This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

Pseudo-Spectral MPSP based Unified Midcourse and Terminal Guidance for Re-Entry Targets

Prayag Sharma, Prem Kumar, Radhakant Padhi, *SM, IEEE*

***Abstract*—A unified optimal guidance scheme is presented in this paper for both midcourse and terminal phases combined. It minimizes the total deceleration resulting in enhanced range and/or higher impact velocity, while satisfying the terminal constraints on impact angle and miss distance. State-dependent terms are included in the cost function of the recently-proposed Generalized QS-MPSP and, because of its similarity with the Pseudo-Spectral philosophy, it is renamed as 'Pseudo-Spectral MPSP'. Extensive simulation studies with different engagement scenarios illustrate the effectiveness of the proposed guidance to engage with incoming high-speed ballistic targets.**

***Index Terms*—Optimal guidance, Midcourse guidance, Terminal Guidance, Angle-constrained guidance, Minimum acceleration guidance, MPSP guidance**

I. Introduction

Guidance of interceptors to intercept high-speed incoming ballistic targets has received increased attention over recent years. Since such targets are not easy to nullify, stringent terminal constraints are necessary that include high impact velocity, minimal miss-distance, and favorable impact angles. However, before an interceptor enters the terminal phase, it travels a significant portion of its trajectory in the midcourse. A midcourse guidance scheme is usually designed by carefully selecting a stationary Predicted Intercept Point (PIP) in 3D space [1] and guiding the vehicle towards it, while the actual non-stationary target is considered in the terminal phase [2].

Various guidance philosophies for both midcourse and terminal phases have been proposed in the literature. Midcourse guidance schemes based on the optimal control theory have been proposed in the literature [3]–[5]. However, many of these are based on the linearized kinematics near the collision triangle as well as two-dimensional engagement, which is far away from reality. 3D impact angle guidance laws based on linearized dynamics have also been proposed in the recent literature [6], [7]. Defining a general interception problem in the 3D space and considering the nonlinear dynamic model of the interceptor, however, leads to a complicated nonlinear optimal control problem. These types of problems can be solved using classical numerical techniques such as the gradient method [8], shooting method [9], transcription method [9], and so on. Nevertheless, classical numerical methods are computationally intensive and are predominantly used for offline applications only.

However, *computational guidance and control* [10] is getting a lot of attention recently. Essentially, the philosophy is to solve the trajectory optimization problem in real-time using 'fast optimal control' computational techniques using space-grade fast onboard processors [11]. This approach facilitates powerful guidance laws which are otherwise not possible. One such promising approach is the computationally efficient *Model Predictive Static Programming (MPSP)* technique [12], [13]; which solves a class of nonlinear fixed final time-optimal control problems by minimizing the control effort and ensures that the output vector satisfies a set of hard constraints at the final time t_f. Similarly, MPSP-based guidance schemes have also been applied to solve terminal phase guidance in a computationally efficient manner [14], [2].

A real-time optimal midcourse guidance using MPSP with alignment angle constraint has been proposed by two co-authors of this paper in [15]. The conventional wisdom of implementing two different guidance laws in two distinct phases has its own advantage. However, it also requires information about the switch-over time apriori, which has an implication of selecting (i.e. time-to-go for the midcourse phase) as well as the desired point in space at the end of the midcourse segment. Hence, any non-optimal selection of these parameters leads to a loss of optimality of the midcourse guidance, leading to additional guidance demand in the terminal phase. However, to the best of the authors' knowledge, these parameters are usually supplied to the guidance logic as *'tuning parameters'* and are not selected judicially. With such a heuristic choice of tuning parameters, in reality, a typical midcourse guidance law does not operate in the best possible optimal manner. To overcome this problem, a unified guidance scheme is proposed in this paper, which can be seamlessly applied to both midcourse as well as terminal guidance phases. Such a guidance approach is presented in this paper using the PS-MPSP approach to engage with incoming high-speed ballistic targets.

The key objective for an endo-atmospheric interceptor is to dissipate minimum energy throughout its trajectory so as to attain a higher speed at the time of interception. This is because a higher speed not only enhances the maneuverability and range of the interceptor but also makes it much more lethal to the target. A higher speed objective can be achieved by minimizing the total retardation experienced by the interceptor due to aerodynamic and propulsion forces acting along its entire trajectory. Thus, minimizing the total drag (i.e. both induced drag and profile drag) from missile launch to the point of interception while accounting for the terminal constraints on impact angle and miss distance forms a valid

Prayag Sharma is a Project Associate at the Dept. of Aerospace Engineering, Indian Institute of Science, Bangalore, India.

Prem Kumar is a senior scientist in the Programme Air-Defense, DRDO, Hyderabad, India.

Radhakant Padhi, SM-IEEE, is a Professor at the Dept. of Aerospace Engineering, Indian Institute of Science, Bangalore, India.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

unified guidance problem. Moreover, accounting for terminal constraints starting from the midcourse phase itself has other advantages as well, such as more time available to account for complex engagement scenarios and increased capability in accounting for handover error as the missile transitions from the midcourse to the terminal phase. However, one can notice that the MPSP-based guidance techniques proposed for midcourse [12], and terminal [16] phases only minimize the control effort, thereby minimizing the *induced drag* only. Keeping this motivation in mind, the co-authors of this paper extended the MPSP method lately to propose a midcourse guidance technique, where the total drag i.e., *induced drag* and *profile drag* both were minimized [1]. However, it should be emphasized here that the guidance technique proposed in [1] has been developed with some restrictive assumptions, because of which it cannot be applied as a *unified guidance scheme* for both the midcourse phase as well as the terminal phase. These restrictive assumptions are explained as follows: [1] proposed a midcourse guidance law based on PIP, whereas in the current guidance design 3D nonlinear point mass dynamics of both interceptor and target are considered to bring the formulation closer to the actual interception scenario. Since the guidance is supposed to start from the missile launch itself, changing mass over time due to propulsion, (also called the boost or thrust phase) should also be accounted for, which is considered in the problem formulation. Rather than solving a fixed final time problem [1], that can lead to sub-optimality in the solution, the guidance problem is formulated with respect to a new independent variable i.e., relative height between the target and the missile, to solve a free final time-optimal control problem.

The authors in [1] proposed a discrete-time solution to the minimum drag optimal control problem for the midcourse phase, due to which the number of optimization variables is dependent on the total time of the trajectory and the selected time step. Since the objective here is to generate a fairly long trajectory (i.e., for both midcourse and terminal), the total flight time increases, and moreover, the guidance update cycle in the terminal phase is quite small as compared to the midcourse phase, which may decrease the computational efficiency of the algorithm. Addressing this issue is one of the main objectives of this paper.

By combining the ideas of Generalised-MPSP (G-MPSP) [17], Quasi-Spectral MPSP (QS-MPSP) [14], and pseudo-spectral methods [18], Zhou et al. recently proposed a method of solving the minimum control effort guidance problem using MPSP in continuous-time, and named it as Generalised QS-MPSP (G-QS-MPSP) [19]. Like QS-MPSP, this formulation retains the continuous nature of the problem. However, like the pseudo-spectral technique, it only requires the computation to be performed at a few collocation points rather than at all the discrete nodes, thus making it computationally more efficient and independent of the total duration of the engagement as well. In order to solve the Minimum drag problem discussed in this paper, however, this technique has been extended further to include a cost-dependent term, which requires substantially additional algebra. Also, because of its similarity with the Pseudo-Spectral philosophy, it is renamed as ‘Pseudo-Spectral MPSP’ (PS-MPSP).

The performance of the unified midcourse-cum-terminal PS-MPSP guidance has been tested for different engagement scenarios and by performing closed-form Monte-Carlo simulations with realistic conditions such as plant parameter uncertainty (such as uncertainties in the normal and axial force coefficients of the interceptor; and the ballistic coefficient of the target) as well as handover errors (i.e. heading error) at the beginning of the terminal guidance phase. The Extensive simulation study shows that the proposed unified MPSP guidance works well and achieves the objective of minimizing the total deceleration and attaining a much higher speed at the time of interception while satisfying the terminal constraints with near-zero error.

II. Optimal Guidance: Problem Formulation

As mentioned in Section I, a unified optimal guidance scheme is presented in this paper for the interception of high-speed ballistic targets. Details about the problem formulation are presented in this section:

A. State dynamics

Both the interceptor as well as the target are modeled in this paper as point mass objects in 3D space. The models assume flat and non-rotating Earth and the associated inertial frame is assumed to be the Vertical-East-North (VEN) frame; where x-axis is along the local vertical up direction, y-axis is along the local east direction and z-axis is along the local north direction, while the origin of this frame is at the interceptor launch location.

1) Interceptor model: With the assumptions mentioned in Section II-A, the interceptor model can be written as

$$\begin{bmatrix} \dot{x}_M \\ \dot{y}_M \\ \dot{z}_M \\ \dot{V}_M \\ \dot{\gamma}_{a_M} \\ \dot{\gamma}_{e_M} \end{bmatrix} = \begin{bmatrix} V_M \sin\gamma_{e_M} \\ V_M \cos\gamma_{e_M} \sin\gamma_{a_M} \\ V_M \cos\gamma_{e_M} \cos\gamma_{a_M} \\ a_x - g_M \sin\gamma_{e_M} \\ \frac{a_y}{V_M \cos\gamma_{e_M}} \\ -\left(\frac{a_z + g_M \cos\gamma_{e_M}}{V_M}\right) \end{bmatrix} \tag{1}$$

where, (x_M, y_M, z_M) is the interceptor position in the VEN frame. V_M is the velocity magnitude of the interceptor and γ_{a_M}, γ_{e_M} are its flight path angles in azimuth and elevation planes respectively. g_M is the acceleration due to gravity at the interceptor's current location. (a_x, a_y, a_z) are the acceleration components of the interceptor in the velocity frame of the interceptor.

2) Target model: The target is assumed to be a non-maneuvering incoming ballistic object in 3D space, which can be captured fairly well by the following state space model

$$\begin{bmatrix} \dot{x}_T \\ \dot{y}_T \\ \dot{z}_T \\ \dot{V}_T \\ \dot{\gamma}_{a_T} \\ \dot{\gamma}_{e_T} \end{bmatrix} = \begin{bmatrix} V_T \sin\gamma_{e_T} \\ V_T \cos\gamma_{e_T} \sin\gamma_{a_T} \\ V_T \cos\gamma_{e_T} \cos\gamma_{a_T} \\ -\frac{\rho_T V_T^2}{2\beta_T} - g_T \sin\gamma_{e_T} \\ 0 \\ -\left(\frac{g_T \cos\gamma_{e_T}}{V_T}\right) \end{bmatrix} \tag{2}$$

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

where most of the variables used are analogous to the interceptor model with the subscript 'T' replacing the subscript 'M'. β_T is the ballistic coefficient of the target, which is typically estimated from a target state estimator and obviates the necessity of knowing the mass, drag coefficient, and aerodynamic surface of the target individually. β_T is a property of the target and cannot be known apriori, but can be estimated fairly accurately based on radar measurements [20]. Regardless, the guidance algorithm is tested by considering significant uncertainty in β_T, for details refer Section IV-2.

B. Guidance constraints

The key objective of the interceptor guidance is to successfully engage with the target with zero miss-distance. This objective can be achieved by enforcing the following boundary conditions.

$$\begin{bmatrix} x_M - x_T \\ y_M - x_T \\ z_M - x_T \end{bmatrix}_{t_f} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

Also it is desirable to intercept the target with specified relative flight path angles to enhance the effectiveness of the interception; this additional constraint is given below

$$\begin{bmatrix} \gamma_{a_M} - \gamma_{a_T} \\ \gamma_{e_M} - \gamma_{e_T} \end{bmatrix}_{t_f} = \begin{bmatrix} \Delta\gamma_a^* \\ \Delta\gamma_e^* \end{bmatrix} \tag{4}$$

It is assumed that $\begin{bmatrix} \gamma_{a_T} & \gamma_{e_T} \end{bmatrix}_{t_f}^T$ can be estimated accurately. Hence equation (4) can be equivalently written as

$$\begin{bmatrix} \gamma_{a_M} \\ \gamma_{e_M} \end{bmatrix}_{t_f} = \begin{bmatrix} \gamma_{a_M}^* \\ \gamma_{e_M}^* \end{bmatrix} \tag{5}$$

where, $\gamma_{a_M}^* \triangleq \gamma_{a_T}(t_f) + \Delta\gamma_a^*$, $\gamma_{e_M}^* \triangleq \gamma_{e_T}(t_f) + \Delta\gamma_e^*$ and these are the desired flight path angles of the interceptor in the azimuth and elevation planes at the final time.

C. Cost function

The cost function that minimizes the total retardation experienced by the interceptor due to aerodynamic and propulsion forces acting along its trajectory, can be written in the following form

$$J = \int_{t_0}^{t_f} (F_D/m)\, dt \tag{6}$$

where, F_D is the total deceleration force on the interceptor due to aerodynamic and thrust forces (as F_D force is having the same direction as drag, hence D subscript is used in the notation), m is the mass of the interceptor. t_0 is the initial time and t_f is the final time of interception.

1) Simplified expression of the deceleration force expression for a point mass interceptor model: A simplified expression for the deceleration force (F_D) for a point mass interceptor model has been obtained in this section. Figure 1, shows the aerodynamic and thrust forces acting on a point mass interceptor model. Where F_L represents the total aerodynamic and thrust force acting perpendicular to the interceptor velocity direction. While, F_D represents the deceleration force, and it is defined as the total aerodynamic and thrust force acting opposite to the interceptor velocity direction. The aerodynamic force components N and A are called the normal force and axial force respectively, T is the axial thrust, α_T is called the total angle of attack, and (V) is defined as the velocity vector of the interceptor. Using Fig. 1 and writing the force equation opposite to the interceptor velocity direction gives the total deceleration force expression as below.

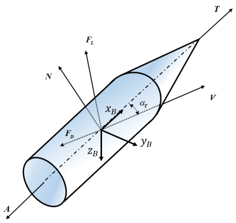

Fig. 1. Aerodynamic and thrust forces acting on the interceptor

$$F_D = (N \sin\alpha_T + A\cos\alpha_T) - T\cos\alpha_T \tag{7}$$

Remark: In general, three types of forces act on the interceptor at any point in time; thrust force, aerodynamic force, and gravity force. Out of these three, the gravity force is a conservative force, and hence work done by it is independent of the path, and it depends only on the initial and final positions. However, the thrust and aerodynamic forces are non-conservative forces, and the work done by these is dependent on the trajectory. Because of this reason, it is reasonable to construct F_D as the components of thrust and aerodynamic forces only.

By writing the axial force and normal forces in terms of their aerodynamic force coefficients, the total deceleration force expression in (7) gets modified as below.

$$F_D = QS\left(C_N \sin\alpha_T + C_A\cos\alpha_T\right) - T\cos\alpha_T \tag{8}$$

where C_A is axial force coefficient, C_N is normal force coefficient, Q is the dynamic pressure, and S is the reference area. Assuming the normal force coefficient to be linear with respect to the angle of attack (that is $C_N = C_{N_\alpha}\alpha$, because interceptors are in general symmetric bodies and hence at zero angle of attack the normal force is also zero). Then for a small angle of attack (such that these approximations are valid, $\sin\alpha_T \approx \alpha_T$ and $\cos\alpha_T \approx 1 - \frac{\alpha_T^2}{2}$) the total deceleration force in (8) can be written as

$$\begin{aligned} F_D &\approx QSC_{N_\alpha}\alpha_T^2 - (T - QSC_A)\left(1 - \frac{\alpha_T^2}{2}\right) \\ &\approx QSC_A - T + QS\left(2C_{N_\alpha} + \frac{T}{QS} - C_A\right)\frac{\alpha_T^2}{2} \end{aligned} \tag{9}$$

where C_{N_α} is called the normal force coefficient slope, and it is the slope of the normal force coefficient (C_N) with respect to the angle of attack.

Also, by writing the force equation perpendicular to the direction of the velocity vector it gives

$$F_L = (N\cos\alpha_T - A\sin\alpha_T) + T\sin\alpha_T \tag{10}$$

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

Equation (10) can be simplified further by writing the forces in terms of their aerodynamic force coefficients and under the same assumption of linear aerodynamic normal force coefficients and under a small angle of attack, the lift force coefficient can be written as

$$\begin{aligned} F_L &\approx QS\left(C_{N_\alpha}\alpha_T - C_A\alpha_T\right) + T\alpha_T \\ &= QS\left(C_{N_\alpha} - C_A + \frac{T}{QS}\right)\alpha_T \end{aligned} \tag{11}$$

Equation (11) gives the method of computation of the force perpendicular to interceptor velocity, and it can be observed that for small angle of attack; force F_L is linear with respect to the total angle of attack (α_T). By defining a new symbol $C_{F_{L_\alpha}}$ as below, it makes the equations (11) more compact.

$$C_{F_{L_\alpha}} = C_{N_\alpha} + \left(\frac{T}{QS} - C_A\right) \tag{12}$$

The total deceleration force expression as given in (9), requires an angle of attack value; because all other terms in the right-hand side of (9) are known by aerodynamic and atmospheric models except the total angle of attack. The total angle of attack can be obtained from the relationship between the demanded acceleration from guidance and the total force generated perpendicular to interceptor velocity as below.

$$F_L = m\sqrt{a_y^2 + a_z^2} \tag{13}$$

where m is the interceptor mass, $(a_y,\ a_z)$ is the guidance demand. Using (11, 12, 13) the total angle of attack can be computed as below

$$\alpha_T = \frac{m\sqrt{a_y^2 + a_z^2}}{QSC_{F_{L_\alpha}}} \tag{14}$$

Putting the total angle of attack expression in (9) gives the explicit expression for the deceleration force as below.

$$F_D = QSC_A - T + \left(1 + \frac{C_{N_\alpha}}{C_{F_{L_\alpha}}}\right)\frac{m^2\left(a_y^2 + a_z^2\right)}{2QSC_{F_{L_\alpha}}} \tag{15}$$

2) Simplified cost function expression: Substitution of the deceleration force expression given in (15) to the cost function expression given in (6), gives the simplified expression of the total cost as below.

$$\begin{aligned} J = -\int_{t_0}^{t_f}\left(\frac{T}{m}\right)dt + \int_{t_0}^{t_f}\left(\frac{QSC_A}{m}\right)dt + \frac{1}{2}\int_{t_0}^{t_f} U^T \times \\ \left(\frac{1}{m}\left(1 + \frac{C_{N_\alpha}}{C_{F_{L_\alpha}}}\right)\left(\frac{m^2}{QSC_{F_{L_\alpha}}}\right)I_{2\times 2}\right)U dt \end{aligned} \tag{16}$$

In general, the interceptor is propelled by a solid propellant motor; where thrust and mass-flow-rate usually depend only on flight time, moreover, the total impulse of these motors is also constant. Consequently, thrust and mass ratio (i.e. $\frac{T}{m}$) is also an explicit function of only flight time. Moreover, in most realistic scenarios, the thrust of the interceptor extinguishes during the initial phase of the midcourse guidance. Under the aforementioned conditions, the integral $(\int_{t_0}^{t_f}(T/m)dt)$ will always turn out to be a fixed quantity for each interceptor trajectory, and hence it can be removed from the cost function (16) during the optimization process. Hence, the final expression of the cost function can be written as

$$J = \int_{t_0}^{t_f} G dt + \frac{1}{2}\int_{t_0}^{t_f} U^T R U dt \tag{17}$$

where, $G = \left(\frac{QSC_A}{m}\right)$, $Q = \frac{1}{2}\rho V^2$, $U = \left[a_y\ a_z\right]^T$ and

$$R = \left(1 + \frac{C_{N_\alpha}}{C_{F_{L_\alpha}}}\right)\left(\frac{m}{QSC_{F_{L_\alpha}}}\right)I_{2\times 2} \tag{18}$$

This specific form of the cost function (17) has been obtained by assuming the dynamic pressure Q as a time-varying parameter in R, and it can be observed that this assumption removes coupling between state and control variables from the cost function expression. The cost function (17) is composed of two terms, where the first term is a function of state only and the second term is a quadratic expression in terms of control input. This specific form of the cost function expression has been selected because it facilitates a simpler algebra during the computation of control inputs.

D. Problem formulation w.r.t Relative height

The proposed guidance problem is a free final time optimal control problem, however, for the present guidance problem formulation, the final time of flight is not known. The computation of the final time can be avoided by selecting a new independent variable. The current engagement scenario consists of an incoming high-speed target during its re-entry phase that follows a ballistic trajectory as shown in Figure. 4 and an endo-atmospheric interceptor that intercepts the target during this phase. It can be observed that $x_{TM} \triangleq x_T - x_M$ (that is, the relative altitude of the target with respect to the interceptor) is a monotonically varying variable. This happens because of the very high speed of the target in its reentry phase in comparison to the interceptor (that is, $|V_T \sin\gamma_{e_T}| > |V_M \sin\gamma_{e_M}|$), under this restriction (which is valid for most of the re-entry targets) the variable x_{TM} varies monotonically with known initial (can be calculated from given initial conditions) and final (at the time of interception this will be zero) values. Moreover, to maximize the chances of a successful interception a head-on/forward attack is desirable which can be achieved by restricting the terminal impact angle accordingly. The desired impact angle can be selected such that the angle $\angle HTI$, (where the vector $\overrightarrow{HT}$ is parallel to the horizon, T defines the point mass target, I denotes the point mass interceptor, and the plane containing $\overrightarrow{HT}$ is perpendicular to the plane containing $\overrightarrow{TI}$) shown in figure 5 is always positive and less than equal to 180°. Under such practical restrictions, the relative altitude of the target with respect to the interceptor would always vary monotonically till the point of interception.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

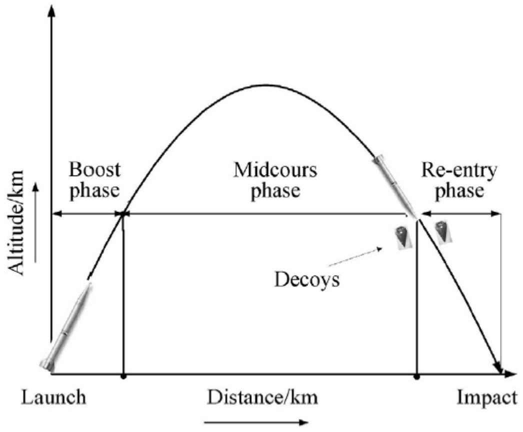

Fig. 2. Ballistic trajectory of the Target during various phases [21]

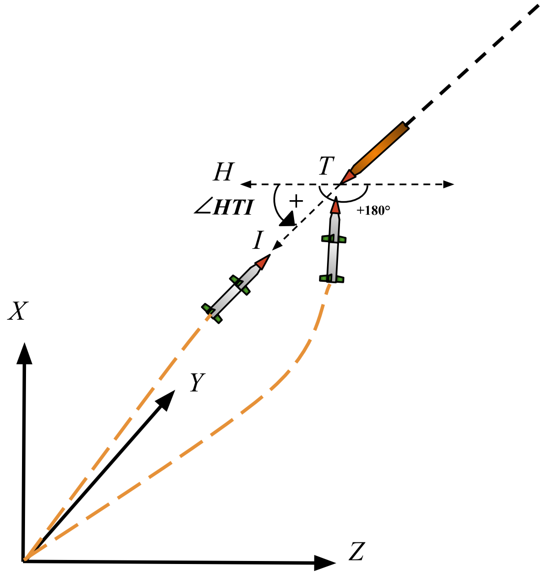

Fig. 3. Desired target and missile interception scenario

1) State dynamics model: Changing the independent variable in the state dynamics from t to x_{TM} can be performed below

$$\begin{aligned}\frac{dt}{dx_{TM}} &= 1/(\dot{x}_T - \dot{x}_M)\\ &= 1/(V_T \sin\gamma_{e_T} - V_M \sin\gamma_{e_M}) \end{aligned}\tag{19}$$

$$\begin{aligned}\frac{dx_M}{dx_{TM}} &= \left(\frac{dx_M}{dt}\right)\Big/\left(\frac{dx_{TM}}{dt}\right)\\ &= \frac{V_M \sin\gamma_{e_M}}{(V_T \sin\gamma_{e_T} - V_M \sin\gamma_{e_M})}\end{aligned}\tag{20}$$

Equation (20) converts interceptor altitude dynamics with respect to x_{TM}. In a similar manner other state dynamic equations have been transformed with respect to x_{TM}. And the final state dynamics can be written as

$$\begin{aligned}\frac{dX}{dx_{TM}} = \Bigg[& \frac{dt}{dx_{TM}} \quad \frac{dx_M}{dx_{TM}} \quad \cdots \\ & \frac{d\gamma_{e_M}}{dx_{TM}} \quad \frac{dx_T}{dx_{TM}} \quad \cdots \quad \frac{d\gamma_{e_T}}{dx_{TM}} \Bigg]^T\end{aligned}\tag{21}$$

Here, time t has also been included as a state because the plant is time-varying and it requires information of time at each time step.

2) Output equation: Clearly, the guidance constraints given in (3, 5) are at the final time. But, the state dynamics (21) is propagated in terms of x_{TM}, till $X_{TM}(t_f) = 0$; hence at the final time $x_M(t_f) = x_T(t_f)$. This leaves only four extra guidance constraints to be satisfied. Defining the output vector as below gives all the parameters of interest for the given guidance problem.

$$\begin{aligned}Y = \big[& y_M - y_T \quad z_M - z_T \\ & \gamma_{a_M} - \gamma_{a_M}^* \quad \gamma_{e_M} - \gamma_{e_M}^* \big]^T\end{aligned}\tag{22}$$

The guidance constraints given in (3, 5) can be re-written as below

$$Y_{t_f}^* = 0 \tag{23}$$

where, $Y_{t_f}^*$ represents the desired output value at the final time.

3) Cost function: Substituting the expression (19), the cost function given in (17) can be written as below

$$J = \int_{x_{TM_0}}^{x_{TM_f}} G\, dx_{TM} + \frac{1}{2}\int_{x_{TM_0}}^{x_{TM_f}} U^T R U dx_{TM} \tag{24}$$

where, $G = \left(\frac{QSC_A}{(\dot{x}_T - \dot{x}_M)m}\right)$, $Q = \frac{1}{2}\rho V^2$, $U = [a_y \ a_z]^T$ and

$$R = \left(\frac{m\left(I_{2\times2}\right)}{\left(QSC_{F_{L_\alpha}}\right)(\dot{x}_T - \dot{x}_M)}\right)\left(1 + \frac{C_{N_\alpha}}{C_{F_{L_\alpha}}}\right) \tag{25}$$

To solve the proposed guidance problem with cost function (17) having additional nonlinear functions of the state, a new methodology named 'Pseudo-Spectral MPSP' is proposed in this paper. This method is an important extension of the existing G-QS-MPSP guidance [19], and it can solve a larger class of guidance problems. The details of the 'Pseudo-Spectral MPSP' are given in Sec. III.

III. Pseudo-Spectral MPSP

The problem formulation of the proposed guidance uses a nonlinear state dynamics and output equation as below

$$\dot{X}(t) = f(X(t), U(t)) \tag{26}$$

$$Y(t) = h(X(t)) \tag{27}$$

where, $X \in \Re^n$, $U \in \Re^r$, $Y \in \Re^m$ are state, control and output variables. The primary objective is to come up with a suitable control history $U(t)$, so that the output at the final time $Y(t_f)$ goes to the desired value $Y_{t_f}^*$, i.e. $Y_{t_f} \rightarrow Y_{t_f}^*$.

The Pseudo-Spectral MPSP technique presented here is an iterative algorithm and starts with a guess history of the control solution. The guess control history is often generated using classical or heuristic approaches, and it is highly unlikely that this guess control would satisfy the guidance objectives. To improve the control history an optimal control deviation history based on the output error information at the final time step has been calculated. This deviation has been added to the present control history to get an improved/updated control

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

history. The control history is improved in an iterative manner until the algorithm converges.

Because the state, output, and control histories are improved in an iterative manner, the relationship of the state, control, and output variables between two consecutive iterations i and $(i+1)$ at any given time t is defined as

$$\begin{aligned} Y^{i+1}(t) &\triangleq Y^i(t) + \delta Y^i(t) \\ X^{i+1}(t) &\triangleq X^i(t) + \delta X^i(t) \\ U^{i+1}(t) &\triangleq U^i(t) + \delta U^i(t) \end{aligned} \tag{28}$$

1) Guidance constraint: The objective of the algorithm at i^{th} iteration is to determine a suitable control history $U^{i+1}(t)$, so that the output at the final time in the next iteration (that is, $Y^{i+1}(t_f)$) should go to the desired final output $Y^*_{t_f}$, (that is $Y^{i+1}_{t_f} \to Y^*_{t_f}$). Hence the control history must be updated in such a way that, it should ensure the following equality

$$Y^{i+1}(t_f) = Y^*_{t_f} \tag{29}$$

Equation (29) can be simplified further as below

$$\begin{aligned} Y^i(t_f) + \delta Y^i(t_f) &= Y^*_{t_f} \\ \delta Y^i(t_f) + Y^i(t_f) - Y^*_{t_f} &= 0 \qquad (30) \\ \delta Y^i(t_f) + \Delta Y^*_{t_f} &= 0 \qquad (31) \end{aligned}$$

where, $\Delta Y^*_{t_f} \triangleq Y^i(t_f) - Y^*_{t_f}$. Equation (31) is the equality constraint imposed at the final time step by the guidance requirements.

2) Cost function: Apart from meeting the output constraint on the final time step, an additional objective of the guidance is to ensure minimization of the cost function J along the trajectory as given below

$$J = \int_{t_0}^{t_f} \left[G\left(X_t^{i+1}\right) + \frac{1}{2}\left(U_t^{i+1}\right)^T R(t) \left(U_t^{i+1}\right) \right] dt \tag{32}$$

where, $U_t^{i+1} \triangleq U^{i+1}(t)$ and $X_t^{i+1} \triangleq X^{i+1}(t)$. The cost function J given in (32), clearly is a function of updated state and updated control. To improve the control solution iteratively, first, the state and output deviations with respect to control changes have been obtained, then the continuous control history is written as a weighted sum of basis functions. After this, the necessary conditions of optimality for the cost function with the additional guidance constraint on the final output are derived to generate a linear system of algebraic equations in terms of coefficients of the basis functions. By solving this linear system of equations one can get the optimal control history update. The steps, as mentioned above, are elaborated in the following subsections.

A. State and final output error due to control deviations

The expressions for state and final output error have been derived in this section. The error or deviation in state at any given time t can be attributed to all the control deviations prior to that time, thus an intermediate variable τ is defined that varies from $t_0 \to t$ to derive the above expression as follows. First multiplying the state dynamic equation with a weighting matrix $W(\tau)$ on both sides:

$$W(\tau)\dot{X}(\tau) = W(\tau) f(X(\tau), U(\tau)) \tag{33}$$

where, the weighting matrix $W(\tau) \in \Re^{n \times n}$, Integrating the above equation both sides from t_0 to t

$$\int_{t_0}^{t} \left[W(\tau)\dot{X}(\tau) \right] d\tau = \int_{t_0}^{t} \left[W(\tau) f(X(\tau), U(\tau)) \right] d\tau \tag{34}$$

Integrating the left-hand side expression,

$$\left[W(\tau) X(\tau)|_{\tau=t} - W(t_0) X(t_0) \right] - \int_{t_0}^{t} \left[\dot{W}(\tau) X(\tau) \right] d\tau = \int_{t_0}^{t} \left[W(\tau) f(X(\tau), U(\tau)) \right] d\tau \tag{35}$$

adding $X(t)$ to both sides,

$$\begin{aligned} X(t) = X(t) - \left[W(\tau) X(\tau) \right]_{\tau=t} + W(t_0) X(t_0) \\ + \int_{t_0}^{t} \left[W(\tau) f(X(\tau), U(\tau)) + \dot{W}(\tau) X(\tau) \right] d\tau \end{aligned} \tag{36}$$

and taking first variation on both sides then simplifying

$$\begin{aligned} \delta X(t) = \left[(I - W(\tau)) \delta X(\tau) \right]_{\tau=t} + W(t_0) \delta X(t_0) + \\ \int_{t_0}^{t} \left[W\left(\tfrac{\partial f}{\partial X} \delta X(\tau) + \tfrac{\partial f}{\partial U} \delta U(\tau) \right) + \dot{W} \delta X(\tau) \right] d\tau \end{aligned} \tag{37}$$

Since, state at time t_0 is known, $\delta X(t_0) = 0$. The weighting matrix $W(t)$ needs to be selected in such a way that $\delta X(t)$ is only a function of deviation in control, keeping this in mind the following matrix dynamic equation with the associated boundary condition and expression of $\delta X(t)$ can be derived as:

$$\dot{W}(\tau) = -W(\tau) \tfrac{\partial f}{\partial X} \tag{38}$$

$$W(t) = I_{n \times n} \tag{39}$$

$$\delta X(t) = \int_{t_0}^{t} \left[B_S^t(\tau) \delta U(\tau) \right] d\tau \tag{40}$$

$$B_S^t(\tau) \triangleq W(\tau) \left[\tfrac{\partial f}{\partial U} \right]_{(X(\tau), U(\tau))} \tag{41}$$

Using (40) the weighting matrix can be computed from t to t_0 by integrating the equation (38) backward. $B_S^t(\tau)$ relates the variation in the state to the variation in control prior to that time. The superscript t in $B_S^t(\tau)$ denotes the time at which the boundary condition (40) is applied to solve the matrix dynamic equation (38). Similarly, the expression for variation in output at the final time t_f can be obtained using (40). Considering output equation (27), and taking the first variation on both sides at final time t_f

$$\delta Y(t_f) = \left[\frac{\partial h}{\partial X} \right]_{X(t_f)} \delta X(t_f) \tag{42}$$

substituting the expression (40) at final time t_f,

$$\delta Y(t_f) = \left[\frac{\partial h}{\partial X} \right]_{X(t_f)} \int_{t_0}^{t_f} \left[B_S^{t_f}(\tau) \delta U(\tau) \right] d\tau \tag{43}$$

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

B. Spectral Transcription of Control Variable

By transcribing the control history in terms of the weighted sum of basis functions, the continuous nature of the control variable can be retained, while the optimization variables can become static, as shown below:

$$U^i\left(t\right)=\sum_{j=1}^{N_p}C_j^iP_j\left(t\right) \tag{44}$$

where, $C_j^i\in\Re^m$, N_p denotes the number of basis functions used, and $P_j\left(t\right)$ denotes the jth basis function. The coefficients $C_j^i,\ j=1,2,\ldots,N_p$ are the new optimization variables. The basis functions can be selected arbitrarily and can be of the form such as Legendre polynomials, Chebychev polynomials, etc. Using (44) the expression for variation in control update can be written as:

$$\delta U^i\left(t\right)=\sum_{j=1}^{N_p}dC_j^iP_j\left(t\right) \tag{45}$$

The above expression can be substituted back to equation (40) to get the updated expression for $\delta X\left(t\right)$

$$\delta X\left(t\right)=\int\limits_{t_0}^{t}\left[B_S^t\left(\tau\right)\sum_{j=1}^{N_p}dC_j^iP_j\left(\tau\right)\right]d\tau$$

$$\delta X\left(t\right)=\sum_{j=1}^{N_p}\left(\int\limits_{t_0}^{t}\left[B_S^t\left(\tau\right)P_j\left(\tau\right)d\tau\right]\right)dC_j^i$$

$$\delta X\left(t\right)=\sum_{j=1}^{N_p}Q_j^tdC_j^i,\quad\left(Q_j^t\triangleq\int\limits_{t_0}^{t}\left[B_S^t\left(\tau\right)P_j\left(\tau\right)d\tau\right]\right) \tag{46}$$

By substituting the above expression back to (42), expression for $\delta Y\left(t_f\right)$ can also be obtained

$$\delta Y\left(t_f\right)=\left[\frac{\partial h}{\partial X}\right]_{X\left(t_f\right)}\sum_{i=1}^{N_p}Q_j^{t_f}dC_j^i \tag{47}$$

Similar to (28), the relation between coefficients of successive iterations can be defined as

$$C_j^{i+1}=C_j^i+dC_j^i \tag{48}$$

Using the above relation, $\delta Y\left(t\right)$ can be re-written as

$$\delta Y\left(t_f\right)=\left[\tfrac{\partial h}{\partial X}\right]_{X\left(t_f\right)}\sum_{i=1}^{N_p}Q_j^{t_f}\left(C_j^{i+1}-C_j^i\right)$$

$$\delta Y\left(t_f\right)=\left[\tfrac{\partial h}{\partial X}\right]_{X\left(t_f\right)}\sum_{i=1}^{N_p}Q_j^{t_f}C_j^{i+1}-\left[\tfrac{\partial h}{\partial X}\right]_{t_f}\sum_{i=1}^{N_p}Q_j^{t_f}C_j^i$$

$$\delta Y\left(t_f\right)=\left[\tfrac{\partial h}{\partial X}\right]_{X\left(t_f\right)}\sum_{i=1}^{N_p}Q_j^{t_f}C_j^{i+1}-D_\lambda \tag{49}$$

where, $D_\lambda\triangleq\left[\tfrac{\partial h}{\partial X}\right]_{X\left(t_f\right)}\sum_{i=1}^{N_p}Q_j^{t_f}C_j^i$

C. Necessary conditions of optimality

Using the cost function given in (32) and equality constraint (31), the augmented cost function can be constructed as below

$$\tilde{J}^i=\int\limits_{t_0}^{t_f}G_t^{i+1}dt+\tfrac{1}{2}\int\limits_{t_0}^{t_f}\left(U_t^{i+1}\right)^TR\left(t\right)\left(U_t^{i+1}\right)dt+\\ \lambda^T\left(\delta Y^i\left(t_f\right)+\Delta Y_{t_f}^*\right) \tag{50}$$

where, $G_t^{i+1}\triangleq G\left(X^{i+1}\left(t\right)\right)$ and λ is the Lagrange multiplier. Also, $\left(G_t(\cdot)\geq 0,\ \forall X\in\Re^n\right)$ and R_t is the control weighing matrix.

From (46) it is clear that variation in state at any time t is a function of the deviation in the coefficients of basis functions, similarly variation in the output (47) at any time is also a function of deviation in the coefficients. Considering these and the expression of the augmented cost function $\tilde{J}^i$ as given in (50); it is clear that the cost function $\tilde{J}^i$ is a function of deviation in the coefficients of basis functions and Lagrange multiplier only. Hence, necessary conditions for optimality for $\tilde{J}^i$ can be written as

$$\frac{\partial\tilde{J}^i}{\partial\lambda}\quad=0 \tag{51}$$

$$\frac{\partial\tilde{J}^i}{\partial\left(dC_j^i\right)}\quad=0,\quad\forall j\leq N_p \tag{52}$$

Simplifying (51) and substituting the expression for $\delta Y^i\left(t_f\right)$ from (49), gives the following equation

$$\left[\frac{\partial h}{\partial X}\right]_{X\left(t_f\right)}\sum_{i=1}^{N_p}Q_j^{t_f}C_j^{i+1}-D_\lambda=-\Delta Y_{t_f}^* \tag{53}$$

Simplification of the expression corresponding to $\dfrac{\partial\tilde{J}^i}{\partial\left(dC_j^i\right)}$ is obtained as below

$$\frac{\partial\tilde{J}^i}{\partial\left(dC_j\right)}=\frac{1}{2}\int\limits_{t_0}^{t_f}\frac{\partial}{\partial\left(dC_j\right)}\left[\left(U_t^{i+1}\right)R_t\left(U_t^{i+1}\right)\right]dt+\\ \int\limits_{t_0}^{t_f}\frac{\partial G_t^{i+1}}{\partial\left(dC_j\right)}dt+\left[\frac{\partial\left(\delta Y\left(t_f\right)\right)}{\partial\left(dC_j\right)}\right]^T\lambda \tag{54}$$

From (54), it is clear that expression corresponding to $\frac{\partial\tilde{J}^i}{\partial\left(dC_j^i\right)}$ is the summation of three terms; next step is to obtain simplified expressions corresponding to each term.

The first term involves the evaluation of the partial derivative of G_t^{i+1} with respect to dC_j^i. But $G_t(.)$ is a function of the state only, hence by chain rule this partial derivative has been evaluated as

$$\frac{\partial G_t^{i+1}}{\partial\left(dC_j^i\right)}\quad=\quad\left[\frac{\partial X_t^{i+1}}{\partial\left(dC_j^i\right)}\right]^T\left[\frac{\partial G}{\partial X}\right]_{\left(X_t^{i+1}\right)} \tag{55}$$

By definition, $X_t^{i+1}\triangleq X_t^i+\delta X_t^i$ (from (28)). Substituting this in (55) gives

$$\frac{\partial G_t^{i+1}}{\partial\left(dC_j\right)}=\left[\frac{\partial X^i\left(t\right)}{\partial\left(dC_j\right)}+\frac{\partial\left(\delta X^i\left(t\right)\right)}{\partial\left(dC_j\right)}\right]^T\left[\frac{\partial G}{\partial X}\right]_{X_t^{i+1}} \tag{56}$$

As state history is independent of deviation in coefficients, this gives $\frac{\partial X_t^i}{\partial\left(dC_j^i\right)}=0,\ \forall i,\ \forall j\leq N_p$. Also, by substituting (46) in (56) and solving

$$\frac{\partial G_t^{i+1}}{\partial\left(dC_j\right)}=\left[Q_j^t\right]^T\left[\frac{\partial G}{\partial X}\right]_{X^{i+1}(t)} \tag{57}$$

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

Equation (57) requires the evaluation of the gradient of G at the state X_t^{i+1}. whereas, X_t^{i+1} is the updated state which in turn depends on the control update. To resolve this issue the gradient $\left[\frac{\partial G}{\partial X}\right]_{\left(X_t^{i+1}\right)}$ has been evaluated at the present state history X_t^i. Thus, the first term in (54) can be evaluated as below

$$\int_{t_0}^{t_f} \frac{\partial G_t^{i+1}}{\partial \left(dC_j\right)} dt = \int_{t_0}^{t_f} \left[\left[Q_j^t\right]^T \left[\frac{\partial G}{\partial X}\right]_{X^i(t)} \right] dt \qquad (58)$$

Substituting the expression of Q_j^t from (46),

$$\int_{t_0}^{t_f} \left[\left[\int_{t_0}^{t} \left[\left[B_{S_2}^t(\tau)\right]^T P_j(\tau) \right] d\tau \right] \left[\frac{\partial G}{\partial X}\right]_{X^i(t)} \right] dt = E_j \quad (59)$$

Let E_j denote the double integral (59) calculated for the jth basis function. To evaluate the second term in the expression (54), first using (45) and (48) expression for U_t^{i+1} is written in terms of basis functions and then the partial derivative $\frac{\partial}{\partial\left(dC_j^i\right)}$ can be evaluated as follows

$$\frac{1}{2}\int_{t_0}^{t_f} \frac{\partial}{\partial(dC_j)} \left[\left(\sum_{k=1}^{N_p} C_k^{i+1} P_k(t) \right)^T R_t \left(\sum_{k=1}^{N_p} C_k^{i+1} P_k(t) \right) \right] dt$$
$$= \sum_{k=1}^{N_p} R_{k,j} C_k^{i+1}, R_{j,k} \triangleq \int_{t_0}^{t_f} P_j(t)\, R(t)\, P_k(t)\, dt \qquad (60)$$

Similarly, the third term in (54) can be evaluated by substituting the expression for $\delta Y(t_f)$ from (47) and evaluating the partial derivative as below

$$\left[\frac{\partial\left(\delta Y(t_f)\right)}{\partial\left(dC_j^i\right)}\right]^T \lambda = \left[\left[\frac{\partial h}{\partial X}\right]_{t_f} Q_j^{t_f}\right]^T \lambda \qquad (61)$$

The optimality criterion given in (51, 52), can be written by using (53, 59, 60, 61) along the optimal trajectory as given below

$$\sum_{k=1}^{N_p} R_{k,j} C_k^{i+1} + \left[\left[\frac{\partial h}{\partial X}\right]_{t_f} Q_j^{t_f}\right]^T \lambda = -E_j$$
$$\left[\frac{\partial h}{\partial X}\right]_{t_f} \sum_{i=1}^{N_p} Q_j^{t_f} C_j^{i+1} + 0_{p\times p}\lambda = D_\lambda - \Delta Y_{t_f}^* \qquad (62)$$

Equations (62) give the necessary conditions of optimality. Clearly these equations are linear in terms of $C_0^{i+1}, C_1^{i+1}, \ldots, C_{N_p}^{i+1}$ and λ. Hence these equations together can be written in the matrix form as below

$$AX = B \qquad (63)$$

where, A, X, and B are the following matrices in their respective order, (Let $\left[\frac{\partial h}{\partial X}\right]_{t_f}.Q_j^{t_f} \triangleq S_j^{t_f}$)

$$\begin{bmatrix} R_{11} & \cdots & R_{1N_p} & \left[S_1^{t_f}\right]^T \\ \vdots & \ddots & \vdots & \vdots \\ R_{N_p1} & \cdots & R_{N_pN_p} & \left[S_{N_p}^{t_f}\right]^T \\ S_1^{t_f} & \cdots & S_{N_p}^{t_f} & 0_{p\times p} \end{bmatrix} \begin{bmatrix} C_1^{i+1} \\ \vdots \\ C_{N_p}^{i+1} \\ \lambda \end{bmatrix} = \begin{bmatrix} E_1 \\ \vdots \\ E_{N_p} \\ D_\lambda - \Delta Y_{t_f}^* \end{bmatrix}$$

Assuming A is non-singular, X can be computed by the following expression

$$X = A^{-1} \cdot B \qquad (64)$$

Since the matrix X contains the updated coefficents $\left(C_j^{i+1}\right)$, using which and the relation (44) the control update can be calculated as follows

$$U^{i+1}(t) = \sum_{j=1}^{N_p} C_j^{i+1} P_j(t) \qquad (65)$$

However, to calculate matrices A and B, first, matrices $R_{k,j}$, E_j, and $Q_j^{t_f}$ need to be evaluated. The following subsection discusses in detail how to calculate these matrices in a computationally efficient manner using Gaussian Quadrature and collocation methods. Matrices $R_{k,j}$, E_j, and $Q_j^{t_f}$ are referred to as control update matrices.

D. Computation of control update matrices using Gaussian Quadrature and Collocation methods

The following sections explain how to evaluate the control update matrices for the current formulation:

1) $R_{k,j}$ matrix: To calculate $R_{k,j}$ matrix using Gaussian quadrature, first the integral is defined in terms of scale time $\omega \in [-1,1]$ from time $t \in [t_0, t_f]$ by doing the following substitution

$$t \equiv \frac{t_f - t_0}{2}\omega + \frac{t_f + t_0}{2}, \; dt = \frac{t_f - t_0}{2} d\omega \qquad (66)$$

$$\Rightarrow R_{j,k} = \frac{t_f - t_0}{2} \int_{-1}^{1} P_j(\omega)\, R(\omega)\, P_k(\omega)\, d\omega \qquad (67)$$

Next, the collocation points $\omega_l\, (l = 1, 2, \ldots, N_{C_\omega})$ are defined and selected such that both endpoints are included i.e, $\omega_1 = -1$ and $\omega_N = -1$. In this formulation, the Legendre-Gauss-Lobatto (LGL) method of selecting collocation points is opted. According to Gaussian Quadrature (GQ), the integral (67) can be evaluated below

$$R_{j,k} = \frac{t_f - t_0}{2} \sum_{l=1}^{l=N_{C_\omega}} P_j(\omega_l)\, R(\omega_l)\, P_k(\omega_l) \cdot \mu(\omega_l) \qquad (68)$$

where, $\mu(\omega_l)$ are the Gaussian quadrature weights corresponding to the collocation point ω_l.

2) E_j matrix: According to Eq (59), E_j is given by the following expression

$$E_j = \int_{t_0}^{t_f} \left[\left[\int_{t_0}^{t} \left[\left[B_{S_2}^t(\tau)\right]^T P_j(\tau) \right] d\tau \right] \left[\frac{\partial G}{\partial X}\right]_{X^i(t)} \right] dt \quad (69)$$

Here, there are two running variables $\tau \in [t_0, t]$ and $t \in [t_0, t_f]$, in order to solve the above double integral expression in a computationally efficient manner, Gaussian quadrature is applied twice in a sequential manner. First, to simplify the algebra E_j is written in terms of Q_j^t as follows

$$E_j = \int_{t_0}^{t_f} \left[\left[Q_j^t\right]^T \left[\frac{\partial G}{\partial X}\right]_{X^i(t)} \right] dt \qquad (70)$$

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

Similar to $R_{k,j}$, time is substituted with scale time $\omega \in [-1,1]$ and GQ is applied as follows

$$E_j = \frac{t_f - t_0}{2} \int\limits_{-1}^{1} \left[\left[Q_j^t\right]^T \left[\frac{\partial G}{\partial X}\right]_{X^i(t)} \right]_{t=\omega} d\omega$$

$$E_j = \frac{t_f - t_0}{2} \sum_{l=1}^{l=N_{C_\omega}} \left[Q_j^t\right]^T \Big|_{t=\omega_l} \left[\frac{\partial G}{\partial X}\right]_{X^i(t=\omega_l)} \cdot \mu(\omega_l) \quad (71)$$

Thus, to compute E_j the gradient $\frac{\partial G}{\partial X}$ needs to be evaluated at a few collocation time instants i.e., $t = \omega_l$ and also the integral $\left[Q_j^t\right]$ needs to be evaluated at these time instants. To Compute the integral $\left[Q_j^t\right]$ at $t = \omega_l$, let time t_c correspond to any of the collocation time instants as follows,

$$Q_j^{t_c} = \int\limits_{t_0}^{t_c} \left[B_S^{t_c}(\tau) P_j(\tau)\right] d\tau \quad (72)$$

To compute the above integral using GQ first the intermediate time variable $\tau \in [t_0, t]$ is substituted in terms of another scale time variable $\alpha \in [-1,1]$ as follows and then GQ is applied

$$\tau \equiv \frac{t_c - t_0}{2}\alpha + \frac{t_c + t_0}{2}, \; d\tau = \frac{t_c - t_0}{2} d\alpha \quad (73)$$

$$Q_j^{t_c} = \frac{t_c - t_0}{2} \sum_{l=1}^{N_{C_\alpha}} \left[B_S^{t_c}(\tau)\big|_{\tau=\alpha_l} P_j(\tau)_{(\tau=\alpha_l)} \right] \cdot \xi(\alpha_l) \quad (74)$$

where, $\alpha_l (l = 1, 2, \ldots, N_{C_\alpha})$ are another set of Legendre-Gauss-lobatto points and $\xi(\alpha_l)$ are the corresponding weights. Value of the basis function at $\tau = \alpha_l$ can be evaluated since the basis functions are already known, but to solve and obtain the value for $B_S^{t_c}(\tau)$ at $\alpha_l (l = 1, 2, \ldots, N_{C_\alpha})$ points, the following relation as derived in (41) needs to be evaluated

$$B_S^{t_c}(\tau)\big|_{\tau=\alpha_l} \triangleq W(\tau)\big|_{\tau=\alpha_l} \left[\frac{\partial f}{\partial U}\right]_{(X(\tau),U(\tau))} \Bigg|_{\tau=\alpha_l} \quad (75)$$

and to evaluate $W(\tau)|_{\tau=\alpha_l}$ for a given collocation time instant t_c, the Matrix Dynamic Equation (MDE) (38) needs to solve with the boundary condition applied at time t_c as follows

$$\dot{W}(\tau) = -W(\tau) \frac{\partial f}{\partial X} \quad (76)$$

$$W(t_c) = I_{n \times n} \quad (77)$$

Since, the value of $W(t)$ at the final time t_c is known, using (76) the value of $W(t)$ can be obtained by using different numerical integration techniques. Section III-D4 discusses the solution of the MDE using the collocation method.

3) $Q_j^{t_f}$ matrix: The matrix $Q_j^{t_f}$ is given by the following expression

$$Q_j^{t_f} = \int\limits_{t_0}^{t_f} \left[B_S^{t_f}(\tau) P_j(\tau)\right] d\tau \quad (78)$$

As a consequence of evaluating E_j matrix at LGL points (which includes points at the boundary i.e., t_0 and t_f), at time $t_c = \omega_{N_{C_\omega}}$, $t_c = t_f$. Thus, while calculating the matrix E_j, $Q_J^{t_f}$ is already evaluated and need not be computed again.

4) Solution for the MDE using collocation method: Since Eq (76) is a matrix differential equation, it can be computationally intensive to compute the solution using traditional methods. To solve the MDE efficiently collocation method is used, first, τ is substituted by α using the relation (73)

$$\dot{W}(\alpha) = -W(\alpha) \left[\frac{\partial f}{\partial X}\right]_{\tau=\alpha} \cdot \frac{t_c - t_0}{2} \quad (79)$$

Let $f_x(\alpha) \triangleq \left[\frac{\partial f}{\partial X}\right]_{\tau=\alpha} \cdot \frac{t_c - t_0}{2}$, then

$$\dot{W}(\alpha) = -W(\alpha) f_x(\alpha) \quad (80)$$

For all collocation points, i.e, $\alpha_l (l = 1, 2, \ldots, N_{C_\alpha})$, the above Eq (80) needs to be satisfied, however matrix $W(\alpha)$ is generally numerically integrated backwards from $\alpha_l = N_{C_\alpha}$ to $\alpha_l = 1$, thus $W(\alpha_1)$ is the result of the last integration performed at $\alpha_l = 2$. Therefore, defining (80) at collocation points as follows

$$\dot{W}(\alpha_l) = -W(\alpha_l) f_x(\alpha_l), \forall l = 2, 3, \cdots, N_{C_\alpha} \quad (81)$$

Moreover, the above equation is solved row-wise, i.e, the solution of the k^{th} row $(k = 1, \ldots, n)$ of matrix $W(\alpha)$ is obtained for all collocation points, which would satisfy the following differential equation

$$\dot{W}_k(\alpha_l) = -W_k(\alpha_l) f_x(\alpha_l) \quad (82)$$

Let $w_{kj}(\alpha)$ denote the j^{th} element of the row vector $W_k(\alpha)$, and let the value of $w_{kj}(\alpha)$ be approximated by N_{C_α}th degree Lagrange interpolating polynomials denoted by $\phi_l(\alpha)$ as follows,

$$w_{kj}(\alpha) \approx \sum_{l=1}^{N_{C_\alpha}} \phi_l(\alpha) w_{kj}(\alpha_l) = \Phi(\alpha) W_{kj} \quad (83)$$

where, $\Phi(\alpha) = \left[\phi_1(\alpha), \phi_2(\alpha), \ldots, \phi_{N_{C_\alpha}}(\alpha)\right]$ and $W_{kj} = \left[w_{kj}(\alpha_1), w_{kj}(\alpha_2), \ldots, w_{kj}\left(\alpha_{N_{C_\alpha}}\right)\right]^T$. Using the above relation, the expression for the derivative of $w_{kj}(\alpha)$ at any collocation point $\alpha_l (l = 2, \ldots, N_{C_\alpha})$ can be written as,

$$\dot{w}_{kj}(\alpha_l) = \sum_{l=1}^{N_{C_\alpha}} \dot{\phi}_l(\alpha) w_{kj}(\alpha_l) = \dot{\Phi}(\alpha_l) W_{kj} \quad (84)$$

The above expression can be computed at $(l = 2, \ldots, N_{C_\alpha})$ and can be written in a vector form as follows

$$\dot{W}_{kj}^{''} = \mathbf{D} W_{kj} \quad (85)$$

where, $W_{kj}^{''} = \left[w_{kj}(\alpha_2), \ldots, w_{kj}\left(\alpha_{N_{C_\alpha}}\right)\right]^T$ and $\mathbf{D} = \left[\dot{\Phi}(\alpha_1)^T, \ldots, \dot{\Phi}\left(\alpha_{N_{C_\alpha}}\right)^T\right]^T$. Similarly, expanding the above expression for every element in W_k and defining matrices $\varsigma_k^{''}$ and ς_k as follows,

$$\left[\dot{W}_{k1}^{''}, \ldots, \dot{W}_{kn}^{''}\right] = \mathbf{D}\left[W_{k1}, \ldots, W_{kn}\right] \quad (86)$$

$$\dot{\varsigma}_k^{''} = \mathbf{D} \cdot \varsigma_k \quad (87)$$

The above equation gives the relation between $\dot{W}_k^{''}$ and W_k at the collocation points using Lagrange's interpolating

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

polynomial. However, using (82), the same relation can be obtained as

$$\begin{bmatrix} \dot{W}_k\left(\alpha_2\right) \\ \vdots \\ \dot{W}_k\left(\alpha_{N_{C_\alpha}}\right) \end{bmatrix} = -\begin{bmatrix} W_k\left(\alpha_2\right) f_x\left(\alpha_2\right) \\ \vdots \\ W_k\left(\alpha_{N_{C_\alpha}}\right) f_x\left(\alpha_{N_{C_\alpha}}\right) \end{bmatrix} \tag{88}$$

It is clear that ς_k and $\varsigma_k^{''}$ can also be defined as $\varsigma_k = \left[W_k(\alpha_1)^T, \cdots, W_k\left(\alpha_{N_{C_\alpha}}\right)^T\right]^T$ and $\varsigma_k^{''} = \left[W_k(\alpha_2)^T, \cdots, W_k\left(\alpha_{N_{C_\alpha}}\right)^T\right]^T$. By defining the vector valued function of ς_k and $\varsigma_k^{''}$ as $\vec{\varsigma}_k = \left[W_k\left(\alpha_1\right), \cdots, W_k\left(\alpha_{N_{C_\alpha}}\right)\right]^T$ and $\vec{\varsigma}_k^{''} = \left[W_k\left(\alpha_2\right), \cdots, W_k\left(\alpha_{N_{C_\alpha}}\right)\right]^T$, the above relation can be defined as

$$\dot{\vec{\varsigma}}_k^{''} = -\mathbf{f} \cdot \vec{\varsigma}_k \tag{89}$$

where, $\mathbf{f} = \left[0_{n(N-1)\times n}, diag\left[f_x\left(\alpha_2\right), \ldots, f_x\left(\alpha_{N_{C_\alpha}}\right)\right]^T\right]$. Re-writing Eq (86) in terms of vector-valued function gives

$$\dot{\vec{\varsigma}}_k^{''} = \mathbf{D}\vec{\cdot}\varsigma_k = \left(\mathbf{D} \otimes I_{n\times n}\right) \cdot \vec{\varsigma}_k \tag{90}$$

Now, comparing Eq (89) and Eq (90) as follows, and also since $W_k\left(\alpha_{N_{C_\alpha}}\right)$ is known as the boundary condition is given, $\vec{\varsigma}_k$ can be re-arranged in terms of known and unknown variables to solve the following relation

$$\left(\mathbf{f} + \left(\mathbf{D} \otimes I_{n\times n}\right)\right) \cdot \vec{\varsigma}_k = A_W \cdot \vec{\varsigma}_k = 0 \tag{91}$$

$$\begin{bmatrix} A_{W1} & A_{W2} \end{bmatrix} \cdot \begin{bmatrix} X_{Wk} \\ W_k\left(\alpha_{N_{C_\alpha}}\right)^T \end{bmatrix} = 0 \tag{92}$$

This leads to

$$\Rightarrow X_{Wk} = -A_{W1}^{-1} \cdot A_{W2} W_k\left(\alpha_{N_{C_\alpha}}\right)^T \tag{93}$$

where, $X_{W_k} = \left[W_k\left(\alpha_1\right), \cdots, W_k\left(\alpha_{N_{C_\alpha}-1}\right)\right]^T$, and using Eq (93) the value of X_{Wk} can be obtained. By varying just the value of $W_k\left(\alpha_{N_{C_\alpha}}\right)$ from $k = 1, \cdots, n$, the value of the weighting matrix can be obtained at the collocation points $\alpha_l (l = 1, 2, \ldots, N_{C_\alpha} - 1)$. This completes the solution for the MDE solved at the boundary condition applied at a given collocation time $t_c = \omega_l$ and by varying the value of ω_l for $l = 1, 2, \ldots, N_{C_\omega}$, the solution to the MDE can be obtained for all the required time instants.

Using the above procedure, all the control update matrices necessary to compute the control update (64) can be computed, and using the relation given in (65) the control update at a given iteration can be calculated. Apart from this, the guidance algorithm has to run at each time step to obtain the converged optimal solution. As the algorithm requires an initial guess to start for the very first time, the guess history is generated using any of the available methods (the guess control history need not be precise and very good). From the next time onwards the guess history is obtained from the previous time step converged history. In this manner, the proposed MPSP-based guidance algorithm becomes independent of the first-time initial guess and becomes self-sustained.

Remark: It can be observed that the entire control update can be evaluated by just solving one matrix equation (64) of small dimension owing to spectral transcription of the control variable. Moreover, the elements of the matrix equation can be computed by just performing a few algebraic calculations at collocation points which further adds to the computational efficiency of the algorithm.

IV. Simulation results

It is worth mentioning here that the simulation results are given in the normalized scale for obvious reasons. The initial conditions of the interceptor missile and the ballistic target for the ideal case are set as $\gamma_{e_M} = 60$ deg, $\gamma_{a_M} = 0$ deg, and $\gamma_{e_T} = -50$ deg, $\gamma_{a_T} = 180$ deg. The desired terminal impact angles for the ideal case are set to $\gamma_{M_d} = \psi_{M_d} = 0°$. Without loss of generality, the first fifteen Legendre polynomials have been used to transcribe the control history (44), and the number of Legendre-Gauss-Lobatto collocation points N_{C_ω}, and N_{C_α} are both selected to be equal to twenty. Moreover, It can be observed that MPSP guidance requires an initial guess of the control history, for the present simulation Generalised explicit (GENEX) guidance has been utilized to obtain the guidance demand over the whole trajectory. The details about GENEX guidance is given in [6]. Once the initial guess of the control history is obtained, the corresponding coefficients of basis functions can also be obtained. Note that for all the simulations performed the convergence criteria for the guidance algorithm was set to ± 1 meter in position and ± 0.15 deg in terminal impact angle. To demonstrate the benefit of the proposed minimum drag guidance, a speed comparison between the minimum drag unified guidance and minimum control effort-based guidance for the ideal case is shown in Figure 4, it can be observed that the minimum drag guidance leads to a huge gain in speed of approximately $\%16.5$ at the time of interception as compared to minimum control effort based guidance. Further, it can be observed from Figure 4 that the overall speed of the interceptor remains higher in the case of minimum drag guidance throughout the trajectory, which leads to the added advantage of increased maneuvering capacity. This section is divided into three parts, Section IV-1 provides the simulation results corresponding to different engagement scenarios. Second IV-2 provides details and results for the closed-form simulation with handover error and parameter perturbation, and the last Section IV-3 provides the run-time details of the algorithm.

1) Different Engagement Scenarios: The Unified Guidance scheme is tested against a wide range of terminal impact angles. Generally, due to limited time available in the terminal phase the terminal guidance schemes [2], [14], are only able to account for a 10 to 20 deg band of variation in terminal impact angles. But, since the unified guidance scheme proposed in this paper starts accounting for terminal constraints from the midcourse phase itself, the ability of the formulation to account for larger impact angles increases which is evident from Table I and Figures 6, 7. The interceptor is able to satisfy the terminal conditions within the desired threshold for a complete 60 deg band of variation in the terminal elevation

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

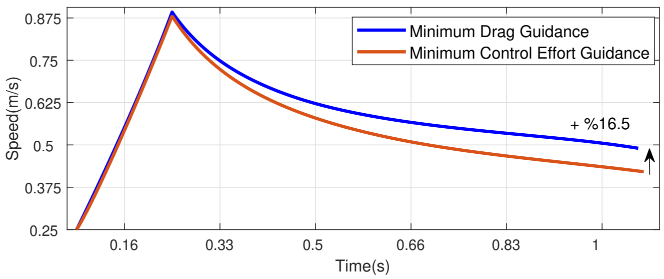

Fig. 4. Interceptor speed vs time

and azimuth angle. Table I shows the miss distance and error in the terminal impact angle for different cases of desired impact angles. Figure 6 shows the azimuth and elevation angles for all the different cases and Figure 7 provides the corresponding guidance demand generated for each case. It can be observed from Figure 6, that the interceptor can satisfy all the different terminal impact angle cases given in Table I, at the point of interception.

2) Handover Error and Perturbation Testing: In a realistic scenario, the midcourse guidance uses target updates from a long-range radar, and the target state accuracy is not very high. However, during the terminal phase of the guidance, an onboard seeker is utilized to obtain the target track. As the target is near the interceptor during the terminal guidance phase of flight, the onboard seeker provides high-accuracy target information to the guidance. Because of this reason, in general, a step jump in the state information of the target appears during the transition from the midcourse to the terminal phase; this is known as a handover error. To replicate this scenario a closed-form simulation based on range-based transfer from midcourse to the terminal phase is designed. As the relative range between the target and the interceptor becomes less than the homing range of the onboard seeker, the guidance switches to the terminal phase, and the true position of the target are passed to the guidance algorithm. By considering ± 50 meters jump in position as the standard deviation (1σ) value, the guidance algorithm was tested for all possible handover error cases with the worst-case scenario being ± 200 meters (4σ) of state jump. Further, all possible combinations of ± 200 meters of error in position $\begin{bmatrix} x_{err} & y_{err} & z_{err} \end{bmatrix}$ were taken, out of which the results for the worst case is reported in Table II, and Figures 9, 8. It can be observed from Figures 9 that when the true position of the target gets known as the missile transitions from midcourse to terminal phase, a sudden change in the guidance demand is experienced. Figure 8 gives the 3D trajectory for the handover error case 1 in Table II, through which the change in the 3D trajectory of the interceptor as compared to the ideal case is clearly visible in the terminal phase of the interception. Table II provides the details about the corresponding error in terminal constraints, which are satisfied within the given tolerance.

Similarly, the simulation results for the worst case of parameter perturbation is also done by considering a $+\%15$, $-\%15$, and $+\%20$ perturbation in axial force coefficient (C_A), normal force coefficient (C_N) and ballistic coefficient (β_T). This combination generates one of the worst-case scenarios, because (i) the higher the axial force coefficient, the higher the profile drag, and (ii) the lower the normal force coefficient, the higher the induced drag. Moreover, since the ballistic coefficient of the target is assumed to be 20% higher which leads to a lower drag on the target, the predicted target position will be always lower. Results for the combined worst-case scenario of parameter perturbation and handover error are also provided. The corresponding guidance demand and error in terminal constraints for both of these cases are given in Table II and Figure 9. It can be observed that the maximum miss distance of only 0.38 meters for the combined worst-case scenario shows the robustness and accuracy of the algorithm. Next, a realistic performance of the guidance algorithm is shown through a Monte-Carlo-like simulation study by varying the plant parameters and injecting handover error. The interceptor parameters (C_A), (C_N), and (β_T) have been varied by using a normal distribution with a 3σ value of 15%, while the handover error is varied with a 3σ value of 150 meters. Five hundred Monte-Carlo runs have been given and the maximum miss distance in these 500 runs has been found to be only $0.2176\ m$. The cumulative distribution function (CDF) corresponding to these Monte-Carlo runs has been given in Fig. 10, which corroborates this fact.

3) Computational Efficiency and Runtime: All the simulations were performed in MATLAB R2021a, running on a desktop with an Intel i5-2400S CPU @ 2.50GHz. With the specified convergence criteria, the average time taken for computing the guidance demand in the Monte-Carlo simulations per guidance cycle was found to be only 0.21 seconds with a standard deviation of 0.016 seconds at the start of the midcourse phase. In the terminal phase, PS-MPSP only took 0.021 seconds, and given the plant dynamics in the closed-form simulation were updated after every 0.2 seconds in the midcourse phase and after 0.02 seconds in the terminal phase, the computation time of PS-MPSP is very close to the guidance update cycle. This demonstrates the algorithm's computational efficiency and real-time capability. After the algorithm becomes self-sustained i.e., independent of the initial guess, it was found that the PS-MPSP-based unified guidance only required one or two iterations to converge each at each guidance cycle. Implementing the guidance algorithm in Embedded C with a dedicated processor can significantly reduce the computational time even further and make it suitable for onboard implementations.

TABLE I
DIFFERENT ENGAGEMENT SCENARIOS

Case	Value*	$\lvert\gamma_M - \gamma_{M_d}\rvert^{\circ}$	$\lvert\psi_M - \psi_{M_d}\rvert^{\circ}$	Miss dist. (m)
1	-30	1.01×10^{-1}	1.1×10^{-1}	7.3×10^{-1}
2	-20	9.1×10^{-2}	2.1×10^{-2}	8.08×10^{-3}
3	-10	1.1×10^{-2}	1.2×10^{-3}	3.02×10^{-3}
4	0	8.7×10^{-3}	1.3×10^{-4}	2.5×10^{-3}
5	10	4.3×10^{-3}	2.5×10^{-3}	9.6×10^{-3}
6	20	3.3×10^{-2}	6.4×10^{-2}	5.8×10^{-2}
7	30	9.4×10^{-2}	1.03×10^{-1}	6.4×10^{-1}

*Here -30 $\Rightarrow \gamma_{M_d} = \psi_{M_d} = -30^{\circ}$

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

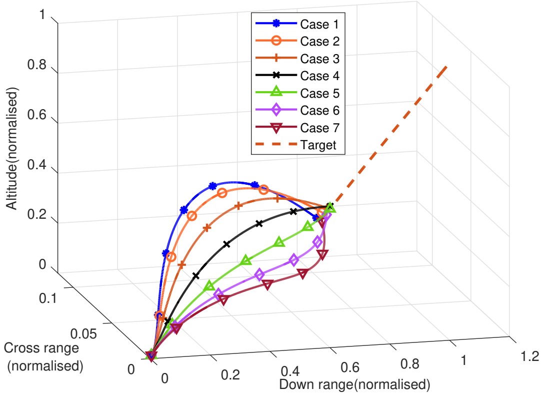

Fig. 5. Different engagement scenarios: 3D trajectory

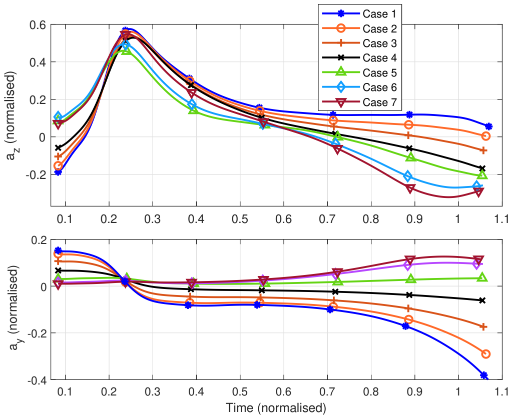

Fig. 7. Different engagement scenarios: Guidance demand

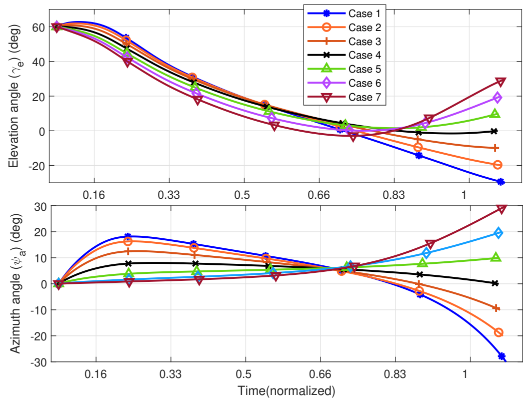

Fig. 6. Different engagement scenarios: Azimuth and elevation angles

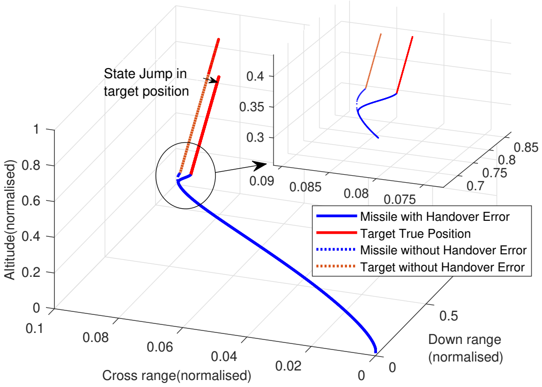

Fig. 8. Handover Error 3D trajectory for Case 1

4) Comparison With Other Guidance Laws: To establish the optimality of the solution obtained using the PS-MPSP, a free final time optimal control problem is formulated that minimizes the cost function (17), subject to state dynamics (1),(2), and guidance constraints defined by (3), (4) and (5). After applying the necessary conditions of optimality, the problem is converted into a two-point boundary value problem which is then solved using the *bvp4c* solver [22] (based on the collocation method) in MATLAB. Further, the minimum drag problem is solved using the MPSP [1] and QS-MPSP [14] techniques, and the corresponding results are summarized in Figures: 11- 15 and Table: III. First, it can be observed from the figures that the solution calculated by all MPSP techniques are very close to the actual optimal solution; moreover, the difference between the total time of engagement computed by the PS-MPSP with relative altitude as the independent variable and the free final time TPBVP was found to be only 0.01 seconds. To maintain equivalence, other MPSP techniques were also formulated with relative altitude as the independent variable. An interesting observation can be made by analyzing the guidance demands generated by different methods. When the thrust phase ends, a sudden change in the guidance demands is observed in MPSP, GENEX, and the TPBVP solutions leading to a point-wise loss of smoothness. However, due to the spectral transcription of the control trajectory, the QS-MPSP and PS-MPSP are able to generate smooth guidance demands that avoid such sudden changes in the solution and adjust the control trajectory accordingly. Smooth guidance demands can be tracked suitably by the inner-loop autopilot; therefore, such solutions provide an additive advantage over the other techniques. The GENEX guidance fails to satisfy the terminal impact angles and ends up with a large miss-distance primarily because the lateral acceleration demanded lies outside the capability of the interceptor and saturates after some time (see Fig. 16). However, guidance demand calculated by the PS-MPSP remains within the capability of the interceptor throughout the trajectory. Note that the total lateral acceleration capability is a function of the altitude and velocity of the interceptor, and since the proposed PS-MPSP minimizes the total drag it ends with a larger velocity facilitating a larger maneuvering capacity closer to the interception, which is desirable.

The computational superiority of the proposed PS-MPSP to solve the minimum drag problem with unified midcourse and terminal phase over other MPSP-based techniques can be corroborated by Table: III. The average time taken by each method per guidance cycle at the start of the midcourse phase and the beginning of the terminal phase was noted over several runs and is reported in Table: III. It is important to note that the computational efficiency of MPSP-based algorithms to solve the current problem mainly depends on two aspects (1) the dimension of the control update equation (64), which

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

TABLE II
TESTING WITH HANDOVER ERROR & PARAMETER PERTURBATION

Case*	$\lvert\gamma_M - \gamma_{M_d}\rvert$ (deg)	$\lvert\psi_M - \psi_{M_d}\rvert$ (deg)	Miss dist. (m)
HE:-200	4.2×10^{-2}	8.2×10^{-2}	1.2×10^{-1}
Ideal:0	8.7×10^{-3}	1.3×10^{-4}	2.5×10^{-3}
P.Per	3.6×10^{-2}	2.1×10^{-3}	1.4×10^{-1}
HE+P.Per	5.1×10^{-2}	2.9×10^{-2}	3.8×10^{-1}

*Here, HE: Handover Error, P.Per: Parameter Perturbation, and $-200 : \begin{bmatrix} x_{err} & y_{err} & z_{err} \end{bmatrix} = \begin{bmatrix} -200 & -200 & -200 \end{bmatrix}$ meters

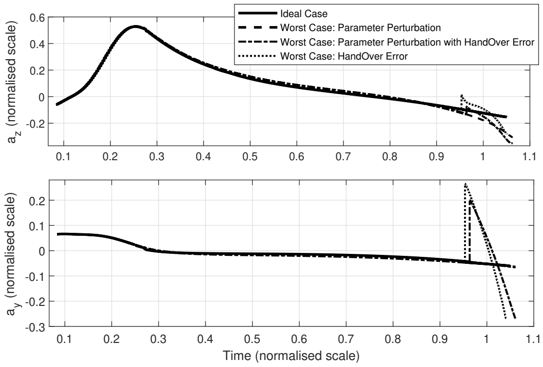

Fig. 9. Guidance Demand: Parameter Perturbation and Handover Error

is required to be computed at each iteration, and (2) how efficiently the elements of the control update matrices A and B can be computed. The solutions of MPSP and QS-MPSP can be written similarly to equation (64). The adverse effect of temporal discretization on the computational efficiency of MPSP can be observed in Table: III. It took MPSP on average 2.51 seconds at the start of the midcourse phase and only 39 ms in the terminal phase. Discrete MPSP results in a large dimensional equation (64) due to many optimization variables present at the start of the trajectory and consequently increase the computational load. Moreover, the sensitivity matrices must be calculated at each node, which is the most time-consuming segment of MPSP-class algorithms. However, the optimization variables decrease as time progresses, and the computation time improves. A small yet not substantial improvement in the computation time is observed in the QS-MPSP, which is a consequence of opting for spectral transcription of the control variable that reduces the dimension of equation (64). But like MPSP, it also discretizes the problem in time and suffers from the considerable computation time taken for computing the sensitivity matrices at each node. A significant improvement over these methods is observed using PS-MPSP, which retains the benefit of the spectral transcription and only performs calculations at a few collocation points, efficiently calculating the control update matrices. Approximately $12\times$ gain in computation time is observed compared to MPSP in the midcourse phase and $2\times$ in the terminal phase.

Moreover, as the total duration of the trajectory increases, the computational efficiency of the PS-MPSP does not deteriorate. Adding a few higher-order polynomials allows the new trajectory to be captured fairly accurately. From the above study, it can be concluded that the PS-MPSP provides a significant advantage over the other techniques compared in this section.

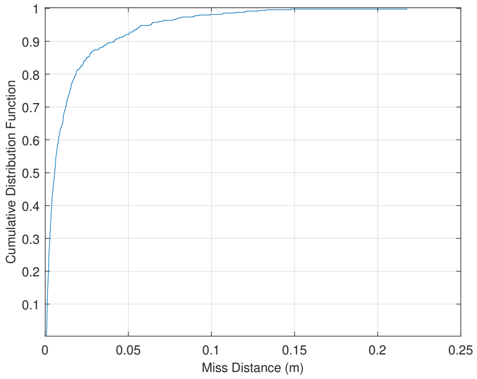

Fig. 10. CDF of miss distance in Monte-Carlo runs

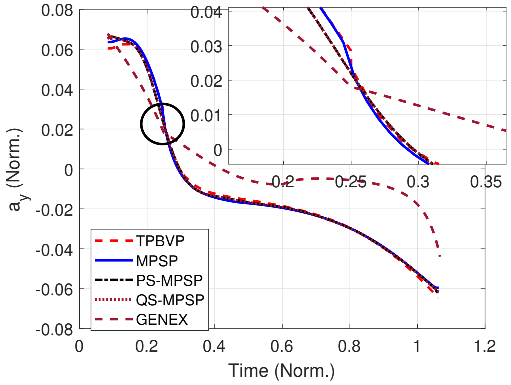

Fig. 11. Guidance Techniques Comparison: a_y

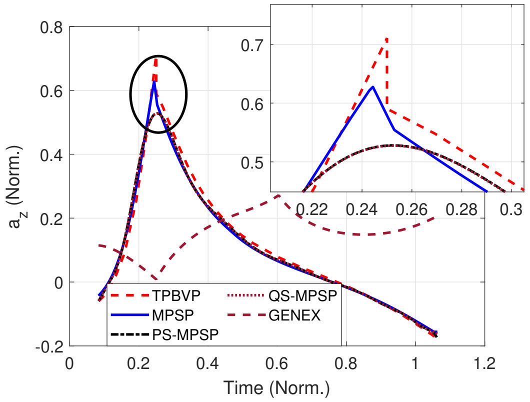

Fig. 12. Guidance Techniques Comparison: a_z

V. CONCLUSION

A computationally-efficient optimal guidance scheme is presented in this paper utilizing the Pseudo-Spectral MPSP, which includes state-dependent terms in the cost function of the recently proposed State-dependent Generalised QS-MPSP technique. This PS-MPSP technique is applied next to solve a unified midcourse-cum-terminal phase guidance problem.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

TABLE III
COMPARISON OF GUIDANCE TECHNIQUES: COMPUTATIONAL AND CONCEPTUAL

Method Method	Computation Time Unified (Mid. + Ter.) (sec)	Computation Time Terminal (sec)	Smooth Guidance Demand	Terminal Contraints	Control Transcription	Discretization
TPBVP (*bvp4c*)	15.16	2.94	✗	✓	–	–
GENEX	0.016	0.002	✗	✗	–	–
MPSP	2.51	0.039	✗	✓	✗	Euler
QS-MPSP	1.73	0.032	✓	✓	Legendre Poly.	Euler
PS-MPSP	0.21	0.021	✓	✓	Legendre Poly.	Gaussian Quad. (LGL)

*Here Mid. ⇒ Midcourse Phase, Ter. ⇒ Terminal Phase, and LGL ⇒ Legendre-Gauss-Lobatto

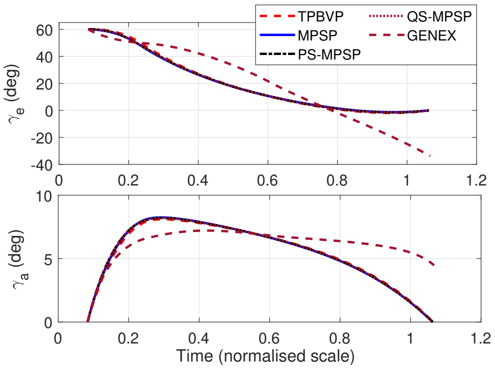

Fig. 13. Guidance Techniques Comparison: Impact Angle

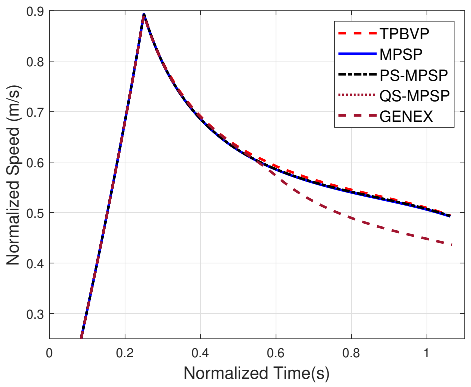

Fig. 14. Guidance Techniques Comparison: Speed

This innovative formulation minimizes the total deceleration acting on the interceptor throughout its trajectory. Besides, it satisfies the terminal constraints on impact angle besides ensuring near-zero miss distance. The problem is formulated in terms of relative height to solve a free final-time optimal control problem. The proposed QS-MPSP computational guidance illustrates good performance against a wide range of terminal impact angles and handover errors. Randomized Monte-Carlo-like randomized simulation of the guidance with the realistic bounds of plant parameter perturbations illustrates its promising performance.

REFERENCES

[1] P. Kumar, A. Bhattacharya, and R. Padhi, "Minimum drag optimal guidance with final flight path angle constraint against re-entry targets," in *AIAA SciTech Forum*. American Institute of Aeronautics and Astronautics, Jan. 2018.

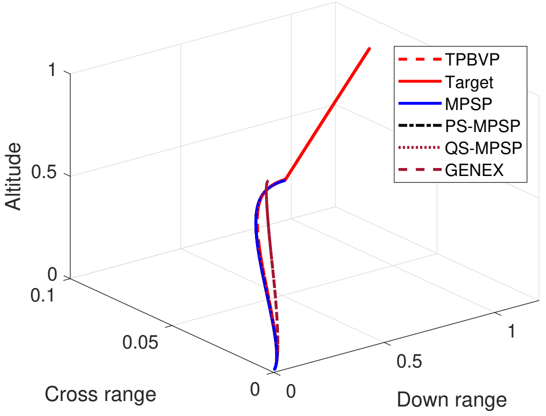

Fig. 15. Guidance Techniques Comparison: Trajectory

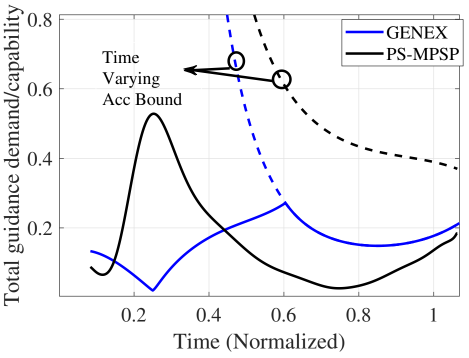

Fig. 16. Guidance Techniques Comparison: GENEX VS PS-MPSP total guidance demand and capability

[2] S. Mondal and R. Padhi, "Constrained Quasi-Spectral MPSP With Application to High-Precision Missile Guidance With Path Constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 143, no. 3, 10 2020. [Online]. Available: https://doi.org/10.1115/1.4048488

[3] S.-M. Yang, "Analysis of optimal midcourse guidance law," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 419–425, January 1996.

[4] U. K. Sahu, P. N. Dwivedi, A. Bhattacharya, and R. Padhi, "Modified generalized explicit guidance for midcourse with near-zero lateral acceleration in terminal phase," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3954 – 3959, 2014, 19th IFAC World Congress.

[5] R. W. Morgan, "Midcourse guidance with terminal handover constraint," in *2016 American Control Conference (ACC)*, July 2016, pp. 6006–6011.

[6] E. J. Ohlmeyer and C. A. Phillips, "Generalized vector explicit guidance," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 261–268, March 2006.

[7] J.-H. Kim, S.-S. Park, K.-K. Park, and C.-K. Ryoo, "Quaternion based three-dimensional impact angle control guidance law," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2311–

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3237796

2323, 2021.

[8] D. E. Kirk, *Optimal Control Theory: An Introduction*. Prentice Hall, 1970, ch. 6, pp. 329–413.

[9] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, ser. Advances in design and control. Society for Industrial and Applied Mathematics, 2001, ch. 3 and 4, pp. 91–216.

[10] P. Lu, "Introducing computational guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 193–193, 2017.

[11] T. M. Lovelly and A. D. George, "Comparative analysis of present and future space-grade processors with device metrics," *Journal of Aerospace Information Systems*, vol. 14, no. 3, pp. 184–197, 2017.

[12] R. Padhi and M. Kothari, "Model predictive static programming: A computationally efficient technique for suboptimal control design," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 2, pp. 399–411, Feb 2009.

[13] O. Halbe, R. G. Raja, and R. Padhi, "Robust reentry guidance of a reusable launch vehicle using model predictive static programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 134–148, 2014.

[14] S. Mondal and R. Padhi, "Angle-constrained terminal guidance using quasi-spectral model predictive static programming," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 3, pp. 783–791, 2018.

[15] P. N. Dwivedi, A. Bhattacharya, and R. Padhi, "Suboptimal midcourse guidance of interceptors for high-speed targets with alignment angle constraint," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 3, pp. 860–877, 2011.

[16] R. Padhi, N. Unnikrishnan, X. Wang, and S. Balakrishnan, "A single network adaptive critic (snac) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Networks*, vol. 19, no. 10, pp. 1648 – 1660, 2006.

[17] A. Maity, H. B. Oza, and R. Padhi, "Generalized model predictive static programming and angle-constrained guidance of air-to-ground missiles," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1897–1913, 2014. [Online]. Available: https://doi.org/10.2514/1.G000038

[18] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[19] C. Zhou, X. Yan, and S. Tang, "Generalized quasi-spectral model predictive static programming method using gaussian quadrature collocation," *Aerospace Science and Technology*, vol. 106, p. 106134, 2020.

[20] S. Ghosh and S. Mukhopadhyay, "Tracking reentry ballistic targets using acceleration and jerk models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 666–683, 2011.

[21] M. Asad, S. Khan, Ihsanullah, Z. Mehmood, Y. Shi, S. A. Memon, and U. Khan, "A split target detection and tracking algorithm for ballistic missile tracking during the re-entry phase," *Defence Technology*, vol. 16, no. 6, pp. 1142–1150, 2020.

[22] J. Kierzenka and L. F. Shampine, "A bvp solver based on residual control and the maltab pse," *ACM Trans. Math. Softw.*, vol. 27, no. 3, p. 299–316, sep 2001.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: TOHOKU UNIVERSITY. Downloaded on March 30,2023 at 04:07:48 UTC from IEEE Xplore. Restrictions apply.