HYBRID NEIGHBORING-OPTIMAL-CONTROL AND LAMBERT-BASED INTERCEPTOR BOOST-PHASE GUIDANCE

John A. Lawton and Craig A. Martell
Naval Surface Warfare Center, Dahlgren Division, Dahlgren, Virginia

Abstract

Finding guidance methods that efficiently and effectively handle the problem of directing the boost-phase portion of an exoatmospheric tactical ballistic missile interceptor is an active area of research. This paper presents a candidate method that is a hybrid combination of three algorithms. The chief method, used throughout the majority of the flight, is a minimum-time neighboring optimal control scheme. As demonstrated through Monte Carlo simulations, perturbations in the predicted intercept point (caused by boost-phase updates of the target state estimate received from an exogenous radar) are corrected for in a smooth manner which preserves performance. Even large corrections are handled with low angles of attack, and less than one percent changes in burnout velocity relative to theoretical open-loop optimal control values. Near the end of the booster burn, the guidance scheme switches over to two Lambert-based guidance methods, which have the effect of taking out residual velocity errors, thus putting the interceptor on a nearly exact intercept path to the latest estimate of the target trajectory. The total scheme, while being somewhat computationally complex in pre-deployment development, is actually computationally simple for on-board computations, and has a small computer memory requirement. Therefore, the robust performance combined with the light computational burden provided by this hybrid algorithm make it a competitive candidate for boost-phase interceptor guidance.

Introduction

The problem of finding a boost-phase guidance method for the exoatmospheric interception of tactical ballistic missiles is a challenge. The flight environment starts where the atmospheric density is the highest, and ends where it is negligible. At boost-phase burnout, the interceptor must be put on a path that intercepts the target after a coast of up to minutes in duration. Furthermore, it must be able to handle large redirections, as boost-phase updates are received from an exogenous radar filter, without losing appreciable performance.

For these reasons, neighboring optimal control is investigated as a candidate guidance algorithm[1,2,3,4]. Optimal control, by its very nature, must preserve performance in order to meet the optimization criterion (which is minimum intercept time in this case). Neighboring optimal control, which is a closed-loop guidance method based on open-loop optimal control solutions, preserves the optimal flight path when the interceptor state and the target state are perturbed from the nominal path.

As the burnout time approaches, however, the neighboring optimal control scheme has some residual errors that amount to on the order of one kilometer of intercept error. While this error level may seem small by some measure, when combined with all other error sources, it makes a nontrivial contribution. Therefore, two algorithms that are based on the solution to Lambert's problem, of classical orbital mechanics origin, are used at the end of the boost phase to efficiently drive the intercept errors to approximately zero[5,6,7].

The hybrid combination of these methods yields an algorithm that achieves all of the goals listed above. This hybrid algorithm is derived in detail in this paper, and representative results from Monte Carlo simulations are used to illustrate its performance.

Guidance Methods

The total guidance scheme proposed in this paper is comprised of three methods. The first method is neighboring optimal control[1,2,3,4]. It is the primary guidance method, being used for the majority of the controlled segment of the interceptor flight. As is typical with neighboring extremal methods, the gains become unstable as the burnout time of the booster approaches. At this point, a switch is made to a combination of two methods that are based on Lambert's problem[5].

The first Lambert-based guidance algorithm is the classical velocity-to-be-gained method[6]. For short burn times outside of the atmosphere, velocity-to-be-gained guidance is nearly optimal, so it moves the intercept vehicle in a path that is consistent with the optimal control path for the last few seconds of flight. It is designed to zero out the residual errors in the interceptor velocity, so that it puts the interceptor on a nearly-exact collision course with the target (to within the target state estimate accuracy).

Velocity-to-be-gained guidance has a couple of problems at the extreme end of the burn time. First of all, it may be that the amount of rocket burn time remaining is slightly shy of that required to zero out all of the residual errors. Second, if the residual errors go to zero before the burn time is up, then classical velocity-to-be-gained requires the thrust to be terminated; if the rocket motor has solid propellant, it generally cannot be terminated at will. For such purposes, matched-characteristic-velocity (MCV) guidance, an offshoot of velocity-to-be-maintained guidance, was derived[7]. At about one-half to one second time to go, the guidance switches over to MCV guidance to obtain and maintain a more precise intercept solution.

Open-Loop Optimal Control

The closed-loop neighboring optimal control scheme is based on open-loop optimal control solutions. In the open-loop problem statement, the interceptor vehicle is launched from the Earth's surface, then flies through the atmosphere with thrust, gravitational, and aerodynamic forces acting on it. Burnout of the rocket motor occurs at a point where the aerodynamic forces have become negligible, at time t_{bo}, after which the interceptor coasts under the influence of gravity to the intercept point. The intercept point, $\vec{r}_T$, is fixed for this open-loop problem; the closed-loop neighboring optimal control problem has a moving intercept point, since the target vehicle position is a function of the intercept time, which is continuously updated in the closed-loop guidance mode. The goal of the optimization problem is find the pitch and yaw history of the missile body that minimizes the time of intercept, t_I.

In the optimal control problem, the state is comprised of the position and velocity vectors in launch-centered Earth-fixed (LCEF) coordinates:

$$x = \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} .$$

The LCEF coordinates are centered at the launch point of the interceptor, with the x-axis in the vertical direction, the y-axis in the plane containing the nominal intercept point, and the z-axis out of plane. The controls are non-standard pitch and yaw angles, θ and ψ, defining the direction of the longitudinal axis of the interceptor missile, $\widehat{e}_b$. ψ is the angle between $\widehat{e}_b$ and the x-y plane, and θ is the angle between the projection of $\widehat{e}_b$ onto the x-y plane and the x-axis. Therefore,

$$\widehat{e}_b = \begin{bmatrix} \cos\psi \cos\theta \\ \cos\psi \; \sin\theta \\ \sin\psi \end{bmatrix} .$$

The governing differential equations are

$$\dot{\vec{r}} = \vec{v}$$

and

$$\dot{\vec{v}} = \vec{A} + \vec{N} + \vec{g} + \vec{T} \; ,$$

where

$$\vec{A} = \frac{q\, S\, C_A(h,v)}{m(t)} \widehat{e}_b$$

is the axial aerodynamic acceleration,

$$\vec{N} = \frac{q\, S\, C_{N\alpha} \alpha_T}{m(t)} \widehat{n}$$

is the normal aerodynamic acceleration, $\vec{g}$ is the inverse-square gravitational acceleration, and

$$\vec{T} = \frac{\dot{m}\, g_e\, I_{sp_{\mathrm{vac}}} - p(h)\, A_{\mathrm{exit}}}{m(t)} \widehat{e}_b$$

is the thrust acceleration. In these expressions,

$$q = \frac{1}{2} \rho(h)\, v^2$$

is the dynamic pressure, S is the reference area, $C_A(h,v)$ is the axial force coefficient, $m(t)$ is the vehicle mass, $C_{N\alpha}$ is the normal force curve slope,

$$\widehat{n} = \widehat{e}_b \times \frac{\widehat{e}_b \times \vec{v}}{\|\widehat{e}_b \times \vec{v}\|}$$

is the normal force direction,

$$\alpha_T = \cos^{-1}\left(\widehat{e}_b \cdot \frac{\vec{v}}{\|\vec{v}\|}\right)$$

is the total angle of attack, $p(h)$ and $\rho(h)$ are the atmospheric pressure and density, h is the altitude, A_{exit} is the motor nozzle exit area, $I_{sp_{\text{vac}}}$ is the vacuum specific impulse of the motor, and g_e is the gravitational acceleration at the Earth's surface. Axial force coefficient data is fit, via a least-squares method, to the functional form

$$C_A = c_1 + c_2 \tan^{-1}(c_3 + c_4 M) + c_5 \mathrm{e}^{\left(\frac{c_6 + c_7 M}{c_8 + c_9 M + c_{10} M^2}\right)},$$

where $M = \|\vec{v}\|/a(h)$ is the Mach number (and $a(h)$ is the speed of sound at the current altitude, h). $\rho(h)$ and $a(h)$ are polynomial fits in h to the 1962 Standard Atmosphere, and $p(h)$ is calculated with

$$p(h) = \rho(h)\frac{a(h)^2}{1.4} \; .$$

The solution method chosen for this problem is to discretize the control variables at N node times, $t_1, \ldots, t_N$, linearly interpolate between the nodes, then solve for the optimal set of $2N + 1$ control and flight time parameters $\theta_1, \ldots, \theta_N$, $\psi_1, \ldots, \psi_N$, and t_I, that minimize the intercept time, t_I. The chief constraint on the resulting parameter optimization problem is that the interceptor must hit the target point at time t_I, $\vec{r}(t_I) - \vec{r}_T = 0$ (which is really 3 constraints). Other constraints are operational, such as launch angle constraints, angle of attack constraints, and dynamic pressure limits.

One advantage of this method is that a wide variety of operational constraints can easily be imposed, whereas a variational solution method would require re-formulation for most constraint types. In an operational environment, this is an important consideration, since various facets of the system architecture frequently impose constraints on the problem that the dynamicist would never dream of. Another advantage is that the complete trajectory is specified by a low number of data points (N is typically between 10 to 15), with an exact intercept (to within the modeling accuracy) being permitted with such a low number of data points. This also means that the memory-requirement to hold the gain data is low. Besides, the full two-point boundary value problem resulting from the variational formulation has been solved to compare with this proposed approximate method using the techniques in refs. 8 and 9, and the intercept time difference is on the order of only a tenth of a second.

Neighboring Optimal Control

For any perturbed trajectory in the vicinity of the reference trajectory, let $\delta x(t)$ represent the perturbed optimal state solution minus the reference solution. Similarly, let $\delta u(t)$ represent the difference in the control history, $\delta\vec{r}_T$ the difference in the final intercept position, and δt_I the difference in the intercept time. Then the gain functions $G(t)$ and $H(t)$ are defined through the relation

$$\begin{bmatrix} \delta u(t) \\ \delta t_I \end{bmatrix} = \begin{bmatrix} G(t) & H(t) \end{bmatrix} \begin{bmatrix} \delta x(t) \\ \delta\vec{r}_T \end{bmatrix} . \quad (1)$$

$G(t)$ is an $(m+1) \times n$ matrix, and $H(t)$ is an $(m+1) \times 3$ matrix, where m is the number of controls (2 in this case), and n is the number of state variables (6 for this problem). They are computed at the nodes used in the optimization problem, and approximated as linear functions between the nodes.

The gains are computed by taking numerical partial derivatives. In lieu of re-starting the optimal solution at each node, one trajectory is flown for each finite difference perturbation in $x(0)$ and in $\vec{r}_T$, for a total of $(n+3)$ perturbed solutions[3,4]. For the first n of these solutions, this is accomplished by first perturbing the kth element of $x(0)$ by ϵ_k from the reference initial condition (while the rest of $x(0)$ is the same as the reference initial condition), then solving the optimal control problem again. The initial perturbations are chosen such that the whole perturbed optimal solution is sufficiently close to the reference optimal solution to yield good numerical derivatives at all of the node values. The last three solutions are with only the target point perturbed by ϵ_k, $k = n+1, \ldots, n+3$ (as before, with care taken to keep the whole trajectory sufficiently close to yield good derivatives). Next, for each node time t, a group of $n+3$ solutions from Eq. (1) are collected into one matrix equation:

$$[G(t)\quad H(t)]\left[\begin{pmatrix}\delta x^1(t)\\ \delta\vec{r}_T^{\,1}\end{pmatrix}\ \cdots\ \begin{pmatrix}\delta x^{n+3}(t)\\ \delta\vec{r}_T^{\,n+3}\end{pmatrix}\right]$$

$$=\left[\begin{pmatrix}\delta u^1(t)\\ \delta t_I^1\end{pmatrix}\ \cdots\ \begin{pmatrix}\delta u^{n+3}(t)\\ \delta t_I^{n+3}\end{pmatrix}\right]. \quad (2)$$

Since for the first n solutions the target point is unperturbed from the reference solution, $\delta\vec{r}_T^{\,k}=0\,,\,k=1,\ldots,n$. With this, the right-hand matrix on the left side of the equal sign in Eq. (2) can be partitioned in the form

$$\begin{bmatrix}A & B\\ 0 & C\end{bmatrix},$$

where A is an $n\times n$ matrix, and C is a 3×3 matrix. Now,

$$\begin{bmatrix}A & B\\ 0 & C\end{bmatrix}^{-1}=\begin{bmatrix}A^{-1} & -A^{-1}BC^{-1}\\ 0 & C^{-1}\end{bmatrix},$$

so that, since

$$C=\begin{bmatrix}\epsilon_{n+1} & 0 & 0\\ 0 & \epsilon_{n+2} & 0\\ 0 & 0 & \epsilon_{n+3}\end{bmatrix},$$

straightforward matrix manipulations yield

$$G(t)=\left[\begin{pmatrix}\delta u^1(t)\\ \delta t_I^1\end{pmatrix}\ \cdots\ \begin{pmatrix}\delta u^n(t)\\ \delta t_I^n\end{pmatrix}\right]\cdot[\,\delta x^1\ \ \ldots\ \ \delta x^n\,]^{-1}$$

and

$$H(t)=\left[\begin{pmatrix}\delta u^{n+1}(t)\\ \delta t_I^{n+1}\end{pmatrix}\cdot\frac{1}{\epsilon_{n+1}}\ \cdots\ \begin{pmatrix}\delta u^{n+3}(t)\\ t_I^{n+3}\end{pmatrix}\cdot\frac{1}{\epsilon_{n+3}}\right]-G(t)\left[\delta x^{n+1}\cdot\frac{1}{\epsilon_{n+1}}\ \ \ldots\ \ \delta x^{n+3}\cdot\frac{1}{\epsilon_{n+3}}\right].$$

These gains, having been pre-computed in this manner, are used in Eq. (1) in real time to continuously modify the intercept path.

Intercept Time Updates

Throughout the neighboring optimal control segment of the interceptor trajectory, the estimate of the intercept time, t_I, is continuously updated. Picking off the last row of Eq. (1) at any time t,

$$\delta t_I = G_3\,\delta x + H_3\,\delta\vec{r}_T(\delta t_I)\ , \qquad (3)$$

where the "3" subscripts on the gain matrices denote the third row of each matrix. $\delta\vec{r}_T$ is denoted as a function of δt_I (or, more explicitly as a function of $t_I^*+\delta t_I$) because the ballistic target is moving in time, so each new intercept time requires a new target position. Hence, Eq. (3) is not solved for any arbitrary value of δt_I. Eq. (3) is, however, in the form of a fixed point iteration method, and experience has shown that it converges.

When large changes are made to the estimate of the threat trajectory, such as occur when an update is received from the exogenous radar filter, the fixed point iteration can require 20 or more iterations. Therefore, a Newton's method is used to solve Eq. (3). Subtracting δt_I from the right side of Eq. (3) creates a function that is zero at the solution point; using this function in a Newton's method yields the following iteration scheme:

$$\delta t_I^{i+1}=\delta t_I^i-\frac{G_3\,\delta x+H_3\,\delta\vec{r}_T(\delta t_I^i)-\delta t_I^i}{H_3\,\vec{v}_T(\delta t_I^i)-1} \qquad (4)$$

where $\vec{v}_T$ is the target velocity. This method typically converges, to a tolerance of less than a microsecond, in two to six iterations for large changes in the threat state estimate. The update occurs on the order of once every half of a second, at what will be called the target-adjust frequency. This occurs more frequently than the target updates from the exogenous radar, but less frequently than the guidance cycle.

Alternatively, Eq. (4) can be used as a single update (that is, non-iteratively), once every target-adjust cycle, because it implicitly converges as time moves forward at small increments. In other words, the relative geometries, the values of the gains, etc., are changing more slowly than the rate at which Eq. (4) is converging in this mode. The control and trajectory results are almost identical to those where Eq. (4) converges to a tight tolerance each target update cycle.

To summarize, Eq. (4) is computed once each target-adjust cycle, and the resultant δt_I is used to compute $\delta\vec{r}_T$. This value of $\delta\vec{r}_T$ is then held constant until the next target-adjust time. Between target-adjust cycles, δt_I does vary through Eq. (1), but with $\delta\vec{r}_T$ held fixed.

Velocity-to-be-Gained Guidance

Velocity-to-be-gained guidance is based on the solution to Lambert's problem. One version of Lambert's problem can be stated as follows: given a source position, a target position, and a time of flight, find the initial velocity vector that yields an orbit that traverses from the source to the target in the given time of flight. At any time, t, in the interceptor's flight, denote the velocity vector that is the solution to Lambert's problem as the required velocity, $\vec{v}_r$. The inputs to the solution of $\vec{v}_r$ are the current interceptor position, $\vec{r}(t)$, the position of the threat state estimate at time t_I (where t_I is held constant at the last value given by the neighboring optimal control scheme), $\vec{r}_T(t_I)$, and the time of flight, $\tau = t_I - t$.

The difference between the required velocity and the current interceptor velocity is termed the velocity to-be-gained:

$$\vec{v}_g(t) = \vec{v}_r(t) - \vec{v}(t) \ .$$

The goal of velocity-to-be-gained guidance is to drive $\vec{v}_g$ to zero. The version of this method used in this hybrid algorithm is to thrust in the direction of $\vec{v}_g$. Although more advanced methods exist, such as cross-product steering[6], experience has shown that thrusting along $\vec{v}_g$ is just as effective for the short burn times used in this hybrid guidance scheme.

Matched Characteristic Velocity Guidance

The problem that velocity-to-be-gained guidance has with not converging to an intercept solution at exactly the right time (that is, at thrust cut-off time), can be handled by letting the intercept time, t_I, vary. Each intercept time yields a different velocity to-be-gained vector, so the idea behind MCV guidance is to choose the intercept time such that the velocity to-be-gained approximately equals the amount of velocity increase that can be achieved with the amount of remaining thrust. Classically, the amount of velocity that can be achieved by a given amount of thrust if instantaneously applied is termed the characteristic velocity. Therefore, let the effective characteristic velocity, v_c, denote the velocity increase under the remaining finite burn time, incorporating the other accelerations as well. The magnitude of the velocity to-be-gained vector at the adjusted intercept time must match v_c.

Letting t_I be the current estimate of the intercept time, and Δt_I be the amount that the intercept time is to be changed, the new velocity to-be-gained vector is approximated by

$$\vec{v}_g(t_I + \Delta t_I) = \vec{v}_g(t_I) + \frac{\partial \vec{v}_r(t_I)}{\partial t_I}\Delta t_I \ . \quad (5)$$

The partial derivative can be analytically computed by the method given in the appendix of Ref. 7. Now, in order for the magnitude of $\vec{v}_g(t_I + \Delta t_I)$ to be matched to v_c, first take the dot product of the right hand side of Eq. (5) with itself, subtract off v_c^2, and equate the result to zero, yielding

$$\left\| \frac{\partial \vec{v}_r}{\partial t_I} \right\|^2 \Delta t_I^2 + 2\vec{v}_g \cdot \frac{\partial \vec{v}_r}{\partial t_I}\Delta t_I + \|\vec{v}_g\|^2 - v_c^2 = 0 \, .$$

Solving this,

$$\Delta t_I = \, - b \pm \sqrt{b^2 - \frac{\left(\|\vec{v}_g\|^2 - v_c^2\right)}{\left\|\frac{\partial \vec{v}_r}{\partial t_I}\right\|^2}} \ , \quad (6)$$

where

$$b = \frac{\vec{v}_g \cdot \frac{\partial \vec{v}_r}{\partial t_I}}{\left\|\frac{\partial \vec{v}_r}{\partial t_I}\right\|^2} \ .$$

The sign in Eq. (6) is chosen such that $|\Delta t_I|$ is smallest. This value of Δt_I is then added to the previous estimate of t_I to obtain the current t_I, which in turn is used to calculate the current $\vec{v}_r$ by solving Lambert's problem, and thence the current $\vec{v}_g$. As with velocity to-be-maintained, the thrust direction is along $\vec{v}_g$.

All that is left undone is the determination of v_c. Assuming thrust and gravity are the only forces acting on the interceptor, its acceleration is

$$\dot{\vec{v}} = \vec{g} + \frac{\dot{m}\, u_e}{m_0 - \dot{m}\, t}\hat{d} \ , \quad (7)$$

where the gravity vector

$$\vec{g} = \, - \frac{\mu}{R^3}\vec{R} \ ,$$

$u_e = I_{sp} \cdot g_e$ is the equivalent exhaust velocity, m_0 is the initial vehicle mass, $\dot{m}$ is the (positive) mass flow rate, $\hat{d}$ is the direction of the thrust, μ is the gravitational parameter, $\vec{R} = \vec{r} + [0 \quad 0 \quad R_E]^T$, and R_E is the Earth's radius. Assuming that $\vec{g}$ and $\hat{d}$ are nearly constant over short burn times, both sides of Eq. (7) can be integrated to yield

$$\Delta\vec{v} = \vec{g}\,t_{\text{rem}} + u_e \ln\left(\frac{m(t)}{m(t_{\text{bo}})}\right)\widehat{d}\ ,$$

where t_{bo} is the burnout time of the rocket motor, and $t_{\text{rem}} = t_{\text{bo}} - t$. The magnitude of this vector is v_c. Therefore, taking the dot product of $\Delta\vec{v}$ with itself,

$$v_c^2 = g^2 t_{\text{rem}}^2 - 2g\,t_{\text{rem}}\,u_e \ln\left(\frac{m(t)}{m(t_{\text{bo}})}\right)\left(\widehat{d}\cdot\frac{\vec{R}}{\|\vec{R}\|}\right) + u_e^2\left[\ln\left(\frac{m(t)}{m(t_{\text{bo}})}\right)\right]^2 .$$

Generally, a fudge factor of 0.6 to 0.8 is applied to v_c to account for inaccuracies in this estimation. This also has the effect of slightly underestimating the thrust capacity, which is the conservative approach, causing MCV guidance to tend to convergence as $t_{\text{rem}} \rightarrow 0$. (whereas overestimating tends to lead to burnout before convergence, whence nontrivial residual errors occur).

Simulation

A TBM interceptor will typically be launched based on a threat target state derived from an exogenous sensor, such as a radar. One important design issue in such a system is the decision as to when to launch the interceptor after the target is in track of the radar. For the simulation results presented herein, the interceptor is launched very early, about three seconds after detection, when the radar filter velocity error drops just below 300 m/s. With the approximately 150 s interceptor flight time for the configuration presented below, this means typical initial predicted intercept point (PIP) errors will be on the order of 45 km.

As the interceptor flight progresses, the exogenous radar filter will continue to improve the target state, so the PIP will move appreciably. The guidance algorithm must be able to handle such large changes in the PIP without degrading performance.

In the case presented, the radar is modeled with a 4 mrad angular error, and about a 50 m range error. The filter associated with the radar is an extended Kalman filter of the same design as presented in Ref. 10. After the square root of the trace of the velocity covariance of the filter drops below 300 m/s, the interceptor is launched and the filter update rate is dropped to once every 5 s. Updates on the target state are passed to the interceptor at this same 5 s cycle.

The target is a hypothetical 1000 km minimum-energy ballistic missile trajectory, with the gravitational acceleration being modeled with an oblate-Earth model. The target ground track is depicted in Fig. (1), along with the interceptor launch point and trajectory. The exogenous radar is co-located with the interceptor launch point.

The resultant interceptor trajectory from one of the Monte Carlo set of runs is depicted in Fig. (2). The trajectory labeled "1" is the initial flight path that would have been flown if no target state updates were received from the radar filter. Similarly, the trajectory labelled "2" is the path that would have been flown after the first update was received, but no later updates were received. "3" is the trajectory after two updates, and so forth. None are shown after "8" because they all lump together in the same region. The endpoints of these trajectories, which are essentially the various PIP points, are projected onto all three planes.

Fig. (3) contains the same endpoints projected onto the three planes, only magnified, with each of the axes on the same scale, to give a better feel how the PIPs migrate. Lines connect adjacent endpoints, so only the first two endpoints are labelled. The total PIP migration is 46 km for this particular case.

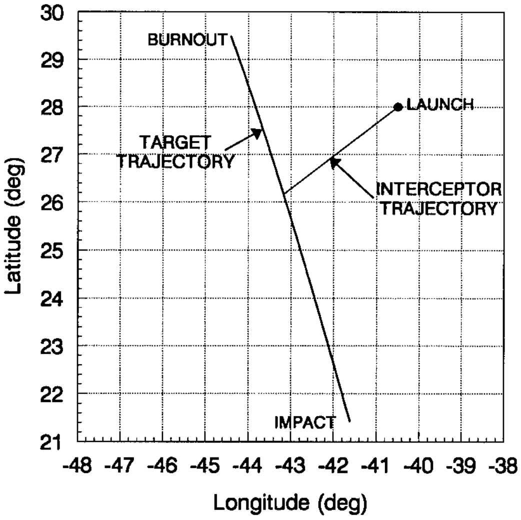

Figure 1. Target and Interceptor Ground Tracks.

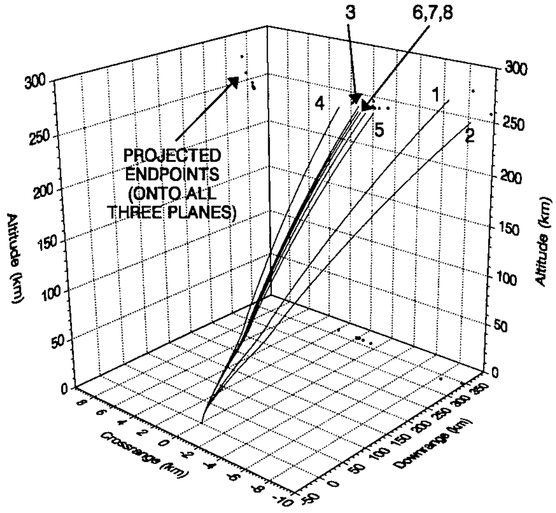

Figure 2. Interceptor Trajectories

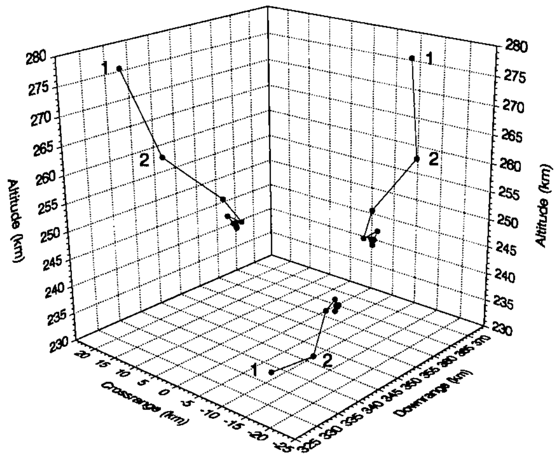

Figure 3. Magnification of the Endpoints.

The corresponding control perturbations for this trajectory are plotted in Figs. (4) and (5), and the intercept time perturbations are plotted in Figs. (6) and (7). In Fig. (7), the flat portion of the curve occurs when the velocity to-be-gained guidance is used, since the intercept time is held constant with that scheme. The portion of the curve between about 64.1 and 64.6 s is when MCV guidance is active. After an initial adjustment into the region of -0.8 s, a damped oscillation ensues, as MCV guidance converges on the appropriate intercept time. Similar oscillations are seen in the controls in Fig. (5).

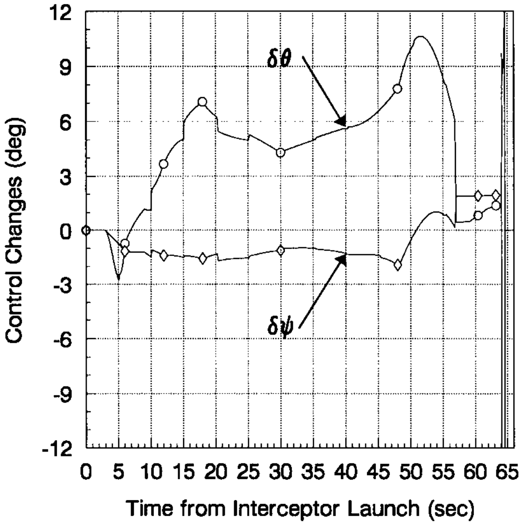

Figure 4. Control Perturbations.

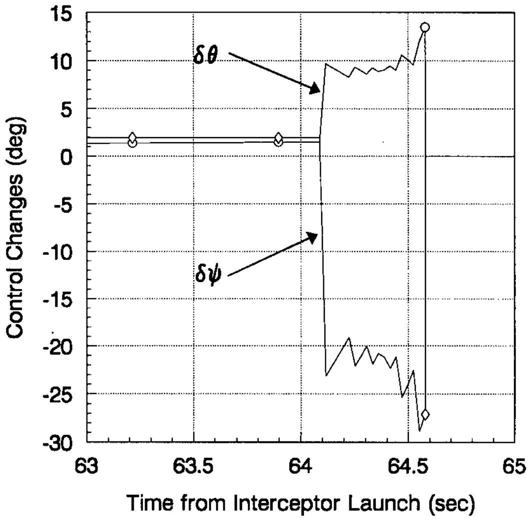

Figure 5. Zoom-in on MCV Portion of Control Perturbation Plot.

Finally, the total angle of attack, α_T, is shown in Fig. (8). Notice the relatively benign angle of attack history, especially in the neighboring optimal control segment, even for the large adjustments in the PIPs. The target update times are at 5 s, 10 s, etc. This low angle of attack history is typical for neighboring optimal control. Another performance indicator is burnout velocity, which typically changes by less that one percent relative to the open loop optimal trajectories, when large perturbations are imposed. Since it knows the future optimal path, as well as the current, the neighboring optimal control method reacts with mild corrections to

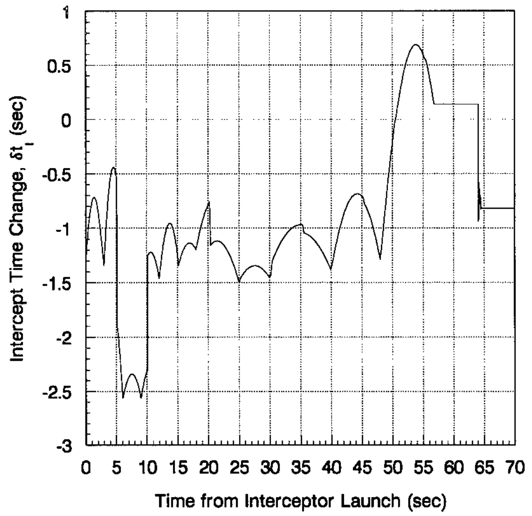

Figure 6. Intercept Time Perturbations.

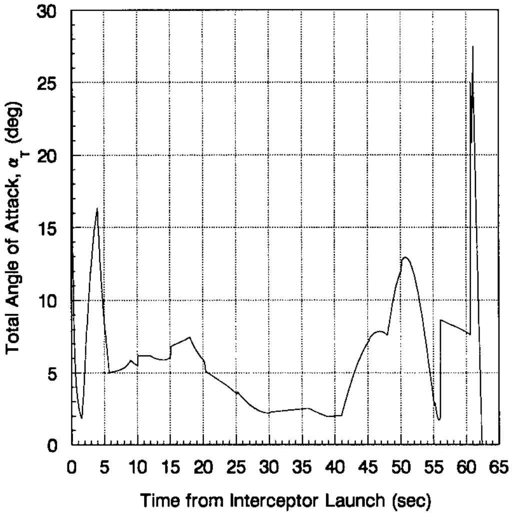

Figure 8. Resultant Total Angle of Attack.

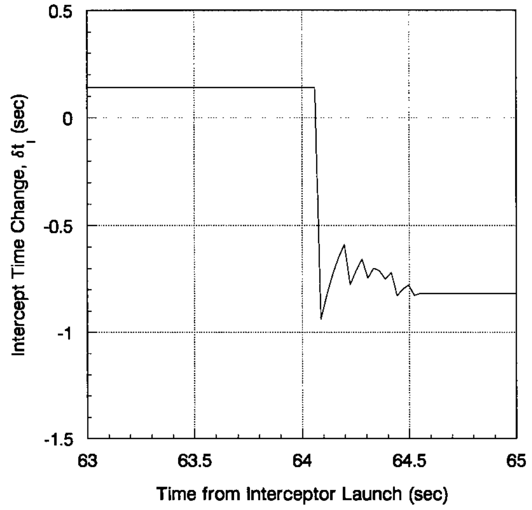

Figure 7. Zoom in on MCV Portion of the Intercept Time Perturbations.

large PIP changes. More precisely, it models the whole trajectory as one entity, so that current control perturbations take into account the future path. This is something that simpler "local" guidance schemes (that is, those that are based on satisfying certain equations that are essentially functions only of the current state and the target point) cannot do. For that reason, the local guidance schemes tend to react more strongly to large PIP changes.

When all of the interceptor errors are turned off (that is, there are zero heading and velocity errors, and the interceptor has perfect knowledge of its own state), the RMS error from this guidance scheme over 100 Monte Carlo runs is 35 m (relative to the last known target state). Even very small interceptor errors would yield propagated errors to the PIP point on the order of about 1 km. Taking the RSS of this and 35 m yields an increase of less than 1 m to the 1 km. In other words, the contribution by the guidance scheme to the total errors that must be taken out by the midcourse corrections and the terminal homing is essentially zero. The residual errors in the target state estimate add even more errors to this. Having the boost-phase guidance contribute nearly zero errors to the total error budget is an important goal, since the lower the total error is, the higher the probability of successful intercept.

As a final note, it should be mentioned that one reference trajectory and one set of gains were used for the whole trajectory. The reference trajectories are available at every 5 km by 5 km in downrange and altitude; so, the initial PIP is rounded to the nearest 5 km by 5 km to select the reference trajectory. This is an arbitrary choice of gridding that works; all indications are that less dense gridding will work just as well. The gains are even more sparsely populated. They are required only every 50 to 100 km in downrange and altitude (more densely at the lower altitudes). In fact, only a total of 10 to 12 sets of gains are needed for the whole battle space. Then, at fire control time, only one reference trajectory with accompanying reference controls and one set of gains (about 2 Kbytes of data) need to be loaded onto the interceptor flight computer.

Conclusions

A hybrid guidance scheme, which mainly relies upon neighboring optimal control guidance, with Lambert-based guidance schemes used at the end of the burn to zero out residual errors, has been shown to be a robust and effective guidance method for tactical ballistic missile exoatmospheric interceptors. While generating the reference trajectories and gains for the neighboring optimal control method is computationally intensive, this is all accomplished before the deployment of the missile, and supplied to the fire control computer and to the interceptor missile in the form of look-up tables. The actual fire control solution process and onboard guidance calculations are computationally simple. Furthermore, the data storage requirements are light.

The method handles large variations in the predicted intercept points, while yielding essentially zero guidance errors when compared to all error contributions. Furthermore, the method tends to preserve performance, as indicated by such measures as relatively low angles of attack and small changes in burnout velocities relative to open-loop optimal levels, even when large intercept point changes are imposed by target state updates received from the exogenous radar.

In closing, the ability of the method to handle large intercept point corrections has at least two important system-level implications. First, this allowance for large intercept point corrections means that the radar does not need to revisit the target at a high frequency, so the radar resources can be spread to other targets and other priorities. Second, this permits the interceptor to be launched early relative to target detection (that is, while large target state estimate velocity errors remain), so that more battle space can be attained, and more assets can be defended.

References

[1] Bryson, A. E., and Ho, Y.-C., "Neighboring Extremals and the Second Variation," *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 177-211.

[2] Kelly, H. J., "An Optimal Guidance Approximation Theory," *IEEE Transactions on Automatic Control*, October, 1964, pp. 375-380.

[3] Kumar, R. R., Seywald, H., and Cliff, E. M., "Near Optimal 3-D Guidance Against a Maneuvering Target," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1989, pp. 482-495.

[4] Seywald, H., and Cliff, E. M., "Neighboring Optimal Control Based Feedback Law for the Advanced Launch System," *Journal of Guidance, Control, and Dynamics*, Vol 17, No. 6, 1994, pp. 1154-1162.

[5] Battin, R. H., "Solving Lambert's Problem," *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, New York, 1987, pp. 295-342.

[6] Battin, R. H., "Powered Orbital Transfer Maneuvers," *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, New York, 1987, pp. 550-558.

[7] Lawton, J. A., and Jesionowski, R. J., "Velocity-to-be-Maintained Guidance," *Spaceflight Mechanics 1994*, Vol. 87, *Advances in the Astronautical Sciences*, American Astronautical Society, 1994.

[8] Bushong, P. M., and Lawton, J. A., "Fast Solution to an Optimal Trajectory Problem," *Spaceflight Mechanics 1994*, Vol. 87, *Advances in the Astronautical Sciences*, American Astronautical Society, 1994.

[9] Martell, C. A., and Lawton, J. A., "Adjoint Variable Solutions via an Auxiliary Optimization Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, 1995, pp. 1267-1272.

[10] Lawton, J. A., Jesionowski, R. J., and Zarchan, P., "Comparison of Four Filtering Options for a Radar Tracking Problem," Paper No. 16-01, *AIAA/BMDO Technology Readiness Conference,* 1997.