International Journal of Control, Automation, and Systems 21(6) (2023) 1866-1884
http://dx.doi.org/10.1007/s12555-022-0660-0
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Fixed Range Horizon MPPI-based Missile Computational Guidance for Constrained Impact Angle

Ki-Pyo Kim and Chang-Hun Lee*

Abstract: This paper presents a new computational guidance algorithm based on the model predictive path integral (MPPI) control for missiles with the impact angle, seeker's look angle, and acceleration constraints. The MPPI control is one of the optimization approaches using the stochastic process, and the optimal control input is determined using sample trajectories generated by propagating the system model. Thus, the MPPI control can be considered as a data-driven method for solving nonlinear and constrained optimization problems. The proposed guidance algorithm consists of the proportional navigation (PN) guidance command with a time-varying gain to be optimized at every guidance cycle by utilizing the iterative path integral technique in conjunction with the importance sampling under the model predictive control (MPC) philosophy. Unlike existing approaches, this approach allows us to effectively solve nonlinear guidance problems without the convexification or linearization process. It can also adapt to environmental changes by reflecting the current system state variables. Furthermore, unlike other computational guidance approaches, the proposed algorithm does not rely on a dedicated solver for optimization problems. In this study, numerical simulations are performed to investigate the effectiveness and applicability of the proposed guidance algorithm.

Keywords: Computational guidance, impact-angle-control, missile guidance, model predictive path integral (MPPI) control.

1. INTRODUCTION

Over the past several decades, in the guidance problems, impact angle control has been widely considered to maximize the warhead effect for the anti-ship and anti-tank missile systems. The most commonly used approach for the impact angle control is to modify the conventional proportional navigation (PN) [1] guidance by adding bias terms or shaping the navigation constant. To this end, several variants of PN guidance laws have been studied, such as biased PN, gain-shaping PN, or switched-gain PN. In [2], PN guidance with a time-varying term was proposed to achieve the impact angle constraint. The authors in [3] proposed a biased PN guidance providing the desired impact angle without the time-to-go information. In [4], an interception angle control guidance was developed based on the PN guidance with the optimal bias term. The authors in [5] introduced PN guidance in conjunction with a gain shaping technique for impact angle control. In [6], a switched-gain PN law was devised for impact angle control.

Another approach is to apply the optimal control theory [7] to guidance problems with impact angle constraints. This approach relies on the linear-quadratic (LQ) control with the linear approximation of the nonlinear engagement geometry in a near-collision course (NCC). In [8], an optimal impact-angle-control guidance for arbitrary missile dynamics was proposed. A time-to-go weighted optimal guidance providing the desired impact angle was developed in order to shape guidance command [9]. Generalized formulation of optimal guidance with impact angle constraint was studied in [10]. This guidance law was further extended by considering autopilot lag and varying missile velocity in [11]. Recently, the physical meaning of optimal guidance law with impact angle constraint was investigated in [12].

The guidance problem with the impact angle constraint is indeed a highly nonlinear control problem due to nonlinear kinematics, the requirement of high precision, and the limitation of the control energy. In this context, the nonlinear control approach has also been used for accomplishing impact angle control. In [13,14], the sliding mode control (SMC) has been applied to impact angle control problems. In [15], an impact angle control guidance law based on the nonsingular terminal sliding mode control was studied. In [16], a homing guidance with an impact angle constraint while ensuring the finite-time conver-

Manuscript received July 12, 2022; revised October 24, 2022; accepted November 6, 2022. Recommended by Senior Editor Bin Jiang.

Ki-Pyo Kim and Chang-Hun Lee are with the Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea (e-mails: {kipyo1871, lckdgns}@kaist.ac.kr).
* Corresponding author.

gence was studied. A nonlinear differential game-based guidance law with an impact angle constraint was suggested in [17]. In [18], an impact angle control guidance was developed based on the feedback linearization control methodology. In [19], the state-dependent Riccati equation (SDRE)-based approach has been applied to guidance problems with the impact angle constraint.

The previous studies on the impact angle control guidance are able to provide the desired impact angle, however these approaches have limited scalability for considering other important practical constraints, such as the seeker's field-of-view (FOV) limit or the maximum acceleration limit. As violation of these constraints results in severe miss distance at the terminal guidance phase, the satisfaction of these constraints is also important as much as achieving the impact angle control. Although some works on impact angle control with the FOV limit have been reported recently [20-23], they still provided only limited solutions for other practical constraints. These PN-based guidance laws can control the impact angle with the FOV constraint, but they have a deficiency in effectively responding to the practical constraint on the maximum lateral acceleration of the missile. Although there is a simple method to react to large guidance commands by clipping the commanded acceleration at the allowable bound, this is unsuitable for severe saturation. In addition, for reliable guidance performance even in inevitable uncertainties, the guidance law should have the ability to adjust PN gains appropriately in a closed-loop form. Here, the closed-loop application means that guidance gains are adequately determined online by solving the nonlinear guidance problem in each guidance cycle by utilizing the current states as the initial values [24].

In this context, the computational guidance concept has recently been introduced [25] with the rapid development of embedded computational capability. The main characteristics of the computational guidance framework are summarized as numerical algorithms and onboard computation involving iterations [26]. One of possible enablers for the computational guidance is the convex optimization [27-30], and it has been recently applied to guidance problems for various aerospace systems [31-33] including missile systems [24,34-37]. An impact angle control guidance with look angle constraint and maximum acceleration command limit was studied by utilizing the convex optimization in [24]. In [34], successive convex programming was applied to guidance problems for hypersonic missile systems. A midcourse guidance algorithm for air-to-ground missiles has been proposed based on the convex programming in [35]. The authors in [36] proposed a midcourse guidance algorithm for guided rockets using sequential convex programming. In [37], an impact time and angle control guidance algorithm was studied based on L1-penalized sequential convex programming. In the convex optimization-based approach, it is required that nonlinear and nonconvex guidance problems are converted into a special class of convex optimization problems such as linear programming (LP), second-order cone programming (SOCP), and semi-definite programming (SDP) by applying appropriate convexification techniques [31]. As the convexification procedure highly depends on types of problems, it is not always possible for all types of problems. Namely, it implies that convex optimization-based approaches can only handle a narrow type of problem with specific performance index that can be transformed into the convex programming problems. Furthermore, these approaches require a dedicated solver to determine optimal solutions for implementing them. This is still an open and challenging issue in the convex optimization-based approach.

For the computational guidance, another potential framework would be the model predictive path integral (MPPI) control. This is mainly based on the information-theoretic interpretation of optimal control using Kullback-Leibler (KL)-divergence and free energy [38], while it was previously based on the linearization of Hamilton-Jacobi-Bellman (HJB) equation and application of Feynman-Kac lemma [39]. The main idea of MPPI control is to sample thousands of trajectories in real-time from the system dynamic model and update the control inputs based on the cost of sampled trajectories by utilizing the importance-sampling technique within the context of stochastic optimal control [40]. After evaluating each sampled trajectory according to the predefined cost function, the control sequence is updated at each guidance cycle. In that sense, the MPPI control can be considered as a data-driven method for solving nonlinear and constrained optimization problems. One of the merits of the MPPI control is that there is no need for derivative information of the system because it is a sampling-based open-loop method. In addition, nonlinear optimal control problems with nonconvex objective and constraint functions can be solved without linearization or convexification [38] process. Moreover, it does not require any dedicated solver for solving optimization problems, unlike the convex optimization-based method. Last but not least, the MPPI control can be thought of as a more efficient approach to take advantage of the parallel nature of sampling by using a graphics processing unit (GPU) to sample thousands of trajectories from the nonlinear dynamics [41]. Accordingly, the MPPI control could be a potential approach for complementing the convex optimization-based approach. Recently, there are some studies applying the MPPI control to various autonomous systems such as an aggressive autonomous driving car [42,43], a fixed-wing aircraft for complex maneuvers [44], a quadrotor navigation in 2D clustered environments [45], and a landing guidance for rotorcraft [46]. On the contrary, few papers applying the MPPI control to missile guidance problems [47,48] are reported in open literature. More specifically, the use of the MPPI

control in missile guidance application is still in the early stage compared to other approaches, including the convex optimization-based method. We still need to study how to make it work.

In this context, this paper aims to investigate the feasibility and potential capability of the MPPI control framework in the missile guidance applications. To be more specific, we establish a procedure and describe the detailed formulation of missile guidance problems to be solved by the MPPI control method. The optimal guidance problem considered is to perform the impact angle control against a stationary target while satisfying the practical limits (i.e., the seeker's FOV limit and the maximum acceleration bound) and minimizing the control energy. To this end, a guidance command is first assumed as the proportional navigation with a time-varying navigation gain (or PN gain) in the proposed method. Using the importance sampling, an iterative path integral optimizes the PN gain under the model predictive control (MPC) setup. The obtained PN gain is then updated using a stochastic optimal approach at every guidance cycle. In order to address the issue of trajectory lengths that vary while always including the target's terminal point, we utilize the relative range as a new independent variable. This approach, referred to as fixed range horizon MPPI, guarantees consistent trajectory lengths at every iteration. For clarity, it is worth mentioning that the objective of this study is not to replace the existing approaches, including the convex optimization-based method in the missile guidance applications, but to extend the methodology with an appealing new addition.

The main contributions of this study can be summarized as fourfold: 1) For solving nonlinear and constrained guidance problems, we propose a new computational guidance algorithm based on the MPPI control that does not require any dedicated effort for convexification or linearization compared to other computational guidance approaches. 2) This study establishes a procedure and presents a detailed formulation for solving nonlinear missile guidance problems using the MPPI control scheme. 3) The proposed method is implementable in a closed-loop fashion to react to unexpected situations or uncertainties by iteratively solving the nonlinear guidance problem at each guidance cycle using the current states as the initial values. Thus, the proposed method would potentially improve the guidance performance compared to existing approaches. 4) We propose a generalizable computational guidance strategy for optimization problems that does not require any dedicated solver.

This paper is organized as follows: Section 2 describes the problem formulation, including the engagement kinematics, objective function, and constraints for guided missiles intended to apply the MPPI control. Section 3 briefly reviews the basic MPPI algorithm. In Section 4, a guidance algorithm based on the MPPI control is discussed. The simulation results are presented in Section 5. Finally, the conclusions are offered in Section 6.

2. PROBLEM DEFINITION

This section explains the optimal guidance problem considered in this study. First, the nonlinear engagement kinematics against a stationary target is discussed. Inspired by [24], the guidance problem is formulated in a way to optimize a time-varying PN gain to satisfy the practical constraints in this study.

2.1. Nonlinear engagement kinematics

Fig. 1 illustrates the planar homing engagement geometry against a stationary target. In this figure, the notation $X-Z$ represents the inertial reference frame. The points M and T denote the missile and target, respectively. The variables r and σ denote the relative range and line-of-sight (LOS) angle between the missile and target, respectively. The notation V_M denotes the missile speed, and the variable a_M represents the normal acceleration. a_M is acting perpendicular to the velocity vector of the missile, which contributes to altering its direction. The parameter γ_f is the desired impact angle. The variables λ_M and γ_M are the look angle and flight path angle. By definition, the look angle is given by

$$\lambda_M \triangleq \gamma_M - \sigma. \tag{1}$$

The sign of the angles is defined as positive for the counterclockwise direction and negative for the clockwise direction. From Fig. 1, the nonlinear engagement kinematics in the polar coordinate system is written as

$$\dot{r} = -V_M \cos\lambda_M, \tag{2}$$

$$r\dot{\sigma} = -V_M \sin\lambda_M, \tag{3}$$

$$\dot{\gamma}_M = \frac{a_M}{V_M}. \tag{4}$$

Note that the normal acceleration a_M can be considered as the control input (i.e., $u = a_M$) in the nonlinear engagement kinematics.

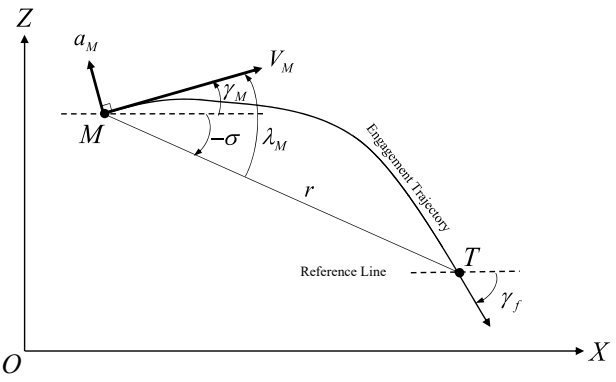


Fig. 1. Engagement geometry.

2.2. Formulation of optimal guidance problem

In this study, we consider an energy optimal guidance problem providing the desired impact angle and zero miss distance while meeting the seeker's FOV limit and maximum acceleration bound. The final boundary conditions for the interception and impact angle control can be written as

$$r(t_f) = 0,\ \gamma_M(t_f) = \gamma_f, \tag{5}$$

where t_f is the final time. From (1) and (3) with (5), the final boundary conditions can be equivalently written as

$$\lambda_M(t_f) = 0,\ \sigma(t_f) = \gamma_f. \tag{6}$$

To maintain the lock on condition, the look angle should be lied in the seeker's FOV limit. This constraint can be expressed as

$$|\lambda_M| \le \lambda_{M\max}, \tag{7}$$

where $\lambda_{M\max} < \pi/2$ is the maximum allowable look angle. In addition, the constraint on the maximum acceleration bound is given by

$$|a_M| \le a_{M\max}. \tag{8}$$

In this study, the seeker's FOV limit $\lambda_{M\max}$ and maximum acceleration bound $a_{M\max}$ are assumed to be known because those can be readily obtained from the missile system. Finally, the optimal guidance problem with the practical constraints can be formulated as follows:

$$P_1:\ \min_{a_M} J = \int_{t_0}^{t_f} a_M(t)^2 dt,$$
$$\text{subject to (1)-(5), (7)-(8),}$$

where t_0 is the initial time. For convenience, the above problem is called P_1 in this study. In the above optimal problem, the normal acceleration a_M can be regarded as a free variable to be optimized.

2.3. Alternative formulation of optimal guidance problem

A direct finding of the optimal control history of a_M in the problem P_1 might cause a slow convergence because possible solution space of a_M is generally huge. In other words, a_M is unstructured in that case. Thus, by taking the structured form of the basic PN guidance command, we can mitigate the problem complexity, which helps improve the convergence characteristic. Furthermore, the previous studies on the trajectory shaping or the command shaping guidance [49-52] provide an insight that all guidance laws can be expressed as an alternative form of PN guidance with a free variable to be determined, as follows:

$$a_M = N(t) V_M \dot{\sigma}, \tag{9}$$

where the variable $N(t)$ denotes the time-varying gain. This fact implies that the free variable a_M can be structured as given in (9), and the time-varying gain can be defined as a new control input.

$$u = N. \tag{10}$$

Note that the time dependence of variables is suppressed for simplicity. By augmenting the structured a_M with u into the problem P_1, the solution space of a_M can be reduced. In order to exploit this favorable property of the structured a_M, we reformulate the optimal guidance problem with respect to u. Substituting (3), (9), and (10) into (4) gives

$$\dot{\gamma}_M = \frac{a_M}{V_M} = -\frac{V_M \sin\lambda_M}{r} u. \tag{11}$$

By combining (1), (3), and (11), we have

$$\dot{\lambda}_M = -\frac{V_M \sin\lambda_M}{r}(u-1). \tag{12}$$

In addition, by substituting (3) into (9), the maximum acceleration bound can be rewritten as

$$|a_M| = \left| u \frac{V_M^2 \sin\lambda_M}{r} \right| \le a_{M\max}. \tag{13}$$

Accordingly, the alternative formulation of the optimal guidance problem with the practical constraints can be stated as follows:

$$P_2:\ \min_{u} J = \int_{t_0}^{t_f} a_M(t)^2 dt$$
$$= \int_{t_0}^{t_f} \frac{V_M^4 \sin^2\lambda_M(t)}{r(t)^2} u(t)^2 dt,$$
$$\text{subject to (1)-(5), (7), (12)-(13).}$$

For convenience, the above problem is called P_2 in this study. In the above optimal problem, the time-varying gain ($N = u$) can be regarded as a free variable to be optimized. In the next section, the MPPI control will be used to solve the problem P_2 instead of the problem P_1.

3. GENERIC MPPI FRAMEWORK

For the completeness of the paper, this section describes the general MPPI framework that is an essential ingredient of the proposed algorithm. The MPPI control is a probabilistic MPC method. As it is a sampling-based and non-derivative approach, it can be applied to the optimal control problems expressed by nonlinear and nonconvex equations without reformulation or convexification. The central idea of MPPI is to sample thousands of trajectories based on Monte-Carlo (MC) simulations using the stochastic system dynamic model. To get optimal control variations of the nominal control sequence, the trajectory

of each sample is then analyzed using specified cost functions. This process is repeated at each guidance cycle [45].

In the MPPI control framework, it is assumed that the discrete-time stochastic dynamical system is given as follows:

$$\begin{aligned} &x_{t+1} = F(x_t, v_t), \\ &v_t = u_t + \epsilon(t), \text{ where, } \epsilon(t) \sim N(0, \Sigma_u), \end{aligned} \tag{14}$$

where $t \in \{0, 1, 2, \cdots, T-1\}$ represents the index of discrete times, in which 0 and T represent the indexes of the start and final time, respectively. $x_t \in \mathbb{R}^n$ and $v_t \in \mathbb{R}^m$ denote the state and the stochastic control input vectors at time t. The stochastic control input v_t is defined as the element-wise sum of $u_t \in \mathbb{R}^m$ and $\epsilon(t) \in \mathbb{R}^m$, where u_t is the nominal control input vector, and $\epsilon(t)$ is the diffusion of control input vector given by the zero-mean Gaussian noise with a variance Σ_u.

Given a finite time-horizon $t \in \{0, 1, 2, \cdots, T-1\}$, we can define a sequence of control and noise inputs over some number of time steps T as

$$\begin{aligned} V &= \{v_0, v_1, \cdots, v_{T-1}\}, \\ U &= \{u_0, u_1, \cdots, u_{T-1}\}, \\ \mathcal{E} &= \{\epsilon_0, \epsilon_1, \cdots, \epsilon_{T-1}\}. \end{aligned} \tag{15}$$

The elements in V are the actual control input at time t and its distribution will be $v_t \sim N(u_t, \Sigma_u)$. If we define Ω as the sample space, the sequence is also a random variable defined as mapping $V : \Omega \to \Omega_V$ where $\Omega_V = \mathbb{R}^m \times \{0, 1, \cdots, T-1\}$. By changing the control input U, we can alter the probability distribution of V [38]. The ultimate goal of the stochastic optimal control is to find a control sequence $V \in \mathbb{R}^{m \times T}$, which minimizes the expectation of the cost function over all sampled trajectories produced by the dynamic model as given in (14).

There are interesting probability distributions created by V. First, there is $\mathbb{P}$ as the probability distribution in the uncontrolled system i.e., ($U \equiv 0$). Second, there is $\mathbb{Q}$ as the probability distribution in the controlled system. Lastly, $\mathbb{Q}^*$ denotes an abstract optimal distribution. The probability density functions for these distributions are denoted as $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{q}^*$, respectively [38]. The density functions of $\mathbf{p}$ and $\mathbf{q}$ have simple forms as follows [39,53]:

$$\begin{aligned} \mathbf{p}(V) &= \prod_{t=0}^{T-1} Z^{-1} \exp\left(-\frac{1}{2} v_t^T \Sigma_u^{-1} v_t\right), \\ \mathbf{q}(V) &= \prod_{t=0}^{T-1} Z^{-1} \exp\left(-\frac{1}{2} (v_t - u_t)^T \Sigma_u^{-1} (v_t - u_t)\right), \\ Z^{-1} &= \left((2\pi)^m |\Sigma_u|\right)^{\frac{1}{2}}. \end{aligned} \tag{16}$$

Given an initial state x_0, V can produce the system state trajectories through the system model F. Accordingly, we can define the space of all state trajectories as

$$\mathcal{G}_{x_0} : \Omega_V \to \Omega_\tau,$$

where τ denotes the state trajectories of space we can express $\Omega_\tau \subset \mathbb{R}^n \times \{0, \cdots, T-1\}$. Now, we can define a state-dependent cost function of the trajectory as follows:

$$C(x_1, x_2, \cdots, x_T) = \phi(x_T) + \sum_{t=1}^{T-1} q(x_t), \tag{17}$$

where ϕ is a terminal cost and q is an instantaneous state cost. To define the cost of the state trajectories, we define $S : \Omega_V \to \mathbb{R}^+$ over input sequences by the composition [38] $S = C \circ \mathcal{G}_{x_0}$. Now, let λ be a positive scalar variable. Then, we can express the free-energy of the control system as follows:

$$\mathcal{F}(V) = -\lambda \log\left(\mathbb{E}_{\mathbb{P}}\left[\exp\left(-\frac{1}{\lambda} S(V)\right)\right]\right). \tag{18}$$

In (18), the expectation can be rewritten with respect to $\mathbb{Q}$ by adding the likelihood ratio $\mathbf{p}(V)/\mathbf{q}(V)$ and its bound by Jensen's inequality as follows [40,54]:

$$\begin{aligned} \mathcal{F}(V) &= -\lambda \log\left(\mathbb{E}_{\mathbb{Q}}\left[\frac{\mathbf{p}(V)}{\mathbf{q}(V)} \exp\left(-\frac{1}{\lambda} S(V)\right)\right]\right) \\ &\leq -\lambda \mathbb{E}_{\mathbb{Q}}\left[\log\left(\frac{\mathbf{p}(V)}{\mathbf{q}(V)} \exp\left(-\frac{1}{\lambda} S(V)\right)\right)\right]. \end{aligned}$$

In the likelihood ratio term, $\mathbf{q}(V)$ is controllable by U as shown in (16). Thus, it is possible to find $\mathbf{q}^*(V)$ minimizing the cost function $S(V)$ by choosing appropriate U. If we make the likelihood ratio $\mathbf{p}(V)/\mathbf{q}(V)$ be proportional to the inverse of $\exp\left(-\frac{1}{\lambda} S(V)\right)$, the term inside the logarithm will be a constant. Accordingly, the cost over input sequence $S(V)$ is bounded by the free energy of system. When the bound is tight with an optimal control distribution, the optimal distribution $\mathbb{Q}^*$ can be defined with the probability density function $\mathbf{q}^*(V)$ as follows [38]:

$$\mathbf{q}^*(V) = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda} S(V)\right) \mathbf{p}(V), \tag{19}$$

where η is the normalizing factor. In other words, we can get the optimal distribution by minimizing a gap between the controlled distribution $\mathbb{Q}$ and the optimal control distribution $\mathbb{Q}^*$. We can use the definition of KL divergence to measure the closeness between the controlled distribution and the optimal one. Therefore, we can rewrite this optimal control problem as follows:

$$U^*(V) = \underset{U}{\operatorname{argmin}}\, \mathbb{D}_{KL}(\mathbb{Q}^* \,\|\, \mathbb{Q}),$$

where the KL divergence is expressed as

$$\begin{aligned} &\mathbb{D}_{KL}(\mathbb{Q}^* \,\|\, \mathbb{Q}) \\ &= \int_{\Omega_V} \mathbf{q}^*(V) \log\left(\frac{\mathbf{q}^*(V)}{\mathbf{q}(V)}\right) dV \\ &= \int_{\Omega_V} \mathbf{q}^*(V) \log\left(\frac{\mathbf{q}^*(V)\mathbf{p}(V)}{\mathbf{p}(V)\mathbf{q}(V)}\right) dV \end{aligned}$$

$$= \int_{\Omega_V} \mathbf{q}^*(V) \log\left(\frac{\mathbf{q}^*(V)}{\mathbf{p}(V)}\right) - \mathbf{q}^*(V) \log\left(\frac{\mathbf{q}(V)}{\mathbf{p}(V)}\right) dV. \tag{20}$$

From (16) and (19), it can be readily observed that the first term of (20) does not depend on U since $\mathbf{p}(V)$ and $\mathbf{q}^*(V)$ are independent of U. Therefore, U^* can be expressed as follows:

$$U^*(V) = \underset{U}{\operatorname{argmax}} \int_{\Omega_V} \mathbf{q}^*(V) \log\left(\frac{\mathbf{q}(V)}{\mathbf{p}(V)}\right) dV. \tag{21}$$

In (21), we can calculate the term $\mathbf{q}(V)/\mathbf{p}(V)$ by using (16). Then, we have

$$\begin{aligned} &\int_{\Omega_V} \mathbf{q}^*(V) \log\left(\frac{\mathbf{q}(V)}{\mathbf{p}(V)}\right) dV \\ &= \Sigma_{t=0}^{T-1} \left(-\frac{1}{2} u_t^T \Sigma_u^{-1} u_t + u_t^T \int \mathbf{q}^*(V) \Sigma_u^{-1} v_t dV\right). \end{aligned} \tag{22}$$

Equation (22) is a concave function with respect to u_t. Therefore, by calculating the gradient of (22) with respect to u_t, we can get the optimal u^* as follows:

$$u_t^* = \int \mathbf{q}^*(V)\, v_t dV. \tag{23}$$

As shown in (23), if we could sample from the optimal distributions $\mathbb{Q}^*$, we could compute u^* by averaging them. Since sampling from the optimal distribution $\mathbb{Q}^*$ is not available directly, the MPPI framework uses the importance sampling method. To this end, we rewrite (23) with respect to $\mathbb{Q}$ as follows:

$$u_t^* = \int \mathbf{q}(V) \frac{\mathbf{q}^*(V)\mathbf{p}(V)}{\mathbf{p}(V)\mathbf{q}(V)} v_t dV.$$

For convenience, the importance sampling weight $w(V)$ can be defined as

$$w(V) \triangleq \frac{\mathbf{q}^*(V)\mathbf{p}(V)}{\mathbf{p}(V)\mathbf{q}(V)}.$$

Finally, the optimal control input can be rewritten as the expectation with respect to $\mathbb{Q}$ as follows [38]:

$$u_t^* = \int \mathbf{q}(V) w(V) v_t dV = \mathbb{E}_{\mathbb{Q}}\left[w(V) v_t\right]. \tag{24}$$

Because V is equal to $U + \mathcal{E}$ and U is a given nominal control sequence, $w(V)$ becomes $w(\mathcal{E})$. The importance sampling weight $w(\mathcal{E})$ can be expressed by using (16) and (19) as follows:

$$\begin{aligned} w(\mathcal{E}) = \frac{1}{\eta} \exp\Bigg[&-\frac{1}{\lambda}\Bigg(S(U+\mathcal{E}) \\ &+ \lambda \sum_{t=0}^{T-1} \frac{1}{2} u_t \Sigma_u^{-1} \left(u_t + 2\epsilon_t\right)\Bigg)\Bigg]. \end{aligned} \tag{25}$$

In the above equation, η can be approximated using the Monte-Carlo estimate

$$\begin{aligned} \eta = \sum_{k=1}^{K} \exp\Bigg[&-\frac{1}{\lambda}\Bigg(S(U+\mathcal{E}_k) \\ &+ \lambda \sum_{t=0}^{T-1} \frac{1}{2} u_t \Sigma_u^{-1} \left(u_t + 2\epsilon_t^k\right)\Bigg)\Bigg], \end{aligned}$$

where k denotes the index of sampling and K is the number of random samples. From (24) and (25), we can have the optimal control input in the form of an iterative update law

$$u_t^{i+1} = u_t^i + \sum_{k=1}^{K} w(\mathcal{E}_k)\, \epsilon_t^k. \tag{26}$$

According to the previous studies [38,40,45,54], the state trajectory cost considering the likelihood ratio is denoted by $\tilde{S}$. We can rewrite (26) with $\tilde{S}$ as follows:

$$u_t^{i+1} = u_t^i + \frac{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda}\tilde{S}^k(V)\right) \epsilon_t^k}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda}\tilde{S}^k(V)\right)},$$

where

$$\tilde{S}^k(V) = S^k(V) + \lambda \sum_{t=0}^{T-1} \frac{1}{2} u_t^T \Sigma_u^{-1} \left(u_t + 2\epsilon_t^k\right).$$

Algorithm 1 summarizes the iterative MPPI control update law, in which the optimization and execution progress simultaneously. The state trajectory is optimized, and a single control u_0 is then executed with *SendToActuators()*. In the next loop, the trajectory is optimized based on the unexecuted portion of the previous control sequence for a warm start.

4. PROPOSED GUIDANCE ALGORITHM

In this section, the MPPI control framework is applied to the problem P_2. To this end, the free-final-time problem is first converted into a fixed-final-range problem by introducing the range as an independent variable. The cost functions for the MPPI control are designed in order to accomplish the prescribed guidance operational goal. Finally, the problem P_2 is solved by the MPPI control in conjunction with the receding horizon control scheme.

4.1. Fixed-final-range problem formulation

In general, the guidance problems, including the problem P_2, are inherently characterized as the free-final-time problem. It means the final time t_f will be varying depending on engagement scenarios. Moreover, for a fixed time step Δt, the length of the time sequence $t \in \{0, 1, 2, \cdots, T-1\}$ will also be different. It could be an obstacle for determining the terminal cost function for the MPPI control. Accordingly, to handle the free-final-time aspect of

Algorithm 1: Iterative MPPI control update law [38].

Given: F: Transition Model;
K: Number of samples;
T: Number of timesteps;
$(u_0, u_1, \ldots u_{T-1})$: Initial control sequence;
$\Sigma_u, \phi, q, \lambda$: Control hyper-parameters;
while *task not completed* **do**
  $x_0 \leftarrow$ `GetStateEstimate()`;
  for $k \leftarrow 0$ **to** $K-1$ **do**
    $x \leftarrow x_0$;
    Sample $\mathcal{E}^k = \{\epsilon_0^k, \epsilon_1^k, \cdots, \epsilon_{T-1}^k\}$;
    for $t \leftarrow 1$ **to** T **do**
      $x_t \leftarrow \mathrm{F}\left(x_{t-1}, u_{t-1} + \epsilon_{t-1}^k\right)$;
      $\tilde{S}\left(\mathcal{E}^k\right)$ += $q(x_t) + \frac{\lambda}{2} u_{t-1} \Sigma_u^{-1}\left(u_{t-1} + 2\epsilon_{t-1}^k\right)$;
    $\tilde{S}\left(\mathcal{E}^k\right)$ += $\phi(x_T)$;
  $\beta \leftarrow \min_k\left[\tilde{S}\left(\mathcal{E}^k\right)\right]$;
  $\eta \leftarrow \sum_{k=0}^{K-1} \exp\left(-\frac{1}{\lambda}\left(\tilde{S}\left(\mathcal{E}^k\right) - \beta\right)\right)$;
for $k \leftarrow 0$ **to** $K-1$ **do**
  $w\left(\mathcal{E}^k\right) \leftarrow \frac{1}{\eta} \exp\left(-\frac{1}{\lambda}\left(\tilde{S}\left(\mathcal{E}^k\right) - \beta\right)\right)$
for $t \leftarrow 0$ **to** $T-1$ **do**
  u_t += $\sum_{k=0}^{K-1} w\left(\mathcal{E}^k\right) \epsilon_t^k$;
SendToActuators(u_0);
for $t \leftarrow 1$ **to** $T-1$ **do**
  $u_{t-1} \leftarrow u_t$;
$u_{T-1} \leftarrow$ `Initialize`(u_{T-1});

the problem, a new independent variable is used in this study. From (2) with a given condition $\lambda_{M\max} < \pi/2$, it can be predicted that the range is monotonically decreasing during the engagement as follows:

$$\dot{r} = -V_M \cos\lambda_M < 0. \tag{27}$$

This fact implies that the range r can be used as the independent variable instead of time t. If the range is chosen as the independent variable, the final range can be always fixed as $r(t_f) = 0$. Additionally, the length of the range sequence can be fixed as well. These are beneficial for determining the terminal cost function in the MPPI control framework. To leverage these favorable characteristics, the nonlinear engagement kinematics for the problem P_2 can be rewritten with respect to r. From (2), (3), and (12), we have

$$\begin{aligned}\sigma' &= \frac{d\sigma}{dr} = \frac{d\sigma}{dt}\frac{dt}{dr} = \frac{\tan\lambda_M}{r},\\ \lambda_M' &= \frac{d\lambda_M}{dr} = \frac{d\lambda_M}{dt}\frac{dt}{dr} = \frac{\tan\lambda_M}{r}(u-1).\end{aligned} \tag{28}$$

Note that u in (28) becomes $N(r)$ in that case. If the state variables are defined as $x \triangleq [\sigma,\ \lambda_M]^T$ and x' represents dx/dr, the nonlinear engagement kinematics with respect to r can be rewritten as follows:

$$x' = f(x) + G(x)\,u, \tag{29}$$

where

$$f(x) \triangleq \begin{bmatrix} \frac{\tan\lambda_M}{r} \\ -\frac{\tan\lambda_M}{r} \end{bmatrix},\ G(x) \triangleq \begin{bmatrix} 0 \\ \frac{\tan\lambda_M}{r} \end{bmatrix}. \tag{30}$$

From (6), the final boundary conditions with respect to r can be rewritten as

$$\lambda_M(r_f) = 0,\ \sigma(r_f) = \gamma_f. \tag{31}$$

Likewise, the look angle and acceleration bound can also be rewritten in terms of r as

$$|\lambda_M(r)| \le \lambda_{M\max}, \tag{32}$$

$$\left| u(r)\,\frac{V_M(r)^2 \sin\lambda_M(r)}{r} \right| \le a_{M\max}. \tag{33}$$

Finally, the fixed-final-range problem can be formulated as follows:

$$\begin{aligned} P_3:\ \min_{u(r)} J &= \int_{r_0}^{r_f} a_M(r)^2\,(-dr) \\ &= \int_{r_0}^{r_f} R(r) u(r)^2\,(-dr), \\ &\text{subject to (29)-(33)}, \end{aligned} \tag{34}$$

where $R(r) = \left(V_M(r)^3 \tan\lambda_M(r) \sin\lambda_M(r)\right)/r^2$. For convenience, the above problem is called P_3 in this study.

4.2. Range discretization

As the MPPI control framework is suited in a discrete domain problem, a continuous domain problem P_3 needs to be transformed into a discrete domain problem. Assuming that the range r is discretized into N_h evenly distributed points as

$$r_h = r_0 + h\Delta r, \quad \forall h \in [0, N_h],$$

with

$$\Delta r \triangleq \frac{r_f - r_0}{N_h}, \tag{35}$$

where r_0 and r_f represent the initial and final range, respectively. h denotes the index of the discretized range steps as shown Fig. 2. Δr is a negative value due to the condition of (27).

The control input sequence to be optimized at h is given as

$$U_h = \{u_0, u_1, \cdots, u_{n_c}\}. \tag{36}$$

where n_c is the number of control steps. As the missile approaches the target, the range decreases. Therefore, the

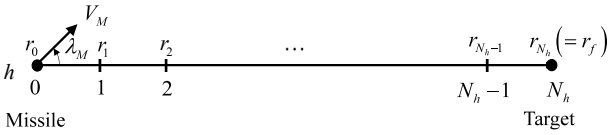


Fig. 2. Range discretization.

Table 1. The control steps according to the range step h.

h	n_c	Control steps
0	N_h-1	$\{0,1,\cdots,n_c-1,n_c\}$
1	N_h-2	$\{0,1,\cdots,n_c\}$
⋮	⋮	⋮
N_h-2	1	$\{0,n_c\}$
N_h-1	0	$\{n_c\}$

number of control steps n_c is also decreased. Table 1 explains the control steps according to the range step h.

In this study, the Euler method is used to perform the numerical integration of the system equation in (29) and (30).

$$\begin{aligned} x_{h+1} &= x_h + x_h' \Delta r \\ &= x_h + [f(x_h) + G(x_h)u_h]\Delta r,\ \forall h \in [0,\ n_c]. \end{aligned} \tag{37}$$

If we consider the diffusion adding directly through the control input with variance Σ_u, the deviation of the state $x_h' \Delta r$ in (37) can be expressed as follows:

$$\begin{aligned} x_h' \Delta r &= f(x_h)\Delta r + G(x_h)\left(u_h \Delta r + \sqrt{\Sigma_u} d\omega\right) \\ &= f(x_h)\Delta r + G(x_h)\left(u_h \Delta r + z\sqrt{\Sigma_u \Delta t}\right), \end{aligned} \tag{38}$$

where $d\omega \in \mathbb{R}^p$ is a Brownian disturbance. If z denotes a standard normal Gaussian random variable, we can express the Brownian disturbance with $z\sqrt{\Delta t}$ after time step Δt [39,40]. From (2), the time step Δt becomes

$$\Delta t = -\frac{\Delta r}{V_M \cos \lambda_M}. \tag{39}$$

By substituting (39) into (38), the deviation of the state after Δr can be rewritten as follows:

$$\begin{aligned} x_h' \Delta r &= f(x_h)\Delta r + G(x_h)\left(u_h \Delta r + z\sqrt{-\frac{\Sigma_u \Delta r}{V_M \cos \lambda_M}}\right) \\ &= \left[f(x_h) + G(x_h)\left(u_h + z\sqrt{-\frac{\Sigma_u}{\Delta r V_M \cos \lambda_M}}\right)\right]\Delta r \\ &= [f(x_h) + G(x_h)(u_h + \epsilon_h)]\Delta r \\ &= [f(x_h) + G(x_h)v_h]\Delta r. \end{aligned}$$

We can define the discretized diffusion of the control input ϵ_h in (15) as follows:

$$\epsilon_h = z\sqrt{-\frac{\Sigma_u}{\Delta r V_M \cos \lambda_M}}.$$

As given in (38), we can see the system dynamics to be solved is affine in control and Brownian disturbance. Accordingly, by referring [40], we can determine the value of λ in (18) to satisfy the linear PDE assumption $BB^T = \lambda G R^{-1} G^T$ in the state trajectory value equation, where B denotes the diffusion matrix of the Brownian disturbance. To get the expression for B, we rewrite the dynamics in the following form:

$$x_h' \Delta r = f(x_h)\Delta r + G(x_h)u_h \Delta r + B(x_h)\, d\omega. \tag{40}$$

Here, B is equal to $G\sqrt{\Sigma_u}$ as shown in (38) and (40). By substituting B into the assumption, we can determine the variable λ_h at r_h as follows:

$$\begin{aligned} BB^T &= G\sqrt{\Sigma_u}\left(G\sqrt{\Sigma_u}\right)^T = \lambda G R^{-1} G^T, \\ \lambda_h &= R_h \Sigma_u \\ &= \frac{V_M^3 \tan \lambda_M(r_h) \sin \lambda_M(r_h)}{r_h^2}\Sigma_u, \end{aligned}$$

where R_h is the weight of control u_h, which is defined in (34) [40].

4.3. Cost function design and normalization

This section describes the design procedure of the cost function to apply to the guidance problem P_3 in the MPPI framework. The guidance objective is to guide the missile to the target while satisfying the desired impact angle with the look angle and acceleration constraints.

In (17), the cost of the state trajectory is composed of the terminal and instantaneous state costs. In other studies [39,40], the instantaneous state costs are also called running costs. It is worth noting that the control costs do not appear in the expression of (17). This means that the MPPI framework optimizes the controls without ever referring to the controlled dynamics of the system or the control costs. This is because the controls in the MPPI framework are linked together in passive dynamics through the assumption of noise and controls. The optimization proceeds considering the control matrix G and the control costs, but only implicitly [40]. Accordingly, we can design the terminal cost function for the desired impact angle and the running cost ones for the seeker's FOV limit and the maximum acceleration bound as follows:

$$\begin{aligned} &\phi(x_{n_c+1}) = (x_{n_c+1} - x_f)^2, \\ &q(x_h) \triangleq \begin{cases} q_{\max}, & \text{if } |x_h| > x_{\max}, \\ (x_h/x_{\max})^2, & \text{otherwise}, \end{cases} \end{aligned} \tag{41}$$

where x_{n_c+1} is the terminal state and x_f is its desired state. In (41), x_h is the instantaneous state at h, and $x_{\max}$ is the maximum limited value for the state x. $q_{\max}$ is the user-defined constant cost value when x_h is over $x_{\max}$. Finally,

we can express the cost vector $\boldsymbol{c}^k(x_0, x_1, \cdots, x_{n_c})$ of the k-th state trajectory for the guidance problem P_3 as follows:

$$\boldsymbol{c}^k(x_0, x_1, \cdots, x_{n_c}) = \begin{bmatrix} \Phi^k \\ \tilde{Q}^k \end{bmatrix} = \begin{bmatrix} \phi(\lambda_M(r_f)) + \phi(\sigma(r_f)) \\ \sum_{i=0}^{n_c} \left(w_{\lambda_M} q(\lambda_M(r_i)) + w_{a_M} q(a_M(r_i)) \right. \\ \left. + \frac{\lambda_i^k}{2} u(r_i) \Sigma_u^{-1} \{ u(r_i) + 2\epsilon(r_i)^k \} \right) \end{bmatrix}, \tag{42}$$

where Φ^k denotes the terminal cost and $\tilde{Q}^k$ is the sum of the instantaneous running cost with the likelihood ratio for all control steps from 0 to n_c. w_{λ_M} and w_{a_M} are the normalized weights of each instantaneous running cost $q(\lambda_M(r_i))$ and $q(a_M(r_i))$. As shown in (42), the cost values are given by the function of the state variables which have significantly different physical quantities. Accordingly, normalizing each cost value is required to reduce variability due to individual physical characteristics. To this end, we define the normalization function as follows:

$$normalize(\boldsymbol{c}_i) = \frac{\boldsymbol{c}_i - c_{\min}}{c_{\max} - c_{\min}} \times SF, \text{ for } i \in [1, 2],$$

where $\boldsymbol{c}_i \in \mathbb{R}^K$ denotes the cost vector element for the K state trajectories. $c_{\min}$ and $c_{\max}$ are the minimum and maximum values of $\boldsymbol{c}_i$, respectively. SF denotes the scale factor of the normalized values. To add the terminal and running costs, the cost weight vector w_c is defined. We also normalize w_c as follows:

$$w_c = \begin{bmatrix} w_\Phi \\ w_{\lambda_M} + w_{a_M} \end{bmatrix} \frac{1}{w_\Phi + w_{\lambda_M} + w_{a_M}}, \tag{43}$$

where w_Φ is the weight of the terminal cost vector Φ. With the normalized cost weight vector, we can construct the cost vector $\tilde{S} \in \mathbb{R}^K$ from the K state trajectories as follows:

$$\tilde{S} = w_c^T \boldsymbol{c}_n = w_c^T \begin{bmatrix} \Phi_n \\ \tilde{Q}_n \end{bmatrix},$$

where Φ_n and $\tilde{Q}_n$ are the normalized $\Phi \in \mathbb{R}^K$ and $\tilde{Q} \in \mathbb{R}^K$, which are the cost vectors for K state trajectories.

4.4. MPPI-based computational guidance algorithm

Algorithm 2 describes the details of the MPPI-based guidance algorithm. The number of range steps and samples need to be determined to implement the algorithm. Additionally, we need to design the cost functions and determine cost weight parameters for the guidance problem. The variance of the input noise and the initial control sequences are the design parameters affecting the outcomes of the guidance solution. In the proposed algorithm, at every range step, we take the cost values of $n_c \times K$ sampled state trajectories with the initial state and controls. In the fixed-final-range problem, n_c is a variable updated as explained in Table 1. The K state trajectories are generated with the noise ϵ_i^k. After adding the likelihood ratio to the running cost, we obtain K normalized costs of the state trajectory. Through the procedure of calculating the minimum value of the state trajectory costs, we can optimize the current control sequence according to the MPPI algorithm. Additionally, the stochastic nature of the sampling procedure can lead to jittering in the resulting control, so we adopt a Savitzky-Golay filter (SGF) to smooth the output control sequence. Most control systems, including

Algorithm 2: MPPI-based guidance algorithm for impact angle control.

Given $\mathtt{f}, \mathtt{G}$: Transition Model;
K: Number of samples;
N_h: Number of range steps;
$U_0 = \{u_0, u_1, \ldots u_{N_h-1}\}$: Initial control sequence;
$\Delta r, x_0, \Sigma_u$: Initial system variables;
ϕ, q: Cost functions;
$w_\Phi, w_{\lambda_M}, w_{a_M}$: Cost weights;
SGF: Savitzky-Golay Filter;
for $h \leftarrow 0$ **to** $N_h - 1$ **do**
  $n_c = N_h - h - 1$;
  for $k \leftarrow 0$ **to** $K - 1$ **do**
    $x_h \leftarrow x_0$;
    $U_h \leftarrow U_0$;
    $\tilde{Q}^k, \mathcal{L} \leftarrow 0$;
    Sample $\mathcal{E}^k = \{\epsilon_0^k, \epsilon_1^k, \cdots, \epsilon_{n_c}^k\}$;
    for $i \leftarrow 0$ **to** n_c **do**
      $r_i \leftarrow r_0 + (h+i)\Delta r$;
      $v_i \leftarrow u_i + \epsilon_i^k$;
      $x_{i+1} \leftarrow x_i + \{\mathtt{f}(x_i) + \mathtt{G}(x_i) v_i\} \Delta r$;
      $\mathcal{L} \leftarrow \frac{\lambda_i^k}{2} u_i \Sigma_u^{-1} \{u_i + 2\epsilon_i^k\}$;
      $\tilde{Q}^k \mathrel{+}= w_{\lambda_M} q(\lambda_{M,i}) + w_{a_M} q(a_{M,i}) + \mathcal{L}$;
    $\Phi^k \leftarrow \phi(\lambda_M(r_f)) + \phi(\sigma(r_f))$;
    $\boldsymbol{c}^k = [\Phi^k \ \tilde{Q}^k]^T$;
  $\Phi_n \leftarrow \mathtt{normalize}(\Phi)$;
  $\tilde{Q}_n \leftarrow \mathtt{normalize}(\tilde{Q})$;
  $\tilde{S} \leftarrow w_c^T \boldsymbol{c}_n$;
  $\beta \leftarrow \min_k[\tilde{S}]$;
  for $i \leftarrow 0$ **to** n_c **do**
    $\eta_i \leftarrow \sum_{k=0}^{K-1} \exp\left(-\frac{1}{\lambda_i^k}\left(\tilde{S}^k - \beta\right)\right)$;
    for $k \leftarrow 0$ **to** $K - 1$ **do**
      $w_i^k \leftarrow \frac{1}{\eta_i} \exp\left(-\frac{1}{\lambda_i^k}\left(\tilde{S}^k - \beta\right)\right)$
    $u_i \mathrel{+}= \sum_{k=0}^{K-1} w_i^k \epsilon_i^k$;
  $U_{\text{sys}} \leftarrow \mathtt{SGF}(U_h)$;
  $x_{\text{sys,h+1}} \leftarrow$
   $x_{\text{sys,h}} + \{\mathtt{f}(x_h) + \mathtt{G}(x_h)\, \mathtt{clipping}(u_{\text{sys},0})\} \Delta r$;
  for $i \leftarrow 0$ **to** $n_c - 1$ **do**
    $u_i \leftarrow u_{sys,i+1}$;
  $x_0 \leftarrow x_{\text{sys,h+1}}$;

missiles, have acceleration limits that the controller must take into account. The method used in this study is to design the cost function and clipping function that restricts system input $u_{sys,0}$ to remain within an allowable acceleration range. For the next iteration, the initial state x_0 is updated with the system state $x_{sys,h+1}$, and the remained portion of the optimized system control input vector U_{sys} is assigned to U_0 for the warm start.

5. SIMULATION RESULTS

In this section, numerical simulations are performed to investigate the characteristics and feasibility of the proposed algorithm. First, we verify that the proposed MPPI control procedure is valid through a well-known energy-minimization guidance problem. We then examine the effectiveness of the proposed algorithm for different impact angle constraints with the look angle and acceleration constraints. We also perform the Monte-Carlo simulation and computation time analysis. Finally, the simulation results obtained under the proposed algorithm and other algorithms are compared to investigate the characteristics of the proposed method.

As a baseline engagement scenario in the simulation study, we consider a ground-launched scenario against a stationary target. It is assumed that the missile is flying with a constant velocity of 200 m/s, and the initial flight path angle is 15 deg. The number of sample trajectories is chosen as $K = 100$. This study implements the MPPI-based guidance algorithm by utilizing GPU on a desktop with an Intel Core i5-8265U CPU@1.60 GHz device and 8 GB RAM. The NVIDIA GeForce MX250 GPU and MATLAB R2018b software are used for parallel processing and interface for GPU device memory.

5.1. Validity of MPPI control procedure

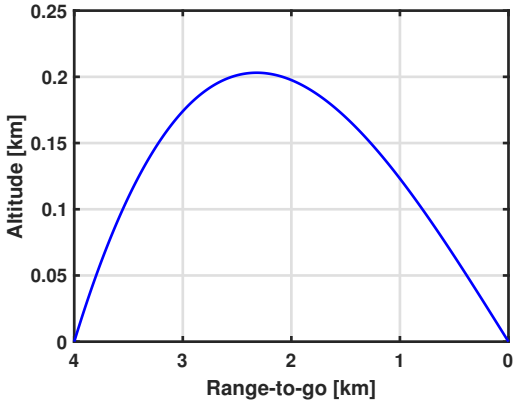
According to reference [1,55,56], it has been well-known that the PN guidance with $N = 3$ is the optimal solution for the guidance problem with the control energy minimization under certain assumptions such as a constant closing velocity, a near-collision course, and more. This fact is also valid even in the nonlinear problem if the impact angle is not constrained. Therefore, in this simulation, the solution obtained by the proposed algorithm for the same guidance problem will be compared to the known optimal solution. This is to examine the validity of the proposed algorithm to the guidance problems.

First, to confirm the fact that $N = 3$ is the optimal solution under the energy minimization condition, the guidance problem P_3 without the impact angle, look angle, and acceleration constraints is solved by a nonlinear optimization algorithm called the interior point method [57]. In this simulation, it is assumed that a stationary target is located 4 km away from the launch point. For the discretization in (35), we set $N_h = 400$, $r_f = 0$ m, and $r_0 = 4000$ m. Fig. 3 shows the optimization results obtained by the nonlinear optimization algorithm. From Fig. 3(c), we can readily observe that the optimized PN gain converges to 3. This result is consistent with the previously known result. In the terminal phase of Fig. 3(c), N goes to a large value and converges to zero. As shown in (11), a_M depends on the variable λ_M, r, and $N(=u)$. Because a_M is not limited, the system can use the large value of N when λ_M is very small. Additionally, when the range-to-go approaches zero, a_M goes to the infinite value, so N has to be zero to minimize the energy consumption. Next, the same guidance problem is solved by the proposed MPPI control procedure. In this case, the initial control input sequence in (36) is set to constant values 5, 7, or 9 across the range steps for each simulation. Additionally, the cost weight vector (i.e., the design parameters) in (43) is chosen as $w_c^T = [1, 0]$ where w_{λ_M} and w_{a_M} are 0. The variance Σ_u in (38) is selected as 0.3. Fig. 4 provides the guidance results determined by the proposed algorithm and the nonlinear optimization algorithm. To be more specific, Figs. 4(a), 4(b), and 4(c) show the look angle profile, guidance command profile, and PN gain profile, respectively. As shown in Fig. 4(c), it can be known that the optimization results obtained by the proposed algorithms converge to the optimal solutions determined by the nonlinear optimization algorithm, regardless of the initial control input sequence setting. Fig. 4(a) shows that the look angle also gradually converges to 0 from the initial value of 15 deg as the range-to-go approaches 0. It means that the intercept condition for a stationary target can be successfully achieved by the proposed algorithm. The results obtained indicate that the proposed MPPI control procedure is applicable to solving the nonlinear optimal guidance problems.

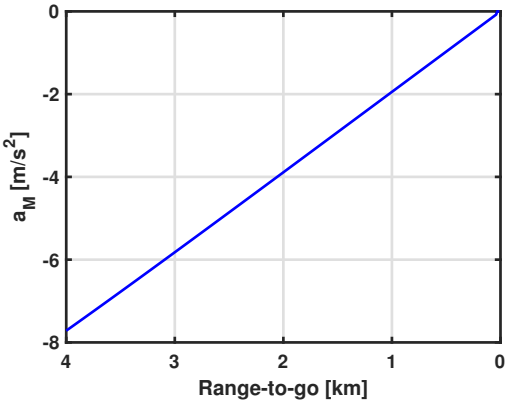
5.2. Different impact angles with acceleration constraint

We apply the proposed guidance algorithm to the nonlinear constrained impact angle guidance problem. In this simulation, it is assumed that a stationary target is located 5 km away from the launch point. We set the desired impact angles γ_f from -30 to -150 deg at intervals of -15 deg. To test various desired impact angles, we set only the acceleration bound $a_{M\max}$ as 45 m/s^2 without the FOV limits. Accordingly, we set $w_c^T = [0.91, 0.09]$ with $w_{\lambda_M} = 0$ in (43). In addition, $q_{\max}$ in (41) is set to 10^3 and the variance of control input (i.e., Σ_u) is set to 0.01. The initial control input sequence U_0 is chosen as shown in Fig. 5(e) for all cases.

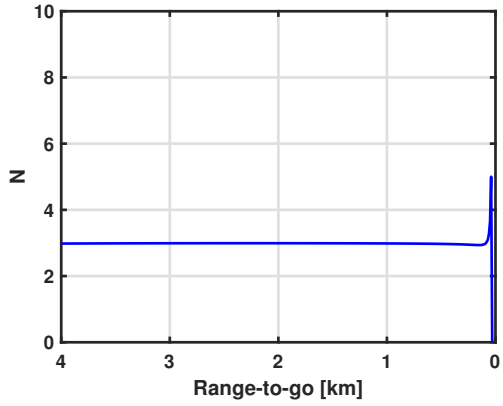
Fig. 5 shows the simulation results of each desired impact angle case. In Fig. 5(a), we can readily see that the final flight path angles approach their desired impact angles from the initial 15 deg. The errors of the impact angles are recorded under 0.001 deg for all cases. Additionally, as shown in Fig. 5(b), the final look angle of each case converges to 0, which means the missile successfully



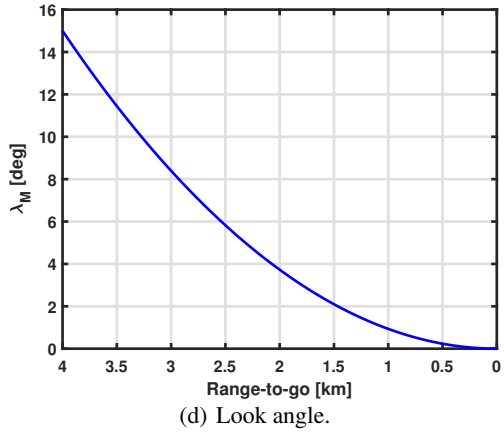
(a) Intercept trajectory.



(b) Guidance command.

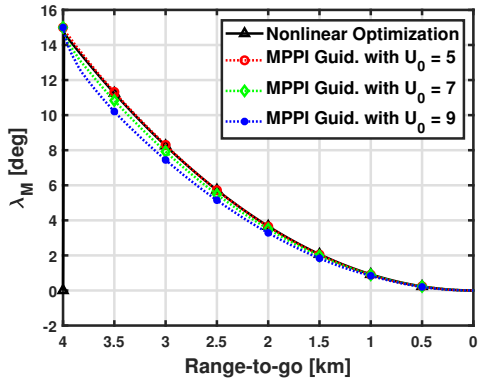


(c) PN gain.

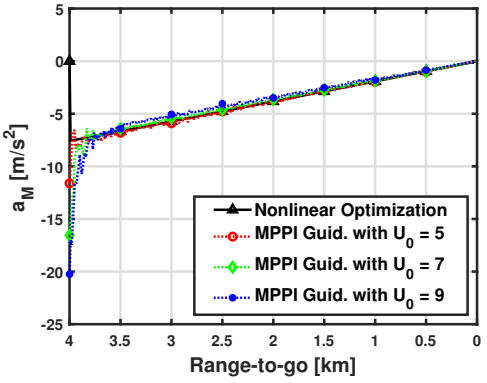


(d) Look angle.

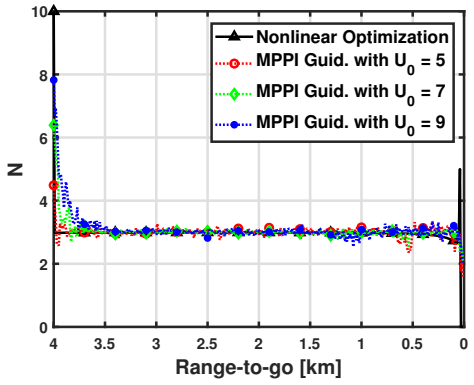
Fig. 3. The energy optimal guidance solution obtained by a nonlinear optimization algorithm.



(a) Look angle.



(b) Guidance command.



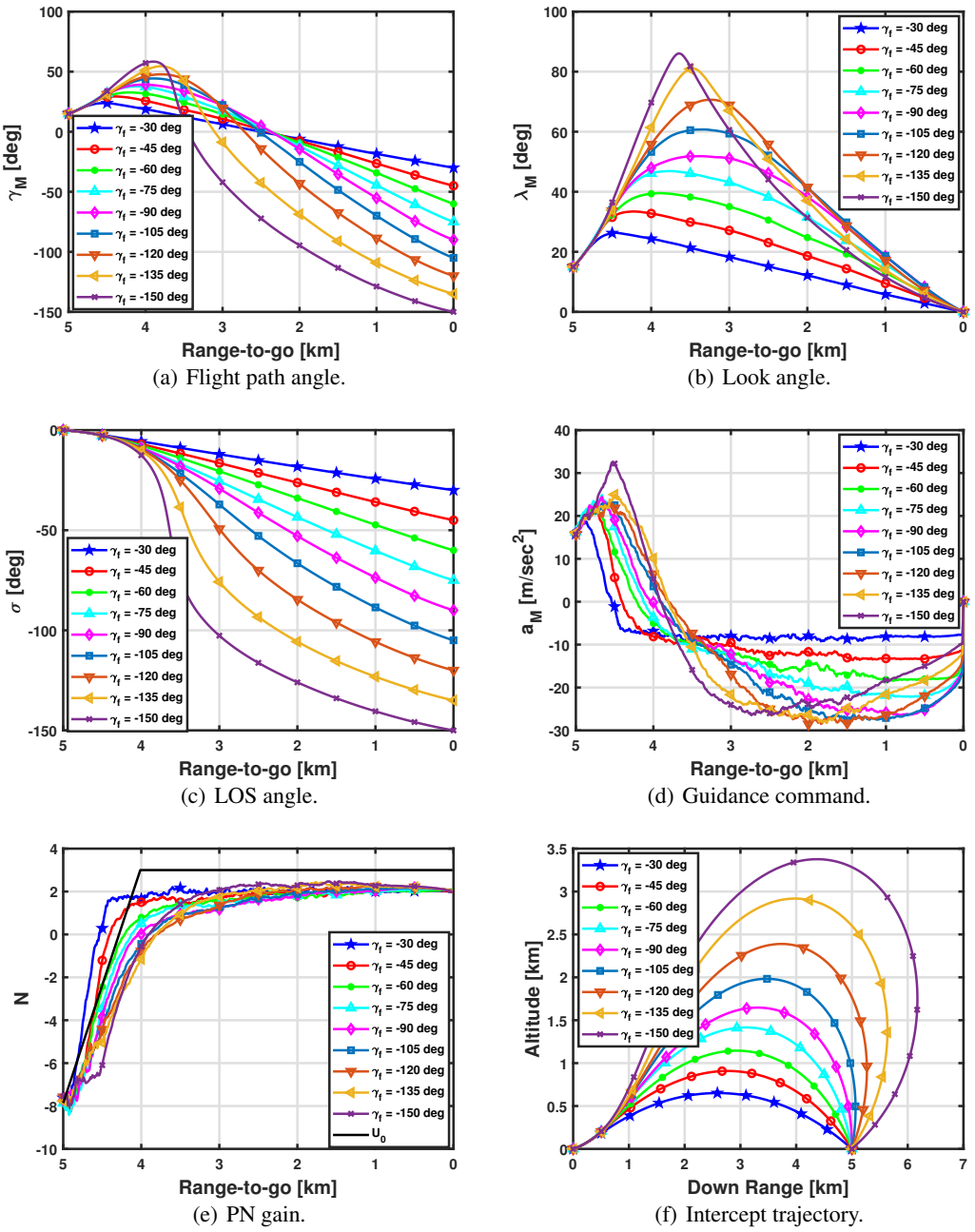
(c) PN gain.

Fig. 4. The energy optimal guidance solution obtained by the proposed algorithm.

intercepts the target. Fig. 5(f) shows the resultant trajectories of each case. Fig. 5(d) depicts the acceleration used to control the flight path angle of the missile, which are under the value of $a_{M\max}$.

5.3. Different impact angles with look angle and acceleration constraints

In this simulation, the proposed MPPI guidance algorithm is tested with the nonlinear constrained optimal guidance problems as given in (34). The same initial engagement condition used in Subsection 5.2 is considered in this simulation. We set $\lambda_{M\max} = 35$ deg and $a_{M\max} = 50$ m/s^2, re-

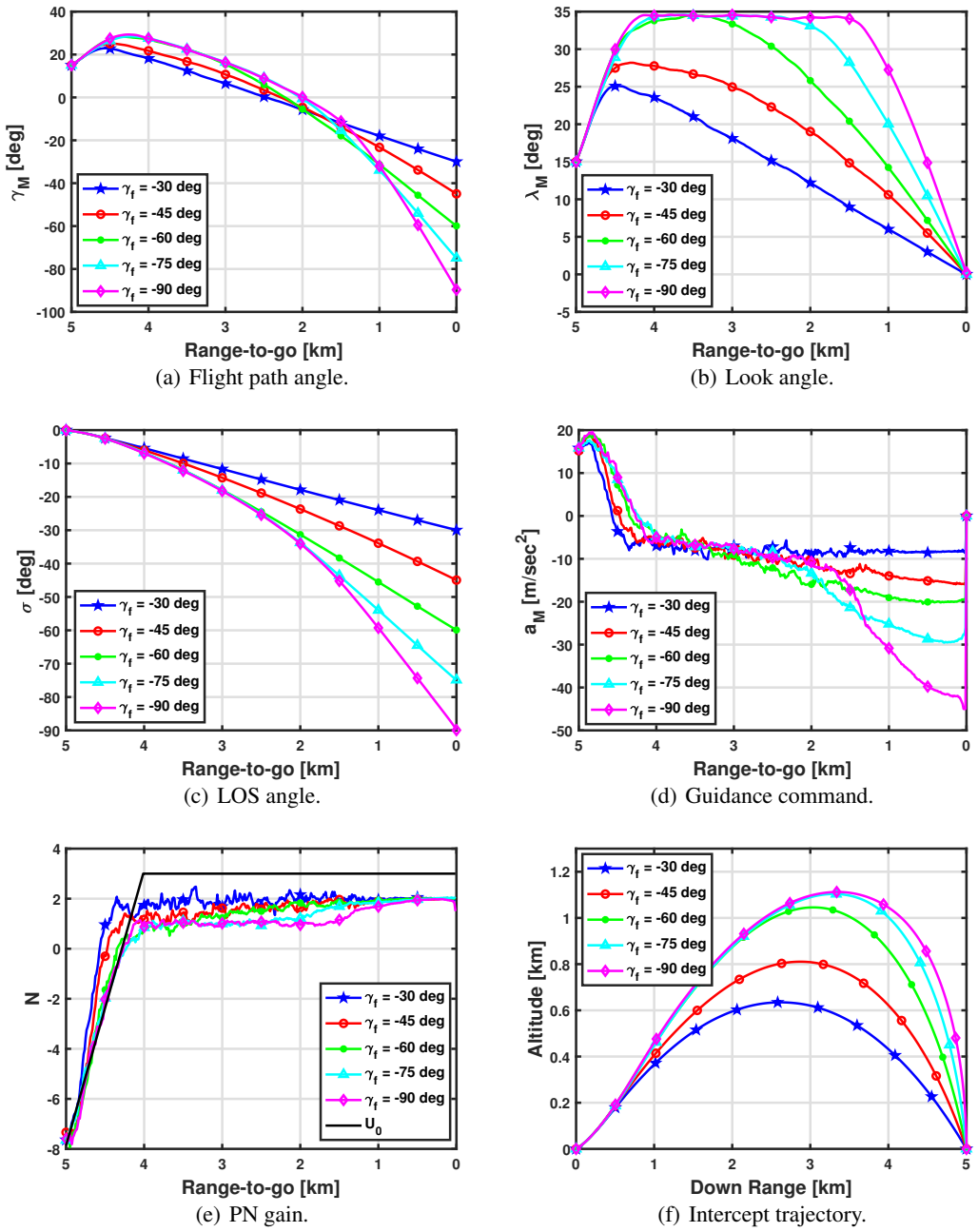


(a) Flight path angle. (b) Look angle. (c) LOS angle. (d) Guidance command. (e) PN gain. (f) Intercept trajectory.

Fig. 5. MPPI simulation results for each desired impact angle with the acceleration constraint.

spectively. The desired impact angles are chosen as $\gamma_f = -30, -45, -60, -75$, and -90 deg. Additionally, $q_{\max}$ in (41) is set to 10^3 and the variance of control input (i.e., Σ_u) is set to 0.01. The same initial control input sequence U_0 used in the previous section is considered as shown in Fig. 6(e). In (43), we set w_Φ, w_{λ_M}, and w_{a_M} as 0.67, 0.27, and 0.06, respectively. Accordingly, we set the normalized cost weight vector $w_c^T = [0.67, 0.33]$ and the normalized weight values of the running costs $w_{\lambda_M} = 0.82$, and $w_{a_M} = 0.18$.

Fig. 6 provides the simulation results for each impact angle constraint under the proposed algorithm. As shown in Fig. 6(a), it can be readily observed that the desired impact angle constraints can be achieved in all cases. Figs. 6(b) and 6(d) show that the look angle limit and the acceleration constraint are all met under the proposed algorithm. Additionally, the results in Fig. 6(b) indicate that the look angles converge to 0 as the missile approaches to the stationary target. It implies that the proposed guidance algorithm can successfully satisfy the intercept condition in all cases as well. The acceleration profiles are plotted in Fig. 6(d). It can be observed that the acceleration



(a) Flight path angle. (b) Look angle.

(c) LOS angle. (d) Guidance command.

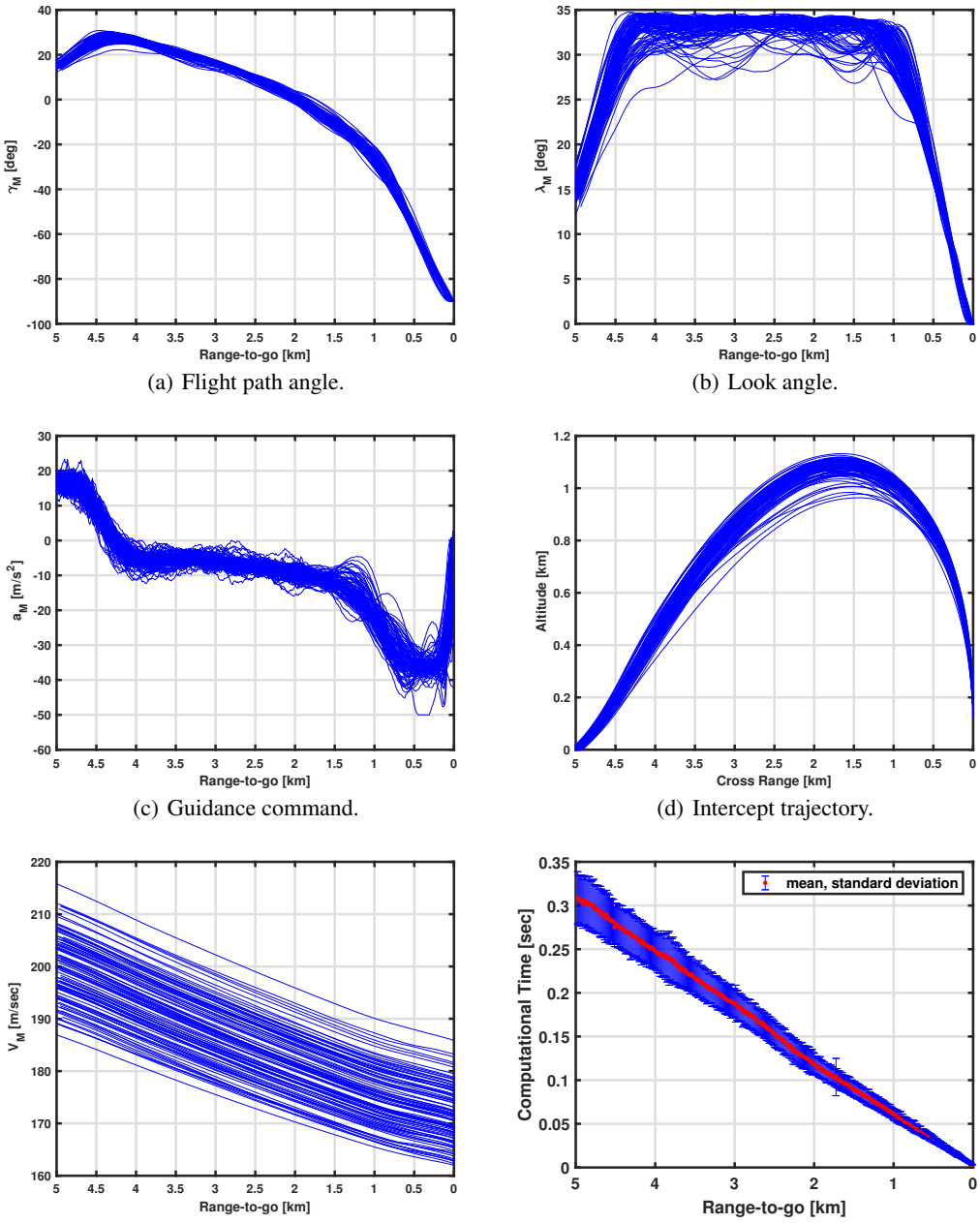
(e) PN gain. (f) Intercept trajectory.

Fig. 6. MPPI simulation results for each desired impact angle with the acceleration and look angle constraints.

demands increase as the desired impact angles increase. The corresponding PN gain profiles for different γ_f are depicted in Fig. 6(e). They have similar patterns: starting with negative values, gradually increasing in the middle, and converging to a specific value (i.e., 2) at the end. It is worth pointing out that these gains profiles are optimized onboard in a closed-loop fashion without relying on any optimization solvers in the proposed algorithm. The simulation results obtained indicate that the proposed computational guidance method can be applied to the challenging issues of the impact angle control problems with the practical constraints.

5.4. Monte-Carlo simulation and computation time analysis

In this subsection, we carry out the Monte-Carlo simulation to verify the robustness of the proposed algorithm in the presence of model uncertainties and disturbances. The computation time for the proposed algorithm is also analyzed. For the Monte-Carlo simulation, we generate a total of 100 simulation cases by disturbing the missile's initial relative range, initial velocity, and initial flight path an-



(a) Flight path angle. (b) Look angle.

(c) Guidance command. (d) Intercept trajectory.

(e) Velocity. (f) Computation time.

Fig. 7. Monte-Carlo simulation results for $\gamma_f = -90$ deg with the acceleration and look angle constraints.

gle using the normal distribution. Each variable's 3-sigma (standard deviation) is chosen as 2% (100 m), 10% (20 m/s), and 20% (3 deg), respectively. Furthermore, the effect of aerodynamic drag on missile velocity is considered. Additionally, when determining the guidance command, the drag coefficient is randomly altered by imposing 10% uncertainty using the normal distribution to consider an aerodynamic error or disturbance. The time-varying velocity is governed by

$$\dot{V}_M = -D/m,$$

where m is the mass, and $D = 0.5\rho V_M^2 S_{\text{ref}} C_D$ represents aerodynamic drag with the reference area S_{ref}, drag coefficient C_D, and the air density ρ. In this simulation, the desired impact angle is set to -90 deg, and the other simulation conditions are identical to those in Subsection 5.3.

As shown in Fig. 7, it can be observed that the desired impact angles can be achieved even in the presence of uncertainties. The look angle and acceleration limits are all satisfied as well. The resultant mean and standard deviation values of the impact angles are recorded as -90.03 deg and 0.05 deg, respectively.

To examine the computational efficiency of the proposed method, the computation time is measured at each guidance cycle. Fig. 7(f) shows the resultant mean and standard deviation values of the computation time for the Monte-Carlo simulation. Considering the standard deviation, the computation times are below 0.35 sec and monotonically decrease as the range-to-go decreases. In [24, 58], the convex (CVX) optimization-based computational guidance law was implemented for real-time guidance, and the computation time for the CVX-based method was recorded as 0.5 sec. Thus, the results show that the proposed MPPI-based approach is comparable in terms of computation time to the CVX-based approach, considered the state-of-the-art method in the computational guidance approach. In general, the computational efficiency considerably depends on the hardware and software specifications. Therefore, the computational time will be further reduced if the proposed algorithm is implemented in an embedded program language such as C with CUDA library with more powerful computing devices and software. As examples, it is worth pointing out that the real-time performance of MPPI framework has already been verified in [40,42,44,45,54] . This computational efficiency of the proposed approach demonstrates its potential for real-time feasibility.

5.5. Comparison with other approaches

In this subsection, the proposed algorithm is compared with two other methods: the biased PN guidance (BPN) [22] as an analytic method and the CVX approach as a computational method [24]. In the CVX-based computational guidance, the software called MOSEK [59] (i.e., one of optimization solvers) is utilized to solve the given guidance problem. In [22,24], each method is applied to address impact angle control guidance (IACG) problems considering the seeker's look angle and acceleration limits.

Fig. 8 provides the simulation results for the comparison study. As shown in Figs. 8(a), 8(b), and 8(d), the look angle limit, acceleration limit, and impact angle constraints are all satisfied in all methods. However, there are several distinctive features. In Fig. 8(e), it can be observed that the PN gain varies with time in the computational guidance methods (i.e., the MPPI-based method and the CVX-based method). However, the PN gain is constant at 3 under the BPN guidance. This is because the BPN guidance introduces a biased command term to satisfy the constraints. According to reference [22], the guidance command for the BPN law is given by

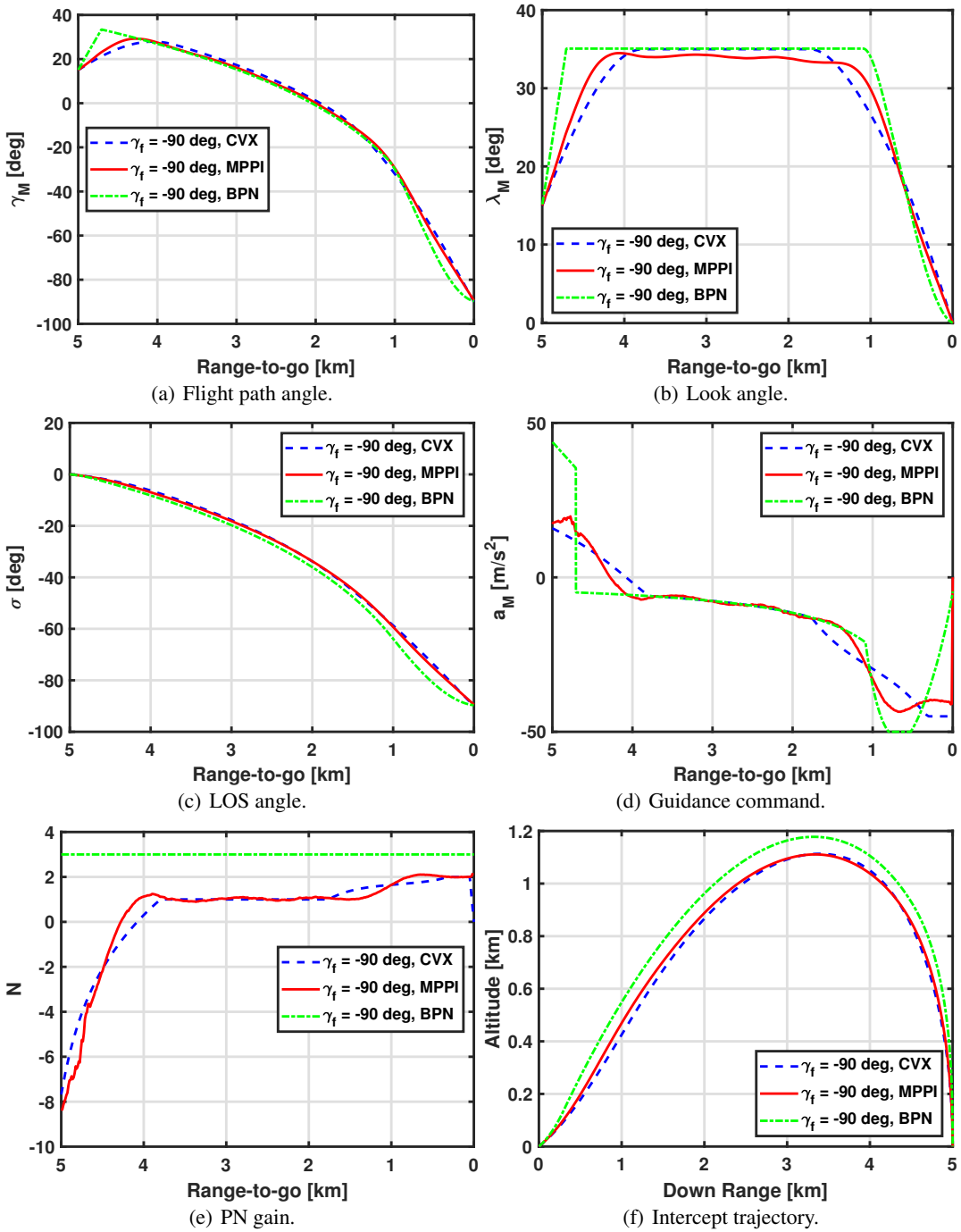
$$\dot{\gamma} = N\dot{\lambda} + b,$$

where b represents the biased command that can be calculated analytically in a state feedback form with fixed guidance parameters. Thus, this guidance law would be vulnerable to unanticipated conditions or uncertainties. It also requires a time-to-go estimation, which is difficult to obtain precisely. Moreover, as shown in Fig. 8(d), abrupt guidance command changes due to the reconfiguration of the guidance command can be observed in the BPN guidance. An abrupt command change is unfavorable for the autopilot because it may cause autopilot instability due to the actuator's slew rate limit.

The performance index values (i.e., the energy costs) of the BPN guidance, CVX-based method, and MPPI-based method are recorded as $14,346\ \mathrm{m}^2/\mathrm{sec}^3$, $11,662\ \mathrm{m}^2/\mathrm{sec}^3$, and $12,185\ \mathrm{m}^2/\mathrm{sec}^3$, respectively. The energy costs of the BPN method and MPPI-based method are about 23% and 4% more than that of the CVX-based approach, respectively. As the BPN guidance has not been devised using the optimal control approach, it has a high energy cost. As shown in Fig. 8, the MPPI-based approach and the CVX-based approach have similar patterns. It can also be observed that the MPPI-based approach and CVX-based approach have similar energy costs compared to the BPN guidance. It implies that the MPPI-based approach has almost equivalent performance to the CVX-based computational guidance. In other words, the MPPI-based guidance can provide a near-optimal solution because the CVX-based approach can be considered as the optimal solution. However, compared to the CVX-based guidance, it is worth noting that the solution is determined onboard without any dedicated optimization solvers in the MPPI-based guidance. It can be regarded as a benefit for implementation. Last but not least, the MPPI-based guidance can also provide more degrees of freedom in selecting the performance index through several design parameters (i.e., the selections of the running cost and terminal cost functions are flexible) compared to the CVX-based guidance.

6. CONCLUSIONS

In this study, a new computational guidance algorithm was proposed based on the model predictive path integral (MPPI) control framework, which is an approach substantially different from the previous studies. We also examined the feasibility and applicability of the MPPI control framework in the practical missile guidance problems with the desired impact angle, seeker's look angle, and maximum acceleration constraints. In the proposed algorithm, the guidance command was assumed as the form of the proportional navigation (PN) guidance with a time-varying navigation gain to be optimized by the MPPI control. A running cost function and a terminal cost function for the guidance problem were proposed to achieve the given path constraints, terminal conditions, and performance index to be minimized. Extensive numerical simulation verified the effectiveness and features of the proposed algorithm compared to other approaches. The results obtained indicate that the proposed algorithm can be



(a) Flight path angle. (b) Look angle. (c) LOS angle. (d) Guidance command. (e) PN gain. (f) Intercept trajectory.

Fig. 8. Comparison results between MPPI, CVX, and BPN methods for $\gamma_f = -90$ deg.

applicable to the challenge issues of the guidance problems with the practical constraints. Additionally, it has been shown that the proposed algorithm has the potential to complement other computational guidance approaches because of its favorable characteristics. The proposed algorithm allows us to efficiently solve nonlinear guidance problems without the effort of convexification and linearization in a closed-loop fashion. It does not require any dedicated solvers for optimization problems. Additionally, the selection of performance index is more flexible than other approaches.

CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

[1] P. Zarchan, *Tactical and Strategic Missile Guidance*, American Institute of Aeronautics and Astronautics, Inc., 2012.

[2] B. S. Kim, J. G. Lee, and H. S. Han, "Biased PNG law for impact with angular constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 277-288, 1998.

[3] K. S. Erer and O. Merttopçuoglu, "Indirect impact-angle-control against stationary targets using biased pure proportional navigation," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 700-704, 2012.

[4] C.-H. Lee, T.-H. Kim, and M.-J. Tahk, "Interception angle control guidance using proportional navigation with error feedback," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 5, pp. 1556-1561, 2013.

[5] A. Ratnoo and D. Ghose, "Impact angle constrained interception of stationary targets," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1817-1822, 2008.

[6] R. Tekin and K. S. Erer, "Switched-gain guidance for impact angle control under physical constraints," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 205-216, 2015.

[7] S. He, C.-H. Lee, H.-S. Shin, and A. Tsourdos, *Optimal Guidance and Its Applications in Missiles and UAVs*, Springer, 2020.

[8] C.-K. Ryoo, H.-J. Cho, and M.-J. Tahk, "Optimal guidance laws with terminal impact angle constraint," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 724-732, 2005.

[9] C.-K. Ryoo, H. Cho, and M.-J. Tahk, "Time-to-go weighted optimal guidance with impact angle constraints," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 483-492, 2006.

[10] C.-H. Lee, M.-J. Tahk, and J.-I. Lee, "Generalized formulation of weighted optimal guidance laws with impact angle constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1317-1322, 2013.

[11] C.-H. Lee and M.-Y. Ryu, "Practical generalized optimal guidance law with impact angle constraint," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 10, pp. 3790-3809, 2019.

[12] C.-H. Lee and M.-G. Seo, "New insights into guidance laws with terminal angle constraints," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 8, pp. 1832-1837, 2018.

[13] C.-H. Lee, T.-H. Kim, and M.-J. Tahk, "Design of impact angle control guidance laws via high-performance sliding mode control," *Proc. of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 227, no. 2, pp. 235-253, 2013.

[14] H.-B. Zhou, S.-M. Song, and J.-H. Song, "Design of sliding mode guidance law with dynamic delay and impact angle constraint," *International Journal of Control, Automation, and Systems*, vol. 15, no. 1, pp. 239-247, 2017.

[15] C. Ming and X. Wang, "Nonsingular terminal sliding mode control-based prescribed performance guidance law with impact angle constraints," *International Journal of Control, Automation, and Systems*, vol. 20, no. 3, pp. 715-726, 2022.

[16] S. R. Kumar, S. Rao, and D. Ghose, "Nonsingular terminal sliding mode guidance with impact angle constraints," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1114-1130, 2014.

[17] R. Bardhan and D. Ghose, "Nonlinear differential games-based impact-angle-constrained guidance law," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 3, pp. 384-402, 2015.

[18] R. Tekin and F. Holzapfel, "Impact angle control based on feedback linearization," *Proc. of AIAA Guidance, Navigation, and Control Conference*, p. 1509, 2017.

[19] C. Wang, W. Dong, J. Wang, and J. Shan, "Nonlinear suboptimal guidance law with impact angle constraint: an SDRE-based approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4831-4840, 2020.

[20] H.-G. Kim, J.-Y. Lee, and H. J. Kim, "Look angle constrained impact angle control guidance law for homing missiles with bearings-only measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 3096-3107, 2018.

[21] K. S. Erer, R. Tekin, and M. K. Ozgoren, "Look angle constrained impact angle control based on proportional navigation," *Proc. of AIAA Guidance, Navigation, and Control Conference*, 0091, 2015.

[22] T.-H. Kim, B.-G. Park, and M.-J. Tahk, "Bias-shaping method for biased proportional navigation with terminal-angle constraint," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1810-1816, 2013.

[23] A. Ratnoo, "Analysis of two-stage proportional navigation with heading constraints," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 1, pp. 156-164, 2016.

[24] X. Liu, Z. Shen, and P. Lu, "Closed-loop optimization of guidance gain for constrained impact," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 453-460, 2016.

[25] P. Lu, "Introducing computational guidance and control," 2017.

[26] S. He, H.-S. Shin, and A. Tsourdos, "Computational missile guidance: A deep reinforcement learning approach," *Journal of Aerospace Information Systems*, vol. 18, no. 8, pp. 571-582, 2021.

[27] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[28] Q. Zhang, Z. Gong, Z. Yang, and Z. Chen, "Distributed convex optimization for flocking of nonlinear multi-agent systems," *International Journal of Control, Automation, and Systems*, vol. 17, no. 5, pp. 1177-1183, 2019.

[29] X. Ping, S. Yang, B. Ding, T. Raissi, and Z. Li, "A convexity approach to dynamic output feedback robust MPC for lpv systems with bounded disturbances," *International Journal of Control, Automation, and Systems*, vol. 18, no. 6, pp. 1378-1391, 2020.

[30] J. Liu, W. Chen, and H. Dai, "Sampled-data based distributed convex optimization with event-triggered communication," *International Journal of Control, Automation and Systems*, vol. 14, no. 6, pp. 1421-1429, 2016.

[31] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, vol. 1, no. 1, pp. 23-40, 2017.

[32] H. Ahn, J. Park, H. Bang, and Y. Kim, "Model predictive control-based multirotor three-dimensional motion planning with point cloud obstacle," *Journal of Aerospace Information Systems*, pp. 1-15, 2021.

[33] J. Bae, S.-D. Lee, Y.-W. Kim, C.-H. Lee, and S.-Y. Kim, "Convex optimization-based entry guidance for spaceplane," *International Journal of Control, Automation, and Systems*, vol. 20, no. 5, pp. 1652-1670, 2022.

[34] X. Liu, Z. Shen, and P. Lu, "Exact convex relaxation for optimal flight of aerodynamically controlled missiles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1881-1892, 2016.

[35] H.-H. Kwon and H.-L. Choi, "A convex programming approach to mid-course trajectory optimization for air-to-ground missiles," *International Journal of Aeronautical and Space Sciences*, pp. 1-14, 2019.

[36] K. Zhang, S. Yang, and F. Xiong, "Rapid ascent trajectory optimization for guided rockets via sequential convex programming," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 13, pp. 4800-4809, 2019.

[37] H. Roh, Y.-J. Oh, M.-J. Tahk, K.-J. Kwon, and H.-H. Kwon, "L1-penalized sequential convex programming for fast trajectory optimization: with application to optimal missile guidance," *International Journal of Aeronautical and Space Sciences*, pp. 1-11, 2019.

[38] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714-1721, IEEE, 2017.

[39] H. J. Kappen, "An introduction to stochastic control theory, path integrals and reinforcement learning," *AIP Conference Proceedings*, vol. 887, pp. 149-181, 2007.

[40] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344-357, 2017.

[41] G. Williams, E. Rombokas, and T. Daniel, "Gpu based path integral control with learned dynamics," *arXiv preprint arXiv:1503.00330*, 2015.

[42] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433-1440, IEEE, 2016.

[43] A. Buyval, A. Gabdullin, K. Sozykin, and A. Klimchik, "Model predictive path integral control for car driving with autogenerated cost map based on prior map and camera image," *Proc. of IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2109-2114, IEEE, 2019.

[44] J. Pravitra, E. Theodorou, and E. N. Johnson, "Flying complex maneuvers with model predictive path integral control," *PRoc. of AIAA Scitech 2021 Forum*, 1957, 2021.

[45] I. S. Mohamed, G. Allibert, and P. Martinet, "Model predictive path integral control framework for partially observable navigation: A quadrotor case study," *Proc. of 16th IEEE International Conference on Control, Automation, Robotics and Vision, ICARCV 2020*, pp. 196-203, 2020.

[46] V. Comandur and J. Prasad, "Rotorcraft shipboard landing guidance using MPPI trajectory optimization," 2018.

[47] C. Liang, W. Wang, Z. Liu, C. Lai, and B. Zhou, "Learning to guide: Guidance law based on deep meta-learning and model predictive path integral control," *IEEE Access*, vol. 7, pp. 47353-47365, 2019.

[48] C. Liang, W. Wang, Z. Liu, C. Lai, and S. Wang, "Range-aware impact angle guidance law with deep reinforcement meta-learning," *IEEE Access*, vol. 8, pp. 152093-152104, 2020.

[49] M.-Y. Ryu, C.-H. Lee, and M.-J. Tahk, "Command shaping optimal guidance laws against high-speed incoming targets," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 10, pp. 2025-2033, 2015.

[50] C.-H. Lee, H.-S. Shin, J.-I. Lee, and M.-J. Tahk, "Zero-effort-miss shaping guidance laws," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 693-705, 2017.

[51] C.-H. Lee, J.-I. Lee, and M.-J. Tahk, "Sinusoidal function weighted optimal guidance laws," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 229, no. 3, pp. 534-542, 2015.

[52] J.-I. Lee, I.-S. Jeon, and C.-H. Lee, "Command-shaping guidance law based on a gaussian weighting function," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 772-777, 2014.

[53] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137-3181, 2010.

[54] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *arXiv preprint arXiv:1509.01149*, 2015.

[55] A. Bryson and Y.-C. Ho, *Applied Optimal Control*, Hemisphere, New York, 1975.

[56] P. Lu and F. Chavez, "Nonlinear optimal guidance," *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 6079, 2006.

[57] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877-900, 1999.

[58] D. Wei, W. Qiuqiu, X. Qunli, and Y. Shengjiang, "Multiple-constraint cooperative guidance based on two-stage sequential convex programming," *Chinese Journal of Aeronautics*, vol. 33, no. 1, pp. 296-307, 2020.

[59] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249-277, 2003.

Ki-Pyo Kim received his B.S. degree in electronic engineering from Kyungbook National University, Daegu, Korea, in 2002, and an M.S. degree in electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2004. He is currently working toward a Ph.D. degree in the Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology. Since 2004, he has been with the Agency for Defense Development (ADD), Daejeon, Korea. His current research interests include missile guidance and control.

Chang-Hun Lee received his B.S., M.S., and Ph.D. degrees in aerospace engineering from Korea Advanced Institute of Science and Technology (KAIST), in 2008, 2010, and 2013, respectively. From 2013 to 2015, he was a Senior Researcher for Guidance and Control Team, Agency for Defense Development (ADD), Daejeon, Korea. From 2016 to 2018, he was a Research Fellow for School of Aerospace, Transportation, and Manufacturing, Cranfield University, Bedford, United Kingdom. Since 2019, he has been with the Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea, where he is currently an Associate Professor. His recent research interests include advanced missile guidance and control, cooperative control for unmanned aerial vehicles, target tracking filter, deep learning, and aviation data analytics. Currently, he is technical editor of International Journal of Aeronautical and Space Science.