AIAA SciTech Forum
12-16 January 2026, Orlando, FL
AIAA SCITECH 2026 Forum

10.2514/6.2026-0548

Incremental Nonlinear Dynamic Inversion based Control of a Generic Hypersonic Vehicle (GHAME)

T. H. Mueller* and S. Theodoulis †
Delft University of Technology, Delft, 2628CD, Netherlands

I. Sarras‡
ONERA - The French Aerospace Lab, Palaiseau, 91123, France

The Generic Hypersonic Aerodynamics Model Example (GHAME) provides a practical benchmark for evaluating advanced control strategies for hypersonic vehicles. Its nonlinear dynamics and strong aero–propulsive coupling create challenges well suited to nonlinear inversion methods. This work develops a hierarchical control architecture based on time–scale separation, combining NDI for attitude and position control with Incremental Nonlinear Dynamic Inversion (INDI) for angular–rate and velocity control. The controller is implemented in MATLAB and Simulink and evaluated under synchronized and desynchronized sensor delays. The results show that delay synchronization markedly increases the admissible delay margin. The study also reveals a fundamental limitation in the lateral axis: the lateral-directional dynamics of GHAME are too fast to satisfy the time-scale separation assumption required by INDI, leading to unreliable linear stability predictions. In contrast, the longitudinal dynamics do satisfy this assumption and remain well suited to inversion-based control. Overall, the NDI–INDI structure is effective for the longitudinal motion when delays are synchronized, but the intrinsic speed of the lateral dynamics imposes a major constraint on its applicability for lateral control.

Nomenclature

$\alpha,\ \mu,\ \beta$	=	AoA, bank, sideslip angles	C_L, C_Y, C_D	=	Stability-axis force coeffs.
C_X, C_Y, C_Z	=	Body-axis force coeffs.	C_l, C_m, C_n	=	Moment coeffs.
$c_{\text{ref}},\ b_{\text{ref}}$	=	Ref. chord and span	$\omega_{act},\ \zeta_{act}$	=	Act. natural freq. and damp.
p, q, r	=	Inertial angular rates	$\delta_a, \delta_e, \delta_r$	=	Aileron, elevator, rudder defl.
δ_{vl}, δ_{vr}	=	Left/right elevon	$u_N,\ u_E,\ u_U$	=	North, East, Up velocity comp.
f_P	=	Propulsive force	$\omega_H,\ \zeta_H$	=	Noise filter natural freq. and damp.
ρ	=	Air density	$\bar{q}$	=	Dynamic pressure
M	=	Mach number	ω_a	=	Anti-aliasing natural freq.
g_0	=	Standard gravity	R_0	=	Standard radius Earth
γ	=	Flightpath angle	χ	=	Heading angle
$\boldsymbol{f}_g$	=	Grav. force	$\boldsymbol{f}_{a,p}$	=	Aero-prop. force
V_K	=	Groundspeed	δ_t	=	Throttle setting
$(\boldsymbol{m}_c)_\delta$	=	Control moment coefs.	S	=	Ref. surface area
$\boldsymbol{m}_a$	=	Aero. moment coefs.	Δt	=	Sample time
$\boldsymbol{m}_B$	=	External moments	τ_{SD}	=	Sensor delay
BW	=	Bandwidth	τ_{CD}	=	Computational delay
$\omega_\oplus$	=	Earth angular rate	$[T]^{VG}$	=	Geoc.-to-vel. transform
ω	=	Inertial angular rate	$[T]^{VB}$	=	Body.-to-vel. transform
λ, l, h	=	Lat., long., altitude	$\boldsymbol{I}$	=	Moment of inertia matrix

*MSc, student, Control & Simulation division, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, Netherlands; thomas-harald.mueller1@outlook.de.

†Associate Professor, Control & Simulation division, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, Netherlands; S.Theodoulis@tudelft.nl. Associate Fellow AIAA.

‡Research engineer, Information Processing and Systems Department, ONERA, Univ. Paris-Saclay, F-91123 Palaiseau, France; ioannis.sarras@onera.fr.

I. Introduction

Recent interest in applying Incremental Nonlinear Dynamic Inversion (INDI) to hypersonic vehicles is driven by the extreme control challenges posed by these platforms. Air-breathing hypersonic vehicles exhibit strong propulsion–airframe coupling, highly nonlinear and speed-dependent stability characteristics, and structural flexibility arising from slender geometries and thermal loading [1, 2]. These effects produce rapid variations in aerodynamic forces and moments that are difficult to model accurately in real time. The reduced reliance of INDI on precise aerodynamic models therefore makes it an appealing candidate for such environments.

INDI achieves this reduced model dependence by using measurements of the controlled-variable derivatives to construct an incremental inversion of the dynamics. In an aircraft rate-control study, Smith [3] demonstrated that the resulting control law becomes largely insensitive to uncertainties in aerodynamic coefficients, centre-of-gravity position, and inertial properties, since these effects are already embedded in the measured angular accelerations. The same study, however, showed that this benefit comes at the cost of increased sensitivity to sensor delay.

To assess the applicability of INDI in the hypersonic regime, this study uses the Generic Hypersonic Aerodynamic Model Example (GHAME), a publicly released NASA model of a single-stage-to-orbit vehicle. GHAME combines aerodynamic and propulsive characteristics of turbojet, ramjet, and scramjet propulsion across Mach 0–24 and captures the dominant nonlinearities governing hypersonic flight [4]. Using this model, the paper develops and evaluates a cascaded INDI controller and examines its robustness under synchronized and desynchronized sensor delays to determine whether the method's inherent robustness extends to flight conditions characterized by strong nonlinearities and rapid variations in aerodynamic behavior. The main contribution is the first systematic application and assessment of an INDI-based architecture on a high-fidelity hypersonic model.

II. Modelling

A. GHAME Vehicle Model

The GHAME is a high-fidelity aerodynamic and geometric model developed at the NASA Ames Research Center and extensively tested at the Dryden Flight Research Center [4]. It was conceived during the early phases of the NASP initiative to provide accurate and physically representative aerodynamic data for hypersonic vehicle research. GHAME was designed as a generic hypersonic configuration capable of performing a SSTO mission, involving horizontal takeoff using air-breathing propulsion, acceleration to orbital velocity, orbital insertion, atmospheric reentry, and unpowered gliding recovery. The aerodynamic database spans the complete mission envelope, including subsonic, transonic, hypersonic, and reentry flight regimes [5].

The model enables high-fidelity Six-Degree-of-Freedom (6 DOF) simulations and produces a dynamic response representative of hypersonic flight. Consequently, GHAME has become a benchmark for evaluating control architectures, developing guidance strategies, and conducting trajectory optimization studies. Multiple versions of the model have been implemented in Fortran by P. Zipfel, using the original NASA Ames aerodynamic dataset as documented in White et al. [5] and detailed in his textbook [6]. More recently, Goz [7] developed a Simulink-based implementation of a simplified GHAME model assuming a nonrotating, flat Earth, constant mass and first-order actuator dynamics.

B. Dynamics

The simulation in this work is a Simulink implementation of the full six-degree-of-freedom Hyper6 C++ flight-dynamics model described by Zipfel [6].* Hyper6 is the most complete and rigorously tested version of the GHAME model. It incorporates a rotating elliptical Earth, propellant depletion with the associated variation in mass and moment of inertia, and the coupled translational and rotational equations of motion. The Simulink implementation used here is a direct port of these dynamics and has been extensively verified, providing a high-fidelity environment for evaluating the proposed control architecture.

*Simulations can be downloaded from: https://arc.aiaa.org/doi/suppl/10.2514/4.107535

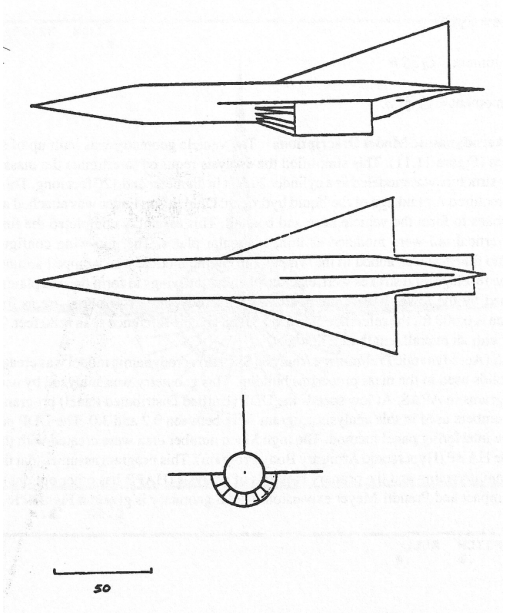

Fig. 1 Schematic configuration of the GHAME vehicle model

C. Aerodynamics and Forces

The GHAME aerodynamic database was developed by combining theoretical flow models and empirical data from multiple legacy hypersonic vehicle configurations. Longitudinal aerodynamic coefficients were constructed using blended data from the Space Shuttle Orbiter, lifting-body vehicles, a swept double-delta wing platform, and a 60° half-angle cone. For lateral-directional coefficients, Space Shuttle and double-delta data were used at Mach numbers below 8. Above this threshold, only Space Shuttle data were retained. Drag coefficients were scaled to match lift-to-drag ratios consistent with measured Space Shuttle performance. All coefficients were normalized using a fixed aerodynamic reference area and span. To extend the aerodynamic model into the hypersonic regime, a modified Newtonian impact theory was applied. This combination of analytical and empirical methods yields an aerodynamic model capable of capturing the nonlinearities, cross-coupling, and regime transitions critical for high-fidelity hypersonic flight simulation [5].

The aerodynamic force and moment coefficients are expressed as linear combinations of the control surface deflections, angular rates, sideslip angle β, and angle of attack α. In simulations with wind, the aerodynamic angles are used directly, whereas in wind-free conditions the corresponding kinematic angles apply. All aerodynamic data follow the GHAME reference model and were generated under the assumption of zero sideslip, although β may vary during simulation. Additionally, the coefficients C_{L_q}, C_{Y_0}, C_{Y_p}, C_{Y_r}, C_{l_0}, and C_{n_0} are negligible across the evaluated flight envelope [5]. The force coefficients are defined as follows:

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha}\,\alpha + C_{L_{\delta_e}}\,\delta_e, \\ C_Y &= C_{Y_\beta}\,\beta + C_{Y_{\delta_a}}\,\delta_a + C_{Y_{\delta_r}}\,\delta_r, \\ C_D &= C_{D_0} + C_{D_\alpha}\,\alpha, \end{aligned} \tag{1}$$

where δ_e is the elevator deflection, δ_a the aileron deflection, and δ_r the rudder deflection. The moment coefficients are calculated as follows:

$$\begin{aligned} C_m &= C_{m_0} + C_{m_\alpha}\,\alpha + C_{m_{\delta_e}}\,\delta_e + C_{m_q}\,\frac{q\,c_{\text{ref}}}{2V_{\text{TAS}}}, \\ C_l &= C_{l_\beta}\,\beta + C_{l_{\delta_a}}\,\delta_a + C_{l_{\delta_r}}\,\delta_r + C_{l_p}\,\frac{p\,b_{\text{ref}}}{2V_{\text{TAS}}} + C_{l_r}\,\frac{r\,b_{\text{ref}}}{2V_{\text{TAS}}}, \\ C_n &= C_{n_\beta}\,\beta + C_{n_{\delta_r}}\,\delta_r + C_{n_{\delta_a}}\,\delta_a + C_{n_p}\,\frac{p\,b_{\text{ref}}}{2V_{\text{TAS}}} + C_{n_r}\,\frac{r\,b_{\text{ref}}}{2V_{\text{TAS}}}. \end{aligned} \tag{2}$$

where p, q, and r denote the roll, pitch, and yaw rates of the vehicle with respect to Earth in body axes, respectively. Then c_{ref} is the reference chord, b_{ref} the reference span of the aircraft and V_{TAS} is the true airspeed. All partial derivatives with respect to angles have units of $1/^\circ$, while those with respect to angular rates have units of $1/\text{rad}$.

The lift coefficient C_L is defined as positive upwards and perpendicular to the velocity vector, the drag coefficient C_D is positive in the direction opposite to flight and the side force coefficient C_Y is positive to the right (starboard). Rolling moment C_l is positive for right–wing–down roll, pitching moment C_m for nose-up rotation, and yawing moment C_n for nose-right yaw. Since the aerodynamic force coefficients are expressed in stability coordinates, they must be transformed into body coordinates using α:

$$\begin{aligned} C_X &= -C_D \cos\alpha + C_L \sin\alpha, \\ C_Z &= -C_D \sin\alpha - C_L \cos\alpha. \end{aligned} \tag{3}$$

The total force vector $[f_{a,p}]^B$ is computed by combining the aerodynamic forces with the propulsive force. The total force vector in body coordinates is given by:

$$[f_{a,p}]^B \equiv \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^B = \begin{bmatrix} \bar{q}SC_X + f_P \\ \bar{q}SC_Y \\ \bar{q}SC_Z \end{bmatrix}, \tag{4}$$

where $\bar{q} = \frac{1}{2}\rho V_{\text{TAS}}^2$ is the dynamic pressure, in which ρ is the air density, S is the wing reference area and f_P is the propulsive force. The aerodynamic moment vector in the body frame is computed as:

$$[m_B]^B \equiv \begin{bmatrix} L \\ M \\ N \end{bmatrix}^B = \bar{q}S \begin{bmatrix} C_l b_{\text{ref}} \\ C_m c_{\text{ref}} \\ C_n b_{\text{ref}} \end{bmatrix}, \tag{5}$$

The aerodynamic coefficients are tabulated over a two–dimensional grid in angle of attack and Mach number, spanning $\alpha \in [-3^\circ, 21^\circ]$ and $M \in [0.4, 24]$. For simulation, most coefficients are interpolated independently on this grid. The only exceptions are the grouped forms $C_D = C_{D_0} + C_{D_\alpha}\alpha$, $C_M = C_{M_0} + C_{M_\alpha}\alpha$, and $C_L = C_{L_0} + C_{L_\alpha}\alpha$, which are combined first and then interpolated as single variables. This follows the implementation approach described in [7].

D. Actuator Model

The vehicle contains one rudder and two elevons located at the trailing edge of the wing. The elevons function simultaneously as elevator and aileron control surfaces. Their symmetric and antisymmetric combinations define the elevator and aileron control inputs, respectively, according to:

$$\delta_e = \frac{\delta_{vl} + \delta_{vr}}{2}, \quad \delta_a = \frac{\delta_{vl} - \delta_{vr}}{2}, \tag{6}$$

where δ_{vl} and δ_{vr} are the left and right elevon deflections, both defined as positive in the upward direction. The rudder deflection δ_r is defined to be positive for a trailing-edge right deflection. Each control surface is subject to deflection limits of $\pm 20^\circ$ and rate limits of $\pm 150^\circ/\text{s}$. To realistically capture actuator dynamics, all surfaces are modeled using a second-order system with rate and position limits. The actuator transfer function from commanded deflection δ_c to realized deflection δ is given by:

$$A(s) = \frac{\delta(s)}{\delta_c(s)} = \frac{\omega_{act}^2}{s^2 + 2\zeta_{act}\omega_{act}s + \omega_{act}^2}, \tag{7}$$

where $\omega_{act} = 50\,\text{rad/s}$ is the natural frequency and $\zeta_{act} = 0.707$ is the damping ratio. To enforce physical constraints, the implementation includes a saturation block limiting the input command δ_c to the allowable deflection range $\pm 20^\circ$ and a rate limiter which is calculated as $\frac{150\pi}{180}\frac{2\zeta_{act}}{\omega_{act}}$ applied to the difference in deflection , ensuring the actuator slew rate does not exceed prescribed limits.

E. Propulsion System Model

The engine models a generic combined-cycle propulsion system that switches automatically between turbojet, ramjet, and scramjet modes based on Mach number. The inlet is represented as a variable-geometry system whose effective capture area depends on Mach number and angle of attack, reflecting the changing shock structure on the forebody. The thrust is computed as a product of throttle setting, specific impulse, atmospheric density, true airspeed, and the inlet capture–area coefficient. Implementation details follow the formulation in [6]. The commanded throttle ranges from idle at 0.05 to a maximum value of 2.

Both the capture–area coefficient and the specific impulse are stored on lookup tables in Mach–angle–of–attack and Mach–throttle space, respectively, and are interpolated using the same method applied to the aerodynamic coefficients. The propulsion model also includes fuel consumption. The instantaneous thrust determines the fuel mass flow rate, which is integrated to update the remaining fuel. The total vehicle mass decreases accordingly. As fuel is depleted, the moment of inertia matrix is updated by interpolating linearly between the full–fuel and dry–mass inertia tensors. The simulation terminates automatically once the fuel level drops below zero.

F. Sensor Dynamics

The rate gyroscopes and accelerometers are modeled with explicit sensor dynamics. These devices are subject to two main effects: filtering to prevent aliasing and computation delays introduced by the sensor processor. The anti–aliasing filter is required because the Inertial Measurement Unit (IMU) signals are sampled at a fixed interval $\Delta t = 0.01$ s. The corresponding sampling frequency and Nyquist frequency are:

$$\omega_s = \frac{2\pi}{\Delta t} = 628.32 \text{ rad/s}, \qquad \omega_N = \frac{\omega_s}{2} = 314.16 \text{ rad/s}. \tag{8}$$

Any signal content above ω_N will fold into the lower frequency band, producing aliasing in the sampled data. Since sensor noise is broadband and not naturally band–limited, a low–pass filter is required before sampling. To guarantee sufficient attenuation of out–of–band noise, the cutoff is conservatively placed at half the Nyquist frequency, $\omega_a = \frac{\omega_N}{2} = 157.08$ rad/s. The anti–aliasing dynamics are represented as a first–order low–pass transfer function:

$$G_a(s) = \frac{\omega_a}{s + \omega_a}. \tag{9}$$

In addition to filtering, the IMU introduces a finite computation delay. This effect can be represented exactly by a pure time delay which is given by the nonrational transfer function:

$$G_{\text{SD}}(s) = e^{-\tau_{SD} s}, \tag{10}$$

which is referred to as the sensor delay transfer function. The exact duration of this sensor delay τ_{SD} is uncertain and therefore not modeled as a fixed constant, but it is expected to be on the order of one to several sampling intervals. Since its precise value cannot be specified a priori, the sensor delay is not fixed in the nominal simulation model. Instead, it is varied parametrically in the high-fidelity nonlinear simulation studies to evaluate how well the controller can tolerate such effects.

The overall sensor model therefore consists of a first-order low-pass filter for anti-aliasing, followed by the optional application of the sensor delay transfer function when robustness to measurement delays is investigated.

III. Flight Control System

In this framework, the main control approach is INDI. To the author's knowledge, INDI has not previously been applied to hypersonic vehicle control. It is well suited to this problem because hypersonic flight is characterized by large variations in aerodynamic coefficients, strong state coupling, and significant uncertainty in aerodynamic models. Classical NDI relies heavily on accurate modeling, which can limit performance under such conditions. INDI, on the other hand, updates the control effectiveness in real time using measured accelerations and angular rates, making it more robust to modeling errors and rapid aerodynamic changes. In the proposed four–loop architecture, INDI is used in the inner angular–rate loop, where fast and reliable response is essential for stabilizing the outer loops. For relationships that are purely kinematics, standard NDI is applied to achieve exact inversion without relying on aerodynamic models.

The cascaded Flight Control System (FCS) is organized into four nested loops, each operating at a distinct dynamic level. The innermost rate loop stabilizes the angular rates, while the successive outer loops regulate attitude, velocity, and finally position. This hierarchical structure, illustrated in Fig. 2, ensures that the fast inner–loop dynamics provide stability for the slower outer–loop objectives, thereby achieving a natural separation of time scales and facilitating controller design. The subscript K is used throughout to denote quantities derived from Earth–relative velocities rather than air–relative ones. This distinction is crucial because the onboard Inertial Navigation System (INS) provides only the Earth–relative velocity vector without direct wind measurements, and thus, it cannot compute the air–relative velocity. Consequently, all kinematic quantities used in the control system are expressed relative to Earth motion.

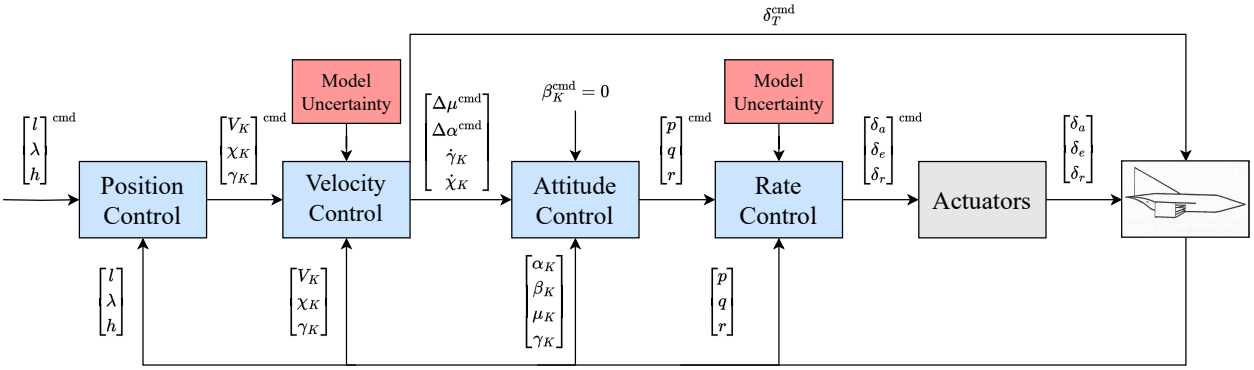

Fig. 2 Cascaded flight control architecture used in this work.

Model Assumptions for Control Law Development

The complete flight dynamics model captures all major physical effects relevant to the vehicle's motion. Such fidelity is essential for accurate simulation and reliable performance evaluation. However, formulating a control law directly from this high-fidelity model would introduce unnecessary complexity, complicating analytical derivations, controller tuning, and practical implementation.

A simplified dynamic model is therefore used for control law development. Only effects whose omission has a negligible impact on the vehicle's response are removed, resulting in a reduced-order model that preserves the dominant control-relevant dynamics. This simplification makes the analytical derivation of the control law feasible while maintaining sufficient accuracy within the intended flight envelope. The final control law is then validated using the complete high-fidelity model to confirm its performance under all relevant physical effects and interactions.

1) *Non-Rotating Earth*
   The angular rate of Earth is assumed to be zero, $\omega_\oplus = 0$. The rotational velocity of the Earth introduces transport and Coriolis accelerations in the equations of motion. For sub-orbital flight in the sensible atmosphere, the transport acceleration is very small compared to aerodynamic and gravitational forces. Its maximum value, occurring at the equator, is about 0.034 m/s^2 ($3.5 \times 10^{-3}\, g_0$), and it decreases with latitude to zero at the poles. The Coriolis acceleration magnitude is bounded by $2\omega_\oplus V$; at $V = 300$ m/s it is 0.044 m/s^2 $\approx 4.5 \times 10^{-3}\, g_0$, at $V = 1500$ m/s it is 0.22 m/s^2 $\approx 2.2 \times 10^{-2}\, g_0$, and it only becomes significant ($\approx 0.116\, g_0$) at orbital velocities. For the control loops designed here, these values are smaller than typical modeling and actuation uncertainties, so omitting them simplifies the derivation without altering closed-loop behavior.

2) *Spherical Earth Model*
   The high-fidelity model represents the Earth as an ellipsoid, but the reduced-order control model assumes a spherical Earth, because the difference between an ellipsoidal and spherical representation is negligible for the purposes of controller design. This simplification removes the distinction between geodetic and geocentric latitude and makes the Earth's radius constant rather than latitude dependent. Quantitatively, the WGS–84 equatorial and polar radii differ by approximately 21.4 km. At a flight altitude of 91 km, the equatorial and polar distances from Earth's center are 6469.137 km and 6447.752 km respectively, based on the WGS–84 radii. The

corresponding gravitational accelerations are $g_{\text{eq}}(91\text{ km}) \approx 9.516\text{ m s}^{-2}$ and $g_{\text{pol}}(91\text{ km}) \approx 9.547\text{ m s}^{-2}$, so the maximum latitude-dependent difference $\Delta g \approx 0.031\text{ m s}^{-2}$ corresponds to roughly 0.33%. This variation is small and has negligible influence on the closed-loop dynamics considered here.

A further simplification to a flat Earth would eliminate the curvature terms from the equations of motion. In particular, omitting the term $V^2/(gR)$ removes the centrifugal contribution that reduces the net force which the aerodynamic lift must balance. As the vehicle's speed approaches the orbital condition $V^2 = gR$, this centrifugal contribution becomes large enough to offset gravity, so the aerodynamic lift required to maintain altitude decreases toward zero. Because induced drag depends directly on lift, it also decreases toward zero in this regime. This reduction is a real physical effect that influences high-speed aerodynamic behavior. For this reason, the flat-Earth assumption is not adopted and the reduced-order control model retains the curvature terms.

3) *Central Gravity Field*
In the reduced-order control model, the gravitational field is taken to be purely central, meaning that zonal harmonics such as J_2 are omitted. Combined with the spherical Earth assumption, which already removes geometric latitude dependence, the only remaining variation in g is its inverse-square dependence on the distance from Earth's center.

Assuming constant gravity introduces a larger error when the $1/r^2$ decrease with altitude is ignored: evaluating $g(h) = g_0(R_0/(R_0 + h))^2$ at $h = 91$ km yields $g(91\text{ km}) \approx 9.530\text{ m s}^{-2}$, so the constant-gravity approximation overestimates gravitational acceleration by about 0.277 m s^{-2}, or approximately 2.83%. Although this altitude-induced variation is comparable to the error introduced by neglecting Coriolis effects at 1500 m/s, it is retained because it is straightforward to include and does not introduce any additional complexity into the reduced-order model.

4) *Stationary Atmosphere Model*
The reduced-order control model assumes steady atmospheric conditions and ignores changes in wind over time or location. At hypersonic speeds, the contribution of wind to the total airspeed is negligible compared to the vehicle's forward velocity, and its rotational influence on the body dynamics is effectively zero [8]. At subsonic speeds, [9] have shown that INDI exhibits excellent disturbance rejection properties against constant wind and wind gusts, further reducing the need to model wind explicitly during control synthesis. Moreover, the onboard INS has no direct access to wind information, making it unrealistic to incorporate wind into the controller as a measured quantity. It is therefore more appropriate to treat wind as a disturbance that must be rejected rather than as a modeled input. This assumption simplifies the controller and allows the design process to focus on the core vehicle dynamics. Wind and turbulence effects are, however, retained in the high-fidelity model to quantify performance in realistic atmospheric conditions.

5) *Coordinated Flight and Constant Speed*
The attitude control loop regulates the sideslip angle β_K to zero, so both the velocity and position control loops are formulated under the assumption $\beta_K = 0$. Similarly, the velocity control loop ensures that the commanded speed is reached and maintained, allowing the position control loop to be formulated under the additional assumption $V_K = V_K^{\text{cmd}}$. Together, these assumptions reduce the number of states that must be considered in the outer loops and provide the additional benefit of filtering out small-scale fluctuations in the computed velocity and sideslip angle, since both quantities are treated as constant.

6) *Small-Displacements*
The change in longitude l and latitude λ is negligible over the short time scales relevant to the velocity dynamics. As a result, $\dot{l}$ and $\dot{\lambda}$ are omitted from the attitude control loop design [10]. Numerical evaluation confirms that their effect on the velocity response is insignificant, with differences in the computed states below 10^{-6}.

7) *Accurate Control Effectiveness and Actuator Dynamics*
Throughout the control-law derivation, it is assumed that the estimated control effectiveness matrix matches the true system behavior and that the modeled actuator dynamics accurately represent the physical actuator response.

These assumptions ensure that the nonlinear inversion remains valid and that the incremental control updates behave as predicted by the reduced-order model.

IV. Control Law Design

This section presents the design of the dynamic inversion control laws for each loop in the cascaded architecture. It begins with the angular–rate control loop, where both NDI and INDI formulations are developed, including the method for reconstructing previous control inputs and the selection criteria for the low–pass filter used in the INDI implementation. The next part addresses the attitude–control loop, describing the derivation of the nonlinear inversion law used to track commanded attitudes. The velocity–control loop is then introduced, outlining its inversion law and its role in regulating airspeed and flight–path dynamics. Finally, the position–control loop is discussed, including its two functional modes for altitude and heading tracking.

A. Position Control Loop

The outermost loop governs the vehicle's position by generating reference commands for velocity, heading, and flight–path angle, which are tracked by the velocity–control loop. Its purpose is twofold. First, it enables trajectory tracking, allowing the vehicle to follow prescribed references in longitude, latitude, and altitude. Second, it enables test–case generation, where the performance of heading and altitude tracking is evaluated. The system is formulated using the state vector, output vector, and control input shown below, where λ denotes the vehicle's latitude, l its longitude, and h its altitude above a spherical Earth[†]:

$$\sum \quad \begin{cases} \boldsymbol{x}_4 = [\lambda \;\; l \;\; h]^\top, \\ \boldsymbol{y}_4 = \boldsymbol{h}_4(\boldsymbol{x}_4) = \boldsymbol{x}_4, \\ \boldsymbol{u}_4 = [u_N \;\; u_E \;\; u_U]^\top. \end{cases} \tag{11}$$

In this context the heading angle χ_K is defined in the standard navigation convention: $\chi_K = 0^\circ$ corresponds to motion due north, $\chi_K = 90^\circ$ to motion due east and so forth. A positive latitude rate corresponds to northward motion, a positive longitude rate corresponds to eastward motion, and a positive altitude rate corresponds to an increase in altitude.

With this convention, the translational kinematics for a point–mass model on a spherical Earth are:

$$\dot{\lambda} = \frac{V_K^{\text{ref}} \cos\chi_K \cos\gamma_K}{R_0 + h}, \quad \dot{l} = \frac{V_K^{\text{ref}} \sin\chi_K \cos\gamma_K}{(R_0 + h)\cos\lambda}, \quad \dot{h} = V_K^{\text{ref}} \sin\gamma_K, \tag{12}$$

with R_0 the mean Earth radius, V_K^{ref} the commanded geographic speed, χ_K heading, and γ_K the flight–path angle. Due to the constant speed assumption V_K^{ref} is treated as constant.

For nonlinear dynamic inversion it is convenient to define an intermediate input vector:

$$\boldsymbol{u} = \begin{bmatrix} u_N \\ u_E \\ u_U \end{bmatrix} = \begin{bmatrix} V_K^{\text{ref}} \cos\chi_K \cos\gamma_K \\ V_K^{\text{ref}} \sin\chi_K \cos\gamma_K \\ V_K^{\text{ref}} \sin\gamma_K \end{bmatrix}, \tag{13}$$

where u_N, u_E, and u_U denote the linear velocity components of the vehicle in the geographic North–East-Up coordinate system. These components serve as intermediate control variables: when mapped through the spherical–dependent matrix $\boldsymbol{G}_4(\boldsymbol{x})$, they yield the correct time derivatives of (λ, l, h). The dynamics of the output vector can then be expressed in control–affine form as:

$$\dot{\boldsymbol{y}}_4 = \boldsymbol{f}_4(\boldsymbol{x}) + \boldsymbol{G}_4(\boldsymbol{x})\,\boldsymbol{u}_4. \tag{14}$$

From the control–affine representation the output dynamics can be written explicitly as:

$$\begin{bmatrix} \dot{\lambda} \\ \dot{l} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{R_0 + h} & 0 & 0 \\ 0 & \dfrac{1}{(R_0 + h)\cos\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_N \\ u_E \\ u_U \end{bmatrix}. \tag{15}$$

[†]For a spherical Earth the distinction between geocentric and geodetic disappears, so that $\lambda = \lambda_c = \lambda_d$ and $h = h_c = h_d$.

In nonlinear dynamic inversion, the control objective is expressed in terms of virtual controls, defined as the desired geographic rates $\boldsymbol{\nu}_4 = [\dot{\lambda} \;\; \dot{l} \;\; \dot{h}]^\top_{\text{cmd}}$ which are obtained from the reference trajectory $\boldsymbol{y}_4^{\text{cmd}} = [\lambda \;\; l \;\; h]^\top_{\text{cmd}}$ by suitable tracking laws. Enforcing $\dot{\boldsymbol{y}}_4 = \boldsymbol{\nu}_4$ gives:

$$\boldsymbol{\nu}_4 = \boldsymbol{G}_4(\boldsymbol{x})\, \boldsymbol{u}_4, \tag{16}$$

so that the required linear velocities are:

$$\boldsymbol{u}_4^{\text{cmd}} = \boldsymbol{G}_4^{-1}(\boldsymbol{x})\, \boldsymbol{\nu}_4, \tag{17}$$

which yields the explicit inversion law:

$$\begin{bmatrix} u_N \\ u_E \\ u_U \end{bmatrix}_{\text{cmd}} = \begin{bmatrix} (R_0 + h) & 0 & 0 \\ 0 & (R_0 + h)\cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \nu_\lambda \\ \nu_l \\ \nu_h \end{bmatrix}, \tag{18}$$

where ν_λ, ν_l, and ν_h denote the virtual control inputs for the longitudinal, latitudinal, and vertical channels, respectively. The commanded flight–path and heading angles are obtained directly by inverting the kinematic relations:

$$\gamma_K^{\text{cmd}} = \arcsin\left(\frac{u_U}{V_K^{\text{cmd}}}\right), \qquad \chi_K^{\text{cmd}} = \text{atan2}\left(u_N, u_E\right). \tag{19}$$

These angles, together with the reference speed, form the commanded inputs to the velocity control loop.

For control–system testing, however, the objective is not to follow arbitrary reference trajectories, but to assess how well the system can maintain a prescribed heading or climb at a fixed climb rate. Therefore, two additional test modes are implemented to facilitate these specific scenarios.

1) **Mode 1: Constant–rate climb to altitude:**
   In this mode, the vehicle is commanded to reach a target altitude h^{ref} while maintaining a specified vertical rate $\dot{h}^{\text{ref}}$. The up–velocity component is therefore imposed directly as:

   $$u_U^{\text{cmd}} = \dot{h}^{\text{ref}}. \tag{20}$$

   This value is substituted into Eq. (19) to compute the commanded flight–path angle γ_K^{cmd}. Once the altitude error is eliminated (i.e. $h = h^{\text{ref}}$), the climb rate command is set to zero. This mode can be combined with Mode 2.
2) **Mode 2: Heading–hold:**
   In this mode, when heading–hold is enabled (e.g. `mheading = 1`), any heading value calculated from Eq. (19) is overridden and the commanded heading is specified directly by the reference heading:

   $$\chi_K^{\text{cmd}} = \chi_K^{\text{ref}}, \tag{21}$$

   while the commanded flight–path angle γ_K^{cmd} is still obtained from Eq. (19) to ensure that the altitude h tracks its reference h^{cmd}.

This completes the design of the position–control loop. Its feasibility depends on several conditions that ensure the inversion law is well defined and that the commanded angles remain physically meaningful. The input–gain matrix $\boldsymbol{G}_4(\boldsymbol{x})$ must be invertible, which requires $\cos\lambda \neq 0$ and therefore excludes the poles where the geographic formulation is ill defined. The inner velocity loop must accurately track the commanded reference speed, and the north and east velocity components (u_N, u_E) cannot both be zero, because the commanded heading χ_K^{cmd} is otherwise undefined.

B. Velocity Control Loop

The velocity control loop governs the translational dynamics of the vehicle and is responsible for tracking the flight-path command $\boldsymbol{u}_4^{\text{cmd}}$. It generates the throttle, δ_T, bank-angle, μ_K, and angle-of-attack, α_K, commands $\boldsymbol{u}_3^{\text{cmd}}$ for the inner attitude control loop. As in the previous section, the controller has access only to the kinematic, groundspeed-relative values α_K and μ_K provided by the INS, which means that the on-board model incurs an error when wind is present.

In contrast to the attitude and position loop, where the dynamics are purely kinematic, the translational dynamics depend directly on uncertain aerodynamic forces. This makes INDI particularly suitable for this loop. Its disturbance–rejection mechanism allows the controller to compensate for modeling errors and neglected effects, which appear

as disturbances on the slower time scale of the velocity dynamics [11]. The system is defined using the following state, output, and input variables:

$$\sum \begin{cases} \boldsymbol{x}_3 = [V_K \;\; \chi_K \;\; \gamma_K]^\top, \\ \boldsymbol{y}_3 = \boldsymbol{h}_3(\boldsymbol{x}_3) = \boldsymbol{x}_3, \\ \boldsymbol{u}_3 = [\delta_T \;\; \mu_K \;\; \alpha_K]^\top. \end{cases} \tag{22}$$

Here, V_K denotes the groundspeed. The angle of attack α_K is positive for nose–up motion relative to the groundspeed velocity vector, and the bank angle μ_K is positive for a right–wing–down rotation about the groundspeed velocity vector. The throttle command δ_T denotes the throttle setting and takes values in the interval $[0.05,\, 2]$. The governing dynamics are formulated by expressing Newton's Second Law in the air-path coordinate system. The resulting time derivatives of the velocity states are given by [12]:

$$\dot{\boldsymbol{x}}_3 = \boldsymbol{M}^{-1}\left([T]^{VB}[f_{a,p}]^B + [T]^{VG}[f_g]^G + m\boldsymbol{a}_{\text{corr}}\right), \text{ where } \boldsymbol{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & mV_K\cos\gamma_K & 0 \\ 0 & 0 & -mV_K \end{bmatrix}. \tag{23}$$

Here, $[f]^B$ is the aero-propulsive force vector expressed in body coordinates, $[f_g]^G$ is the gravitational force vector in expressed in geocentric coordinates, $[T]^{VG}$ and $[T]^{VB}$ are transformation matrices from geocentric and body coordinates to the velocity coordinates, which are given by Eq. (26) and $\boldsymbol{a}_{\text{corr}}$ contains rotation and curvature corrections [12]. The spherical Earth assumption implies that the Earth radius function $R_0(\lambda_d)$ reduces to the constant mean spherical Earth radius R_0. In this case, the corrections are:

$$\boldsymbol{a}_{\text{corr}} = \begin{bmatrix} \omega_\oplus^2 R_0\cos\lambda\,(\sin\gamma_K\cos\lambda - \cos\gamma_K\sin\lambda\cos\chi_K) \\ 2\omega_\oplus V_K(\sin\lambda\cos\gamma_K - \cos\lambda\sin\gamma_K\cos\chi_K) + \dfrac{V_K^2}{R_0}\cos^2\gamma_K\tan\lambda\sin\chi_K + \omega_\oplus^2 R_0\cos\lambda\sin\lambda\sin\chi_K \\ -2\omega_\oplus V_K\cos\lambda\sin\chi_K - \dfrac{V_K^2}{R_0}\cos\gamma_K - \omega_\oplus^2 R_0\cos\lambda(\cos\lambda\cos\gamma_K + \sin\gamma_K\sin\lambda\cos\chi_K) \end{bmatrix}. \tag{24}$$

With the non-rotating Earth assumption, all terms containing $\omega_\oplus$ vanish. Together, this yields the simplified spherical-Earth form of the velocity dynamics:

$$\begin{bmatrix} \dot{V}_K \\ \dot{\chi}_K \\ \dot{\gamma}_K \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & mV_K\cos\gamma_K & 0 \\ 0 & 0 & -mV_K \end{bmatrix}^{-1}\left([T]^{VB}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^B + m[T]^{VG}\begin{bmatrix} 0 \\ 0 \\ \frac{GM}{(R_0+h)^2} \end{bmatrix}^G + m\begin{bmatrix} 0 \\ \frac{V_K^2}{R_0}\cos^2\gamma_K\tan\lambda\sin\chi_K \\ -\frac{V_K^2}{R_0}\cos\gamma_K \end{bmatrix}\right). \tag{25}$$

In the gravitational force term, GM denotes the Earth's gravitational parameter. Although $[T]^{VB}$ is normally written in its full three–angle form, the condition $\beta_K = 0$ reduces it to the simplified expression shown below. The corresponding transformation matrix $[T]^{VG}$ taken from the same reference is placed alongside it [13].

$$[T]^{VB} = \begin{bmatrix} \cos\alpha_K & 0 & \sin\alpha_K \\ -\sin\alpha_K\sin\mu_K & -\cos\mu_K & \cos\alpha_K\sin\mu_K \\ -\sin\alpha_K\cos\mu_K & \sin\mu_K & \cos\alpha_K\cos\mu_K \end{bmatrix}, \qquad [T]^{VG} = \begin{bmatrix} \cos\gamma\cos\chi & \cos\gamma\sin\chi & -\sin\gamma \\ -\sin\chi & \cos\chi & 0 \\ \sin\gamma\cos\chi & \sin\gamma\sin\chi & \cos\gamma \end{bmatrix}. \tag{26}$$

Next, it is noted that μ_K, α_K, and δ_T do not appear in an affine form in Eq. (25) because the aerodynamic forces contain nonlinear trigonometric couplings with these inputs. The following step is to express the nonlinear dynamics in the standard input-affine form:

$$\dot{\boldsymbol{x}}_3 = \boldsymbol{f}_3(\boldsymbol{x}_3) + \boldsymbol{G}_3(\boldsymbol{x}_3)\,\boldsymbol{u}_3. \tag{27}$$

This can be partially accomplished by separating the force contributions to isolate the effects of bank angle, angle of attack, and thrust setting on the dynamics. The propulsive force is modeled as $f_p = \bar{q}SC_{X_T}\,\delta_T$, and, using the axial force definition in Eq. (4), the body–axis force relation becomes $[X]^B = \bar{q}S\left(C_X + C_{X_T}\delta_T\right)$, showing that thrust enters the dynamics in the same nondimensional manner as the aerodynamic coefficients. Since the velocity dynamics are formulated in the air–path coordinate system, the body–axis force vector is first transformed using $[T]^{VB}$. The resulting

expressions contain products of $\cos\alpha_K$, $\sin\alpha_K$, and the body–axis coefficients C_X and C_Z. To simplify these terms, the standard relations between body–axis and stability–axis coefficients are used:

$$C_D = -\left(\cos\alpha_K\, C_X + \sin\alpha_K\, C_Z\right), \qquad C_L = \sin\alpha_K\, C_X - \cos\alpha_K\, C_Z,$$

allowing the trigonometric combinations in the velocity–axis forces to be replaced by the physically meaningful drag and lift coefficients:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^V = [T]^{VB} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^B = \bar{q}S \begin{bmatrix} \cos\alpha_K\, C_{X_T}\delta_T - C_D \\ -\sin\mu_K\left(C_{X_T}\delta_T \sin\alpha_K + C_L\right) - \cos\mu_K\, C_Y \\ -\cos\mu_K\left(C_{X_T}\delta_T \sin\alpha_K + C_L\right) + \sin\mu_K\, C_Y \end{bmatrix}. \tag{28}$$

Finally, C_L and C_D are decomposed using the definitions in Eq. (1) so that the α_K-dependent terms required by the controller appear explicitly. Substituting the velocity–axis force vector from Eq. (28) into the translational dynamics of Eq. (25), and explicitly evaluating the resulting expressions, gives the nonlinear velocity equations shown in Eq. (29):

$$\begin{aligned}
\dot{V}_K &= \frac{\bar{q}S}{m}\left[C_{X_T}\delta_T\,\cos\alpha_K - C_{D_0} - C_{D_\alpha}\alpha_K\right] - g\sin\gamma_K, \\
\dot{\chi}_K &= \frac{\bar{q}S}{mV_K\cos\gamma_K}\left[-\sin\mu_K\left(C_{X_T}\delta_T\sin\alpha_K + C_{L_0} + C_{L_{\delta_e}}\delta_e + C_{L_\alpha}\alpha_K\right) - C_Y\cos\mu_K\right] + \frac{V_K}{R_0}\cos\gamma_K\tan\lambda\sin\chi_K, \\
\dot{\gamma}_K &= \frac{\bar{q}S}{mV_K}\left[\cos\mu_K\left(C_{X_T}\delta_T\sin\alpha_K + C_{L_0} + C_{L_{\delta_e}}\delta_e + C_{L_\alpha}\alpha_K\right) - C_Y\sin\mu_K\right] + \cos\gamma_K\left(\frac{V_K}{R_0} - \frac{g}{V_K}\right).
\end{aligned} \tag{29}$$

Wind effects are not included but could be incorporated, in which case all subscripts $_K$ change to the wind case $_A$. Looking at Eq. (29), the bank angle μ_K enters the dynamics only through the nonlinear terms $\sin\mu_K$ and $\cos\mu_K$. As a result, the velocity subsystem is not control-affine in μ_K, meaning it cannot be written in the form $f(x) + g_\mu(x)\,\mu_K$. This violates the standard requirement for applying NDI directly with μ_K treated as a control input.

INDI does not have this limitation [14]. By using a first-order Taylor expansion, it can produce an inversion law that works locally around the operating point. A standard Taylor series expansion is taken about the values at the previous sampling instant, where $\boldsymbol{x}_{3,0} = \boldsymbol{x}_3(t-\Delta t)$ and $\boldsymbol{u}_{3,0} = \boldsymbol{u}_3(t-\Delta t)$. This yields:

$$\begin{aligned}
\dot{\boldsymbol{x}}_3 &= \dot{\boldsymbol{x}}_{3,0} + \left.\frac{\partial\dot{\boldsymbol{x}}_3(\boldsymbol{x}_3,\,\boldsymbol{u}_3)}{\partial\boldsymbol{x}_3}\right|_0 \Delta\boldsymbol{x}_3 + \left.\frac{\partial\dot{\boldsymbol{x}}_3(\boldsymbol{x}_3,\,\boldsymbol{u}_3)}{\partial\boldsymbol{u}_3}\right|_0 \Delta\boldsymbol{u}_3 + O(\Delta\boldsymbol{x}_3^2), \\
&\approx \dot{\boldsymbol{x}}_{3,0} + \left.\frac{\partial\dot{\boldsymbol{x}}_3(\boldsymbol{x}_3,\,\boldsymbol{u}_3)}{\partial\boldsymbol{u}_3}\right|_0 \Delta\boldsymbol{u}_3,
\end{aligned} \tag{30}$$

here $\Delta\boldsymbol{x}_3 = \boldsymbol{x}_3 - \boldsymbol{x}_{3,0}$, $\Delta\boldsymbol{u}_3 = \boldsymbol{u}_3 - \boldsymbol{u}_{3,0}$, and the term $O(\Delta\boldsymbol{x}_3^2)$ collects all higher–order contributions.

As in the standard INDI argument, a time–scale separation assumption is introduced: over one sampling step the state increment is much smaller than the control increment, $\Delta\boldsymbol{x}_3 \ll \Delta\boldsymbol{u}_3$. Attitude dynamics typically evolve on a much faster time scale than the flight path states, so the term involving $\Delta\boldsymbol{x}_3$ and the higher–order contributions can be neglected [15]. From Eq. (29) and by seeing that $C_L = C_{L_0} + C_{L_{\delta_e}}\delta_e + C_{L_\alpha}\alpha_K$, the control effectiveness matrix is found by taking the partial derivatives of each state rate $\dot{\boldsymbol{x}}_3$ with respect to each control input $\boldsymbol{u}_3$, which yields:

$$\frac{\partial\dot{\boldsymbol{x}}_3}{\partial\boldsymbol{u}_3} = \frac{\bar{q}S}{mV_K}\begin{bmatrix} C_{X_T}\cos\alpha_K V_K & 0 & -V_K(C_{X_T}\delta_T\sin\alpha_K + C_{D_\alpha}) \\ \dfrac{-C_{X_T}\sin\mu_K\sin\alpha_K}{\cos\gamma_K} & \dfrac{-\cos\mu_K(C_L + C_{X_T}\delta_T\sin\alpha_K) + C_Y\sin\mu_K}{\cos\gamma_K} & \dfrac{-\sin\mu_K(C_{X_T}\delta_T\cos\alpha_K + C_{L_\alpha})}{\cos\gamma_K} \\ C_{X_T}\cos\mu_K\sin\alpha_K & -\sin\mu_K(C_L + C_{X_T}\delta_T\sin\alpha_K) - C_Y\cos\mu_K & \cos\mu_K(C_{X_T}\delta_T\cos\alpha_K + C_{L_\alpha}) \end{bmatrix}. \tag{31}$$

To evaluate these partial derivatives, the vehicle is assumed to be in vertical force equilibrium, expressed in a form consistent with the spherical Earth model:

$$\bar{q}S\cos\mu_K(C_L + C_{X_T}\delta_T\sin\alpha_K) = m\cos\gamma_K\left(g - \frac{V_K^2}{R_0}\right). \tag{32}$$

This condition makes it possible to replace the aerodynamic term on the left in Eq. (32) by the gravitational–curvature term on the right. Additionally, the influence of thrust on lift and drag derivatives is neglected, as aerodynamic forces dominate in magnitude:

$$C_{L_\alpha}\bar{q}S \gg \bar{q}SC_{X_T}\delta_T \cos\alpha_K, \qquad C_{D_\alpha}\bar{q}S \gg \bar{q}SC_{X_T}\delta_T \sin\alpha_K.$$

The lateral coefficient C_Y is also neglected. With $\beta_K = 0$ its magnitude is much smaller than the lift- and thrust-related terms, making its contribution to the velocity dynamics negligible. Substituting Eq. (32) into Eq. (31) and removing negligible terms yields the following expression for the control effectiveness matrix:

$$\boldsymbol{G}_3(\boldsymbol{x}_3,\, u_3) \approx \frac{\bar{q}S}{mV_K}\begin{bmatrix} C_{X_T}\cos\alpha_K V_K & 0 & -C_{D_\alpha}V_K \\ -\dfrac{C_{X_T}\sin\mu_K\sin\alpha_K}{\cos\gamma_K} & -\dfrac{m}{\bar{q}S}\left(g - \dfrac{V_K^2}{R_0}\right) & -\dfrac{C_{L_\alpha}\sin\mu_K}{\cos\gamma_K} \\ C_{X_T}\cos\mu_K\sin\alpha_K & -\dfrac{m\tan\mu_K\cos\gamma_K}{\bar{q}S}\left(g - \dfrac{V_K^2}{R_0}\right) & C_{L_\alpha}\cos\mu_K \end{bmatrix}. \tag{33}$$

A virtual input $\boldsymbol{\nu}_3 = [\nu_V \;\; \nu_\chi \;\; \nu_\gamma]^\top$ is now defined to represent the desired rates of the velocity states. Linearizing the dynamics around the previously measured operating point yields the incremental inversion law:

$$\boldsymbol{u}_3^{\text{cmd}} = \boldsymbol{u}_{3,0} + \boldsymbol{G}_3^{-1}(\boldsymbol{x}_{3,0},\, \boldsymbol{u}_{3,0}) \;\left(\boldsymbol{\nu}_3 - \dot{\boldsymbol{x}}_{3,0}\right). \tag{34}$$

The subscript "0" refers to measurements from the previous timestep. The matrix $\boldsymbol{G}_3$ becomes non-invertible only in vertical flight ($\gamma_K = \pm 90°$), knife-edge flight ($\mu_K = \pm 90°$), or in the limiting case $g = V_K^2/R_0$, where centrifugal force cancels gravity and lift can no longer steer the velocity vector.

C. Attitude Control Loop

The objective of the attitude loop is to follow the commands $\boldsymbol{u}_3^{\text{cmd}}$ generated by the velocity control loop. It does so by producing the angular-rate commands $\boldsymbol{u}_2^{\text{cmd}}$ for the inner rate loop. The loop regulates the aerodynamic angles μ_K, α_K, and β_K, which are used instead of Euler angles because they are directly provided by the INS and match the commanded quantities from the velocity loop. For the control design, the attitude subsystem is written as:

$$\sum \;\begin{cases} \boldsymbol{x}_2 = [\mu_K \;\; \alpha_K \;\; \beta_K]^\top, \\ \boldsymbol{y}_2 = \boldsymbol{h}_2(\boldsymbol{x}_2) = \boldsymbol{x}_2, \\ \boldsymbol{u}_2 = [p \;\; q \;\; r]^\top. \end{cases} \tag{35}$$

Here, $[p \;\; q \;\; r]$ are the inertial angular rates. For the NDI design, the attitude kinematics are written in the control-affine form:

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2) + \boldsymbol{G}_2(\boldsymbol{x}_2)\,\boldsymbol{u}_2. \tag{36}$$

A set of dynamic attitude equations suitable for expressing this form was derived by Mooij [16]. The general dynamics of the aerodynamic angles for a rotating, geodetic Earth are given by:

$$\begin{aligned} \dot{\mu}_K &= -\dot{\alpha}_K\sin\beta_K + \dot{\chi}_K\sin\gamma_K + \dot{\lambda}\sin\chi_K\cos\gamma_K - (\dot{l}+\omega_\oplus)(\cos\lambda\cos\chi_K\cos\gamma_K + \sin\lambda\sin\gamma_K) \\ &\quad + p\cos\alpha_K\sin\beta_K + q\sin\beta_K + r\sin\alpha_K\cos\beta_K, \\ \dot{\alpha}_K &= -\frac{\sin\mu_K}{\cos\beta_K}\left[\dot{\chi}_K\cos\gamma_K - \dot{\lambda}\sin\chi_K\sin\gamma_K + (\dot{l}+\omega_\oplus)(\cos\lambda\cos\chi_K\sin\gamma_K - \sin\lambda\cos\gamma_K)\right] \\ &\quad - \frac{\cos\mu_K}{\cos\beta_K}\left[\dot{\gamma}_K - \dot{\lambda}\cos\chi_K - (\dot{l}+\omega_\oplus)\cos\lambda\sin\chi_K\right] - p\cos\alpha_K\tan\beta_K + q - r\sin\alpha_K\tan\beta_K, \\ \dot{\beta}_K &= -\sin\mu_K\left[\dot{\gamma}_K - \dot{\lambda}\cos\chi_K - (\dot{l}+\omega_\oplus)\cos\lambda\sin\chi_K\right] + \cos\mu_K\left[\dot{\chi}_K\cos\gamma_K - \dot{\lambda}\sin\chi_K\sin\gamma_K\right] \\ &\quad + \cos\mu_K\left[(\dot{l}+\omega_\oplus)(\cos\lambda\cos\chi_K\sin\gamma_K - \sin\lambda\cos\gamma_K)\right] + p\sin\alpha_K - r\cos\alpha_K, \end{aligned} \tag{37}$$

where l is the longitude, λ is the geodetic latitude, $\dot{l}$ and $\dot{\lambda}$ are their time derivatives, and $\omega_\oplus$ is Earth's angular rate.

The full expression in Eq. (37) contains several terms that have negligible influence on the control law. The body–rate contributions must be retained, and the flight–path rates $[\dot{\gamma}_K \;\; \dot{\chi}_K]$ remain significant for maneuvering. Under the *Small-Displacement Assumption*, variations in l and λ_d and their rates $(\dot{l}, \dot{\lambda}_d)$ are neglected [10]. Consistent with the *Non-rotating Earth Assumption*, the Earth's rotation rate is set to $\omega_\oplus = 0$ for control design.

For dynamic inversion, each aerodynamic–angle derivative must be expressed explicitly in terms of measurable quantities rather than in terms of other angle derivatives. In Mooij's formulation, the equation for $\dot{\mu}_K$ contains the term $\dot{\alpha}_K$, whose value is already given by the second line of Eq. (37). Substituting this expression into the $\dot{\mu}_K$ equation removes the derivative coupling and yields a set of equations in which $\dot{\mu}_K$, $\dot{\alpha}_K$, and $\dot{\beta}_K$ depend only on INS–measurable states, flight–path rates, and body angular rates. Because the resulting dynamics contain only kinematic terms, they admit a direct nonlinear dynamic inversion without the need for an incremental control law. The resulting equations are:

$$\begin{aligned}
\dot{\mu}_K &= \dot{\chi}_K(\sin\gamma_K + \sin\mu_K \tan\beta_K \cos\gamma_K) + \dot{\gamma}_K \cos\mu_K \tan\beta_K + p\frac{\cos\alpha_K}{\cos\beta_K} + r\frac{\sin\alpha_K}{\cos\beta_K},\\
\dot{\alpha}_K &= -\dot{\chi}_K \frac{\sin\mu_K \cos\gamma_K}{\cos\beta_K} - \dot{\gamma}_K \frac{\cos\mu_K}{\cos\beta_K} - p\cos\alpha_K \tan\beta_K + q - r\sin\alpha_K \tan\beta_K,\\
\dot{\beta}_K &= \dot{\chi}_K \cos\mu_K \cos\gamma_K - \dot{\gamma}_K \sin\mu_K + p\sin\alpha_K - r\cos\alpha_K.
\end{aligned} \tag{38}$$

Equation 38 can be separated into the contribution from the body angular rates and the terms arising from the flight–path kinematics which yields the following control-affine form from Eq. 36:

$$\begin{bmatrix}\dot{\mu}_K\\ \dot{\alpha}_K\\ \dot{\beta}_K\end{bmatrix} = \underbrace{\begin{bmatrix}\sin\gamma_K + \sin\mu_K\tan\beta_K\cos\gamma_K & \cos\mu_K\tan\beta_K\\ -\frac{\sin\mu_K\cos\gamma_K}{\cos\beta_K} & -\frac{\cos\mu_K}{\cos\beta_K}\\ \cos\mu_K\cos\gamma_K & -\sin\mu_K\end{bmatrix}\begin{bmatrix}\dot{\chi}_K\\ \dot{\gamma}_K\end{bmatrix}}_{\boldsymbol{f}_2(\boldsymbol{x}_2)} + \underbrace{\begin{bmatrix}\frac{\cos\alpha_K}{\cos\beta_K} & 0 & \frac{\sin\alpha_K}{\cos\beta_K}\\ -\cos\alpha_K\tan\beta_K & 1 & -\sin\alpha_K\tan\beta_K\\ \sin\alpha_K & 0 & -\cos\alpha_K\end{bmatrix}}_{\boldsymbol{G}_2(\boldsymbol{x}_2)}\underbrace{\begin{bmatrix}p\\ q\\ r\end{bmatrix}}_{\boldsymbol{u}_2}. \tag{39}$$

The influence of $\dot{\gamma}_K$ and $\dot{\chi}_K$ can alternatively be expressed using the specific forces measured by the onboard accelerometers, from which the contributions of weight, lift, and side force to the flight–path angles are reconstructed [17].

A virtual input $\boldsymbol{\nu}_2 = [\nu_\mu \;\; \nu_\alpha \;\; \nu_\beta]^\top$ is defined to represent the desired attitude angle rates. By enforcing $\dot{\boldsymbol{x}}_2 = \boldsymbol{\nu}_2$ and inverting Eq. (36), the required body rates become:

$$\boldsymbol{u}_2^{\text{cmd}} = \boldsymbol{G}_2^{-1}(\boldsymbol{x}_2)\left(\boldsymbol{\nu}_2 - \boldsymbol{f}_2(\boldsymbol{x}_2)\right). \tag{40}$$

The matrix $\boldsymbol{G}_2$ is invertible for all α_K provided $\cos\beta_K \neq 0$, i.e. $\beta_K \neq \frac{\pi}{2} + n\pi,\ n \in \mathbb{Z}$. This inversion yields the commanded body rates required to track the desired changes in attitude angles.

D. Angular Rate Control Loop

The angular rate loop forms the innermost part of the FCS. It is responsible for tracking angular rate commands $\boldsymbol{u}_2^{\text{cmd}}$ generated by the attitude controller. It calculates the actuator commands The system is described using the following state, output, and input vectors:

$$\sum \begin{cases}\boldsymbol{x}_1 = [p \;\; q \;\; r]^\top,\\ \boldsymbol{y}_1 = \boldsymbol{h}_1(\boldsymbol{x}_1) = \boldsymbol{x}_1,\\ \boldsymbol{u}_1 = [\delta_a \;\; \delta_e \;\; \delta_r]^\top,\end{cases} \tag{41}$$

where p, q, r are the inertial body rates, and the control inputs $\delta_a, \delta_e, \delta_r$ are the aileron, elevator, and rudder deflections, respectively. The variable–mass attitude equations of motion for a symmetric aircraft, written in the body coordinate system with superscripts omitted for clarity, are:

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{I}^{-1}\left(\boldsymbol{m}_B - \dot{\boldsymbol{I}}\boldsymbol{x}_1 - \boldsymbol{x}_1 \times \boldsymbol{I}\,\boldsymbol{x}_1\right). \tag{42}$$

where I are the vehicle's moment of inertia and m_B are the external moments from Eq. (5). These moments are the sum of the aerodynamic moments $\boldsymbol{m}_a$ and the control–generated moments $\boldsymbol{m}_c$. Assuming linear control derivatives $(\boldsymbol{m}_c)_\delta = \frac{\partial}{\partial\delta}\boldsymbol{m}_c$, the total moment becomes linear in the deflection inputs:

$$\boldsymbol{m}_B = \boldsymbol{m}_a + (\boldsymbol{m}_c)_\delta\,\boldsymbol{u}_1. \tag{43}$$

Substituting Eq. (43) into the rotational equation of motion and solving for the input-affine form yields:

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{I}^{-1}\left[-\dot{\boldsymbol{I}}\boldsymbol{x}_1 - \boldsymbol{x}_1 \times I\,\boldsymbol{x}_1 + \boldsymbol{m}_a\right] + \boldsymbol{I}^{-1}(\boldsymbol{m}_c)_\delta\,\boldsymbol{u}_1. \tag{44}$$

The control effectiveness matrix and plant dynamics are consequently:

$$\boldsymbol{G}_1 = \boldsymbol{I}^{-1}(\boldsymbol{m}_c)_\delta, \quad \text{and} \quad \boldsymbol{f}_1(\boldsymbol{x}_1) = \boldsymbol{I}^{-1}\left(-\dot{\boldsymbol{I}}\,\boldsymbol{x}_1 - \boldsymbol{x}_1 \times I\,\boldsymbol{x}_1 + \boldsymbol{m}_a\right). \tag{45}$$

To reduce the dependency of the control law on the full nonlinear model, the angular–rate loop also employs an INDI formulation. The key idea here is to replace part of the model information by locally measured state–rate data, thereby improving robustness to modeling uncertainty. Take the local Taylor series expansion on the control-affine Eq. (44) and remove state dependent contributions by using the time scale separation assumption. This reduces Eq. (44) to the incremental approximation:

$$\dot{\boldsymbol{x}}_1 \approx \dot{\boldsymbol{x}}_{1,0} + \boldsymbol{G}_1\,\Delta\boldsymbol{u}_1. \tag{46}$$

The matrix $\boldsymbol{G}_1$ is treated as constant over the interval and is obtained from the OBM, which updates it online. To enforce desired angular acceleration dynamics, a virtual control input $\boldsymbol{\nu}_1$ is introduced. Substituting this into the approximation Eq. (46) and solving for $\boldsymbol{u}_1$ yields the INDI control law:

$$\boldsymbol{u}_1^{\text{cmd}} = \boldsymbol{u}_{1,0} + \boldsymbol{G}_1^{-1}\left(\boldsymbol{\nu}_1 - \dot{\boldsymbol{x}}_{1,0}\right). \tag{47}$$

This control law preserves the input-affine structure of feedback linearization while avoiding explicit dependence on the full nonlinear model. Provided that measurements of $\dot{\boldsymbol{x}}_1(t-\Delta t)$ and $\boldsymbol{u}_1(t-\Delta t)$ are accurate and the delay Δt is sufficiently small, the INDI formulation achieves robust inner-loop angular rate tracking. All quantities are expressed in the body-axis coordinate system:

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix}, \quad (\boldsymbol{m}_c)_\delta = \bar{q}\,S \begin{bmatrix} b\,C_{l_{\delta_a}} & 0 & b\,C_{l_{\delta_r}} \\ 0 & \bar{c}\,C_{m_{\delta_e}} & 0 \\ b\,C_{n_{\delta_a}} & 0 & b\,C_{n_{\delta_r}} \end{bmatrix}, \quad \boldsymbol{m}_a = \bar{q}\,S \begin{bmatrix} b\,C_{l_a} \\ \bar{c}\,C_{m_a} \\ b\,C_{n_a} \end{bmatrix}.$$

E. State Estimation, Filtering, and Synchronization

In the SISO implementation, both the delayed control input δ_0 and the delayed angular acceleration $\dot{\omega}_0$ must be reconstructed, since neither quantity is directly measurable. The delayed actuator deflection is obtained by passing the commanded signal $\delta_{\text{cmd}}(t)$ through the second–order actuator model in Eq. (7) together with the fixed delay Δt:

$$\delta_0 = \delta(t-\Delta t) = \mathcal{L}^{-1}\left\{e^{-\Delta t s} A(s)\,\delta_{\text{cmd}}(s)\right\}. \tag{48}$$

The same delay appears in the angular–acceleration term because the INDI control law is linearized about a previously measured operating point. In the Discrete Time (DT) implementation, this delay coincides with the Flight Control Computer (FCC) sampling time and is represented by a single Unit Delay (UD) block. The delayed angular acceleration follows from differentiation of the measured angular rate:

$$\dot{\omega}_0 = \dot{\omega}(t-\Delta t) = \mathcal{L}^{-1}\left\{s\,e^{-\Delta t s}\,\omega(s)\right\}. \tag{49}$$

Direct numerical differentiation amplifies sensor noise, so the angular–rate measurement is first passed through the second–order low–pass filter:

$$H(s) = \frac{\omega_H^2}{s^2 + 2\zeta_H\omega_H s + \omega_H^2}, \qquad \text{using} \quad \omega_H = 25\,\text{rad/s},\ \zeta_H = 1, \tag{50}$$

This filter produces the angular–acceleration estimate:

$$\dot{\omega}_{f_0} = \mathcal{L}^{-1}\left\{s\,H(s)\,e^{-\Delta t s}\,\omega(s)\right\}. \tag{51}$$

The filter introduces a phase lag that would desynchronize $\dot{\omega}_{f_0}$ from the unfiltered actuator signal δ_0. Because INDI relies on a first–order Taylor expansion evaluated at a common delayed operating point, both paths must experience the same temporal shift. The actuator path is therefore filtered identically:

$$\delta_{f_0} = \mathcal{L}^{-1}\left\{H(s)\,e^{-\Delta t s}\,A(s)\,\delta_{\text{cmd}}(s)\right\}. \tag{52}$$

With the two signals synchronized, the filtered INDI control law becomes:

$$\delta_{\text{cmd}} = \delta_{f_0} + G_{1,0}^{-1}\left(\nu_1 - \dot{\omega}_{f_0}\right). \tag{53}$$

The complete structure, including the filtering and reconstruction paths, is illustrated in Fig. 3. The diagram summarizes the roles of the actuator dynamics $A(s)$, the noise–attenuating filter $H(s)$, and the fixed delay Δt within the INDI formulation.

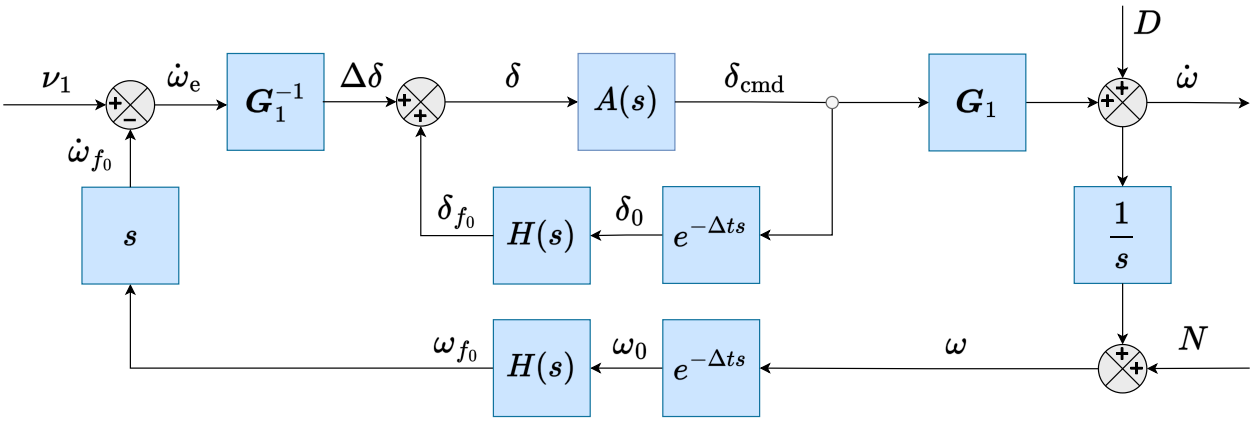

Fig. 3 Frequency-domain implementation of the INDI control loop for angular rate tracking. The loop includes angular acceleration estimation, actuator modeling, and consistent filtering of both the feedback and control branches.

V. Linear Controller Design

Nonlinear dynamic inversion reshapes the system into approximate chains of integrators, forming the basis for hierarchical feedback design. A time–scale separation assumption is adopted so that the faster inner loops appear instantaneous from the perspective of the slower outer loops, allowing each loop to be designed independently.

With the dynamics rendered integrator–like and the hierarchy established, the remaining task is to specify the desired transient and steady–state characteristics through linear controllers placed on top of the inverted dynamics. The hierarchical control architecture follows a set of design requirements:

1) **Damping requirements:** The rate loop is designed to be critically damped ($\zeta_\omega = 1$) with no overshoot in the angular rate. The outer loops are designed to be well damped, with

$$\zeta_\theta = 0.9, \qquad \zeta_V = 0.7, \qquad \zeta_X = 0.9,$$

ensuring fast responses with minimal overshoot.
2) **Rate Requirement:** The rate loop bandwidth BW_ω is selected as high as possible subject to the critical–damping requirement, as a faster inner loop eases the design constraints imposed on the slower outer loops.
3) **Time–scale separation:** Each loop is designed to be significantly slower than the loop beneath it. A practical and commonly used choice is a bandwidth ratio of approximately four between adjacent loops, leading to:

$$\text{BW}_\theta \leq \frac{\text{BW}_\omega}{4}, \qquad \text{BW}_V \leq \frac{\text{BW}_\theta}{4}, \qquad \text{BW}_X \leq \frac{\text{BW}_V}{4}.$$

4) **Robustness margins:** All loops must achieve a phase margin of at least 30° and a gain margin of at least 6 dB.

Rate Loop

The design of the angular–rate controller follows from the observation that, under ideal INDI and in the absence of disturbances and sensor noise, the airframe dynamics are cancelled by the inversion, leaving only the actuator dynamics

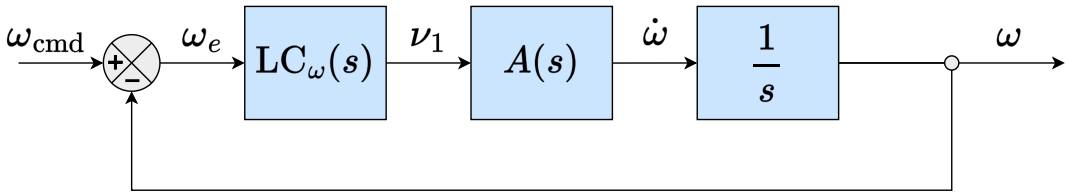

Fig. 4 Idealized closed–loop structure of the angular–rate control loop.

$A(s)$ in the loop, and that rapid disturbance rejection together with effective noise suppression can be ensured by a suitable choice of the noise filter $H(s)$ [18].

Figure 4 illustrates the corresponding idealized closed–loop structure of the angular–rate controller, where the linear controller $LC_\omega(s)$ generates the virtual input ν_1 from the rate error, and the integrator yields the angular rate $\boldsymbol{\omega}$. Thus, in this simplified setting the closed–loop $\boldsymbol{\omega}_{\text{cmd}}(s)$ to $\boldsymbol{\omega}(s)$ is:

$$H_\omega(s) = \frac{A(s)LC_\omega(s)}{s + A(s)LC_\omega(s)}. \tag{54}$$

The actuator dynamics are often assumed to be sufficiently fast to justify the approximation $A(s) \approx 1$ over the relevant frequency range [18]. This assumption holds for systems with very high actuator bandwidths, but becomes inaccurate when the actuator bandwidth is more moderate. In such cases, neglecting $A(s)$ leads to optimistic predictions of loop performance, since the actuator still limits stability and achievable bandwidth. For the present aircraft, the actuator bandwidth of approximately 50 rad/s is not high enough for $A(s) \approx 1$ to be a valid approximation within the desired control bandwidth. A substantially lower rate-loop bandwidth would be required to make actuator dynamics negligible, which is undesirable for performance. Therefore, the actuator model $A(s)$ is included explicitly in both the controller design and the stability analysis, ensuring that achievable bandwidth and robustness are evaluated with the true actuator limitations in mind.

The controller is designed for the third-order plant $P(s) = A(s)\,\frac{1}{s}$, where the actuator dynamics $A(s)$ from Eq. (7) act in series with the integrator, as shown in Fig. 4. For this plant, a proportional control strategy is adopted, using the control law:

$$LC_\omega(s) = K_p.$$

From the requirements it follows that the proportional gain K_p must be chosen to maximize the closed–loop bandwidth while ensuring negligible overshoot and sufficient robustness. To this end, a bisection search is employed to determine the largest feasible gain subject to the constraints of at most 0.1% overshoot and a minimum phase margin of 30°. The search begins with a broad interval $[K_{p,\min}, K_{p,\max}]$. At each iteration, the midpoint value of K_p is evaluated by forming the closed-loop transfer function:

$$T(s) = \frac{K_p P(s)}{1 + K_p P(s)}. \tag{55}$$

From this closed–loop system, the overshoot is extracted from the time–domain step response, while the phase margin is obtained from the frequency response of the open–loop transfer function $L(s) = K_p P(s)$. If the constraints are satisfied, the gain is deemed feasible and the lower bound of the search interval is raised, otherwise, the upper bound is reduced. This procedure continues until the interval width falls below the prescribed tolerance ε, guaranteeing convergence to the largest feasible proportional gain. The result is expressed as:

$$K_p^\star = \max\left\{K_p \;\middle|\; \text{OS} \le 0.1\%,\ \text{PM} \ge 30^\circ\right\}, \tag{56}$$

which yields the fastest critically damped response consistent with the actuator dynamics described in Eq. (7). The bisection search is characterized by the iteration bound:

$$N = \left\lceil \log_2\left(\frac{K_{p,\max} - K_{p,\min}}{\varepsilon}\right)\right\rceil, \tag{57}$$

where $K_{p,\min}$ and $K_{p,\max}$ represent the initial search interval. With $K_{p,\min} = 10^{-3}$, $K_{p,\max} = 10^{2}$, and a convergence tolerance of $\varepsilon = 10^{-6}$, the method is guaranteed to converge, if a solution exists, within $N = 27$ iterations. For the plant under consideration, the optimized gain and maximum achievable bandwidth and phase margin are:

$$LC_\omega(s) = 13.5625, \qquad \mathrm{BW}_\omega = 24.96 \text{ rad/s} \qquad \mathrm{PM} = 67.57 \text{ deg.} \tag{58}$$

This yields the following closed loop relation:

$$H_\omega(s) = \frac{33906}{(s + 24.96)(s^2 + 45.74s + 1358)}. \tag{59}$$

Allowing a small overshoot would enable a higher crossover frequency and bandwidth at the expense of reduced phase margin[‡]. It should also be noted that digital implementation effects, such as sampling and computation delays, introduce extra phase lag. For this reason, retaining sufficient phase and gain margins is essential to ensure robustness against these delays.

Table 1 Pole locations, natural frequencies, and damping ratios of the angular rate dynamics

Real ± Imag	ω_n, rad/s	ζ	Type
$-22.87 \pm 28.90i$	36.85	0.62	Actuator poles
$-24.96 \pm 0.00i$	24.96	1.00	Rate pole

Attitude Loop

For the linear controller in the attitude loop, the first step is to obtain the closed–loop expression of the system, shown schematically in Fig. 5. Here, standard NDI is used rather than the incremental form. Unlike INDI, where additional dynamics appeared as disturbances to reject, the NDI formulation retains these explicitly. In this case it is the term $\boldsymbol{f}_2$, representing the flight–path dynamics given in Eq. (39):

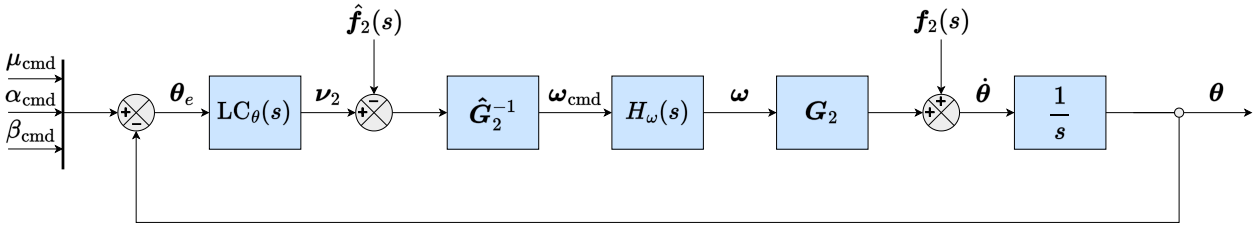

Fig. 5 Closed-loop architecture for attitude control using model-based inversion and linear compensation.

The commands for μ_{cmd} and α_{cmd} are supplied by the outer velocity loop, while the sideslip reference is fixed at $\beta_{\mathrm{cmd}} = 0$ to enforce coordinated flight. A linear controller $LC_\theta(s)$ regulates the commanded attitude error $\boldsymbol{\theta}_e$ to produce the virtual control vector $\boldsymbol{\nu}_2(s)$, which defines the desired attitude rate.

The angular rate command is obtained by subtracting the estimated kinematic term $\hat{\boldsymbol{f}}_2(s)$ from $\boldsymbol{\nu}_2(s)$ and mapping the result through the inverse of the kinematic matrix. The matrices $\boldsymbol{G}_2(s)$ and $\hat{\boldsymbol{G}}_2(s)$ map angular rates to attitude rates, and since this relation is known analytically, they are exactly equal. The resulting closed-loop relation for $\boldsymbol{\theta}(s)$ is:

$$\boldsymbol{\theta}(s) = \frac{H_\omega(s)\, LC_\theta(s)}{s + H_\omega(s)\, LC_\theta(s)}\, \boldsymbol{\theta}_{\mathrm{cmd}}(s) - \frac{H_\omega(s)}{s + H_\omega(s)\, LC_\theta(s)}\, \hat{\boldsymbol{f}}_2(s) + \frac{1}{s + H_\omega(s)\, LC_\theta(s)}\, \boldsymbol{f}_2(s). \tag{60}$$

At this stage, the bandwidth separation assumption becomes relevant. Including the full inner-loop dynamics makes the system effectively fourth order, complicating controller design. To simplify the problem, the standard approach in the literature is to assume that the inner rate loop operates on a sufficiently faster timescale than the outer attitude loop.

[‡]This trade-off is intrinsic to the structure of the system and the absence of any compensatory phase lead in the proportional controller. Achieving higher closed-loop bandwidths would require a lead or PD compensator to provide additional phase margin.

If the inner-loop bandwidth is sufficiently higher than that of the attitude loop, it is reasonable to approximate the rate loop as ideal, i.e. $\omega(s) \approx \omega_{\text{cmd}}(s)$, which implies $H_\omega(s) \approx 1$. With this approximation, the attitude dynamics reduce to:

$$\boldsymbol{\theta}(s) = \frac{LC_\theta(s)}{s + LC_\theta(s)}\,\boldsymbol{\theta}_{\text{cmd}}(s) + \frac{1}{s + LC_\theta(s)}\left(\boldsymbol{f}_2(s) - \hat{\boldsymbol{f}}_2(s)\right), \tag{61}$$

where $\boldsymbol{f}_2(s)$ represents the true flight-path kinematics and $\hat{\boldsymbol{f}}_2(s)$ denotes its model estimate used in the controller. Due to the assumptions in the attitude control loop design, $\boldsymbol{f}_2(s) \neq \hat{\boldsymbol{f}}_2(s)$, but the mismatch is extremely small in magnitude and varies slowly with time. As a result, the induced tracking error remains bounded and does not compromise overall system performance. The closed loop from $\boldsymbol{\theta}_{\text{cmd}}(s)$ to $\boldsymbol{\theta}(s)$ is therefore:

$$\lim_{H_\omega(s)\to 1} H_\theta(s) = \frac{LC_\theta(s)}{s + LC_\theta(s)}. \tag{62}$$

The bandwidth separation assumption effectively decouples the rotational axes in the outer loop, allowing each attitude channel to be treated independently. While this simplification introduces a potential risk of instability, theoretical results show that exponential stability of the outer-loop states around their commanded values is still guaranteed if the inner-loop bandwidth is sufficiently high [19]. In practice, it is recommended that the rate loop bandwidth exceeds the attitude loop bandwidth by at least a factor of four to ensure that the approximation remains accurate [20]. For this single–integrator loop structure, exact second–order closed–loop dynamics with the desired natural frequency ω_θ and damping ratio ζ_θ can be obtained through pole placement. This is achieved using a proportional–with–roll–off controller of the form:

$$LC_\theta(s) = \frac{K_\theta}{1 + s/\omega_f} = \frac{K_\theta \omega_f}{s + \omega_f}.$$

Substituting the controller into the attitude dynamics yields the second–order pitch response:

$$H_\theta(s) = \frac{K_\theta \omega_f}{s^2 + \omega_f s + K_\theta \omega_f}, \qquad \omega_f = 2\zeta_\theta \omega_\theta, \qquad K_\theta = \frac{\omega_\theta}{2\zeta_\theta}.$$

These expressions directly link the controller parameters to the desired dynamic characteristics. Using the design values specified in the control requirements yields the second order closed-loop system:

$$H_\theta(s) = \frac{38.84}{s^2 + 11.22s + 38.84}, \quad \text{using} \quad LC_\theta(s) = \frac{38.84}{s + 11.22}. \tag{63}$$

The corresponding pole locations are summarized in Table 2. Unlike the rate control loop, however, the desired damping ratio is not fully achieved in the full closed loop. Increasing the time-scale separation would cause this value to approach 0.9, but such an adjustment would also slow down the overall system response. Hence, the trade-off lies between achieving the exact damping ratio or maintaining a faster response.

Table 2 Pole locations, natural frequency and damping for the full and the reduced attitude closed-loop system

System	Real ± Imag	ω_n, rad/s	ζ	Type
$H_{\theta_{\text{full}}}$	$-23.2 \pm 29.4i$	37.5	0.619	Actuator dynamics
	$-28.2 \pm 0.00i$	28.2	1.00	Rate pole
	$-3.66 \pm 4.45i$	5.77	0.635	Attitude poles
$H_{\theta_{\text{red}}}$	$-5.61 \pm 2.72i$	6.23	0.900	Attitude poles

Velocity Loop

The outer velocity loop regulates the geographic airspeed V_K, the heading angle χ_K, and the flight–path angle γ_K. This regulation is achieved indirectly by generating small changes in attitude and thrust, which the inner attitude and rate loops convert into linear accelerations.

The cascaded control structure used for velocity regulation is illustrated in Fig. 6. Commands originate from the position control loop, and the resulting command error is processed by the linear controller LC_V to produce the desired attitude and thrust setting rates. The error in the commanded derivatives is mapped through the control effectiveness matrix from Eq. (29), generating incremental attitude commands. These increments are added to the previous values and forwarded to the inner loops for bank angle and angle of attack, while the throttle command is sent directly to the engine.

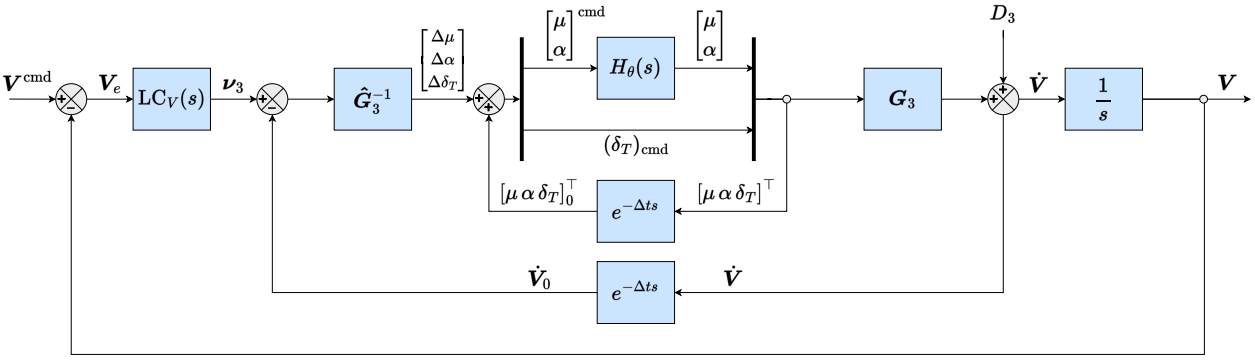

Fig. 6 Closed-loop block diagram of the velocity loop, assuming perfect attitude and angular rate tracking.

For the linear controller design, the closed–loop expression plays a central role. Under the assumption that the estimated control effectiveness matrix $\hat{\boldsymbol{G}}_3$ is exact and that disturbances are rejected sufficiently fast, the closed–loop relation is:

$$\boldsymbol{V}(s) = \frac{H_\theta(s)LC_V(s)}{s + H_\theta(s)LC_V(s)}\,\boldsymbol{V}_{\text{cmd}}(s), \qquad \lim_{H_\theta(s)\to 1} H_V(s) = \frac{LC_V(s)}{s + LC_V(s)}. \tag{64}$$

This relation applies to the angle-of-attack and bank-angle channels, where the inversion-based commands pass through inner dynamics represented by $H_\theta(s)$. The throttle channel does not include this term, since it acts directly on the propulsion system. When the inner loops are sufficiently fast such that $H_\theta(s) \approx 1$, the closed-loop map reduces to the expression shown on the right.

The velocity controller is obtained using the same second–order pole–matching approach applied in the attitude-loop design together with the bandwidth-separation and damping requirements, resulting in the closed-loop relation:

$$H_V(s) = \frac{2.428}{s^2 + 2.181s + 2.428}, \quad \text{using} \quad LC_V(s) = \frac{2.428}{s + 2.181}. \tag{65}$$

The poles of the reduced and complete transfer function, which includes the inner control loops, are given in Table 3:

Table 3 Pole locations, natural frequency and damping for the full and the reduced velocity closed-loop system

System	Real ± Imag	ω_n [rad/s]	ζ [–]	Type
$H_{V_{\text{full}}}$	$-23.2 \pm 29.4i$	37.5	0.62	Actuator poles
	$-28.2 + 0.00i$	28.2	1.00	Rate pole
	$-4.08 \pm 4.43i$	6.0	0.68	Attitude poles
	$-0.67 \pm 1.33i$	1.5	0.45	Velocity poles
$H_{V_{\text{red}}}$	$-1.09 \pm 1.11i$	1.6	0.70	

Position Loop

The position control loop determines the required heading χ_K and flight-path angle γ_K commands to track the commanded waypoints in latitude, longitude, and altitude. As shown in Fig. 7, these waypoint commands originate from the onboard guidance system, and the resulting command error is processed by the linear controller $LC_X(s)$ to

produce the desired changes in heading and flight-path angle. The errors in the commanded derivatives are then mapped through the inverse of the matrix $\boldsymbol{G}_4$ to account for spherical Earth effects, yielding intermediate commanded inputs. These are used in the inverse of Eq. (13) to generate the corresponding commands passed to the inner loops, which realize the actual heading and flight-path angle. The resulting control inputs are subsequently remapped through $\boldsymbol{G}_4$ to obtain the geodetic rate derivatives, which are integrated to produce the achieved tracking vector.

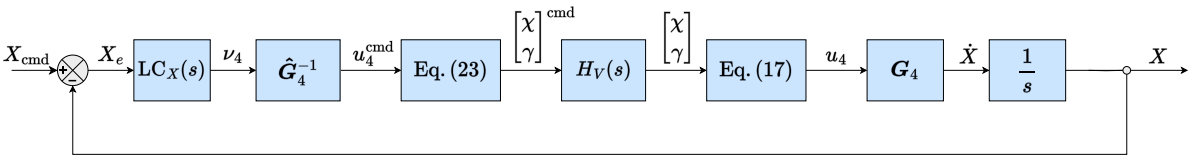

Fig. 7 Closed loop block diagram of the position control loop.

Analogous to the attitude loop, and assuming the estimated mapping $\hat{\boldsymbol{G}}_4$ is exact, the position closed–loop transfer function is given by the expression on the left below. This formulation incorporates the inner velocity dynamics through $H_V(s)$. When the velocity loop is sufficiently fast such that $H_V(s) \approx 1$, the closed–loop map reduces to the simplified first–order form shown on the right:

$$\boldsymbol{X}(s) = \frac{H_V(s)\, LC_X(s)}{s + H_V(s)\, LC_X(s)}\, \boldsymbol{X}_{\text{cmd}}(s), \qquad \lim_{H_V(s)\to 1} H_X(s) = \frac{LC_X(s)}{s + LC_X(s)}. \tag{66}$$

As with the attitude and velocity loops, the controller is derived using the same second-order pole-matching procedure applied to the inverted dynamics, together with the bandwidth-separation and damping requirements, producing the second-order closed-loop system:

$$H_X(s) = \frac{0.1517}{s^2 + 0.7012s + 0.1517}, \quad \text{using} \quad LC_X(s) = \frac{0.1517}{s + 0.7012}. \tag{67}$$

The poles corresponding to the reduced transfer function and the complete transfer function, which includes the inner control loops, are given in Table 4:

Table 4 Pole locations, natural frequency and damping for the full and the reduced position closed-loop system

System	Real ± Imag	ω_n, rad/s	ζ	Type
$H_{X_{\text{full}}}$	$-23.2 \pm 29.4i$	37.5	0.619	Actuator poles
	$-28.2 \pm 0.00i$	28.2	1.000	Rate pole
	$-4.08 \pm 4.43i$	6.02	0.677	Attitude poles
	$-0.74 \pm 1.28i$	1.48	0.502	Velocity poles
	$-0.28 \pm 0.27i$	0.39	0.716	Position poles
$H_{X_{\text{red}}}$	$-0.35 \pm 0.17i$	0.39	0.900	

The stability margins in Table 5 show that all control loops maintain sufficient gain, phase, and delay margins, consistent with well-damped and stable behavior. As expected, the rate loop has the highest crossover frequency and smallest delay margin, while the outer loops exhibit lower bandwidths and larger delay margins, preserving the desired time-scale separation. Overall, the ideal Continuous Time (CT) control structure achieves zero steady-state error and robust stability across all levels of the cascaded NDI–INDI architecture.

VI. Digital Control Design

In practice, FCSs must be implemented on digital FCCs, which operate in DT. A straightforward approach is to design a controller in CT and then transform it into a discrete controller using, for example, the bilinear transform or the matched pole–zero method. This approach, however, neglects the inherent properties of digital implementation: the

controller interacts with sampled signals, Zero-Order Hold (ZOH) devices, computation delays, and on-board filtering, none of which are captured in a purely CT design [12]. Only when the sampling interval Δt is very small can one argue that the discrete controller obtained from such a transformation accurately reproduces the performance of its CT counterpart.

A more sophisticated approach is the so-called modified CT controller design. Here, the discrete effects of sampling, holding, and computation are modeled explicitly as CT transfer functions and included in the synthesis process. The controller is then designed in this modified CT framework, so that the delay and hold dynamics are taken into account from the outset. Finally, the bilinear transform or a similar mapping is applied to obtain the actual digital controller for implementation. This approach ensures that the implemented digital controller preserves the intended performance characteristics even for practical sampling rates. Within this framework, the additional dynamics introduced by digital implementation must be modeled explicitly, as they influence both stability margins and transient response. The considered effects are:

1) **Sensor anti-aliasing filter.** Before sampling, sensor signals are band-limited to prevent aliasing, which folds high-frequency content above the Nyquist frequency into the measured bandwidth [12]. The anti-aliasing low-pass filter defined in Eq. (9) attenuates these components but introduces phase lag and gain reduction, both of which must be included in the plant model $G(s)$ for accurate analysis.
2) **Sampler and ZOH.** After sampling, the controller updates commands only at intervals of Δt, while the ZOH holds the actuator input constant between updates. This piecewise-constant reconstruction attenuates high-frequency content and introduces an effective delay of roughly $\Delta t/2$. The corresponding CT transfer functions are:

$$G_{ZOH}(s) = \frac{1 - e^{-\Delta t s}}{s}, \qquad G_{SH}(s) = \frac{1 - e^{-\Delta t s}}{s\,\Delta t}. \tag{68}$$

   The second expression includes the sampling action through normalization by the sampling interval.
3) **Computation delay.** The FCC requires finite time to read measurements, compute the control input, and send the command to the actuator. A conservative model assumes the worst case which is a full-sample computation delay of Δt, represented in CT as:

$$G_D(s) = e^{-\Delta t s}. \tag{69}$$

In addition to the modeled effects, real sensors may introduce further delays, as noted in Eq. (10). Because these delays are typically unknown at the design stage, they are not included in the modified CT controller synthesis. Instead, potential sensor delays are introduced in the high-fidelity simulation to evaluate how much delay the controller can tolerate.

The overall control structure, shown in Fig. 8, includes the key effects introduced by digital implementation. The controller $LC(s)$ operates together with the sample-and-hold element $G_{SH}(s)$, computation delay $G_D(s)$, and sensor anti-aliasing filter $G_a(s)$. Combined with the plant dynamics $G(s)$, these components form a CT representation of how the digital controller interacts with the aircraft.

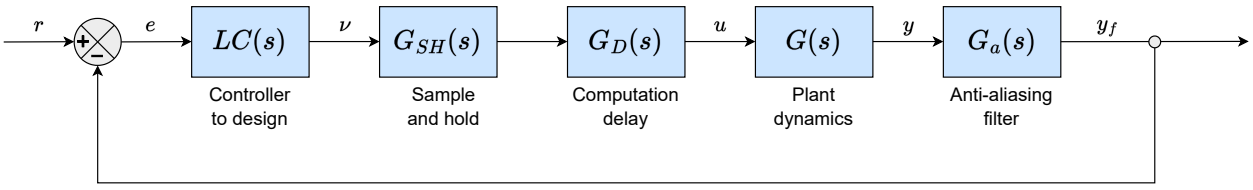

Fig. 8 CT control structure with digital implementation effects [12].

The transfer functions $G_{SH}(s)$ and $G_D(s)$ contain exponential terms and are therefore irrational, which is inconvenient for controller synthesis where rational transfer functions are preferred. A standard remedy is to approximate these exponentials with Padé expansions. The (m, n) Padé approximation represents a function by a ratio of two polynomials whose Taylor series matches the first $m + n$ terms of the original function [12]. Padé forms capture the key delay behavior, including phase lag, with low-order rational models suitable for analysis and simulation. The third-order approximations used here follow [12].

For the sample-and-hold and the computation delay transfer function, a third-order Padé approximation of the exponential yields:

$$G_{SH}(s) \approx \frac{1 - \frac{1}{14}(s\Delta t) + \frac{23}{840}(s\Delta t)^2 - \frac{1}{840}(s\Delta t)^3}{1 + \frac{3}{7}(s\Delta t) + \frac{1}{14}(s\Delta t)^2 + \frac{1}{120}(s\Delta t)^3}, \qquad G_D(s) \approx \frac{1 - \frac{1}{2}(s\tau_{CD}) + \frac{1}{10}(s\tau_{CD})^2 - \frac{1}{120}(s\tau_{CD})^3}{1 + \frac{1}{2}(s\tau_{CD}) + \frac{1}{10}(s\tau_{CD})^2 + \frac{1}{120}(s\tau_{CD})^3}. \tag{70}$$

It is important to note that the influence of digital implementation effects strongly depends on the sampling interval. For small sampling times, the impact of sample–and–hold behavior, computation delay, and sensor filtering is minor and can often be neglected. However, as the sampling interval increases, these effects introduce additional phase lag and attenuation that can noticeably degrade closed–loop performance. Consequently, explicitly accounting for these effects in the controller design is advisable to preserve the intended performance.

A representative block diagram illustrating the inclusion of discrete effects (shown in pink), as well as sensor delay and anti-aliasing effects, is presented in Fig. 9.

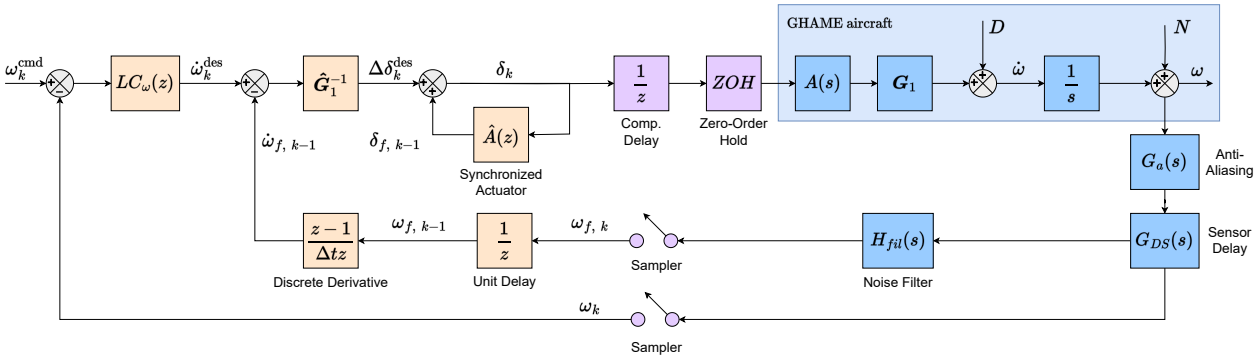

Fig. 9 DT INDI rate control loop including actuator dynamics, sensor dynamics, sampling, and noise filtering.

The digital effects on the rate control loop are now considered. When the control structure from Fig. 8 is applied to the angular rate loop described in Sec. V, the additional dynamics introduced by the sample-and-hold, computation delay, and anti-aliasing filter invalidate the previously calculated open-loop gain and phase margins.

The phase effects of the digital elements can be assessed directly at the crossover frequencies. A one–sample computational delay with $\Delta t = 0.01$ s introduces about $-7.7°$ of lag at the phase crossover frequency of 13.5 rad/s. The anti–aliasing filter contributes roughly $-4.9°$, and the ZOH adds about $-3.9°$, giving a total of approximately $-16.5°$. This reduces the CT phase margin from $67.6°$ to about $51.1°$ in the digital implementation. At the gain crossover frequency of 50 rad/s, the same mechanisms generate nearly $-60°$ of lag. Although the exact gain–margin reduction cannot be computed analytically, the additional phase clearly lowers the achievable margin. These reductions confirm that a sampling time of $\Delta t = 0.01$ s is not small enough for digital effects to be ignored.

Table 5 Comparison of stability margins for each control loop without and with digital effects (D)

Loop	Gain margin, dB	Phase margin, deg	Delay margin, s	Crossover, rad/s
Rate	14.3	67.6	0.0872	13.5
Rate (D)	7.67	51.2	0.0664	13.5
Attitude	13.5	59.7	0.315	3.3
Attitude (D)	10.9	57.9	0.301	3.35
Velocity	10.0	48.1	0.831	1.01
Velocity (D)	9.78	47.7	0.816	1.02
Position	12.8	62.4	5.19	0.21
Position (D)	12.7	62.4	5.19	0.21

The impact of discretization becomes smaller as the dynamics slow down and is therefore most relevant in the rate and attitude loops. Table 5 summarizes the resulting gain and phase margins for the continuous and digital cases. With the nominal CT gain from Eq. (58), the digital implementation yields a gain margin of 7.67 dB, a phase margin of 51.20°, and a delay margin of 0.066 s. The corresponding step response shows an overshoot of 5.2%, meaning the zero–overshoot requirement is no longer met. The reduction in stability is not severe, but it is significant enough that it should be considered when choosing controller parameters.

Table 6 Final stability margins obtained with the redesigned controllers from Eq. (71), based on the modified CT control design

Loop	Gain margin, dB	Phase margin, deg	Delay margin, s	Crossover, rad/s
Rate	12.3	67.3	0.148	7.95
Attitude	12.9	58.7	0.52	2.0
Velocity	9.83	47.8	1.37	0.61
Position	12.8	62.4	8.64	0.125

Two design paths can now be considered. The first option is to accept the reduced stability margins and the small overshoot as sufficient for operation, since the system remains stable with acceptable stability margin. The second option is to redesign the linear controllers with the digital effects explicitly included, thereby recovering the intended performance targets. This approach is adopted here to restore the zero-overshoot behavior of the rate loop and to improve the limited delay margins observed under digital implementation. The resulting four linear controllers obtained from this reoptimization are presented in Eq. (71), and their corresponding stability margins are summarized in Table 6:

$$LC_\omega = 7.9663, \quad LC_\theta(s) = \frac{13.96}{s+6.726}, \quad LC_V(s) = \frac{0.8726}{s+1.308}, \quad \text{and} \quad LC_X(s) = \frac{0.05454}{s+0.4204}. \tag{71}$$

To summarize the key characteristics of the developed cascaded controller, the control architecture consists of four nested loops: rate, attitude, velocity, and position. Each outer loop is tuned to operate at a lower bandwidth than the one inside it, ensuring that the inner dynamics are already attenuated within its operating range. This arrangement minimizes coupling between loops and simplifies both tuning and analysis. The corresponding loop transfer functions are shown in Fig. 10. The magnitude plot illustrates clear bandwidth separation among the loops. For clarity, reference lines at 0 dB and −180° are added to visualize crossover frequencies and phase margins.

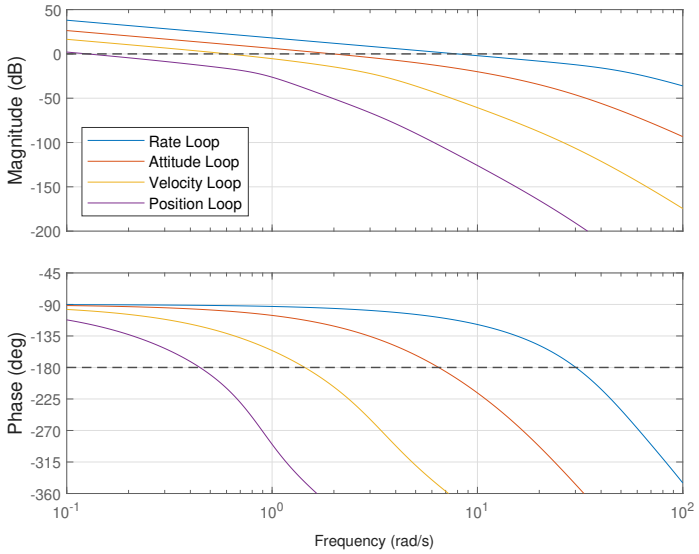

Fig. 10 Open loop Bode magnitude and phase for the rate, attitude, velocity, and position loops.

The parameters used in the modified CT control design are listed in Table 7. They include the selected bandwidths, damping ratios, filter and actuator characteristics, and timing values that define the dynamic properties of each loop. These parameters produce the frequency responses shown in Fig. 10.

Table 7 Design parameters used for control law development using the modified continuous control design.

Parameter	Symbol	Value	Unit
Filter corner freq.	ω_H	25	rad/s
Filter damping	ζ_H	1	-
Actuator nat. freq.	ω_{act}	50.0	rad/s
Actuator damping	ζ_{act}	0.707	-
Rate bandwidth	BW_ω	15.0	rad/s
Attitude damping	ζ_θ	0.9	-
Attitude bandwidth	BW_θ	3.75	rad/s

Parameter	Symbol	Value	Unit
Velocity nat. freq.	BW_V	0.9375	rad/s
Velocity damping	ζ_V	0.7	-
Position nat. freq.	BW_X	0.234375	rad/s
Position damping	ζ_X	0.9	-
Anti-aliasing freq.	ω_a	157.07	rad/s
Comp. delay	τ_{CD}	0.01	s
Sampling time	Δt	0.01	s

VII. Digital Implementation

Digital implementation is required because the FCS and the hierarchical controllers in Eq. (71) were developed in CT, whereas the FCC operates with a finite sampling interval. The effects of sampling and delays have already been incorporated in the digital control design of Sec. VI. The objective here is therefore to discretize the CT components that must execute on the FCC, including the derivative blocks and the hierarchical INDI controllers. All signals entering the FCC must be sampled accordingly. The aircraft dynamics, actuator models, sensor dynamics, and noise filters remain in CT, requiring the FCC outputs to interface with continuous-time subsystems. Although this interaction would nominally require a zero-order hold, Simulink manages the discrete–continuous rate transitions automatically, so no explicit conversion blocks are needed.

The final layout of the hierarchical control system is shown in Fig. 11. The diagram presents all five main components: the sampling of input signals into the FCC, followed by the four control loops described previously. This configuration represents the DT Simulink implementation of the architecture shown in Fig. 2.

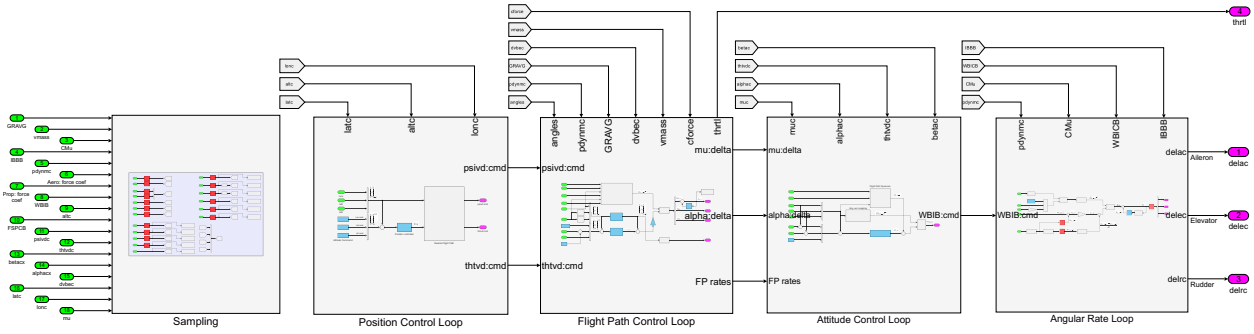

Fig. 11 DT hierarchical INDI control architecture implemented in Simulink, including the sampling subsystem.

All controllers are executed with a sampling time of $\Delta t = 0.01$ s. Simulations use a fixed-step solver to capture discretization effects, with the step size set to $\Delta t/25$, which corresponds to 4×10^{-4} s.

Position Control

The Simulink representation of the DT position control loop is shown in Fig. 12. The function implements the equations in Eq. (18) and Eq. (19). The green inputs represent sampled variables from the sampling subsystem, while the blue blocks correspond to the commanded way points and the DT position controller. The blue color indicates that

these elements operate at a defined sampling time. The pink blocks denote the outputs, which are the commanded flight path and heading signals passed to the next control loop.

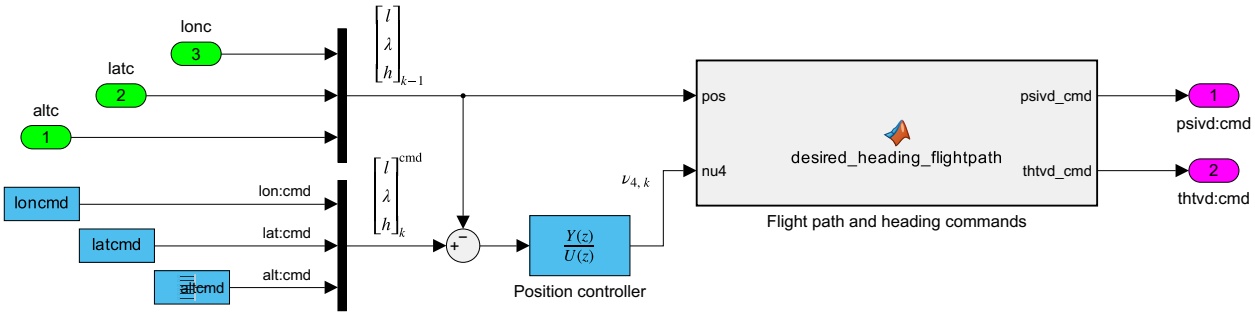

Fig. 12 Simulink implementation of the DT position control loop.

The DT position controller is obtained by discretizing the CT controller in Eq. (71). The Tustin transformation is used for this purpose, mapping the s-domain to the z-domain while preserving stability and accurately capturing the low-frequency dynamics. With a sampling period of $\Delta t = 0.01$ s, the resulting DT controller is:

$$LC_X(z) = \frac{4.0319 \times 10^{-5} + 4.0319 \times 10^{-5} z^{-1}}{1 - 0.99838\, z^{-1}}. \tag{72}$$

The DT controller maintains the dominant low-frequency behavior of the CT design, yielding comparable closed-loop performance within the operating bandwidth. Differences appear mainly at higher frequencies due to frequency warping and the finite sampling period.

Velocity and Flight Path Control

The DT velocity control loop is shown in Fig. 13. The inverse of Eq. (31) is computed for control allocation within this loop. The throttle actuator is assumed to track the commanded change within one sampling period, so a UD block provides a sufficient approximation for generating the incremental input for speed control. The remaining incremental inputs for commanded bank angle and angle of attack are passed to the attitude loop and combined with values from the previous sampling instant. The heading and flight-path derivatives are likewise provided to that loop for its control law. The speed command is set by a constant block defining the trimmed velocity, and a step block introduces heading changes when the hold-heading mode is active.

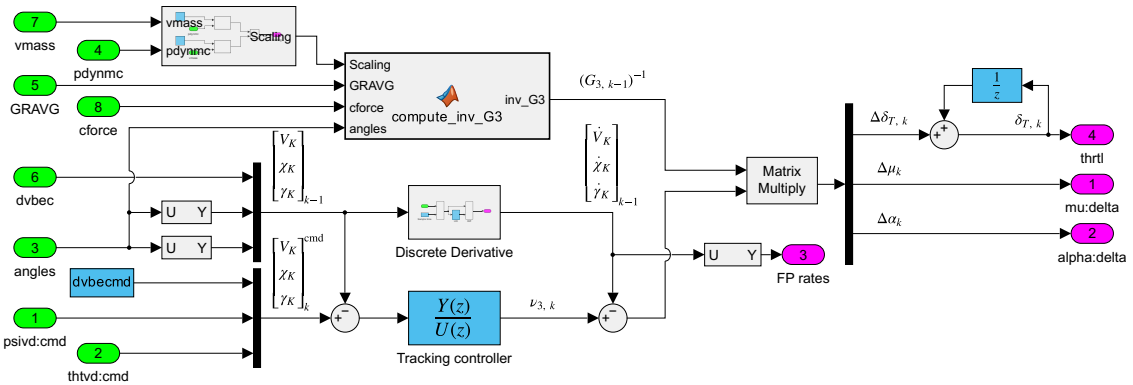

Fig. 13 Simulink implementation of the DT tracking control loop.

A key feature of this loop is that the derivatives required for the INDI control law are computed using the backward difference numerical differentiation method, as shown in Fig. 14.

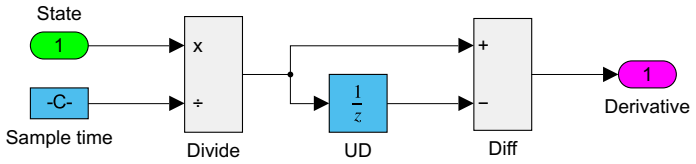

Fig. 14 Discrete differentiation algorithm used for computing derivatives.

Unlike the rate loop, no second–order noise filter is applied before differentiation. The INS model used here introduces errors only in angular rate and excludes tilt, velocity, and position errors. The resulting airspeed, flight path, and heading signals are therefore treated as noise-free, and additional filtering is not required.

The DT velocity controller is again obtained by discretizing the CT design in Eq. (71) using the Tustin transformation. With a sampling period of $\Delta t = 0.01$ s and normalized coefficients, the resulting discrete transfer function is:

$$LC_V(z) = \frac{0.0003297 + 0.0006595\, z^{-1} + 0.0003297\, z^{-2}}{1 - 1.9220\, z^{-1} + 0.9249\, z^{-2}}. \tag{73}$$

Attitude Control:

The attitude control loop is implemented using NDI, which eliminates the need for discrete derivative approximations and thereby simplifies implementation in Simulink. The overall structure of the loop is shown in Fig. 15. At each integration step, the terms $\boldsymbol{G}_2$ and $\boldsymbol{f}_2$ are computed according to Eq. (39) to perform the dynamic inversion. The attitude command is formed by combining the incremental control inputs from the preceding velocity loop with the sampled states provided by the INS. The sideslip command is fixed at zero to maintain coordinated flight, and the resulting rate command is passed to the rate control loop.

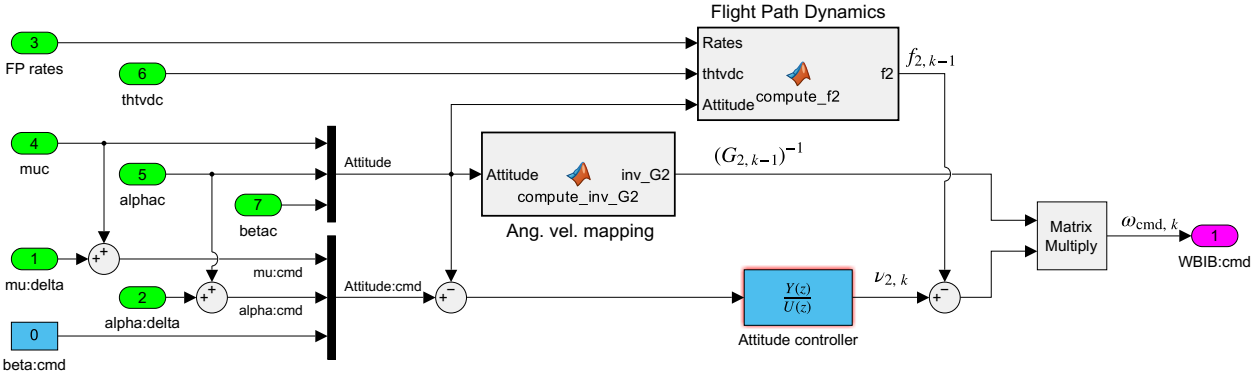

Fig. 15 Simulink implementation of the attitude control loop. The discrete controller $LC_\theta(z)$, obtained via Tustin discretization, is located in the linear control subsystem.

The attitude controller is obtained by discretizing the CT design in Eq. (71) using the Tustin transformation. With a sampling period of $\Delta t = 0.01$ s and normalized coefficients, the resulting discrete transfer function is:

$$LC_\theta(z) = \frac{0.1271 + 0.1271\, z^{-1}}{1 - 0.9113\, z^{-1}}. \tag{74}$$

Rate Control:

The rate control loop is implemented using INDI and therefore requires discrete derivatives. Unlike the velocity control loop, noise filtering is applied in this case and sensor dynamics are present, which introduces the need for actuator synchronization. The DT block diagram of the INDI rate controller, including sensor, noise, and actuator dynamics, is shown in Fig. 16 it is essentially the Simulink implementation of Fig. 9.

In the upper path, the control effectiveness matrix $\boldsymbol{G}_1$ is computed according to Eq. (45). The lower path contains the CT implementation of the sensor dynamics, representing anti-aliasing effects plus an unknown delay. The INS state is subsequently processed through a CT noise filter before being sampled to obtain the discrete derivatives. The angular rate state that is not differentiated bypasses the noise filter and is directly sampled. Another important aspect is the worst-case computational delay in the simulation, this effect is modeled using a UD block placed immediately before the control signal is sent to the actuators. For the rate control loop, the controller reduces to a constant linear gain, which requires no translation into DT.

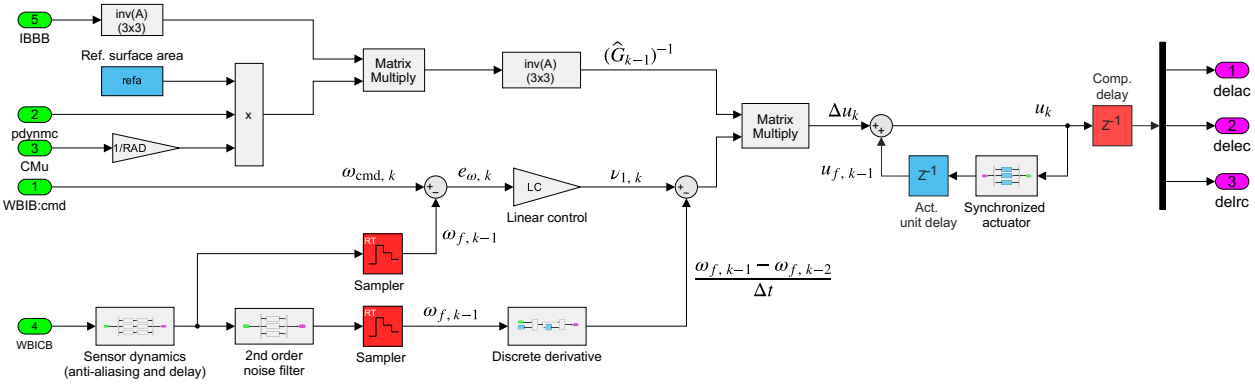

Fig. 16 Simulink implementation of the INDI subsystem for angular rate control, including sensor, noise, and actuator dynamics.

The key aspect here is the discretization of the actuator path, which includes the synchronization transfer functions. The corresponding discrete formulation for implementation within the digital control loop is presented next.

Discrete Synchronized Actuators:

The physical actuators operate in CT, but their representation within the FCC must be discrete. The CT formulation of the combined actuator, synchronization dynamics, and filtering is based on the sensor synchronization method defined in [21] and is expressed as:

$$\hat{A}(s) = A(s)\, H_{\text{sync}}(s)\, H_{\text{fil}}(s)\, e^{-\Delta t s}, \tag{75}$$

where $A(s)$ denotes the actuator dynamics, $H_{\text{fil}}(s)$ represents the noise filter from Eq. (50), and $e^{-\Delta t s}$ models a one–sample delay aligning the actuator output with the previous control step. The term $H_{\text{sync}}(s)$ represents the synchronization dynamics, capturing all phase and timing mismatches between the actuator and measurement paths. In this system, these correspond to $G_a(s)$ and $G_{SD}(s)$ from Eqs. (9) and (10), respectively. The synchronized simulation includes both. In the unsynchronized case, only $G_a(s)$ is applied, and the sensor delay is modeled as an uncertainty.

All components except the delay are combined in the CT domain prior to discretization, because the transformation from the s–domain to the z–domain is nonlinear and must, in general, be applied to the complete transfer function [22]:

$$z\{S(s)G(s)\} \neq z\{S(s)\}\, z\{G(s)\}. \tag{76}$$

Here, $z\{\cdot\}$ denotes the z–transform, and discretization is performed using the ZOH method rather than the Tustin method, since it reproduces the sampled-data behavior of actuators with constant input during each sampling period.

By separating the non delayed portion of Eq. (75), obtain $A(s)\, H_{\text{sync}}(s)\, H_{\text{fil}}(s)$. Using this expression directly inside the ZOH discretization gives:

$$H_c(z) = z\left\{\frac{1-e^{-\Delta t s}}{s}\, A(s)\, H_{\text{sync}}(s)\, H_{\text{fil}}(s)\right\}. \tag{77}$$

The exponential term $e^{-\Delta t s}$ corresponds directly to a UD in the z–domain, i.e. multiplication by z^{-1}. The final DT synchronized actuator is therefore:

$$\hat{A}(z) = H_c(z)\, z^{-1}. \tag{78}$$

The resulting discretized synchronized actuator dynamics can be expressed as follows:

$$\hat{A}(z) = \frac{z^{-2}\left(1.3389 \cdot 10^{-4} + 0.0023z^{-1} + 0.0039z^{-2} + 9.7198 \cdot 10^{-4}z^{-3} + 2.3291 \cdot 10^{-5}z^{-4}\right)}{1 - 3.1777z^{-1} + 3.9921z^{-2} - 2.4448z^{-3} + 0.7099z^{-4} - 0.0722z^{-5}}. \tag{79}$$

The final step is the initialization of the discrete synchronized actuator. To avoid spurious transients at $t = 0$, the actuator must start at the calculated trim input; otherwise, a mismatch between the trim condition and the actuator state would occur. In a DT transfer function block in Simulink, however, the trim deflection cannot be assigned directly as the initial output. The parameter *Initial states* instead specifies the contents of the internal delay registers of the filter. Consequently, the internal states must be computed such that the filter output equals the trim deflection u_{trim} at the start of the simulation.

Simulink realizes a DT transfer function in state-space form as:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\,\boldsymbol{u}_k, \\ y_k &= \boldsymbol{C}\boldsymbol{x}_k + D\,\boldsymbol{u}_k, \end{aligned} \tag{80}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ are the filter states, $\boldsymbol{u}_k$ is the vector of control inputs, and y_k is the output. The matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, D)$ are obtained from the actuator numerator and denominator coefficients using the MATLAB command `tf2ss`. At trim, the input is constant, $\boldsymbol{u}_k = \boldsymbol{u}_{\text{trim}}$. Requiring the state to remain constant, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$, leads to the steady-state condition:

$$\boldsymbol{x}_0 = \left(\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B}\,\boldsymbol{u}_{\text{trim}}, \tag{81}$$

with $\boldsymbol{I}$ the identity matrix of the same size as $\boldsymbol{A}$. Assigning the vector $\boldsymbol{x}_0$ to the actuator block as its initial state ensures that, at $t = 0$, the actuator outputs match the trimmed control deflections $\boldsymbol{u}_{\text{trim}}$. The corresponding initial states associated with the trimmed states are summarized in Table 8.

Table 8 Initial condition for discrete actuators

Condition	Initial state $\boldsymbol{x}_0$	δ_{trim}, deg	α_{trim}, rad	θ_{trim}, rad
Mach 3, 60,000 ft	[0, -12.21, 0]	[0.0, -5.1039, 0.0]	0.0602	0.0602

VIII. Simulation Results

Effect of Sensor Delay:

The effect of unsynchronized sensor delay is analyzed by progressively increasing the measurement delay while keeping the actuator dynamics and filters unchanged. The sensor delay is increased by 0.01 s for each simulation run. The corresponding altitude, flight path, and heading responses are shown in Figs. 17, 18, and 19. The effect of sensor delay is twofold. First, it causes a desynchronization between the angular acceleration and actuator paths, which directly disturbs the incremental feedback mechanism. Second, it introduces a pure delay in the angular rate feedback loop, reducing both the gain and phase margins. While proper synchronization can mitigate the first issue by aligning the actuator and measurement paths, it cannot eliminate the second effect.

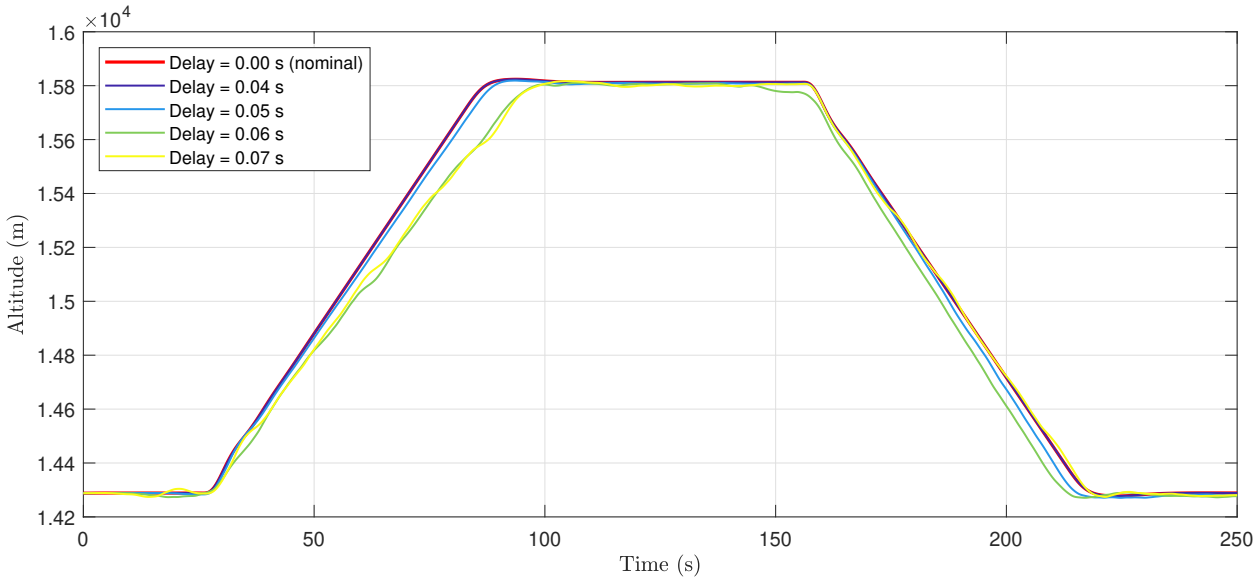

Fig. 17 Effect of increasing sensor delay on altitude tracking without sensor delay synchronization.

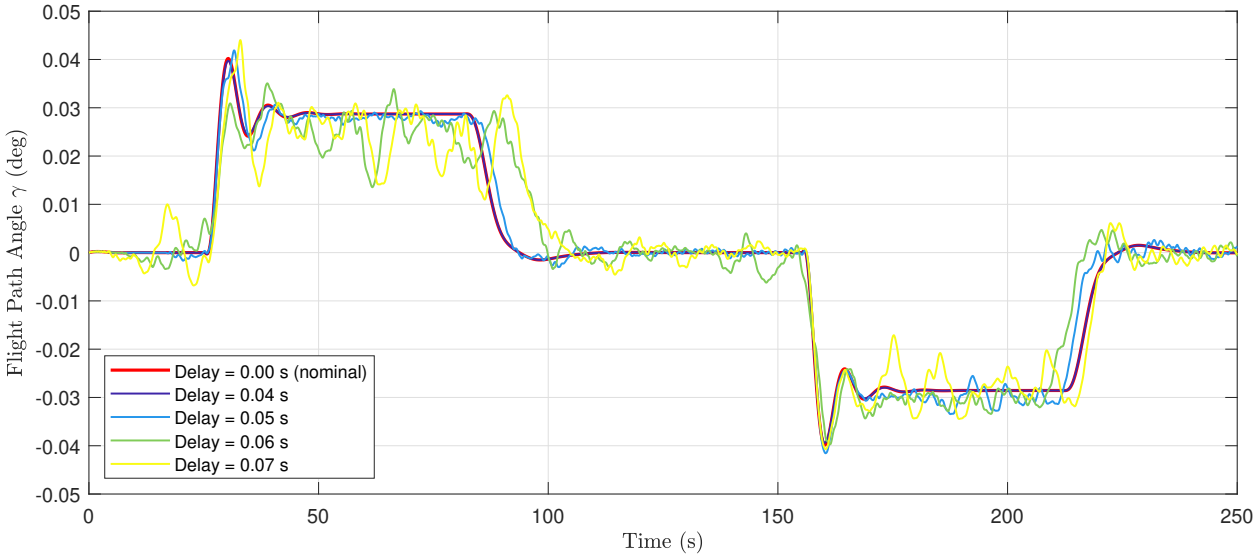

Fig. 18 Effect of increasing sensor delay on flight-path angle without sensor delay synchronization.

The controller maintains accurate tracking up to a sensor delay of approximately 0.04 s. Beyond this point, the response begins to deteriorate, as evident from Fig. 17. Although the altitude error remains moderate, the inner control loops exhibit growing oscillations that indicate the onset of instability, as shown in Fig. 18. Once the delay exceeds 0.04 s, the closed-loop dynamics degrade rapidly and eventually the simulation automatically terminates at a delay of 0.08 s when the angle of attack reaches 21°, which lies outside the aerodynamic data set. A similar trend appears in the heading and bank angle responses shown in Fig. 19 and Fig. 20. Oscillations develop once the sensor delay exceeds approximately 0.04 s, matching the onset observed in the altitude and flight path responses. However, the simulation remains stable for a longer duration and only terminates at 0.12 s. This delayed onset of instability is likely caused by the nonlinear effect introduced by the bank angle limitation, which constrains the commanded bank to within ±30°. Simulations performed without this constraint diverge earlier, at a delay of approximately 0.07 s.

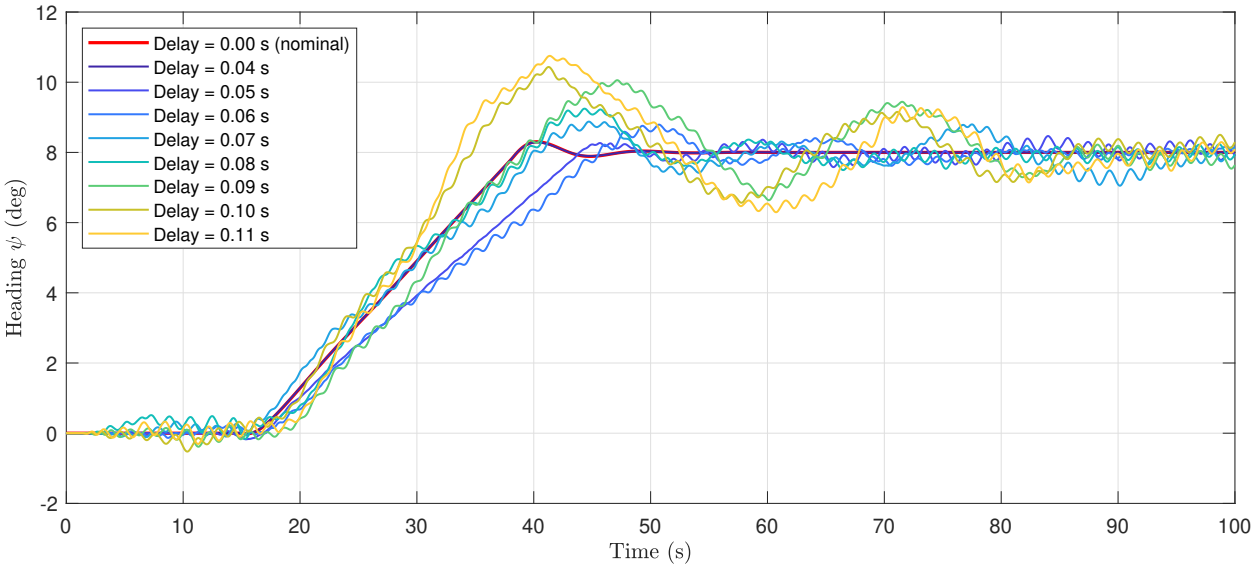

Fig. 19 Effect of increasing sensor delay on heading response without sensor delay synchronization.

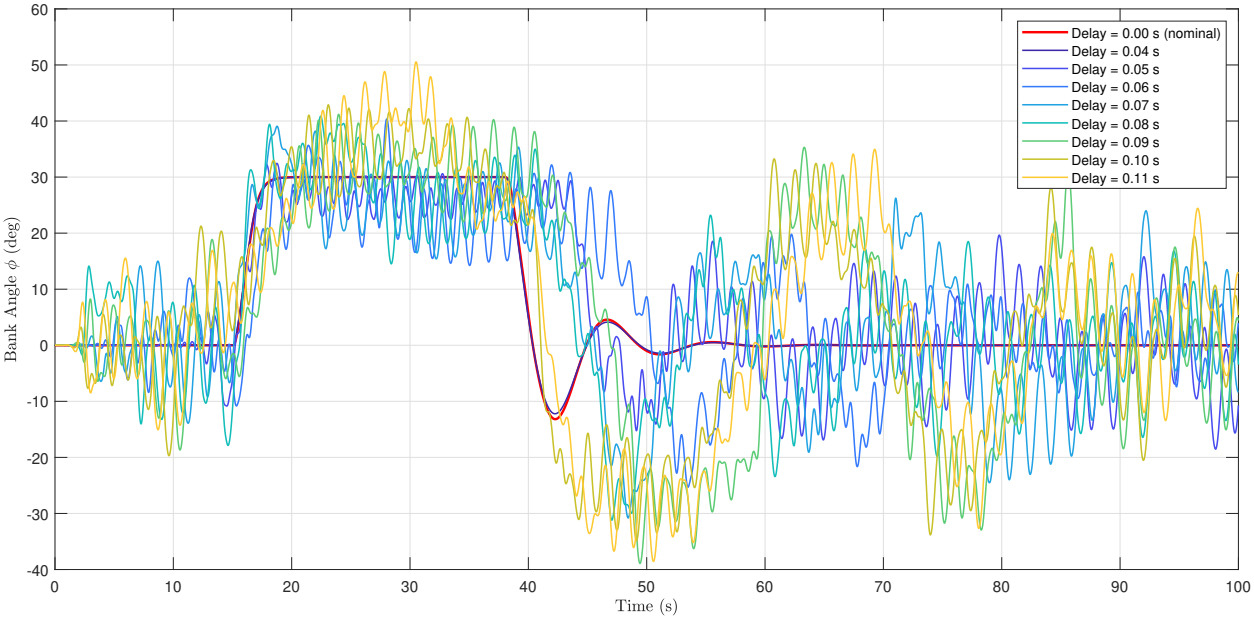

Fig. 20 Effect of increasing sensor delay on bank angle response without sensor delay synchronization.

Synchronized Time Delay

When sensor delay synchronization is not applied, the effective delay margin is only 0.04 s, far below the values in Table 6. Practical implementations therefore require sensor delay synchronization to compensate for timing offsets between the measurement and actuation paths. The idea is to introduce an equivalent delay in the actuator or reference signal so that both signals are aligned when processed by the controller, ensuring consistent information despite sensor latency. The following analysis presents results obtained with synchronized delays.

The synchronized altitude-response results are shown in Fig. 21 and Fig. 22. With synchronization enabled, the controller tolerates delays up to approximately 0.13 s before degradation occurs, more than twice the unsynchronized limit. At 0.14 s, sustained oscillations drive the angle of attack outside its valid aerodynamic range, causing the

simulation to terminate; the system is close to divergence. This follows from the continued erosion of gain and phase margins in the rate loop as the pure delay increases. The critical delay of 0.14 s matches the rate-loop delay margin in Table 6, showing that synchronization extends the stable delay range to the theoretical limit for the longitudinal case.

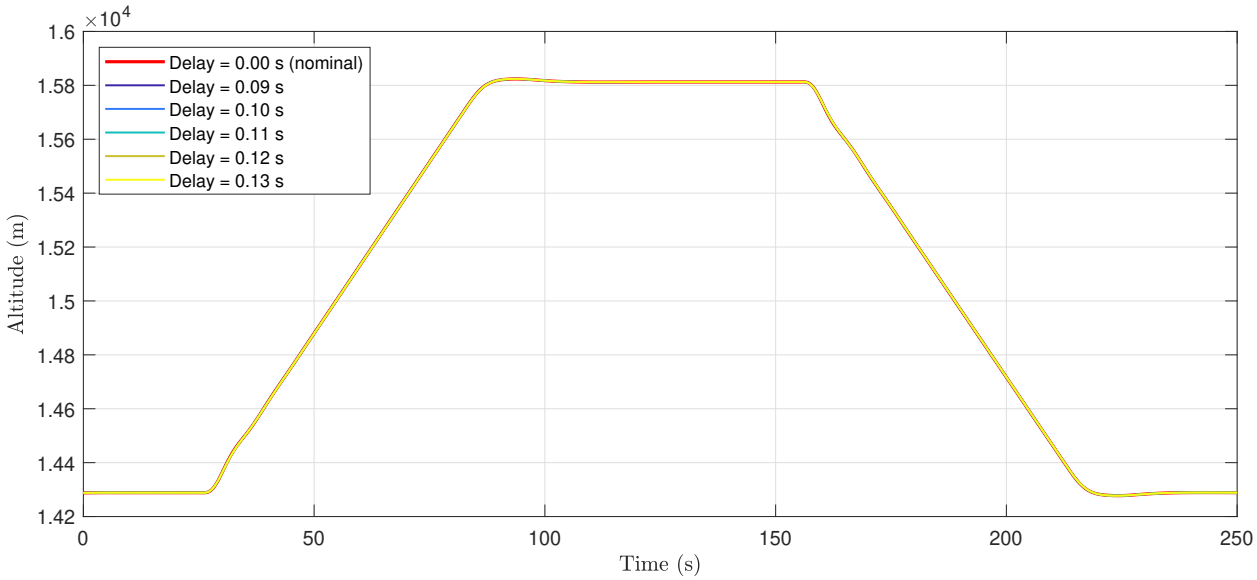

Fig. 21 Effect of increasing sensor delay on altitude tracking with sensor delay synchronization.

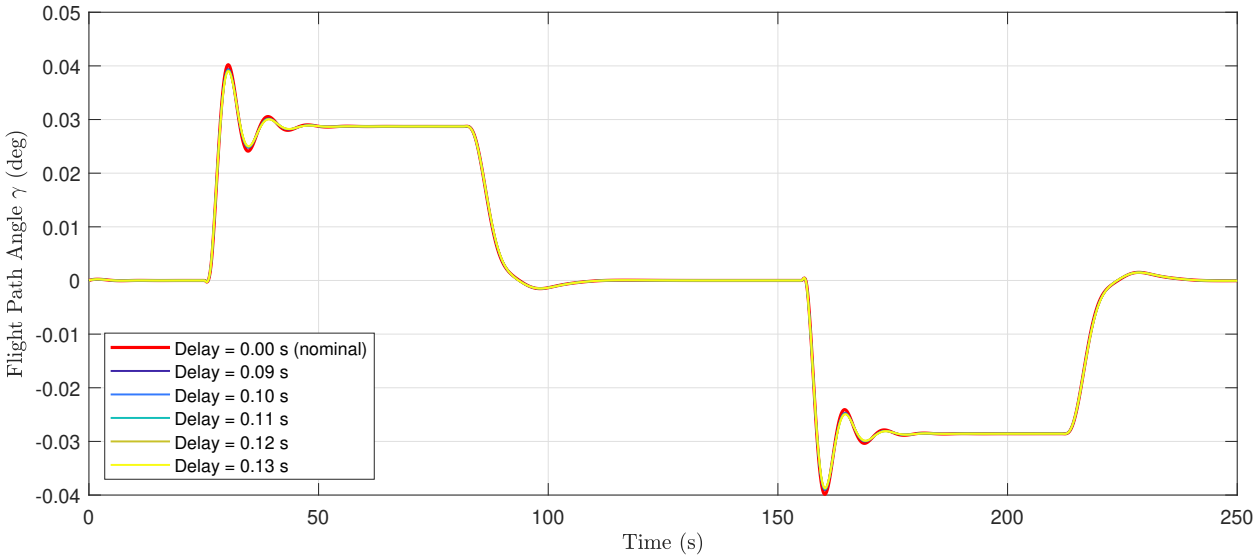

Fig. 22 Effect of increasing sensor delay on flight-path angle tracking with sensor delay synchronization.

For the heading command case shown in Fig. 23 and Fig. 24, the response differs notably from the longitudinal case. Oscillations appear at a sensor delay of about 0.08 s, much earlier than predicted by the calculated delay margin. This indicates that the lateral–directional loop is considerably more sensitive to delay. Beyond this point, the system still tracks the heading command but with large oscillations. The apparent stability up to around 0.16 s results mainly from nonlinear effects, particularly the roll command limit of ±30°, which prevents full divergence. Without this limit, the simulation would fail at smaller delays.

The early onset of oscillations indicates that a core assumption of the INDI formulation may not be fully satisfied for the lateral–directional dynamics of the GHAME vehicle. Examination of the airframe provides some clues. The

GHAME's slender body, low–aspect–ratio wings, and tight wing–body integration yield an unusually low roll moment of inertia, producing exceptionally fast roll dynamics. As a result, the lateral states evolve on time scales close to those of the control inputs rather than being clearly separated as assumed in the incremental formulation. Under these conditions, the system departs from the ideal chain–of–integrators behavior on which INDI relies. The measured acceleration increment is then affected not only by the control input but also by the natural evolution of the states, reducing the validity of the quasi–static mapping between input and acceleration. This coupling introduces additional phase lag in the feedback path and plausibly explains the premature oscillations observed in simulation.

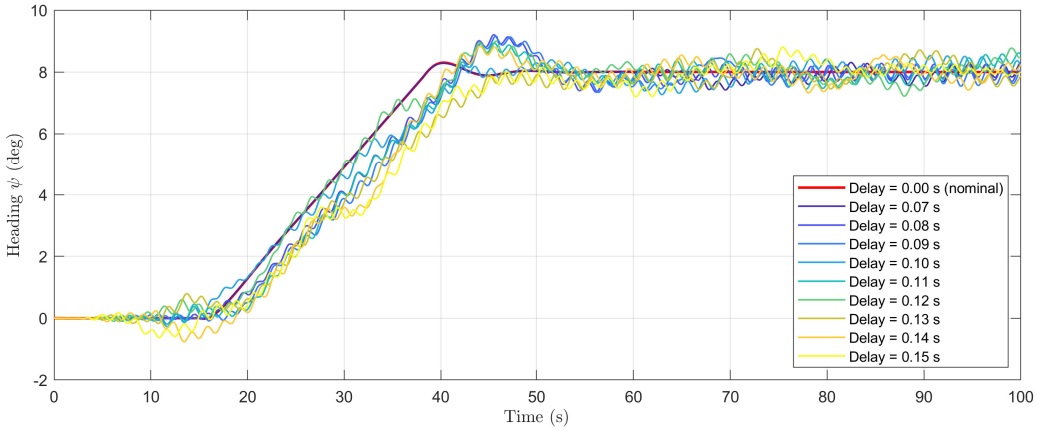

Fig. 23 Effect of increasing sensor delay on heading response with sensor delay synchronization.

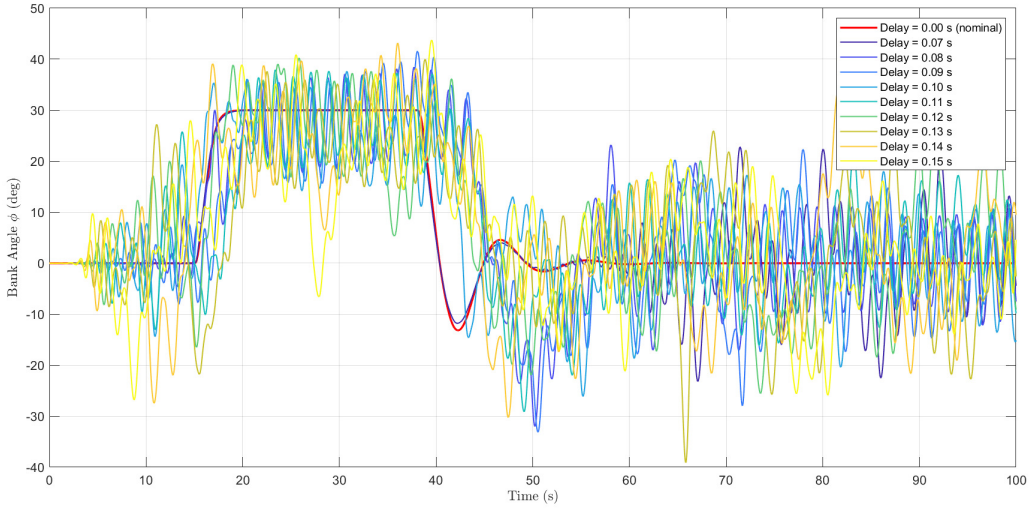

Fig. 24 Effect of increasing sensor delay on bank angle response with sensor delay synchronization.

If that is the case, the linear model used for design no longer provides an accurate representation of the system. Consequently, gain and phase margins derived from the linearized model are not predictive of the actual closed–loop behavior. Retuning the controller gains can slow the response and give the impression of increased robustness, yet instability still occurs at delay values inconsistent with linear predictions. The discrepancy is therefore structural rather than tuning related.

Conclusion

The analysis demonstrates that the NDI–INDI control architecture performs well for GHAME when no sensor delay is present. Introducing unsynchronized delay causes the system to fail rapidly, confirming that misaligned measurement and actuation paths severely degrades the incremental control law formulation. Synchronizing the delay substantially increases the admissible delay margin and is therefore essential for reliable performance.

The delay study also exposed a deeper structural limitation in the lateral–directional axis. Even with synchronized delays, the lateral subsystem remains far more sensitive than the longitudinal subsystem. This arises from the intrinsic characteristics of GHAME: low roll inertia and very fast lateral dynamics leave insufficient separation between the control input and the airframe response. As a result, the system cannot approximate the chain of integrators required by INDI. Incremental inversion becomes only approximate, and linear stability margins lose predictive value. In contrast, the longitudinal axis exhibits adequate separation between control and dynamics, satisfies the INDI assumptions, and supports accurate inversion and stable control. Overall, the study shows that while NDI–INDI performs effectively for the longitudinal motion, particularly when sensor delays are synchronized and the lateral axis suffers from a fundamental structural limitation. In its current form, INDI should only be applied to GHAME's lateral–directional dynamics with caution. Only physical or implementation-level modifications, such as higher actuator bandwidth, faster sampling, or increased roll inertia, could establish the conditions required for reliable inversion-based control.

Appendix

Table 9 GHAME Geometric, Mass Properties, and WGS-84 Physical Constants (SI Units)

Parameter	Symbol	Constant	Takeoff	Fuel Burnout	Unit
Reference area	S	557.42	–	–	m^2
Reference chord	c	22.86	–	–	m
Reference span	b	24.38	–	–	m
Mass	m	–	136,080	54,432	kg
Moment of inertia	I_{xx}	–	1.573×10^6	1.180×10^6	$kg{\cdot}m^2$
Moment of inertia	I_{yy}	–	31.60×10^6	19.25×10^6	$kg{\cdot}m^2$
Moment of inertia	I_{zz}	–	32.54×10^6	20.20×10^6	$kg{\cdot}m^2$
Product of inertia	I_{xz}	–	0.380×10^6	0.240×10^6	$kg{\cdot}m^2$
Earth gravitational const.	GM	3.986005×10^{14}	–	–	m^3/s^2
Mean Earth radius	R_0	6.370987308×10^6	–	–	m
Earth rotation rate	$\omega_\oplus$	7.292115×10^{-5}	–	–	rad/s
Stand. gravity at sea level	g_0	9.80675445	–	–	m/s^2

References

[1] Schmidt, D., and Velapoldi, J., *Flight Dynamics and Feedback Guidance Issues for Hypersonic Air-Breathing Vehicles*, American Institute of Aeronautics and Astronautics Inc., 1999. https://doi.org/10.2514/6.1999-4122.

[2] Gilbert, M. G., Heeg, J., and Pototzky, A. S., "The application of active controls technology to a generic hypersonic aircraft configuration," Tech. Rep. NASA-TM-101689, NASA, 1990. URL https://ntrs.nasa.gov/citations/19900010755.

[3] Smith, P., "A Simplified Approach to Nonlinear Dynamic Inversion Based Flight Control," *23rd Atmospheric Flight Mechanics Conference*, AIAA, 1998, pp. 762–770. https://doi.org/10.2514/6.1998-4461, aIAA Paper 1998-4461.

[4] Bowers, A. H., et al., "A Generic Hypersonic Aerodynamic Model Example (GHAME)," Technical report, NASA Dryden Flight Research Facility, 1989.

[5] White, D. A., Bowers, A., Iliff, K., and Menousek, J., "Flight, Propulsion, and Thermal Control of Advanced Aircraft and Hypersonic Vehicles," *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, Multiscience Press, Inc., New York, NY, 1992, pp. 357–465.

[6] Zipfel, P. H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, 3rd ed., AIAA, Reston, Virginia, 2014. https://doi.org/https://doi.org/10.2514/4.102509.

[7] Goz, E., "Robust Multi-Objective H-Infinity Control of a Generic Hypersonic Vehicle," Master's thesis, Delft University of Technology, 2024.

[8] Autenrieb, J., "Data Fusion-Based Incremental Nonlinear Model Following Control Design for a Hypersonic Waverider Configuration," *AIAA SciTech 2023 Forum*, American Institute of Aeronautics and Astronautics, 2023. https://doi.org/10.2514/6.2023-1997.

[9] Pfeifle, O., and Fichter, W., "Cascaded Incremental Nonlinear Dynamic Inversion for Three-Dimensional Spline-Tracking with Wind Compensation," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 8, 2021, pp. 1559–1571.

[10] Juliana, S., Chu, Q. P., Mulder, J. A., and van Baten, T. J., "Flight Control of Atmospheric Re-entry Vehicle with Nonlinear Dynamic Inversion," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004. https://doi.org/https://doi.org/10.2514/6.2004-5330.

[11] Smeur, E. J. J., de Croon, G. C. H. E., and Chu, Q., "Cascaded Incremental Nonlinear Dynamic Inversion for MAV Disturbance Rejection," *Control Engineering Practice*, Vol. 73, 2018, pp. 79–90. https://doi.org/10.1016/j.conengprac.2018.01.003.

[12] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., John Wiley & Sons, Hoboken, NJ, 2016.

[13] Mooij, E., *Re-entry Systems*, Springer, 2024. https://doi.org/10.1007/978-3-031-62174-1.

[14] Bacon, B., Ostroff, A., and Joshi, S., "Reconfigurable NDI controller using inertial sensor failure detection and isolation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 4, 2001, pp. 1373–1383. https://doi.org/https://doi.org/10.1109/7.976972.

[15] Bacon, B., Ostroff, A., and Joshi, S., "Nonlinear dynamic inversion reconfigurable controller utilizing a fault tolerant accelerometer," *19th Digital Avionics Systems Conference*, Vol. 2, 2000. https://doi.org/https://doi.org/10.1109/DASC.2000.884920.

[16] Mooij, E., *The Motion of a Vehicle in a Planetary Atmosphere*, Delft University Press, The Netherlands, 1997.

[17] Snell, S. A., Enns, D. F., and Garrard, W. L., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 976–984. https://doi.org/https://doi.org/10.2514/3.20932.

[18] Smeur, E. J. J., Chu, Q. P., and de Croon, G. C. H. E., "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 450–461. https://doi.org/https://doi.org/10.2514/1.G001490.

[19] Schumacher, C., and Khargonekar, P. P., "Stability Analysis of a Missile Control System with a Dynamic Inversion Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 3, 1998, pp. 508–515.

[20] Naidu, D. S., and Calise, A. J., "Singular Perturbations and Time Scales in Guidance and Control of Aerospace Systems: A Survey," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1057–1078.

[21] Steffensen, R., Steinert, A., and Smeur, E. J. J., "Nonlinear Dynamic Inversion with Actuator Dynamics: An Incremental Control Perspective," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 4, 2022, pp. 709–717. https://doi.org/https://doi.org/10.2514/1.G006813.

[22] Nise, N. S., *Control Systems Engineering*, 7th ed., Wiley, Hoboken, NJ, 2015.