JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 19, No. 2, March–April 1996

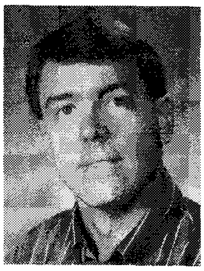
Dynamic Programming Approach for Burnout-to-Apogee Guidance of Precision Munitions

Clark R. Dohrmann, G. Richard Eisler, and Rush D. Robinett
Sandia National Laboratories, Albuquerque, New Mexico 87185

A method is presented for burnout-to-apogee munitions guidance that compensates for launch offsets from a nominal, ground-to-ground trajectory. The munition is assumed to be configured for low-cost, externally received navigation updates that are only available up to the time of apogee. Using a novel technique, a guidance problem with an unspecified final time is transformed to a form solvable by the method of dynamic programming. A set of feedback gains is generated for times between motor burnout and apogee to compensate for launch-induced offsets. Corrected ballistic trajectories from apogee are obtained by including a quadratic term of target miss distance in the problem formulation. A single set of gains is shown to be robust for a range of launch offsets.

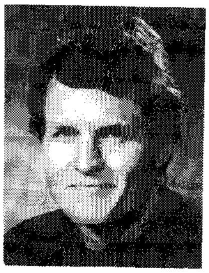
Nomenclature

A_k, B_k = state transition matrices at the kth time step
C, d, E, F = sensitivity matrices
$C^s_{D_0}$ = drag coefficient used to simulate modeling errors
C_L, C_S, C_D = lift, side-force, and drag coefficients
$C_{L_\alpha}, C_{D_0}, k_D$ = Mach number dependent coefficients used in aerodynamic model
c = speed of sound
e_D = constant error term in drag coefficient
g = acceleration of gravity
h = time step for discretization of input
K_k = feedback gain matrix at the kth time step
m = mass of munition after motor burnout
r = radius of Earth
S = reference area of munition
T, t_f = times of impact for nominal and controlled trajectories
t_N = time at apogee for nominal trajectory
$u(t), x(t)$ = input and state for guidance problem
V = magnitude of velocity vector
W_k, H_{2k}, H_{3k} = matrices used in dynamic programming algorithm at the kth time step
x_c, y_c, z_c = downrange, crossrange, and altitude coordinates



Clark Dohrmann received his B.S. and M.S. degrees in mechanical engineering from the Ohio State University in 1985 and 1986, respectively. He joined Sandia National Laboratories in 1987 and worked as a staff member in the Engineering Analysis Department for three years. In 1990, Clark returned to Ohio State's Mechanical Engineering Department on a company-sponsored educational program and received his Ph.D. in 1992. Clark is currently a Senior Member of the Technical Staff at Sandia working in the Structural Dynamics Department. His areas of technical interest include structural and spacecraft dynamics, system identification, robotics, stability analysis, inverse problems, and controls.

Rick Eisler received his B.S. at Boston University, M.S. at Iowa State, and Ph.D. from the University of Texas at Austin, all in aerospace engineering. He has been a member of the Technical Staff at Sandia National Laboratories since 1974. His current research interests include optimal control for missile guidance, flexible structure robotics, and optimization of process models. He has served on the AIAA Guidance, Navigation, and Control Committee and is an Associate Fellow of AIAA.



Rush D. Robinett received his B.S. (1982) and Ph.D. (1987) degrees from Texas A&M University and M.S. (1984) degree from the University of Texas at Austin in Aerospace Engineering. At present, he is a Distinguished Member of the Technical Staff at Sandia National Laboratories in the Aerospace Systems Development Center. His current work includes dynamics, stability, and optimal and nonlinear control of reentry systems, spacecraft, and flexible robotic manipulators. He is a Senior Member of AIAA.

Received June 13, 1995; revision received Oct. 27, 1995; accepted for publication Nov. 9, 1995.

x_{ct}, y_{ct}, z_{ct}	= coordinates of target
$\bar{x}(t)$	= ballistic trajectory from t_N to t_f
$\hat{x}(t)$	= nominal trajectory
α, β	= angles of attack and sideslip
γ, ψ	= vertical and horizontal heading angles
λ	= parameter used in definition of guidance problem
ρ	= density of atmosphere

I. Introduction

THE use of aided navigation via satellite updates promises unparalleled, cost-effective, and accurate systems for both civilian and defense purposes. Contemporary defense systems routinely integrate conventional inertial measurement unit (IMU) technology with global positioning system (GPS) updating to produce impressive targeting results. This has been most notable for standoff munitions, such as cruise missiles and guided bombs.[1-3] A recent study by Singh et al.[4] was directed at fabricating the IMU/GPS combination in solid-state form able to withstand artillery gun setback loads. The present study seeks to capitalize on the cost-effectiveness issue by formulating a guidance algorithm for a field munition solely reliant on offboard navigation aids. A major drawback of using only offboard sources, which this work addresses, is that the update signals can be obscured by electronic jamming. Given that this issue is addressed here, it is further assumed that retrofits of existing, extensively used, ballistic munitions with low-cost navigation and actuation hardware add ons would be justified by significantly lowering the cost of destroying defended targets via increased accuracy.

Precision and accuracy are the two major issues in munitions targeting. Important factors to both are the effects of launch offsets and winds. Although this study is concerned primarily with the former, winds are included in one of the aerodynamic models and their effects summarized. The method presented herein remains essentially the same regardless of the aerodynamic model used.

As already stated, jamming of aided navigation signals by the adversary must be taken into account for viability of the proposed scheme. If not, the current IMU/GPS style integration must be relied on, which entails a significant cost penalty. The approach taken here is to apply guidance during the initial phase of flight where jamming effects are minimal and to complete all guidance updates and reestablish nominal ballistic accuracy before descent to the target (Fig. 1). To date, previous work for burnout-to-apogee guidance appears nonexistent.

The flight regime for this munition is below 50,000 ft and supersonic. In this regime aerodynamic and gravity forces are of comparable magnitudes, and both are changing significantly along the flight path. As such, none of the usual simplifying approximations to the equations of motion for atmospheric flight can be made,[5] and an analytic solution can not be found. Nonlinear programming (NP) solutions are certainly viable alternatives to generate control histories for a problem involving both boundary conditions and interior constraints. However, these solutions would be of an open-loop, rather than a feedback, form. NP solutions would need to be computed for a large number of different trajectories, with highly involved interpolations required to keep this number manageable and to span a useful guidance space.

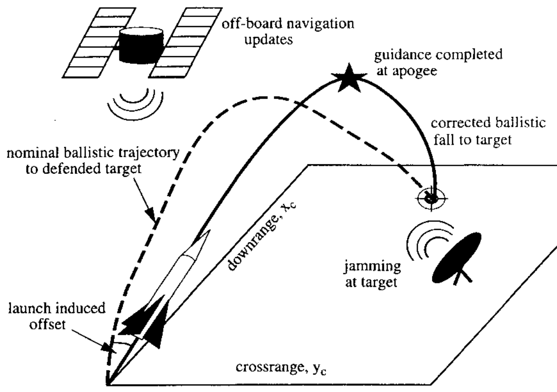

off-board navigation updates
guidance completed at apogee
corrected ballistic fall to target
nominal ballistic trajectory to defended target
downrange, x_c
jamming at target
launch induced offset
crossrange, y_c

Fig. 1 Proposed guided munition scenario.

As an alternative, a feedback control scheme based on the method of dynamic programming is presented in this study. Dynamic programming offers the benefits of 1) retaining all relevant physics of the problem in computing guidance commands, 2) accommodating a variety of different launch offsets, and 3) providing a set of feedback gains for optimal return from any point in the guided portion of the trajectory. The price for this capability is the necessity of computing and storing a matrix of feedback gains at as many update points as are necessary to provide acceptable performance. Given that contemaporary field munition operations require online targeting computations to account for such variables as weather conditions, it is felt that the proposed computational requirements would be compatible with such exercises. This observation is supported by experience with calculating feedback gains on a workstation and is a result of the computational efficiency of the dynamic programming algorithm. In addition, to minimize the attendant actuation hardware costs, the guidance scheme would ideally generate control efforts of minimal magnitude. This would allow considerable latitude in the use of internal[6] vs external methods for attitude control.

In the interest of simplicity and to highlight the salient features of the proposed algorithm, a retrofitted munition capable of independent lift and side-force generation is assumed. (This method can be generalized to an oriented lift-plane control system as well.) Representative aerodynamics for a multiple launch rocket system (MLRS)[7] are used. The physical model is developed initially, followed by the guidance control problem formulation and dynamic programming solution. Finally, simulation results are shown for a single set of feedback gains applied to two sets of launch errors. The effects of modeling errors and unaccounted winds are also presented.

II. Physical Model

In this study an artillery rocket is assumed to have launch and propulsive phases that are unguided. Guidance commences after motor burnout. The physical model for the guided portion of the rocket flight is that of an unpowered, point mass flying over a spherical, nonspinning Earth at an altitude that is small with respect to the radius of the Earth. The equations of motion are

$$\begin{aligned} \dot{x}_c &= V \cos\gamma \cos\psi & \dot{V} &= -\frac{\rho V^2 C_D S}{2m} - g \sin\gamma \\ \dot{y}_c &= V \cos\gamma \sin\psi & \dot{\gamma} &= \frac{\rho V^2 C_L S}{2mV} - \left[\frac{g}{V} - \frac{V}{r}\right] \cos\gamma \qquad (1) \\ \dot{z}_c &= V \sin\gamma & \dot{\psi} &= \frac{\rho V^2 C_S S}{2mV \cos\gamma} \end{aligned}$$

where the overdot in terms in Eq. (1) denotes the time derivative. The lift and side-force coefficients C_L and C_S are assumed linear over a small aerodynamic angle range and related to angles of attack α and sideslip β via $C_L = C_{L_\alpha}\alpha$ and $C_S = C_{L_\alpha}\beta$, assuming an aerodynamically symmetric vehicle. Drag variation with lift and side force are fitted as a quadratic function according to $C_D = C_{D_0} + k_D[C_L^2 + C_S^2]$, where C_{D_0} and k_D are Mach number dependent and plotted with C_{L_α} in Fig. 2. Density and speed of sound are tabulated for a standard atmosphere and shown in Fig. 3. This model complexity was felt necessary because the munition will be flying in the transonic to moderate-supersonic regime where aerodynamic changes are significant. More complex aerodynamic models that account for current weather conditions could also be accommodated.

III. Guidance Problem

The input u for the guidance problem is chosen as

$$u = [\dot{C}_L \quad \dot{C}_S]^T \qquad (2)$$

where the superscript T denotes the matrix transpose. The physical model given by Eq. (1) along with Eq. (2) can be expressed together in the standard form

$$\dot{x}(t) = f[x(t), u(t), t] \qquad x(t_1) = x_1 \qquad (3)$$

where t_1 is the time at motor burnout and x is the state defined as

$$x = [x_c \quad y_c \quad z_c \quad V \quad \gamma \quad \psi \quad C_L \quad C_S]^T \qquad (4)$$

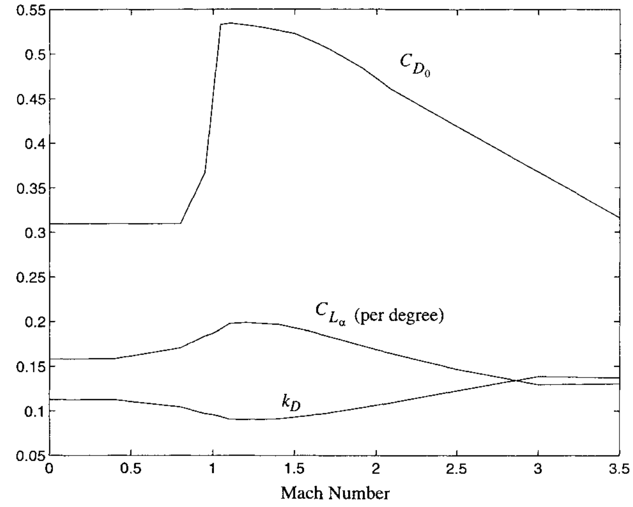


Fig. 2 Aerodynamic data.

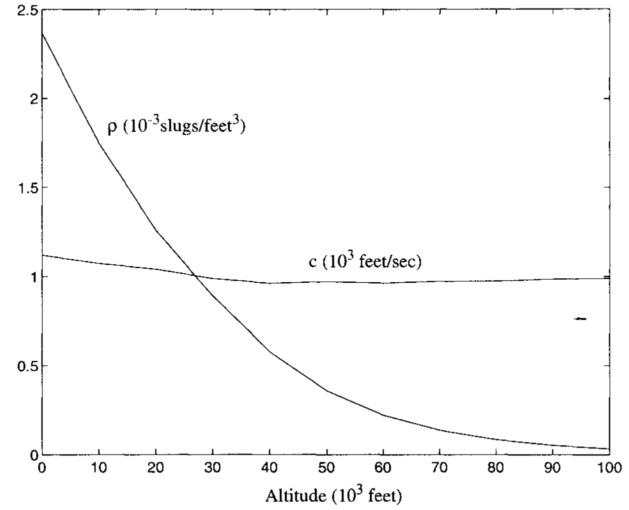


Fig. 3 Atmospheric data.

The nominal trajectory $\hat{x}(t)$ corresponds to the input $u(t) = 0$ and the initial conditions

$$\hat{x}(t_1) = [\hat{x}_c(t_1) \quad \hat{y}_c(t_1) \quad \hat{z}_c(t_1) \quad \hat{v}(t_1) \quad \hat{\gamma}(t_1) \quad \hat{\psi}(t_1) \quad 0 \quad 0]^T \tag{5}$$

It is assumed that the nominal trajectory hits the target at the final time T. That is,

$$\hat{x}_c(T) = x_{ct} \qquad \hat{y}_c(T) = y_{ct} \qquad \hat{z}_c(T) = z_{ct} \tag{6}$$

where (x_{ct}, y_{ct}, z_{ct}) are the coordinates of the target.

In practice, it is unlikely that the nominal trajectory will hit the target because the correct initial conditions cannot be enforced exactly and because of limitations in the accuracy of the aerodynamic model. The approach taken herein is to control the portion of the flight between the times t_1 and $t_N (t_N < T)$ in such a manner that the corrected ballistic trajectory of the munition is directed toward the target. In the present study, t_N is set equal to the time at apogee. It is assumed for $t \geq t_N$ that navigation signals are jammed and the lift and side-force coefficients are set to zero.

The guidance problem is stated as follows. Find the input $u(t)$ that minimizes the functional

$$\Gamma(u) = \lambda[(x_c(t_f) - x_{ct}) \quad (y_c(t_f) - y_{ct})]\begin{bmatrix} x_c(t_f) - x_{ct} \\ y_c(t_f) - y_{ct} \end{bmatrix} + \int_{t_1}^{t_f} [u(t)]^T[u(t)]\,\mathrm{d}t \tag{7}$$

subject to Eq. (3) and the constraints

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{for} \qquad t = t_1 \qquad \text{and} \qquad t_N \leq t \leq t_f \tag{8}$$

where t_f denotes the time of impact. The term premultiplied by the scalar $\lambda > 0$ in Eq. (7) is equal to the square of the distance between the point of impact and the target. Thus, λ directly affects the tradeoff between accuracy and control effort as measured by the integral term in Eq. (7). For $\lambda = 0$, the input is zero for all times resulting in a ballistic trajectory. In the limit as $\lambda \to \infty$, the resulting trajectory hits the target. The preceding statement implicitly assumes that the physical model and measurements of the initial conditions are exact. This assumption clearly is not valid in practice; therefore, care must be taken in choosing an appropriate value for λ. The topic of choosing λ is discussed further in Sec. VI. Equation (8) states that the lift and side-force coefficients are zero initially and zero during the portion of the trajectory between times t_N and t_f.

Notice that the time at impact t_f (final time) in Eq. (7) is unspecified and need not equal the time at impact T for the nominal trajectory. In the next section, the guidance problem is transformed to a standard form in which the final time is specified and the constraints given by Eq. (8) are satisfied automatically.

IV. Transformation of Guidance Problem

It proves useful to introduce the ballistic trajectory $\bar{x}(t)$ from t_N to t_f corresponding to the input $u(t) = 0$ and initial conditions at $t = t_N$ of

$$\bar{x}(t_N) = [\bar{x}_c(t_N) \quad \bar{y}_c(t_N) \quad \bar{z}_c(t_N) \quad \bar{V}(t_N) \quad \bar{\gamma}(t_N) \quad \bar{\psi}(t_N) \quad 0 \quad 0]^T \tag{9}$$

Defining t_f as the time at impact for $\bar{x}(t)$ and

$$\Delta t_f = t_f - T \tag{10}$$

$$\Delta x(t_N) = \bar{x}(t_N) - \hat{x}(t_N) \tag{11}$$

the following linear approximations in Δt_f and $\Delta x(t_N)$ are obtained:

$$\bar{z}_c(t_f) = z_{ct} + C\Delta x(t_N) + d\Delta t_f \tag{12}$$

$$\begin{bmatrix} \bar{x}_c(t_f) \\ \bar{y}_c(t_f) \end{bmatrix} = \begin{bmatrix} x_{ct} \\ y_{ct} \end{bmatrix} + E\Delta x(t_N) + F\Delta t_f \tag{13}$$

The last two columns of the sensitivity matrices C and E in Eqs. (12) and (13) can be set equal to zero since the last two elements of $\Delta x(t_N)$ are zero. All of the sensitivity matrices in Eqs. (12) and (13) can be determined from numerical integration of the aerodynamic model along with finite difference calculations. Another option is to use an automatic differentiation program to obtain these terms. In this study, finite difference calculations were used.

Setting $\bar{z}_c(t_f)$ equal to z_{ct} in Eq. (12) leads to

$$\Delta t_f = -\frac{C\Delta x(t_N)}{d} \tag{14}$$

Combining Eqs. (13) and (14), one obtains

$$\begin{bmatrix} \bar{x}_c(t_f) - x_{ct} \\ \bar{y}_c(t_f) - y_{ct} \end{bmatrix} = G\Delta x(t_N) \tag{15}$$

where

$$G = E - (1/d)FC \tag{16}$$

Equation (15) is a key relation that is used shortly to transform the guidance problem to an equivalent standard form.

Returning for a moment to the controlled portion of the trajectory, the input is assumed to be discretized for any time $t_1 \leq t \leq t_N$ as

$$u(t) = u_k$$

$$\text{for} \qquad t_k \leq t < t_{k+1} \qquad \text{and} \qquad k = 1, \ldots, N-1 \tag{17}$$

where

$$t_k = t_1 + h(k-1) \tag{18}$$

and

$$h = \frac{t_N - t_1}{N-1} \tag{19}$$

The input is set equal to zero for $t \geq t_N$.

From its definition [see Eq. (9)], it is clear that $\hat{x}(t)$ satisfies Eq. (8) for $t_N \leq t \leq t_f$. Thus, without loss of generality, one can consider $x(t) = \hat{x}(t)$ for $t_N \leq t \leq t_f$. Consequently, substitution of Eqs. (15) and (17) into Eq. (7) yields

$$\Gamma(u_k) = \lambda \tilde{x}_N^T G^T G \tilde{x}_N + h \sum_{k=1}^{N-1} u_k^T u_k \tag{20}$$

where

$$\tilde{x}_k = x(t_k) - \hat{x}(t_k) \qquad k = 1, \ldots, N \tag{21}$$

To satisfy Eq. (8) at $t = t_N$, the seventh and eighth elements of $\tilde{x}_N$ must equal zero. Thus, for an input of the form given by Eq. (17), integration of Eq. (2) between the limits of t_{N-1} and t_N implies that

$$u_{N-1} = -D\tilde{x}_{N-1} \tag{22}$$

where

$$D = \frac{1}{h}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

Linearizing the dynamics about the nominal trajectory, one obtains

$$\tilde{x}_{k+1} = A_k \tilde{x}_k + B_k u_k \qquad k = 1, \ldots, N-1 \tag{24}$$

where the state transition matrices A_k and B_k can be determined with recourse to a numerical integration scheme (see Appendix). Substituting Eqs. (22) and (24) with $k = N - 1$ into Eq. (20) yields

$$\Gamma(u_k) = \tilde{x}_{N-1}^T W_{N-1} \tilde{x}_{N-1} + h \sum_{k=1}^{N-2} u_k^T u_k \tag{25}$$

where

$$W_{N-1} = \lambda (A_{N-1} - B_{N-1}D)^T G^T G (A_{N-1} - B_{N-1}D) + hD^T D \tag{26}$$

In summary, the original guidance problem has been transformed from minimization of a functional with an unspecified final time to minimization of the function given by Eq. (25) with a specified final time t_N.

V. Dynamic Programming Solution

Minimization of Eq. (25) subject to Eq. (24) can be accomplished in a straightforward manner using the method of dynamic programming. Detailed explanations of the theory and applications of dynamic programming are available in several texts.[8,9] The algorithm for the guidance problem of interest is summarized in the following two-step procedure.

1) Starting with W_{N-1} given by Eq. (26), calculate W_k recursively for $k = N - 2, \ldots, 1$ from the equation

$$W_k = A_k^T W_{k+1} A_k - H_{2k} H_{3k}^{-1} H_{2k}^T \tag{27}$$

where

$$H_{2k} = A_k^T W_{k+1} B_k \tag{28}$$

$$H_{3k} = h \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + B_k^T W_{k+1} B_k \tag{29}$$

In the process, store the feedback gain matrices

$$K_k = H_{3k}^{-1} H_{2k}^T \tag{30}$$

All of these calculations are performed prior to launch of the munition.

2) Calculate the inputs $u_1, \ldots, u_{N-2}$ online during ascent from the equation

$$u_k = -K_k \tilde{x}_k \tag{31}$$

and u_{N-1} from Eq. (22). Equations (27–31) are equivalent to Eqs. (3.7–60) and (3.7–63) of Ref. 9. The $\tilde{x}_k$ terms in Eqs. (22) and (31) are calculated by substituting the measured and nominal trajectories into Eq. (21). The constraint of zero lift and side-force coefficients at the beginning of the trajectory, time t_1, is satisfied by setting the elements in rows 7 and 8 of $\tilde{x}_1$ equal to zero. As the inputs are calculated, C_L and C_S are determined by integrating Eq. (2). The result is

$$\begin{bmatrix} C_L(t_{k+1}) \\ C_S(t_{k+1}) \end{bmatrix} = \begin{bmatrix} C_L(t_k) \\ C_S(t_k) \end{bmatrix} + hu_k \qquad k = 1, \ldots, N-1 \tag{32}$$

Between the times t_k and t_{k+1}, coefficients C_L and C_S vary linearly.

An attractive feature of the control law given by Eq. (31) is that it provides the optimal return from any point in the guided portion of the trajectory (see p. 281 of Ref. 9). That is, the control law is optimal in the sense that it minimizes the function

$$\Gamma_k(\tilde{x}_k, u_i) = \tilde{x}_{N-1}^T W_{N-1} \tilde{x}_{N-1} + h \sum_{i=k}^{N-2} u_i^T u_i \tag{33}$$

subject to Eq. (24) for all values of $\tilde{x}_k$, not just the one that is on the optimal trajectory for the original launch offset $\tilde{x}_1$. Thus, if at times intermediate to t_1 and t_N, the trajectory is bumped away from the optimal owing to modeling, measurement, or actuation errors, the control law is still optimal in the sense stated earlier. Although this feature of the control law is attractive, success of the algorithm ultimately rests in having adequate physical models and measurement systems. As a final note, the control law presented is equivalent to what one would obtain using discrete linear-quadratic guidance[9] owing to the use of linearized dynamics, a nominal trajectory that hits the target, and the quadratic form of the function given by Eq. (25).

VI. Results

The MLRS-type vehicle for this study has a burnout mass of 14.25 slugs and a reference area of 0.4356 ft^2. Boundary conditions for the nominal trajectory are shown in Table 1. Both the nominal and controlled trajectories were obtained by numerically integrating the nonlinear differential equations given by Eq. (3) using a fixed-step, fourth-order, Runge–Kutta method.[10] Apogee altitude is approximately 27,600 ft and Mach number variations are from 1.09 to 2.33. The nominal trajectory was discretized in the time domain with 500 time steps. The accuracy of the numerical integration was checked by comparing the calculated trajectory with one obtained using 1000 time steps. The maximum distance at any time between the two calculated trajectories was less than 0.01 ft.

Feedback gains for the control scheme were generated by choosing $\lambda/h = 1 \times 10^{-6}$. The states $x_1, \ldots, x_N$ $(N = 209)$ for the controlled portion of the trajectory $(t_1 \leq t \leq t_N)$ were obtained using the inputs generated from Eqs. (31) and (22). The corrected ballistic trajectory for $t > t_N$ was obtained by setting both the lift and side-force coefficients to zero.

Two different sets of launch offsets were considered: case 1 with γ, ψ offsets of 0.5 and 1 deg from nominal at burnout and case 2 with

Table 1 Boundary conditions for the nominal trajectory

Variable	Initial value	Final value
x_c, ft	0	92,952
y_c, ft	0	0
z_c, ft	7390	0
V, ft/s	2532	1222
γ, deg	35	−57.64
ψ, deg	0	0

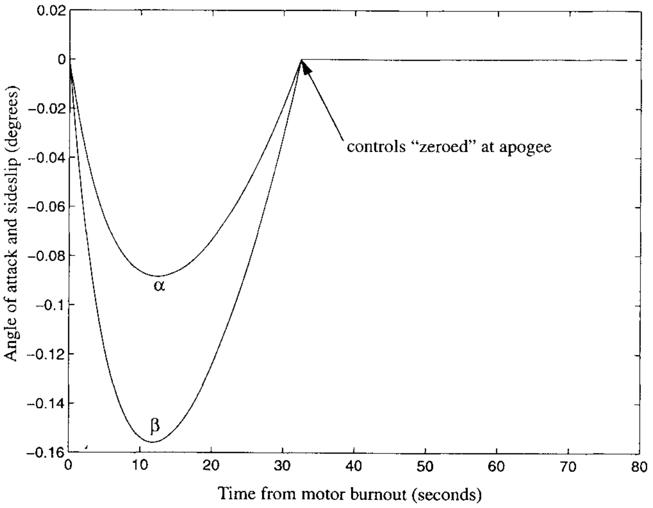


Fig. 4 Angle of attack α and sideslip β time histories for case 1.

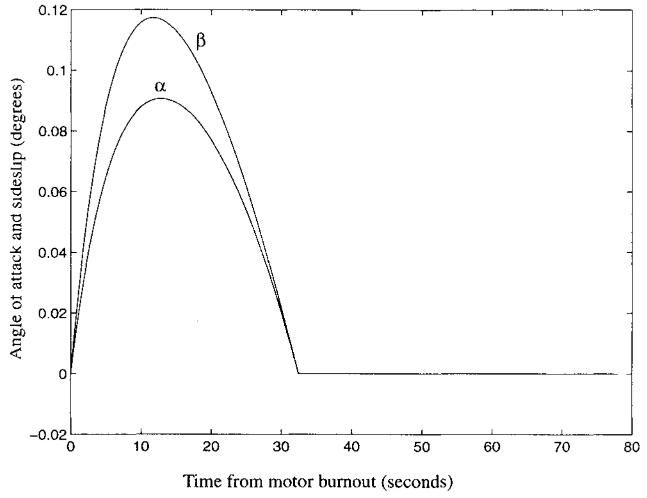


Fig. 5 Angle of attack α and sideslip β time histories for case 2.

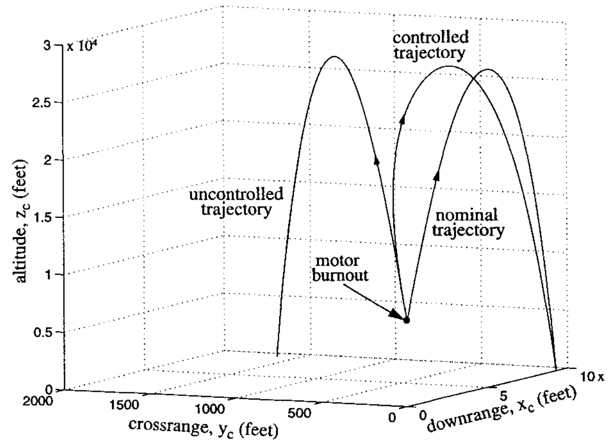


Fig. 6 Case 1 trajectory comparisons.

Table 2 Case 1 downrange and crossrange errors for modeling errors in C_{D_0}

C_{D_0} error	Error (uncontrolled), ft		Error (controlled), ft	
e_D	Downrange	Crossrange	Downrange	Crossrange
−0.02	1423	1647	157	0.6
−0.01	897	1638	87	0.4
0	377	1629	−0.2	0.3
0.01	−136	1620	−105	0.3
0.02	−645	1611	−227	0.4

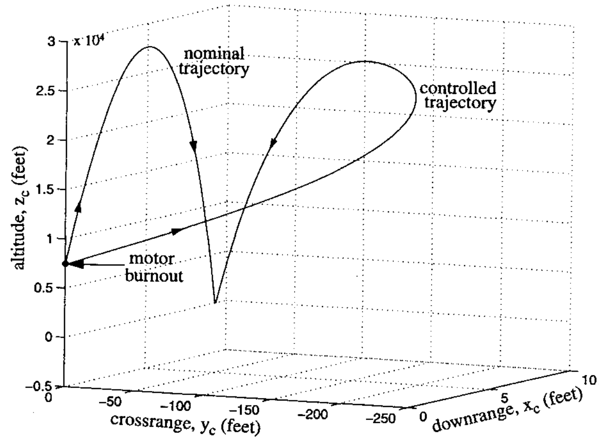


Fig. 7 Case 2 trajectory comparisons.

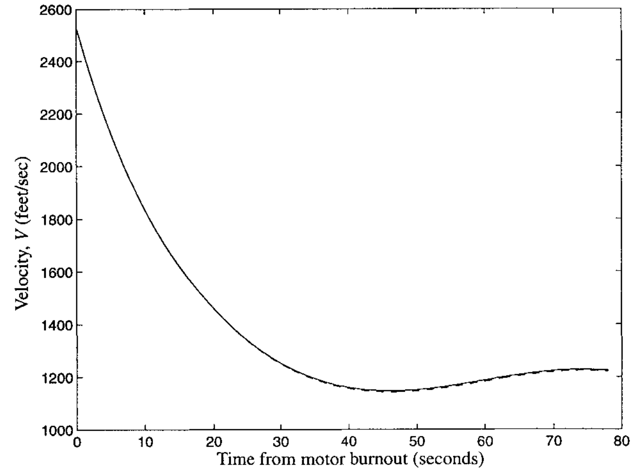


Fig. 8 Velocity time histories for cases 1 and 2: ——, case 1 and - - - -, case 2.

γ, ψ offsets of −0.5 and −0.75 deg. Figures 4 and 5 show the aerodynamic control angles for the two cases computed using the same set of feedback gains. These angles were obtained from the equations $\alpha = C_L/C_{L_\alpha}$ and $\beta = C_S/C_{L_\alpha}$ (see Sec. II) using the calculated values for C_L and C_S. In both cases the angular magnitudes are well below 1 deg. Figures 6 and 7 show the point mass trajectories for the two cases. In Fig. 6, the controlled and uncontrolled trajectories, as well as the nominal, are plotted. Note that even for the small γ, ψ offsets at burnout, the target miss distance, as shown by the uncontrolled trajectory, can be substantial (>1600 ft). The controlled trajectory for case 2 is plotted on an expanded crossrange scale in Fig. 7. The miss distances for cases 1 and 2 were approximately 0.4 and 1.2 ft, respectively. Velocity and horizontal heading-angle time histories for the two cases are shown in Figs. 8 and 9. Note that the horizontal heading angle is a constant after apogee in both cases.

Examples are also provided to show the effects of modeling errors on the performance of the guidance algorithm. The effects of a parametric modeling error in the drag coefficient are shown in Table 2. The results were obtained using the same set of feedback gains and launch offsets that were used for case 1. The only difference is that the drag coefficient used in the simulations, $C^s_{D_0}$, was calculated from the equation

$$C^s_{D_0} = (1 + e_D)C_{D_0} \tag{34}$$

where e_D is a constant error term. Notice from the table that the errors for the guided trajectory are primarily in the downrange direction. Although the performance is degraded, it is clear that the miss distances for the controlled trajectories remain significantly smaller than those for the uncontrolled ones.

Table 3 Case 1 downrange and crossrange errors for unaccounted headwinds

Headwind speed, mph	Error (uncontrolled), ft		Error (controlled), ft	
	Downrange	Crossrange	Downrange	Crossrange
−3.0	657	1579	485	4.2
−1.5	518	1603	243	2.3
0	377	1629	−0.2	0.3
1.5	235	1655	−244	−1.7
3.0	92	1682	−488	−3.7

Table 4 Case 1 downrange and crossrange errors for unaccounted crosswinds

Crosswinds speed, mph	Error (uncontrolled), ft		Error (controlled), ft	
	Downrange	Crossrange	Downrange	Crossrange
−3.0	365	−1605	−8.5	−1176
−1.5	387	12	−2.6	−588
0	377	1629	−0.2	0.3
1.5	336	3245	−1.5	588
3.0	263	4858	−6.4	1176

Table 5 Case 1 downrange and crossrange errors for modeling errors and unaccounted winds when guided portion of trajectory is extended

c_{D_0} error e_D	Headwind, mph	Crosswind, mph	Downrange, ft	Crossrange, ft
0	0	0	−2.1	−2.3
0.02	0	0	−8.4	−11
0	3	0	−147	−8.3
0	0	3	−28	166
0.02	3	3	−190	125

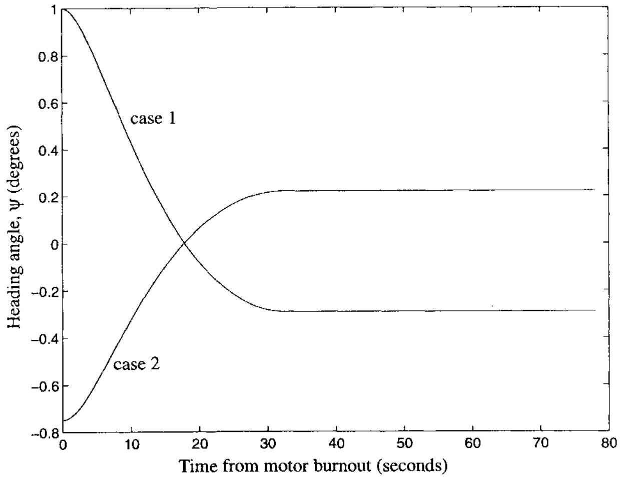


Fig. 9 Horizontal heading-angle time histories for cases 1 and 2.

The effects of unaccounted headwinds and crosswinds are shown in Tables 3 and 4. Again, the results were obtained using the same set of feedback gains and launch offsets that were used for case 1. The difference is that the aerodynamic model was modified to include the effects of steady winds. Not surprisingly, headwinds lead to larger downrange errors whereas crosswinds result in larger crossrange errors. The guided trajectory results, however, generally remain more accurate than the unguided ones. Modifying the guidance algorithm to account for current wind conditions can be done in a straightforward manner and would lead to more impressive targeting results.

In the absence of jamming of aided navigation signals, the duration of the guided portion of the trajectory can be extended. This is done simply by increasing the value of N and, thus, t_N. In the preceding examples, N was set equal to 209, which corresponded to the time at apogee. The effects of increasing the value of N to 475 for $\lambda/h = 1 \times 10^{-6}$ is shown in Table 5 for a variety of modeling errors and unaccounted winds. The performance is improved by extending the duration of the guided portion of the trajectory.

As was already mentioned, selection of the scalar parameter λ [see Eq. (7)] is an important consideration in the practical implementation of the proposed guidance scheme. One option is to choose a relatively large value for λ such that the corrected ballistic trajectory impact is very close to the target. Indeed, such an approach would lead to desirable results in computer simulations. In practice, however, choosing a large value for λ could lead to undesirable performance of the control system.

In the limit as $\lambda \to \infty$, the feedback gains [see Eq. (30)] asymptotically approach a finite fixed value. As the time step for discretization h becomes very small, the gains for the portion of the trajectory nearby $t = t_N$ become very large. A similar situation is present with proportional navigation[11] schemes where feedback gains become infinite as the time to go approaches zero. One consequence of large gains is that disturbances caused by modeling or measurement errors can lead to undesirably large control inputs near $t = t_N$.

In selecting an appropriate value for λ, one must balance the competing goals of targeting accuracy and control system insensitivity to disturbances. One starting point is to choose λ as small as possible such that the specified targeting requirements are met for the anticipated range of launch offsets. The value of λ could then be increased depending on the accuracy of the measurement system and physical model. Although the topic of choosing the value of λ is not addressed in complete detail by this study, it is hoped that some of the important issues involved have been conveyed.

VII. Conclusions

A method to correct for launch offsets of precision guided munitions, operating with low-cost satellite navigation updates against defended targets, was presented. The method provides guidance commands up to the time of apogee and is unaffected by jamming of external navigation updates subsequent to this time. A guidance problem with an unspecified final time was successfully transformed to a standard form amenable to the method of dynamic programming. The feedback gains computed using the guidance algorithm were shown to be robust for a range of small angle launch offsets in the absence of modeling errors. The performance of the guidance algorithm was degraded by modeling errors, but remained adequate for the level and type of errors considered. The method requires only a single pass to compute gains and could be done quickly to meet the needs of a battle environment.

Appendix: State Transition Matrices

Given the existence and uniqueness of the solution to the initial value problem given by Eq. (3), adjacent states in time can be related as

$$x_{k+1} = g_k(x_k, u_k) \tag{A1}$$

The state transition matrices introduced in Eq. (24) are defined as

$$A_k = \frac{\partial g_k}{\partial x_k} \tag{A2}$$

$$B_k = \frac{\partial g_k}{\partial u_k} \tag{A3}$$

where the partial derivatives in Eqs. (A2) and (A3) are evaluated at the nominal trajectory. As was already mentioned, a fixed-step, fourth-order, Runge–Kutta method is used for numerical integration of Eq. (3). The formula used by this method is

$$g_k = x_k + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} \tag{A4}$$

where

$$k_1 = hf(x_k, u_k, t_k) \tag{A5}$$

$$k_2 = hf(x_k + k_1/2, u_k, t_k + h/2) \tag{A6}$$

$$k_3 = hf(x_k + k_2/2, u_k, t_k + h/2) \tag{A7}$$

$$k_4 = hf(x_k + k_3, u_k, t_k + h) \tag{A8}$$

Using the chain rule for differentiation, one obtains

$$A_k = I + \frac{1}{6}\left(\frac{\partial k_1}{\partial x_k} + 2\frac{\partial k_2}{\partial x_2} + 2\frac{\partial k_3}{\partial x_k} + \frac{\partial k_4}{\partial x_k}\right) \tag{A9}$$

$$B_k = \frac{1}{6}\left(\frac{\partial k_1}{\partial u_k} + 2\frac{\partial k_2}{\partial u_k} + 2\frac{\partial k_3}{\partial u_k} + \frac{\partial k_4}{\partial u_k}\right) \tag{A10}$$

where I is the identity matrix and

$$\frac{\partial k_1}{\partial x_k} = h\frac{\partial f}{\partial x_k} \qquad \frac{\partial k_1}{\partial u_k} = h\frac{\partial f}{\partial u_k} \tag{A11}$$

$$\frac{\partial k_2}{\partial x_k} = h\frac{\partial f}{\partial x_k}\left(I + \frac{1}{2}\frac{\partial k_1}{\partial x_k}\right) \qquad \frac{\partial k_2}{\partial u_k} = h\left(\frac{\partial f}{\partial u_k} + \frac{1}{2}\frac{\partial f}{\partial x_k}\frac{\partial k_1}{\partial u_k}\right) \tag{A12}$$

$$\frac{\partial k_3}{\partial x_k} = h\frac{\partial f}{\partial x_k}\left(I + \frac{1}{2}\frac{\partial k_2}{\partial x_k}\right) \qquad \frac{\partial k_3}{\partial u_k} = h\left(\frac{\partial f}{\partial u_k} + \frac{1}{2}\frac{\partial f}{\partial x_k}\frac{\partial k_2}{\partial u_k}\right) \tag{A13}$$

$$\frac{\partial k_4}{\partial x_k} = h\frac{\partial f}{\partial x_k}\left(I + \frac{\partial k_3}{\partial x_k}\right) \qquad \frac{\partial k_4}{\partial u_k} = h\left(\frac{\partial f}{\partial u_k} + \frac{\partial f}{\partial x_k}\frac{\partial k_3}{\partial u_k}\right) \tag{A14}$$

The partial derivatives $\partial f/\partial x_k$ and $\partial f/\partial u_k$ in Eqs. (A11–A14) are evaluated at the same values of the arguments used to calculate $k_1, \ldots, k_4$, respectively.

Acknowledgment

This work, performed at Sandia National Laboratories, was supported by the U.S. Department of Energy under Contract DE-AC04-76DP00789.

References

[1] Synder, S., Tellman, L., Torrey, P., and Kohli, S., "INS/GPS Operational Concept Demonstration (OCD) High Gear Program," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 9, No. 9, 1994, pp. 38–44.

[2] Frost, G. P., and Schweitzer, B. P., "Operational Issues of GPS Aided Precision Missiles," *Proceedings of the 1993 National Technical Meeting of the Institute of Navigation* (San Francisco, CA), 1993, pp. 325–338.

[3] Kessler, K. M., Karkalik, F. G., Sun, J., and Brading, J. D., "Performance Analysis of Integrated GPS/INS Guidance for Air-to-Ground Weapons in a Jamming Environment," *Proceedings of the 4th International Technical Meeting of the Satellite Division of the Institute of Navigation* (Albuquerque, NM), 1991, pp. 139–148.

[4] Singh, M., Fraysee, J., and Kohli, S., "Silicon Inertial and GPS Guidance for Gun-Launched Munitions,"*Proceedings of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation* (Salt Lake City, UT), 1994, pp. 477–482.

[5] Loh, W. H. T., *Dynamics and Thermodynamics of Planetary Entry*, Prentice–Hall, Englewood Cliffs, NJ, 1963.

[6] Robinett, R. D., Rainwater, B., and Kerr, S. A., "Moving Mass Trim Control for Aerospace Vehicles," AIAA Missile Sciences Conf., Monterey, CA, 1994.

[7] Chavez, K. V., "MLRS Aerodynamic Coefficients," Sandia National Lab. Internal Memorandum, Albuquerque, NM, Feb. 1993.

[8] Dreyfus, S. E., and Averill M. L., *The Art and Theory of Dynamic Programming*, Academic, New York, 1977.

[9] Stengel, R. F., *Stochastic Optimal Control: Theory and Application*, Wiley, New York, 1986.

[10] Ralston, A., and Rabinowitz, P., *A First Course in Numerical Analysis*, 2nd ed., McGraw–Hill, New York, 1978.

[11] Friedland, B., *Control System Design: An Introduction to State-Space Methods*, McGraw–Hill, New York, 1986.