perigee at POI	$\cos\omega$
3. Control or independent variables	
a. Roll rate during stage 0 roll maneuver to flight azimuth	ω_{XB}
b. Pitch rate during stage 0 load relief and beyond	ω_{YB}
c. Yaw rate during stage 0 load relief	ω_{ZB}
d. Pitch angular acceleration during load relief and beyond	$\dot\omega_{YB}$
e. Yaw angular acceleration during load relief	$\dot\omega_{ZB}$
f. Angular momentum vector components at stage 0–I separation and stage I, II shutdowns	h_x, h_y, h_z
g. Radius vector magnitude at stage I, II shutdowns	R
h. Velocity vector magnitude at stage I, II shutdowns	V
i. Radius rate at stage I, II shutdowns	VR

For the optimization of a Titan IV launch vehicle trajectory, F represents the stage II propellant margin to be maximized and $\boldsymbol{u}$ represents a set of control variables such as body rates, angular accelerations, and the target aim points for each stage. The equality and inequality constraints are used to satisfy such conditions as vehicle load limits, aeroheating, and park orbit injection parameters. Table 1 defines the objective function, constraint, and control variables used for this analysis.

Up to 22 constraints are applied at specific time points, or phase events, which are dependent on the vehicle performance up to that event. Some constraints can be specified as continuous functions of time over a specified period or specified at a time that is dependent on the trajectory. Examples of such constraints are 1) $q\alpha$ and $q\beta$ constraints specified as continuous functions of time over the load relief phase and 2) maximum dynamic pressure. Traditionally, constraints α and $\beta = 0$ are used over the load relief phase for three-DOF optimization. A definite performance gain will be realized by reducing both $q\alpha$ and $q\beta$ to reach minimum values at or near the maximum dynamic pressure point. Further, $q\alpha$ and $q\beta$ constraints are more realistic ones for a boost trajectory since the vehicle load depends on the products $q\alpha$ and $q\beta$ rather than α and β alone.

Up to 50 controls are activated at various events. There are two types of controls: 1) segment controls such as stage 0 pitch rates and accelerations, which are applied over a period of time, and 2) guidance controls such as guidance targets at stage I and II shutdown, which are applied at a specific performance time point. The controls are transformed into guidance parameters or variables, which are used to calculate the steering commands to the flight control logic.

The dynamic equations represented by Eq. (3) are the equations of motion that determine the trajectory. The set of state differential equations representing translational, rotational, and vehicle mass change can be written in the Earth-centered inertial (ECI) frame as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{e}} \\ \dot{\boldsymbol{\omega}}_B \\ \dot{W}_G \end{bmatrix} \tag{5}$$

where

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \qquad \dot{\boldsymbol{v}} = [BI][\boldsymbol{A}_{TB} + \boldsymbol{A}_{AB}] + \boldsymbol{G}_I \qquad \dot{\boldsymbol{e}} = \tfrac{1}{2}[E]\boldsymbol{\omega}_B$$

$$\dot{\boldsymbol{\omega}}_B = [II][\boldsymbol{M}_{TB} + \boldsymbol{M}_{AB} - [\dot{I}]\boldsymbol{\omega}_B - \boldsymbol{\omega}_B \times [I]\boldsymbol{\omega}_B]$$

$$\dot{W}_G = -\left(\sum_i \eta_i \dot{W}_{Pi} + \sum_j \dot{W}_{Sj}\right)$$

The dynamic equations, when integrated through a series of phases, determine the state variables x, y, z, v_x, v_y, v_z, e_0, e_1, e_2, e_3, ω_{XB}, ω_{YB}, ω_{ZB}, and W_G at any time from go-inertial to park orbit inject (POI).

Curve Fit

A large number of input data representing physical processes are needed to simulate a booster trajectory through the Earth's atmosphere. Thrust and propellant flow characteristics of multiple engines are represented as functions of time from ignition. Mass properties of the vehicle, such as center of gravity and moments of inertia, are functions of the vehicle gross weight. Atmospheric characteristics such as pressure, density, and temperature are modeled by tabular data or approximated by several polynomial curve fits as functions of altitude. Wind characteristics, such as speed and direction as a function of altitude, are the most uncertain and volatile and the most demanding on control authority and vehicle dynamic loads. Aerodynamic characteristics of the vehicle, such as force and moment coefficients, are presented as functions of the angle-of-attack α, angle of sideslip β, and Mach number.

The dependence of tabular data on auxiliary variables requires a small integration step size, on the order of 0.04 s. Derivative discontinuity at tabular break points imposes further constraints on finite differencing in the optimization process. To achieve a fast and accurate trajectory simulation, as well as stability and rapid convergence during the optimization process, the tabular data are replaced by appropriate curve fits (cubic spline or polynomial).

Figure 2 shows the cubic spline curve fit with 44 break points (spline knots) for typical Titan IV SRM thrust profile tabular data, with accuracy shown in Table 2. The trajectory integration for each of the 200 or more tabular data is reduced to the 44 break points, thereby allowing a larger integration step size. Discontinuities in the slope at tabular data points are completely eliminated by the cubic spline fit, representing a more realistic thrust profile.

Although the majority of tabular data can be replaced by appropriate curve fits, certain tabular data representing sharp gradients, as in thrust buildup and tailoff or wind profiles, cannot be replaced without loss of accuracy. Central finite differencing is used to compute derivatives from tabular data.

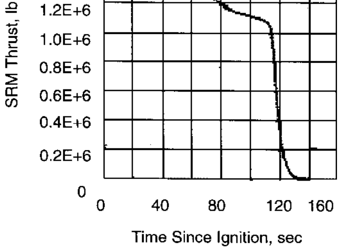

Fig. 2 SRM thrust vs time.

Table 2 Sample curve fit accuracy

Sample data	Sum of squared errors	% of max. deviation	Difference in area under curve, %
	Cubic spline curve fit: thrust		
Stage 0, SRM	2.861E−4	2.860E−1	−2.274E−2
Stage I	1.191E−3	3.045E−2	−6.069E−3
Stage II, main engine	2.270E−3	2.150E−1	2.472E−2
	Polynomial curve fit		
Stage 0, $x_{\mathrm{c.g.}}$	−2.00E−3	1.041E−2	−7.346E−3
Stage 0, I_{xx}	−7.80E−5	2.370E+0	9.489E−2
Stage 0, I_{yz}	−9.31E−3	8.391E−1	3.217E−1

Simplified Guidance

To achieve fast trajectory simulations representing Titan IV missions, the guidance logic is simplified. The Titan IV flight software uses open-loop steering during stage 0 flight and closed-loop linear sine steering during stage I and II flight. The flight software derives its navigational state from an inertial measurement unit (IMU) by accumulating quantized sensor counts over 0.04 s and computing delta velocities. The ECI delta velocities are integrated to compute the navigational state vector. In the trajectory simulation program, the IMU model is eliminated and dynamic equations are integrated to provide dynamic state vectors. Since the difference between the dynamic and navigation state vectors is typically less than 50 ft in magnitude over a boost trajectory to POI, it is logical to eliminate the sensor model and replace navigational computations by the dynamic state to achieve higher computational speed and larger integration step sizes without loss of accuracy. Further, the actual attitude of the vehicle, normally determined by the IMU gimbal angles, is replaced by the attitude determined by integrating the rotational dynamics. The open-loop and closed-loop steering logic provides accurate attitude error, attitude rate, and lateral acceleration inputs to the simplified autopilot logic.

Simplified Autopilot

A simplified autopilot model is required to perform flight control computations quickly without degrading the response characteristics of the system. In recent years, Robert Luke[2] developed an enhanced three-DOF optimization program incorporating an autopilot logic to simulate the vehicle response during stage 0 flight using simplified rotational dynamic equations.

In this analysis, the rotational equations are integrated without any simplifying assumptions to provide an accurate vehicle attitude truth model. Inputs to the flight control system are 1) roll, pitch, and yaw attitude errors, 2) body lateral accelerations computed by guidance, and 3) body rates from the stage I rate gyros. Outputs are the thrust vector deflection angles required to affect the commanded attitude. Considerable simplification of the six-DOF simulation is realized by replacing the digital autopilot with an equivalent analog autopilot. Feedback loops include body attitude, angular rate, angular acceleration, and lateral acceleration to correct the attitude and to reduce the aerodynamic loads.

For optimizing the stage II margin, a rigid-body simulation of the vehicle is adequate. Since only gross vehicle attitude behavior is required for the simulation, smaller dynamic effects such as slosh, vibration, buffeting, engine dithering, and vehicle flexible modes are neglected. There are no high-frequency inputs into the system; therefore, high-frequency filters are not needed. The feedback loop execution (and integration) step size of 0.5 s was found to be sufficient to simulate the rotational motion and flight control system response.

A continuous-time analog autopilot replaces the digital autopilot. As a result, only Bode gains are required, and all filter terms are set to unity. Note, however, that there is a "pseudo" integration performed on the lateral acceleration signals to ensure adequate vehicle response to the low-frequency wind inputs. Since the vehicle center of gravity and moments of inertia vary widely as propellant is consumed during the flight, the gains are varied linearly during segments as in the original Titan IV autopliot. This piecewise linear scheme, with break points located more or less equidistantly in time

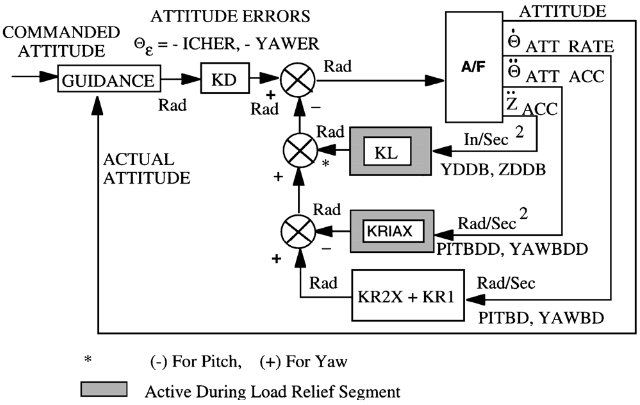

Fig. 3 Simplified autopilot for stage 0 pitch and yaw channels.

and with gains updated at each sampler cycle, is used for six-DOF optimization. The simplified autopilot block diagram for stage 0 pitch and yaw channels is shown in Fig. 3.

Numerical Integration

A large number of ordinary differential equations are integrated during the optimization process. Most trajectory optimization tools use finite differencing, where control parameters are perturbed, and simulations are run to construct constraint and objective function sensitivities. In six-DOF TOP the number of trajectory simulations required to compute the Jacobian and objective gradients is reduced by numerically integrating analytical representations of the variational equations (NAVE) instead of finite differences. In addition to the integration of 14 state dynamic equations, the NAVE approach requires integration of 14 × 14 state transition variational equations, and a maximum of 14 × 4 control transition variational equations. A third-order Runge–Kutta–Lear (RKL) integrator,[3] which requires only three derivative passes, provides improved accuracy and a significant reduction in trajectory simulation time. The RKL is also a self-starting integrator and accommodates the variable step size needed for a launch vehicle trajectory simulation.

Partial Derivatives for Optimization

The optimization process requires three key ingredients: 1) function evaluations, that is, for a given set of control parameter values, running a trajectory simulation to evaluate constraint and objective-function values, 2) a Jacobian matrix $[J]$ of constraints sensitivities to control variations, and 3) an objective gradient of objective change with respect to control variations. The NAVE process generates both Jacobian matrix and objective gradient simultaneously.

Function Evaluation

The simulation process provides function evaluations by integrating the six-DOF equations of motion. A function evaluation also provides a time history of the vehicle state. The state is driven by the independent control parameters, such as stage 0 pitch rates and stage II guidance targets. The vehicle state, in turn, determines the value of the dependent constraints and the objective function. Let the 14 × 1 vehicle state $\boldsymbol{x}$ be defined as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \\ \boldsymbol{e} \\ \boldsymbol{\omega}_B \\ W_G \end{bmatrix} = \begin{bmatrix} \text{inertial position, } 3 \times 1 \\ \text{inertial velocity, } 3 \times 1 \\ \text{inertial body attitude quaternion, } 4 \times 1 \\ \text{inertial rotation rate in body frame, } 3 \times 1 \\ \text{vehicle gross weight, } 1 \times 1 \end{bmatrix}$$

The state is propagated by integrating the equations of motion,

$$\boldsymbol{x} = \int_{t_0}^{t} \dot{\boldsymbol{x}}(\tau)\,\mathrm{d}\tau \qquad \text{with} \qquad \boldsymbol{x} = \boldsymbol{x}_0 \qquad \text{at} \qquad t = t_0 \tag{6}$$

Jacobian Matrix

For each element of the Jacobian matrix $[J] = [\partial \boldsymbol{c}/\partial \boldsymbol{u}]$, the sensitivity of the constraint c_j at time t_j with respect to the kth control u_k, which is activated at time t_a and deactivated at time t_d, is as follows.

1) If the constraint at t_j is in the active region of the control u_k, then

$$\frac{\partial c_j(t_j)}{\partial u_k} = \left(\frac{\partial c_j(t_j)}{\partial \boldsymbol{x}(t_j)}\right)_{1 \times 14} \left[\boldsymbol{\theta}_{k,t_j}\right]_{14 \times 1} \qquad \text{for} \qquad t_a < t_j < t_d$$

2) If the constraint at t_j is beyond the deactivation time t_d of the control u_k, then

$$\frac{\partial c_j(t_j)}{\partial u_k} = \left(\frac{\partial c_j(t_j)}{\partial \boldsymbol{x}(t_j)}\right)_{1 \times 14} [\Phi_{d,j}]_{14 \times 14} \left[\boldsymbol{\theta}_{k,td}\right]_{14 \times 1}$$

$$\text{for} \quad t_a < t_d \le t_j$$

3) If the constraint at t_j is beyond the control guidance parameter (target aim point) activation–deactivation time t_d, then

$$\frac{\partial c_j(t_j)}{\partial u_k} = \left(\frac{\partial c_j(t_j)}{\partial \boldsymbol{x}(t_j)}\right)_{1 \times 14} [\Phi_{d,j}]_{14 \times 14} \left(\frac{\partial \boldsymbol{x}(t_d)}{\partial u_k(t_d)}\right)_{14 \times 1}$$

$$\text{for} \quad t_a = t_d \le t_j$$

Here $[\Phi]$ and $\boldsymbol{\theta}$ are the state transition matrix and control transition vector, respectively. The transformation vectors $\partial c_j/\partial \boldsymbol{x}$ and $\partial \boldsymbol{x}/\partial u_k$ are derived analytically. The vector $\partial \boldsymbol{x}/\partial u_k$ represents the variation of state vector with respect to each control guidance parameter, or target aim point.

The 14 × 14 state transition matrix $[\Phi]$ maps changes in the state at any event time t_i to changes in the state at a future event time t_{i+1}:

$$[\Phi_{i,i+1}] = \frac{\partial \boldsymbol{x}(t_{i+1})}{\partial \boldsymbol{x}(t_i)} \tag{7}$$

$[\Phi]$ is computed by integrating the state variational equations of motion:

$$[\dot{\Phi}_{i,i+1}] = \int_{t_i}^{t_i+1} [F(\tau)][\Phi_{i,\tau}]\,\mathrm{d}\tau \qquad \text{with} \qquad [\Phi_{i,i}] = [I] \qquad \text{at} \qquad t = t_i$$

where $[I]$ is a 14 × 14 identity matrix and $[F(\tau)]$ is the 14 × 14 state variational matrix,

$$[F(\tau)] = \left[\frac{\partial \dot{\boldsymbol{x}}(\tau)}{\partial \boldsymbol{x}(\tau)}\right] \tag{8}$$

$[F]$ is constructed by analytically differentiating the equations of motion (5) with respect to the state components $\boldsymbol{x}$. The state transition matrix is generated across each simulation phase. It is initialized at the start of each phase because of potential state discontinuities at any given event. Chaining properties are used to compute $[\Phi]$ from any time t_i to any future time t_j across a number of events.

The 14 × 1 control transition vector $\boldsymbol{\theta}$ maps changes in the state vector $\boldsymbol{x}$ at a time t_j to the changes in the kth control parameter u_k activated at the time t_a, as

$$\boldsymbol{\theta}_{k,t_j} = \frac{\partial \boldsymbol{x}(t_j)}{\partial u_k(t_a)} \tag{9}$$

The control transition vector is computed in an analogous fashion as the state transition matrix

$$\boldsymbol{\theta}_{k,t_j} = \int_{t_a}^{t_j} \{[F(\tau)]\boldsymbol{\theta}_{k,\tau} + \boldsymbol{G}\}\,\mathrm{d}\tau \qquad \text{with} \qquad \boldsymbol{\theta}_{k,t_a} = \boldsymbol{0}$$

where $\boldsymbol{0}$ is a 14 × 1 vector with all zeros, $[F(\tau)]$ is the state variational matrix defined earlier, and $\boldsymbol{G}(\tau)$ is the control variational vector, constructed analytically from the state equations of motion as

$$\boldsymbol{G}(\tau) = \frac{\partial \dot{\boldsymbol{x}}(\tau)}{\partial u_k(\tau)} \tag{10}$$

$\boldsymbol{\theta}$ is evaluated only from the time of control activation to the time of control deactivation.

Objective Gradient

The objective gradient $\partial F/\partial u_K$ (the stage II propellant margin at the stage II shutdown time t_f with respect to the control parameters u_k) is approximated by

$$\frac{\partial F}{\partial u_k} = \dot{W}_P \left[\frac{\partial t_f}{\partial \boldsymbol{x}(t_f)}\right]_{1\times 14} [\Phi_{d,f}]_{14\times 14}\left[\boldsymbol{\theta}_{k,td}\right]_{14\times 1}$$

where $\dot{W}_P$ is the propellant flow rate and

$$\left[\frac{\partial t_f}{\partial \boldsymbol{x}(t_f)}\right]_{1\times 14} =$$

$$\left[\frac{\partial t_f}{\partial R_X}\ \frac{\partial t_f}{\partial R_Y}\ \frac{\partial t_f}{\partial R_Z}\ \frac{\partial t_f}{\partial V_X}\ \frac{\partial t_f}{\partial V_Y}\ \frac{\partial t_f}{\partial V_Z}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\right]_{1\times 14}$$

NAVE techniques are used to compute the elements of the matrix $[F]$, vector $\boldsymbol{G}$, Jacobian matrix $[J]$, and control transition vector $\boldsymbol{\theta}$ when the control parameters are represented by the guidance target parameters. Also, the partial derivatives of polynomial curve fits are derived analytically. Partial derivatives of tabular data are derived using central finite differencing. The six-DOF trajectory optimization program, six-DOF TOP, computes and integrates all appropriate partials and chains them together at the appropriate times; only the necessary partials are computed.

Optimization Technique

The six-DOF trajectory optimization program uses NPSOL 4.0 (Ref. 4) to solve the constrained optimization problem. NPSOL uses a projected Lagrangian formulation with sequential quadratic programming (SQP) to minimize F. The Lagrangian function is defined as

$$L(\boldsymbol{u}, \lambda) = F(\boldsymbol{u}) - \sum_{i=1}^{m} \lambda_i c_i(\boldsymbol{u}) \tag{11}$$

where λ_i is the Lagrange multiplier associated with the constraint c_i. Successive iterations are used to determine the optimal solution for $\boldsymbol{u}$. Iteration convergence is strongly dependent on the weights, or scaling values, associated with each control and constraint parameter.[5,6] Six-DOF TOP selects weighting values that normalize the Jacobian matrix in order to minimize its condition number. The Jacobian matrix may be computed as often as each iteration, but usually is recomputed only as needed.

Trajectory and Optimization Results

Timing Analysis

A typical six-DOF POST run requires approximately 160 s of execution time to propagate a vehicle's state from liftoff to park orbit inject. The six-DOF TOP trajectory simulation time was reduced by 1) polynomial or cubic spline curve fits, 2) simplified guidance, 3) a simplified analog autopilot, and 4) an RKL third-order integrator. The execution time for a single simulation pass was reduced from 160 to 16 s. The timing improvement due to each design change is shown in Table 3.

Simulation Accuracy

Comparisons were made of the entire simulation by comparing timing and states at various points through the trajectory against the six-DOF POST simulation. Severe wind profiles (Fig. 4) were also introduced to verify the robustness of the autopilot and the ability of the 0.5-s integration step to respond to this wind. Summaries of this final comparison are shown in Tables 4 and 5.

Table 3 Timing improvement

Change	Reduction in execution time, %
Curve fits	5.5
Guidance simplification	32.1
Analog autopilot	32.1
RKL integrator	20.6

Table 4 Comparison of events

Quantity	Six-DOF POST	Six-DOF TOP
Integration step size, s	0.04	0.5
Time, s		
Go-inertial	−4.0	−4.0
SRM ignition	0.0	0.0
Roll maneuver to flight azimuth	6.0	6.0
End roll maneuver	9.0	9.0
Pitch–yaw open-loop steering	10.0	10.0
Start load relief	25.0	25.0
Time at max dynamic pressure	57.48	57.50
End load relief	85.0	85.0
Stage 0–I staging (1.3 g sense)	116.909	116.895
Start stage I main engine	116.989	116.975
Stage 0–I separation	125.091	125.081
Stage I closed-loop steering	131.0	131.0
Payload fairing separation	244.005	243.365
Stage I shutdown, stage II ignition	310.066	310.046
Stage I–II separation	310.756	310.735
Stage II closed-loop steering	315.0	315.0
Stage II shutdown	530.985	531.005
Stage II–upper-stage separation	540.095	540.095

Table 5 Comparison of POI conditions

Variable	Six-DOF POST	Six-DOF TOP
Integration step size, s	0.04	0.5
Time, s	540.095	540.095
Perigee altitude, n mile	84.550	84.540
Apogee altitude, n mile	118.328	117.873
Inclination, deg	28.62	28.62
Arg. of perigee, deg	84.20	83.98
Long. of ascending node, deg	30.29	30.50

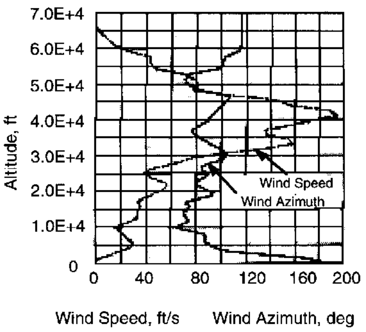

Fig. 4 Wind speed and direction.

The results on angle of attack and body pitch rate from six-DOF TOP (0.5-s step size) simulation are shown in Figs. 5 and 6. These results compare very well with six-DOF POST (0.04-s step size), showing differences less than 0.1% in flight parameters and less than 0.7% in end conditions. Thus, six-DOF TOP simplified guidance and autopilot models, with large integration step sizes, provide the simulation accuracy and computational speed needed for effective six-DOF optimization.

Optimization Results

The Titan IV K-14 mission, flown successfully in December 1994, has been used to supply data for six-DOF TOP simulation and optimization studies.[7] The Titan targeting cycle uses extensive three-DOF optimization to shape the flight profile, while maximizing the stage II propellant margin. The optimized flight parameters, such as body rates and target conditions, are transformed into guidance parameters. Detailed six-DOF simulations verify results from the optimized three-DOF performance. Because of model differences, a propellant margin bias of 10 to 60 lb is levied on three-DOF performance. This also recognizes that detailed six-DOF simulations are optimal only from a three-DOF perspective, and are six-DOF-suboptimal. Just prior to launch, wind conditions are measured, and a three-DOF optimization with six-DOF simulation is exercised to provide updated guidance parameters. These parameters are used by the autopilot to steer the vehicle and provide load relief in the presence of launch winds.

Table 6 Optimization performance

Method	Stage II propellant margin, lb	
	No winds	With winds
Three-DOF optimization	2119.0	2124.4
Six-DOF simulation (three-DOF controls)	2090.2	2088.7
Six-DOF TOP, finite differencing partials	2138.5	2185.2
Six-DOF TOP, NAVE partials	2145.2	N/A

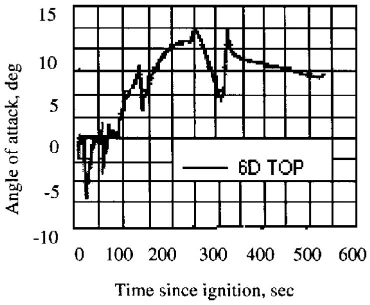

Fig. 5 Angle of attack vs time.

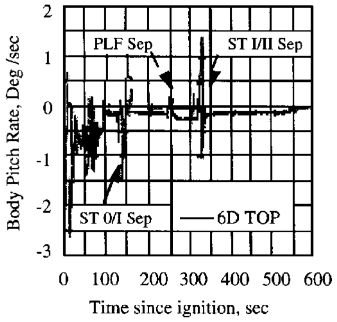

Fig. 6 Body pitch rate vs time.

Six-DOF TOP control, constraint, and objective parameters have been formulated to support both the prelaunch and day-of-launch targeting cycles. The six-DOF TOP tool eliminates the need for any three-DOF-to-six-DOF performance penalty. Table 6 summarizes the performance results. All optimization results are after successful convergence.

The difference between three-DOF optimization and six-DOF simulation verification shows why a propellant margin bias is necessary. With six-DOF TOP, a margin bias is unnecessary because a true six-DOF optimal solution has been found. This performance improvement allows a commensurate increase in payload weight capability. Even with winds, the six-DOF TOP performance is excellent. The advantage of NAVE over finite differencing is also evident. Not only is there a performance benefit, but optimization with NAVE-derived partials is over four times faster than optimization with finite differencing.

Six-DOF TOP optimization usually converges in less than 15 min on a CDC/MIPS 4680 computer. All constraints, such as maximum dynamic pressure and POI conditions, are met, even in the presence of day-of-launch winds. About 95% of constraint and optimal propellant margin is achieved within the first 15% of run time and iterations. The rapid convergence rate is a reflection of the accuracy inherent in NAVE-derived partials and the fidelity of the simulation process used in function evaluations.

Conclusion

A six-DOF trajectory simulation and optimization program (six-DOF TOP) has been developed for a typical launch-vehicle boost mission. Simplified guidance and autopilot functions, along with curve fits, result in an increase in trajectory simulation computational speed by a factor of 10. Simulation accuracy better than 0.01 n mile is achieved at park orbit inject. A more realistic $q\alpha$ constraint during the load relief phase of the trajectory provides greater flexibility in optimization and enhances the launch-vehicle performance. Use of numerical integration of analytically derived variational equations to compute Jacobian matrix and objective gradients enhances the robustness, computational speed, stability, and accuracy of the optimization process. Net performance, in terms of added payload weight capability, is improved.

Acknowledgments

This study was supported by the Titan IV Continuous Product and Process Improvement task, Advanced Integration Analysis, U.S. Air Force Contract F04701-85-C-0019. The authors wish to acknowledge the support provided by Gary Brauer, Scientific Software Manager, and Wayne Hallman of The Aerospace Corporation.

References

[1] Brauer, G. L., Cornick, D. E., Olson, D. W., Peterson, F. M., and Stevenson, R., "Program to Optimize Simulated Trajectories (POST)," Martin Marietta Corp., MCR-87-583, NAS1-18147, Denver, CO, Sept. 1987.

[2] Luke, R. A., "Rotational Motion and Guidance System Approximations in Optimizable Operational Launch Vehicle Simulations," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 3, 1993, pp. 477–483.

[3] Lear, W. M., "Accuracy and Speed of 38 Self-Starting Integrators," NASA Johnson Space Center, Rept. JSC 14364, Houston, TX, July 1978, pp. 35, 36.

[4] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL (Version 4.0): A FORTRAN Package for Nonlinear Programming," Dept. of Operations Research, Stanford Univ., Rept. SOL 86-2, Stanford, CA, Jan. 1986.

[5] Hallman, W. P., "Sensitivity Analysis for Trajectory Optimization Problems," AIAA Paper 90-0471, Jan. 1990.

[6] Hallman, W. P., "Optimal Scaling Techniques for the Nonlinear Programming Problem," AIAA Paper 94-4417, Sept. 1994.

[7] Rao, P. P., Hong, P. E., Sutter, B. M., and Kent, P. D., "Six Degrees-of-Freedom Trajectory Optimization Program," Lockheed Martin Corp., MCR-95-2617, Denver, CO, March 1996.

F. H. Lutze Jr.
Associate Editor