Journal of Guidance, Control, and Dynamics
Vol. 34, No. 3, May–June 2011

Suboptimal Midcourse Guidance of Interceptors for High-Speed Targets with Alignment Angle Constraint

Prasiddha Nath Dwivedi* and Abhijit Bhattacharya*
Defence Research and Development Organization, Hyderabad 500 058, India
and
Radhakant Padhi†
Indian Institute of Science, Bangalore, Bangalore, 560 012, India

DOI: 10.2514/1.50821

Using the recently developed computationally efficient model predictive static programming and a closely related model predictive spread control concept, two nonlinear suboptimal midcourse guidance laws are presented in this paper for interceptors engaging against incoming high-speed ballistic missiles. The guidance laws are primarily based on nonlinear optimal control theory, and hence imbed effective trajectory optimization concepts into the guidance laws. Apart from being energy efficient by minimizing the control usage throughout the trajectory (minimum control usage leads to minimum turning, and hence leads to minimum induced drag), both of these laws enforce desired alignment constraints in both elevation and azimuth in a hard-constraint sense. This good alignment during midcourse is expected to enhance the effectiveness of the terminal guidance substantially. Both point mass as well as six-degree-of-freedom simulation results (with a realistic inner-loop autopilot based on dynamic inversion) are presented in this paper, which clearly shows the effectiveness of the proposed guidance laws. It has also been observed that, even with different perturbations of missile parameters, the performance of guidance is satisfactory. A comparison study, with the vector explicit guidance scheme proposed earlier in the literature, also shows that the newly proposed model-predictive-static-programming-based and model-predictive-spread-control-based guidance schemes lead to lesser lateral acceleration demand and lesser velocity loss during engagement.

Nomenclature

a_z, a_y	=	sense latex in pitch and yaw plan
a_{z_c}, a_{y_c}	=	demanded latex in pitch and yaw plan
C_D, C_L, C_Y	=	drag, lift, and side force coefficient in velocity frame
C_l, C_m, C_n	=	rolling moment, pitch moment, and yaw moment coefficient
$C_{l_p}, C_{m_q}, C_{n_r}$	=	damping coefficient
$C_{N_\alpha}, C_{N_{\delta_q}}, C_{m_\alpha}, C_{m_{\delta_q}}$	=	aerodynamic coefficient for pitch plan
$C_{N_\beta}, C_{y_\alpha}, C_{m_\beta}, C_{n_\alpha}$	=	cross-plan aerodynamic coefficient
C_x, C_N, C_y	=	drag, lift, and side force coefficient in body-axis frame
$C_{y_\beta}, C_{y_{\delta_r}}, C_{n_\beta}, C_{n_{\delta_r}}$	=	aerodynamic coefficient for yaw plan
I_x, I_y, I_z	=	moment of inertia
M, V	=	Mach number and total velocity
m, S, d	=	mass, reference area, and diameter of missile
p, q, r	=	roll rate, pitch rate, and yaw rate
p_c, q_c, r_c	=	demanded roll rate, pitch rate, and yaw rate
$T_\alpha, T_{\mathrm{ap}}$	=	turning rate time constant and autopilot time constant
X, Y, Z	=	target/missile position in launcher fixed frame
α, β, ζ	=	angle of attack, sideslip angle, and roll angle
$\alpha_c, \beta_c, \phi_c$	=	demanded angle of attack, sideslip angle, and roll angle
γ, ϕ	=	flight-path angle and heading angle of interceptor in point mass model
γ_i, μ_i	=	flight-path angle and heading angle of interceptor used in six-degree-of-freedom model
γ_t, ϕ_t	=	flight-path angle and heading angle of target
$\delta_p, \delta_q, \delta_r$	=	aileron elevator and rudder deflection
η_γ, η_ϕ	=	acceleration in g in pitch and yaw planes
ξ	=	desired damping for roll, pitch, and yaw rates
ρ, v_s	=	density and speed of sound
ϕ_i, θ_i, ψ_i	=	Euler angle of interceptor used in six-degree-of-freedom model
$\omega_\alpha, \omega_\beta, \omega_\zeta$	=	desired bandwidth and damping for slow dynamic
$\omega_p, \omega_q, \omega_r$	=	desired bandwidth for roll, pitch, and yaw rates

I. Introduction

The trajectory of an interceptor missile mainly consists of three phases: namely, 1) boost phase, 2) midcourse phase, and 3) terminal phase. All of these phases play crucial roles in making a mission successful by intercepting the target successfully. The midcourse guidance phase is typically the longest one, and one of its primary objectives is to guide the missile properly that leads to the appropriate initial condition for the terminal phase so that the terminal phase can subsequently be very effective in taking the interceptor to close proximity of the target. To ensure this favorable condition for the terminal phase (which includes a low seeker look angle, a low requirement of line of sight (LOS) rate, etc.), alignment constraints in both elevation and azimuth (i.e., flight-path and heading angles, respectively) play a significant role, which must be ensured at the end of the midcourse phase. Moreover, being the longest phase (with a relatively longer time to go t_{go}), another restriction on the midcourse guidance is to minimize maneuvers as much as possible. This is mainly to minimize the induced drag, which in turn prevents a significant drop in the missile velocity before initiation of the terminal phase. In an intuitive sense, a minimum drag loss leads to range enhancement as well.

Numerous midcourse missile guidance schemes have been proposed in the literature, which include 1) general energy management (GEM) guidance [1], 2) singular perturbation (SP) guidance

Received 18 May 2010; revision received 18 September 2010; accepted for publication 29 November 2010. Copyright © 2010 by Radhakant Padhi. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/11 and $10.00 in correspondence with the CCC.

*Scientist.

†Associate Professor; padhi@aero.iisc.ernet.in.

[2], 3) optimal guidance [3,4], 4) proportional navigation (PN) with a shaping term [5], etc. GEM guidance is typically used for long-range applications (of the order of more than 100 km range), which is not relevant to the case addressed in this paper. SP guidance is a well-known energy-efficient technique and suitable for midcourse guidance in general, but it does not enforce the alignment constraint. In the PN guidance with shaping term, one can achieve the angle constraint, but that needs to be done from a tedious offline heuristic tuning process, which depends on the predicted target trajectory as well. Optimal control theory is perhaps the right tool to address a number of challenging guidance design problems in general (including midcourse guidance).

In fact, the idea of using optimal control techniques for guidance of flight vehicles is not new (see, for example, [6,7] for more details). However, such a formulation leads to two-point boundary value problems (TPBVPs) [8], which in turn lead to large computational requirements that are infeasible to implement in real time. Moreover, the control solution becomes open loop, which is highly undesirable. The solution approach of the TPBVP requires numerical techniques like the shooting method [8], the gradient method [9], etc. These are iterative processes that require a lot of memory storage and computation time as well. Because of their computational difficulties, they have largely been restricted to either linear (or linearized) guidance problems or offline trajectory optimization problems. The infinite-time linear quadratic regulator (LQR) theory, however, is an exception that facilitates a feedback solution of the control variable after solving the associated algebraic Riccati equation [3] offline. Because of this, many conventional optimal guidance laws are based on LQR theory. However, since they are based on the linearized engagement dynamics, they are not powerful enough to address the stringent requirements for true engagements (which are invariably nonlinear).

There have recently been attempts to extend the LQR theory to nonlinear problems. The most notable idea in this regard is perhaps the state-dependent Riccati equation (SDRE) approach [10]; hence, it has recently been applied for guidance and control of missiles as well [11]. However, the SDRE solution is usually suboptimal. Moreover, the approach assumes that the Riccati equation can be solved sufficiently fast for the control update to be carried out online, which may not always be feasible. A relatively recent alternate approach, known as the θ-D method [12,13], partially overcomes this problem, since it requires that a small set of linear Lyapunov equations needs to be solved instead of the nonlinear Riccati equation. However, it has significantly more design parameters, which need to be tuned very carefully. Although both SDRE and θ-D methods attempt to provide a rapid solution, their utility is limited only to infinite-horizon quadratic regulator problems for control affine systems.

Even though a considerably fair amount of attention has been given in literature to develop computationally efficient techniques for infinite-time regulator problems and their extensions, the same cannot be said for finite-time problems in general (which are more relevant for missile guidance problems). A notable contribution in this regard is perhaps the pseudospectral method [14–16], which has been proposed recently and has been used in a number of guidance design applications as well. Similarly, combining the philosophies of nonlinear model predictive control (MPC) theory [17–21] and approximate dynamic programming (ADP) [22], a new nonlinear optimal control design method called model predictive static programming (MPSP) has also been proposed recently [23,24]. This technique solves a class of nonlinear optimal control problems that require that 1) the output vector satisfy a set of hard constraints at the final time t_f and 2) the control effort should be minimized in the entire interval of $[t_0, t_f]$. The MPSP technique has been applied to a variety of aerospace guidance problems recently, including missile guidance problems [24]. Innovation of the MPSP technique lies with the fact that it successfully converts a dynamic programming problem (which an optimal control problem leads to if formulated from the Hamilton–Jacobi–Bellman approach [4]) to a static programming problem. It requires that a static costate vector (of the same dimension as the state vector) be solved to get the control history update for the entire interval. Moreover, this costate vector has a symbolic solution; hence, it leads to a closed-form solution of the optimal control history update. Moreover, the sensitivity matrices (which are required to be computed in this technique) can be computed in a recursive manner, which further saves the computational time. This avoids the numerical complexities of optimal control theory, making it computationally very efficient, and hence suitable for online implementation. Among various applications, the MPSP technique holds good promise for optimal missile guidance. Note that the MPSP technique brings in the philosophy of trajectory optimization into the framework of guidance design, which in turn results in very effective missile guidance. For more details about the technique, one can refer to some of the available literature [23–25].

The MPSP technique has been further extended recently by considering a parameterized version of the control variable as polynomial expressions of either time t or, equivalently, time to go t_{go} [26,27]. This has been named as the model predictive spread control (MPSC), since it essentially retains most of the features of the MPSP design and yet spreads the control requirement over the entire t_{go} in a very deterministic way that is decided a priori. The advantage of this approach is that the control action is guaranteed to be smooth (by enforcement). Moreover, the computational time is further reduced, since instead of solving a free control variable at each grid point in time, it essentially requires the solution of only a very small dimensional set of coefficients of the polynomial expressions. Note that, even though only linear and quadratic polynomials have been used in this paper, one can essentially use polynomials of any order (or even polynomials other than power series) so long as the number of free variables (i.e., coefficients) are more than or equal to the number of output constraints in the problem formulation.

Details of both the MPSP as well as MPSC techniques are given in Sec. II of this paper, which are subsequently applied to solve a challenging midcourse guidance problem for interceptors that are intended to intercept high-speed incoming ballistic missiles. In this design, the desired alignment constraints in both elevation and azimuth (i.e., flight-path angle and heading angle) are enforced in a hard-constraint sense, and hence lead to better accuracy of alignment of the interceptor velocity vector with respect to the target. This is rather a necessity, because the t_{go} in the terminal phase of such engagements are usually small and, unless there is good alignment to begin with, the terminal phase gets strained, which may lead to compromise in the final miss distance. Moreover, the seeker look angle (which is a critical factor for seeker lock on) is usually small; hence, a good alignment is necessary for the seeker to capture the target to initiate the terminal phase.

Note that the MPSP and MPSC techniques are primarily formulated with time t as the independent variable in the framework of finite-time problems. However, the guidance law developed here uses the ground range as the free variable, essentially converting it to a finite-range problem. This makes the guidance formulation more generic and eliminates the open-loop nature of the solution. Also note that, along with the alignment angle constraints, position of the interceptor at the end of the midcourse phase is also enforced in a hard-constraint sense as part of the problem formulation. Apart from the angle and position constraints at the end of the midcourse, throughout the midcourse phase, the control usage (i.e., lateral acceleration demand) is also minimized in this formulation by minimizing a quadratic cost function. This minimum lateral acceleration formulation leads to minimum turning, and hence leads to minimum induced drag, which in turn minimizes the velocity loss of the vehicle. This is another requirement of a good midcourse guidance logic, as the midcourse is the longest phase of the missile's flight path, and it operates in the relatively dense atmosphere.

In addition to the simulation studies with a point-mass model (which is usually used for guidance design), the effectiveness of the proposed guidance law has also been verified from the six-degree-of-freedom (six-DOF) simulation studies. However, this is a not a trivial task and includes the task of realizing the lateral acceleration demands by generating the necessary control surface deflections, depending on the system dynamics of the vehicle. Toward this objective, a realistic inner-loop autopilot based on dynamic inversion [28] has also been carried out in this paper, and the details of this inner-loop control design are presented in Sec. V.B. Both point mass as well as six-DOF simulation results are presented in this paper, which clearly shows the effectiveness of the proposed guidance laws. Note that, even though both the approaches lead to very similar performance results (which are better than the conventional approaches because of alignment constraints), it has been observed that the MPSC guidance is marginally better over the MPSP guidance, since it enforces lateral acceleration smoothness in addition to being computationally more efficient. Moreover, comparison studies with the existing VEG scheme [29,30] (which proposes a closed-form solution for a linearized version of the problem) show that the newly proposed MPSP- and MPSC-based guidance schemes lead to lesser lateral acceleration demand and lesser velocity loss during engagement.

The rest of the paper is organized as follows. In Sec. II, the MPSP and MPSC approaches are explained in detail. In Sec. III, the mathematical model of the vehicle is presented. The guidance design with MPSP and MPSC is presented in Sec. IV. Numerical simulation results are presented in Sec. V. Finally, some conclusions are drawn in Sec. VI.

II. Model Predictive Static Programming/Model Predictive Spread Control Design: Mathematical Formulation

A. Model Predictive Static Programming Design

In this section, we combined the philosophies of MPC [17–20] and ADP [22] to propose an innovative technique for a class of finite-horizon optimal control problems. Here, we present the mathematical details of the new MPSP design, taking into account the state-space model of dynamic system. In this design, we consider general nonlinear systems in discrete form, the state and output dynamics of which are given by

$$X_{k+1} = F_k(X_k, U_k) \tag{1}$$

$$Y_k = h(X_k) \tag{2}$$

where $X \in \Re^n$, $U \in \Re^m$, $Y \in \Re^p$, and $k = 1, 2, \dots, N$ are the time steps. The primary objective is to come up with a suitable control history U_k, $k = 1, 2, \dots, N-1$, so that the output at the final time step Y_N goes to a desired value Y_N^*: that is, $Y_N \to Y_N^*$. In addition, we aim to achieve this task with minimum control effort (which will be clearer later in this section).

For the technique presented here, one needs to start from a guess history of the control solution. With the application of such a guess history, obviously, the objective is not expected to be met; hence, there is a need to improve this solution. In this section, we present a way to compute an error history of the control variable, which needs to be subtracted from the previous history to get an improved control history. This iteration continues until the objective is met: i.e., until $Y_N \to Y_N^*$. Note that the technique presented here comes up with a control update history in closed form; hence, the computational requirement is substantially lesser and the algorithm can be used online. Next, we present the mathematical details of the MPSP design.

Expanding Y_N about Y_N^* using the Taylor series expansion,

$$Y_N = Y_N^* + \left[\frac{\partial Y_N}{\partial X_N}\right] \mathrm{d}X_N + \mathrm{HOT} \tag{3}$$

where HOT denotes the higher-order terms. From Eq. (3), we can write the error in the output as

$$Y_N - Y_N^* = \left[\frac{\partial Y_N}{\partial X_N}\right] \mathrm{d}X_N + \mathrm{HOT} \tag{4}$$

Using small error approximation, we write

$$\triangle Y_N \cong \mathrm{d}Y_N = \left[\frac{\partial Y_N}{\partial X_N}\right] \mathrm{d}X_N \tag{5}$$

However, from Eq. (1), we can write the error in state at time step $(k+1)$ as

$$\mathrm{d}X_{k+1} = \left[\frac{\partial F_k}{\partial X_k}\right] \mathrm{d}X_k + \left[\frac{\partial F_k}{\partial U_k}\right] \mathrm{d}U_k \tag{6}$$

where $\mathrm{d}X_k$ and $\mathrm{d}U_k$ are the error of state and control at time step k, respectively.

Expanding $\mathrm{d}X_N$, as in Eq. (6) (for $k = N-1$), and substituting it in Eq. (5), we get

$$\mathrm{d}Y_N = \left[\frac{\partial Y_N}{\partial X_N}\right]\left(\left[\frac{\partial F_{N-1}}{\partial X_{N-1}}\right] \mathrm{d}X_{N-1} + \left[\frac{\partial F_{N-1}}{\partial U_{N-1}}\right] \mathrm{d}U_{N-1}\right) \tag{7}$$

Similarly, the error in state at time step $(N-1)$, $\mathrm{d}X_{N-1}$, can be expanded in terms of the errors in state and control at time step $(N-2)$, and Eq. (7) can be rewritten as

$$\mathrm{d}Y_N = \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F_{N-1}}{\partial X_{N-1}}\right]\left(\left[\frac{\partial F_{N-2}}{\partial X_{N-2}}\right] \mathrm{d}X_{N-2} + \left[\frac{\partial F_{N-2}}{\partial U_{N-2}}\right] \mathrm{d}U_{N-2}\right) + \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F_{N-1}}{\partial U_{N-1}}\right] \mathrm{d}U_{N-1}$$

Next, $\mathrm{d}X_{N-2}$ can be expanded in terms of $\mathrm{d}X_{N-3}$ and $\mathrm{d}U_{N-3}$, and so on. Continuing the process $k = 1$, we can write

$$\mathrm{d}Y_N = A\,\mathrm{d}X_1 + B_1\,\mathrm{d}U_1 + B_2\,\mathrm{d}U_2 + \cdots + B_{N-1}\,\mathrm{d}U_{N-1} \tag{8}$$

where the matrix A is defined as

$$A = \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F_{N-1}}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F_1}{\partial X_1}\right]$$

and the sensitivity matrices are defined as

$$B_k = \left[\frac{\partial Y_N}{\partial X_N}\right]\left[\frac{\partial F_{N-1}}{\partial X_{N-1}}\right] \cdots \left[\frac{\partial F_{k+1}}{\partial X_{k+1}}\right]\left[\frac{\partial F_k}{\partial U_k}\right], \qquad k = 1, \dots, N-1$$

Since the initial condition is specified, there is no error in the first term, which means $\mathrm{d}X_1 = 0$. With this, Eq. (8) reduces to

$$\mathrm{d}Y_N = B_1\,\mathrm{d}U_1 + B_2\,\mathrm{d}U_2 + \cdots + B_{N-1}\,\mathrm{d}U_{N-1} = \sum_{k=1}^{N-1} B_k\,\mathrm{d}U_k \tag{9}$$

Note that, while deriving Eq. (9), we have assumed that the control variable at each time step is independent of the previous values of states and/or control. Intuitive justification of this assumption comes from the fact it is a decision variable; hence, an independent decision can be taken at any point of time.

At this point, we would like to point out that, if one evaluates each of the B_k, $k = 1, \dots, (N-1)$ as in Eq. (9), it will be a computationally intensive task (especially when N is high). However, fortunately, it is possible to compute them recursively. For doing this, first we define B_{N-1}^0 as follows:

$$B_{N-1}^0 = \left[\frac{\partial Y_N}{\partial X_N}\right] \tag{10}$$

Next, we compute B_k^0, $k = (N-2), (N-3), \dots, 1$, as

$$B_k^0 = B_{k+1}^0\left[\frac{\partial F_{k+1}}{\partial X_{k+1}}\right] \tag{11}$$

Finally, B_k, $k = (N-1), (N-2), \dots, 1$, can be computed as

$$B_k = B_k^0\left[\frac{\partial F_k}{\partial U_k}\right] \tag{12}$$

Equations (10–12) provides a recursive way of computing B_k^0, $k = (N-1), (N-2), \ldots, 1$, which leads to the enormous saving of computational time.

In Eq. (9), we have $(N-1)m$ unknowns and p equations. Usually, $p < (N-1)m$; hence, it is an underconstrained system of equations and there is a scope for meeting additional objectives. We take advantage of this opportunity and aim to minimize the following objective (cost) function:

$$J = \frac{1}{2}\sum_{k=1}^{N-1}(U_k^0 - \mathrm{d}U_k)^T R_k (U_k^0 - \mathrm{d}U_k) \tag{13}$$

where U_k^0, $k = 1, \ldots, (N-1)$, is the previous control history solution, and $\mathrm{d}U_k$ is the corresponding error in the control history. The cost function in Eq. (13) needs to be minimized subjected to the constraint in Eq. (9), where $R_k > 0$ (a positive definite matrix) is the weighting matrix, which needs to be chosen judiciously by the control designer. The selection of such a cost function is motivated from the fact that we are interested in finding an l_2-norm minimizing control history [31], since $(U_k^0 - \mathrm{d}U_k)$ is the updated control value at k [see Eq. (22)].

Equations (9) and (13) formulate an appropriate constrained static optimization problem. Hence, using optimization theory [4,32], the augmented cost function is given by

$$\bar{J} = \frac{1}{2}\sum_{k=1}^{N-1}(U_k^0 - \mathrm{d}U_k)^T R_k (U_k^0 - \mathrm{d}U_k) + \lambda^T\left(\mathrm{d}Y_N - \sum_{k=1}^{N-1} B_k\,\mathrm{d}U_k\right) \tag{14}$$

Then, the necessary conditions of optimality are given by

$$\frac{\partial \bar{J}_k}{\partial\,\mathrm{d}U_k} = -R_k(U_k^0 - \mathrm{d}U_k) - B_k^T\lambda = 0 \tag{15}$$

$$\frac{\partial \bar{J}_k}{\partial \lambda} = \mathrm{d}Y_N - \sum_{k=1}^{N-1} B_k\,\mathrm{d}U_k = 0 \tag{16}$$

Solving for $\mathrm{d}U_k$ from Eq. (15), we get

$$\mathrm{d}\,U_k = R_k^{-1}B_k^T\lambda + U_k^0 \tag{17}$$

Substituting for $\mathrm{d}U_k$ from Eq. (17) into Eq. (16) leads to

$$-A_\lambda\lambda + b_\lambda = \mathrm{d}Y_N \tag{18}$$

where

$$A_\lambda = \left[-\sum_{k=1}^{N-1} B_k R_k^{-1} B_k^T\right], \qquad b_\lambda = \left[\sum_{k=1}^{N-1} B_k U_k^0\right] \tag{19}$$

Note that A_λ is a $p \times p$ matrix and b_λ is a $p \times 1$ vector. Assuming A_λ to be nonsingular, the solution for λ from Eq. (18) is given by

$$\lambda = -A_\lambda^{-1}(\mathrm{d}Y_N - b_\lambda) \tag{20}$$

Using Eq. (20) in Eq. (17) leads to

$$\mathrm{d}\,U_k = -R_k^{-1}B_k^T A_\lambda^{-1}(\mathrm{d}Y_N - b_\lambda) + U_k^0 \tag{21}$$

Hence, the updated control at time step $k = 1, 2, \ldots, (N-1)$ is given by

$$U_k = U_k^0 - \mathrm{d}U_k = R_k^{-1}B_k^T A_\lambda^{-1}(\mathrm{d}Y_N - b_\lambda) \tag{22}$$

This is how the control history is updated in the MPSP algorithm. At this point, we would like to point out that we have used small error approximation in deriving the closed-form control update. This approximation may not hold good in general. Hence, the process needs to be repeated in an iterative manner before one arrives at the converged (optimal) solution, which is defined as the solution when $Y_N \to Y_N^*$.

Note that the MPSP technique is, computationally, a very efficient algorithm because of several reasons:

1) Unlike the classical optimal control solution techniques, only a static costate variable is necessary to update the entire control history.

2) The static costate vector is computed symbolically.

3) The sensitivity matrices are computed recursively.

Because of these reasons, the algorithm usually converges quite rapidly (within a couple of iterations). Moreover, for finite-time computation, the idea of iteration unfolding is very useful. In this philosophy, only a finite number of iterations are carried out at any time step. This does not alter the nature of the results much. In fact, in all the results included in this paper, we have implemented this idea with only one iteration at any point of time. Quite naturally, there is no problem in implementing the algorithm without the iteration unfolding as well. However, those results are not included here to contain the length of this paper (the results are very similar in any case).

We also mention that the relative magnitude of the control input at various time steps can be adjusted by properly adjusting the weight matrixes R_k, $k = 1, \ldots, (N-1)$, associated with the cost function. This gives additional flexibility in lateral acceleration shaping, which is an additional advantage. Note that details on the MPSP technique can be found in some of the earlier published literature as well [23,24].

B. Model Predictive Spread Control Design

In this section, we present the mathematical details of the MPSC design, which is closely related to the MPSP technique. The main difference here is a priori parameterizations of control, which leads to a few further advantages over the MPSP technique. Note that, even though the following discussion is carried out with only two parameterizations (namely, the straight line and quadratic parameterizations), the generic theory can be extended to other parameterizations in an analogous manner.

1. *Model Predictive Spread Control Design: Straight Line Formulation*

In this formulation, the control is considered to be a function of time t:

$$U_k = at_k + b \tag{23}$$

$$U_k = U_k^0 - \mathrm{d}U_k \tag{24}$$

where U_k^0 is the initial guess history of control obtained from PN law. Therefore, the error in control can be given as

$$\begin{aligned}\mathrm{d}U_k = U_k^0 - U_k &= (a_0 t_k + b_0) - (at_k + b)\\ &= (a_0 - a)t_k + (b_0 - b)\end{aligned} \tag{25}$$

Substituting $\mathrm{d}U_k$ for $k = 1, \ldots, N-1$ in

$$\mathrm{d}\,Y_N = B_1\,\mathrm{d}U_1 + B_2\,\mathrm{d}U_2 + \ldots \cdots + B_{N-1}\,\mathrm{d}U_{N-1} \tag{26}$$

$$= \sum_{k=1}^{N-1} B_k\,\mathrm{d}U_k \tag{27}$$

we get

$$\begin{aligned}&B_\lambda - \left(\sum_{k=1}^{N-1} B_k t_k\right)a - \left(\sum_{k=1}^{N-1} B_k\right)b = \mathrm{d}Y_N\\ &C_y a + D_y b = B_\lambda - \mathrm{d}Y_N\end{aligned} \tag{28}$$

where

$$B_\lambda = (B_1 U_1^0 + B_2 U_2^0 + \cdots B_{N-1} U_{N-1}^0) \tag{29}$$

$$C_y = \left(\sum_{k=1}^{N-1} B_k t_k\right) \tag{30}$$

$$D_y = \left(\sum_{k=1}^{N-1} B_k\right) \tag{31}$$

If the number of unknown variables is the same as the number of equations, then Eq. (28) can be write as

$$[C_y \quad D_y]\begin{bmatrix} a \\ b \end{bmatrix} = B_\lambda - \mathrm{d}Y_N \tag{32}$$

and the solution can be obtained as

$$\begin{bmatrix} a \\ b \end{bmatrix} = [C_y \quad D_y]^{-1}[B_\lambda - \mathrm{d}Y_N] \tag{33}$$

If the number of unknowns is greater than the number of equations, however, the optimal solution can be obtained by putting additional objectives in the formulation. One possible option is to minimize the following cost function, subject to the constraint in Eq. (28):

$$J = \tfrac{1}{2}(r_1 a^2 + r_2 b^2) \tag{34}$$

where $r_1, r_2 > 0$ are the tuning parameters. In Eq. (34), a represents the slope and b represents the y intercept. Hence, by reducing J, the aim is to obtain a control history with minimum slope and intercept, which in turn assures minimum lateral acceleration demand.

The augmented cost function in this case is given by

$$\bar{J} = \tfrac{1}{2}(r_1 a^2 + r_2 b^2) + \lambda^T(\mathrm{d}Y_N - B_\lambda + C_y a + D_y b) \tag{35}$$

The necessary conditions of optimality are

$$\frac{\partial \bar{J}}{\partial a} = r_1 a + C_y^T \lambda = 0 \tag{36}$$

$$\frac{\partial \bar{J}}{\partial b} = r_2 b + D_y^T \lambda = 0 \tag{37}$$

Substituting for C_y and D_y, and solving for a and b, we get

$$a = -\left(\frac{1}{r_1}\right)\left[\left(\sum_{k=1}^{N-1} B_k t\right)\lambda\right] \tag{38}$$

$$b = -\left(\frac{1}{r_2}\right)\left[\left(\sum_{k=1}^{N-1} B_k\right)\lambda\right] \tag{39}$$

Substituting for a and b in Eq. (28), we get

$$\mathrm{d}Y_N = B_1 U_1^0 + \cdots + B_{N-1} U_{N-1}^0 + \left(\sum_{k=1}^{N-1} B_k t\right)\left(\sum_{k=1}^{N-1} B_k^T t\right)\frac{\lambda}{r_1} + \left(\sum_{k=1}^{N-1} B_k\right)\left(\sum_{k=1}^{N-1} B_k^T\right)\frac{\lambda}{r_1} \tag{40}$$

which leads to

$$A_\lambda \lambda + B_\lambda = \mathrm{d}Y_N \tag{41}$$

where

$$A_\lambda = \left(\sum_{k=1}^{N-1} B_k t\right)\left(\sum_{k=1}^{N-1} B_k^T t\right) + \left(\sum_{k=1}^{N-1} B_k\right)\left(\sum_{k=1}^{N-1} B_k^T\right) \tag{42}$$

$$B_\lambda = [B_1 U_1^0 + \cdots + B_{N-1} U_{N-1}^0] \tag{43}$$

Assuming A_λ to be nonsingular, the solution for λ from Eq. (41) is given by

$$\lambda = A_\lambda^{-1}[\mathrm{d}Y_N - B_\lambda] \tag{44}$$

Therefore, the error in control can be computed as

$$\mathrm{d}\, U_k = U_k^0 + \left(\sum_{k=1}^{N-1} B_k^T t\right)\lambda t + \left(\sum_{k=1}^{N-1} B_k^T\right)\lambda \tag{45}$$

Hence, the updated control at time step $k = 1, 2, \ldots, (N-1)$ is given by

$$U_k = U_k^0 - \mathrm{d}U_k \tag{46}$$

$$= -\left[\left(\sum_{k=1}^{N-1} B_k^T t\right)\lambda t + \left(\sum_{k=1}^{N-1} B_k^T\right)\lambda\right] \tag{47}$$

From Eq. (46), a closed-form solution of the control history can be obtained. By varying the weights r_1 and r_2 present in the cost function, the slope and intercept values can be changed. It is clear that the updated control history solution in Eq. (46) is a closed-form solution.

2. *Model Predictive Spread Control Design: Quadratic Formulation*

In this formulation, the control is considered to be a function of time t:

$$U_k = a t_k^2 + b t_k + c \tag{48}$$

$$U_k = U_k^0 - \mathrm{d}U_k \tag{49}$$

where U_k^0 is the initial guess history of control obtained from PN law. Therefore, the error in control can be given as

$$\begin{aligned} \mathrm{d}U_k = U_k^0 - U_k &= (a_0 t_k^2 + b_0 t_k + c_0) - (a t_k^2 + b t_k + c) \\ &= (a_0 - a)t_k^2 + (b_0 - b)t_k + (c_0 - c) \end{aligned}$$

Substituting $\mathrm{d}U_k$ for $k = 1, \ldots, N-1$ in

$$\mathrm{d}\, Y_N = B_1\, \mathrm{d}U_1 + B_2\, \mathrm{d}U_2 + \cdots + B_{N-1}\, \mathrm{d}U_{N-1} \tag{50}$$

$$= \sum_{k=1}^{N-1} B_k\, \mathrm{d}U_k \tag{51}$$

we get

$$\begin{aligned} &B_\lambda - \left(\sum_{k=1}^{N-1} B_k t_k^2\right)a - \left(\sum_{k=1}^{N-1} B_k t_k\right)b - \left(\sum_{k=1}^{N-1} B_k\right)c = \mathrm{d}Y_N \\ &C_y a + D_y b + E_y c = B_\lambda - \mathrm{d}Y_N \end{aligned} \tag{52}$$

where

$$B_\lambda = (B_1 U_1^0 + B_2 U_2^0 + \cdots B_{N-1} U_{N-1}^0) \tag{53}$$

$$C_y = \left(\sum_{k=1}^{N-1} B_k t_k^2\right) \tag{54}$$

$$D_y = \left(\sum_{k=1}^{N-1} B_k t_k\right) \tag{55}$$

$$E_y = \left(\sum_{k=1}^{N-1} B_k\right) \tag{56}$$

If the number of unknown variables is the same as the number of equations, then Eq. (28) can be written as

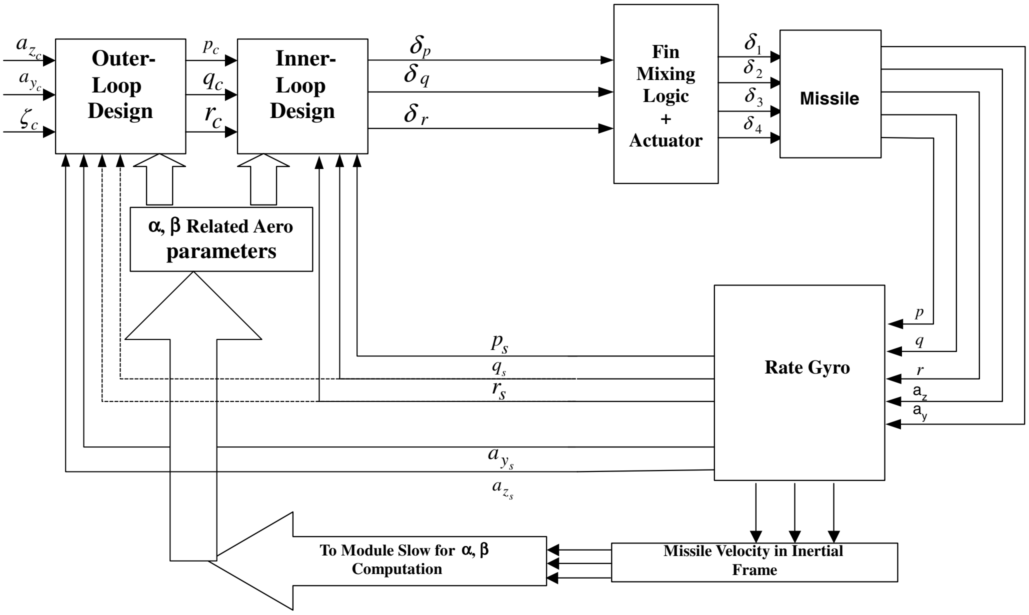

Fig. 1 Block diagram of controller.

$$[C_y \quad D_y \quad E_y]\begin{bmatrix} a \\ b \\ c \end{bmatrix} = B_\lambda - \mathrm{d}Y_N \tag{57}$$

and the solution can be obtained as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = [C_y \quad D_y \quad E_y]^{-1}[B_\lambda - \mathrm{d}Y_N] \tag{58}$$

However, if the number of unknowns is greater than the number of equations, the optimal solution can be obtained by minimizing the following objective (cost) function, subject to the constraint in Eq. (28):

$$J = \tfrac{1}{2}(r_1 a^2 + r_2 b^2 + r_3 b^2) \tag{59}$$

Further detail derivations of this formulation, however, are similar to the algebra discussed before, and hence omitted for brevity.

At this point, we would like to point out that, in this approach as well, the control history update process needs to be repeated in an iterative manner before one arrives at the converged (optimal) solution. Even though the number of iterations required is typically small, to assure a deterministic computational time, one can also use the philosophy of iteration unfolding, which has been used in this paper by updating the control history only once at any point of time. Note that the MPSC technique can also be found in [23,24].

III. Mathematical Models for Interceptor and Target

The primary objective of this paper is to present an effective midcourse guidance logic. Hence, in principle, one requires point-mass mathematical models for both the interceptor (a missile) as well as the target (which is also a missile). However, since our intension is to validate the results with the six-DOF simulation studies as well, we need to have a six-DOF model of the interceptor as well. These are discussed in this section.

A. Point-Mass Model of Interceptor

The following point-mass model is used for the guidance design:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{V} \\ \dot{\phi} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} V\cos\gamma\cos\phi \\ V\cos\gamma\sin\phi \\ V\sin\gamma \\ \frac{T-D}{m} - g\sin\gamma \\ \frac{g\eta_\phi}{V\cos\gamma} \\ \frac{g}{V}(\eta_\gamma - \cos\gamma) \end{bmatrix} \tag{60}$$

where x, y, and z are the missile positions in the launcher fixed inertial frame. V is the velocity of the missile. T is the thrust, and D is the aerodynamic drag of the missile.

B. Six-Degree-of-Freedom Model of Interceptor

The nonlinear six-DOF model used for the inner-loop control design, as well as the realistic validation, is described by the following system of equations [28]:

$$\begin{bmatrix} \dot{M} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \frac{1}{mv_s}[-0.5\rho V^2 S C_D - mg\sin\gamma_i] \\ q - \frac{1}{\cos\beta}\left[(p\cos\alpha + r\sin\alpha)\sin\beta + \frac{1}{mV_sM}(-0.5\rho V^2 S C_L - mg\cos\gamma_i\cos\mu_i)\right] \\ p\sin\alpha - r\cos\alpha + \frac{1}{mV_sM}(0.5\rho V^2 S C_Y + mg\cos\gamma_i\sin\mu_i) \end{bmatrix} \tag{61}$$

Table 1 Initial and desired final conditions of missile

	Position X, m	Position Y, m	Position Z, m	Velocity, m/s	ϕ, deg	γ, deg
Initial condition	0	0	6500	1400	45	45
Desired final condition for case 1	6600	6200	15000	——	20	70
Desired final condition for case 2	6600	6200	15000	——	50	10

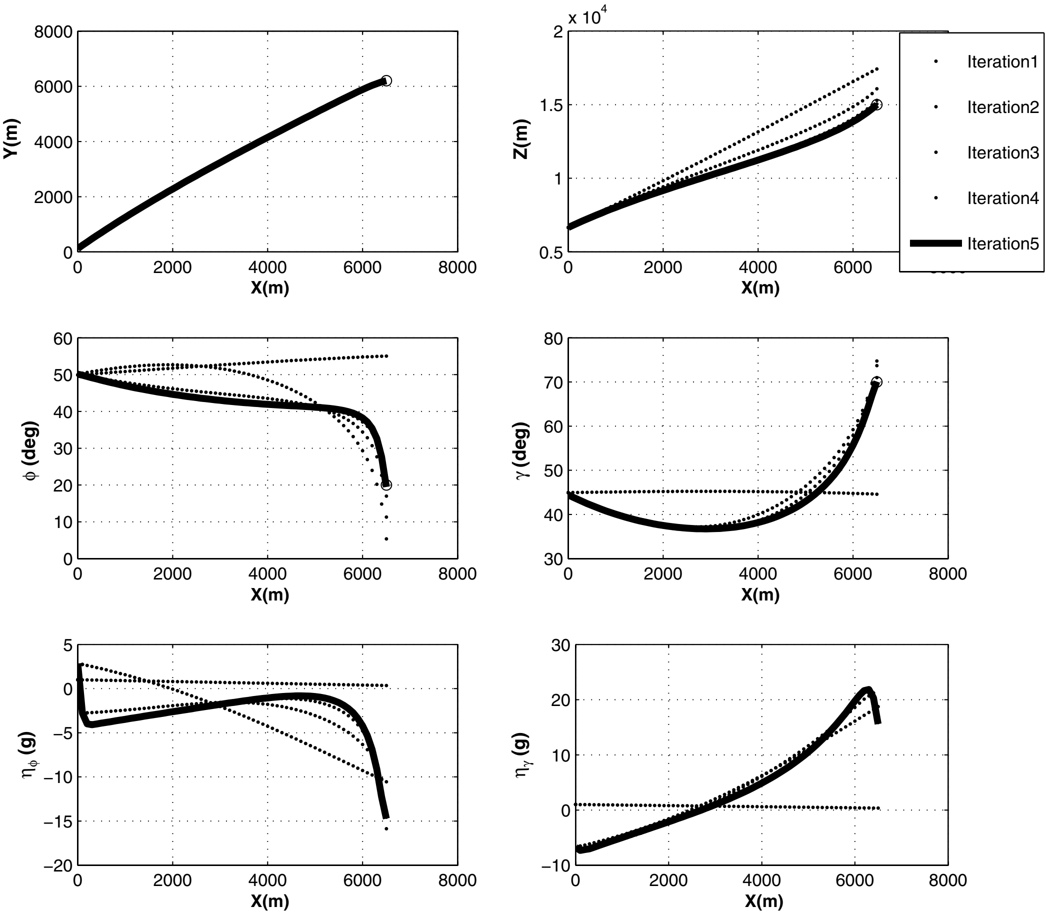

Fig. 2 Results of MPSP formulation (case 1).

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{I_x}[(I_y - I_z)rq + 0.5\rho V^2 S\,\mathrm{d}C_l] \\ \frac{1}{I_y}[(I_z - I_x)rp + 0.5\rho V^2 S\,\mathrm{d}C_m] \\ \frac{1}{I_z}[(I_x - I_y)pq + 0.5\rho V^2 S\,\mathrm{d}C_n] \end{bmatrix} \tag{62}$$

$$\begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} p + q\sin\phi_i \tan\theta_i + r\cos\phi_i \tan\theta_i \\ q\cos\phi_i - r\sin\phi_i \\ \frac{q\sin\phi_i + r\cos\phi_i}{\cos\theta_i} \\ p \end{bmatrix} \tag{63}$$

where

$$\begin{bmatrix} C_D \\ C_L \\ C_Y \\ C_N \\ C_y \\ C_l \\ C_m \\ C_n \end{bmatrix} = \begin{bmatrix} C_x\cos\alpha\cos\beta + C_N\sin\alpha + C_y\sin\beta \\ C_N\cos\alpha - C_x\sin\alpha \\ C_y\cos\beta - C_x\sin\beta \\ C_{N_\alpha}\alpha + C_{N_{\delta_q}}\delta_q \\ C_{y_\beta}\beta + C_{y_{\delta_r}}\delta_r \\ \frac{C_{l_p}\,\mathrm{d}p}{2V} + C_{l_{\delta_p}}\delta_p \\ C_{m_\alpha}\alpha + C_{m_q}q + C_{m_{\delta_q}}\delta_q \\ C_{n_\beta}\beta + C_{n_r}r + C_{n_{\delta_r}}\delta_r \end{bmatrix} \tag{64}$$

and, γ_i and μ_i are computed from the following expressions:

$$\begin{bmatrix} \sin\gamma_i \\ \sin\mu_i \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta\sin\theta_i - \sin\beta\sin\phi_i\cos\theta_i - \sin\alpha\cos\beta\cos\phi_i\cos\theta_i \\ \frac{1}{\cos\gamma_i}(\cos\alpha\sin\beta\sin\theta_i + \cos\beta\sin\phi_i\cos\theta_i - \sin\alpha\sin\beta\cos\phi_i\cos\theta_i) \end{bmatrix} \tag{65}$$

C. Target Model and Prediction of Interception Point

The following point-mass model is being used for target modeling:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{V} \\ \dot{\phi}_t \\ \dot{\gamma}_t \end{bmatrix} = \begin{bmatrix} V\cos\gamma_t\cos\phi_t \\ V\cos\gamma_t\sin\phi_t \\ V\sin\gamma_t \\ \frac{-D}{m} - g\sin\gamma_t \\ 0 \\ \frac{-g\cos\gamma_t}{V} \end{bmatrix} \tag{66}$$

where x, y, and z are the target positions in the launcher fixed frame. V is the velocity of the target.

For calculation of the predicted intercept point (PIP), the target model is extrapolated until the desired height of kill to extract other states of the target. Once the predicted heading angle and flight-path angle are known, the desired heading angle and flight-path angle to be achieved by the missile can be calculated, which is typically done for maximizing the warhead lethality.

Table 2 Convergence results of MPSP (case 1)

Iteration	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg
1	2614.4	2598.4	35	25
2	850.67	1526.5	−78	19
3	232.3	231.1	21.2	0.42
4	−58.5	−49.6	−6.23	−1.72
5	−0.54	0.91	0.42	−0.26

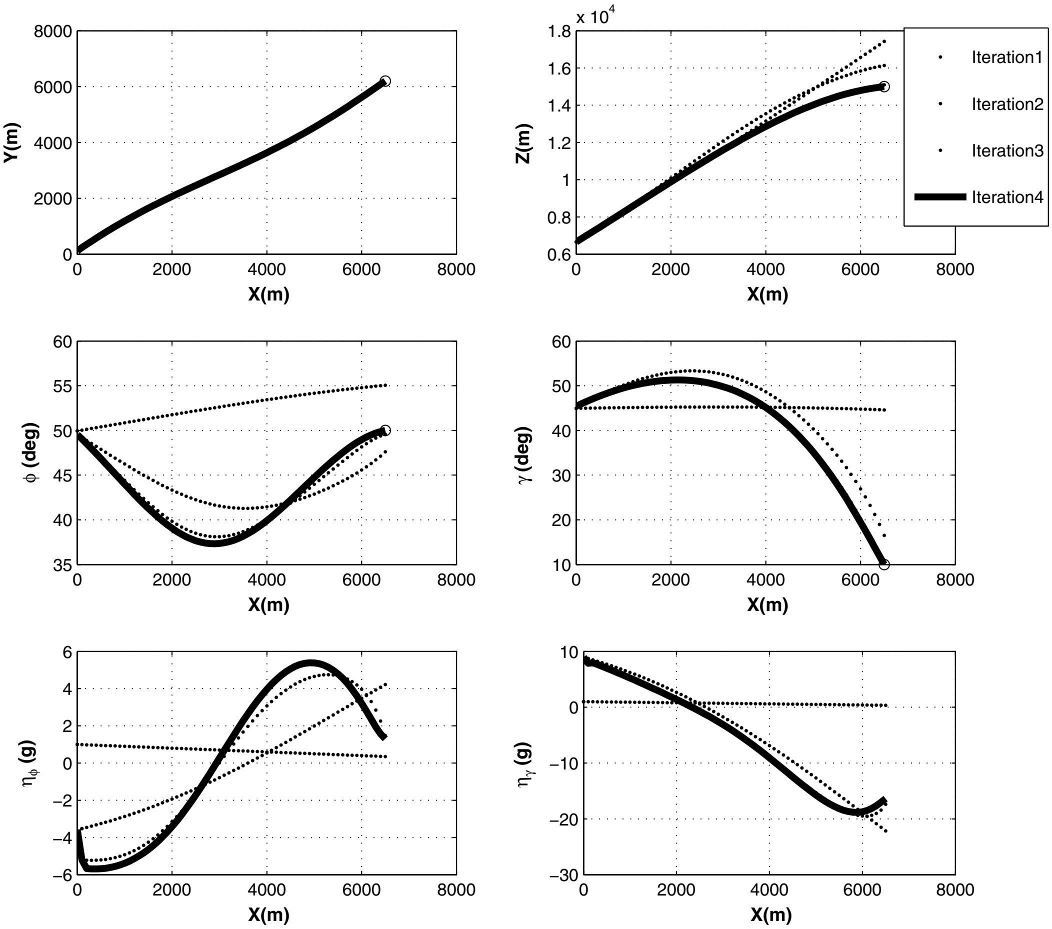

Fig. 3 Results of MPSP formulation (case 2).

IV. Midcourse Guidance with Model Predictive Static Programming/Model Predictive Spread Control

A. Objective of Midcourse Guidance

For an interceptor, guidance works in three phases. The first is the boost phase, the second is the midcourse, and the third and last is the terminal phase. For successful interception of a high-speed target, it is very important that, during the terminal guidance phase, the interceptor and target velocity vectors are near antiparallel. Moreover, the interceptor must have sufficient capability to fulfill the terminal guidance requirement. So it is quite important for midcourse guidance to provide the proper initial condition to the terminal guidance phase. As the interceptor spends most of its time in the midcourse guidance phase, it is quite imperative that this phase should be energy efficient while simultaneously achieving its primary performance-related objectives. Hence, an optimal midcourse guidance must enable the interceptor to reach a particular point at a particular range-to-go with the desired velocity vector, also with minimum effort. This desired point is decided by a prelaunch computation scheme. Moreover, the missile must reach the desired location with the proper flight-path angle and heading angle while retaining sufficient velocity to satisfy the terminal guidance requirement due to handover errors and the subsequent requirements due to the target maneuver. The desired flight-path angle and heading angle are important for the warhead point of view.

Therefore, the objective of midcourse guidance is the following: the interceptor has to reach the desired point (x_d, y_d, z_d) with the desired heading angle ϕ_d and flight-path angle γ_d using minimum acceleration η_ϕ and η_γ.

B. Guidance Design with Model Predictive Static Programming

In the state equation of the interceptor, time is used as an independent variable. However, to eliminate the open-loop nature of the solution, instead of time, the downrange x has been used as an independent variable. This is possible because x is a monotonic variable and the final value of x is known (since the missile has to reach a desired point after the midcourse phase). For this purpose, the missile model is first written as

$$\begin{bmatrix} t' \\ y' \\ z' \\ V' \\ \phi' \\ \gamma' \end{bmatrix} = \begin{bmatrix} \frac{1}{V\cos\gamma\cos\phi} \\ \tan\phi \\ \frac{\tan\gamma}{\cos\phi} \\ \frac{T-D}{mV\cos\gamma\cos\phi} \\ \frac{g\eta_\phi}{V^2\cos^2\gamma\cos\phi} \\ \frac{g(\eta_\gamma-\cos\gamma)}{V^2\cos\gamma\cos\phi} \end{bmatrix} \tag{67}$$

where X' represents the derivative of state with respect to position x. For the MPSP, the design state model has to be in discreet form as

$$X_{k+1} = F_k(X_k, U_k) \tag{68}$$

Where states X_k, U_k, and F_k are defined as

$$X_k = [t_k \quad y_k \quad z_k \quad V_k \quad \phi_k \quad \gamma_k]^T \tag{69}$$

$$U_k = [\eta_{\phi_k} \quad \eta_{\gamma_k}]^T \tag{70}$$

Table 3 Convergence results of MPSP (case 2)

Iteration	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg
1	2614.4	2598.4	5.1	34.5
2	197.79	1061.8	2.2	4.3
3	1.090	77.06	−0.85	0.88
4	0.95	−0.63	0.02	−0.02

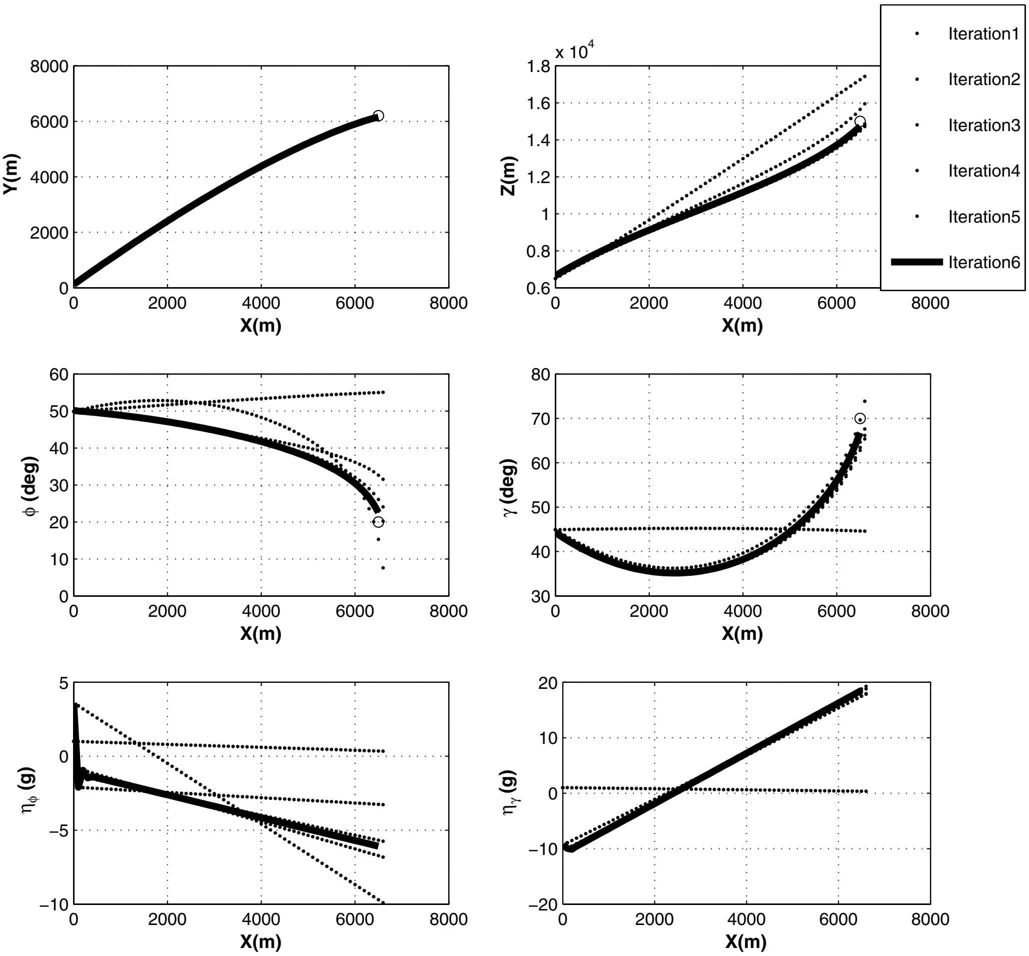

Fig. 4 Results of MPSC: straight line formulation (case 1).

$$F_k = \begin{bmatrix} t_k + \frac{h}{V_k \cos\gamma_k \cos\phi_k} \\ y_k + h\tan\phi_k \\ z_k + \frac{h\tan\gamma_k}{\cos\phi_k} \\ V_k + \frac{h(T_k - D_k)}{mV_k \cos\gamma_k \cos\phi_k} \\ \phi_k + \frac{hg\eta_{\phi_k}}{V_k^2 \cos^2\gamma_k \cos\phi_k} \\ \gamma_k + \frac{hg(\eta_{\gamma_k} - \cos\gamma_k)}{V_k^2 \cos\gamma_k \cos\phi_k} \end{bmatrix} \tag{71}$$

The following steps are used for midcourse guidance based on the MPSP technique:

1) Initialize η_{ϕ_k} and η_{γ_k} with some guess, such that the guess trajectory is not very far from the desired trajectory.

2) Define the present state as $k = 1$ and the desired state as $k = N$ and calculate the increment of $x(h)$ as

$$h = \frac{x_d - x_1}{N - 1} \tag{72}$$

3) Propagate the point-mass model of the missile using η_{ϕ_k} and η_{γ_k} until x_d to get the final state of the missile X_f and find $\mathrm{d}Y_N$ as

$$\mathrm{d}\,Y_N = \begin{bmatrix} y_f - y_d \\ z_f - z_d \\ \phi_f - \phi_d \\ \gamma_f - \gamma_d \end{bmatrix} \tag{73}$$

4) If either element of $\mathrm{d}Y_N$ is not within the desired limit, a correction has to be made for η_{ϕ_k} and η_{γ_k}. For this, B^0_{N-1} is calculated as

$$B^0_{N-1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{74}$$

B_{N-1} can be calculated as

$$B_{N-1} = B^0_{N-1}\left[\frac{\partial F_{N-1}}{\partial U_{N-1}}\right] \tag{75}$$

where $\partial F_{N-1}/\partial U_{N-1}$ is obtained from Eq. (71) at the $(N-1)$th step.

5) For $k = 1, \dots, N-1$, B_k can be calculated as

$$B_k = B^0_k\left[\frac{\partial F_k}{\partial U_k}\right] \tag{76}$$

where

$$B^0_k = B^0_{k+1}\frac{\partial F_{k+1}}{\partial X_{k+1}} \tag{77}$$

Table 4 Convergence results of MPSC: straight line formulation (case 1)

Iteration	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg
1	2614.4	2598.4	35	25
2	882.1	1412.4	−36.1	12.2
3	183.3	126.4	10.1	−1.6
4	−30.04	37.2	−3.6	1.3
5	14.63	−18.04	1.34	−0.55
6	−0.81	0.30	−0.58	0.25

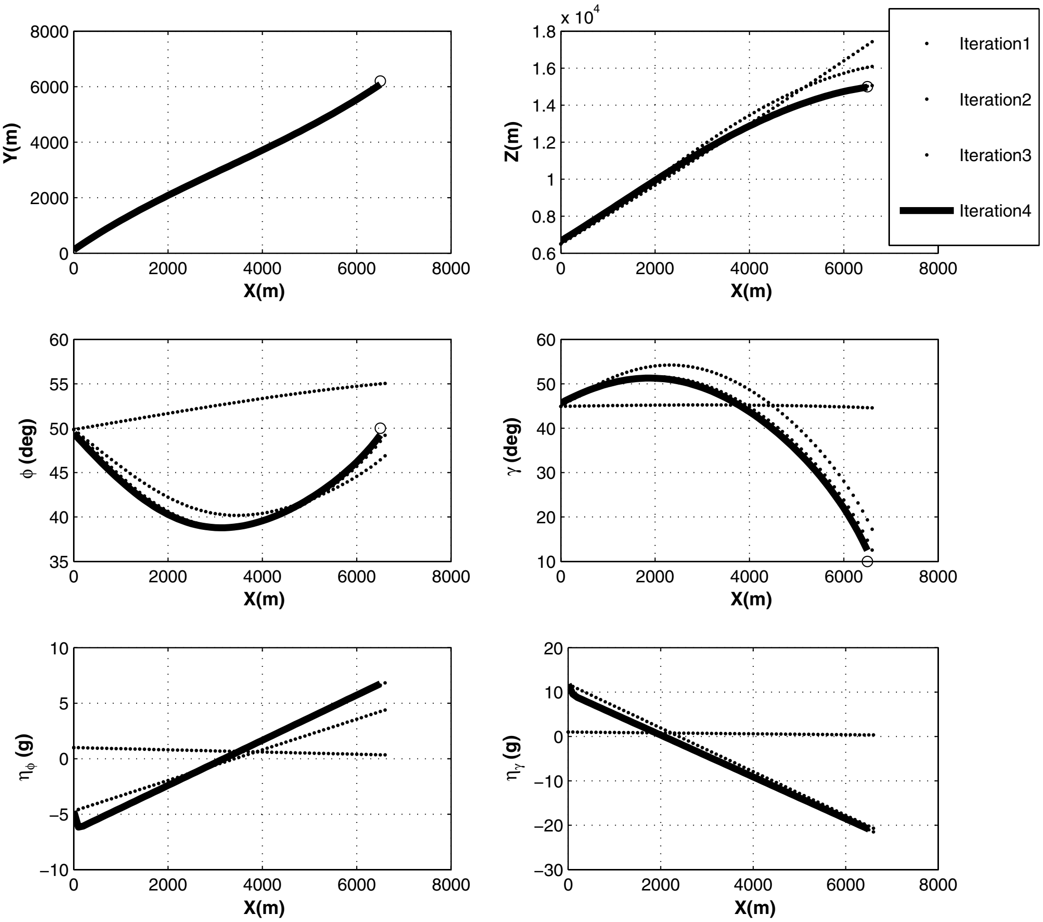

Fig. 5 Results of MPSC: straight line formulation (case 2).

$\partial F_{k+1}/\partial U_{k+1}$ and $\partial F_{k+1}/\partial X_{k+1}$ can be calculated from Eq. (71) at the $(k+1)$th step.

6) Once B_k is calculated, B_λ and A_λ can be calculated from Eq. (19), and η_{ϕ_k} and η_{γ_k} can be written as

$$\begin{bmatrix} \eta_{\phi_k} \\ \eta_{\gamma_k} \end{bmatrix} = B_k^T A_\lambda^{-1}[B_\lambda - \mathrm{d}Y_N] \tag{78}$$

Here, R_k is chosen as the unit matrix.

7) Once we obtain the new value of η_{ϕ_k} and η_{γ_k}, this will be used as the guidance command.

C. Guidance Design with Model Predictive Spread Control: Straight Line Formulation

This formulation is the same as the MPSP. Here, η_{ϕ_k} and η_{γ_k} are approximated as

$$\eta_{\phi_k} = a_1 x_k + b_1 \tag{79}$$

$$\eta_{\gamma_k} = a_2 x_k + b_2 \tag{80}$$

The steps followed in this method are the same as the MPSP, apart from the first and sixth steps:

1) The variables a_1, a_2, b_1, and b_2 have to be initialized.

6) The variables a_1, a_2, b_1, and b_2 can be obtained as

$$\begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \left[\sum_{k=1}^{N-1} B_k x_k \sum_{k=1}^{N-1} B_k \right]^{-1} [B_\lambda - \mathrm{d}Y_N] \tag{81}$$

The rest of the steps are same as the MPSP design.

D. Guidance Design with Model Predictive Spread Control: Quadratic Formulation

This formulation is the same as the MPSP. Here, η_{ϕ_k} and η_{γ_k} are approximated as

$$\eta_{\phi_k} = a_1 x_k^2 + b_1 x_k + c_1 \tag{82}$$

$$\eta_{\gamma_k} = a_2 x_k^2 + b_2 x_k + c_2 \tag{83}$$

In this case, however, the number of free variables are more than the number of constraint equations. Hence, the coefficients are updated by formulating an appropriate optimization problem as described before (see Sec. II.B.2). Further specific details are omitted here to contain the length of the paper.

V. Nonlinear Controller Design for Six-Degree-of-Freedom Simulation

Since, in addition to the results from point-mass simulation studies, our aim was to validate the results from six-DOF simulation studies as well, there is a need for realizing the lateral acceleration demands from the guidance loop through an appropriate inner loop for control design. For this purpose, we have used the nonlinear

Table 5 Convergence results of MPSC: straight line formulation (case 2)

Iteration	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg
1	2614.4	2598.4	5.1	34.5
2	114.5	1132.9	−2.6	4.9
3	9.48	86.7	−0.03	0.08
4	0.73	0.14	−0.005	0.08

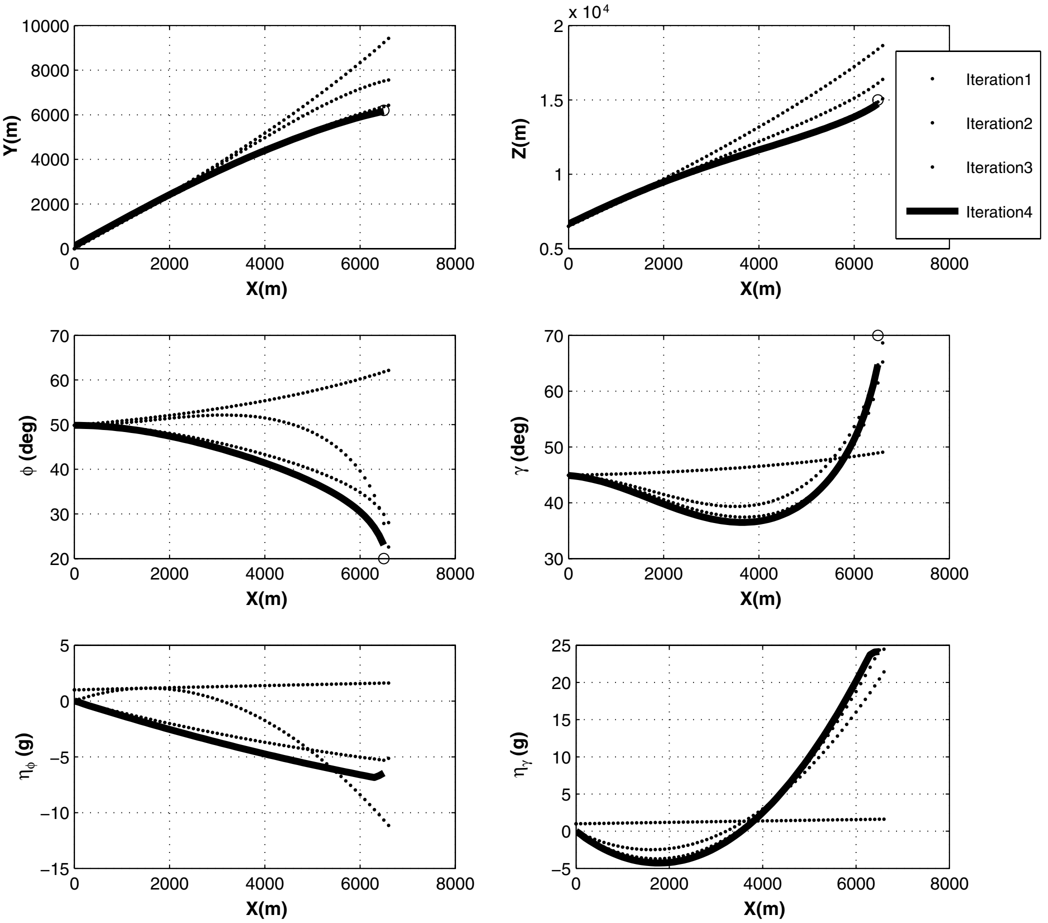

Fig. 6 Results of MPSC: quadratic formulation (case 1).

dynamic inversion philosophy, the details of which are discussed in this section. Note that the dynamic inversion control design approach presented here exploits the time-scale separation that inherently exists in the missile dynamics, and hence avoids the difficulty of dealing with the nonminimum phase problem. This is possible because, although the whole system is of a nonminimum phase, each fast and slow dynamics, separated from the original dynamics according to the time response with respect to the input variables, is of a minimum phase.

Therefore, the separated dynamics can be controlled by the feedback liberalization technique. That is, the outer-loop inversion controller uses the states of the fast dynamics to control those of the slow dynamics, and the inner-loop inversion controller uses the control fin deflections to control the states of the fast dynamics. In this approach, states p, q, and r are identified as the faster dynamic response, while α, β, and ζ are characterized as the slow state variables.

The angular rates p, q, and r strongly depend upon the fin deflections. Thus, to start with, fast-state controllers for p, q, and r were designed. Having designed a fast-state controller, a separate, approximate inversion procedure was carried out to design the slow state controller for α, β, and ζ. It may be noted that such a model reduction was possible, as there is a significant difference in the time-scale between the fast and slow states in the open-loop dynamics of the tactical flight vehicle.

A. Outer-Loop Design

As stated earlier, the design of this controller is dependent on two time-scale separations (see Fig. 1). The outer loop operates on an error between the commanded and achieved slow variables (lateral acceleration and desired roll orientation). Its dynamics are related to slowly varying scheduling parameters: i.e., turning rate time constant. It generates the commanded roll rate, pitch rate, and yaw rate, which are input to the inner loop of faster dynamics. The acceleration command coming from the estimator will be used to compute the desired dynamics. Here, first-order dynamics are considered for desired dynamics as

$$\begin{bmatrix} \dot{a}_{y_c} \\ \dot{a}_{z_c} \end{bmatrix} = \begin{bmatrix} \omega_\alpha(a_{y_c} - ay) \\ \omega_\beta(a_{z_c} - az) \end{bmatrix} \tag{84}$$

Once the $\dot{a}y_c$ and $\dot{a}z_c$ are known, these can be converted to $\dot{\alpha}_c$ and $\dot{\beta}_c$ as follows:

$$\begin{bmatrix} \dot{\alpha}_c \\ \dot{\beta}_c \end{bmatrix} = \begin{bmatrix} \dot{a}_{z_c} \frac{\partial\alpha}{\partial az} \\ \dot{a}_{y_c} \frac{\partial\beta}{\partial ay} \end{bmatrix} \tag{85}$$

where $\partial\alpha/\partial az$ and $\partial\beta/\partial a_y$ are defined as

$$\begin{bmatrix} \frac{\partial\alpha}{\partial az} \\ \frac{\partial\beta}{\partial ay} \end{bmatrix} = \begin{bmatrix} \frac{m}{QSC_{N_\alpha}} \\ \frac{m}{QSC_{Y_\beta}} \end{bmatrix} \tag{86}$$

Table 6 Convergence results of MPSC: quadratic formulation (case 1)

Iteration	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg
1	3415.9	3903.12	42.5	20.8
2	1391.8	1698.8	−8.3	5
3	273.7	360.4	5.1	0.2
4	−0.59	0.31	−0.28	−0.39

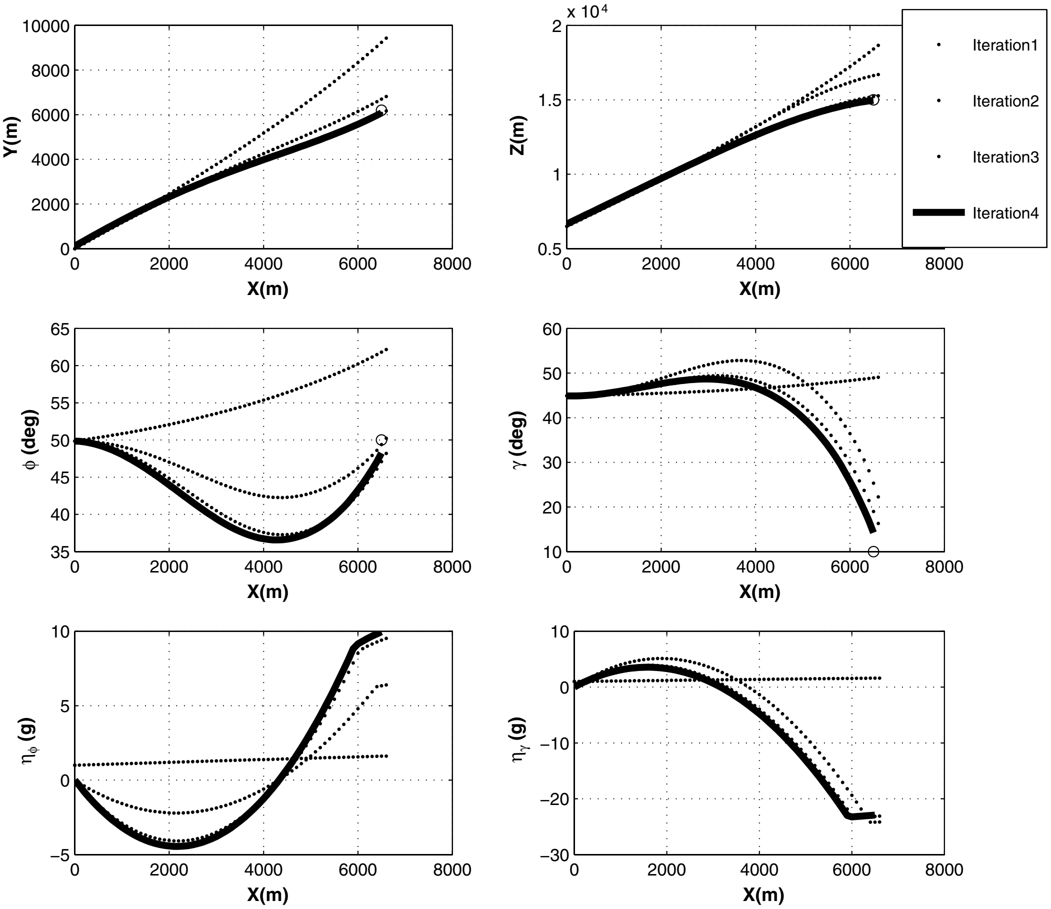

Fig. 7 Results of MPSC: quadratic formulation (case 2).

Now, the force equation can be inverted to get the body rate as

$$\begin{bmatrix} p_c \\ q_c \\ r_c \end{bmatrix} = \begin{bmatrix} \omega_\zeta(\zeta_c - \zeta) \\ \dot{\alpha}_c + \frac{1}{\cos\beta}[(p\cos\alpha + r\sin\alpha)\sin\beta + \frac{1}{mV_sM}(-0.5\rho V^2 SC_L - mg\cos\gamma_i\cos\mu_i)] \\ \frac{1}{-\cos\alpha}[\dot{\beta}_c - p\sin\alpha - \frac{1}{mV_sM}(0.5\rho V^2 SC_Y + mg\cos\gamma_i\sin\mu_i)] \end{bmatrix} \tag{87}$$

where the demanded roll angle ζ_c is zero.

B. Inner-Loop Design

This loop generates fin deflections that control tactical flight vehicle dynamics.

The rate command coming from the outer loop will be used to compute the desired dynamics. Here, second-order dynamics are considered for the desired dynamics, which are given as

$$\begin{bmatrix} \dot{p}_c \\ \dot{q}_c \\ \dot{r}_c \end{bmatrix} = \begin{bmatrix} -2\xi\omega_p p + \omega_p^2 \int(p - p_c) \\ -2\xi\omega_q q + \omega_q^2 \int(q - q_c) \\ -2\xi\omega_r r + \omega_r^2 \int(r - r_c) \end{bmatrix} \tag{88}$$

Once the $\dot{p}_c$, $\dot{q}_c$, and $\dot{r}_c$ are known, the rate equation can be inverted to get the fin deflection as

$$\begin{bmatrix} C_{l_c} \\ C_{m_c} \\ C_{n_c} \end{bmatrix} = \begin{bmatrix} \frac{\{(\dot{p}_c - [(I_y - I_z)rq/I_x]\}I_x}{0.5\rho V^2 Sd} \\ \frac{\{\dot{q}_c - [(I_z - I_x)rp/I_y]\}I_y}{0.5\rho V^2 Sd} \\ \frac{\{\dot{r}_c - [(I_x - I_y)qp/I_z]\}I_z}{0.5\rho V^2 Sd} \end{bmatrix} \tag{89}$$

where C_{l_c}, C_{m_c}, and C_{n_c} are the commanded aerodynamic coefficients, which are used further to obtain the fin deflection as

$$\begin{bmatrix} \delta_p \\ \delta_q \\ \delta_r \end{bmatrix} = \begin{bmatrix} \frac{C_{l_c} - [(C_{l_p} dp)/2V]}{C_{l_{\delta_p}}} \\ \frac{C_{m_c} - C_{m_q} q - C_{m_\alpha}\alpha}{C_{m_{\delta_q}}} \\ \frac{C_{n_c} - C_{n_r} r - C_{n_\beta}\beta}{C_{n_{\delta_r}}} \end{bmatrix} \tag{90}$$

For further detail, one can refer to [28], where the scheduling of the autopilot (gain selection) is given in detail.

VI. Results and Discussion

To show the performance of MPSP guidance, the results have been divided into three sections. First, the point-mass model of the missile

Table 7 Convergence results of MPSC: quadratic formulation (case 2)

Iteration	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg
1	3415.9	3903.12	12.5	39.1
2	736	1758.6	1.2	8.6
3	82.76	308.4	−0.5	3.2
4	−0.57	0.71	−0.46	1.21

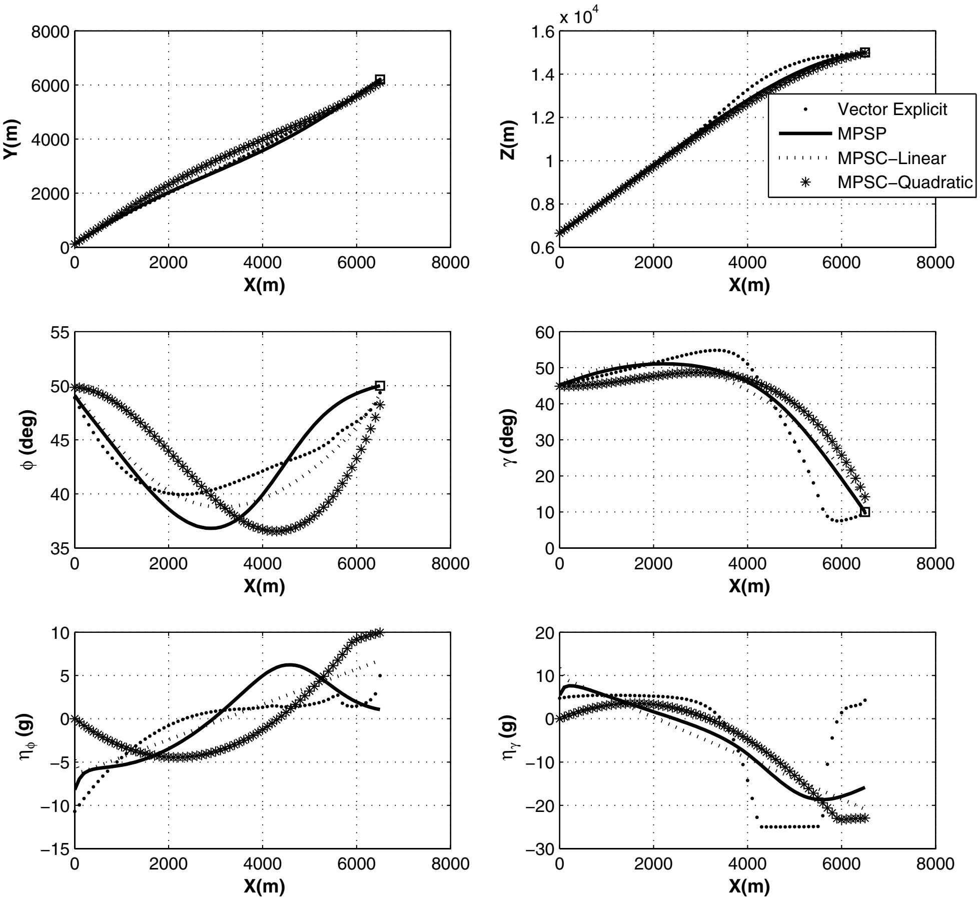

Fig. 8 Comparison of result with VEG for case 2.

has been considered without autopilot lag to evaluate these guidance. Here, the objective is to reach a specified point with a specified flight-path angle and heading angle. In the second part, the complete six-DOF equation has been used to carry out the simulation. A nonlinear dynamic inversion autopilot has been used here as a controller in the six-DOF simulation given in [28]. A ballistic missile has been used as a target for the simulation. Lastly, the robustness results of this guidance are discussed.

A. Results with Point-Mass Model of Interceptor

In this section, the point-mass model of the missile has been used for simulation. For each guidance methodology, the same initial condition of the missile has been considered and is shown in Table 1. Two cases for different desired final conditions have been chosen to test each method. These cases are shown in Table 1.

In both cases, the desired final position is the same, but the desired final heading angle ϕ and flight-path angle γ are different.

As discussed before, the proposed guidance needs an initial guess for the guidance command history. The choice of initial guess should be such that the initial trajectory should not be very far from the desired trajectory. In this work, the initial guess has been chosen as a straight line based on a bit of heuristic study, the details of which are omitted.

1. Results with Model Predictive Static Programming

In case 1, the missile is required to reach the desired position with $\phi = 20°$ and $\gamma = 70°$. The first subplot (Fig. 2) shows the X versus Y position of the missile. The first guess of acceleration is represented as a straight dotted line. The final path achieved by the missile is shown as a solid line, and the rest are the intermediate paths that converge to the final path after five iterations. The desired final position is shown as a circle. A similar pattern of convergence can be observed in the second subplot in Fig. 2, which shows the X versus Z position of the interceptor. The third and fourth subplots (Fig. 2) show the demanded (circle point) and achieved heading angle and flight-path angle of the missile (solid line). The fifth and sixth subplots (Fig. 2) show the acceleration required (solid line) to achieve the desired goal. The remaining lines represent the intermediate history of acceleration. It can be observed here that a total of five iterations is required to achieve the desired performance. The convergence results are tabulated in Table 2. It can be observed here that convergence is very fast, even if the initial error is quite high.

In case 2, the missile is required to reach the desired position with $\phi = 50°$ and $\gamma = 10°$. Figure 3 shows the X versus Y and X versus Z

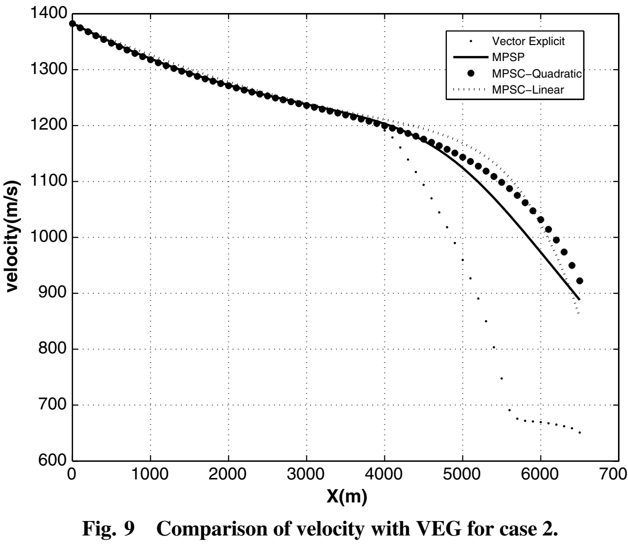

Fig. 9 Comparison of velocity with VEG for case 2.

Table 8 Comparison of results (case 2)

Guidance	$y_f - y_d$, m	$z_f - z_d$, m	$\phi_f - \phi_d$, deg	$\gamma_f - \gamma_d$, deg	Final velocity, m/s	Integration of total acceleration, I
Vector explicit	−0.1407	−0.0582	−0.6491	−0.2698	650	1772.6
MPSP	0.95	−0.63	0.02	−0.02	890	1353.3
MPSC: linear	0.73	0.14	−0.005	0.08	860	1390.2
MPSC: quadratic	−0.57	0.71	−0.46	1.21	920	1310.2

positions of the missile, the heading angle, the flight-path angle, and the acceleration required. It was observed here that only four iterations are required in this case to achieve the desired performance. The output errors $\mathrm{d}Y_N$ are shown in Table 3, which clearly show the good convergence nature of the algorithm.

2. *Results with Model Predictive Spread Control: Straight Line Formulation*

In case 1, the missile is required to reach the desired position with $\phi = 20°$ and $\gamma = 70°$. Figure 4 shows the X versus Y and X versus Z positions of the missile, the heading angle, the flight-path angle, and the acceleration required. It can be observed here that a total of six iterations are required to achieved desired performance, compared with five iterations in the MPSP guidance. The convergence results are given in Table 4.

In case 2, the missile is required to reach the desired position with $\phi = 50°$ and $\gamma = 10°$. Figure 5 show the X versus Y and X versus Z positions of the missile, the heading angle, the flight-path angle, and the acceleration required. It can be observed here that the total number of iterations is the same as the MPSP guidance. The convergence results are given in Table 5.

3. *Result with Model Predictive Spread Control: Quadratic Formulation*

In case 1, the missile has to reach a desired position with $\phi = 20°$ and $\gamma = 70°$. The first subplot (Fig. 6) shows the X versus Y position of the missile. Here, the convergence of the trajectory from the initial guess to the final iteration has been shown. A similar phenomena can be observed in the second subplot, which shows the X versus Z position of the interceptor. The third and fourth subplots of Fig. 6 show the heading angle and flight-path angle demanded and achieved by the missile. The fifth and sixth subplots of Fig. 6 show the acceleration required to achieve the desired goal (solid line). The remaining lines are the intermediate history of acceleration. It can be observed here that a total of four iterations are required to achieved desired performance compared with five iterations in the MPSP guidance and six iterations in the MPSC: linear guidance. The convergence results are given in Table 6.

In case 2, the missile has to reach a desired position with $\phi = 50°$ and $\gamma = 10°$. The first subplot of Fig. 7 shows the X versus Y position of the missile. The second subplot of Fig. 7 shows the X versus Z position of the interceptor. The third and fourth subplots of Fig. 7 show the heading angle and flight-path angle demanded (circle point) and achieved by the missile (solid line). The fifth and sixth subplots of Fig. 7 show the acceleration required to achieve the desired goal. It can be observed here that the total number of iterations is the same as the MPSP guidance. The convergence result is given in Table 7, which clearly shows the rapid convergence nature.

Based on the preceding results, it can be observed that MPSC: quadratic has better convergence properties than the other two for the same conditions, perhaps because the nature of the final solution is quadratic.

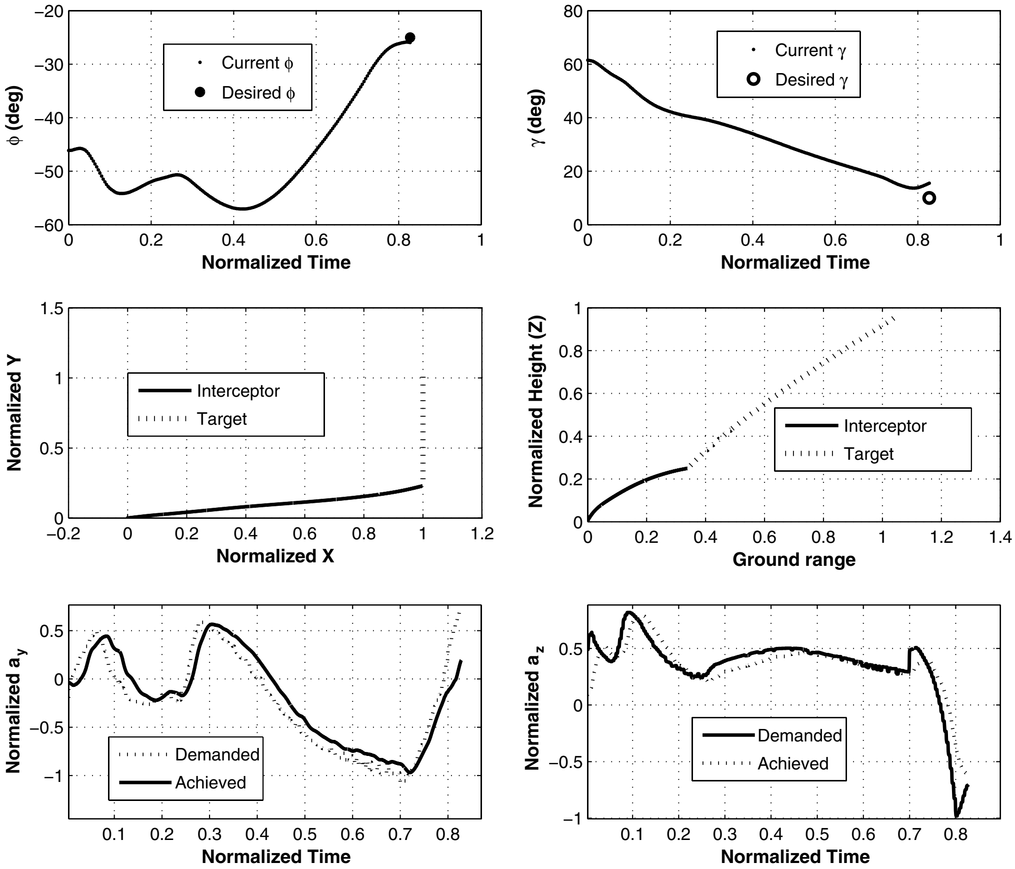

Fig. 10 Six-DOF results with MPSP.

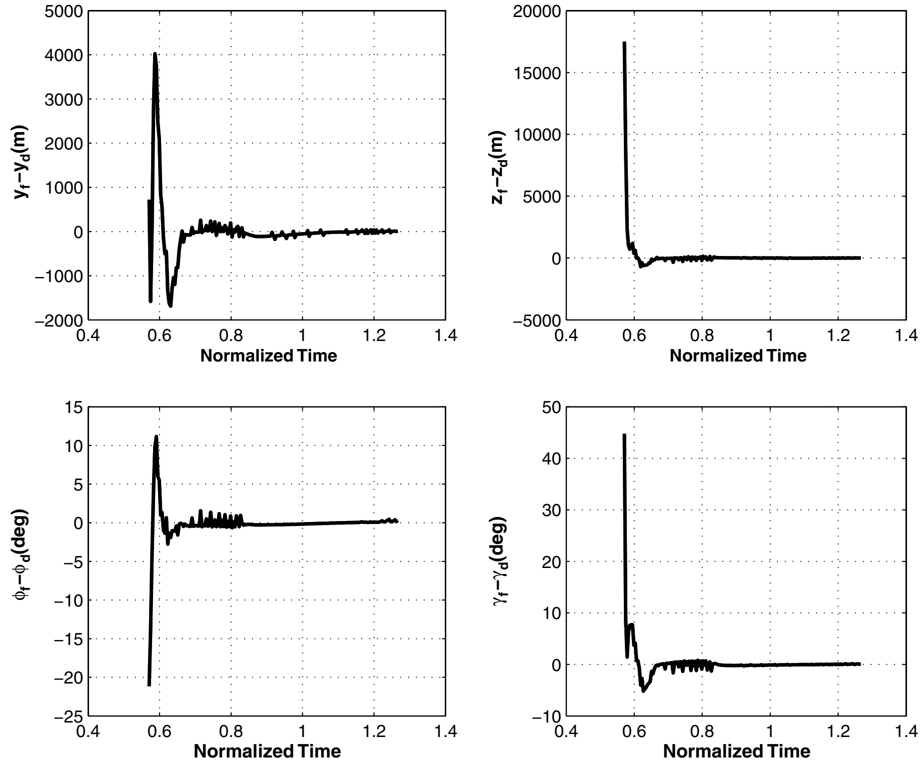

Fig. 11 Convergence of MPSP.

4. Comparison of Results with Vector Explicit Guidance

Even though the results of the proposed guidance laws are quite promising, a natural question arises about how they compare with some of the existing guidance laws that are capable of achieving the same objective. In this connection, the results of the three proposed guidance schemes have also been compared with the generalized vector explicit guidance (VEG) scheme [29,30], which proposes a closed-form solution for the linearized version of the problem. In the

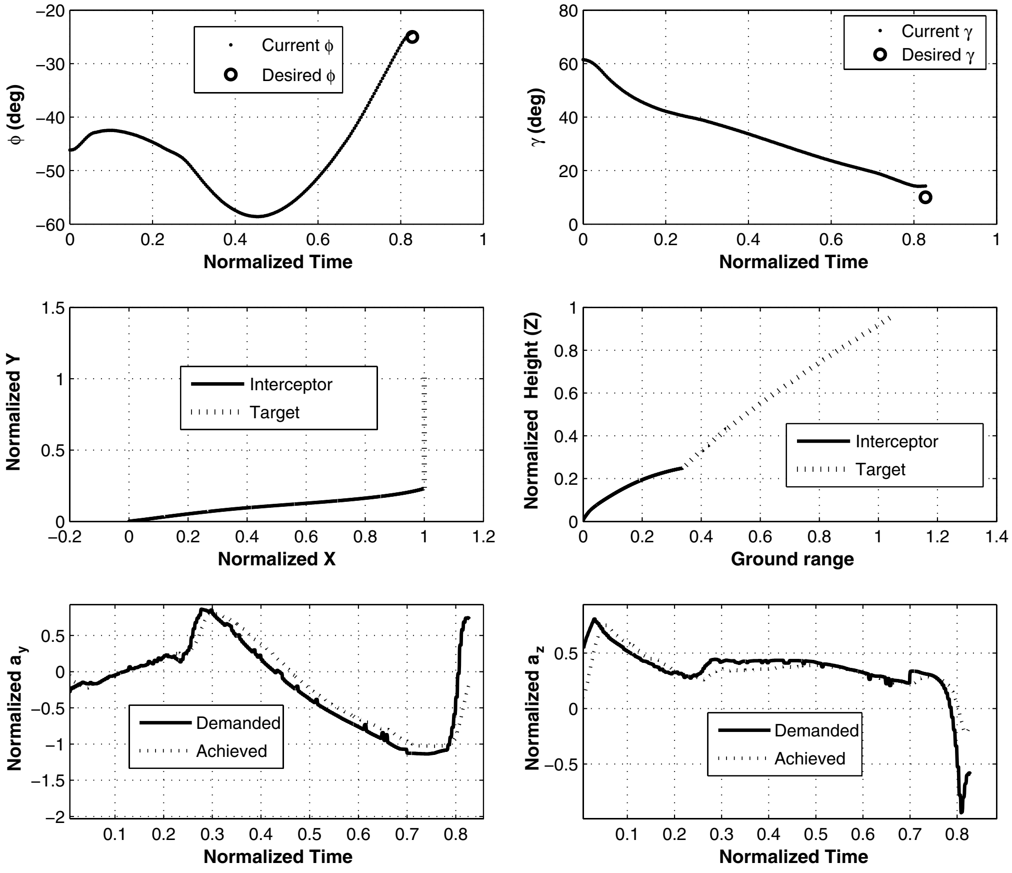

Fig. 12 Six-DOF results with MPSC: linear.

VEG scheme, the acceleration command a_d can be expressed as the sum of two terms as follows:

$$\mathbf{a}_{\mathbf{d}} = \frac{1}{t_{\text{go}}^2}(K_1\mathbf{ZEM} + K_2\mathbf{\Delta LOS}) \tag{91}$$

where

$$\mathbf{ZEM} = \mathbf{X}_{\mathbf{T}}(t) - \mathbf{X}(t) + \{\mathbf{V}_{\mathbf{T}}(t) - \mathbf{V}(t)\}t_{\text{go}}$$
$$\mathbf{\Delta LOS} = \mathbf{X}_{\mathbf{T}}(t) - \mathbf{X}(t) + \{\mathbf{V}_{\mathbf{T}}(t_f) - \mathbf{V}(t_f)\}t_{\text{go}}$$

where $\mathbf{ZEM}$ is the zero effort miss vector, $\mathbf{\Delta LOS}$ is the difference between the current and desired LOS directions, and K_1 andK_2 are generalized guidance gains. The value of gains K_1 and K_2 have been chosen as six and two, respectively. Note that the first term $K_1(\text{ZEM}/t_{\text{go}}^2)$ is equivalent to the PN guidance law with navigation gain K_1, while the second term $K_2(\Delta\text{LOS}/t_{\text{go}}^2)$ is the trajectory shaping part that ensures a specific terminal aspect angle requirement. For more details, one can refer to [29,30].

The comparison results have been shown in Figs. 8 and 9. It can be seen here that all four guidance laws are able to meet the end condition requirements. However, in VEG, the velocity loss is relatively high compared with the proposed guidance laws in this paper. This is due to the fact that the lateral acceleration demands are higher in VEG. Because of this, the induced drag is higher, which leads to higher velocity loss. It can also be observed here that the final velocity is maximum for the MPSC: quadratic guidance law. It is because the acceleration demand is lesser in the MPSC: quadratic scheme. Note that fairly similar results were obtained for a number of cases with different initial conditions.

5. *Comparison of Results for Total Lateral Acceleration Demand*

Comparing the results further, the total root-mean-square lateral acceleration demand I has also been calculated for all cases, which have been defined as

$$I = \int_0^t \sqrt{\eta_\phi^2 + \eta_\gamma^2}\,\mathrm{d}t \tag{92}$$

The summary of results has been tabulated in Table 8. It can be observed here that the value of I is maximum for VEG and minimum for the MPSC: quadratic case. It can also be observed that the integration of total acceleration I is comparable for the proposed three guidance laws in this paper, and they are substantially lesser than the VEG scheme.

In Fig. 8, it can be seen that, in the VEG law, the acceleration demand is saturating, but for the proposed guidance laws, the guidance demand saturation has been avoided. The comparison of the lateral acceleration of the four cases has been shown in Table 8.

B. Results with Six-Degree-of-Freedom Model of Interceptor

In this section, the complete six degrees of freedom of the missile have been used with the nonlinear dynamic inversion autopilot [28]. The target used here is a ballistic target with an initial velocity of 2500 m/s. Midcourse guidance is executed until a 10 km range-to-go following the terminal guidance: i.e., PN guidance with navigation gains three. In this simulation, the objective of midcourse guidance is to intercept the target with a heading angle ϕ of $-25°$ and a flight-path angle γ of 10°. To calculate the desired PIP, the target model has to be extrapolated up to the desired height of kill, as discussed in Sec. III.C. Six-DOF simulation is carried out with the first-order actuator model having a 10 Hz bandwidth [4].

1. *Results with Model Predictive Static Programming*

The first and second subplots in Fig. 10 show the ϕ and γ achieved by the missile. Here, it can be observed that MPSP guidance has driven the interceptor to intercept the target with the required ϕ and γ as specified. The deviation observed in the angles toward the end is because of the switch to terminal guidance at a 10 km range-to-go, and terminal guidance does not guarantee the angular constraints.

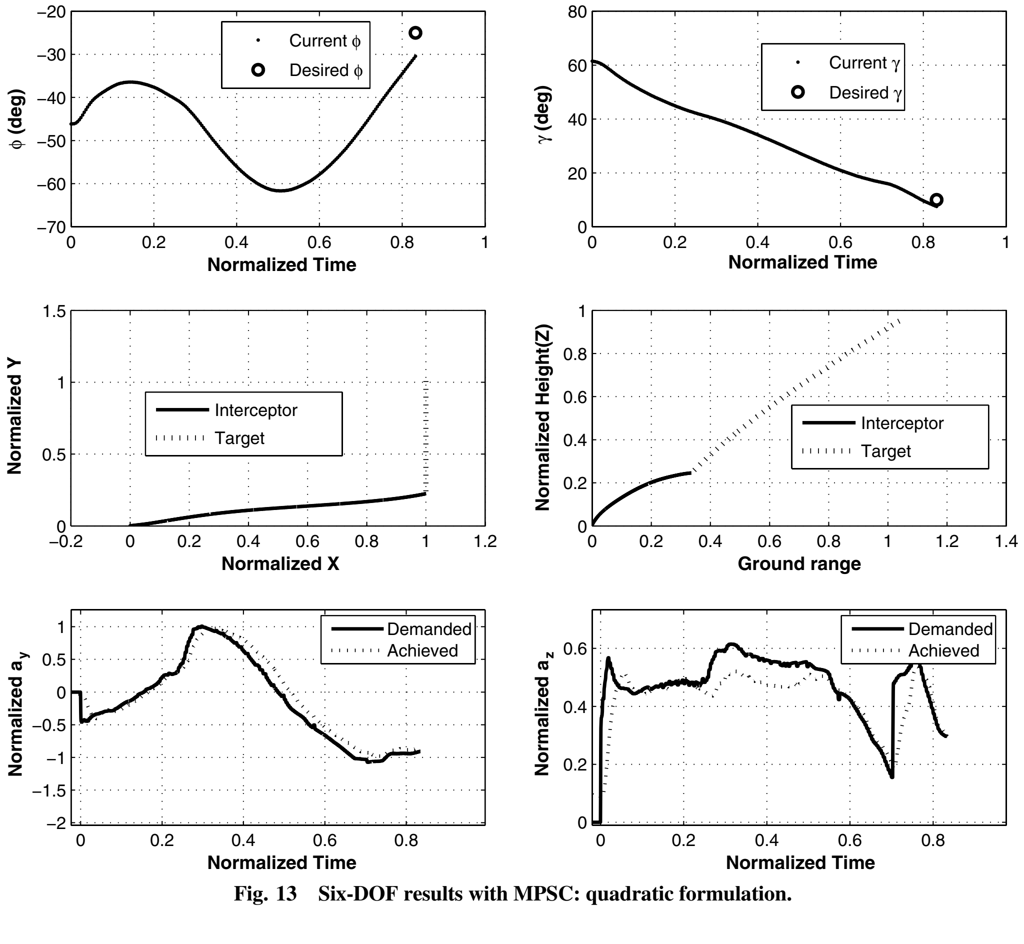

Fig. 13 Six-DOF results with MPSC: quadratic formulation.

The third and fourth subplots in Fig. 10 show the trajectories of the missile and the target. The fifth and sixth subplots in Fig. 10 show the demanded and achieved accelerations by the autopilot. Figure 11 shows the convergence of the MPSP method. It can be observed here that convergence is very fast, even when the initial error is very high. Hence, it can be inferred that the initial guess is not very critical in the MPSP method. Here, the initial guess of acceleration is taken as zero, because the initial error is relatively quite high. Some oscillation has been observed in guidance command generation and in convergence results. It is because the predicted model of the missile used in MPSP guidance is the simplified point-mass model, and the real plant is the complete six-DOF with autopilot lag. Because of this, once the error is converged within the desired bound, it is not guaranteed to stay in that bound. However, as soon as the error crosses the desired bound, the MPSP guidance again brings it back within the desired bound. Because of this, some oscillation is present in the lateral acceleration generation process, as observed in Fig. 11.

2. *Results with Model Predictive Spread Control: Straight Line Formulation*

In this section, MPSC guidance with linear approximation has been used. The missile has to reach a desired position with $\phi = -25°$ and $\gamma = 10°$. The first and second subplots of Fig. 12 show ϕ and γ achieved by the missile. It can be observed here that, with help, this guidance interceptor is able to hit the target with specified ϕ and γ. The third and foruth subplots of Fig. 12 show the engagement trajectory of the missile and the target. The fifth and sixth subplots of Fig. 12 show the acceleration demanded and the acceleration achieved by the autopilot. Here, it can be observed that the results of both MPSP and MPSC are nearly similar but not the same, because in the MPSC method, straight line approximation of acceleration has been done whether the acceleration is approximately quadratic in nature.

Note that the robustness results of the MPSC in the straight line formulation are quite similar to the MPSP results and are omitted here for brevity.

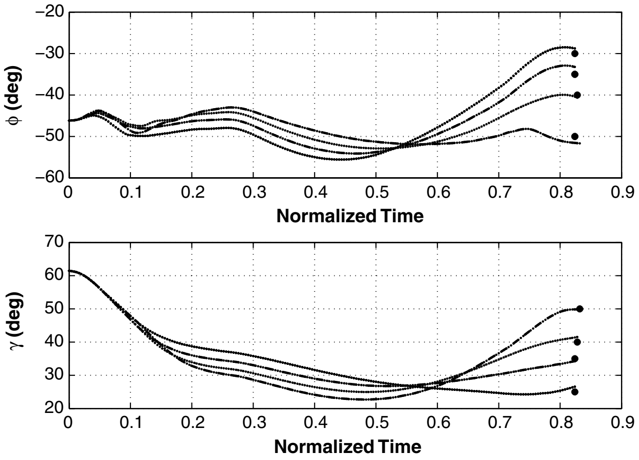

Fig. 14 Heading angle ϕ and flight-path angle γ vs time for different ϕ and γ.

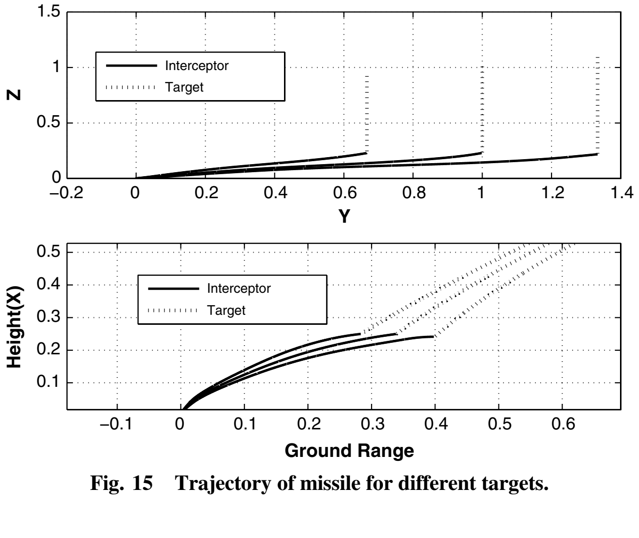

Fig. 15 Trajectory of missile for different targets.

Table 9 Results of MPSP with different perturbation cases of missile

Case	Thrust	C_N	$C_{N\delta}$	C_m	ϕ, deg	γ, deg
1	Nominal	Nominal	Nominal	Nominal	−26.5	13
2	+10%	+10%	+10%	+10%	−27	14.1
3	+10%	−10%	−10%	−10%	−26.6	13.9
4	−10%	+10%	+10%	+10%	−24.4	12.4
5	−10%	−10%	−10%	−10%	−24.6	12.6

3. *Results with Model Predictive Spread Control: Quadratic Formulation*

In this section, the MPSC with the quadratic approximation of the lateral acceleration has been used. The missile has to reach a desired position with $\phi = -25°$ and $\gamma = 10°$. The first and second subplots of Fig. 13 show the ϕ and γ achieved by the missile. It can be observed here that, with help, this guidance interceptor is able to hit the target with the specified ϕ and γ. The third and fourth subplots of Fig. 13 show the trajectory of the missile as well as the target. The fifth and sixth subplots of Fig. 13 show the acceleration demanded and achieved by the autopilot. Note that the results of the MPSC in quadratic formulation are quite similar to the MPSP formulation and are omitted here to contain the length of the paper.

C. Robustness of Proposed Guidance

To show the robustness of the proposed guidance algorithms (which has been verified with complete six-DOF simulations), many cases have been simulated for different trajectories of the target and for a different desired heading angle ϕ and flight-path angle γ, which are given in Fig. 14.

Figure 14 shows that with the help of this guidance, a different combination of ϕ and γ can be achieved. Here, in this figure, the initial condition for the missile and the target is the same for all cases, but the desired ϕ and γ are different. Also, in all the cases, the desired angles have been achieved. Figure 15 shows that this guidance works for different kinds of engagement scenarios without any change in guidance.

Note that these experiments have been carried out with various plant parameter perturbations as well, which are given in Table 9.

Table 9 summarizes the results of the MPSP guidance with a different perturbation of missile parameters. In the six-DOF model of the plant, various parameters like thrust, aerodynamic force coefficient CN, aerodynamic moment coefficient Cm, and control force coefficient CN_δ have been varied, as given in Table 9. For all the cases, the desired heading angle ϕ and flight-path angle γ have been chosen as 25 and 15°, respectively. It can be observed here that, even with different perturbations of the missile parameters, the performance of the guidance is satisfactory. Note that the achieved heading angle ϕ and flight-path angle γ are slightly different from the desired. This is because, during the last 10 km to go, PN-based guidance has been used, which does not impose any angle constraint.

Note that the robustness results of the MPSC formulation are quite similar and are omitted here for brevity.

VII. Conclusions

In this paper, MPSP and MPSC techniques have been used to propose effective alignment constrained suboptimal midcourse guidance laws for engaging incoming high-speed ballistic missile targets. Apart from being energy efficient by minimizing the lateral acceleration demands throughout the trajectory, both of these laws enforce desired alignment constraints in both elevation and azimuth.

Both point mass as well as six-DOF simulation results (with a realistic inner-loop autopilot based on dynamic inversion) are presented in this paper, which clearly shows the effectiveness of the proposed guidance laws. It has also been observed that, even with different perturbations of the missile parameters, the performance of the guidance is satisfactory. The time-to-go uncertainty issue has been avoided in this formulation by making use of the desired downrange-to-go, for which the accuracy is quite good, as the missile position is accurately known from the inertial navigation system and the desired position is known from prelaunched computation. The computational efficiency of the algorithms has also been demonstrated. Both MPSP as well as MPSC lead to promising results. However, the MPSC approach is marginally better, since it guarantees lateral acceleration smoothness as well as leads to a minor advantage from a computational time point of view. A comparison study with the VEG scheme also shows that the newly proposed MPSP- and MPSC-based guidance schemes lead to advantages like lesser lateral acceleration demand and lesser velocity loss during engagement.

References

[1] Zarchan, P., Tactical and Strategic Missile Guidance, 5th ed., Vol. 219, AIAA, Reston, VA, 1997, pp. 281–289.

[2] Naidu, D. S., “Singular Perturbations and Time Scales in Guidance and Control of Aerospace Systems: A Survey,” *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1057–1078. doi:10.2514/2.4830

[3] Bryson, A. E., “Application of Optimal Control Theory in Aerospace Engineering,” *Journal of Spacecraft and Rockets*, Vol. 4, No. 5, 1967, pp. 545–553. doi:10.2514/3.28907

[4] Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 424–426.

[5] Kumar, P., Dwivedi, P. N., and Bhattacharya, A., “Integrated Guidance and Control Against High Speed Targets with Desired Aspect Angle,” 2006 AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, CO, AIAA Paper 2006-6783, 2006.

[6] Padhi, R., “An Optimal Explicit Guidance Scheme for Ballistic Missile with Solid Motors,” AIAA Guidance, Navigation, and Control Conference, AIAA Paper 1999-4140, 1999.

[7] Sinha, S. K., and Shrivastava, S. K., “Optimal Explicit Guidance of Multistage Launch Vehicle Along Three-Dimensional Trajectory” *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 394–403. doi:10.2514/3.25350

[8] Roberts, S. M., and Shipman, J. S., *Two Point Boundary Value Problems: Shooting Methods*, Elsevier, New York, 1972.

[9] Kirk, D. E., “Numerical Determination of Optimal Trajectories,” *Optimal Control Theory: An Introduction*, Prentice–Hall, Upper Saddle River, NJ, 1975, pp. 329–413.

[10] Curtis, C. P., and Cloutier, J. R., “Control Designs for the Nonlinear Benchmark Problem via the State-Dependent Riccati Equation Method,” *International Journal of Robust and Nonlinear Control*, Vol. 8, Nos 4–5, 1998, pp. 401–433. doi:10.1002/(SICI)1099-1239(19980415/30)8:4/5<401::AID-RNC361>3.0.CO;2-U

[11] Menon, P. K., and Ohlmeyer, E. J., “Integrated Design of Agile Missile Guidance and Autopilot Systems,” *Control Engineering Practice*, Vol. 9, No. 10, Oct. 2001, pp. 1095–1106.

[12] Xin, M., and Balakrishnan, S. N., “A New Method for Suboptimal Control of a Class of Nonlinear Systems,” *Optimal Control Applications and Methods*, Vol 26, No. 2, 2005, pp. 55–83. doi:10.1002/oca.750

[13] Padhi, R., Xin, M., and Balakrishnan, S. N., “Suboptimal control of a One-Dimentionl Nonlinear Heat Equation Using POD and θ-*D* Techniques,” *Optimal Control Applications and Methods*, Vol. 29, No. 3, 2008, pp. 191–224. doi:10.1002/oca.822

[14] Gong, Q., Fahroo, F., and Ross, I. M., “Spectral Algorithm for Pseudospectral Methods in Optimal Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471. doi:10.2514/1.32908

[15] Fahroo, F., and Ross, I. M., “Direct Trajectory Optimization via a Chebyshev Pseudospectral Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166. doi:10.2514/2.4862

[16] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., “Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, Nov.–Dec. 2006, pp. 1435–1440. doi:10.2514/1.20478

[17] Rossiter, J. A., *Model Based Predictive Control: A Practical Approach*, CRC Press, New York, 2003, pp. 1–84.

[18] Yang, H., and Li, S., “Subspace-Based Adaptive Predictive Control for a Class of Nonlinear Systems,” *International Journal of Innovative Computing, Information and Control*, Vol. 1, No. 4, 2005, pp. 743–753.

[19] Soloway, D., and Haley, P. J., “Neural Generalized Predictive Control,” *Proceedings of the IEEE International Symposium on Intelligent Control*, IEEE Publ., Piscataway, NJ, 1996, pp. 277–282.

[20] Cannon, M., “Efficient Nonlinear Model Predictive Control Algorithms,” *Annual Reviews in Control*, Vol. 28, No. 2, 2004, pp. 229–237. doi:10.1016/j.arcontrol.2004.05.001

[21] Siegers, N., and Costello, M., “Model Predictive Control of a Parafoil and Payload System,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 816–821. doi:10.2514/1.12251

[22] Werbos, P. J., “Approximate Dynamic Programming for Real Time Control and Neural Modelling,” *Handbook of Intelligent Control*, edited by D. A. White and D. A. Sofge, Multiscience Press, New York, 1992, pp. 493–526.

[23] Padhi, R., “Model Predictive Static Programming: A Promising Technique for Optimal Missile Guidance,” *Annals of the Indian National Academy of Engineering*, Vol. 5, April 2008, pp. 185–194.

[24] Padhi, R., and Kothari, M., “Model Predictive Static Programming: A Computationally Efficient Technique for Suboptimal Control Design,” *International Journal of Innovative Computing, Information and Control*, Vol. 5, No 2, Feb. 2009, pp. 399–411.

[25] Dwivedi, P. N., Bhattacharya, A., and Padhi, R., “Computationally Efficient Suboptimal Midcourse Guidance Using Model Predictive Static Programming,” *Proceedings of IFAC World Congress*, Seoul, ROK, July 2008.

[26] Dwivedi, P. N., Padhi, R., Bhattacharya, A., and Bhattacharjee, R. N., “Suboptimal Midcourse Model Predictive Spread Acceleration Guidance,” *Proceedings of the International Conference and Exhibition in Aerospace Engineering*, Bangalore, India, May 2009.

[27] Tatiya, M., Bhatia, N., and Padhi, R., “Impact Angle Constrained Suboptimal Terminal Guidance of Interceptors for Air Defence,” *Proceedings of 18th IFAC Symposium on Automatic Control in Aerospace*, Nara, Japan, Sept. 2010.

[28] Bhattacharyaa, A., Dwivedi, P. N., Kumar, P., Bhale, P. G., and Bhattacharjee, R. N., “A Practical Approach for Robust Scheduling of Nonlinear Time-Scale Separated Autopilot,” *Proceedings of the International Conference on Advances in Control and Optimization of Dynamical Systems* [CD-ROM], Indian Inst. of Science, Bangalore, India, Feb. 2007.

[29] Ohlmeyer, E. J., and Phillips, C. A., “Generalized Vector Explicit Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, March–April 2006, pp. 261–268. doi:10.2514/1.14956

[30] Lukacs, J. A., and Yakimenko, O. A., “Trajectory-Shape-Varying Missile Guidance for Interception of Ballistic Missiles During the Boost Phase,” AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, SC, AIAA Paper 2007-6538, Aug. 2007.

[31] Atkinson, K. E., *An Introduction to Numerical Analysis*, Wiley, New York, 2001.

[32] Elbert, T. F., *Optimal Theory: Estimation and Control System*, Van Nostrand Reinhold, New York, 1984.