Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 3104397, 17 pages
https://doi.org/10.1155/2018/3104397

Research Article

Optimal Path-Following Guidance with Generalized Weighting Functions Based on Indirect Gauss Pseudospectral Method

Qi Chen,[1] **Xugang Wang**,[1] **and Jing Yang**[2]

[1] *School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *Xi'an Institute of Modern Control Technology, Xi'an 710065, China*

Correspondence should be addressed to Qi Chen; qichen@njust.edu.cn

Received 15 May 2018; Revised 16 July 2018; Accepted 25 July 2018; Published 9 August 2018

Academic Editor: Renato Vidoni

An indirect Gauss pseudospectral method based path-following guidance law is presented in this paper. A virtual target moving along the desired path with explicitly specified speed is introduced to formulate the guidance problem. By establishing a virtual target-fixed coordinate system, the path-following guidance is transformed into a terminal guidance with impact angle constraints, which is then solved by using indirect Gauss pseudospectral method. Meanwhile, the acceleration dynamics are modeled as the first-order lag to the command. Using the receding horizon technique a closed-loop guidance law, which considers generalized weighting functions (even discontinuous) of both the states and the control cost, is derived. The accuracy and effectiveness of the proposed guidance law are validated by numerical comparisons. A STM32 Nucleo board based on the ARM Cortex-M7 processor is used to evaluate the real-time computational performance of the proposed indirect Gauss pseudospectral method. Simulations for various types of desired paths are presented to show that the proposed guidance law has better performance when compared with the existing results for pure pursuit, a nonlinear guidance law, and trajectory shaping path-following guidance and provides more degrees of freedom in path-following guidance design applications.

1. Introduction

During the last decade, there is a growing interest in unmanned aerial vehicles (UAVs) in both civilian and military applications like geological surveys, power line patrol, reconnaissance, etc. In most of these applications, the UAVs are usually required to follow a desired path accurately. The desired paths are commonly planned as straight lines or circular lines with specified constraints. To obtain a satisfactory path-following performance, a robust and efficient path-following guidance law is needed.

In recent years, a variety of path-following guidance laws have been developed for UAVs. Nelson et al. [1] and Lawrence et al. [2] proposed approaches based on a notion of vector field. The vector field-based path-following approach uses vector fields to represent the desired headings to drive the UAV onto the defined path. This approach has high robustness but is complicated in construction of the vector fields and also difficult in implementation. A waypoint-based path-following law was developed by Tsourdos et al. [3], in which a number of waypoints are selected on the desired path for the vehicle to pass through. However, the results based on this approach usually have low accuracy, and, specifically, a large path-following error may occur when the path is highly curved. Another waypoint-based guidance law was proposed by Liang et al. [4] for entry vehicle. In this study, the prescribed waypoints and the expected heading angle are imposed on the vehicle as additional constraints. The proposed guidance can successfully generate a lateral trajectory that satisfies the waypoint constraint and thus guarantee that the vehicle is able to reach all the waypoints. Yang et al. [5] studied the path tracking problem for a fixed-wing unmanned aircraft using the error-regulation philosophy, in which an adaptive nonlinear model predictive controller is designed to minimize both the mean and the maximum error between the reference trajectory and the UAV, and thus provides accurate tracking performance.

Virtual target-based approaches lead to another class of path-following guidance laws. The main objective of these approaches is to chase a virtual target point moving along

the desired path, which is ahead of the UAV. The virtual target is initially placed at the beginning of the desired path. Once the vehicle starts to track the desired path, a virtual speed related to the vehicle's speed and the separation between the vehicle and the virtual target is generated and imposed on the virtual target. By using the virtual speed and the curvature of the desired path, the states of the virtual target can be explicitly propagated along the desired path till the end of the engagement [6–8]. As a consequence, the position of the virtual target is always available during the entire guidance process. The line-of-sight guidance [9] and proportional navigation guidance [10] were used to drive the vehicle to chase the virtual target, which eventually drives the vehicle onto the path, and the same problem has been considered by Medagoda and Gibbens [11] using pure pursuit guidance. However, a heading error will be caused for curved paths because pure pursuit guidance consistently compels the vehicle to head toward the target. Therefore, a path-following error will occur. Further improvement to pure pursuit guidance was proposed by Cho et al. [12], in which differential geometry of space curves is used to extend the pure pursuit method for 3D path following. A nonlinear path-following guidance law adapted from pure pursuit based methods was proposed by Park et al. [13, 14]. This method is prominent due to its robustness of convergence for all initial geometries, the simple guidance command, and the so-called "look-ahead effect" which guarantees accurate following of curved paths. However, the lateral acceleration is undefined when the initial position of the vehicle is outside of the specified look-ahead distance from the desired path. Moreover, an overshoot response in the initial phase is another issue. Exploiting the concept of terminal missile guidance law with impact angle constraint, Ratnoo et al. [15] proposed a new path-following law based on trajectory shaping guidance. The advantages of this method are the fast rate of convergence, the negligible path-following error, and the strong robustness with respect to the minimum distance.

However, the research works mentioned above have not considered the autopilot delay, which cannot be ignored in practice, especially for the UAVs with low control authority. Most of the existing works assume that the actual lateral acceleration is the same as the command, which means that the acceleration response is instantaneous. However, this is not practical as it always takes some time to achieve the desired guidance command in practical system. This delay may degrade the overall path-following performance and even cause instability. To obtain satisfactory path-following performance, it is of necessity to take the autopilot delay into account. In addition, previous works mentioned above generally utilize simple guidance laws to chase the virtual target and do not take the weighting functions into account to improve the path-following guidance performance. As can be shown that through appropriate selections of the weighting functions, the vehicle's trajectory and acceleration profile can be shaped as desired for achieving different path-following objectives. Furthermore, if the weighting functions can be chosen arbitrarily, then the flexibility of the path-following guidance design could be largely enhanced. Various weighting functions, such as constant function [16], Gaussian function [17], time-to-go function [18], exponential function [19], hyperbolic tangent function [20], and sinusoidal function [21], have been used to devise terminal guidance laws for different guidance objectives. These weighting functions have their corresponding advantages, for example, reducing sensitivity with respect to initial heading error, extending the operational margin to cope with the external disturbances in the terminal phase, or alleviating the acceleration command at the initial phase. Discontinuous functions that consist of the combination of the above-mentioned weighting functions give rise to attractive and prominent types of the weighting functions, in which different weighting factors are applied to weigh the states and/or controls in different guidance phases. Consequently, discontinuous weighting functions can take advantages of several weighting functions during the entire guidance process and thus significantly enhance the guidance performance. But, meanwhile, these discontinuous weighting functions are usually intractable for most of the guidance laws in existing literatures. However, the method proposed in this paper can easily cope with these discontinuous weighting functions. The results presented in this paper are the first attempts in the literature to derive the optimal path-following guidance with generalized weighting functions using the indirect Gauss pseudospectral method. This novel approach can handle complex weighting functions (even though they are discontinuous) which are intractable for most of the guidance laws in previous studies and thus provides more degrees of freedom in path-following guidance design applications. Similar to the use of terminal missile guidance law for path following in [15], a novel guidance logic with impact angle constraint considering generalized weighting functions as well as the autopilot delay is proposed in this paper to follow the virtual target on a planar path. Detailed numerical comparisons are presented to demonstrate the high path-following performance of the proposed guidance.

This paper is organized as follows. In Section 2, the path-following problem based on virtual target pursuit is formulated. In Section 3, an indirect Gauss pseudospectral method based approach is derived, and then a closed-loop path-following guidance law is proposed. The validation of the Gauss pseudospectral method based approach and the performance of the proposed guidance law are presented by numerical simulations in Section 4. Conclusions are given in Section 5.

2. Problem Formulation

Consider the path-following guidance geometry shown in Figure 1. Here, M denotes a vehicle; the curve is the desired path; T is a virtual target moving along the path and governed by the curvature of the desired path. The vehicle pursues the virtual target and reduces the distance R to converge to the desired path. V_m, γ_m, and γ_t denote the vehicle velocity, vehicle heading angle, and the virtual target heading angle, respectively. a_m is the vehicle acceleration perpendicular to the velocity vector to change the heading angle γ_m. Motivated

by [11, 15], the speed of the virtual target is chosen as a function of the vehicle speed v_m and the closing distance R as follows:

$$v_t = v_m \frac{R^*}{R} \tag{1}$$

where R^* is a design parameter within the guidance algorithm and represents the minimum allowed separation between the vehicle and the virtual target, because the virtual target's speed is inversely proportional to the closing distance. Therefore, the speed of the virtual target increases as the vehicle approaches the virtual target, which makes the vehicle always in pursuit of the virtual vehicle. This constraint links the dynamics of the vehicle and the virtual target and implies that the vehicle can never be closer to the virtual target than the minimum separation R^* (e.g., $R \geq R^*$). The choice of R^* and the guidance law of the vehicle affect the vehicle following performance. In order to achieve better following performance, the vehicle is expected to approach the virtual target in tail chase. To this end, the guidance law of the vehicle had better maintain a capacity of impact angle control (the expected impact angle is the virtual target heading angle γ_t). Trajectory shaping guidance law [16] is analyzed in [15] and is proven to be an effective logic for virtual target following on a planar path. However, when taking the autopilot dynamics into account, this method will have some limitation (for instance, oscillation and instability). Moreover, weighting functions are not considered in the previous studies. To eliminate this limitation and enhance the flexibility of the design of the path-following guidance, a new guidance law with impact angle constraint is proposed in this paper.

For convenience, a virtual target-fixed coordinate $Tx_{\mathrm{d}}y_{\mathrm{d}}$ is defined in Figure 1. The coordinate $Tx_{\mathrm{d}}y_{\mathrm{d}}$ is fixed with the virtual target and rotates with the virtual target heading angle γ_t from the inertial reference coordinate $x_{\mathrm{I}}y_{\mathrm{I}}$. The engagement kinematics and proposed guidance are derived in $Tx_{\mathrm{d}}y_{\mathrm{d}}$. The equations of motion for the engagement defined in $Tx_{\mathrm{d}}y_{\mathrm{d}}$ are given by

$$\begin{aligned} \dot{y}(t) &= V_m(t)\sin\gamma(t), \\ y(0) &= y_0 \\ \dot{\gamma}(t) &= \frac{a_m(t)}{V_m(t)}, \\ \gamma(0) &= \gamma_0 \end{aligned} \tag{2}$$

where y and γ are the vehicle cross-range and heading angle defined in $Tx_{\mathrm{d}}y_{\mathrm{d}}$, respectively. Under the assumption that V_m is constant and γ is small, (2) can be linearized as

$$\begin{aligned} \dot{y}(t) &= V_m(t)\gamma(t), \\ \dot{\gamma}(t) &= \frac{a_m(t)}{V_m(t)} \end{aligned} \tag{3}$$

Suppose that the vehicle autopilot model is a first-order delay system:

$$\dot{a}_m(t) = \frac{1}{t_\tau}\left[a_c(t) - a_m(t)\right] \tag{4}$$

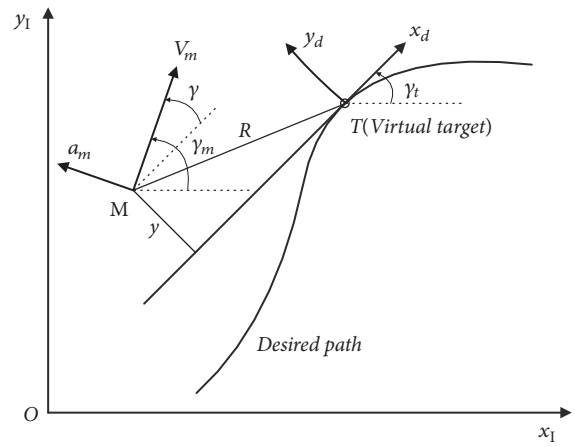


Figure 1: Path-following guidance geometry.

where a_c and t_τ denote the guidance command and autopilot time delay constant, respectively. Equations (3) and (4) can be rewritten in a compact form as follows:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \end{aligned} \tag{5}$$

where

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} y & \gamma & a_m \end{bmatrix}^T, \\ \boldsymbol{u} &= a_c, \\ \boldsymbol{A} &= \begin{bmatrix} 0 & V_m & 0 \\ 0 & 0 & \dfrac{1}{V_m} \\ 0 & 0 & \dfrac{-1}{t_\tau} \end{bmatrix}, \\ \boldsymbol{B} &= \begin{bmatrix} 0 \\ 0 \\ \dfrac{1}{t_\tau} \end{bmatrix} \end{aligned} \tag{6}$$

Note that this linearized kinematics has been widely used to devise optimal guidance laws with diverse constraints from many researchers in [17, 18, 21–23]. It should be pointed out that the existing work among these studies mainly concentrates on various weighting functions only on the control energy cost. To enhance the design flexibility, we consider the generalized weighting functions, even if they are discontinuous, on both the states and control to construct a novel design framework of path-following guidance law.

It can be observed from Figure 1 that if the speed vector coincides with the axis Tx_d, namely, $y = 0$ and $\gamma = 0$, then the impact angle constraint will be satisfied. Therefore, we consider the following finite-time optimal control problem

$\mathcal{P}$: find $\boldsymbol{u}(t)$ in the time interval $[t_0, t_f]$ that minimize the cost function

$$J = \frac{1}{2}\boldsymbol{x}^T\left(t_f\right)\boldsymbol{S}_f\boldsymbol{x}\left(t_f\right) + \frac{1}{2}\int_{t_0}^{t_f}\left[\boldsymbol{x}^T(t)\,\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\,\boldsymbol{R}\boldsymbol{u}(t)\right]dt \tag{7}$$

subject to (5).

Here, t_0 and t_f are the initial and final time of flight, respectively. $\boldsymbol{S}_f$ is used to control the terminal state of the vehicle. $\boldsymbol{Q}$ and $\boldsymbol{R}$ are used to weight the state variables and the control effort and are considered as the weighting functions.

3. Design of the Optimal Path-Following Guidance Law

3.1. Optimal Solution Based on Indirect Gauss Pseudospectral Method. The Hamiltonian function of the problem $\mathcal{P}$ is

$$H = \frac{1}{2}\left[\boldsymbol{x}(t)^T\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}(t)^T\boldsymbol{R}\boldsymbol{u}(t)\right] + \boldsymbol{\lambda}(t)^T\left[\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)\right] \tag{8}$$

where $\boldsymbol{\lambda}(t)$ is the costate vector that satisfies the dynamics

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial H}{\partial \boldsymbol{x}(t)} = -\left[\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{A}^T\boldsymbol{\lambda}(t)\right] \tag{9}$$

with the transversality condition

$$\boldsymbol{\lambda}\left(t_f\right) = \boldsymbol{S}_f\boldsymbol{x}\left(t_f\right) \tag{10}$$

According to the minimum principle, the necessary optimality condition is

$$\frac{\partial H}{\partial \boldsymbol{u}(t)} = 0 \tag{11}$$

which yields the optimal control

$$\boldsymbol{u}^*(t) = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{\lambda}(t) \tag{12}$$

Substituting (12) into (5), and combining (9), the linear two-point boundary-value problem can be described by

$$\dot{\boldsymbol{x}}(t) = \frac{\partial H}{\partial \boldsymbol{\lambda}(t)} = \boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{\lambda}(t), \tag{13a}$$

$$\boldsymbol{x}\left(t_0\right) = \boldsymbol{x}_0$$

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial H}{\partial \boldsymbol{x}(t)} = -\left[\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{A}^T\boldsymbol{\lambda}(t)\right], \tag{13b}$$

$$\boldsymbol{\lambda}\left(t_f\right) = \boldsymbol{S}_f\boldsymbol{x}\left(t_f\right)$$

For some complex or discontinuous weighting functions, it is intractable to solve (13a) and (13b) analytically. A conventional method to solve problem ((13a), (13b)) is the backward sweep method. In this method, it is assumed that $\boldsymbol{\lambda}(t) = \boldsymbol{P}(t)\boldsymbol{x}(t)$, which is substituted into (12), where $\boldsymbol{P}(t)$ is determined by solving the differential Riccati matrix equation. It is well-known that this method needs iterations and is numerically intensive and potentially unstable. To overcome these disadvantages, a novel approach based on indirect Gauss pseudospectral method that can deal with arbitrary weighting functions is derived to tackle problem ((13a), (13b)). Because the computation interval used in the Gauss pseudospectral method is [-1, 1], it is necessary to transfer the time interval $[t_0, t_f]$ to [-1, 1] via the following transformation:

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0} \tag{14}$$

By applying (14), ((13a), (13b)) is transferred to the following form:

$$\dot{\boldsymbol{x}}(\tau) = \frac{t_f - t_0}{2}\left[\boldsymbol{A}\boldsymbol{x}(\tau) - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{\lambda}(\tau)\right], \tag{15a}$$

$$\boldsymbol{x}(-1) = \boldsymbol{x}_0$$

$$\dot{\boldsymbol{\lambda}}(\tau) = -\frac{t_f - t_0}{2}\left[\boldsymbol{Q}\boldsymbol{x}(\tau) + \boldsymbol{A}^T\boldsymbol{\lambda}(\tau)\right], \tag{15b}$$

$$\boldsymbol{\lambda}(1) = \boldsymbol{S}_f\boldsymbol{x}(1)$$

Next, ((15a), (15b)) is transformed into a set of algebraic equations based on the approximated state variable $\boldsymbol{x}(\tau)$ and costate variable $\boldsymbol{\lambda}(\tau)$ using interpolating polynomials. It should be noted that the initial value is known for the state variable $\boldsymbol{x}(\tau)$ in (15a), whereas the terminal value is known for the costate variable $\boldsymbol{\lambda}(\tau)$ in (15b). Therefore, there is a slight difference between approaches used to approximate $\boldsymbol{x}(\tau)$ and $\boldsymbol{\lambda}(\tau)$. For $\boldsymbol{x}(\tau)$, the boundary point, -1, and the N Gauss points [24], τ_k, $k = 1, \ldots, N$, which are all in the interior of the interval [-1, 1], are used as the interpolation points to form the interpolating polynomials

$$\boldsymbol{x}(\tau) \approx X(\tau) = \boldsymbol{x}(-1)\,L_0(\tau) + \sum_{k=1}^{N}\boldsymbol{x}\left(\tau_k\right)L_k(\tau) \tag{16}$$

where $L_i(\tau)$ $(i = 0, \ldots, N)$ are $N + 1$ Lagrange interpolating polynomials and defined as

$$L_i(\tau) = \prod_{j=0, j\neq i}^{N}\frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{17}$$

Differentiating (16) yields

$$\dot{\boldsymbol{x}}(\tau) \approx \dot{X}(\tau) = \boldsymbol{x}(-1)\,\dot{L}_0(\tau) + \sum_{k=1}^{N}\boldsymbol{x}\left(\tau_k\right)\dot{L}_k(\tau) \tag{18}$$

Denote $\overline{D}_i = \dot{L}_0(\tau_i)$ and $D_{ik} = \dot{L}_k(\tau_i)$; (18) can be written as

$$\dot{\boldsymbol{x}}(\tau_i) \approx \dot{X}(\tau_i) = \boldsymbol{x}(-1)\,\overline{D}_i + \sum_{k=1}^{N} \boldsymbol{x}(\tau_k)\, D_{ik} \tag{19}$$

for $i = 1, 2, \ldots, N$, where $\overline{\boldsymbol{D}} \in \mathbb{R}^N$ and $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ are differential approximation matrices.

Because the terminal value is known for $\boldsymbol{\lambda}(\tau)$, the boundary point, 1, and the N Gauss points, τ_k, $k = 1, \ldots, N$, are used as the interpolation points to form the interpolating polynomials

$$\dot{\boldsymbol{\lambda}}(\tau_i) \approx \dot{\Lambda}(\tau_i) = \boldsymbol{\lambda}(1)\,\overline{D}_i^* + \sum_{k=1}^{N} \boldsymbol{\lambda}(\tau_k)\, D_{ik}^* \tag{20}$$

for $i = 1, 2, \ldots, N$, where $\overline{\boldsymbol{D}}^* \in \mathbb{R}^N$ and $\boldsymbol{D}^* \in \mathbb{R}^{N\times N}$ are adjoint differential approximation matrices. According to (15b), $\boldsymbol{\lambda}(1) = \boldsymbol{S}_f \boldsymbol{x}(1)$, where $\boldsymbol{x}(1)$ is the terminal value of state and can be determined via Gauss quadrature [25]:

$$\begin{aligned}\boldsymbol{x}(1) = {} & \boldsymbol{x}(-1) \\ & + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \left[\boldsymbol{A}\boldsymbol{x}(\tau_k) - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{\lambda}(\tau_k)\right]\end{aligned} \tag{21}$$

where w_k are Gauss weights for $k = 1, 2, \ldots, N$. Thus the terminal value of $\boldsymbol{\lambda}$ is

$$\begin{aligned}\boldsymbol{\lambda}(1) = {} & \boldsymbol{S}_f \boldsymbol{x}(-1) \\ & + \boldsymbol{S}_f \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \left[\boldsymbol{A}\boldsymbol{x}(\tau_k) - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{\lambda}(\tau_k)\right]\end{aligned} \tag{22}$$

$$\widetilde{\overline{\boldsymbol{D}}}^* = \begin{bmatrix} \overline{D}_1^* \boldsymbol{S}_f \\ \overline{D}_2^* \boldsymbol{S}_f \\ \vdots \\ \overline{D}_N^* \boldsymbol{S}_f \end{bmatrix} \in \mathbb{R}^{nN\times n},$$

$$\widetilde{\overline{\boldsymbol{D}}} = \begin{bmatrix} \overline{D}_1 \boldsymbol{I}_n \\ \overline{D}_2 \boldsymbol{I}_n \\ \vdots \\ \overline{D}_N \boldsymbol{I}_n \end{bmatrix} \in \mathbb{R}^{nN\times n},$$

Substituting (22) into (20) and including (19), it follows that

$$\begin{aligned}& \overline{D}_i^* \boldsymbol{S}_f \boldsymbol{x}(-1) \\ & + \overline{D}_i^* \boldsymbol{S}_f \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \left(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{\lambda}_k\right) \\ & + \sum_{k=1}^{N} D_{ik}^* \boldsymbol{\lambda}_k = -\frac{t_f - t_0}{2}\left[\boldsymbol{Q}\boldsymbol{x}_i + \boldsymbol{A}^T \boldsymbol{\lambda}_i\right]\end{aligned} \tag{23a}$$

$$\overline{D}_i \boldsymbol{x}(-1) + \sum_{k=1}^{N} D_{ik} \boldsymbol{x}_k = \frac{t_f - t_0}{2}\left[\boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{\lambda}_i\right] \tag{23b}$$

where $i = 1, 2, \ldots, N$, $\boldsymbol{x}_k \equiv \boldsymbol{x}(\tau_k)$, $\boldsymbol{\lambda}_k \equiv \boldsymbol{\lambda}(\tau_k)$ for $k = 1, 2, \ldots, N$.

As proven in [24], the differential approximation matrices $\overline{\boldsymbol{D}}$ and $\boldsymbol{D}$, and the adjoint differential approximation matrices $\overline{\boldsymbol{D}}^*$ and $\boldsymbol{D}^*$ have the following relationships:

$$\overline{D}_i = -\sum_{k=1}^{N} D_{ik} \tag{24a}$$

$$\overline{D}_i^* = -\sum_{k=1}^{N} D_{ik}^* \tag{24b}$$

$$D_{ik}^* = -\frac{w_k}{w_i} D_{ki} \tag{24c}$$

It can be seen from (24a), (24b), and (24c) that as long as the matrix $\boldsymbol{D}$ is determined via differentiating the Lagrange interpolating polynomials $L_k(t)$, the other matrices $\overline{\boldsymbol{D}}$, $\overline{\boldsymbol{D}}^*$, and $\boldsymbol{D}^*$ can be directly calculated by employing (24a), (24b), and (24c). Furthermore, the matrix $\boldsymbol{D}$ and the Gauss weights w_k can be computed offline, and thus all the other matrices can be also computed offline.

Denote $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1^{\mathrm{T}}, \boldsymbol{\lambda}_2^{\mathrm{T}}, \ldots, \boldsymbol{\lambda}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nN}$, $\boldsymbol{X} = [\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}, \ldots, \boldsymbol{x}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nN}$, $\boldsymbol{X}_0 = \boldsymbol{x}(-1) \in \mathbb{R}^n$. Equations (23a) and (23b) can be rearranged as

$$\widetilde{\overline{\boldsymbol{D}}}^* \boldsymbol{X}_0 + \widetilde{\boldsymbol{D}}_+^* \boldsymbol{\Lambda} + \widetilde{\boldsymbol{Q}} \boldsymbol{X} = \boldsymbol{0} \tag{25a}$$

$$\widetilde{\overline{\boldsymbol{D}}} \boldsymbol{X}_0 + \widetilde{\boldsymbol{D}}_- \boldsymbol{X} + \widetilde{\boldsymbol{G}} \boldsymbol{\Lambda} = \boldsymbol{0} \tag{25b}$$

where

(26)

$$\widetilde{\boldsymbol{D}}_{+}^{*} = \begin{bmatrix} D_{11}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\left(\overline{D}_1^{*}\boldsymbol{S}_f w_1 \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T - \boldsymbol{A}^T\right) & D_{12}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\overline{D}^{*}\boldsymbol{S}_f w_2 \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T & \cdots & D_{1N}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\overline{D}_1^{*}\boldsymbol{S}_f w_N \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \\ D_{21}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\overline{D}_2^{*}\boldsymbol{S}_f w_1 \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T & D_{22}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\left(\overline{D}_2^{*}\boldsymbol{S}_f w_2 \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T - \boldsymbol{A}^T\right) & \cdots & D_{2N}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\overline{D}_2^{*}\boldsymbol{S}_f w_N \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \\ \vdots & \vdots & \ddots & \vdots \\ D_{N1}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\overline{D}_N^{*}\boldsymbol{S}_f w_1 \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T & D_{N2}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\overline{D}_N^{*}\boldsymbol{S}_f w_2 \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T & \cdots & D_{NN}^{*}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\left(\overline{D}_N^{*}\boldsymbol{S}_f w_N \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T - \boldsymbol{A}^T\right) \end{bmatrix} \in \mathbb{R}^{nN\times nN} \tag{27}$$

$$\widetilde{\boldsymbol{D}}_{-} = \begin{bmatrix} D_{11}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\boldsymbol{A} & D_{12}\boldsymbol{I}_n & \cdots & D_{1N}\boldsymbol{I}_n \\ D_{21}\boldsymbol{I}_n & D_{22}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\boldsymbol{A} & \cdots & D_{2N}\boldsymbol{I}_n \\ \vdots & \vdots & \ddots & \vdots \\ D_{N1}\boldsymbol{I}_n & D_{N2}\boldsymbol{I}_n & \cdots & D_{NN}\boldsymbol{I}_n - \frac{t_f - t_0}{2}\boldsymbol{A} \end{bmatrix} \in \mathbb{R}^{nN\times nN}$$

$$\widetilde{\boldsymbol{Q}} = \begin{bmatrix} \frac{t_f - t_0}{2}\left(\overline{D}_1^{*}\boldsymbol{S}_f w_1 \boldsymbol{A} + \boldsymbol{Q}\right) & \frac{t_f - t_0}{2}\overline{D}_1^{*}\boldsymbol{S}_f w_2 \boldsymbol{A} & \cdots & \frac{t_f - t_0}{2}\overline{D}_1^{*}\boldsymbol{S}_f w_N \boldsymbol{A} \\ \frac{t_f - t_0}{2}\overline{D}_2^{*}\boldsymbol{S}_f w_1 \boldsymbol{A} & \frac{t_f - t_0}{2}\left(\overline{D}_2^{*}\boldsymbol{S}_f w_2 \boldsymbol{A} + \boldsymbol{Q}\right) & \cdots & \frac{t_f - t_0}{2}\overline{D}_2^{*}\boldsymbol{S}_f w_N \boldsymbol{A} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{t_f - t_0}{2}\overline{D}_N^{*}\boldsymbol{S}_f w_1 \boldsymbol{A} & \frac{t_f - t_0}{2}\overline{D}_N^{*}\boldsymbol{S}_f w_2 \boldsymbol{A} & \cdots & \frac{t_f - t_0}{2}\left(\overline{D}_N^{*}\boldsymbol{S}_f w_N \boldsymbol{A} + \boldsymbol{Q}\right) \end{bmatrix} \in \mathbb{R}^{nN\times nN} \tag{28}$$

$$\widetilde{\boldsymbol{G}} = \begin{bmatrix} \frac{t_f - t_0}{2}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T & \boldsymbol{0}_n & \cdots & \boldsymbol{0}_n \\ \boldsymbol{0}_n & \frac{t_f - t_0}{2}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T & \cdots & \boldsymbol{0}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_n & \boldsymbol{0}_n & \boldsymbol{0}_n & \frac{t_f - t_0}{2}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \end{bmatrix} \in \mathbb{R}^{nN\times nN}$$

In the preceding matrices, $\boldsymbol{I}_n$ and $\boldsymbol{0}_n$ are the $n \times n$ identity and zero matrices, respectively. Next, we transform (25a) and (25b) into a concise form as

$$\begin{bmatrix} \widetilde{\overline{\boldsymbol{D}}}^{*} \\ \widetilde{\overline{\boldsymbol{D}}} \end{bmatrix} \boldsymbol{X}_0 + \begin{bmatrix} \widetilde{\boldsymbol{D}}_{+}^{*} & \widetilde{\boldsymbol{Q}} \\ \widetilde{\boldsymbol{G}} & \widetilde{\boldsymbol{D}}_{-} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda} \\ \boldsymbol{X} \end{bmatrix} = \boldsymbol{0} \tag{29}$$

Denote $\overline{\boldsymbol{D}}_{Aug} = \begin{bmatrix} \widetilde{\overline{\boldsymbol{D}}}^{*} \\ \widetilde{\overline{\boldsymbol{D}}} \end{bmatrix} \in \mathbb{R}^{2nN\times n}$, $\boldsymbol{D}_{Aug} = \begin{bmatrix} \widetilde{\boldsymbol{D}}_{+}^{*} & \widetilde{\boldsymbol{Q}} \\ \widetilde{\boldsymbol{G}} & \widetilde{\boldsymbol{D}}_{-} \end{bmatrix} \in \mathbb{R}^{2nN\times 2nN}$. It follows that

$$\overline{\boldsymbol{D}}_{Aug}\boldsymbol{X}_0 + \boldsymbol{D}_{Aug}\begin{bmatrix} \boldsymbol{\Lambda} \\ \boldsymbol{X} \end{bmatrix} = \boldsymbol{0} \tag{30}$$

By solving (30), the following solution to the two-point boundary-value problem of equations (15a) and (15b) is obtained:

$$\begin{bmatrix} \boldsymbol{\Lambda} \\ \boldsymbol{X} \end{bmatrix} = -\boldsymbol{D}_{Aug}^{-1}\overline{\boldsymbol{D}}_{Aug}\boldsymbol{X}_0 \tag{31}$$

It is important to note from (31) that once the initial state value is given, the states and costates at all the Gauss points can be simultaneously calculated without any explicit integration process. However, (31) cannot provide the values of the states and costates at the boundary points because Gauss points do not include the boundary points. Since the initial value of $\boldsymbol{x}$ has been given, its terminal value can be calculated via (21), and the terminal value of $\boldsymbol{\lambda}$ can be calculated via (22). Regarding the initial value of $\boldsymbol{\lambda}$, the following equation can be used:

$$\boldsymbol{\lambda}(-1) = \boldsymbol{\lambda}(1) - \frac{t_f - t_0}{2}\sum_{k=1}^{N} w_k\left(\boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda}_k\right) \tag{32}$$

Now, all the costates at the discretization points (Gauss points and the boundary points) have been obtained from (21), (22), (31), and (32); the optimal control can be thus determined via the following equation:

$$\boldsymbol{u}_i^{*} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{\lambda}_i \quad (i = 0, 1, 2, \ldots, N+1) \tag{33}$$

3.2. Implementation of Path-Following Guidance. Note that the final time of the flight t_f is necessary in the computation

of matrices $\widetilde{\boldsymbol{D}}_{+}^{*}$, $\widetilde{\boldsymbol{D}}_{-}$, $\widetilde{\boldsymbol{Q}}$, and $\widetilde{\boldsymbol{G}}$. In order to obtain an approximation of t_f, the following equation is employed:

$$t_f = t_0 + t_{go} \approx t_0 + \frac{R}{V_m} \tag{34}$$

As the proposed guidance with impact angle constraint is derived in the new coordinate $Tx_{\mathrm{d}}y_{\mathrm{d}}$, the vehicle cross-range y and heading angle γ in $Tx_{\mathrm{d}}y_{\mathrm{d}}$ are computed as

$$y = -\left(x_m - x_t\right)\sin\gamma_t + \left(y_m - y_t\right)\cos\gamma_t \tag{35a}$$

$$\gamma = \gamma_m - \gamma_t \tag{35b}$$

where (x_m, y_m) and (x_t, y_t) are the coordinates of the vehicle and the virtual vehicle in the inertial coordinate $Ox_{\mathrm{I}}y_{\mathrm{I}}$, respectively.

Note that the optimal solution to the problem $\mathscr{P}$ described previously is open-loop. To obtain a closed-loop solution, the receding horizon technique is used in this study. Finally, the procedure for implementing the proposed guidance law with impact angle constraint for the path following can be summarized as follows.

(1) Initialize the number of Gauss point N. Set t_0=0.

(2) Solve the problem $\mathscr{P}$:

  (a) Compute t_f from (34) and $\boldsymbol{X}_0$ from (35a) and (35b).

  (b) Compute the matrices $\overline{\boldsymbol{D}}_{Aug}$ and $\boldsymbol{D}_{Aug}$, and solve (31).

  (c) Get the optimal control $\boldsymbol{u}_i$ $(i = 0, 1, \ldots, N+1)$ using (33).

(3) Apply the optimal control at the first point, $\boldsymbol{u}_0$, to the vehicle's dynamics and update the states of the vehicle.

(4) Update the states of the virtual vehicle according to (1) and the curvature of the desired path.

(5) Update t_0 by $t_0+\Delta t$ (Δt is the guidance time step).

(6) Repeat steps (2)-(5) until the vehicle reaches the end of the desired path.

4. Hardware Experiment and Simulation Results

In this section, a hardware experimental platform based on the ARM Cortex-M7 processor will be addressed to evaluate the real-time computational performance of the proposed method. Simulation will be provided to validate the accuracy and effectiveness of the proposed method for solving problem $\mathscr{P}$ via comparing with GPOPS [26] which is an open-source software for solving optimal control problems, and then the performance of the proposed guidance law in a path-following problem, taken from [15], will be investigated. In all of the following simulations, unless stated otherwise, the vehicle considered in this study has a constant velocity V_m as 50 m/s, a lateral acceleration saturation ±15g, and an autopilot dynamic delay t_τ=0.5s. It should be pointed out that the lateral acceleration saturation of ±15g might be unfeasible in practice for common UAVs. However, because the emphasis of this section is to demonstrate the effectiveness and the superiority of the proposed guidance law, in order to make fair comparisons with other guidance laws, here, we chose the same value of the lateral acceleration saturation that used in [15].

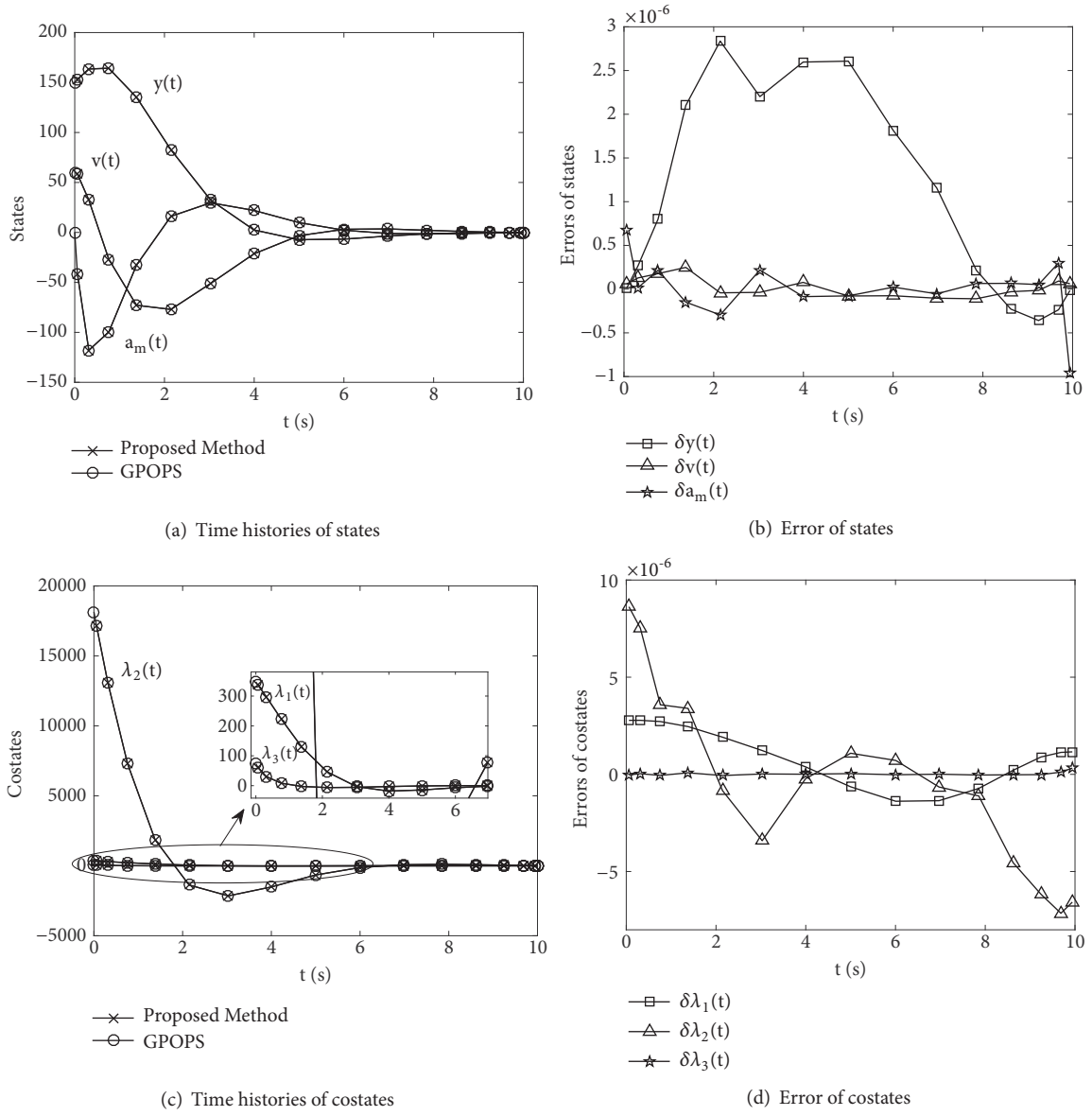
Table 1: Computational time of the proposed method and GPOPS performing on PC.

No.	Computational time		t_2/t_1
	Proposed Method t_1/(ms)	GPOPS t_2/(ms)	
1	2.39	462.04	193.32
2	2.50	467.73	187.09
3	2.46	463.50	188.41

4.1. Validation. In this section, the proposed method is compared with GPOPS by assuming that the initial value of the problem $\mathscr{P}$ is $\boldsymbol{x}_0 = \begin{bmatrix}150 & 50 & 0\end{bmatrix}^{\mathrm{T}}$ and the terminal time t_f is 10s. Actually, the terminal time and initial value are determined by (34), (35a), and (35b) in the path-following guidance law as stated above. The assumed value of $\boldsymbol{x}_0$ and t_f is only used for validation in this section. The weighting functions are chosen as $\boldsymbol{Q}$ = diag($\begin{bmatrix}1 & 1 & 1\end{bmatrix}$), $\boldsymbol{R}$ = 1, $\boldsymbol{S}_f$=diag($\begin{bmatrix}1\times10^5 & 1\times10^5 & 0\end{bmatrix}$), R^*=20m. The number of Gauss points, N, is chosen as 15. The comparison results of the states and costates presented in Figure 2 clearly show that the proposed method is consistent with GPOPS. Specifically, the maximum errors in the three components of the states are less than 3×10^{-6} and the maximum errors in the three components of the costates are less than 1×10^{-5}. These results demonstrate the high accuracy of the proposed method.

Table 1 summarizes the computational time of the two methods performing on a 2.66GHz Core 2 Duo personal computer (PC) with 2 GB RAM running Windows 7. As can be seen, the proposed method is computationally superior to GPOPS. This is mainly because GPOPS must invoke an optimization algorithm (for instance, ipopt or snopt) to iteratively find the optimal solution, which is time consuming and results in a serious computational burden. However, the proposed method obtains the optimal solution by just solving algebraic equations and does not need any iterative optimization process, which makes the proposed method has high computational efficiency.

To further evaluate the real-time computational capacity of the proposed method on a real digital processor, the guidance law is performed on a STM32 Nucleo-F767ZI development board. The hardware experimental platform is shown in Figure 3. The microcontroller (STM32F767ZIT6U) used in the development board is based on the high-performance ARM Cortex-M7 32-bit RISC core operating at up to 216MHz frequency and features a floating-point unit (FPU) which supports ARM double-precision and single-precision processing instructions. To implement the guidance



(a) Time histories of states

(b) Error of states

(c) Time histories of costates

(d) Error of costates

Figure 2: Comparisons of results between the proposed method and GPOPS.

law on the hardware platform, the STM32Cube MX software is used to generate the HAL-based embedded project of IAR compiler. All the matrix operations needed in the guidance law are carried out using the corresponding C functions provided by CMSIS DSP library, which is built-in suite of common signal processing functions. An oscilloscope is used to measure the computational time in the following procedure: keep the first port of GPIOB at a high level during the whole computation process, while set the port to a low level at other times. Hence the computation time can be captured by measuring the duration time of the high level. The single-precision FPU and the compiler's code optimization are turned on and off, respectively, in the test. Table 2 summarizes the computational time of the proposed guidance law running one loop (e.g., step 2-step 5 in the receding horizon procedure). It is seen that the computational time increases as the discretization point increases. When turning on FPU, the computational time is far less than that of the case with FPU OFF. In addition, the compiler optimization for the embedded code is also helpful to reduce the computation time. When we use the compiler optimization and turn on FPU at the same time, the computational time can be reduced to a satisfactory level, which means the proposed method is computationally viable for onboard implementation.

4.2. Straight-Line Following. A straight-line path is considered in this section for path-following performance analysis. The proposed guidance law is compared with trajectory shaping guidance in [15], nonlinear path-following guidance in [13, 14], and pure pursuit in [11]. The value of the

Table 2: Computational time of the proposed guidance law performing on hardware.

C/C++ Compiler Optimization	FPU: VFPv5 single precision	Computation time (ms)		
		N=5	N=10	N=15
High level with speed mode	Turn ON	3	18	87
	Turn OFF	40	303	1070
None	Turn ON	7	46	128
	Turn OFF	46	357	1183

autopilot delay t_τ varies as t_τ=0.1s, 0.3s, 0.5s, and 0.7s. Comparisons of trajectories and lateral command acceleration are presented in Figure 4. Trajectories for t_τ=0.1s, as shown in Figure 4(a), present good path following for all four guidance laws. However, the proposed guidance converges faster to the desired path than the other work and does not cause drastic control effort against the large initial control requirement of trajectory shaping guidance as shown in Figure 4(b). For t_τ=0.3s, the proposed guidance still obtains good path-following performance as shown in Figure 4(c), whereas pure pursuit has an overshoot; nonlinear guidance and trajectory shaping guidance start to oscillate, which cause path-following errors and slow convergences. Moreover, it is clear in Figure 4(d) that the control effort of trajectory shaping guidance is increased with oscillations until it is increased to the saturation. For an increased t_τ=0.5s as shown in Figures 4(e) and 4(f), the overshoot of pure pursuit and the oscillation of nonlinear guidance start to degenerate, and the corresponding control efforts increase to saturation. However, the proposed guidance still follows the desired path with negligible errors. While increasing the dynamic delay to 0.7s, the path-following performances for trajectory shaping guidance, nonlinear guidance, and pure pursuit deteriorate drastically, as shown in Figure 4(g). Nonlinear guidance and trajectory shaping guidance cannot even converge to the path because of oscillations. Figure 4(h) indicates that the proposed guidance also causes control saturation during the initial phase, but as the vehicle approaches the desired path, the control effort can return to reasonable values. However, the control efforts of nonlinear guidance and trajectory shaping guidance are approximately switching between their upper and lower bounds.

The initial heading angle of the vehicle could be different for different missions, a robust path-following guidance law should achieve good path following and be unaffected by the value of initial heading angle. Next, we analyze the path-following performance with different initial heading angles. For fair comparison, the autopilot lag of the vehicle is set as t_τ =0.1s and R^* is 20m, where trajectory shaping, pure pursuit, and nonlinear guidance law could achieve a satisfactory performance without oscillation as shown in Figure 4(a). Trajectories for $\gamma_m(t_0)$=-90deg, 0deg, and 180deg, as shown in Figure 5, present good path following for the proposed guidance, trajectory shaping, and nonlinear guidance. However, the proposed guidance converges faster to the desired path with negligible errors, whereas the pure pursuit fails to converge to the desired path. Moreover, it can be clearly seen that the trajectory shaping guidance always drives the vehicle to first move backward and then approach the desired path during the engagement. This is mainly because the trajectory shaping guidance always utilizes a tail-chase approach to chase the virtual target, which degrades the rate of convergence. Results show that the proposed guidance can achieve satisfactory path-following performances under different initial heading angles $\gamma_m(t_0)$ and is very robust with respect to the value of $\gamma_m(t_0)$.

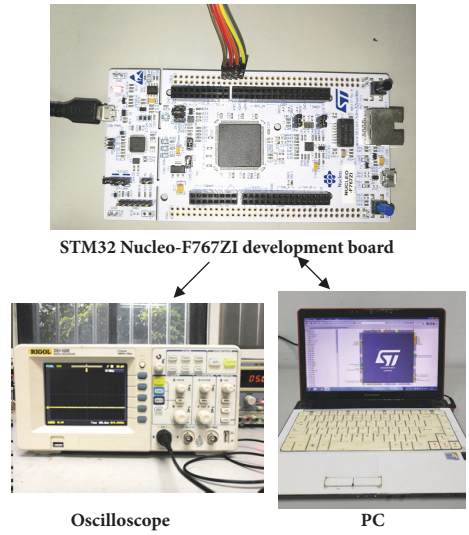
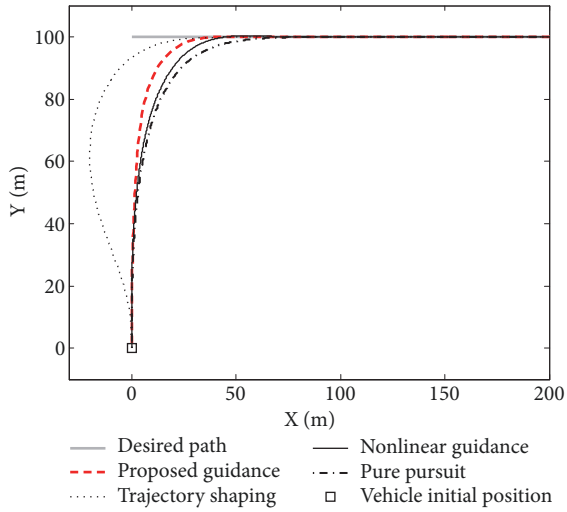
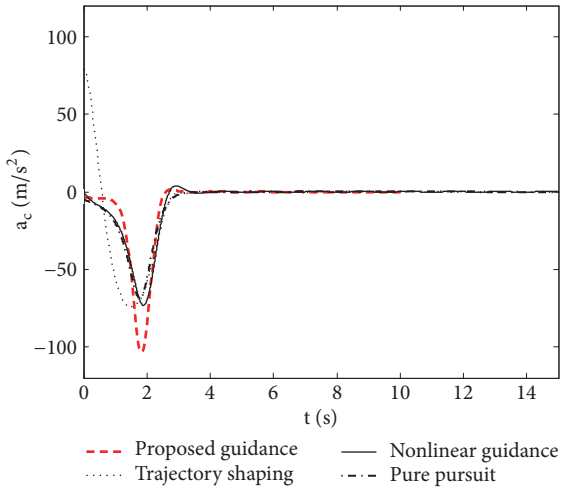


Figure 3: Hardware experimental platform based on STM32F767ZIT6U microcontroller.

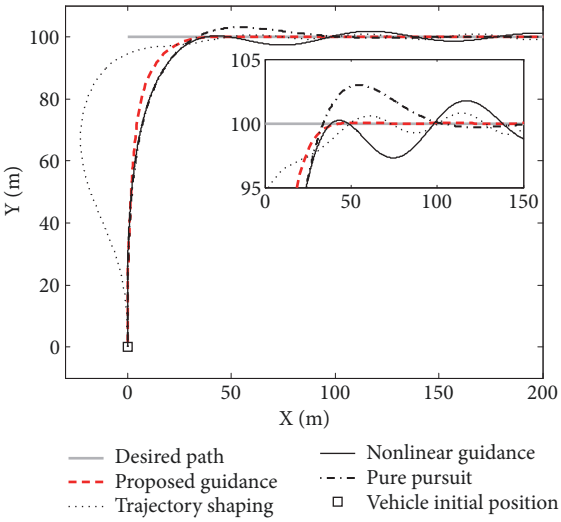
4.3. Straight-Line Following with Different Weighting Functions. Since we do not impose any constrained conditions on the weighting functions during the whole derivation of the guidance law, the proposed method can easily and directly tackle different weighting functions without any modifications, which means different weighting functions make no difference to the structure of the proposed method. This property makes the path-following guidance design very flexible to accomplish the specified objective. In this section, we take the straight-line path-following scenario as



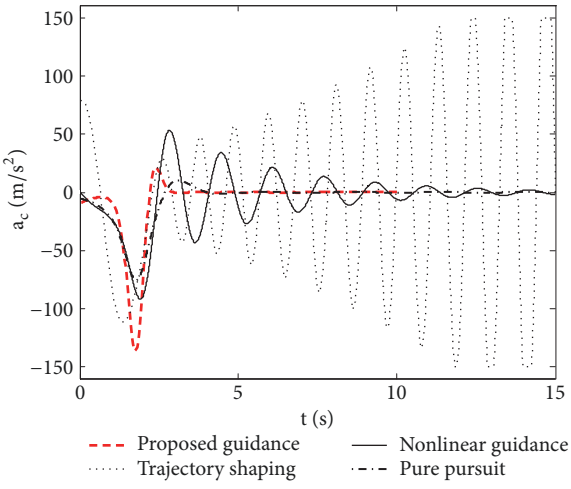
(a) Trajectories with t_τ=0.1s



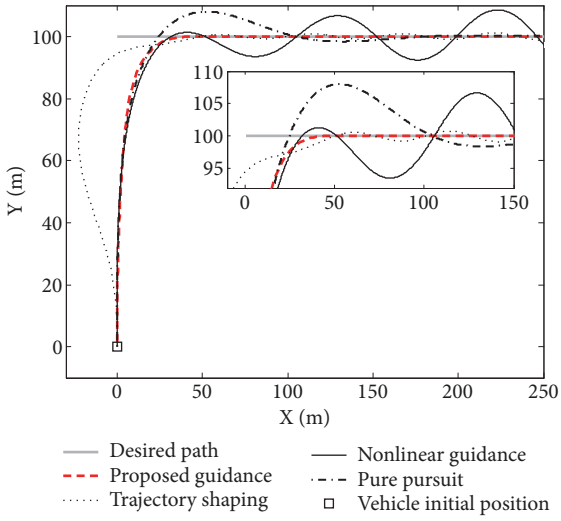
(b) Lateral command acceleration with t_τ=0.1s



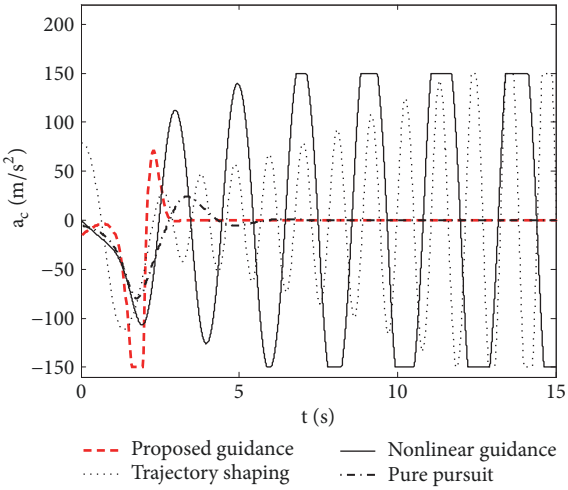
(c) Trajectories with t_τ=0.3s



(d) Lateral command acceleration with t_τ=0.3s

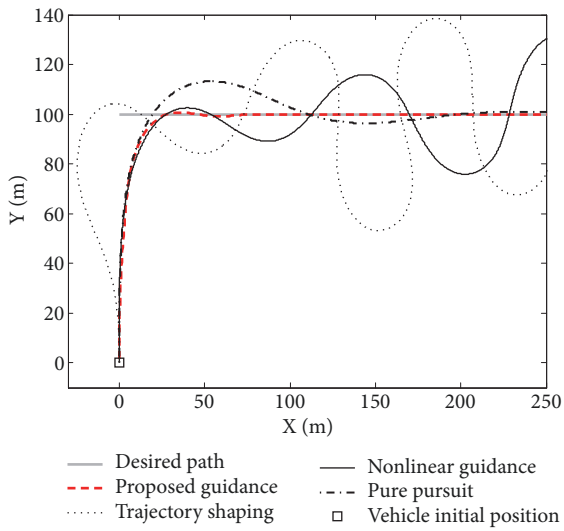


(e) Trajectories with t_τ=0.5s

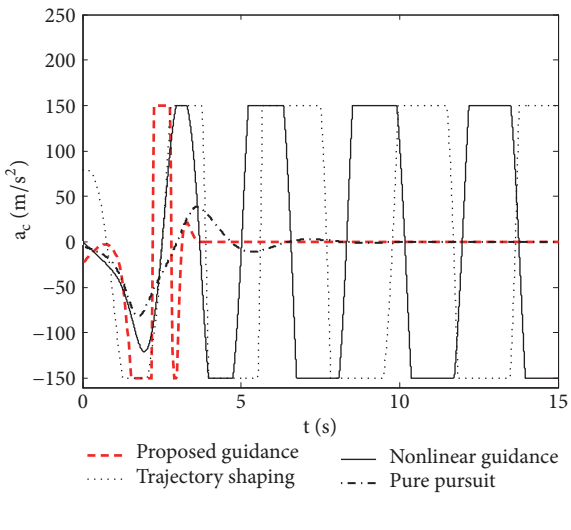


(f) Lateral command acceleration with t_τ=0.5s

Figure 4: Continued.



(g) Trajectories with t_τ=0.7s



(h) Lateral command acceleration with t_τ=0.7s

FIGURE 4: Straight-line following with R^*=20m.

an example and utilize different weighting functions of both states and control cost to demonstrate the path-following performance of the proposed guidance law. Three cases with different weighting functions, showed in (36)–(39), are investigated in the simulation. Case 1 considers simple weighting functions, in which both $\boldsymbol{Q}$ and $\boldsymbol{R}$ are constants. Note that one special case of case 1, $\boldsymbol{Q} = \boldsymbol{0}$, $\boldsymbol{R} = 1$, is frequently used to derive the optimal guidance, as studied in [15, 16, 27, 28]. In the second case, $\boldsymbol{Q}$ is chosen as an exponential function with respect to the time-to-go, while $\boldsymbol{R}$ is chosen as a Gaussian weighting function. Case 3 presents hybrid weighting functions that are of discontinuous form, and switches between those in Cases 1 and 2 by the path-following error d. Note that discontinuous weighting functions are intractable for most of the guidance laws in existing literatures.

Case 1.

$$\boldsymbol{Q}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{R}(t) = 1 \tag{36}$$

Case 2.

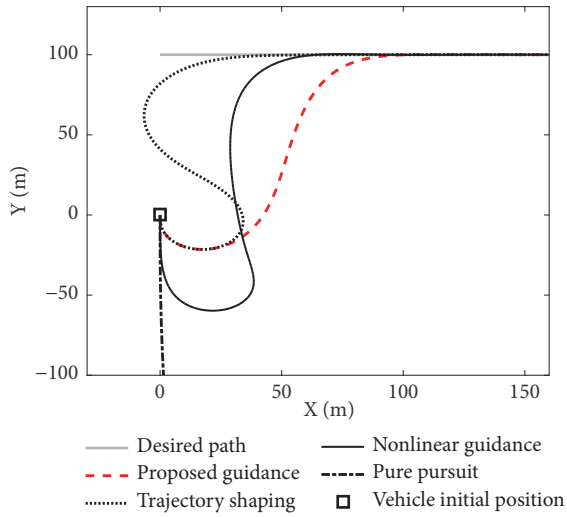
$$\boldsymbol{Q}(t) = \begin{bmatrix} \dfrac{10}{e^{0.2t_{go}}} & 0 & 0 \\ 0 & \dfrac{10}{e^{0.2t_{go}}} & 0 \\ 0 & 0 & \dfrac{10}{e^{0.2t_{go}}} \end{bmatrix}, \qquad \boldsymbol{R}(t) = e^{-t_{go}^2/0.5((t+t_{go})/8)^2} \tag{37}$$

Case 3.

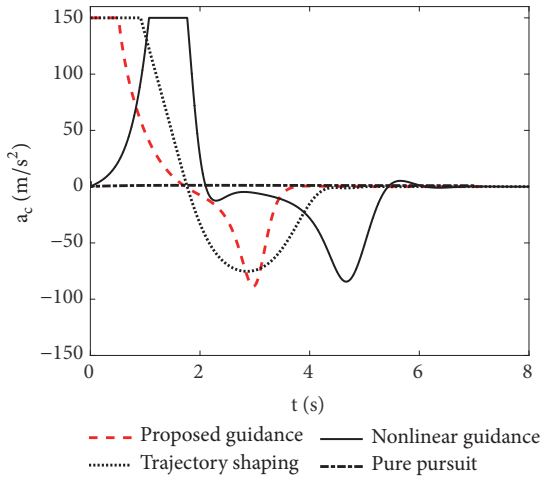
$$\boldsymbol{Q}(t) = \begin{cases} \begin{bmatrix} \dfrac{10}{e^{0.2t_{go}}} & 0 & 0 \\ 0 & \dfrac{10}{e^{0.2t_{go}}} & 0 \\ 0 & 0 & \dfrac{10}{e^{0.2t_{go}}} \end{bmatrix} & \text{if } |d| > 50\text{m} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{else} \end{cases} \tag{38}$$

$$\boldsymbol{R}(t) = \begin{cases} e^{-t_{go}^2/0.5((t+t_{go})/8)^2} & \text{if } |d| > 50\text{m} \\ 1 & \text{else} \end{cases} \tag{39}$$

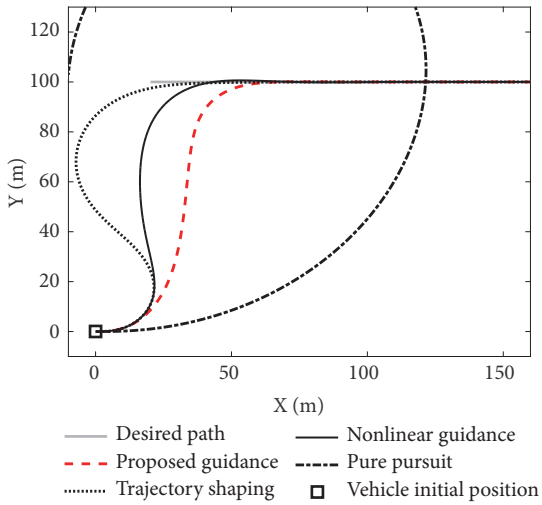
As shown in Figure 6, the proposed guidance law can successfully drive the vehicle converge to the desired path for all cases. However, different weighting functions lead to different convergence rate and thus obtain different path-following performance. For case 1, the constants $\boldsymbol{Q}$ and $\boldsymbol{R}$ do not make any special considerations on shaping the vehicle's path and acceleration profile during different phases in the transition and result in a moderate path-following performance. Case 2 performs a faster convergence during the forepart of the transition, but a slower convergence when the vehicle is approaching the desired path and an obvious overshoot. It is clearly observed that case 3 possesses both the advantages of Cases 1 and 2 and performs the best path-following performance not only in the convergence rate but also in the accuracy. Figure 7 depicts the time history of the command acceleration of each case. The switching point of the weighting functions in case 3 is clearly observed in this figure. The results of simulations show that the path-following



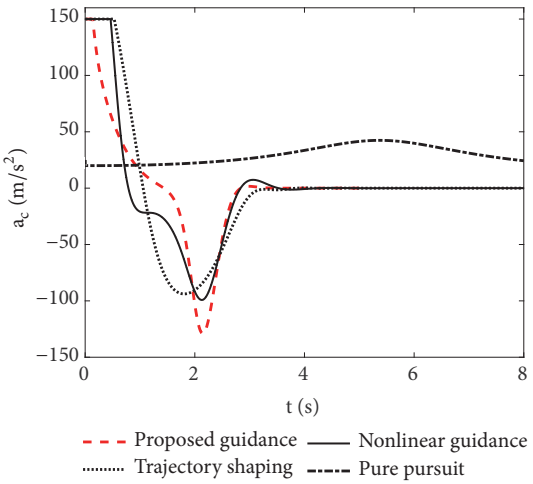
(a) Trajectories with $\gamma_m(t_0)$=-90deg



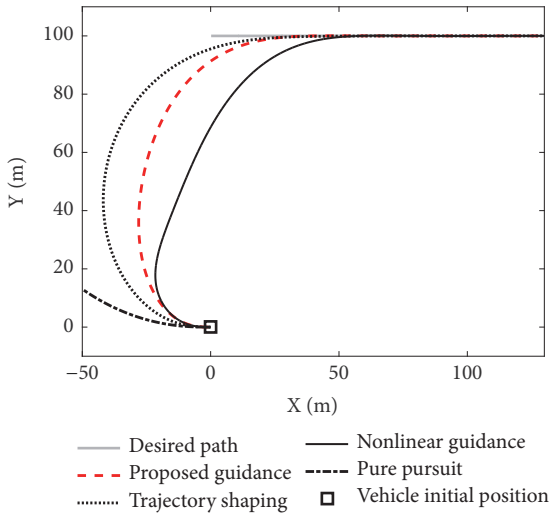
(b) Lateral command acceleration with $\gamma_m(t_0)$=-90deg



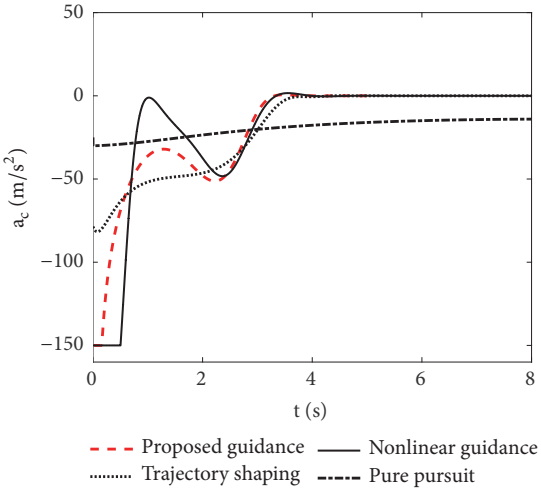
(c) Trajectories with $\gamma_m(t_0)$=0deg



(d) Lateral command acceleration with $\gamma_m(t_0)$=0deg



(e) Trajectories with $\gamma_m(t_0)$=180deg



(f) Lateral command acceleration with $\gamma_m(t_0)$=180deg

Figure 5: Straight-line following with different initial heading angles.

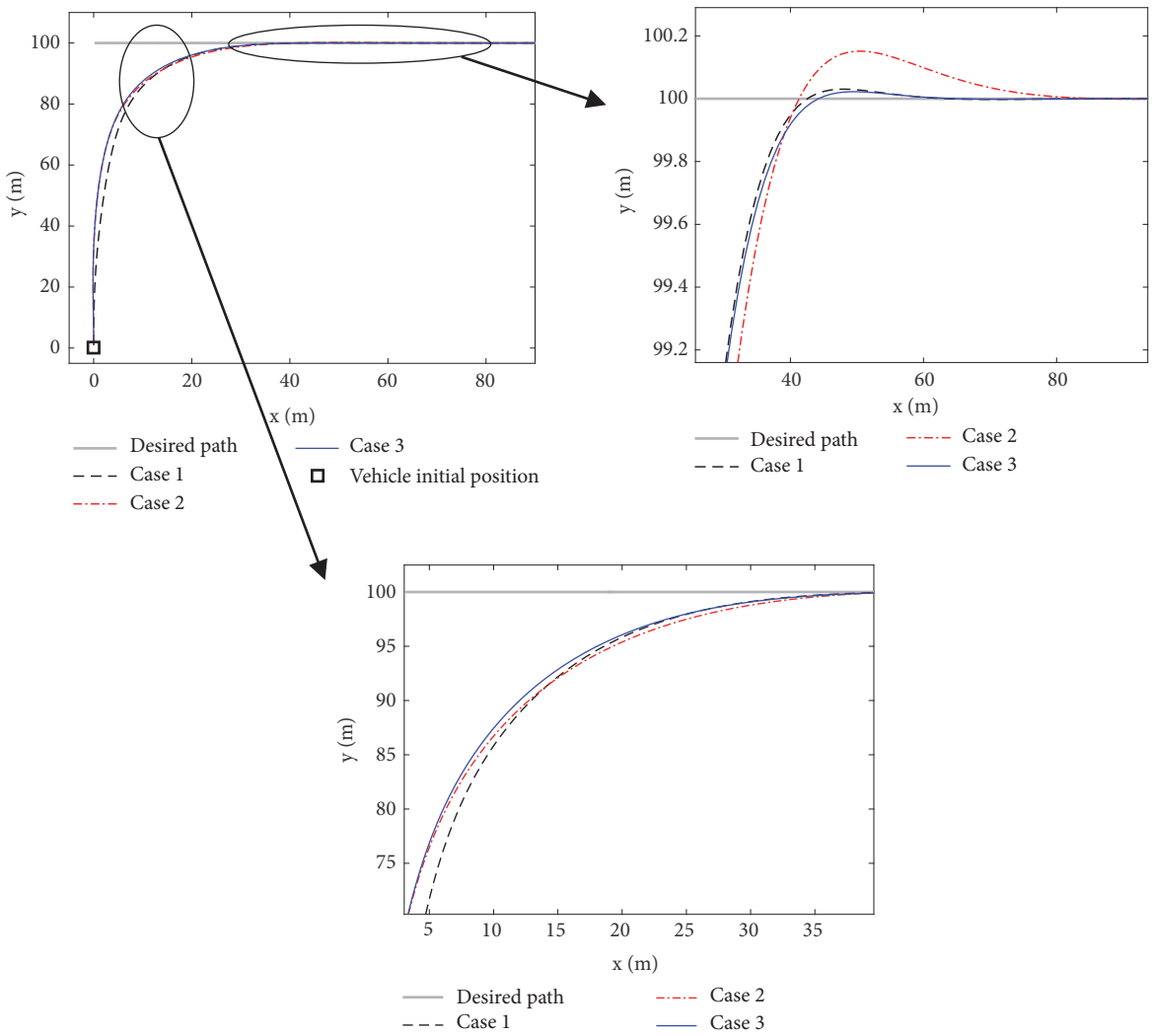


Figure 6: Straight-line following trajectories with different weighting functions.

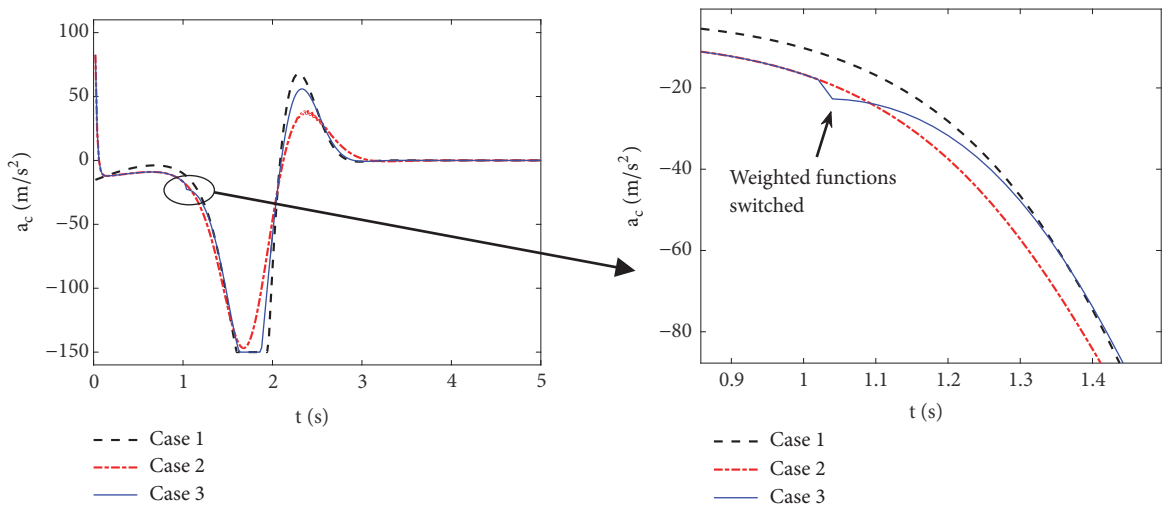


Figure 7: Lateral command acceleration with different weighting functions.

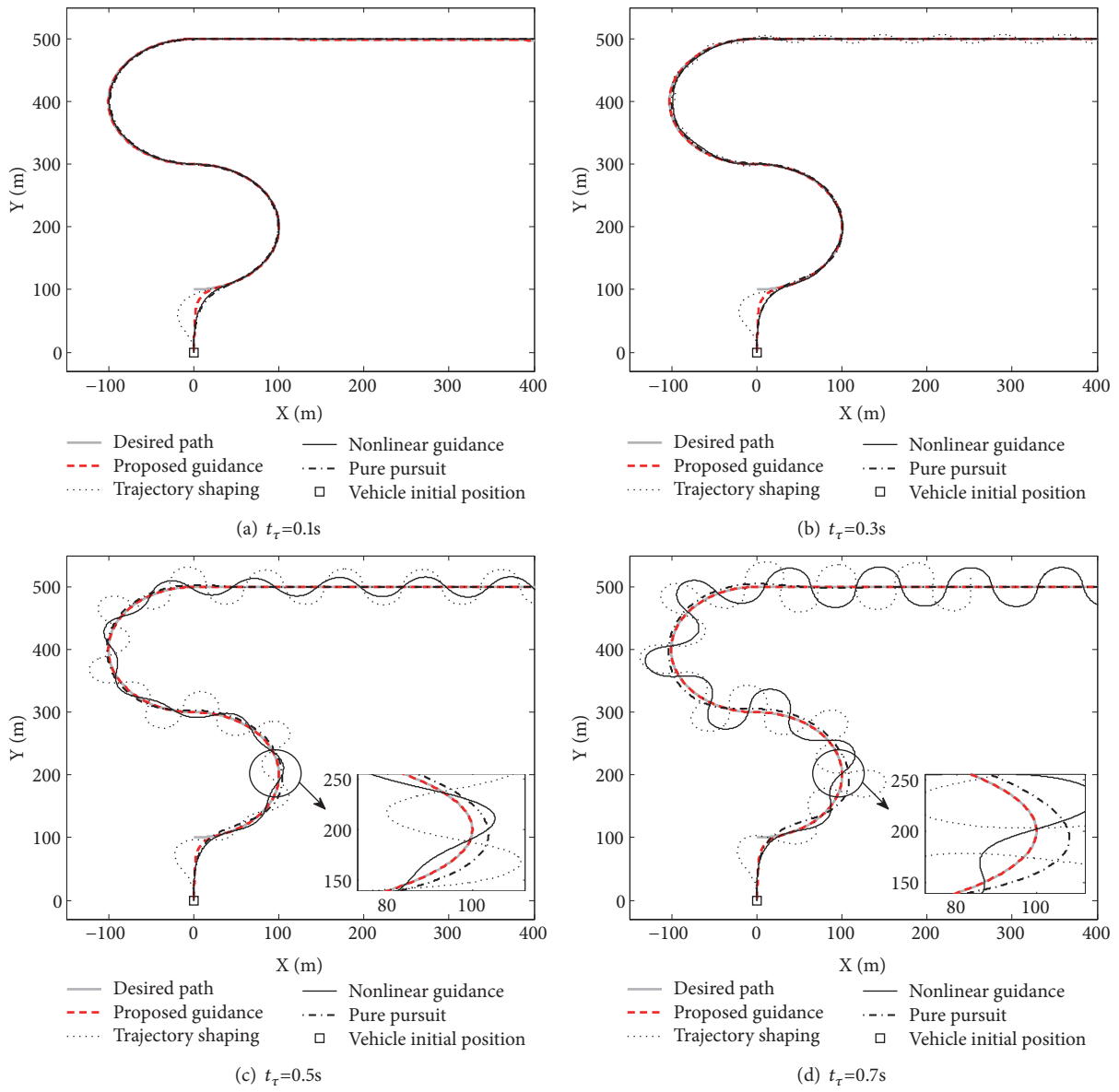


Figure 8: Composite path following with R^*=20m.

performance of the proposed method can be flexibly designed by adjusting the weighting functions, which provides more degrees of freedom in designing a path-following guidance law for various guidance objectives.

4.4. Composite Path Following. Next, a composite path consisting of counterclockwise semicircle, clockwise semicircle, and straight line is considered. The radius of both the semicircles is 100m. The performance of composite path following for different guidance laws is shown in Figure 8. For small values of t_τ, all the guidance laws can provide satisfactory path-following performances. However, as the value of t_τ increases, trajectory shaping guidance and nonlinear guidance start to oscillate and have considerable path-following errors. This phenomenon is similar to that of straight-line following. Pure pursuit is not very sensitive to t_τ and performs better than trajectory shaping guidance and nonlinear guidance, but it also has a large path-following error in curve following for large values of t_τ. The proposed guidance law is independent of t_τ and can follow the path with negligible errors all the time. Simulation results show that the proposed guidance law is more effective than the other guidance laws in both straight-line following and curve following.

4.5. Effect of Parameter R^*. As pointed out in [15], a robust path-following law should be fairly unaffected by the choice of R^*. In this study, we consider the composite path given in the previous section again and vary the value of R^* as R^*=15m, 45m, 75m, and 105m to evaluate the path-following performance of each guidance law. For R^*=15m, as shown in Figure 9(a), trajectory shaping guidance and nonlinear guidance cannot converge to the desired path and have

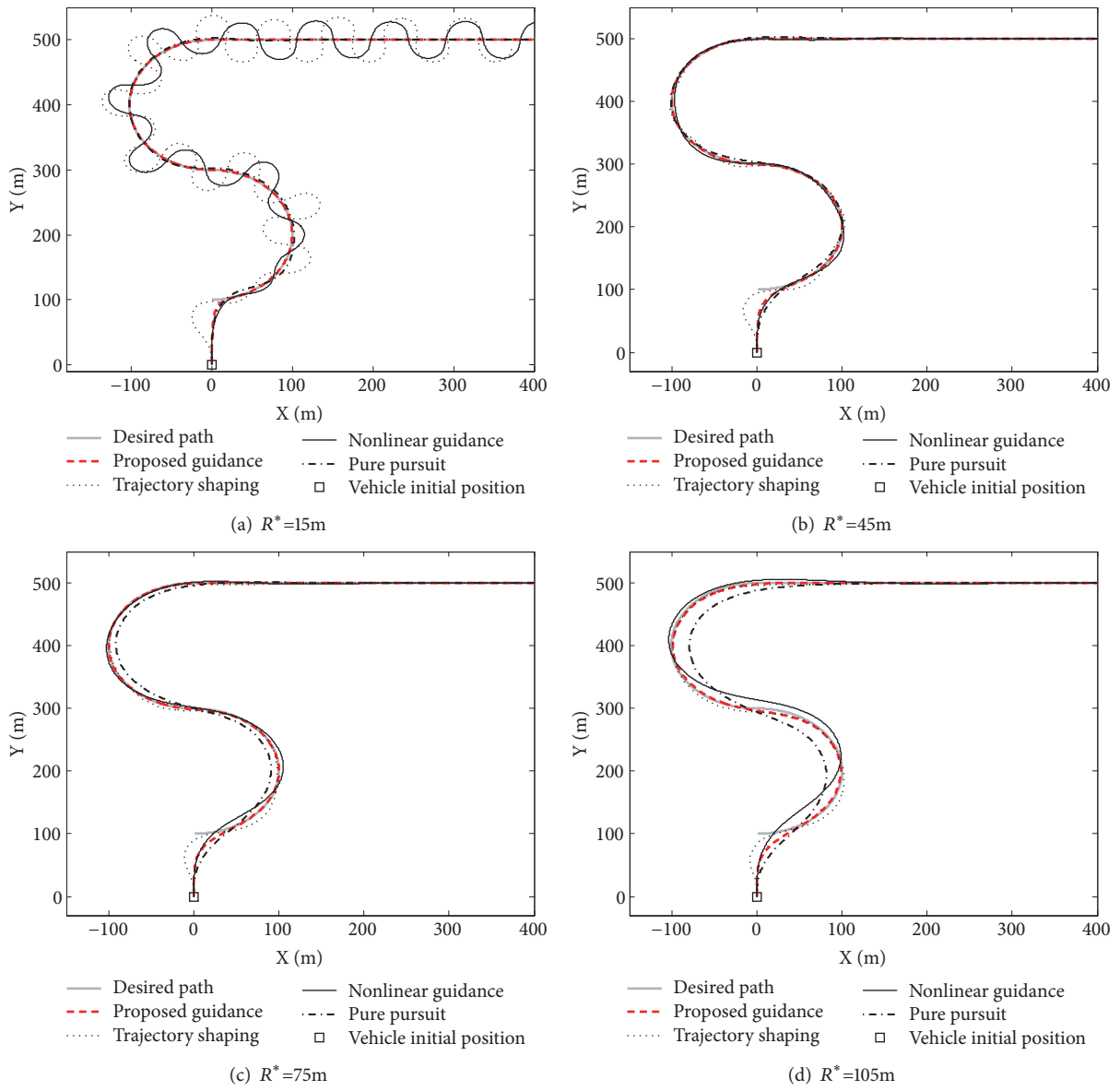
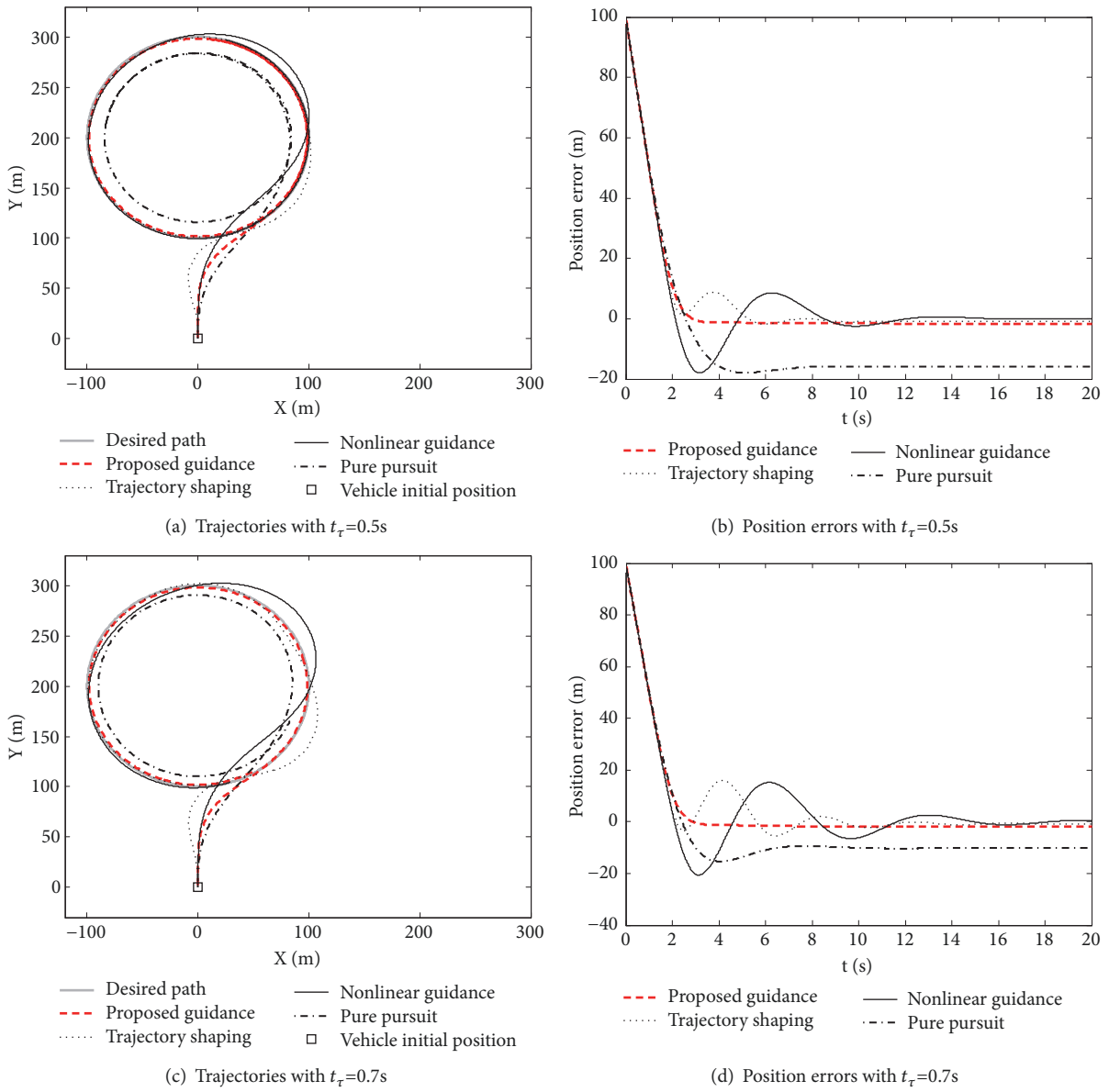


Figure 9: Trajectories with different R^* when t_τ=0.5s.

drastic oscillations around the desired path, whereas the proposed guidance and pure pursuit have negligible path-following errors. Figure 9(b) shows that the oscillations of trajectory shaping guidance and nonlinear guidance are alleviated for R^*=45m, but the proposed guidance still has the least errors. For R^*=75m, as shown in Figure 9(c), trajectory shaping guidance and nonlinear guidance provide slow convergence, and pure pursuit performance starts to deteriorate with obvious path-following errors. However, the proposed guidance still has a satisfactory path-following performance. When further increasing R^* to 105m as shown in Figure 9(d), the proposed guidance still follows the desired path with high accuracy, whereas both pure pursuit and nonlinear guidance have considerable path-following errors, and trajectory shaping guidance converges slowly. These results demonstrate that the proposed guidance law is fairly unaffected by the choice of R^*.

4.6. Convergence Rate Comparison. To further analyze the path-following performance of the proposed guidance, consider a circular path following and focus on the convergence rate. The radius of the circle is chosen as 100m, and R^*=100m. Trajectories with t_τ=0.5s are plotted in Figure 10(a), and the corresponding position errors are shown in Figure 10(b). As can be seen in Figure 10(a), the proposed guidance presents a very good following performance, whereas trajectory shaping guidance and nonlinear guidance converge to the path slowly. Moreover, pure pursuit has a steady following error because of a higher R^* and the curvature of the path. This is the inherent property of pure pursuit as shown in [15]. The settling time (1%), as shown in Figure 10(b), for the proposed guidance, trajectory shaping guidance, and nonlinear guidance is 2.68s, 5.12s, and 14.00s, respectively. Thus, the proposed guidance possesses the fastest convergence to the path compared with the other guidance laws. A similar



(a) Trajectories with t_τ=0.5s

(b) Position errors with t_τ=0.5s

(c) Trajectories with t_τ=0.7s

(d) Position errors with t_τ=0.7s

FIGURE 10: Circular path following with R^*=100m for convergence rate comparison.

study is carried out for a larger dynamic delay, t_τ=0.7s. The trajectories and corresponding position errors are presented in Figures 10(c) and 10(d), respectively. The settling time for the proposed guidance, trajectory shaping guidance, and nonlinear guidance is 2.70s, 15.05s, and 16.88s, respectively. As can be seen, an increased t_τ deteriorates the convergence rates for all of the four guidance laws. However, the proposed guidance still provides the fastest convergence to the path and is slightly affected by t_τ. Results further highlight the significant advantage of the proposed guidance law.

5. Conclusions

In this paper, a novel indirect Gauss pseudospectral method based path-following guidance law with generalized weighting functions by pursuing a virtual target is presented, where the autopilot dynamic delay is considered to compensate for the vehicle's response to the guidance command. The accuracy and effectiveness of the proposed guidance are validated by comparing with GPOPS. The real-time computational performance of the proposed method is validated on a hardware platform based on the ARM Cortex-M7 processor. Simulation results show the oscillations and instability caused by the autopilot delay and demonstrate that the proposed guidance possesses a very robust path-following ability and a fast convergence rate for different autopilot delays. Simulation also shows that the proposed approach has high performance when compared with the trajectory shaping guidance, nonlinear guidance, and pure pursuit. Additionally, according to selections of the weighting functions (even discontinuous), the proposed guidance law

can flexibly shape the path-following trajectories and the acceleration commands, providing more degree of freedoms for the whole path-following guidance design process.

Data Availability

All data generated or analyzed during this study are included in this manuscript.

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for small unmanned air vehicles," in *Proceedings of the 2006 American Control Conference*, pp. 5788–5794, IEEE, USA, June 2006.

[2] D. A. Lawrence, E. W. Frew, and W. J. Pisano, "Lyapunov vector fields for autonomous UAV flight control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA Paper 2007-6317*, USA, August 2007.

[3] Y. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*, vol. 32, pp. 4–6, John Wiley & Sons, Ltd, West Sussex, UK, 2011.

[4] Z. Liang, Q. Li, and Z. Ren, "Waypoint constrained guidance for entry vehicles," *Aerospace Science and Technology*, vol. 52, pp. 52–61, 2016.

[5] K. Yang, S. Sukkarieh, and Y. Kang, "Adaptive Nonlinear Model Predictive path tracking control for a fixed-wing unmanned aerial vehicle," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2009-5622*, USA, August 2009.

[6] D. J. Gates, "Nonlinear path following method," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 321–332, 2010.

[7] Z. Liang, Q. Li, and Z. Ren, "Virtual Terminal–Based Adaptive Predictor–Corrector Entry Guidance," *Journal of Aerospace Engineering*, vol. 30, no. 4, Article ID 04017013, 2017.

[8] J. M. Zhang, Q. Li, N. Cheng, and B. Liang, "Path-following control for fixed-wing unmanned aerial vehicles based on a virtual target," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 228, no. 1, pp. 66–76, 2012.

[9] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1092–1100, 2006.

[10] T. Yamasaki, H. Sakaida, K. Enomoto, H. Takáno, and Y. Babal, "Robust trajectory-tracking method for UAV guidance using proportional navigation," in *Proceedings of the International Conference on Control, Automation and Systems, ICCAS 2007*, pp. 1404–1409, Republic of Korea, October 2007.

[11] E. D. B. Medagoda and P. W. Gibbens, "Synthetic-waypoint guidance algorithm for following a desired flight trajectory," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 601–606, 2010.

[12] N. Cho, Y. Kim, and S. Park, "Three-dimensional nonlinear path-following guidance law based on differential geometry," in *Proceedings of the 19th IFAC World Congress*, pp. 2503–2508, Cape Town, South Africa, 2014.

[13] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, pp. 2004–4900, USA, August 2004.

[14] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path-following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.

[15] A. Ratnoo, S. Y. Hayoun, A. Granot, and T. Shima, "Path following using trajectory shaping guidance," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 1, pp. 106–116, 2015.

[16] C.-K. Ryoo, H. 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.

[17] 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.

[18] 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.

[19] I. Rusnak, "Exponential criterion-based guidance law for acceleration constrained missile and maneuvering target," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 718–721, 1996.

[20] S. Xiong, M. Wei, M. Zhao, H. Xiong, W. Wang, and B. Zhou, "Hyperbolic tangent function weighted optimal intercept angle guidance law," *Aerospace Science and Technology*, vol. 78, pp. 604–619, 2018.

[21] 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.

[22] 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.

[23] C.-H. Lee, T.-H. Kim, M.-J. Tahk, and I.-H. Whang, "Polynomial guidance laws considering terminal impact angle and acceleration constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 74–92, 2013.

[24] D. A. Benson, *A Gauss Pseudospectral Transcription for Optimal Control*, Department of Aeronautics and Astronautics, MIT, USA, Nov. 2004.

[25] P. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, NY, USA, 1984.

[26] A. V. Rao, D. A. Benson, G. T. Huntington et al., *User's Manual for GPOPS Veision 3.0*, Jan. 2010, http://vdol.mae.ufl.edu.

[27] C.-K. Ryoo, H. Cho, and M.-J. Tahk, "Closed-form solutions of optimal guidance with terminal impact angle constraint," in *Proceedings of the IEEE Conference on Control Applications*, pp. 504–509, Istanbul, Turkey, June 2003.

[28] V. Shaferman and T. Shima, "Linear quadratic guidance laws for imposing a terminal intercept angle," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1400–1412, 2008.