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.

Embedded True Proportional Navigation Guidance for a High-Dynamic Flight Vehicle: Real-Time MIL--HIL Validation

Ahmed M. Khalifa

kalifaahmed51@gmail.com

Military Technical College (MTC)

Ehab Safwat

Military Technical College (MTC)

Mostafa S. Mohamed

Military Technical College (MTC)

Mohammed Abozaid Hassan

Military Technical College (MTC)

Research Article

Keywords: Hardware-in-the-loop, Model-in-the-loop, Terminal guidance, True proportional navigation, Embedded guidance and control, ARM Cortex-M4, Six-degree-of-freedom dynamics, Von K, 'arm, 'an turbulence

Posted Date: October 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11304521/v1

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

Additional Declarations: No competing interests reported.

Embedded True Proportional Navigation Guidance for a High-Dynamic Flight Vehicle: Real-Time MIL–HIL Validation

Ahmed M. Khalifa [1], Ehab Safwat [2], Mostafa S. Mohamed [3], Mohammed Abozaid Hassan [4]

[1,2,3,4] Military Technical College (MTC), Cairo, Egypt.

*Corresponding author(s). E-mail(s): kalifaahmed51@gmail.com;
Contributing authors: e.khattab@mtc.edu.eg; mostafasaad43@gmail.com;
mohammed.abozied@mtc.edu.eg;

Abstract

Real-time embedded implementation can produce substantial differences between the terminal performance predicted by numerical simulation and that realized by a physical guidance-and-control processor. This study characterizes that gap through a controlled Model-in-the-Loop (MIL) and Hardware-in-the-Loop (HIL) evaluation of terminal guidance for a high-dynamic flying vehicle. The two environments use the same nonlinear six-degree-of-freedom plant, aerodynamic database, actuator model and constraints, fixed-step plant solver, controller update rate, and paired seeded von Kármán turbulence realizations. True Proportional Navigation Guidance (TPNG) and a gain-scheduled modified PI autopilot with direct angular-rate damping are executed on an STM32F429 ARM Cortex-M4 microcontroller at 50 Hz, while the nonlinear vehicle and disturbance models are propagated on a real-time target computer. For the four investigated disturbance realizations, the MIL configuration yields planar terminal errors of 0.027–0.460 m, whereas the HIL experiments yield 137.6–1865.6 m. The embedded guidance-and-control computation requires an average of 0.356 ms within the 20 ms control interval, corresponding to 1.78% execution-time occupancy. Thus, computational overload is excluded as the direct source of the observed degradation. The results demonstrate that end-to-end implementation effects, including communication, task scheduling, sample alignment, and data representation, must be treated as part of the closed-loop guidance design. The current experiments quantify their aggregate effect; causal attribution to individual mechanisms requires controlled delay and interface-ablation tests.

Keywords: Hardware-in-the-loop; Model-in-the-loop; Terminal guidance; True proportional navigation; Embedded guidance and control; ARM Cortex-M4; Six-degree-of-freedom dynamics; Von Kármán turbulence

1 Introduction

Terminal guidance of high-dynamic flying vehicles requires accurate interception in the presence of strongly coupled rigid-body dynamics, rapidly varying aerodynamic conditions, atmospheric disturbances, actuator constraints, and stringent real-time execution requirements [1–3]. Guidance laws are often designed and first assessed in numerically controlled environments; after deployment, however, communication delay, finite-rate sampling, data quantization, task scheduling, and asynchronous hardware interfaces can alter the realized closed-loop response [4–6]. These effects become increasingly important near interception, where the remaining time-to-go is short and corrective authority must be delivered with minimal delay. Atmospheric turbulence further increases the required control activity and can interact with actuator dynamics and autopilot bandwidth [7–9]. Consequently, agreement between a guidance design and its numerical simulation does not by itself establish equivalent performance on embedded hardware.

Hardware-in-the-Loop (HIL) testing provides an intermediate validation stage between numerical simulation and flight testing by executing the guidance and control software on the target processor while a real-time simulator propagates the vehicle and environment models [10–13]. Recent aerospace work also emphasizes target-hardware timing, synchronization, and software verification as integral parts of flight-control validation [6, 14].

Nevertheless, direct paired MIL–HIL comparisons of terminal guidance remain limited when nonlinear six-degree-of-freedom dynamics, actuator constraints, and stochastic atmospheric disturbances are retained simultaneously. If the plant, solver, disturbance realization, or actuator model changes between the two environments, the resulting performance difference cannot be interpreted cleanly as an implementation effect.

The STM32F429 used in the present study is based on an ARM Cortex-M4 core with a single-precision floating-point unit and a maximum clock frequency of 180 MHz [15]. Processor capability alone, however, is not a sufficient measure of closed-loop implementation quality. Communication interfaces, packet handling, task release, sample alignment, and measurement or command delays can introduce phase lag and timing uncertainty even when the control algorithm occupies only a small fraction of the available frame time [6, 14, 16]. These considerations motivate an experimental comparison that separates algorithm execution cost from the broader end-to-end implementation chain.

True Proportional Navigation Guidance (TPNG) is adopted as the baseline terminal law because of its compact mathematical structure, low computational burden, and well-established interception properties [2, 17–19]. The guidance law is coupled to a gain-scheduled modified PI autopilot with direct angular-rate damping and an experimentally identified actuator model with explicit magnitude and rate constraints. A prescribed von Kármán turbulence realization is paired between the MIL and HIL cases so that disturbance variability is not confused with the implementation difference.

Accordingly, the main contributions of this study are:

(1) a controlled MIL–HIL comparison framework in which the nonlinear plant, aerodynamic database, actuator model and limits, numerical settings, controller update rate, and paired turbulence realization are maintained consistently;
(2) real-time implementation of TPNG and the modified PI autopilot with angular-rate damping on an STM32F429 ARM Cortex-M4 microcontroller coupled to an ARK5000 real-time target executing the nonlinear vehicle and disturbance models;
(3) characterization of the embedded timing budget using measured controller execution time together with packet-level serial-transmission timing, while explicitly distinguishing these quantities from unmeasured scheduler and end-to-end interface delays;
(4) a paired comparison of MIL and HIL terminal performance under ideal, light, moderate, and severe prescribed turbulence conditions, providing a direct measure of the aggregate implementation-induced performance gap for the investigated realizations; and
(5) identification of validation requirements needed to attribute that gap to individual mechanisms, including delay injection, timing-jitter characterization, interface quantization tests, and repeated stochastic trials.

The remainder of the paper is organized as follows. Section 2 presents the nonlinear 6-DOF model, reference frames, aerodynamic formulation, and atmospheric disturbance model. Section 3 describes the autopilot, gain scheduling, control allocation, actuator dynamics, and physical limits. Section 4 formulates the TPNG law and its discrete-time implementation. Section 5 presents the HIL architecture and timing model. Section 6 compares the paired MIL and HIL results, and Section 7 summarizes the principal findings, limitations, and recommended extensions.

2 Vehicle Modeling and Dynamics

A consistent nonlinear vehicle model is required for a meaningful MIL–HIL comparison. Any change in the plant, aerodynamic representation, actuator model, solver configuration, or atmospheric disturbance between the two environments would introduce an additional source of discrepancy [4]. The same nonlinear six-degree-of-freedom (6-DOF) rigid-body plant is therefore used in the MIL and HIL configurations. State propagation on the real-time plant uses a fixed-step fourth-order Runge–Kutta scheme with the same plant step and numerical settings in the paired cases [5, 20].

Coordinate frames and state representation.

Three frames are distinguished. The inertial frame is denoted by $\mathcal{F}_I$. The local navigation frame $\mathcal{F}_n$ uses the North–Up–East (NUE) ordering, with x_n directed north/downrange, y_n upward, and z_n east/lateral. The body-fixed frame $\mathcal{F}_b$ uses x_b forward, y_b right, and z_b down. This ordering is retained throughout the vehicle, guidance, and plotting equations. The direction-cosine matrix $\mathbf{C}_b^n$ maps body-frame vectors into the navigation frame,

$$\mathbf{v}_n = \mathbf{C}_b^n \mathbf{v}_b, \qquad \mathbf{C}_n^b = (\mathbf{C}_b^n)^\top. \tag{1}$$

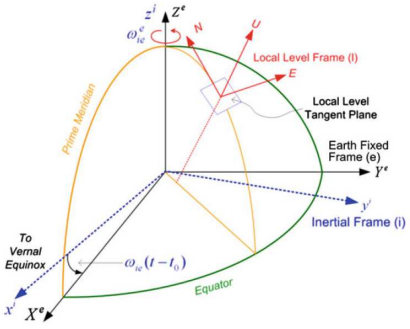
Table 1: Principal parameters used in the MIL–HIL terminal-guidance validation scenario.

Parameter	Symbol	Value	Unit
Vehicle properties at the initial condition			
Mass	m_0	550	kg
Roll inertia	$J_{xx,0}$	8.5	$\mathrm{kg\,m^2}$
Pitch inertia	$J_{yy,0}$	420	$\mathrm{kg\,m^2}$
Yaw inertia	$J_{zz,0}$	420	$\mathrm{kg\,m^2}$
Reference area	S	0.196	$\mathrm{m^2}$
Reference diameter	d	0.5	m
Reference length	L	5.4	m
Initial engagement conditions			
Initial range	R_0	90,000	m
Initial altitude	h_0	12,000	m
Initial speed	V_0	250	m/s
Operational Mach range	M	0.8–3.5	–
Engagement duration	T_f	≈ 210	s
Target	–	Stationary, ground-level	
Actuator model and constraints			
Deflection limit	$\delta_{\max}$	± 25	deg
Rate limit	$\dot{\delta}_{\max}$	350	deg/s
Identified DC gain	$G_{\mathrm{act}}(0)$	0.901	–
Guidance and atmospheric-disturbance parameters			
Navigation constant	$N_n = N_p$	4	–
Controller update rate	f_s	50	Hz
Hit radius	R_{hit}	5.0	m
Nominal σ_w: Ideal	–	0	m/s
Nominal σ_w: Light	–	0.76	m/s
Nominal σ_w: Moderate	–	1.52	m/s
Nominal σ_w: Severe	–	3.05	m/s

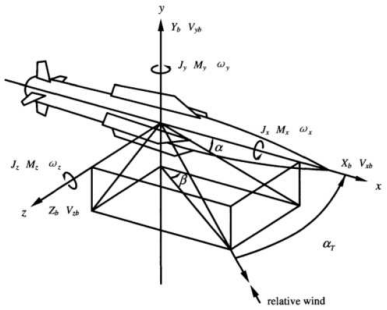
The propagated state is

$$\mathbf{x} = \left[\mathbf{r}_n^\top \ \mathbf{v}_n^\top \ \mathbf{q}^\top \ \boldsymbol{\omega}_b^\top\right]^\top, \tag{2}$$

where $\mathbf{r}_n = [X_N,\, Y_U,\, Z_E]^\top$, $\mathbf{q}$ is a unit quaternion, and $\boldsymbol{\omega}_b = [p,\, q,\, r]^\top$ contains roll, pitch, and yaw rates.



(a) Local navigation frame; the implemented state ordering is NUE.



(b) Body-fixed frame: forward, right, and down.

Fig. 1: Reference frames used in the nonlinear 6-DOF model. The navigation-state ordering is North–Up–East, whereas aerodynamic loads and angular rates are evaluated in the body-fixed frame.

Rigid-body equations of motion.

The position kinematics are

$$\dot{\mathbf{r}}_n = \mathbf{v}_n. \tag{3}$$

The translational dynamics are written as

$$m\dot{\mathbf{v}}_n = \mathbf{F}_n + m\mathbf{a}_{\mathrm{env,n}}, \tag{4}$$

where $\mathbf{F}_n$ contains the aerodynamic and propulsive forces transformed from the body frame and $\mathbf{a}_{\mathrm{env,n}}$ denotes the environmental acceleration returned by the gravity/Earth-rotation model. In the implemented plant this quantity contains the gravitational, Coriolis, and centrifugal contributions expressed in the local navigation frame,

$$\mathbf{a}_{\mathrm{env,n}} = \mathbf{a}_{g,n} + \mathbf{a}_{\mathrm{Cor,n}} + \mathbf{a}_{\mathrm{cf,n}}. \tag{5}$$

The individual terms are evaluated from the geographic state and Earth-rotation model before transformation to the local NUE frame. The non-gravitational force is

$$\mathbf{F}_n = \mathbf{C}_b^n \left(\mathbf{F}_{a,b} + \mathbf{F}_{T,b}\right). \tag{6}$$

Rotational motion about the center of mass follows the Euler rigid-body equation

$$\mathbf{J}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times (\mathbf{J}\boldsymbol{\omega}_b) = \mathbf{M}_b, \tag{7}$$

and attitude is propagated using

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_b)\mathbf{q}. \tag{8}$$

The quaternion is renormalized after each integration step to suppress numerical drift. If mass, inertia, or center-of-gravity properties are varied by the propulsion model, the instantaneous values are used in Eqs. (4) and (7); otherwise the initial values in Table 1 are retained.

Air-relative motion and aerodynamic model.

Aerodynamic loads are evaluated from the velocity relative to the atmosphere,

$$\mathbf{V}_{a,n} = \mathbf{v}_n - \mathbf{W}_n, \qquad \mathbf{V}_{a,b} = \mathbf{C}_n^b \mathbf{V}_{a,n} = \begin{bmatrix} u_a & v_a & w_a \end{bmatrix}^\top, \tag{9}$$

with

$$V_a = \sqrt{u_a^2 + v_a^2 + w_a^2}. \tag{10}$$

For the forward–right–down body convention,

$$\alpha = \mathrm{atan2}(w_a, u_a), \qquad \beta = \sin^{-1}\left(\frac{v_a}{V_a}\right). \tag{11}$$

The dimensional aerodynamic forces are

$$\mathbf{F}_{a,b} = \bar{q}S \begin{bmatrix} C_X & C_Y & C_Z \end{bmatrix}^\top, \qquad \bar{q} = \frac{1}{2}\rho V_a^2. \tag{12}$$

The aerodynamic moments are written in the database-consistent form

$$\mathbf{M}_{a,b} = \bar{q}S\,\mathbf{L}_{\mathrm{ref}} \begin{bmatrix} C_l & C_m & C_n \end{bmatrix}^\top, \qquad \mathbf{L}_{\mathrm{ref}} = \mathrm{diag}(\ell_l, \ell_m, \ell_n), \tag{13}$$

where ℓ_l, ℓ_m, and ℓ_n are the reference lengths used when the source aerodynamic coefficients are nondimensionalized. This notation avoids silently mixing aircraft span/chord conventions with an axisymmetric guided-vehicle database; the same reference lengths are used in all MIL and HIL runs.

The aerodynamic coefficient vector is obtained by interpolation of the same lookup database in both environments,

$$\mathbf{C}_a = \begin{bmatrix} C_X & C_Y & C_Z & C_l & C_m & C_n \end{bmatrix}^\top = \mathcal{F}(M, \alpha, \beta, \delta_e, \delta_r, \delta_a, p, q, r, X_{cg}), \tag{14}$$

with the implemented interpolation method and extrapolation limits held fixed between paired tests.

Atmospheric disturbance model.

The atmospheric velocity is represented as the superposition of a deterministic steady-wind component and a stochastic turbulence component,

$$\mathbf{W}_n(t) = \mathbf{W}_{\mathrm{steady}}(h) + \mathbf{W}_{\mathrm{gust}}(t), \tag{15}$$

where $\mathbf{W}_n$ is expressed in the local North–Up–East (NUE) navigation frame. The steady-wind component is prescribed as an altitude-dependent power-law profile,

$$V_{\text{steady}}(h) = V_{\text{ref}} \left(\frac{h}{h_{\text{ref}}} \right)^{\alpha_s}, \tag{16}$$

where V_{ref} is the reference wind speed at altitude h_{ref} and α_s is the atmospheric shear exponent. For a prescribed wind azimuth χ_w measured from North toward East and elevation angle γ_w, the steady wind vector is

$$\mathbf{W}^n_{\text{steady}}(h) = V_{\text{steady}}(h) \begin{bmatrix} \cos\gamma_w \cos\chi_w \\ \sin\gamma_w \\ \cos\gamma_w \sin\chi_w \end{bmatrix}. \tag{17}$$

The stochastic component is generated using the von Kármán turbulence spectrum, which preserves the characteristic inertial-range behavior of atmospheric turbulence and is widely used in flight-dynamics simulation [7–9, 21]. Let u_g, v_g, and w_g denote the longitudinal, lateral, and vertical turbulence components in a turbulence-aligned reference frame. Their stationary random processes satisfy

$$\mathbb{E}[u_g(t)] = \mathbb{E}[v_g(t)] = \mathbb{E}[w_g(t)] = 0, \tag{18}$$

with root-mean-square intensities

$$\sigma_u = \sqrt{\mathbb{E}\left[u_g^2\right]}, \qquad \sigma_v = \sqrt{\mathbb{E}\left[v_g^2\right]}, \qquad \sigma_w = \sqrt{\mathbb{E}\left[w_g^2\right]}. \tag{19}$$

For spatial angular frequency Ω, the longitudinal von Kármán spectrum is expressed as

$$\Phi_u(\Omega) = \frac{2\sigma_u^2 L_u}{\pi} \frac{1}{\left[1 + (1.339 L_u \Omega)^2\right]^{5/6}}, \tag{20}$$

where L_u is the longitudinal turbulence scale length. The corresponding transverse spectra are

$$\Phi_i(\Omega) = \frac{\sigma_i^2 L_i}{\pi} \frac{1 + \frac{8}{3}(1.339 L_i \Omega)^2}{\left[1 + (1.339 L_i \Omega)^2\right]^{11/6}}, \qquad i \in \{v, w\}, \tag{21}$$

where L_v and L_w are the lateral and vertical turbulence scale lengths, respectively.

Under the frozen-turbulence assumption, the spatial frequency is mapped to temporal frequency through the airspeed V_a according to

$$\Omega = \frac{\omega}{V_a}, \tag{22}$$

so that the corresponding temporal spectrum satisfies

$$S_i(\omega) = \frac{1}{V_a} \Phi_i \left(\frac{\omega}{V_a} \right), \qquad i \in \{u, v, w\}. \tag{23}$$

Because Eqs. (20) and (21) contain fractional spectral powers, direct implementation is not convenient for real-time simulation. Each spectrum is therefore represented by a finite-dimensional rational shaping-filter approximation,

$$H_i(s) \approx H_{i,\text{VK}}(s), \qquad |H_i(j\omega)|^2 \approx S_i(\omega), \qquad i \in \{u, v, w\}, \tag{24}$$

following the real-time implementation procedures described in [8, 21]. Unit-variance Gaussian white-noise sequences η_i are supplied to the shaping filters,

$$\mathbb{E}[\eta_i(t)] = 0, \qquad \mathbb{E}\left[\eta_i(t)\eta_j(t+\tau)\right] = \delta_{ij}\delta(\tau), \tag{25}$$

where δ_{ij} is the Kronecker delta and $\delta(\tau)$ is the Dirac delta. The resulting turbulence-filter dynamics can be expressed in continuous state-space form as

$$\dot{\mathbf{x}}_{g,i}(t) = \mathbf{A}_{g,i}\mathbf{x}_{g,i}(t) + \mathbf{B}_{g,i}\eta_i(t), \tag{26}$$

$$g_i(t) = \mathbf{C}_{g,i}\mathbf{x}_{g,i}(t) + D_{g,i}\eta_i(t), \qquad i \in \{u, v, w\}, \tag{27}$$

where $g_i \in \{u_g, v_g, w_g\}$.

For real-time implementation with simulation step T_s, the turbulence filters are discretized as

$$\mathbf{x}_{g,i}[k+1] = \mathbf{A}_{d,i}\mathbf{x}_{g,i}[k] + \mathbf{B}_{d,i}\eta_i[k], \tag{28}$$

$$g_i[k] = \mathbf{C}_{d,i}\mathbf{x}_{g,i}[k] + D_{d,i}\eta_i[k], \tag{29}$$

where, for zero-order-hold discretization,

$$\mathbf{A}_{d,i} = e^{\mathbf{A}_{g,i}T_s}, \qquad \mathbf{B}_{d,i} = \int_0^{T_s} e^{\mathbf{A}_{g,i}\tau}\mathbf{B}_{g,i}\, d\tau. \tag{30}$$

The resulting turbulence vector is

$$\mathbf{W}^g_{\text{gust}} = \begin{bmatrix} u_g & v_g & w_g \end{bmatrix}^\top, \tag{31}$$

and is transformed to the navigation frame through

$$\mathbf{W}^n_{\text{gust}} = \mathbf{C}^n_g \mathbf{W}^g_{\text{gust}}, \tag{32}$$

where $\mathbf{C}^n_g$ denotes the transformation from the turbulence reference frame to the local NUE navigation frame. The complete air-relative velocity used by the aerodynamic model is then

$$\mathbf{V}_{a,n} = \mathbf{v}_n - \left(\mathbf{W}^n_{\text{steady}} + \mathbf{W}^n_{\text{gust}}\right). \tag{33}$$

For every paired MIL–HIL experiment, identical turbulence-model parameters and pseudorandom-number-generator seeds are prescribed. If $\mathcal{R}(\cdot)$ denotes the pseudorandom generator, the discrete excitation sequence may be represented as

$$\eta_i[k] = \mathcal{R}\left(k; \text{seed}_i\right), \qquad i \in \{u, v, w\}. \tag{34}$$

Thus, the same normalized stochastic excitation sequence is applied to the corresponding MIL and HIL cases. Independent streams may be used for the longitudinal, lateral, and vertical channels while retaining the same channel seeds between each paired MIL–HIL run.

The nominal vertical gust RMS levels used in the reported experiments are

$$\sigma_{w,\text{nom}} = \begin{cases} 0, & \text{Ideal}, \\ 0.76, & \text{Light}, \\ 1.52, & \text{Moderate}, \\ 3.05, & \text{Severe}, \end{cases} \quad \text{m/s}, \tag{35}$$

while the corresponding severity bands are

$$\sigma_w = \begin{cases} 0 \text{ m/s}, & \text{Ideal}, \\ [0.60,\ 0.91] \text{ m/s}, & \text{Light}, \\ [1.22,\ 1.82] \text{ m/s}, & \text{Moderate}, \\ [2.44,\ 3.66] \text{ m/s}, & \text{Severe}. \end{cases} \tag{36}$$

Here, σ_w denotes the RMS intensity of the vertical turbulence component rather than the instantaneous body-axis velocity component w_a. The turbulence intensity can therefore be interpreted through

$$\sigma_w^2 = \mathbb{E}\left[w_g^2(t)\right], \tag{37}$$

for the zero-mean stochastic component.

Because only one fixed realization is reported for each nonzero turbulence severity, the light, moderate, and severe cases should be interpreted as paired deterministic MIL–HIL experiments rather than as statistical turbulence ensembles. Consequently, differences among the three reported nonzero-turbulence cases cannot by themselves establish a monotonic statistical relationship between turbulence intensity and terminal miss distance.

3 Control System Design and Autopilot

The autopilot converts the signed transverse-acceleration commands generated by the terminal guidance law into virtual pitch and yaw control commands and stabilizes roll. A gain-scheduled modified proportional–integral (PI) architecture with direct angular-rate damping is used rather than a conventional PID law. Integral action reduces low-frequency acceleration-tracking error, while measured body-rate feedback adds damping without differentiating the acceleration error [22–26]. The same controller equations, gain tables, allocation logic, actuator model, and limits are used in MIL and HIL.

Pitch and yaw acceleration channels.

Let $j \in \{p, y\}$ denote the pitch and yaw channels. The signed acceleration error is

$$e_{a,j}(t) = a_{c,j}(t) - a_j(t), \tag{38}$$

with the channel mapping

$$\begin{aligned} a_{c,p} &= A_{y,c}, \quad a_p = A_y, \quad \omega_p = q, \quad \delta_p = \delta_e, \\ a_{c,y} &= A_{z,c}, \quad a_y = A_z, \quad \omega_y = r, \quad \delta_y = \delta_r. \end{aligned} \tag{39}$$

The virtual control command is

$$\delta_{u,j}(t) = K_{p,j} e_{a,j}(t) + K_{i,j} \xi_j(t) - K_{d,j} \omega_j(t), \qquad \dot{\xi}_j(t) = e_{a,j}(t), \tag{40}$$

where $K_{p,j}$ and $K_{i,j}$ are the proportional and integral acceleration gains and $K_{d,j}$ is the direct angular-rate damping gain. The associated PI compensator is

$$C_{\mathrm{PI,j}}(s) = K_{p,j} + \frac{K_{i,j}}{s} = K_{p,j}\left(1 + \frac{1}{\tau_{a,j} s}\right), \qquad K_{i,j} = \frac{K_{p,j}}{\tau_{a,j}}. \tag{41}$$

For local interpretation, define the linearized control-to-acceleration and control-to-rate dynamics at an operating point as

$$G_{a,j}(s) = \frac{a_j(s)}{\delta_j(s)}, \qquad G_{\omega,j}(s) = \frac{\omega_j(s)}{\delta_j(s)}. \tag{42}$$

With the actuator $G_{\mathrm{act}}(s)$ included, $P_{a,j} = G_{\mathrm{act}} G_{a,j}$ and $P_{\omega,j} = G_{\mathrm{act}} G_{\omega,j}$. Neglecting saturation for the local linear analysis gives

$$\frac{a_j(s)}{a_{c,j}(s)} = \frac{P_{a,j}(s) C_{\mathrm{PI,j}}(s)}{1 + P_{a,j}(s) C_{\mathrm{PI,j}}(s) + K_{d,j} P_{\omega,j}(s)}. \tag{43}$$

Equation (43) is used only to interpret the feedback structure; all reported results are generated with the full nonlinear 6-DOF model.

Roll channel.

The roll channel regulates bank angle with roll-rate damping,

$$e_\phi = \phi_c - \phi, \qquad \delta_{u,a} = K_{p,\phi} e_\phi + K_{i,\phi} \xi_\phi - K_{d,p} p, \qquad \dot{\xi}_\phi = e_\phi. \tag{44}$$

Gain scheduling.

Aerodynamic control effectiveness varies strongly with flight condition, so fixed gains are not used over the complete engagement. The implemented schedule is indexed by flight phase and a dynamic-pressure scheduling coordinate Q_v. The phase logic is

$$\mathcal{P} = \begin{cases} \text{Boost}, & T > 0 \text{ and } V_U > 5 \text{ m/s}, \\ \text{Coast}, & T \le 0 \text{ and } V_U \ge -5 \text{ m/s}, \\ \text{Reentry}, & T \le 0 \text{ and } V_U < -5 \text{ m/s}, \end{cases} \tag{45}$$

where T is thrust and V_U is the upward navigation-frame velocity. Within each phase, the controller gains are obtained by one-dimensional interpolation of the stored phase-specific gain tables,

$$K_{\ell,j} = \mathcal{K}_{\ell,j}^{(\mathcal{P})}(Q_v), \qquad \ell \in \{p, i, d\}. \tag{46}$$

For adjacent breakpoints $Q_{v,k} \le Q_v \le Q_{v,k+1}$, a generic scheduled gain is

$$K(Q_v) = K_k + \frac{Q_v - Q_{v,k}}{Q_{v,k+1} - Q_{v,k}} \left(K_{k+1} - K_k\right). \tag{47}$$

The same phase state and lookup tables are used in paired MIL and HIL experiments. This formulation describes the implemented scheduler without implying a Mach–dynamic- pressure grid that is not present in the flight code.

X-fin control allocation.

The virtual roll, pitch, and yaw commands are mapped to four physical fins in an X-configuration. With $k = 1/\sqrt{2}$, the allocation is

$$\begin{aligned} \delta_1 &= \delta_a - k\delta_e - k\delta_r, \\ \delta_2 &= \delta_a + k\delta_e - k\delta_r, \\ \delta_3 &= \delta_a + k\delta_e + k\delta_r, \\ \delta_4 &= \delta_a - k\delta_e + k\delta_r. \end{aligned} \tag{48}$$

This fixed linear allocation distributes the three virtual control channels among the four effectors without an online optimization step; related aerospace control-allocation considerations are discussed in [27].

Actuator dynamics and physical limits.

Each fin is represented by the experimentally identified two-pole/one-zero model

$$G_{\text{act}}(s) = \frac{\delta(s)}{\delta_{\text{cmd}}(s)} = \frac{10.5104s + 9.6778}{s^2 + 13.7988s + 10.7378}. \tag{49}$$

The denominator corresponds to

$$\omega_n = \sqrt{10.7378} \approx 3.277 \text{ rad/s}, \qquad \zeta = \frac{13.7988}{2\sqrt{10.7378}} \approx 2.105, \tag{50}$$

and the DC gain is

$$G_{\text{act}}(0) = \frac{9.6778}{10.7378} \approx 0.901. \tag{51}$$

The corresponding differential equation is

$$\ddot{\delta} + 13.7988\dot{\delta} + 10.7378\delta = 10.5104\dot{\delta}_{\text{cmd}} + 9.6778\delta_{\text{cmd}}. \tag{52}$$

Each physical fin is subject to

$$|\delta_i| \le 25^\circ, \qquad |\dot{\delta}_i| \le 350^\circ/\text{s}, \qquad i = 1, \dots, 4. \tag{53}$$

The nonlinear magnitude and rate limits are enforced identically in MIL and HIL. The reported terminal results therefore include the same actuator authority in both environments.

4 Terminal Guidance Law Implementation

True Proportional Navigation Guidance (TPNG) is employed as the baseline terminal-guidance law because of its simple mathematical structure, low computational burden, and well-established interception characteristics [2, 17]. Under the standard proportional navigation assumptions, the guidance objective is to drive the line-of-sight (LOS) angular rate toward zero while maintaining positive closing velocity, thereby establishing a collision geometry between the vehicle and the target.

TPNG is particularly suitable for the present MIL–HIL investigation because its computational requirements are modest, allowing implementation-related effects to be studied without introducing unnecessary algorithmic complexity. All LOS quantities are evaluated analytically from relative position and velocity rather than through numerical differentiation. The same guidance equations, numerical sequence, and update rate are retained in the MIL and HIL implementations.

Three-dimensional engagement geometry.

Consider a pursuer vehicle M and a target T. All engagement quantities are expressed in the North–Up–East navigation frame $\mathcal{F}_n$, so that X, Y, and Z denote north/downrange, up/height, and east/lateral components, respectively. The relative position and relative velocity are defined as

$$\mathbf{r} = \mathbf{r}_T - \mathbf{r}_M = \begin{bmatrix} X & Y & Z \end{bmatrix}^\top, \tag{54}$$

and

$$\mathbf{v}_{\text{rel}} = \mathbf{v}_T - \mathbf{v}_M = \begin{bmatrix} V_x & V_y & V_z \end{bmatrix}^\top. \tag{55}$$

For the stationary target considered in the present engagement,

$$\mathbf{v}_T = \mathbf{0}, \qquad \mathbf{v}_{\text{rel}} = -\mathbf{v}_M. \tag{56}$$

The instantaneous vehicle-to-target range is

$$R = \|\mathbf{r}\| = \sqrt{X^2 + Y^2 + Z^2}, \tag{57}$$

and the corresponding LOS unit vector is

$$\hat{\mathbf{e}}_R = \frac{\mathbf{r}}{R}. \tag{58}$$

The closing velocity is obtained from the projection of relative velocity onto the LOS direction,

$$V_c = -\hat{\mathbf{e}}_R^{\top}\mathbf{v}_{\text{rel}} = -\frac{\mathbf{r}^{\top}\mathbf{v}_{\text{rel}}}{R}. \tag{59}$$

Thus, $V_c > 0$ denotes a closing engagement, whereas $V_c \leq 0$ indicates constant or increasing separation.

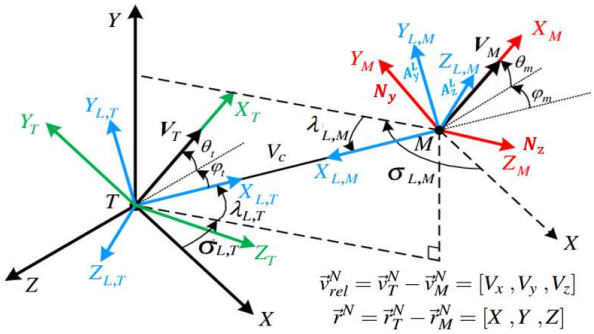


Fig. 2: Three-dimensional terminal-engagement geometry defining the pursuer M, stationary target T, relative position $\mathbf{r} = \mathbf{r}_T - \mathbf{r}_M$, relative velocity $\mathbf{v}_{\text{rel}} = \mathbf{v}_T - \mathbf{v}_M$, LOS azimuth σ, and LOS elevation λ in the navigation frame.

Line-of-sight angles and angular rates.

The three-dimensional LOS geometry is parameterized by azimuth σ and elevation λ. To preserve the correct angular quadrant, the LOS angles are evaluated using the two-argument arctangent function,

$$\sigma = \text{atan2}(Z, X), \qquad \lambda = \text{atan2}\left(Y, \sqrt{X^2 + Z^2}\right). \tag{60}$$

Define the horizontal projection of the relative-position vector as

$$R_h = \sqrt{X^2 + Z^2}. \tag{61}$$

Analytical differentiation of Eq. (60) gives the LOS azimuth rate

$$\dot{\sigma} = \frac{XV_z - ZV_x}{X^2 + Z^2}, \tag{62}$$

and the LOS elevation rate

$$\dot{\lambda} = \frac{(X^2 + Z^2)V_y - Y(XV_x + ZV_z)}{R_h R^2}. \tag{63}$$

An equivalent compact three-dimensional expression for the LOS angular velocity is

$$\boldsymbol{\omega}_{\text{LOS}} = \frac{\mathbf{r} \times \mathbf{v}_{\text{rel}}}{R^2}. \tag{64}$$

Equation (64) is useful for interpreting the guidance geometry: a vanishing LOS angular velocity corresponds to a constant-bearing condition. Provided that the engagement remains closing, this condition approaches the collision geometry required for interception.

Time-to-go estimate and termination logic.

For a stationary target and positive closing velocity, a first-order time-to-go estimate is

$$t_{\mathrm{go}} = \frac{R}{V_c}. \tag{65}$$

This estimate is used only while $V_c > 0$. In the numerical implementation, small positive lower bounds are applied to denominators involving R, R_h, and V_c to avoid singular numerical operations near the terminal condition.

The engagement is terminated when the prescribed hit condition is satisfied according to

$$R \le R_{\mathrm{hit}}, \tag{66}$$

The guidance calculation is also protected by small positive denominator bounds near intercept. The terminal performance metric used later is defined separately in Section 6; it is not inferred from this numerical safeguard.

TPNG acceleration-command formulation.

To formulate TPNG in three dimensions, define the LOS elevation and azimuth basis vectors as

$$\hat{\mathbf{e}}_\lambda = \begin{bmatrix} -\sin\lambda\cos\sigma \\ \cos\lambda \\ -\sin\lambda\sin\sigma \end{bmatrix}, \tag{67}$$

and

$$\hat{\mathbf{e}}_\sigma = \begin{bmatrix} -\sin\sigma \\ 0 \\ \cos\sigma \end{bmatrix}. \tag{68}$$

The commanded transverse acceleration is then expressed as

$$\mathbf{a}_c = N_n V_c \dot{\lambda} \hat{\mathbf{e}}_\lambda + N_p V_c \dot{\sigma} \cos\lambda \hat{\mathbf{e}}_\sigma, \tag{69}$$

where N_n and N_p denote the proportional-navigation constants in the elevation and azimuth guidance channels, respectively. In the present implementation,

$$N_n = N_p = 4. \tag{70}$$

Expansion of Eq. (69) gives the navigation-frame components

$$\begin{aligned} A_{x,c} = & -N_n V_c \dot{\lambda} \sin\lambda\cos\sigma \\ & - N_p V_c \dot{\sigma} \cos\lambda \sin\sigma, \end{aligned} \tag{71}$$

$$A_{y,c} = N_n V_c \dot{\lambda} \cos\lambda, \tag{72}$$

$$A_{z,c} = N_p V_c \dot{\sigma} \cos\lambda\cos\sigma - N_n V_c \dot{\lambda} \sin\lambda \sin\sigma. \tag{73}$$

The guidance system controls the transverse acceleration components $A_{y,c}$ and $A_{z,c}$, while the longitudinal vehicle motion is determined primarily by propulsion and aerodynamic forces. Accordingly, the autopilot receives the signed commands

$$A_{y,c} \longrightarrow \text{pitch acceleration channel}, \qquad A_{z,c} \longrightarrow \text{yaw acceleration channel}. \tag{74}$$

This preserves both the magnitude and direction of the guidance correction. For diagnostic purposes only, the transverse command magnitude may be defined as

$$A_{\perp,c} = \sqrt{A_{y,c}^2 + A_{z,c}^2}, \tag{75}$$

but this scalar quantity is not used as the control input to the pitch and yaw autopilot channels.

Discrete-time embedded implementation.

The guidance algorithm is executed at

$$f_s = 50 \text{ Hz}, \qquad T_s = \frac{1}{f_s} = 0.02 \text{ s}, \tag{76}$$

with update instants

$$t_k = kT_s, \qquad k = 0, 1, 2, \ldots. \tag{77}$$

At each update, the embedded algorithm performs the sequence

$$\{\mathbf{r}[k], \mathbf{v}_{\text{rel}}[k]\} \rightarrow \left\{R[k], V_c[k], \sigma[k], \lambda[k], \dot{\sigma}[k], \dot{\lambda}[k]\right\} \rightarrow \{A_{y,c}[k], A_{z,c}[k]\}\,. \tag{78}$$

The resulting acceleration commands are held constant between consecutive guidance updates,

$$\mathbf{a}_c(t) = \mathbf{a}_c[k], \qquad t_k \le t < t_{k+1}, \tag{79}$$

representing the zero-order-hold behavior of the digital implementation.

The same guidance update rate and mathematical sequence are used in the MIL and HIL configurations. Consequently, the nominal 50 Hz discretization is treated as part of the common controller implementation rather than, by itself, as a HIL-specific source of performance degradation. Additional end-to-end delays introduced by embedded computation, data transmission, packet handling, and synchronization are characterized separately in Section 5. This separation is important for distinguishing the effects of nominal digital implementation from those introduced specifically by the physical HIL communication chain.

5 Hardware-in-the-Loop Testing Framework

The HIL framework is constructed to preserve environmental equivalence between the software reference and the embedded experiment. The nonlinear plant, aerodynamic database, actuator dynamics and limits, plant integration settings, controller update rate, and paired atmospheric-disturbance realization are maintained consistently. The remaining difference is therefore concentrated in the realization of the controller and its physical communication interface [4, 10, 13].

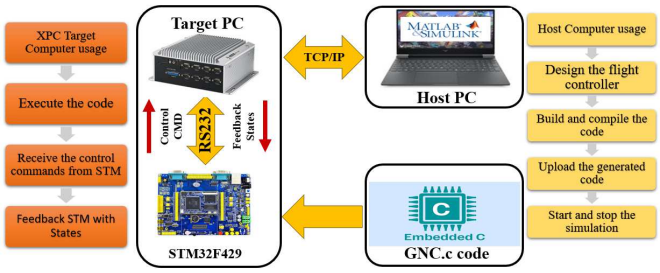


Fig. 3: HIL architecture showing the real-time plant, embedded guidance-and-control processor, and supervisory host computer.

System architecture.

The ARK5000 real-time target executes the nonlinear 6-DOF plant, aerodynamic model, actuator model, and atmospheric disturbance in MathWorks Simulink Real-Time. The STM32F429 executes the TPNG and autopilot code. The host computer is supervisory only. This arrangement follows the controller-HIL separation commonly used for aerospace real-time validation [11, 12, 14]. The closed-loop sequence is

$$\text{plant} \rightarrow \text{input packet} \rightarrow \text{embedded guidance/control} \rightarrow \text{command packet} \rightarrow \text{plant}. \tag{80}$$

Serial interface and packet-level timing.

The physical link uses RS232 at $B = 115200$ bit/s with 8N1 framing and no hardware flow control. Each byte therefore occupies ten serial bits. For a packet of N_b bytes, the ideal wire serialization time is

$$T_{\text{ser}}(N_b) = \frac{10N_b}{B}. \tag{81}$$

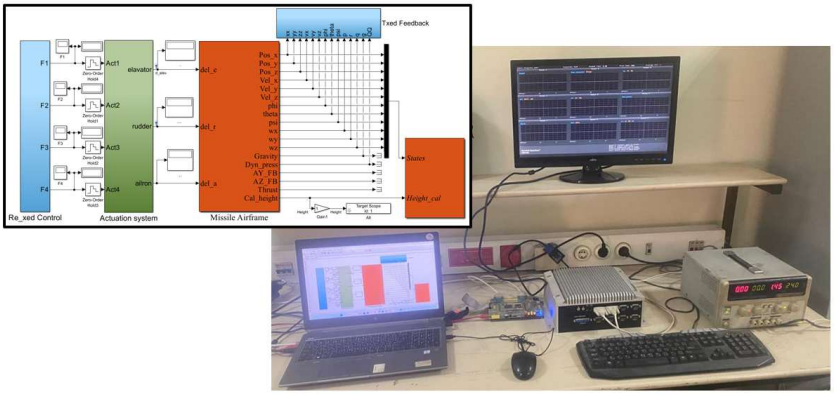


Fig. 4: Experimental HIL validation platform. The STM32F429 executes TPNG and the autopilot, the ARK5000 real-time target propagates the nonlinear 6-DOF vehicle and disturbance models, and the host computer performs deployment, monitoring, and data acquisition without participating directly in the feedback computation.

The controller-input packet has a measured length of 12 bytes and the command-return packet 6 bytes. Their theoretical serialization times are

$$T_{\text{ser,in}} = \frac{10(12)}{115200} = 1.042 \text{ ms}, \qquad T_{\text{ser,out}} = \frac{10(6)}{115200} = 0.521 \text{ ms}. \tag{82}$$

Thus, the minimum two-way wire time is

$$T_{\text{ser,2way}} = 1.563 \text{ ms}, \tag{83}$$

before packet parsing, task release, controller computation, buffering, or scheduler alignment are included. The paper therefore distinguishes this analytically known serialization time from an experimentally measured end-to-end loop delay.

End-to-end timing model.

For control frame k, the physical implementation delay is represented as

$$\tau_{\text{loop,k}} = \tau_{\text{tx,k}} + \tau_{\text{proc,k}} + \tau_{\text{rx,k}} + \tau_{\text{sched,k}}, \tag{84}$$

where the terms denote plant-to-controller transmission, embedded computation, controller-to-plant transmission, and residual scheduler/buffering/sample-alignment effects, respectively. A stochastic timing component may be represented by

$$\tau_{\text{loop,k}} = \bar{\tau}_{\text{loop}} + \tilde{\tau}_k, \qquad E[\tilde{\tau}_k] = 0, \tag{85}$$

with sample standard deviation

$$\sigma_\tau = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(\tau_{\text{loop,k}} - \bar{\tau}_{\text{loop}})^2}. \tag{86}$$

No miss-distance contribution is assigned to an individual timing term without a controlled sensitivity experiment.

Embedded execution time.

The STM32F429 operates at 180 MHz with a single-precision hardware FPU [15]. Controller execution time is measured by setting a GPIO at the beginning of the guidance/control update and clearing it at completion. The mean and maximum measured execution times are

$$T_{\text{exec,avg}} = 0.356 \text{ ms}, \qquad T_{\text{exec,max}} = 0.392 \text{ ms}. \tag{87}$$

At $f_s = 50$ Hz, $T_s = 20$ ms. Define the control-frame execution occupancy as

$$\eta_{\text{exec}} = \frac{T_{\text{exec}}}{T_s} \times 100\%, \qquad M_t = 1 - \frac{T_{\text{exec}}}{T_s}. \tag{88}$$

The mean values are $\eta_{\text{exec}} = 1.78\%$ and $M_t = 98.22\%$, while the maximum observed execution time corresponds to $\eta_{\text{exec,max}} = 1.96\%$. These quantities characterize only the algorithm computation within the frame; they are not interpreted as total microcontroller utilization.

Combining the ideal serial wire time with measured controller computation gives a lower bound on the deterministic packet-plus-compute chain,

$$T_{\text{chain,lb}} = T_{\text{ser,in}} + T_{\text{exec}} + T_{\text{ser,out}}. \tag{89}$$

Hence

$$T_{\text{chain,lb,avg}} \approx 1.919 \text{ ms}, \qquad T_{\text{chain,lb,max}} \approx 1.955 \text{ ms}. \tag{90}$$

These are lower bounds, not measured round-trip delays, because scheduler, buffering, and packet-processing overhead remain outside the calculation.

Table 2: Timing quantities for the embedded HIL control frame.

Quantity	Value	Basis
Controller period T_s	20.000 ms	configured
12-byte input serialization	1.042 ms	analytical, 115200 baud 8N1
6-byte output serialization	0.521 ms	analytical, 115200 baud 8N1
Mean controller execution	0.356 ms	measured GPIO pulse
Maximum controller execution	0.392 ms	measured GPIO pulse
Mean chain lower bound	1.919 ms	serialization + execution
Maximum chain lower bound	1.955 ms	serialization + execution

Clocking and experiment reproducibility.

The ARK5000 real-time scheduler paces the experiment at 50 Hz. The STM32F429 performs one guidance/control update after receipt of each new input packet. Since the two devices use independent clocks and no explicit phase synchronization is imposed, cycle-to-cycle arrival and scheduler variations are treated as potential jitter rather than folded into the deterministic serialization calculation. For each paired MIL and HIL experiment, the same initial conditions, aerodynamic database, actuator model, controller configuration, plant integration step, and prescribed turbulence seed are used. This paired protocol reduces scenario variability when interpreting the MIL–HIL difference.



Fig. 5: Representative real-time monitoring interface during HIL execution. Displayed channels include altitude, range, velocity, attitude, angular rates, Mach number, aerodynamic angles, dynamic pressure, and actuator deflections. The monitoring interface is not part of the feedback path.

6 Experimental Results and Analysis

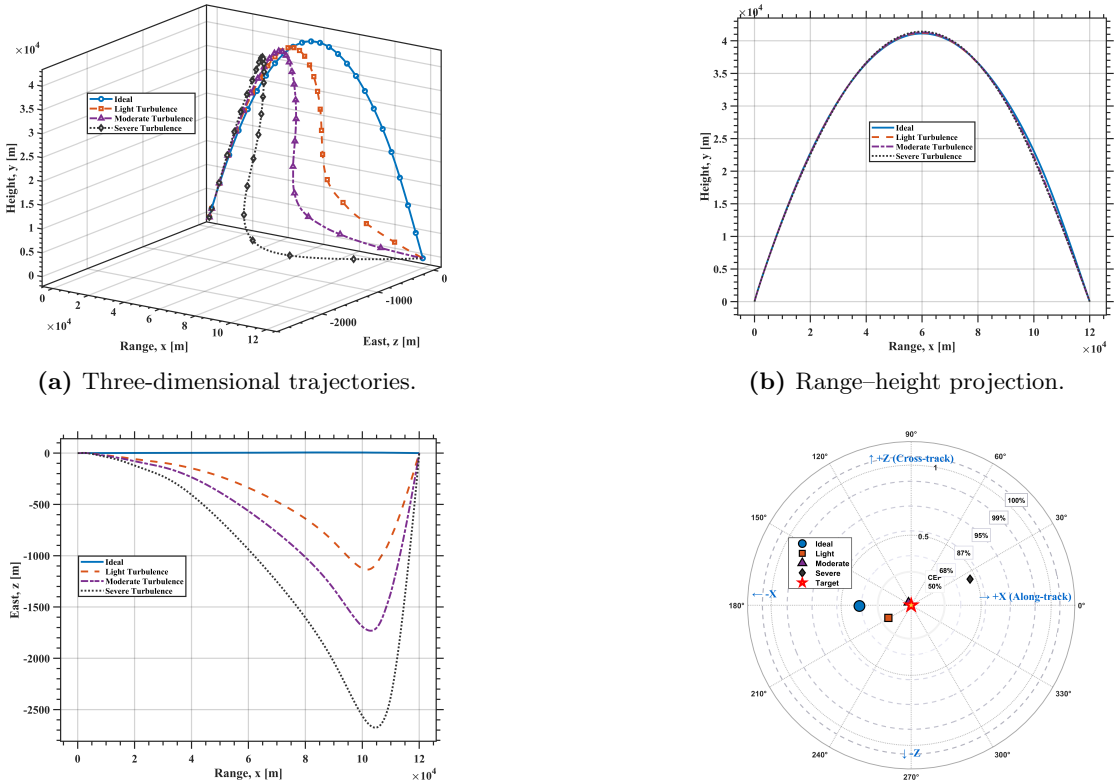
Terminal-guidance performance is evaluated under ideal, light, moderate, and severe atmospheric-turbulence conditions using the controlled MIL–HIL framework described in Section 5. The comparison focuses on three aspects: terminal interception accuracy, closed-loop vehicle response, and real-time implementation performance.

For every turbulence condition, the corresponding MIL and HIL cases use the same initial state, nonlinear 6-DOF plant, aerodynamic model, actuator constraints, controller parameters, and prescribed turbulence realization. Consequently, each MIL–HIL pair represents a controlled comparison of the software reference implementation against its real-time embedded implementation.

All results reported in this section correspond to one fixed turbulence realization for each severity level. The results therefore characterize the specific paired experiments considered here rather than the statistical distribution of performance over the complete turbulence ensemble. In particular, the ordering of miss distance with turbulence severity should not be interpreted as a monotonic statistical trend without additional Monte Carlo experiments.

Model-in-the-Loop reference performance.

The MIL configuration provides the reference closed-loop response before the additional hardware communication, scheduling, and embedded-interface effects associated with the physical HIL configuration are introduced. Figure 6 presents the resulting trajectories and terminal locations for the four atmospheric conditions.



(a) Three-dimensional trajectories.

(b) Range–height projection.

(c) Range–lateral projection.

(d) Terminal impact locations.

Fig. 6: MIL trajectories and terminal impact locations under ideal, light, moderate, and severe turbulence conditions. Each turbulence case represents one fixed disturbance realization.

For consistency with the available terminal-position data, the planar range–lateral terminal error is defined as

$$d_{\text{miss}} = \sqrt{(X_f - X_{\mathcal{T}})^2 + (Z_f - Z_{\mathcal{T}})^2}, \tag{91}$$

where X_f and Z_f are the terminal vehicle coordinates and $X_{\mathcal{T}}$ and $Z_{\mathcal{T}}$ denote the corresponding target coordinates. Equation (91) therefore represents the terminal error in the range–lateral plane and should not be interpreted as the minimum three-dimensional separation unless the terminal event coincides with the target-altitude plane.

The MIL implementation maintains sub-meter terminal error for all four investigated disturbance realizations, with $d_{\text{miss}} = 0.027$–0.460 m. These results demonstrate satisfactory closed-loop tracking for the selected engagement and disturbance sequences.

The minimum error occurs in the moderate-turbulence realization rather than in the ideal case. Because only one seeded realization is evaluated at each nonzero turbulence level, this non-monotonic ordering cannot be interpreted as evidence that moderate turbulence improves guidance performance. Instead, it indicates that

Table 3: MIL terminal errors for the four atmospheric conditions.

Condition	Range error [m]	Lateral error [m]	d_{miss} [m]
Ideal	−0.369	−0.005	0.369
Light turbulence	−0.163	−0.090	0.186
Moderate turbulence	−0.019	0.020	0.027
Severe turbulence	0.420	0.187	0.460

terminal error is sensitive to the temporal realization of the disturbance and its interaction with the finite-duration guidance response. Statistical conclusions regarding turbulence severity therefore require an ensemble of independent realizations.

Figure 7 presents representative MIL flight-state histories, including angle of attack α, sideslip angle β, static margin, pitch attitude, Mach number, and vehicle speed.

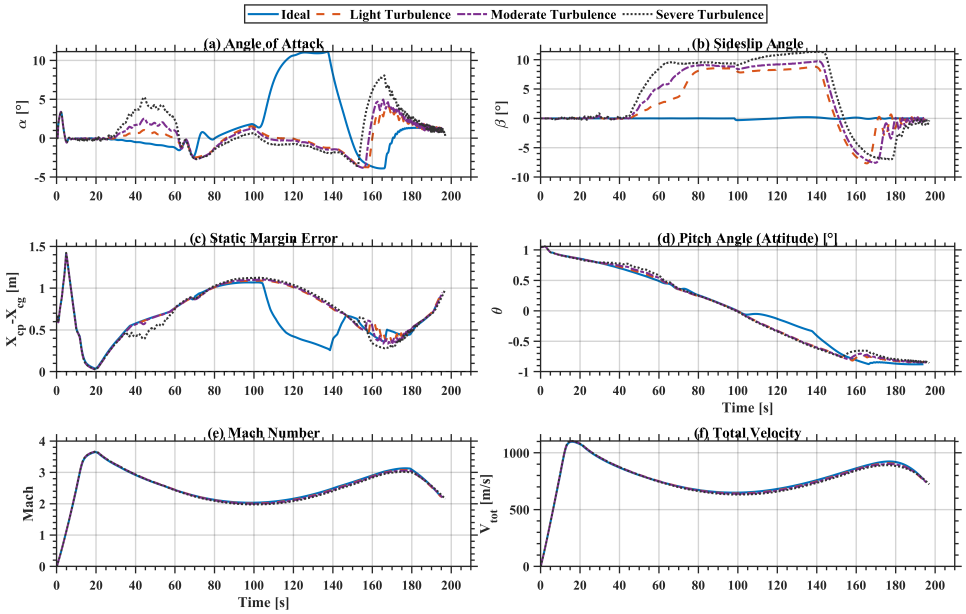


Fig. 7: MIL flight-state histories under ideal, light, moderate, and severe turbulence conditions, showing angle of attack α, sideslip β, static margin $(X_{cp} - X_{cg})$, pitch angle θ, Mach number, and vehicle speed.

Hardware-in-the-Loop experimental performance.

The corresponding HIL experiments retain the same nonlinear plant, initial conditions, actuator model, controller configuration, and prescribed turbulence realization as the MIL cases. The principal difference is that the TPNG and autopilot algorithms execute on the STM32F429 and communicate with the real-time plant through the physical serial interface described in Section 5.

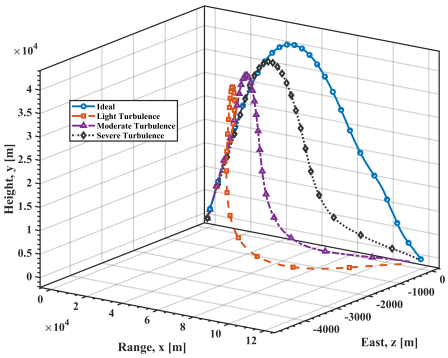
Figure 8 presents the resulting HIL trajectories and terminal locations.

The terminal errors measured in the HIL experiments are summarized in Table 4.

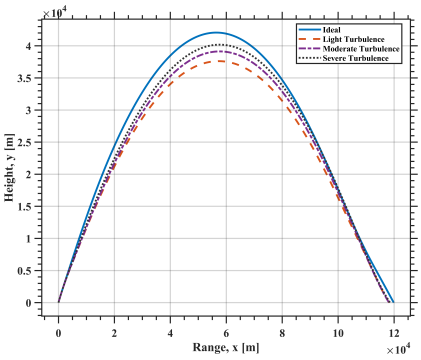
Table 4: HIL terminal errors under the four atmospheric conditions.

Condition	Range error [m]	Lateral error [m]	d_{miss} [m]
Ideal	−137.6	−3.77	137.6
Light turbulence	−1814.5	−433.5	1865.6
Moderate turbulence	−1829.4	−255.0	1847.1
Severe turbulence	−1472.5	−26.6	1472.7

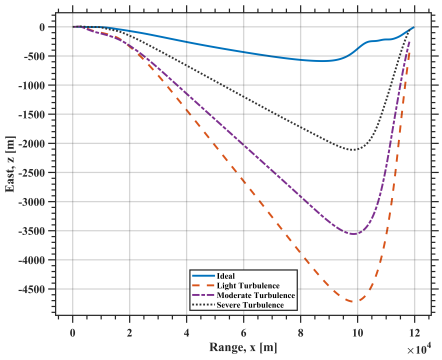
The HIL experiments exhibit substantially larger terminal errors than the corresponding MIL cases. Even in the absence of turbulence, the terminal error increases from 0.369 m in MIL to 137.6 m in HIL. Since the ideal case



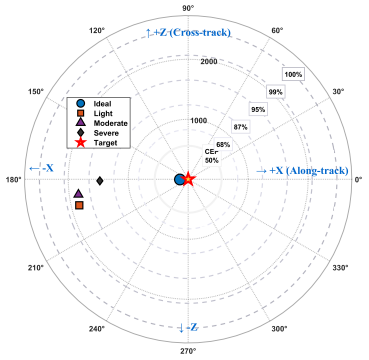
(a) Three-dimensional trajectories.



(b) Range–height projection.



(c) Range–lateral projection.



(d) Terminal impact locations.

Fig. 8: HIL trajectories and terminal impact locations under the four test conditions. Relative to the corresponding MIL runs, substantially larger terminal errors are observed after deployment of the guidance and control algorithms on the embedded hardware.

contains no stochastic atmospheric forcing, this result provides direct evidence that the physical implementation chain introduces a substantial performance difference relative to the software reference configuration.

For the three nonzero-turbulence realizations, the HIL terminal errors range from approximately 1.47 to 1.87 km. Their ordering is again non-monotonic: the severe-turbulence realization produces a smaller terminal error than the light and moderate realizations. Because each turbulence condition is represented by only one seeded run, the present results do not support a statistical ranking of performance with disturbance severity.

The measurements instead demonstrate that the investigated HIL runs exhibit substantially larger errors than their paired MIL references. Determining whether this increased sensitivity is caused predominantly by communication delay, timing jitter, data quantization, sample alignment, or interactions among these effects requires separate controlled sensitivity experiments.

MIL–HIL implementation gap.

To quantify the observed implementation-induced performance difference, define

$$\Delta d_{\mathrm{MIL-HIL}} = d_{\mathrm{HIL}} - d_{\mathrm{MIL}}. \tag{92}$$

The resulting paired comparison is given in Table 5.

Table 5: Paired MIL–HIL comparison of terminal miss distance.

Condition	**MIL [m]**	**HIL [m]**	$\Delta d_{\mathrm{MIL-HIL}}$ **[m]**
Ideal	0.369	137.6	137.231
Light turbulence	0.186	1865.6	1865.414
Moderate turbulence	0.027	1847.1	1847.073
Severe turbulence	0.460	1472.7	1472.240

An absolute difference is reported instead of relying primarily on a multiplicative degradation factor. Because the MIL errors are extremely small, particularly the 0.027 m moderate-turbulence result, a ratio can become numerically very large and may exaggerate the practical interpretation of the MIL–HIL difference.

The ideal experiment is especially useful for separating turbulence from embedded implementation effects. With atmospheric turbulence removed, the observed difference is

$$\Delta d_{\text{ideal}} = 137.6 - 0.369 = 137.231 \text{ m}. \tag{93}$$

This value represents the experimentally observed aggregate implementation gap for the ideal test configuration. It should not, however, be assigned directly to a single mechanism such as serial latency without further controlled experiments.

Closed-loop flight-state comparison.

Figure 9 presents the corresponding HIL flight-state histories. Compared with the MIL reference, the HIL responses exhibit greater variation in the aerodynamic angles, particularly in α and β, while the overall Mach-number and velocity histories remain of comparable scale.

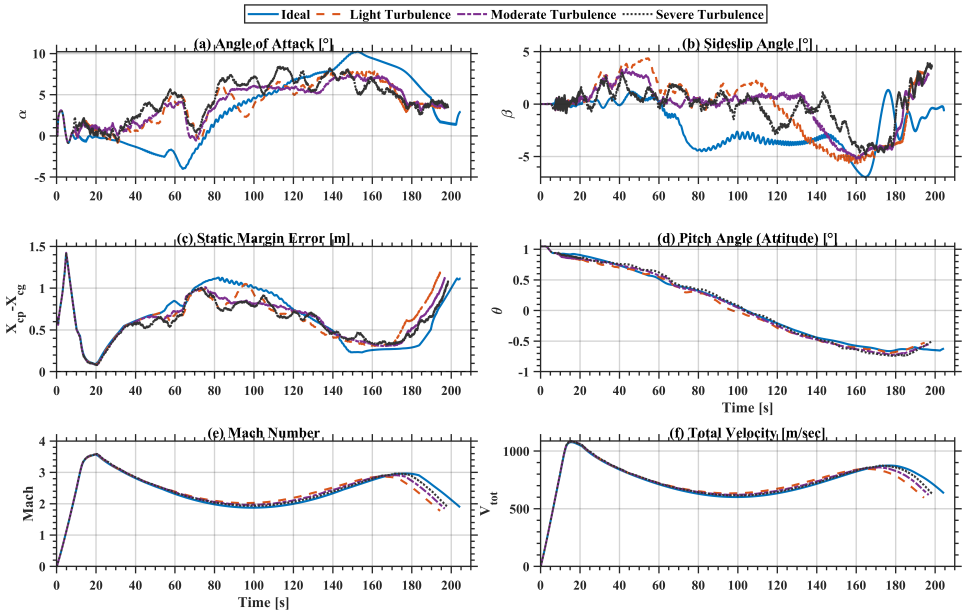


Fig. 9: HIL flight-state histories under ideal, light, moderate, and severe turbulence conditions. Increased variation in aerodynamic angles is observed relative to the corresponding MIL responses, while the principal vehicle-energy trends remain comparable.

The positive static-margin histories indicate that the investigated vehicle remains statically stable over the reported trajectories. However, static margin alone is not sufficient to establish closed-loop dynamic stability. Accordingly, the larger HIL terminal errors are interpreted primarily as a loss of guidance and tracking accuracy rather than as evidence of complete vehicle instability.

Real-time computational performance and interpretation.

As described in Section 5, the STM32F429 requires an average of

$$T_{\text{exec,avg}} = 0.356 \text{ ms} \tag{94}$$

to execute the embedded guidance and control algorithms within the 20 ms control interval. The corresponding average execution-time occupancy is

$$\eta_{\text{cpu}} = \frac{0.356}{20} \times 100 \approx 1.78\%. \tag{95}$$

Therefore, embedded algorithm execution occupies only a small fraction of the available sampling interval. This finding rules out computational overload as the direct explanation for the measured performance degradation; however, it does not by itself identify the dominant source of the MIL–HIL discrepancy.

The remaining implementation chain includes serial transmission, serialization and deserialization, sample alignment, task scheduling, data representation, and timing jitter. Their individual effects cannot be identified uniquely from the four terminal experiments alone. A rigorous causal decomposition requires controlled ablation or delay-injection experiments in which these effects are introduced independently into the MIL configuration.

The present experimental results therefore support two conclusions. First, the measured algorithm execution time leaves substantial computational margin for the implemented TPNG and autopilot calculations at 50 Hz. Second, the complete physical implementation chain produces a substantial degradation in terminal accuracy relative to the matched MIL reference. Quantitative attribution of this degradation to individual communication and timing mechanisms is treated as a separate validation requirement rather than inferred directly from the terminal miss-distance measurements.

7 Conclusions and Design Recommendations

A controlled MIL–HIL framework was used to examine how a real embedded implementation changes terminal-guidance performance relative to a matched numerical reference. TPNG and a gain-scheduled modified PI autopilot with direct angular-rate damping were executed on an STM32F429 ARM Cortex-M4 processor, while an ARK5000 real-time target propagated the nonlinear 6-DOF vehicle, actuator, aerodynamic, and atmospheric models. The paired configurations retained the same plant assumptions, actuator limits, controller update rate, and prescribed turbulence realization.

For the four investigated realizations, MIL planar terminal errors were 0.027–0.460 m, whereas the corresponding HIL errors were 137.6–1865.6 m. The ideal paired case produced an aggregate MIL–HIL difference of 137.231 m even without stochastic turbulence forcing. This demonstrates a substantial implementation-induced loss of terminal accuracy for the tested hardware/interface configuration, but the current experiment does not uniquely assign that loss to a single mechanism.

The embedded guidance-and-control computation required 0.356 ms on average and 0.392 ms at the measured maximum, corresponding to 1.78% and 1.96% of the 20 ms control frame, respectively. Computational overload can therefore be excluded as the direct cause of the measured degradation. The remaining implementation chain includes serial transmission, packet handling, task release, scheduler alignment, clock asynchrony, and numerical data representation. Their individual contributions require controlled ablation or delay-injection tests rather than inference from terminal error alone.

Design recommendations.

Three practical recommendations follow from the experiments. First, embedded guidance feasibility should be assessed using an end-to-end timing budget rather than processor execution time alone. Second, packet serialization, computation, buffering, and scheduler alignment should be measured or bounded separately before a HIL miss error is attributed to communication latency. Third, paired stochastic disturbances should be retained when comparing MIL and HIL so that a change in the random forcing is not confused with an implementation effect.

Limitations.

The present study reports one fixed turbulence realization for each nonzero severity level; it therefore does not provide ensemble statistics or a monotonic turbulence- severity trend. The target is stationary and the initial engagement geometry is fixed. The current experiments quantify the aggregate physical implementation effect but do not independently isolate serial delay, scheduler jitter, quantization, or sample alignment. In addition, the aerodynamic database, actuator model, controller schedule, and X-fin allocation correspond to the investigated vehicle configuration. Finally, the reported miss metric is the planar range–lateral terminal error at the defined terminal event, not an event-interpolated minimum three-dimensional separation.

Future work.

The highest-priority extension is a controlled MIL delay-injection campaign in which deterministic delay and timing jitter are varied independently while all other conditions are fixed. This will permit direct identification of

$$d_{\text{miss}} = f(\tau_{\text{delay}}, \sigma_{\text{jitter}}) \tag{96}$$

and comparison of the resulting sensitivity with the HIL measurements. Paired Monte Carlo MIL–HIL experiments should then be performed using multiple independent von Kármán realizations to report means, standard deviations, confidence intervals, and distributions of the paired implementation gap. Further extensions include a higher-bandwidth communication interface, explicit packet-quantization tests, moving and maneuvering targets, seeker noise and latency, and comparison of TPNG with impact- angle-constrained and optimal terminal-guidance laws.

Declarations

- **Author Contributions:** Conceptualization and methodology: A.M.K., E.S., M.S.M., and M.A.H.; software development, investigation, data curation, and formal analysis: A.M.K., M.S.M., E.S., and M.A.H.; writing—original draft preparation: A.M.K.; writing—review and editing: all authors. All authors have read and approved the final manuscript.
- **Funding:** The authors received no external funding for this study.
- **Competing Interests:** The authors declare that they have no competing financial or non-financial interests relevant to this study.
- **Data Availability:** The data generated and analyzed during this study are available from the corresponding author upon reasonable request, subject to applicable institutional approval.

References

[1] Siouris, G.M.: Missile Guidance and Control Systems. Springer, New York, NY, USA (2004)

[2] Zarchan, P.: Tactical and Strategic Missile Guidance, 6th edn. American Institute of Aeronautics and Astronautics, Reston, VA, USA (2012)

[3] Wang, Z., Cai, P., Gong, Z., Zhang, C., Zhao, S., Wu, J., Dong, P.: Review on guidance and control of aerospace vehicles: Recent progress and prospect. Aerospace Systems **7**(2), 175–192 (2024) https://doi.org/10.1007/s42401-024-00273-6

[4] Oberkampf, W.L., Roy, C.J.: Verification and Validation in Scientific Computing. Cambridge University Press, Cambridge, UK (2010)

[5] Stevens, B.L., Lewis, F.L., Johnson, E.N.: Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd edn. Wiley, Hoboken, NJ, USA (2015)

[6] Steffensen, R., Steinert, A., Mbikayi, Z., Raab, S., Angelov, J., *et al.*: Filter and sensor delay synchronization in incremental flight control laws. Aerospace Systems **6**, 285–304 (2023) https://doi.org/10.1007/s42401-022-00186-2

[7] Ji, H., Chen, R., Li, P.: Analysis of helicopter handling quality in turbulence with recursive von Kármán model. Journal of Aircraft **54**(5), 1631–1639 (2017) https://doi.org/10.2514/1.C034189

[8] Beal, T.R.: Digital simulation of atmospheric turbulence for Dryden and von Kármán models. Journal of Guidance, Control, and Dynamics **16**(1), 132–138 (1993) https://doi.org/10.2514/3.11437

[9] U.S. Department of Defense: Flying qualities of piloted aircraft. Military Handbook MIL-HDBK-1797, U.S. Department of Defense, Washington, DC, USA (1997)

[10] Mihalič, F., Truntič, M., Hren, A.: Hardware-in-the-loop simulations: A historical overview of engineering challenges. Electronics **11**(15), 2462 (2022) https://doi.org/10.3390/electronics11152462

[11] Kaden, A., Boche, B., Luckner, R.: Hardware-in-the-loop flight simulator: An essential part in the development process for the automatic flight control system of a utility aircraft. In: Advances in Aerospace Guidance, Navigation and Control, pp. 491–504. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38253-6_34

[12] Benninghoff, H., Rems, F., Boge, T.: Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing. Acta Astronautica **102**, 67–80 (2014) https://doi.org/10.1016/j.actaastro.2014.05.014

[13] Panchal, P., Hein, L., Myschik, S.: DO-178C software development lifecycle with ARP4754A system-level verification. Aerospace Systems (2026) https://doi.org/10.1007/s42401-026-00455-4

[14] Saleab, M., Sax, F., Schumann, J., *et al.*: Low-level memory and timing analysis of flight code for unmanned aerial systems. Aerospace Systems **7**, 209–225 (2024) https://doi.org/10.1007/s42401-024-00279-0

[15] STMicroelectronics: STM32F427xx/STM32F429xx: 32-bit Arm Cortex-M4 MCU with FPU, up to 180 MHz.

STMicroelectronics, (2025). STMicroelectronics. Production datasheet

[16] Wang, Y., Zhao, S., Wu, J., Tan, L., Dong, P., *et al.*: Sampled-data self-learning observer based attitude tracking control against sensor-actuator faults. Aerospace Systems **8**, 659–671 (2025) https://doi.org/10.1007/s42401-025-00341-5

[17] Shneydor, N.A.: Missile Guidance and Pursuit: Kinematics, Dynamics and Control. Horwood Publishing, Chichester, UK (1998)

[18] Kumar, A., Ojha, A., Yadav, S., Kumar, A.: Real-time interception performance evaluation of certain proportional navigation based guidance laws in aerial ground engagement. Intelligent Service Robotics **15**(1), 95–114 (2022) https://doi.org/10.1007/s11370-021-00404-4

[19] Pandey, D., Wairya, S., Sharma, M., Gupta, A.K., Kakkar, R., Pandey, B.K.: An approach for object tracking, categorization, and autopilot guidance for passive homing missiles. Aerospace Systems **5**(4), 553–566 (2022) https://doi.org/10.1007/s42401-022-00150-0

[20] Zipfel, P.H.: Modeling and Simulation of Aerospace Vehicle Dynamics, 3rd edn. American Institute of Aeronautics and Astronautics, Reston, VA, USA (2014)

[21] Hajjem, M., Victor, S., Melchior, P., Lanusse, P., Thomas, L.: Wind turbulence modeling for real-time simulation. Fractional Calculus and Applied Analysis **26** (2023) https://doi.org/10.1007/s13540-023-00165-0

[22] Blakelock, J.H.: Automatic Control of Aircraft and Missiles, 2nd edn. Wiley, New York, NY, USA (1991)

[23] Franklin, G.F., Powell, J.D., Emami-Naeini, A.: Feedback Control of Dynamic Systems, 8th edn. Pearson, London, UK (2019)

[24] Thai, S., Theodoulis, S., Roos, C., Biannic, J.-M.: Robust gain-scheduled autopilot design with anti-windup compensation for a guided projectile. CEAS Aeronautical Journal **14**(3), 765–786 (2023) https://doi.org/10.1007/s13272-023-00668-9

[25] Huang, B., Zhai, M., Lu, B., Li, Q.: Gain-scheduled anti-windup PID control for LPV systems under actuator saturation and its application to aircraft. Aerospace Systems **5**, 445–454 (2022) https://doi.org/10.1007/s42401-022-00143-z

[26] Kotitschke, C., Rupprecht, T., Bouhlel, E., *et al.*: Kriging-accelerated worst-case search for envelope-wide flight control law optimization. Aerospace Systems (2026) https://doi.org/10.1007/s42401-026-00498-7

[27] Zhang, J., Lu, Z., Holzapfel, F.: Null-space control allocation for generalized load reduction in eVTOL aircraft. Aerospace Systems **9**, 397–411 (2026) https://doi.org/10.1007/s42401-026-00483-0