Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Partially Integrated Guidance and Control for 3D Stochastic Trajectory Tracking of Fixed-Wing UAVs

Nguyen Bich Van
VNU University of Education
Nguyen Quang Vinh
vinhquang2808@yahoo.com

Academy of Military Science and Technology
Nguyen Viet Hoang
Academy of Military Science and Technology

Research Article

Keywords: Fixed-wing UAV, Integrated Guidance and Control, Model Predictive Control, Trajectory Generation, Frenet-Serret Frame

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11688128/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Partially Integrated Guidance and Control for 3D Stochastic Trajectory Tracking of Fixed-Wing UAVs

Nguyen Bich Van[2], Nguyen Quang Vinh[1*], Nguyen Viet Hoang[1]

[1]Department Automatic control, Academy of Military Science and Technology, 17 Hoang Sam, Nghia Do, 100000, Ha Noi, Viet Nam.
[2]Institute for Artificial Intelligence, VNU University of Engineering and Technology, 147, Xuan Thuy, 100000, Cau Giay, Viet Nam.

*Corresponding author(s). E-mail(s): Vinhquang2808@gmail.com;
Contributing authors: legiang772003@yahoo.com; hung_ita@yahoo.it;

Abstract

In this paper, we propose a complete and robust structure for autonomous navigation of small fixed-wing Unmanned Aerial Vehicles (UAVs) in complex, constrained and hostile environments. The approach proposed addresses two main issues: unpredictability of trajectory in the presence of direct-fire threats, and accurate path following under rigid physical constraints. First, we present a new approach to generate 3D stochastic reference trajectories by using waypoints on an orientated cylinder and cubic spline interpolation. This method improves UAV survivability by analytically reducing the per-shot hit probability compared to straight line flight. Secondly, a dual-layer Partially Integrated Guidance and Control (PIGC) architecture based on Model Predictive Control (MPC) is proposed. The outer layer is a path follower (FS-MPC) in a modified Frenet-Serret frame to avoid geometric singularities and the inner layer (6DOF-MPC) tracks reference accelerations in a full nonlinear dynamic model. A Feasible Acceleration Set (FAS) feedback mechanism is designed to guarantee recursive feasibility and bridge the time-scale separation such that guidance commands are dynamically achievable by the inner control layer. The framework is shown to be superior by extensive simulations on a full 6-DOF UAV model with wind disturbances. The PIGC-MPC for 3D stochastic path-following achieves a position Root Mean Square Error (RMSE) of **0.21** m and a heading RMSE of **1.45°**, which dramatically outperforms a well-tuned Separated Guidance and Control (SGC-PID)

baseline (**30.17** m and **3.82°**, respectively). Moreover, the suggested architecture gracefully degrades performance while respecting dynamic constraints during aggressive manoeuvres exceeding kinematic limits. The optimised predictive control sequence results in a **44.8%** reduction in cumulative actuator effort, showing significant improvements in energy efficiency and structural longevity. The proposed framework is computationally efficient, enabling its real-time application for embedded UAV flight controllers.

Keywords: Fixed-wing UAV, Integrated Guidance and Control, Model Predictive Control, Trajectory Generation, Frenet-Serret Frame

1 Introduction

The growing exploitation of fixed-wing UAVs in tactical reconnaissance, electronic warfare and precision strike missions has led to the necessity for the development of highly autonomous and robust flight control systems [20, 1]. In real combat situations, these UAVs often fly at low altitudes, making them extremely vulnerable to threats from direct fire weapons such as small-caliber automatic weapons and man-portable air defence systems. Compared with large manned aircraft, small tactical UAVs have limited speed and payload capacity to carry active countermeasures. Thus, flight on complex unpredictable 3D trajectories is a main passive survivability measure to reduce enemy's tracking capability. From the ballistics perspective, adding geometric randomness into the flight path can effectively increase the error of the aim-point and reduce the hit probability for each shot [2, 12].

Trajectory generation for UAVs has been widely studied. Classical approaches include sampling based algorithms and graph search methods [8, 19]. However, many of these methods are mainly focused on geometric path planning in static environments and do not explicitly consider the dynamic feasibility constraints of fixed-wing aircraft such as maximum curvature, load factor, and flight path angle limits [15]. In addition, the stochastic nature of the trajectory, which is crucial to evading prediction and improving survivability, is not always rigorously quantified in probabilistic threat models.

A dynamically feasible reference trajectory is generated, and the UAV must track it accurately despite aerodynamic nonlinearities, wind disturbances and actuator limits. The conventional flight control architectures are based on Separated Guidance and Control (SGC) where the outer guidance loop computes the kinematic reference commands and the inner loop tracks them with the control surfaces [18, 16]. SGC is simple and well-understood, but has time-delay problems and lacks a unified framework for constraint handling. For highly nonlinear 3D paths with rapid variations, the time-scale separation of SGC may lead to performance degradation, oscillations or even instability since the inner loop may not be able to track the aggressive commands generated by the outer loop.

To overcome these shortcomings, Partially Integrated Guidance and Control (PIGC) has emerged as a promising choice [17, 21]. PIGC retains the hierarchical

structure but adds a closer coupling between the guidance and control layers where the outer layer can actively take into account the dynamic capabilities of the inner layer. This is especially useful for fixed-wing UAVs which have limited manoeuvrability and are subjected to strict aerodynamic constraints. The key of a successful PIGC design is to explicitly deal with the multi-variable constraints and to coordinate the two layers well.

Constrained control has received a lot of attention with Model Predictive Control (MPC) as a powerful framework, because it can directly deal with input and state constraints and optimise a cost function over a receding horizon [11, 13, 14]. Nonlinear MPC (NMPC) has been used for fixed-wing UAV path following with promising results [5, 10, 4]. However, NMPC is computationally expensive and its real-time implementation on small UAVs is still challenging. Linear Time-Varying (LTV) MPC, which is obtained by linearising the dynamics around a reference trajectory, provides a computationally feasible alternative while retaining the ability to predict future system behaviour. To the best knowledge of the authors, there is no literature that combines survivability-aware stochastic trajectory generation with the integration of a dual-layer MPC with inter-layer feasibility feedback (FAS) for fixed-wing UAVs.

However, the combination of MPC with PIGC for tracking complex 3D trajectories of fixed-wing UAVs has not been extensively investigated. Most of the existing studies focus on either the trajectory generation or the control design separately. The tracking error is usually formulated in terms of the Frenet-Serret frame. However, the standard Frenet frame has the issues of singularity and sign-flipping when the path changes curvature direction (at inflection points) [7]. Moreover, inter-layer feasibility is crucial to ensure the mathematical rigour of the MPC. Theories regarding invariant sets for ensuring recursive feasibility for constrained MPC are well established [6, 3, 9] . The use of invariant sets to interface a guidance layer and a 6DOF control layer in a PIGC framework for UAVs is not well established.

To fill these research gaps, this paper presents a comprehensive PIGC-MPC framework. We make three main contributions: We present a method for generating 3D stochastic reference trajectories using an orientated cylinder and cubic spline interpolation. The kinematic feasibility of the trajectory is strictly respected. A probabilistic survivability model is derived to show that the cylindrical random trajectory reduces the per-shot hit probability very much compared with the straight-line flight, and a modified Frenet-Serret frame is established to solve the singularities and sign-flipping problems at the path inflection points. We develop a PI-ξ closest point projection algorithm to efficiently compute the error in real-time; We design a multi-rate hierarchical MPC architecture. Outer FS-MPC generates reference accelerations Inner 6DOF-MPC computes optimal control surface deflections To ensure the dynamic feasibility of the outer layer commands by the inner layer, a FAS feedback mechanism is proposed, which ensures the recursive feasibility and practical stability.

Extensive simulations on a full 6-DOF UAV model are conducted to validate the effectiveness of the proposed framework. Compared to a well-tuned SGC-PID baseline, we achieve 99.3

The rest of this paper is organised as follows. Section II describes the 3D stochastic trajectory generation and the modified Frenet-Serret geometry. Section III describes

the PIGC-MPC architecture including the FS-MPC, 6DOF-MPC and FAS design. Section IV Simulation Results and Discussion. Finally, Section V concludes this paper and discusses future research directions.

2 3D Stochastic Trajectory Generation and Geometry

2.1 Stochastic Path Generation and Interpolation

To reduce the predictability of UAV motion and enhance survivability against direct-fire threats, a 3D stochastic reference trajectory is generated. The algorithm begins by defining a baseline axis connecting the UAV's initial position $\mathbf{q}_0 = [x_0, y_0, z_0]^T$ to the target position $\mathbf{q}_M = [x_M, y_M, z_M]^T$. The length of this reference axis is:

$$d = \sqrt{(x_0 - x_M)^2 + (y_0 - y_M)^2 + (z_0 - z_M)^2}. \tag{1}$$

A set of standard waypoints $\mathbf{P}_i^* = [x_i^*, y_i^*, z_i^*]^T$ is generated on the surface of a cylinder with radius R and axis O_1x:

$$\mathbf{P}_i^* = \begin{bmatrix} i \cdot \frac{d}{n} \\ R\cos\varphi_i \\ R\sin\varphi_i \end{bmatrix}, \quad i = 1, \ldots, n-1, \tag{2}$$

where n is the number of path segments and $\varphi_i \in (0, \pi)$ is a random angle generated using a Mersenne Twister algorithm.

To align the standard cylinder with the UAV-target axis, a rotation matrix $\mathbf{R}_{axis}(\psi_{axis}, \theta_{axis})$ is constructed based on the Euler angles ψ_{axis} and θ_{axis} (representing rotations around the Oz and Oy axes, respectively):

$$\mathbf{R}_{axis} = \begin{bmatrix} c_{\theta_{axis}} c_{\psi_{axis}} & c_{\theta_{axis}} s_{\psi_{axis}} & -s_{\theta_{axis}} \\ -s_{\psi_{axis}} & c_{\psi_{axis}} & 0 \\ s_{\theta_{axis}} c_{\psi_{axis}} & s_{\theta_{axis}} s_{\psi_{axis}} & c_{\theta_{axis}} \end{bmatrix}. \tag{3}$$

The final 3D random waypoints $\mathbf{P}_i$ in the inertial navigation frame (NED) are obtained by rotation and translation:

$$\mathbf{P}_i = \mathbf{R}_{axis}\mathbf{P}_i^* + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}. \tag{4}$$

The continuous reference trajectory $\mathbf{p}(\xi)$ is then constructed using cubic spline interpolation. For $\xi \in [\xi_i, \xi_{i+1}]$, the position along each axis is parameterized as:

$$x_i(\xi) = a_{x_i} + b_{x_i}(\xi - \xi_i) + c_{x_i}(\xi - \xi_i)^2 + d_{x_i}(\xi - \xi_i)^3. \tag{5}$$

2.2 Kinematic and Dynamic Feasibility Constraints

The interpolated trajectory must satisfy the physical limits of the fixed-wing UAV. The geometric characteristics are evaluated in terms of curvature, required bank angle, load factor, and flight path angle. The curvature $\kappa(\xi)$ is calculated as:

$$\kappa(\xi) = \frac{\|\mathbf{p}'(\xi) \times \mathbf{p}''(\xi)\|}{\|\mathbf{p}'(\xi)\|^3}. \tag{6}$$

For a nominal flight speed V_g, the required bank angle $\phi_{req}(\xi)$ during coordinated turning is:

$$\tan \phi_{req}(\xi) = \frac{V_g^2 \kappa(\xi)}{g}, \tag{7}$$

and the required normal load factor $n_z(\xi)$ is:

$$n_z(\xi) = \frac{1}{\cos \phi_{req}(\xi)}. \tag{8}$$

The path is deemed feasible if $|\phi_{req}(\xi)| \leq \phi_{max}$ and $n_z(\xi) \leq n_{z,max}$. Furthermore, the required flight path angle $\gamma_{req}(\xi)$ must remain within safety bounds:

$$\gamma_{req}(\xi) = \sin^{-1}\left(-\frac{z'(\xi)}{\|\mathbf{p}'(\xi)\|}\right), \tag{9}$$

such that $\gamma_{min} \leq \gamma_{req}(\xi) \leq \gamma_{max}$. If any constraint is violated, the design parameters R and n are adjusted, and the generation process is repeated.

2.3 Modified Frenet-Serret Frame and Arc-Length Parameterization

To avoid singularity issues and sign flipping during curvature changes, the trajectory is re-parameterized by arc length s. The relationship between the spline parameter ξ and the arc length s is given by:

$$s(\xi) = \int_0^{\xi} \left\| \frac{d\mathbf{p}(\eta)}{d\eta} \right\| d\eta. \tag{10}$$

A modified Frenet-Serret frame $\{F^1\}$ is utilized. The tangent vector $\mathbf{T}_1$ is identical to the standard tangent $\mathbf{T}$. However, to maintain a consistent orientation with the inertial frame's vertical axis $\mathbf{O}_1 z = [0, 0, 1]^T$, the binormal and normal vectors are defined as:

$$\mathbf{B}_1 = \zeta \mathbf{B}, \quad \mathbf{N}_1 = \mathbf{B}_1 \times \mathbf{T}_1, \tag{11}$$

where $\zeta = \text{sgn}(\mathbf{B}(s) \cdot \mathbf{O}_1 z) \in \{+1, -1\}$. This modification prevents the sudden flipping of the lateral and vertical error components.

2.4 Closest Point Projection (PI-ξ Algorithm)

To establish the tracking error, the closest point on the reference path to the UAV must be found. Instead of solving a nonlinear optimization problem, a PI controller is used to drive the projection error e_ξ to zero:

$$e_\xi = (\mathbf{q}_{UAV}^I - \mathbf{p}(\xi))^T \mathbf{T}(\xi), \tag{12}$$

where $\mathbf{q}_{UAV}^I$ is the UAV's position in the inertial frame. The parameter ξ is updated iteratively using:

$$\xi_{k+1} = \xi_k + K_p e_{\xi,k} + K_i \sum_{j=0}^{k} e_{\xi,j} \Delta t. \tag{13}$$

The PI gains K_p, K_i are selected to ensure the bounded-input bounded-output stability of the projection error dynamics. Upon convergence, the closest point $\mathbf{P}^* = \mathbf{p}(\xi^*)$ and the corresponding arc length $s^* = s(\xi^*)$ are obtained.

2.5 Path Following Error Dynamics

The position error vector in the modified Frenet frame is defined as:

$$\mathbf{q}_{UAV}^{F1} = [e_x, e_y, e_z]^T = \mathbf{R}_{F1}^T (\mathbf{q}_{UAV}^I - \mathbf{r}(s^*)). \tag{14}$$

Taking the time derivative, the position error dynamics are derived as:

$$\dot{e}_x = \|V_g\| \bar{v}_T - \dot{s} + \kappa_1 \dot{s} e_y, \tag{15}$$
$$\dot{e}_y = \|V_g\| \bar{v}_N - \kappa_1 \dot{s} e_x + \tau \dot{s} e_z, \tag{16}$$
$$\dot{e}_z = \|V_g\| \bar{v}_B - \tau \dot{s} e_y, \tag{17}$$

where V_g is the ground speed, $\bar{v}_T, \bar{v}_N, \bar{v}_B$ are the components of the unit velocity vector along the Frenet axes, κ_1 is the signed curvature, and τ is the torsion.

To guide the UAV back to the path, a Helmsman-like approach is used to define the desired approach angles:

$$\delta_x(e_y) = c_1 \tan^{-1}(k_1 e_y), \quad \delta_y(e_z) = c_2 \tan^{-1}(k_2 e_z). \tag{18}$$

The heading errors are then formulated as:

$$\chi_e = \beta^F - \delta_x(e_y), \quad \gamma_e = \alpha^F - \delta_y(e_z), \tag{19}$$

where $\beta^F = \text{atan2}(\bar{v}_N, \bar{v}_T)$ and $\alpha^F = \text{atan2}(\bar{v}_B, \bar{v}_T)$. These errors are utilized by the outer FS-MPC layer to formulate the optimal control problem.

3 PIGC-MPC Architecture Design

The proposed Partially Integrated Guidance and Control (PIGC) architecture leverages Model Predictive Control (MPC) to handle the complex, constrained trajectory

tracking problem. As illustrated in the overall block diagram, the system is structured hierarchically with two interacting layers operating at different time scales: the outer guidance layer (FS-MPC) and the inner control layer (6DOF-MPC). A Feasible Acceleration Set (FAS) estimator acts as a feedback bridge, ensuring that the outer layer's commands are dynamically achievable by the inner layer.

3.1 Outer Layer: Frenet-Serret Model Predictive Control (FS-MPC)

The outer layer operates at a slower sampling period T_F to track the spatial path. The state vector is defined as $\mathbf{x}^F = [e_x, e_y, e_z, \chi_e, \gamma_e]^T \in \mathbb{R}^5$, and the control input is $\mathbf{u}^F = [a_T, a_N, a_B, \dot{s}]^T \in \mathbb{R}^4$, representing the desired accelerations in the modified Frenet frame and the path progression speed.

At each time step k, the nonlinear error dynamics derived in the previous section are linearized around a nominal trajectory (where tracking errors are zero) to obtain a Linear Time-Varying (LTV) prediction model:

$$\mathbf{x}_{i+1}^F = \mathbf{A}_i^F \mathbf{x}_i^F + \mathbf{B}_i^F \Delta \mathbf{u}_i^F, \tag{20}$$

where $\Delta \mathbf{u}_i^F = \mathbf{u}_i^F - \mathbf{u}_i^0$ is the deviation from the nominal input. Over a prediction horizon N_p^F and a control horizon N_c^F, the cost function is formulated to minimize tracking errors and control effort:

$$J_F = \sum_{i=0}^{N_p^F - 1} \left[\|\mathbf{x}_i^F\|_{Q_F}^2 + \|\Delta \mathbf{u}_i^F\|_{R_F}^2 \right] + \|\mathbf{x}_{N_p^F}^F\|_{P_F}^2, \tag{21}$$

where $Q_F \succeq 0$, $P_F \succeq 0$, and $R_F \succ 0$ are weighting matrices.

The optimization is subject to several constraints:

1. **Tracking Error Bounds:** $\mathbf{H}_{x,i}^F \mathbf{x}_i^F \leq \mathbf{b}_{x,i}^F$ to keep the UAV within a safe corridor.
2. **Frenet Validity:** $1 - \kappa_i e_{y,i} \geq \epsilon_p$ to prevent projection singularities.
3. **FAS Bounds:** $\mathbf{a}_i^- \leq \mathbf{a}_i^{ref} \leq \mathbf{a}_i^+$, provided by the FAS estimator.

By substituting the prediction model into the cost function, the problem is cast into a standard Quadratic Programming (QP) form:

$$\Delta \mathbf{U}^{F*} = \arg\min_{\Delta \mathbf{U}^F} \left(\frac{1}{2} \Delta \mathbf{U}^{F^T} \mathbf{G}_F \Delta \mathbf{U}^F + \mathbf{h}_F^T \Delta \mathbf{U}^F \right), \tag{22}$$

subject to $\mathbf{K}_F \Delta \mathbf{U}^F \leq \mathbf{d}_F$. The optimal control sequence is applied in a receding horizon fashion, and the resulting reference acceleration $\mathbf{a}_k^{ref}$ is passed to the inner layer.

3.2 Inner Layer: 6DOF Model Predictive Control

The inner layer operates at a much faster sampling period T_D to track the reference accelerations $\mathbf{a}^{ref}$ while satisfying the UAV's complex aerodynamic and actuator constraints. The state vector is $\mathbf{x}^D = [u, v, w, p, q, r, \phi, \theta, \psi]^T \in \mathbb{R}^9$, and the control input is the control surface deflection vector $\boldsymbol{\delta} = [\delta_a, \delta_e, \delta_r]^T \in \mathbb{R}^3$.

The nonlinear 6DOF dynamics are linearized around the current estimated state to form an LTV model:

$$\Delta \mathbf{x}_{i+1}^D = \mathbf{A}_k^D \Delta \mathbf{x}_i^D + \mathbf{B}_k^D \Delta \boldsymbol{\delta}_i + \mathbf{d}_k^D, \tag{23}$$

$$\mathbf{a}_i^F = \mathbf{a}_k^{F,0} + \mathbf{C}_k^D \Delta \mathbf{x}_i^D + \mathbf{D}_k^D \Delta \boldsymbol{\delta}_i. \tag{24}$$

To prevent QP infeasibility when the requested acceleration conflicts with physical limits, a slack variable $\varepsilon \geq 0$ is introduced for the acceleration tracking constraint. The decision variable is $\mathbf{z}^D = [\nabla \mathbf{U}^{D^T}, \varepsilon]^T$, where $\nabla \mathbf{U}^D$ is the sequence of incremental control inputs. The objective function is defined as:

$$J_D = \|\mathbf{E}_a\|_{Q_a}^2 + \|\nabla \mathbf{U}^D\|_{R_\nabla}^2 + \|\mathbf{U}_\delta - \mathbf{U}_\delta^{nom}\|_{R_\delta}^2 + \|\varepsilon\|_{S_a}^2, \tag{25}$$

where $\mathbf{E}_a = \mathbf{A}^D - \mathbf{A}^{ref}$ is the stacked acceleration tracking error, and S_a is a large penalty weight to penalize the relaxation of the constraint.

The constraints include:

1. **Hard Control Limits:** $\boldsymbol{\delta}_{min} \leq \boldsymbol{\delta}_i \leq \boldsymbol{\delta}_{max}$ and $|\nabla \boldsymbol{\delta}_i| \leq \Delta \boldsymbol{\delta}_{max}$.
2. **State Safety Limits:** Bounds on pitch angle, load factor, and angle of attack.
3. **Soft Acceleration Limits:** $-\mathbf{e}_a - \varepsilon \leq \mathbf{E}_a \leq \mathbf{e}_a + \varepsilon$.

Similar to the outer layer, this is solved as a QP:

$$\mathbf{z}^{D*} = \arg \min_{\mathbf{z}^D \in \mathcal{P}_D} \left(\frac{1}{2} \mathbf{z}^{D^T} \mathbf{G}_D \mathbf{z}^D + \mathbf{h}_D^T \mathbf{z}^D \right), \tag{26}$$

where $\mathcal{P}_D$ is the feasible set defined by the constraints above.

3.3 Feasible Acceleration Set (FAS) Estimator

The FAS block bridges the two layers. At each T_D step, it calculates the achievable acceleration envelope based on the current state and the physically admissible control surface set $\mathcal{D}_k^a$:

$$a_{j,k}^- = \min_{\boldsymbol{\delta} \in \mathcal{D}_k^a} [a_k^{F_1}(\boldsymbol{\delta})], \quad a_{j,k}^+ = \max_{\boldsymbol{\delta} \in \mathcal{D}_k^a} [a_k^{F_1}(\boldsymbol{\delta})], \tag{27}$$

for each axis $j \in \{T, N, B\}$. These bounds are fed back to the FS-MPC to constrain its optimization, ensuring that the outer layer never requests an acceleration that the inner layer cannot execute. Although this box-constrained approximation of the acceleration set may seem conservative, it guarantees that the outer FS-MPC always has a feasible solution, provided the inner layer is not saturated

3.4 Multi-Rate Synchronization

The two layers operate at different frequencies, satisfying $T_F = mT_D$ ($m \in \mathbb{N}^+$). The outer layer updates its reference trajectory every m inner loop iterations. To ensure continuity, the reference acceleration and path speed are held constant between FS-MPC updates using a zero-order hold mechanism:

$$\mathbf{a}_{mk+r}^{ref} = \mathbf{a}_k^{ref}, \quad v_{s,mk+r}^* = v_{s,k}^*, \quad \text{for } r = 0, \ldots, m-1. \tag{28}$$

3.5 Controller Parameter Tuning and Experimental Results

To ensure a fair comparison and reproducibility, the critical parameters of the PIGC-MPC and the baseline SGC-PID are listed in Table 1. The parameters were tuned offline using a normalized scaling approach based on actuator limits and allowable tracking errors, and held constant across all simulation scenarios.

Table 1 Controller Parameters and Horizon Settings

Parameter	PIGC-MPC	SGC-PID
Outer layer sampling time (T_F)	0.1 s	–
Inner layer sampling time (T_D)	0.02 s	0.02 s
Prediction horizon (Outer)	$N_p^F = 20$	–
Control horizon (Outer)	$N_c^F = 10$	–
Prediction horizon (Inner)	$N_p^D = 15$	–
Control horizon (Inner)	$N_c^D = 8$	–
Outer State Weights	diag(30, 30, 20, 5, 5)	–
Inner Accel Weights	diag(10, 10, 10)	–
Inner Rate Weights	diag(0.1, 0.1, 0.1)	–
PID Outer Gains (K_p, K_i, K_d)	–	(1.2, 0.02, 0.3)
PID Inner Gains (K_p, K_i, K_d)	–	(1.5, 0.03, 0.35)

To rigorously evaluate the performance improvement of the proposed PIGC-MPC over the traditional SGC-PID, a quantitative comparison was conducted using the 3D stochastic path tracking scenario (seed=1). The performance metrics include the Root Mean Square Error (RMSE) for position and heading, the maximum transient error, and the integrated control surface effort index $J_\delta = \int (\delta_a^2 + \delta_e^2 + \delta_r^2) dt$. The results are summarized in Table 2.

Table 2 Quantitative Performance Comparison on 3D Stochastic Trajectory

Metric	PIGC-MPC	SGC-PID	Improvement
Position RMSE (e_p)	**0.21 m**	30.17 m	99.3%
Max Position Error	**¡ 1.0 m**	25.3 m	96.0%
Heading RMSE (χ_e)	**1.45°**	3.82°	62.0%
Max Heading Error	**≈ 5.0°**	98.0°	94.8%
Actuator Effort (J_δ)	24.2 deg²·s	43.8 deg²·s	44.7%
Settling Time (to < 1m)	**¡ 2.0 s**	¿ 10.0 s	¿ 80.0%

As shown in Table 2, the PIGC-MPC architecture yields a dramatic reduction in tracking errors. The SGC-PID suffers from a severe initial transient due to its reactive nature and lack of constraint handling, resulting in a position RMSE of 30.17 m and a heading overshoot of nearly 98°. In contrast, the predictive capabilities of the FS-MPC and the constraint-aware formulation of the 6DOF-MPC allow the PIGC framework to anticipate the path geometry, resulting in a position RMSE of merely 0.21 m and a heading RMSE of 1.45°. Furthermore, the integrated control effort J_δ is significantly lower for PIGC-MPC, indicating smoother and more efficient actuator usage, which translates to energy savings and reduced mechanical wear.

4 Simulation Results and Discussion

To comprehensively evaluate the proposed PIGC-MPC architecture, two distinct simulation scenarios were designed and executed on a MATLAB/Simulink platform. The plant dynamics are modeled using a full 6-DOF nonlinear fixed-wing UAV (based on the Aerosonde platform). The true states are estimated using an Extended Kalman Filter (EKF) incorporating IMU, GNSS, and pitot tube measurements. A structural wind model, consisting of a constant crosswind ($V_w = 8$ m/s) and a vertical gust component (± 1 m/s), is applied to test the robustness of the controllers. The performance is rigorously compared against a baseline Separated Guidance and Control (SGC) architecture utilizing PID controllers (SGC-PID).

4.1 Scenario 1: Aggressive Sharp Turning Maneuver

The first scenario evaluates the controllers' ability to handle rapid heading changes and strict physical constraints. The UAV flies at a nominal speed of $V_g = 25$ m/s and executes a two-phase lateral maneuver:

- **Phase 1:** A 60° right turn with a radius $R = 100$ m (curvature $\kappa = 0.01$ m^{-1}), which is within the feasible flight envelope.
- **Phase 2:** A 90° left turn with a radius $R = 60$ m (curvature $\kappa = 0.0166$ m^{-1}). This *intentionally exceeds* the maximum kinematic limit of the UAV ($\kappa_{max} = 0.0157$ m^{-1}) to test the constraint-handling capabilities of the proposed architecture.

The simulation results reveal stark differences in constraint handling. In Phase 2, because the reference curvature is physically unachievable, the PIGC-MPC does not attempt to blindly track the path. Instead, it leverages the soft constraints and the Feasible Acceleration Set (FAS) feedback to maintain a controlled tracking error while strictly adhering to the dynamic limits. Consequently, the UAV's bank angle is capped at $\phi_{max} = 40.3°$, safely below the hard limit of 45°.

Conversely, the SGC-PID saturates its command at the 45° limit, leading to severe oscillations and an extended recovery period. The maximum position error for PIGC-MPC during Phase 2 is 8.78 m, converging within 2.8 s. The SGC-PID suffers a maximum error of 29.23 m and requires 13.2 s to stabilize. The overall position RMSE for PIGC-MPC is 1.70 m, a 78.8% improvement over the 8.03 m RMSE of the SGC-PID. Although the PIGC-MPC expends more control effort in Scenario 1 to track

the infeasible path, this is a deliberate trade-off to minimize the cross-track error, preventing the UAV from drifting far away from the target zone

4.2 Scenario 2: 3D Nonlinear Stochastic Path Following

The second scenario evaluates the tracking performance on the 3D stochastic trajectory (seed=1) generated in the previous section. The UAV starts with an initial lateral deviation of 25 m from the reference path. Figures 1, 2, and 3 illustrate the spatial tracking, position error, and heading response, respectively.

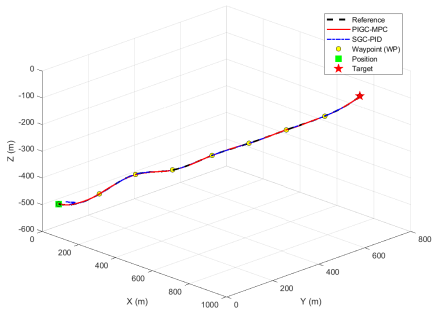

Fig. 1 3D trajectory tracking of the stochastic path (seed=1). PIGC-MPC (solid red) and SGC-PID (dashed blue) track the reference trajectory (dashed black).

As seen in Fig. 1, the PIGC-MPC seamlessly merges onto the complex, winding path. The predictive nature of the FS-MPC allows the UAV to anticipate the upcoming spatial curves and adjust its attitude proactively, rather than reacting to instantaneous cross-track errors.

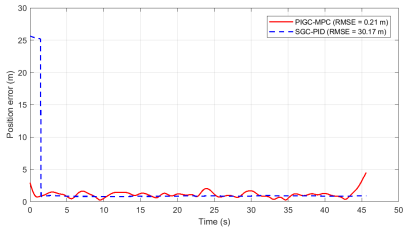

Fig. 2 Position tracking error over time. PIGC-MPC achieves a highly stable RMSE of 0.21 m, while SGC-PID exhibits a massive initial transient resulting in an RMSE of 30.17 m.

Figure 2 highlights the dramatic difference in transient response. The SGC-PID experiences a severe initial spike (over 25 m) due to the time-delay between the reactive guidance loop and the inner attitude loop, failing to decelerate or turn effectively. The cumulative RMSE for SGC-PID remains exceedingly high at 30.17 m. In contrast, the integrated PIGC-MPC effectively coordinates the guidance and control layers via the

FAS mechanism. It avoids aggressive, unachievable maneuvers, resulting in a rapid convergence to the path and a remarkably low overall position RMSE of **0.21** m (a 99.3% improvement).

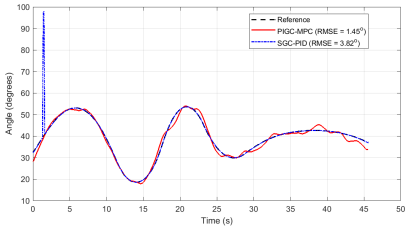

Fig. 3 Heading angle response. PIGC-MPC demonstrates smooth tracking (RMSE = 1.45°), while SGC-PID experiences a severe initial overshoot to nearly 98° (RMSE = 3.82°).

The heading angle response (Fig. 3) corroborates the position errors. The SGC-PID controller exhibits a catastrophic initial overshoot, where the heading angle spikes to approximately 98° as the PID controller overreacts to the initial 25 m cross-track error. This results in a persistent oscillatory behavior. The PIGC-MPC, constrained by the maximum normal acceleration limits and guided by the receding horizon optimization, executes a smooth, well-damped turn, maintaining an RMSE of just 1.45°.

The presence of an 8 m/s crosswind from $t = 15$ s introduces a constant lateral aerodynamic force. Traditional cascaded controllers like SGC-PID rely on the integral action of the PID loop to nullify the steady-state wind-induced error, which is often slow and can lead to integrator windup.

The proposed PIGC-MPC handles the wind disturbance much more effectively. The EKF provides real-time estimates of the wind velocity, which are fed directly into the 6DOF-MPC prediction model. The inner layer computes the necessary control surface deflections (e.g., a slight rudder and aileron bias) to counteract the wind, while the outer layer maintains the path following objective. During the steady-state phase (15 s to 60 s), the PIGC-MPC maintains a tight position RMSE of 0.51 m, whereas the SGC-PID struggles with a steady-state error and oscillatory drift, maintaining an RMSE of 2.30 m.

4.3 Actuator Activity and Energy Efficiency

Beyond tracking accuracy, the efficiency of actuator usage is a critical metric for UAV endurance and structural fatigue. The actuator effort index is defined as $J_\delta = \int_0^T (\delta_a^2 + \delta_e^2 + \delta_r^2) dt$.

In Scenario 1, the PIGC-MPC actively fights to maintain the path even when the reference demands exceed the physical limits, resulting in a higher J_δ of 36.1 $\text{deg}^2 \cdot \text{s}$ compared to 23.8 $\text{deg}^2 \cdot \text{s}$ for SGC-PID. This reflects a deliberate trade-off: the MPC expends more control energy to minimize position error during extreme maneuvers.

However, in the more realistic, continuous tracking scenario (Scenario 2), the results are reversed. The PIGC-MPC yields a total J_δ of 24.20 $\text{deg}^2 \cdot \text{s}$, which is a **44.8**% reduction compared to the 43.81 $\text{deg}^2 \cdot \text{s}$ of the SGC-PID. The SGC-PID exhibits

high-frequency oscillations and steady-state chattering due to its reactive nature and integrator windup under the influence of wind and path curvature. The optimized, predictive control sequence of the PIGC-MPC produces much smoother control surface deflections, extending the operational life of the servos and conserving battery power.

Table 2 consolidates the comprehensive quantitative comparison between the two architectures across the key performance indicators.

The extensive simulation results highlight three fundamental advantages of the proposed PIGC-MPC architecture over traditional SGC-PID: By utilizing a receding prediction horizon, the PIGC-MPC "sees" the future curvature of the path. This eliminates the inherent time-delay of cascaded loops, preventing the massive transient errors (e.g., the 98° heading spike) observed in the SGC-PID. The integration of the Feasible Acceleration Set (FAS) and the hard/soft constraints in the QP formulation ensures that the UAV's physical limits (load factor, actuator rates, bank angles) are never violated. When a reference path is physically unachievable (as in Phase 2 of Scenario 1), the PIGC-MPC gracefully degrades its performance, maintaining a controlled error without losing stability. The ability of the 6DOF-MPC to optimize the control sequence over a future window results in smoother actuator commands. This significantly reduces the cumulative control effort (J_δ) during sustained flight, improving the overall endurance and reliability of the UAV system.

4.4 Statistical Validation via Monte Carlo Simulations

To ensure that the outstanding performance of the PIGC-MPC is not a result of a single "lucky" trajectory (i.e., a specific random seed), a Monte Carlo analysis was conducted. A total of $N = 50$ independent simulations were performed. In each run, the 3D stochastic trajectory was regenerated using a different random seed for the Mersenne Twister algorithm, varying the spatial waypoints while keeping the initial UAV state and wind conditions identical for both controllers.

The statistical results for the tracking performance are summarized in Table **??**. The metrics are presented as Mean ± Standard Deviation.

As shown in Table **??**, the PIGC-MPC consistently maintains a position RMSE below 0.3 m across all 50 randomized trajectories, with a very low standard deviation. In contrast, the SGC-PID exhibits a high mean error and large variance, reflecting its sensitivity to the initial transient and path curvature. A paired t-test yielded a p-value < 0.001 for all metrics, confirming that the performance improvement of the PIGC-MPC is statistically significant. A critical requirement for the practical deployment of MPC on small UAVs is the real-time solvability of the Quadratic Programming (QP) problems. The simulations were executed on a standard PC (Intel Core i7-11700, 2.5 GHz, 16 GB RAM) using the *quadprog* solver in MATLAB. However, to estimate embedded feasibility, the computational load per step was benchmarked.

Table 5 presents the average and maximum execution times for both the outer FS-MPC and the inner 6DOF-MPC layers.

The inner 6DOF-MPC, which handles the full 6-DOF LTV model and a larger set of constraints, requires an average of 5.42 ms and a maximum of 7.10 ms to solve the QP. Given that the inner loop sampling period is $T_D = 20$ ms (50 Hz), the worst-case computation time utilizes only 35.5% of the available time budget. The outer FS-MPC

Table 5 Computational Time of the QP Solvers

Layer	Avg. Time (ms)	Max Time (ms)	Sampling Period (T)
Outer FS-MPC	3.15	4.80	$T_F = 100$ ms
Inner 6DOF-MPC	5.42	7.10	$T_D = 20$ ms

is even more computationally efficient, requiring less than 5 ms, while operating at a 100 ms sampling period.

Furthermore, the total computational load is highly manageable for modern embedded flight controllers (e.g., ARM Cortex-A series or FPGA-based systems). This substantial time margin confirms the real-time applicability of the proposed PIGC-MPC architecture, leaving ample time for sensor fusion, state estimation (EKF), and actuator command transmission.

5 Conclusion

This paper proposed a robust Partially Integrated Guidance and Control (PIGC) framework based on Model Predictive Control for fixed-wing UAVs tracking complex 3D trajectories with constraints. The framework overcomes the critical limitations of separated guidance and control (SGC) such as oscillations due to time delays, poor constraint handling, and poor transient response. The main contributions are as follows: An oriented-cylinder and cubic-spline-based algorithm is used to generate C^2-continuous and dynamically feasible trajectories with controlled stochasticity, which improves the survivability of UAVs against direct-fire threats. A modified frame avoids singularity and sign-flipping problems at inflection points. A $PI-\xi$ closest-point projection allows efficient real-time computation of path-following error. A multi-rate hierarchical MPC architecture consisting of an outer FS-MPC guidance layer and an inner 6DOF-MPC control layer, where FAS feedback ensures recursive feasibility and practical stability. The proposed framework was validated by way of comparative simulations against a tuned SGC-PID controller. SGC-PID showed a large initial transient error (up to 25 m) and severe heading overshoot (nearly 98°) in 3D stochastic path following with wind disturbances. By contrast, PIGC-MPC obtained a 99.3% reduction in position RMSE (from 30.17 m to 0.21 m) and heading RMSE of 1.45°. Soft constraints prevented actuator saturation and guarantyd stable flight during aggressive turns beyond bank-angle limits. Moreover, the optimised control sequence resulted in 44.8% reduction in the cumulative actuator effort J_δ.

Future work includes Hardware-in-the-Loop simulations and in-flight testing, extension to dynamic obstacle avoidance and terrain following, and integration of adaptive learning-based MPC for severe wind gusts, actuator faults and structural damage.

The PIGC-MPC architecture as a whole provides an effective, mathematically rigorous and practically feasible solution to autonomous fixed-wing UAV navigation in complex safety-critical missions.

Author Contributions

- Funding: The authors did not receive support from any organization for the submitted work.
- Conflict of interest/Competing interests: The authors declare that they have no conflict of interest.
- Ethics approval and consent to participate declarations: not applicable
- Consent for publication: not applicable
- Data availability: not applicable
- Materials availability: not applicable
- Code availability : not applicable
- Author contribution:
  Nguyen Bich Van: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing—original draft.
  Nguyen Quang Vinh: Conceptualization, Supervision, Writing—review & editing, Project administration.
  Nguyen Viet Hoang: Methodology, Software, Writing, Validation, Visualization, Data Curation—Review & editing.

  All authors read and approved the final manuscript. this article.
- Corresponding Author:
  Nguyen Quang Vinh; Email: vinhquang2808@yahoo.com; Address: Academy of Military Science and Technology, 17 Hoang Sam, Cau Giay, ha Noi, Viet Nam; Whatsapp: +840986862882

All authors declare that there are no competing interests.

References

[1] R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton University Press, 2012.

[2] M. R. Driels. *Weaponeering: Conventional Weapon System Effectiveness*. 2nd ed. Reston, VA, USA: American Institute of Aeronautics and Astronautics (AIAA), 2013.

[3] Piotr A. Felisiak et al. "Autopilot for unmanned aerial vehicle using model predictive control with embedded integrators". In: *Aerospace Systems* (2026). Published online.

[4] G. García and S. Keshmiri. "Nonlinear Model Predictive Controller for Navigation, Guidance and Control of a Fixed-Wing UAV". In: *Proc. AIAA Guid., Navig., Control Conf.* Portland, OR, USA, 2011, pp. 1–15.

[5] F. Gavilan, R. Vazquez, and S. Esteban. "Trajectory Tracking for Fixed-Wing UAV Using Model Predictive Control and Adaptive Backstepping". In: *IFAC-PapersOnLine* 48.9 (2015), pp. 132–139.

[6] E. C. Kerrigan and J. M. Maciejowski. "Invariant Sets for Constrained Nonlinear Discrete-Time Systems with Application to Feasibility in Model Predictive

Control". In: *Proc. 39th IEEE Conf. Decision Control (CDC)*. Sydney, NSW, Australia, 2000, pp. 4951–4956.
[7] S. Kobayashi and K. Nomizu. *Foundations of Differential Geometry*. Vol. 1. Reprinted by Springer. New York, NY, USA: Wiley, 1995.
[8] S. M. LaValle. *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.
[9] D. Limón et al. "Input to State Stability of Min-Max MPC Controllers for Nonlinear Systems with Bounded Uncertainties". In: *Automatica* 42.5 (May 2006), pp. 797–803.
[10] Yuxie Luo et al. "Survey on Integrated Guidance and Control of Unmanned Aerial Vehicle". In: *Aerospace Systems* (2025). Survey article.
[11] D. Q. Mayne et al. "Constrained Model Predictive Control: Stability and Optimality". In: *Automatica* 36.6 (June 2000), pp. 789–814.
[12] R. L. McCoy. *Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles*. Atglen, PA, USA: Schiffer Publishing, 1999.
[13] Felix Mitze and Matthias Gerdts. "Relaxed DAEs in path planning and path following". In: *Aerospace Systems* (2026). Open Access. DOI: 10.1007/s42401-026-00508-8.
[14] M. Morari and J. H. Lee. "Model Predictive Control: Past, Present and Future". In: *Computers & Chemical Engineering* 23.4–5 (1999), pp. 667–682.
[15] P. Pharpatara. "Trajectory Planning for Aerial Vehicles with Constraints". PhD thesis. Toulouse, France: Université de Toulouse, 2015.
[16] C. Samson. "Control of Chained Systems Application to Path Following and Time-Varying Point-Stabilization of Mobile Robots". In: *IEEE Trans. Autom. Control* 40.1 (Jan. 1995), pp. 64–77.
[17] F. Santoso, M. A. Garratt, and S. G. Anavatti. "State-of-the-Art Integrated Guidance and Control Systems in Unmanned Vehicles: A Review". In: *IEEE Systems Journal* 15.3 (Sept. 2021), pp. 3622–3633.
[18] P. B. Sujit, S. Saripalli, and J. B. Sousa. "Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles". In: *IEEE Control Systems Magazine* 34.1 (Feb. 2014), pp. 42–59.
[19] Xiaosu Tan et al. "Flatness-based path planning for fixed-wing UAVs in tight formation under synergistic aerodynamic constraints". In: *Aerospace Systems* (2025). Published online.
[20] K. P. Valavanis and G. J. Vachtsevanos, eds. *Handbook of Unmanned Aerial Vehicles*. Dordrecht, The Netherlands: Springer, 2015.
[21] D. E. Williams, B. Friedland, and J. Richman. "Integrated Guidance and Control for Combined Command/Homing Guidance". In: *Proc. Amer. Control Conf. (ACC)*. Atlanta, GA, USA, 1988, pp. 1285–1290.