Robust Multi-Objective H∞ Control of GHAME Hypersonic Vehicle in Subsonic Flight

Goz, E.; Theodoulis, S.T.

DOI
10.2514/6.2025-2266

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2025 Forum

AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-2266

Robust Multi-Objective $\mathcal{H}_\infty$ Control of GHAME Hypersonic Vehicle in Subsonic Flight

E. Goz* and S. Theodoulis†
Delft University of Technology, Delft, 2628CD, Netherlands

This research is aimed at developing a comprehensive approach for robust hypersonic vehicle (HV) control utilizing modern $\mathcal{H}_\infty$ techniques. Initial focus is placed on subsonic flight condition to validate the framework and controller design in a relatively familiar field, for which the HV are not primarily optimized. A 6-degree-of-freedom non-linear model of the GHAME hypersonic vehicle was constructed in MATLAB/Simulink, incorporating tensor-based equations of motion and embedded parametric uncertainty in the aerodynamic coefficients. The linear short-period longitudinal dynamics were then extracted at multiple operating points. A controller of fixed structure was synthesized using multi-objective (multidisk) $\mathcal{H}_\infty$ mixed-sensitivity techniques with various performance and robustness requirements covering the pitch moment coefficient parametric uncertainty domain. Additionally, the design is extended to handle variations in Mach number, altitude, and fuel mass around the trim point using a multi-model approach. A single, structured control system successfully stabilized, rejected input and output disturbances and provided reference tracking for the uncertain short-period models and met the robustness margin requirements for the entire grid. It was then tested on the non-linear model and successfully performed the same tasks under parameter variations across the flight point grid.

I. Nomenclature

α	=	Angle of attack
β	=	Angle of sideslip
δ_a	=	Aileron deflection
δ_e	=	Elevator deflection
δ_r	=	Rudder deflection
δ_t	=	Throttle tab deflection
δ_{vl}	=	Left elevon deflection
δ_{vr}	=	Right elevon deflection
γ	=	Performance tuning metric
ρ	=	Air density
σ	=	Singular value
a	=	Speed of sound
A_c	=	Engine cowl area
C_a	=	Capture area ratio
C_D	=	Drag force coefficient
C_{D_1}	=	Partial drag force coefficient
C_L	=	Lift force coefficient
C_{L_1}	=	Partial lift force coefficient
C_l	=	Roll moment coefficient
C_m	=	Pitch moment coefficient
C_{m_1}	=	Partial pitch moment coefficient
C_n	=	Yaw moment coefficient
C_X	=	Longitudinal force coefficient
C_Y	=	Side force coefficient
C_Z	=	Normal force coefficient
f_P	=	Thrust force
g_0	=	Gravitational constant
M	=	Mach number
n_z	=	Vertical load factor at IMU location
$\bar{q}$	=	Dynamic pressure

II. Introduction

Hypersonic vehicles (HV), capable of traveling at speeds and altitudes unreachable by conventional supersonic aircraft, represent a significant advancement in aerospace technology, and require a careful consideration of challenges associated with market, operational infrastructure, and, of course, engineering [1]. Their extraordinary speed capabilities enable drastic reductions in travel time, presenting potential applications for high-priority point-to-point

*MSc, student, Control & Simulation division, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, Netherlands; eggoz55@gmail.com.
†Associate Professor, Control & Simulation division, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, Netherlands; S.Theodoulis@tudelft.nl. Associate Fellow AIAA.

transport and emergency response scenarios where time is a critical factor. Furthermore, the high altitudes achieved by these vehicles not only facilitate reconnaissance missions above conventional flight levels, but also extend beyond the Earth's atmosphere. This capability positions hypersonic vehicles as promising candidates for reduced-cost space access, functioning as single-stage-to-orbit (SSTO) reusable vehicles [2, 3]. Such a development could revolutionize space utilization for the Low Earth Orbit operations. Although HV designs vary depending on mission profiles, such as rocket-propelled space launchers [1], in the context of this paper the term HV specifically refers to the horizontal take-off and landing air-breathing hypersonic transport vehicle, which is able to sustain cruise speed above Mach 5 within the Earth's atmosphere.

In order to reach these operating conditions, some unique design characteristics must be employed. In general, the HV tend to have long, slender bodies, where the propulsion system is tightly integrated into the airframe, with lightweight yet strong structure able to withstand aero-thermodynamic loads [1, 2]. The forward-extended fuselage is needed to create a series of shock-waves that compress the airflow fed into the engine and to create lift force from underneath, whereas the rear fuselage is shaped for expanding the exhaust gases externally, hence also contributing to the lift force. The propulsion system is typically mounted underneath, below the center of gravity (CG), producing a pith-up moment [2].

Among many engineering challenges, the HV operating conditions and design traits introduce significant problems related to their stability and control. The vibrations of the fuselage change the pressure distributions across the airframe, resulting in lift, drag, pitching moment, and intake airflow perturbations [2, 4]. Due to the airframe structure, the vibrations tend to be of relatively low frequencies posing elastic-body mode interactions, whereas the changes in the angle of attack during pitch-up maneuvers further alter the flow field at the inlet and can result in variations in thrust vector magnitude and direction [5], or even in engine flameout altogether [6]. In fact, the airframe-propulsion interactions are argued in [7] to possibly be the most complex of any vehicle. All of the aforementioned problems introduce uncertainties into the system, along with a lack of high-fidelity models of accurate aerodynamic data (at least among those available to the public). Therefore, the flight control system (FCS) designs for HV must account for the unmodelled nonlinearities, as well as consider specific operational requirements such as atmospheric turbulence rejection [7], angle of attack variation limitation [6, 8], etc.

The discussed aeroelastic effects have been the primary concern in almost all HV-related works referenced in this paper, so the modelling and control design approaches must be carefully considered. There are only a handful of HV aerodynamic models available to the general public for research. Among the widely spread ones are the Winged-Cone Configuration Hypersonic Vehicle [9] and the Generic Hypersonic Vehicle Model Example (GHAME) [10]. Both are rigid-body, 6 degrees-of-freedom (DoF), and developed in the 1990's. However, the former is developed completely via analytical computer modelling programs, whereas GHAME is a combination of analytical and real empirical data. There is another model developed later in 2005 [11, 12], which aims to capture the interactions between the airframe, propulsion system, and aerodynamics, although in longitudinal plane only. The model was derived from first principles, and incorporates structural bending into the equations of motion. It was then incorporated into Simulink® framework, for which the initial flight control system was designed with Linear Quadratic Regulator (LQR) [13, 14]. However, the uncertainties in the original model were not quantified [2], and the LQR does not guarantee any robustness margins at the plant output. The focus switched later to make the developed LQR controller robust to uncertainties using the servomechanism theory in [15], but the unmodelled dynamics were considered implicitly as parameter variations with changes of fuel mass. On the other hand, the $\mathcal{H}_\infty$ robust control framework allows to address the uncertainties directly, and guarantees robustness margins.

One of the early attempts of implementing $\mathcal{H}_\infty$ mixed-sensitivity method on HV using basic S/T structure, described in [16], concluded that the real parameter uncertainties were too large to handle by this method and switched to $\mathcal{H}_\infty$ μ-synthesis method instead, which treats the worst-case uncertainty scenario. In fact, μ-synthesis was mostly used in the earlier designs of robust $\mathcal{H}_\infty$ controllers for HV. In [7] it was also concluded that $\mathcal{H}_\infty$ controllers suffer performance degradation with introduction of simultaneous uncertainties into the system, and called for μ-synthesis as a better performing control design methodology for this application. A major drawback of μ-synthesis is that it leads to controllers of very high order, which is computationally inefficient and complicates follow-up gain-scheduled designs. Additionally, it cannot address real parametric uncertainty directly and rather treats it as complex, introducing conservatism in the solution. Further work on HV control in [17–19], where the aeroelastic effects are primarily modeled as parametric uncertainty in C_m partial coefficients, concluded that the controller order reduction does not provide any robustness guarantees, whereas synthesizing a fixed-order controller still results in the orders of 5 to 9. Ref. [5] further outlines the aforementioned problems, stating the need for fixed-structure robust controllers of reduced conservatism, whereas μ-synthesis treats real parameter uncertainty as complex. It also stated that it is crucial to attenuate the effects

of atmospheric turbulence in hypersonic flight, for which an additional $\mathcal{H}_2$ constraint is required. This results in a problem of mixed $\mathcal{H}_2/\mathcal{H}_\infty$, which had no solution in the 1990's even in a full-order variant [5]. The public focus then shifted more towards nonlinear and adaptive control methods for HV [20, 21].

Fortunately, all of the aforementioned problems associated with $\mathcal{H}_\infty$ controllers have been solved later in [22, 23] using non-smooth optimization algorithms to synthesize controllers of predefined fixed structure. This led to multi-objective and multi-model approaches (known as multidisk problem) [24], making it possible to synthesize a single controller for a variety of performance specifications simultaneously, as well as for multiple plant models and directly specified parametric uncertainty. The latter is of special interest, as it presents a direct competitor to μ-synthesis. The multidisk solution not only outperforms μ-synthesis because of controller structure specification ability, but also because it addresses real parametric uncertainty directly, reducing conservatism [25]. The new methods were incorporated into functions *hinfstruct* and *systune* in the MATLAB® Robust Control Toolbox [26] in 2010 and 2012, respectively, whereas solution for parametric uncertainty case with multidisk method was incorporated in 2015 [22]. The main difference between the two is that the former uses a joint performance metric for all constraints like the classical mixed-sensitivity, whereas the latter fully utilizes multidisk capabilities and puts a performance measurement on each constraint individually, thereby making the controller less conservative and the constraining filter selection more convenient. Surprisingly, there is an evident lack of application of either of these methods to HV control. The few recent (after 2015) works on $\mathcal{H}_\infty$ control of HV still use the basic techniques under a single performance metric that lead to controllers of full order, such as in [27–30].

The field of hypersonic vehicle control is relatively new and under-researched, partly due to the scarcity of experimental data and the complexity of the operating conditions. At this point, most focus is aimed at modelling or accounting for the highly nonlinear aeroelastic effects at hypersonic speeds. The models available to the public are of low fidelity for high Mach numbers, and are either completely mathematical, or approximations from methods like DATCOM [31] combined with flight data of other aircraft. Most of the implemented $\mathcal{H}_\infty$ designs are simply not in line with the modern tools, which are able to provide flexible frameworks and satisfy various control design requirements in time and frequency domains. There is still much to be done for the HV flight control system development. One example of a relatively untreated area is the hypersonic airframe behavior in subsonic flight. The HV are not primarily designed for it, with long slender airframes and typically limited control surfaces, making them potentially challenging to optimize for robust performance within that flight regime. An adaptive nonlinear control method has been used for that purpose in [32], but the implementation of any other methods was not found during the literature survey.

This research establishes a foundation for future investigation of hypersonic vehicle control at TU Delft using modern $\mathcal{H}_\infty$ robust control design methods and tools. The scope consists of constructing a 6-degree-of-freedom (DoF) nonlinear model of the GHAME vehicle in MATLAB [33] and Simulink [26, 34] using tensor-based equations of motion with a flexible subsystem-separated block structure and embedded parametric uncertainty, setting up trimming and linearization programs that produce separated longitudinal and lateral linear models in the form of uncertain state-space systems at any given flight point and condition, synthesizing a flight control system of fixed structure at one of the flight points using multi-objective (multidisk) $\mathcal{H}_\infty$ mixed sensitivity techniques to explicitly design for robustness against structured parametric uncertainty. The design is further extended to be robust against trim condition variations in airspeed, altitude, and fuel mass using a multi-modelling approach. A model grid is formed around the nominal flight point (FP), and a single controller is synthesized for all parametrically uncertain models in the grid simultaneously, and then implemented and tested on the nonlinear model. As a first initial step, the focus is put on the short-period longitudinal dynamics at a subsonic flight condition. It seems as an appropriate starting point to test the framework and the controller design in an untreated field.

This paper is structured in three consecutive phases: modelling in Sec. III, FCS design in Sec. IV, and nonlinear implementation and simulation in Sec. V. The modelling phase consists of GHAME HV aerodynamic model description and analysis in Sec. III.A, its nonlinear tensor-based model description in Sec. III.B, and trimming and linearization process outlined in Sec. III.C. The FCS design includes a brief theoretical background in Sec. IV.A, the FCS structure and synthesis description in Sec. IV.B, and the results and analysis discussion in Sec. IV.C. Finally, conclusion and recommendations for future work are presented in Sec. VI.

III. Modelling

A. The GHAME model

1. General description

The hypersonic vehicle model selected is the NASA's Generic Hypersonic Aerodynamic Model Example developed at the Dryden Flight Research Facility. The original publication of the model with full description can be found in [10]. GHAME was specifically developed to provide realistic data of hypersonic flight that is unclassified and available to the general public for performance calculations, trajectory optimization, simulation, and control design purposes. The main two arguments for selecting this vehicle model are the fact that its aerodynamic data are based on a combination of realistic data from existing aircraft and theoretical data of a double-delta wing configuration, and the fact that it was directly available in FORTRAN code and extensively discussed in [35]. Additionally, it is 6-DoF and therefore allows future work with lateral motion.

GHAME is a horizontal take-off and landing SSTO vehicle with a gross take-off weight of 300,000 pounds (≈ 136,080 kg) and a dry weight of 120,000 pounds (≈ 54,432 kg). Its mass and size properties were based on the XB-70, and the moments of inertia were estimated from simplified geometry. The airframe features a 70° delta-wing configuration with a single vertical tail rudder and two elevons. The vehicle is 243 feet (≈ 74.07 m) long with a span of 80 feet (≈ 24.38 m). Its configuration is built from simple geometrical shapes and excludes landing gear and any variable geometry apart from the aforementioned control surfaces. The wing area is 6,000 square feet (≈ 557.42 m^2). The fuselage consists of a 10° half-angle cone ending in a cylinder with a 20-foot (≈ 6.1 m) diameter, terminating in an integrated boattail/nozzle. The schematics are shown in Fig.1.

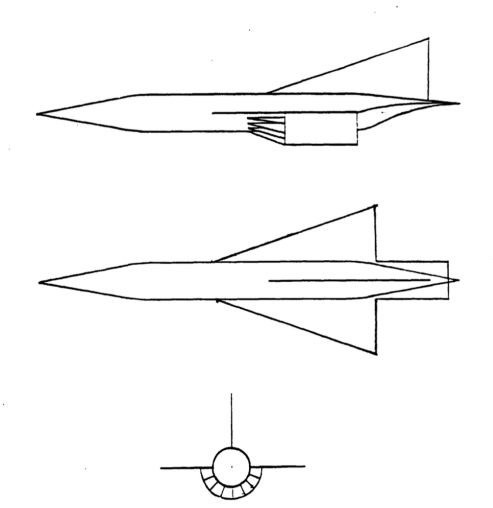


Fig. 1 Schematic of GHAME configuration [36]

2. Aerodynamic model

As has been mentioned, the aerodynamic model was comprised of analytical and empirical data. For lower Mach numbers, it is a combination of a swept double-delta wing using modified Newtonian Impact Flow method and an actual flight test data from the Space Shuttle. Above Mach 8, the data are exclusively based on the latter, but properly scaled. The force and moment coefficient equations have been linearized around a range of angle of attack α numbers for a range of Mach numbers at zero sideslip angle β. The force coefficient equations are presented in Eq. 1, and the moment coefficient equations in Eq. 2 [35].

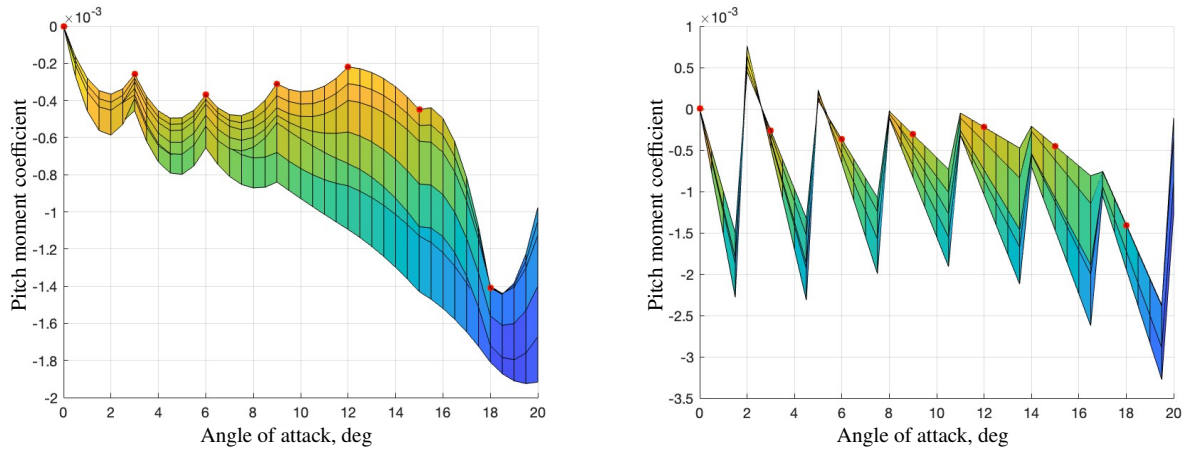
$$
\begin{aligned}
C_L &= C_{L_0}(M,\alpha) + C_{L_\alpha}(M,\alpha)\alpha + C_{L_{\delta_e}}(M,\alpha)\delta_e \\
C_D &= C_{D_0}(M,\alpha) + C_{D_\alpha}(M,\alpha)\alpha \\
C_Y &= C_{Y_\beta}(M,\alpha)\beta + C_{Y_{\delta_a}}(M,\alpha)\delta_a + C_{Y_{\delta_r}}(M,\alpha)\delta_r
\end{aligned} \tag{1}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

$$
\begin{aligned}
C_m &= C_{m_0}(M,\alpha) + C_{m_\alpha}(M,\alpha)\alpha + C_{m_{\delta_e}}(M,\alpha)\delta_e + C_{m_q}(M,\alpha)\frac{qc}{2V} \\
C_l &= C_{l_\beta}(M,\alpha)\beta + C_{l_{\delta_a}}(M,\alpha)\delta_a + C_{l_{\delta_r}}(M,\alpha)\delta_r + C_{l_p}(M,\alpha)\frac{pb}{2V} + C_{l_r}(M,\alpha)\frac{rb}{2V} \\
C_n &= C_{n_\beta}(M,\alpha)\beta + C_{n_{\delta_a}}(M,\alpha)\delta_a + C_{n_{\delta_r}}(M,\alpha)\delta_r + C_{n_p}(M,\alpha)\frac{pb}{2V} + C_{n_r}(M,\alpha)\frac{rb}{2V}
\end{aligned} \tag{2}
$$

All partial coefficients wrt angles have units of 1/°, and all partial coefficients wrt rotation rates have units of 1/rad. Lift coefficient C_L is defined positive upwards and perpendicular to velocity vector, drag coefficient C_D is positive aft and parallel to velocity vector. The rest of the coefficients have conventional positive directions. Each of the partial coefficients is given in the form of a look-up table of 9 α rows and 13 Mach number columns. Their indices, respectively, are shown in Eq. 3.

$$
\begin{aligned}
\alpha &= [-3 \quad 0 \quad 3 \quad 6 \quad 9 \quad 12 \quad 15 \quad 18 \quad 21] \\
M &= [0.4 \quad 0.6 \quad 0.8 \quad 0.9 \quad 0.95 \quad 1.05 \quad 1.2 \quad 1.5 \quad 2.0 \quad 3.0 \quad 6.0 \quad 12.0 \quad 24.0]
\end{aligned} \tag{3}
$$

Jumping a little ahead, a fundamental discrepancy in the aerodynamic data was discovered during the implementation of this aerodynamic model in the simulation. The solution process led to a certain alternative adaptation of the partial coefficients, so it makes more sense do discuss it in this subsection. The problem only concerns the first two terms in the computations of C_L, C_D, and especially C_m, essentially at subsonic Mach numbers. Since its impact on C_m is more severe than on the first two, the following discussion is explained on C_m.



(a) Pitch moment coefficient computed with interpolation for M = 0.4-1.0

(b) Pitch moment coefficient as a set of linear functions for M = 0.4-1.0

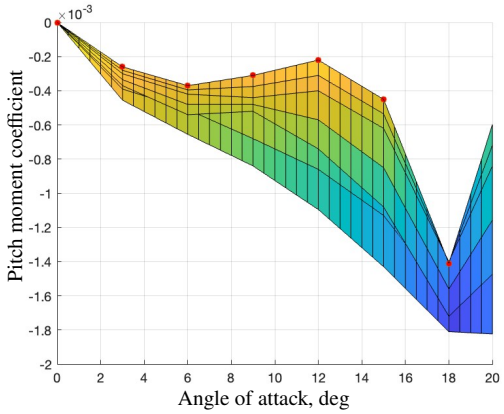
Fig. 2 Comparison of pitch moment coefficient computation using various methods

Naturally, the aerodynamic model with look-up tables means interpolating each of the partial coefficients for active (i.e., currently set) α and M values in the simulation. However, it was directly stated by the authors in [10] that the equations have been linearized around α values. When linearizing the entire coefficient equations wrt α at equilibrium at some Mach number, it is the first two terms that contribute to it. Plotting $C_m = C_{m_0}(\alpha) + C_{m_\alpha}(\alpha)\alpha$, where the terms

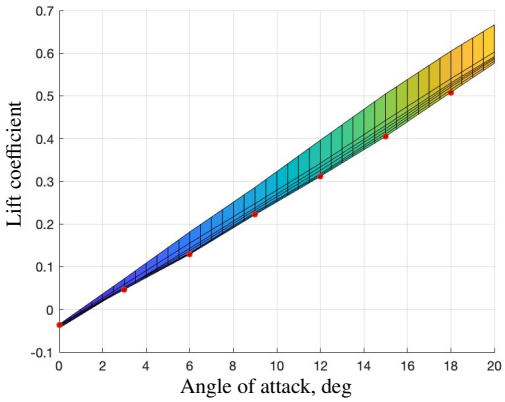
are interpolated at (and multiplied with) respective α first and then summed, for a set of Mach numbers between 0.4 and 1.0, leads to the graph shown in Fig. 2a. The red dots are placed at intersections with α indices along $M = 0.4$ for better comprehension.

This unnatural pattern contradicts the values of C_{m_α} look-up table, which has all entries negative, whereas here the slope becomes positive just before the next α index. As an even better indication of data self-contradiction, note that the red dots at the indices show a positive slope wrt α, whereas the C_{m_α} table states that the slope is always negative. The phenomena of clearly positive slope between the indices only happens at Mach < 1. There is a possible explanation to this discrepancy. The coefficient equations have already been linearized in the aerodynamic tables, so $C_m = C_{m_0} + C_{m_\alpha}\alpha$ can be perceived as a basic linear function $y = ax + b$. Therefore, the coefficients $C_{m_0}(\alpha)$ and $C_{m_\alpha}(\alpha)$ at some fixed Mach number should not be treated as continuous functions of alpha, but rather as a set of "scheduled" linear functions $y = ax + b$ at discreet α values. It is best illustrated in Fig. 2b for the same set of Mach numbers.

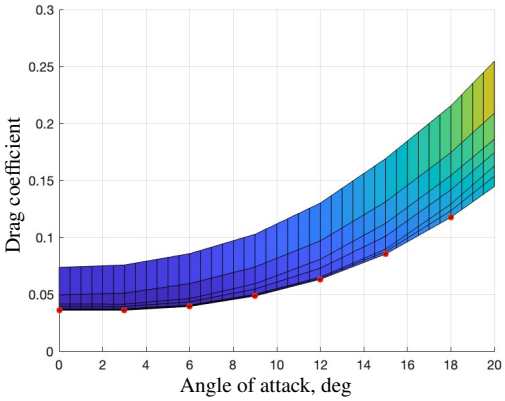
In this interpretation, the closest linear function at some active α is selected and then its deviation from the index is multiplied with C_{m_α}. Although a linear pitching moment coefficient slope may be a useful simplification around the indices, the "saw" shape is not suitable for full scale simulation, either. As a result, it was decided to compute the $C_m = C_{m_0} + C_{m_\alpha}\alpha$ at all α indices first, and then interpolate with straight lines between the indices for the entire Mach range, producing a new aerodynamic table. The same procedure was applied to C_L and C_D, since they have the same structure. The new partial coefficients are called C_{m_1}, C_{L_1}, C_{D_1}, and are defined as a summation of the first two corresponding aerodynamic tables at the indices (i, j), as shown in Eq. 4 and plotted in Fig. 3 (red dots again at M = 0.4 for comprehension). The new partial coefficients are then substituted into their respective formulas previously shown in Eq. 1 and Eq. 2, and they are the ones to be interpolated at the active Mach and α.



(a) Pitch moment coefficient C_{m_1}



(b) Lift coefficient C_{L_1}



(c) Drag coefficient C_{D_1}

Fig. 3 Modified lift C_{L_1}, drag C_{D_1}, and pitch moment C_{m_1} partial coefficients, plotted for Mach = 0.4-1.0

$$
\begin{aligned}
C_{m_1}(M,\alpha) &= C_{m_0}(M_i,\alpha_j) + C_{m_\alpha}(M_i,\alpha_j)\alpha_j \\
C_{L_1}(M,\alpha) &= C_{L_0}(M_i,\alpha_j) + C_{L_\alpha}(M_i,\alpha_j)\alpha_j \\
C_{D_1}(M,\alpha) &= C_{D_0}(M_i,\alpha_j) + C_{D_\alpha}(M_i,\alpha_j)\alpha_j
\end{aligned} \tag{4}
$$

3. Actuator model

The model features two elevons that function both as elevators and ailerons, located at the trailing edge of the wing. The conversions between them are shown in Eq. 5:

$$
\delta_e = \frac{\delta_{vl} + \delta_{vr}}{2}, \qquad \delta_a = \frac{\delta_{vl} - \delta_{vr}}{2} \tag{5}
$$

where δ_{vl} and δ_{vr} are, respectively, left and right elevon deflections, both defined positive downwards. There is also a single rudder, which is defined positive trailing edge left. There is no further information on the actuators in the original model, so it was assumed that limits on deflection angles and rates are intentionally left free to choose. Actuator modelling is further discussed in Sec. III.B.

4. Propulsion system model

Being not the primary focus of research, the propulsion system was intentionally simplified by the authors just to "fly" the aerodynamic model [10]. The generic engine model approximates a combined-cycle propulsion system, where a turbojet operates between Mach 0-2, ramjet operates between Mach 2-6, and supersonic combustion ramjet operates above Mach 6. The engine cycles are assumed to change automatically. The inlet has variable geometry and its size is scheduled wrt α and Mach. The thrust f_P is then computed with Eq. 6.

$$
f_P = 0.029\delta_t I_{sp}(M,\delta_t) g_0 \rho M a C_a(M,\alpha) A_c, \qquad \delta_t = [0-2] \tag{6}
$$

Where δ_t is throttle deflection tab, 0.029 comes from the fact that the pilot indirectly regulates the stoichiometric ratio with the throttle tab, a is the speed of sound, and A_c is engine cowl area factor constant. Specific impulse I_{sp} is dependent on the mass flow, so it is provided in the form of a look-up table for indices of Mach (same as in Eq. 3) and δ_t. The effective capture area C_a of the engine inlet is determined by the properties of bow shock wave under the vehicle fore-body, which is dependent on the angle of attack and dynamic pressure [4], and is therefore given as a look-up table for indices as from Eq. 3. Evidently, the thrust force is computed instantaneously, i.e., the propulsion system model has no dynamics. The thrust force is assumed to be aligned with the center of gravity, and parallel to the airframe x-axis.

B. Non-linear model

The modelling of GHAME was largely based on techniques discussed in [35] and its model written in FORTRAN code, which was built for purposes of full simulation with elliptical Earth equations of motion. The model of GHAME presented in this paper, however, has its own unique traits and assumptions, which are generally outlined below:

1) The model is 6-DoF, built in MATLAB and Simulink environment, with subsystem-separated blocks mindset.
2) The EoM assume flat, non-rotating Earth, which is an understandable simplification since the model is built for control law development purposes.
3) The mass of the vehicle m and the moment of inertia (MOI) are assumed constant in the EoM, for the same reason as the previous assumption.
4) The model is tensor-based, using a mix of relations for an aircraft and a hypersonic vehicle, both described in ref. [35].
5) The center of gravity position is fixed, since its variation is not included explicitly in the original aerodynamic model.
6) The inertial measurement unit (IMU) is loosely approximated to be located 80 ft (24.384 m) ahead of CG along the center line, which is geometrically just before the conical nose in Fig. 1. It is merely an initial coarse reference value for load factor computation near pilot location, and can easily be changed later.
7) The measurement noise is not explicitly included in the initial model and will be added in the future.

The forces and moments computations directly utilize the aerodynamic coefficients derived in Sec. III.A, and have the following expressions in body coordinates, respectively:

$$[f_{a,p}]^B = \begin{bmatrix} f_{a,p_1} \\ f_{a,p_2} \\ f_{a,p_3} \end{bmatrix} = \begin{bmatrix} \overline{q}SC_X + f_P \\ \overline{q}SC_Y \\ \overline{q}SC_Z \end{bmatrix} \tag{7}$$

$$[m_B]^B = \begin{bmatrix} m_{B_1} \\ m_{B_2} \\ m_{B_3} \end{bmatrix} = \begin{bmatrix} \overline{q}SbC_l \\ \overline{q}ScC_m \\ \overline{q}SbC_n \end{bmatrix} \tag{8}$$

where f_P is propulsion force (thrust), and the body-frame coefficients C_X and C_Z are computed directly from C_L, C_D, and α using Eq. 9.

$$C_X = -C_D\cos(\alpha) + C_L\sin(\alpha), \qquad C_Z = -C_D\sin(\alpha) - C_L\cos(\alpha); \tag{9}$$

The equations of motion originate from the first principles of Newton's and Euler's laws. The translational equations are based on the Newton's second law in the inertial frame of reference I, but with the flat-Earth assumption the Earth frame of reference E becomes identical to I. The general equation of motion, expressed in body coordinate system B, is given in Eq. 10 in matrix form:

$$m\left[\frac{dv_B^E}{dt}\right]^B + m\left[\Omega^{BE}\right]^B\left[v_B^E\right]^B = \left[f_{a,p}\right]^B + m\,[T]^{BL}\,[g]^L \tag{10}$$

where v_B^E is the velocity vector of the vehicle's CG wrt inertial frame E, Ω^{BE} is the angular velocity vector of the vehicle wrt frame E, $[T]^{BL}$ is the direction cosine matrix (DCM) which is used to transform the gravity vector $[\bar{g}]^L = [0 \quad 0 \quad g]$ from local frame L to the frame B. In coordinate form the translational EoM become:

$$m\left\{\begin{bmatrix} du/dt \\ dv/dt \\ dw/dt \end{bmatrix}^B + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}^B \begin{bmatrix} u \\ v \\ w \end{bmatrix}^B\right\} = \begin{bmatrix} f_{a,p_1} \\ f_{a,p_2} \\ f_{a,p_3} \end{bmatrix}^B + \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}^{BL} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}^L \tag{11}$$

The rotational EoM are derived from Euler's law, and are worked out in a similar fashion as the translational ones, with the final general expression shown in Eq. 12:

$$\left[\frac{d\omega^{BE}}{dt}\right]^B = \left(\left[I_B^B\right]^B\right)^{-1}\left[-\left[\Omega^{BE}\right]^B\left(\left[I_B^B\right]^B\left[\omega^{BE}\right]^B\right) + [m_B]^B\right] \tag{12}$$

where ω^{BE} is the vehicle's angular velocity vector, $\left[I_B^B\right]^B$ is the vehicle's moment of inertia, and $[m_B]^B$ is from Eq. 8. Assuming symmetry in longitudinal plane, the MOI has $I_{13} = I_{31}$. Expanded matrices of Eq. 12 are shown in Eq. 13.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}^B = \left(\begin{bmatrix} I_{11} & 0 & I_{13} \\ 0 & I_{22} & 0 \\ I_{31} & 0 & I_{33} \end{bmatrix}^B\right)^{-1}\left(-\begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}^B \times \left(\begin{bmatrix} I_{11} & 0 & I_{13} \\ 0 & I_{22} & 0 \\ I_{31} & 0 & I_{33} \end{bmatrix}^B \begin{bmatrix} p \\ q \\ r \end{bmatrix}^B\right) + \begin{bmatrix} m_{B_1} \\ m_{B_2} \\ m_{B_3} \end{bmatrix}^B\right) \tag{13}$$

Next, the kinematic relations include familiar expressions for the angle of attack and the angle of sideslip:

$$\alpha = \arctan\left(\frac{w}{u}\right), \qquad \beta = \arcsin\left(\frac{v}{\sqrt{u^2+v^2+w^2}}\right) \tag{14}$$

The DCM introduced earlier in Eq. 10 is needed to compute the orientation of the vehicle wrt the E reference frame. It is derived with Euler angles $[\phi \quad \theta \quad \psi]$ using the angular velocities in the body reference frame $[p \quad q \quad r]$. The Euler angles method was chosen to compute the DCM because of simplicity of implementation, as it directly uses Euler angles as states in the program calculated from rotational rates, and then the DCM is easily calculated afterwards. The Euler

angles are updated with with Eq. 15, and the DCM is then calculated with Eq. 16. There is a singularity at $\theta = \pm 90°$, which can be avoided with simple program operations. Additionally, the HV are not high maneuverability vehicles, so the singularity region is not their usual operating condition.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \cos\theta & \cos\phi / \cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{15}$$

$$T^{BL} = \begin{bmatrix} \cos\psi \cos\theta & \sin\psi \cos\theta & -\sin\theta \\ \cos\psi \sin\theta \sin\phi - \sin\psi \cos\phi & \sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi & \cos\theta \sin\phi \\ \cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi & \sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi & \cos\theta \cos\phi \end{bmatrix} \tag{16}$$

The speed of sound a and air density ρ are computed using International Standard Atmosphere. The load factor at the CG is calculated as $n = [a_B^E]^B / g_0$ (acceleration/gravity), in body reference frame. However, it is desirable to measure the load factor at the pilot location, where additional forces due to centrifugal and angular acceleration take place. Therefore, the acceleration at the IMU location is calculated using the Grubin's form of Newton's second law, where all the terms are divided by common mass [35]:

$$[a_S^E]^B = [a_B^E]^B + \left[\Omega^{BE}\right]^B \left[\Omega^{BE}\right]^B [s_{SB}]^B + \left[\dot{\Omega}^{BE}\right]^B [s_{SB}]^B \tag{17}$$

where $[a_S^E]^B$ is the acceleration vector at IMU location, $[a_B^E]^B = [f_{a,p}]^B / m$ is the acceleration vector at CG, and $[s_{SB}]^B$ is IMU location vector relative to CG. The load factor at the pilot's location is then $[n_S^E]^B = [a_S^E]^B / g_0$, and the vertical load factor is $n_z = -[n_S^E]^B(3)$ to make it positive upwards.

The nonlinear model is initialized at a specific flight point using a separate file containing the initial conditions and airframe constants. The vehicle mass and MOI are specified for gross take-off weight and at burn-out (dry mass). The active mass and MOI at initial condition are regulated with fuel mass fraction, e.g., 0.5 of fuel tank corresponds to the median values of vehicle mass and MOI. The actuator model is taken directly from the FORTRAN version of GHAME [35] and put into Simulink. Both elevons and the rudder have exactly the same characteristics, and are modelled as second order systems. The initialization parameters and their values are outlined in Table 7.

C. Uncertainty, Trimming and Linearization

The objective is to retrieve a linear state-space model at some operating point with uncertainty in aerodynamic parameters. For that purpose, 2 additional Simulink models were created, one for the airframe excluding the actuators, and one for the propulsion system. The process can be outlined as follows: first, the uncertainty is incorporated into both models. Then, the airframe is trimmed for equilibrium, so that the actuator deflections and thrust that keep the airframe in desired state are computed. The throttle tab deflection is then trimmed in the propulsion model to produce the required thrust. Both models are then linearized at their operating points, producing a full linear uncertain state-space (uss) system with all states together. The system is then decoupled into lateral, longitudinal, and short-period dynamics.

1. Uncertainty implementation

The created framework allows the parametric uncertainty to be explicitly defined for all partial aerodynamic coefficients independently, and can be easily extended to include additional parameters. The real parametric uncertainty can be implemented in MATLAB using $ureal$ function, which takes the mean value and the uncertainty range as inputs. The latter can be specified using ± absolute value range or percentage variation, while the former is trickier. Unlike some constant parameter such as actuator natural frequency, the partial aerodynamic coefficients vary along the flight path, which means their values are computed in the simulation directly, and calculating their value at trim condition by hand is cumbersome. It is significant for the framework to be flexible, such that parameter variation would be specified at the start, and then it would be carried over to the uncertain state-space system directly and automatically. Therefore, the nominal values of the partial aerodynamic coefficients computations are left unchanged, and there are generally two ways to incorporate their variations into Simulink, depending on whether uncertainty is specified as percentage or absolute value range.

When specifying it as percentage, a separate set of $ureal$ parameters is created during initialization for all partial aerodynamic coefficients, with a mean value of 1 and the desired percentage, as in the example of uncertainty variable

for C_{m_q} at the top of Eq. 18. They are then added as uncertain state-space *uss* blocks in Simulink along the path of nominal parameter values, so that they are multiplied with a mean of 1 and vary by the percentage.

$$\begin{aligned} uCm\_q\_percent &= \text{ureal}('\text{uCm\_q}', 1, '\text{ Percentage}', 20) \\ uCm\_q\_range &= \text{ureal}('\text{uCm\_q}', 0, '\text{ PlusMinus}', 0.0001) \end{aligned} \tag{18}$$

When the uncertainty is desired to be in absolute value range format, the initialization is done in a similar way, but with a different specification as at the bottom of Eq. 18, for example. However, in this case, the pitch rate signal q is split in two in the Simulink model. The first branch is multiplied with the nominal computed value of C_{m_q}, while the other branch is multiplied with its twin uncertainty variable uCm_q_range with a mean of 0 and absolute value ± range. They are then added together to produce an uncertain partial aerodynamic coefficient with a range specified for uCm_q, while the 0-mean does not intervene into calculations.

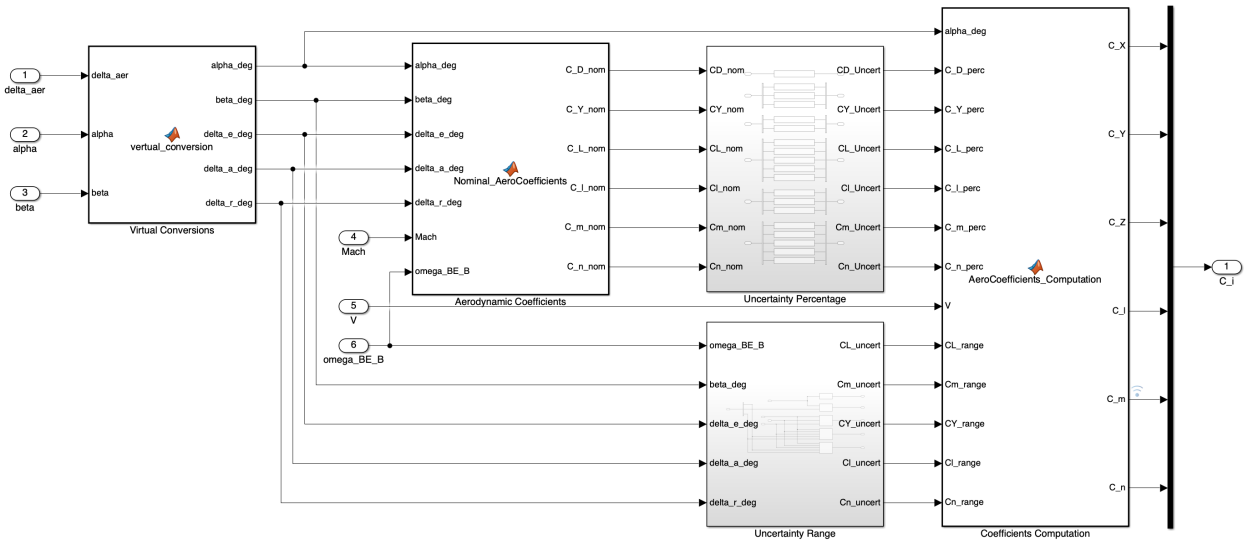


Fig. 4 Simulink model snippet of uncertainty implementation into aerodynamic coefficients

In this model, both variants of uncertainty were implemented into the framework. The schematic is shown in Fig. 4, where the *Uncertainty Percentage* block contains the percentage $ureal$ variables that are multiplied with their respective coefficients, whereas the *Uncertainty Range* block contains the absolute value range $ureal$ variables that are multiplied with their respective control deflections and rotational rates. The *Coefficients Computation* block is where all of them are added together. When one uncertainty variant is chosen, the other must be replaced by a real value of its mean, i.e., if percentage variation is chosen, then its twin variable uCm_q_range must be set to 0. Likewise, the uncertainty for some specific parameter can be "turned off" altogether by setting both of its respective uncertainty variables to their mean values. The uncertainty variant can be selected independently between parameters - one can be specified in percentage, and the other in range. However, the absolute value variation can not be implemented on parameters that are both continuously computed and not multiplied with some continuous signal. This is the case for C_{m_1}, C_{L_1}, and C_{D_1}, as well as engine parameters. These can only be specified in percentage.

For this initial design, the uncertainty was specified for C_{m_1} (which includes C_{m_0} and C_{m_α}), $C_{m_{\delta e}}$, and C_{m_q}. All three partial aerodynamic coefficients were arbitrarily chosen to vary by ±20%, while the uncertainty variables of the other coefficients were turned off.

2. Trimming

Before the model is linearized, it must be trimmed to some operating point. At that point, the control inputs are computed that would keep the system at steady-state (forces and moments are either 0 or constant). The appropriate selection of which states to keep constant, zero, or free depending on the equilibrium condition is described in [37].

The nominal operating point was selected at 0.75 Mach, 5000 m altitude, 0.5 fuel fraction of tank, in a steady, straight, wings-level flight. The trimming process was done via $findop$ function in MATLAB, which allows for code trimming specifications for the Simulink model. At this point, the inputs and outputs that would be used in the next steps must be selected. The uncertain airframe model has 4 inputs: thrust f_P, left δ_{vl} and right δ_{vr} elevon deflections, and rudder deflection δ_r. Note that these are actuator outputs, not the commanded inputs. The actuator models are added separately in the next step. The outputs of the uncertain airframe are the IMU-measured load factor and the rotational rates.

The states of the model and their trimming specifications are outlined for the nominal operating point in Table 1. The model is then trimmed automatically to find the necessary control inputs. The required thrust is then used to trim the propulsion system model in a similar fashion.

Table 1 Trimming Specifications for Model States and Inputs

Description	Known	Initial Value	Steady State	Min	Max
Roll rate p	Yes	0	Yes	$-\infty$	∞
Pitch rate q	Yes	0	Yes	$-\infty$	∞
Yaw rate r	Yes	0	Yes	$-\infty$	∞
Roll angle ϕ	Yes	0	Yes	$-90°$	$90°$
Pitch angle θ	No	$3°$	Yes	$-90°$	$90°$
Yaw angle ψ	No	0	Yes	$-360°$	$360°$
Airspeed V	Yes	240.4 m/s	Yes	0	∞
Angle of attack α	No	$3°$	Yes	0	$21°$
Sideslip angle β	No	0	Yes	$-\infty$	∞
Position x	No	0	No	$-\infty$	∞
Position y	No	0	Yes	$-\infty$	∞
Position z	Yes	-5000 m	Yes	$-\infty$	0

3. Linearization

The uncertain airframe model in Simulink can then be conveniently linearized around the selected operating point using the function $ulinearize$. The advantage of this method is that it recognizes the uss blocks with specified uncertainty in the model, and directly produces an uncertain state space model with embedded uncertain parameters specified earlier.

The computed full linear state-space model of the airframe has the following properties:

$$\begin{aligned} \text{Inputs} &= [f_P \quad \delta_{vl} \quad \delta_{vr} \quad \delta_r] \\ \text{States} &= [p \quad q \quad r \quad \phi \quad \theta \quad \psi \quad V \quad \alpha \quad \beta \quad z^E] \\ \text{Outputs} &= [p \quad q \quad r \quad n_x \quad n_y \quad n_z] \end{aligned}$$

The objective is to retrieve the longitudinal short-period model with virtual elevator input δ_e from the full model. The selected states are angle of attack α and pitch rate q, and the regulated outputs for the control system are the vertical load factor n_z measured by IMU and the pitch rate q. The short-period state-space model is retrieved by selecting appropriate rows and columns from the full state-space model matrices corresponding to the desired states and outputs. The two elevon inputs are converted to a single elevator input δ_e by simply adding the two elevons together in the rows corresponding to the longitudinal motion. The reason why the Eq. 5 is not used here is because the two elevons are represented separately with identical values for longitudinal states and outputs in the full linear model, so in this case δ_e action is divided equally between them.

The resulting short-period state-space model is represented by Eq. 19 with respective stability derivatives, according to [37]. The lateral and full longitudinal models were obtained in a similar way, but any further discussion on them is outside the scope of this research.

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -Z_\alpha/V & 1 \\ M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -Z_{\delta_e}/V \\ M_s \end{bmatrix} \delta_e = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$
$$\begin{bmatrix} n_z \\ \alpha \end{bmatrix} = \begin{bmatrix} -A_\alpha/g & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -A_{\delta_e}/g \\ 0 \end{bmatrix} \delta_e = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{19}$$

The stability derivatives were verified using Jacobian linearization on the EoM analytically. The computed nominal short-period system matrices are presented in Eq. 32 in the Appendix.

The nominal short-period model input-output pole-zero map is shown in Fig. 5. It has a non-minimum-phase (NMP) zero in the right-half-plane for vertical load factor, which is consistent with this vehicle type since the elevons are located aft the vehicle's CG. Small frequencies indicate that the system is relatively slow, and real negative pole locations indicate that it is open-loop stable and critically damped. The short-period model is in the form of uncertain state-space system, its properties and operating point conditions are summarized in Table 2.

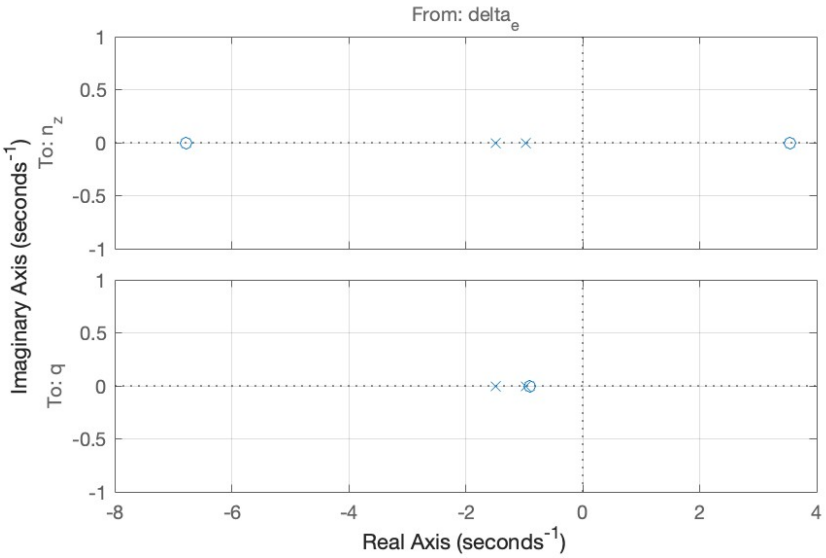


Fig. 5 Short-period input-output pole-zero map

Table 2 Short-period nominal linear system properties

Trim condition property	Value	Linear system property	Value
Mach number	0.75	Pole 1 λ_1	-0.965
Altitude	5000 m	Pole 2 λ_2	-1.48
Fuel fraction	0.5	NMP zero z_{nmp}	3.55
Angle of attack α	3.89°	Damping ratio ζ	1.0
Pitch angle θ	3.88°	Uncertainty C_{m_1}	±20%
Elevator deflection δ_e	-1.925°	Uncertainty $C_{m_{\delta_e}}$	±20%
		Uncertainty C_{m_q}	±20%

The virtual elevator δ_e is modelled as a 2-nd order linear system with the same natural frequency and damping ratio as for the elevons, presented in Table 7.

4. Model grid

To account for variations around the nominal flight condition, additional models were trimmed and linearized at specified deviations from the trim point. As mentioned earlier, the variations include fuel mass, Mach number, and altitude. The fuel mass is measured as a fuel fraction that is present in the tank, effectively varying vehicle mass and MOI. The variations around the nominal flight point are summarized in Table 3.

Table 3 Nominal flight point and variations around it

Property	Nominal FP value	Variation
Fuel fraction	0.5	0.25-0.75
Mach number	0.75	0.70-0.80
Altitude	5000 m	4500-5500 m

Taking into account the fuel fraction variation at the nominal trim point, a total grid of 11 models was assembled. All of them are short-period *uss* models and include parametric uncertainty in C_m specified earlier. The produced models and their properties are outlined in Table 4, where the first model corresponds to the nominal flight point

Table 4 Full short-period uncertain linear model grid

Model number	Model code name	Fuel fraction	Mach number	Altitude
1	G_sp_05_75_50	0.5	0.75	5000 m
2	G_sp_025_75_50	0.25	0.75	5000 m
3	G_sp_075_75_50	0.75	0.7	5000 m
4	G_sp_025_70_45	0.25	0.7	4500 m
5	G_sp_025_70_55	0.25	0.7	5500 m
6	G_sp_025_80_45	0.25	0.8	4500 m
7	G_sp_025_80_55	0.25	0.8	5500 m
8	G_sp_075_70_45	0.75	0.7	4500 m
9	G_sp_075_70_55	0.75	0.7	5500 m
10	G_sp_075_80_45	0.75	0.8	4500 m
11	G_sp_075_80_55	0.75	0.8	5500 m

IV. Robust flight control system design

A. H_∞ mixed-sensitivity theory

Control system performance and robustness to uncertainty specifications can be addressed with closed-loop (CL) transfer functions within the system. Consider a typical multiple-input-multiple-output (MIMO) control system shown in Fig. 6 with plant G, controller K, reference signal r, controller input signal e, controller output signal u, plant input signal u_p, plant output y_p, system output y, input and output disturbances d_I and d_O, and measurement noise n. If the system is closed-loop stable, then the fundamental relations between the signals can be derived as in Eq. 20.

$$\begin{bmatrix} e \\ y \\ u \\ u_p \end{bmatrix} = \begin{bmatrix} S_O & -S_O & -S_O & T_O \\ T_O & S_O & S_O & -T_O \\ KS_O & -T_I & -KS_O & -KS_O \\ KS_O & 0 & S_I & -KS_O \end{bmatrix} \begin{bmatrix} r \\ d_I \\ d_O \\ n \end{bmatrix} \tag{20}$$

where S_I and S_O are input and output sensitivity transfer functions, and T_I and T_O are input and output complementary sensitivity transfer functions, respectively, with following relations: $S_I + T_I = I$, $S_O + T_O = I$. Basic performance

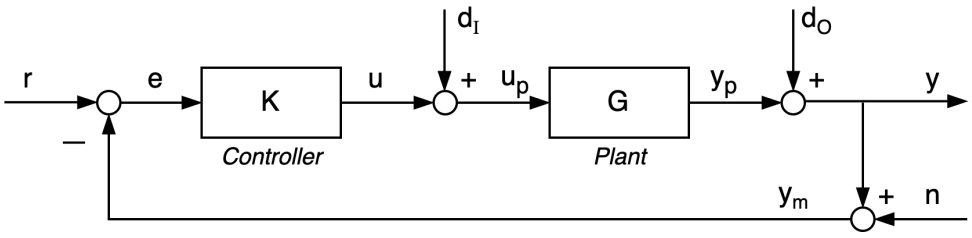


Fig. 6 General closed loop control system

objectives include disturbance rejection, noise attenuation, control effort minimization, reference tracking. Attenuating some signal's effects on the system involves minimizing the $\mathcal{H}_\infty$ norm of the transfer function between that signal and the analysis point that it affects. On top of that, enforcing robust stability to uncertainty involves minimizing the $\mathcal{H}_\infty$ norms of KS_O and both complementary sensitivity functions. A more detailed description of the relations between design specifications and the CL transfer functions can be found in [38]. However, fulfilling all of the requirements simultaneously at all frequencies is impossible, and thus trade-off must be made between robustness and performance.

The general $\mathcal{H}_\infty$ control problem can be formulated as [22]:

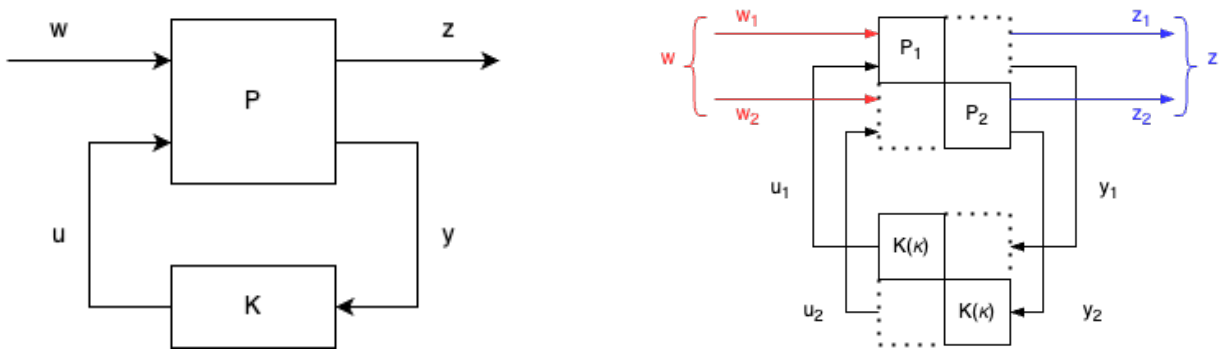
$$\begin{aligned} \text{minimize} \quad & ||T_{w\to z}(P,K)||_\infty \\ \text{subject to} \quad & K \text{ stabilizes } P \end{aligned}$$

which is schematically represented in Fig. 7a. In its core, the classical mixed-sensitivity $\mathcal{H}_\infty$ method is a way to synthesize a controller by putting appropriate weights on the selected closed loop (CL) transfer functions that attenuate the signals at certain frequencies. Notation is important here: a high frequency (HF) gain of *inverted* filter corresponds to HF maximum gain constraint for the transfer function (TF), and low frequency (LF) gain of inverted filter thus constraints the LF gain of the TF. Essentially, the filters are inverted transfer function gain constraints. For example, for output disturbance rejection at y, one can specify a performance channel $T_{w\to z}$ from d_O to y with a low-pass filter W_{S_O}, so that the disturbances are attenuated at low frequencies, where they normally happen. Likewise, for minimization of control effort at high frequencies and robustness to additive uncertainty, a high-pass filter W_{KS_O} can be put on on the channel from d_O to u. The classic mixed-sensitivity S/KS bundle is then put in a single minimization cost function under a single performance metric γ [38]:

$$\left\| \begin{bmatrix} W_{S_O} S_O \\ W_{KS_O} K S_O \end{bmatrix} \right\|_\infty \leq \gamma$$

The original $\mathcal{H}_\infty$-synthesis algorithms developed in 80-90's that use linear matrix inequalities (LMI) to solve this problem produce full-order controllers (order is the same as number of states in P plus order of all filters), which is in many cases undesirable. Furthermore, the optimization of the S/KS two-block problem can result in cancellation of the stable poles in the plant by the controller zeros because the disturbance is considered only at the plant input or output at a time [38]. On the other hand, a four-block problem with both input and output disturbances complicates filter selection [39]. The use of LMI-based optimization corresponds to $hinfsyn$ command in MATLAB. Relatively recently, in 2006, non-smooth optimization algorithms that allow synthesis of controllers with predefined structure were developed [22, 23]. The cost of this method is that it converges to local optima instead of the global one, like the former method does. However, it was shown that in practice the convergence to local optima performs even better, as it can handle problems of large scale, unlike the classical method which experiences numerical problems [23]. The non-smooth optimization then led to new solutions of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ and multidisk problems. The latter is of special interest in this case, as it allows controller synthesis for multi-objective and multi-modelling problems.

The multidisk problem formulation in a nutshell means having multiple separate performance channels $T_{w^i\to z^i}$, for which multiple plants P_i are formulated and the controller $K(\kappa)$ of fixed structure is connected to all of them



(a) General $\mathcal{H}_\infty$ control problem illustration **(b) $\mathcal{H}_\infty$ multidisk control problem illustration**

Fig. 7 Standard and multidisk $\mathcal{H}_\infty$ control problems [22]

simultaneously. The multidisk problem for two performance channels is illustrated in Fig. 7b and can be formulated as:

$$\begin{aligned} &\text{minimize} && ||T_{w_1 \to z_1}(P_1, K)||_\infty \\ &\text{subject to} && ||T_{w_2 \to z_2}(P_2, K)||_\infty \leq \gamma_2 \\ & && K \text{ stabilizes } P_1 \text{ and } P_2 \\ & && K = K(\kappa) \text{ is structured} \end{aligned}$$

Naturally, the multidisk approach can be extended to include more than two performace channels, so that it can be used to design for multiple models, multiple performance objectives, and parametric uncertainty. Its broader formulation is then [22]:

$$\min_{\kappa \in \mathbb{R}^n} \max_{i=1,\ldots,N} \alpha_i \left\| F_\ell(P_i, K(\kappa)) \right\|_\infty \tag{21}$$

where α_i is cost weight per channel, and N can also be an uncertainty set Δ. The trick lies in the fact that the maximum $\mathcal{H}_\infty$ norm among multiple separate norms is equal to the joint $\mathcal{H}_\infty$ norm. Thus, minimizing the maximum $\mathcal{H}_\infty$ among the channels also minimizes the total $\mathcal{H}_\infty$ norm of the system. This process iterates on a worst-case basis, where at each iteration a new maximum $\mathcal{H}_\infty$ norm is found and the controller is updated according to it, until a satisfactory solution is reached. In the case of real parametric uncertainty, it uses an inner relaxation method with generally the same logic as above. The solution is certified using μ-analysis as a conservative metric in the end rather than during synthesis, which outperforms outer relaxation methods such as for μ-synthesis. A more detailed description of the multidisk problem solution and application can be found in the original publication paper [24], and its application for parametric uncertainty with inner relaxation can be found in [25]. The non-smooth optimization methods have been put into the *hinfstruct* and *systune* MATLAB functions, and therefore both offer structured $\mathcal{H}_\infty$ controller synthesis with multi-modelling capabilities. The focus in this paper is put on *systune*, because it optimizes each constraint individually in a separate performance channel with its own γ metric for analysis, which allows a more flexible oversight over the performance objectives.

B. FCS synthesis

The controlled variables for the flight control system are the vertical load factor n_z and the pitch rate q, and the control input is the commanded elevator deflection angle $\delta_{e,cmd}$. Due to the exploratory nature of this research it is unknown how much robust performance can the GHAME provide, so there are no initial concrete numbers on achievable robustness margins or on how effectively the disturbances can be rejected. The general margin requirements for flight control systems include a minimum of 6 dB gain margin (GM) and 35° phase margin (PM) evaluated on a Nichols exclusion region [38]. These requirements are a good first indication of available robustness margins and are thus included in the design specifications for the FCS. Therefore, the design requirements are of general nature at this point, and are outlined below:

1) The FCS must robustly stabilize the vehicle for the nominal flight point under parametric uncertainties Δ of $\pm 20\%$ in C_m partial coefficients.

2) The FCS must provide a minimum disk GM of 6 dB and a minimum disk PM of 35° at the plant input and outputs evaluated on a Nichols exclusion region for all Δ.
3) The FCS must provide adequate n_z disturbance rejection at the plant output for all Δ.
4) The FCS must provide adequate δ_e disturbance rejection at the plant input for all Δ.
5) The FCS must provide adequate tracking of n_z reference commands for all Δ.
6) The closed loop short-period natural frequency and damping ratio must abide the flying qualities criteria outlined in [40].
7) The FCS must attenuate high-frequency control gains near actuator bandwidth.
8) The commanded actuator response overshoot must be within adequate limits to avoid actuator saturation.
9) The FCS must meet the design requirements using fixed structure controllers of minimum possible order.

1. Set-up

The Simulink set-up is shown in Fig. 8. The gray G_a block represents the actuator model with parameters presented in Table 7, and the gray G_{sp} block represents the uncertain short-period state-space model from Eq. 19. The blue blocks are the tunable controllers, and the green blocks are the inputs into the system, namely: reference n_z value represented by signal r, input disturbance d_I, and output disturbance d_O.

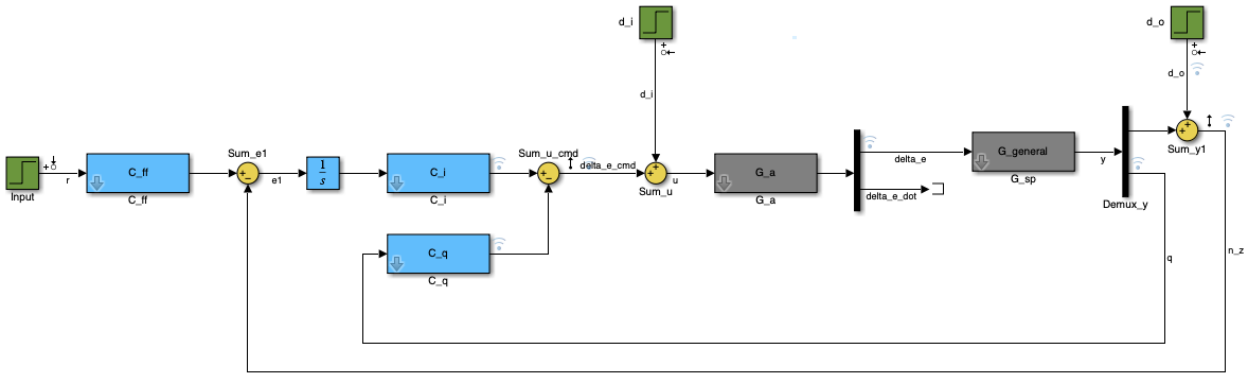


Fig. 8 Short-period FCS layout

The actuator model G_a has 2nd-order dynamics, takes commanded elevator deflection $\delta_{e,cmd}$ as input, and produces the elevator deflection δ_e and elevator deflection rate $\dot{\delta}_e$ as output. The first output δ_e is then fed as an input into the short-period airframe model G_{sp}. The joint actuator-airframe plant G_{asp} can be represented in zero-pole-gain format:

$$G_{asp}(\delta_{e,cmd} \rightarrow n_z): \quad \frac{291.44(s+6.785)(s-3.547)}{(s+0.9654)(s+1.479)(s^2+70s+2500)} \tag{22}$$

$$G_{asp}(\delta_{e,cmd} \rightarrow q): \quad \frac{-316.85(s+0.9046)}{(s+0.9654)(s+1.479)(s^2+70s+2500)} \tag{23}$$

Here, the left half plane (LHP) complex poles correspond to the actuator, and the real negative poles correspond to the airframe, they were already identified back in Table 2. The NMP zero in the n_z channel is of particular interest here, as it is located at a relatively low frequency of 3.547 rad/s. This zero imposes a limitation on achievable bandwidth of the system. The maximum possible crossover frequency ω_c (and thus the bandwidth ω_B) for a real RHP zero is [39]:

$$\omega_B \approx \omega_c < \frac{z_{NMP}}{2} = 1.7735 \quad \text{rad/s} \tag{24}$$

This is the case for an "ideal" controller without any penalty on the input u, thus making it the theoretical ceiling. In practice, however, the input would be weighted in the cost function, and the controller would be structured, so the achievable bandwidth is expected to be considerably lower. Additionally, the bandwidth ω_B is defined here as a frequency where the sensitivity S crosses the -3.01 dB line, so $\omega_B < \omega_c$, thereby decreasing it further. In a nutshell, this makes tight control (reference command following and disturbance rejection) possible only in the lower frequency range.

It is also possible to flip it around and have tight control in the frequencies above the NMP zero, but makes no practical sense in this case.

A way to bypass the bandwidth limitation is to use non-casual controllers where the output depends on future inputs, which is unfortunately unrealizable in practice [39]. Only casual controllers can be implemented in real world systems, and with them the bandwidth limitation from NMP zero simply has to be tolerated. To ensure that the controllers are casual and thus realizable, the controller transfer function must be proper, i.e., numerator cannot be of higher order than the denominator.

2. Controller structure

There are a total of 3 controllers to be tuned. Integral controller C_i is put on the error between the reference signal and the output n_z. An integral itself is placed on the line with C_i to ensure zero steady-state error. Proportional controller C_q utilizes the pitch rate q to damp rotation accordingly. A feed-forward controller C_{ff} is placed outside of the loop to tune transient response to reference commands, for which a convenient choice is a lead-lag compensator [39]. A proportional controller on n_z was found to be redundant. The C_i and C_q controller blocks were initially set as tunable gains, which led to good stability margins yet poor performance in time domain, so the order had to be increased. A final well-balanced structure was found to be the following:

$$C_i = \frac{K_i(s+z_i)}{(s+p_i)}, \quad C_q = K_q, \quad C_{ff} = \frac{K_{ff}(s+z_{ff})}{(s+p_{ff})} \tag{25}$$

Which corresponds to setting C_i and C_{ff} blocks to tunable 1-st order transfer functions with one zero and one pole, and the C_q to tunable gain.

3. Feedback tuning goals

Unlike the conventional mixed-sensitivity approach, there is no need to restructure the Simulink model from Fig. 8 into a $T_{w\to z}$ performance channel configuration with $systune$. Here, the analysis points are used to specify the desired inputs and outputs of the performance channels for constraints using the related signal names. The program then computes the corresponding transfer functions and applies the constraints on them. The selected analysis points are the reference signal r, the disturbance inputs d_i and d_o, the commanded actuator deflection $\delta_{e,cmd}$, and the output n_z. They can be seen as input perturbation and output measurement arrows on the corresponding signals in Fig. 8.

The Simulink model, analysis points, tuned blocks, and model substitutions are input into the $slTuner$ object within $systune$, which is used to create tuning goals. The tuning goals are basically the performance channels, and can handle time-domain, frequency domain, pole placement, minimum margins, and other constraints. The process is now similar to setting the conventional weighting filters for classical mixed-sensitivity. The procedure below is described for the nominal flight point.

First, the output sensitivity function S_O is constrained for output disturbance rejection at the controlled output. The $Gain$ tuning goal is created with input d_o and output n_z. The disturbances are expected to be in LF, so the inverse weighting function must be a high-pass filer. That way, output sensitivity is attenuated at low frequencies, and allowed to pass at HF, where the disturbances are not present. LF gain is thus set to a near-zero value of -60 dB. The HF gain constraint corresponds to the maximum peak value of S_O, which is directly related to the modulus disk margin. Generally, the disk margin α_{max} is computed as in Eq. 26 [41]:

$$\alpha_{max} = \frac{1}{\|S + \frac{\sigma-1}{2}\|_\infty} \tag{26}$$

where σ is the disk skew. When $\sigma = 1$, it becomes the S-based (modulus) margin with the disk centered exactly at the critical point in the Nyquist plot, and thus corresponds to the distance from the n_z open-loop curve to the critical point. Constraining $\|S\|_\infty$ puts a bound on the minimum distance to the critical point in the Nyquist plot. A common maximum peak criteria is to constrain the peak gain of S to be ≤ 2, which guarantees a GM ≥ 6 dB and PM $\geq 30°$ [39] within the n_z loop. To slightly increase the margins, the HF gain of S_O was set to 1.9. The achievable bandwidth of S_O is part of the trade-off between performance and robustness, and is specified with a gain of -3.01 dB at the selected frequency. The achieved bandwidth ω_B of S_O is 0.4 rad/s. The low bandwidth is consistent with the expected range from Eq. 24, and increasing it any further leads to actuator saturation. The inverted filter is set as a proper 1st order transfer function using a $makeweight$ MATLAB command. It is important to note that the inverted weights are passed directly into $systune$, and are inverted back automatically within the cost function.

The second tuning goal is to constrain the gain of $-KS_O$ (which is actually not a negative equivalent of KS_O since it is a multiple-output system, but the name is kept for convention). The tuning goal analysis points are d_o as input and $\delta_{e,cmd}$ as output. To attenuate HF control effort and provide robustness to additive uncertainty, the gain of $-KS_O$ must roll-off sufficiently at the actuator bandwidth. The LF gain is essentially unconstrained and set to 60 dB, since the actual gain would be the actuator DC-gain. The HF-gain must approach zero, so it is set to -60 dB. The bandwidth frequency is equal to actuator bandwidth of 50 rad/s, and the gain at that point is the performance-robustness trade-off parameter set to -16 dB for sufficient attenuation. The function is scaled with inverse of a singular value of the G_{asp} model at zero frequency. The weighting filter is 1st order proper TF.

Two more constraints follow a similar pattern for input disturbance. The tuning goal on S_OG constraints the sensitivity of output n_z to input disturbance d_i, so the analysis points are set accordingly. The DC gain limit is set to -60 dB for disturbance rejection at low frequency range. The HF-gain is set to 2, although the function rolls off at HF due to system gain G. The bandwidth at -3.01 dB is set to 0.8 rad/s, which is again a trade-off parameter. The filter is again a 1st order TF.

Finally, constraining the input complementary sensitivity T_i bounds the control effort wrt input disturbance. The transfer function T_i is set from input d_i to output $\delta_{e,cmd}$. Its peak value is located in the LF range and is constrained to a magnitude of 2. The attenuation at the actuator bandwidth of 50 rad/s is set to -23 dB. The T_i has a steeper slope than the transfer functions above – a 3rd order filter is necessary to properly constrain the roll-off, and the high-frequency gain is set to -80 dB. The singular values of all the respective filter inverses are shown in Fig. 9.

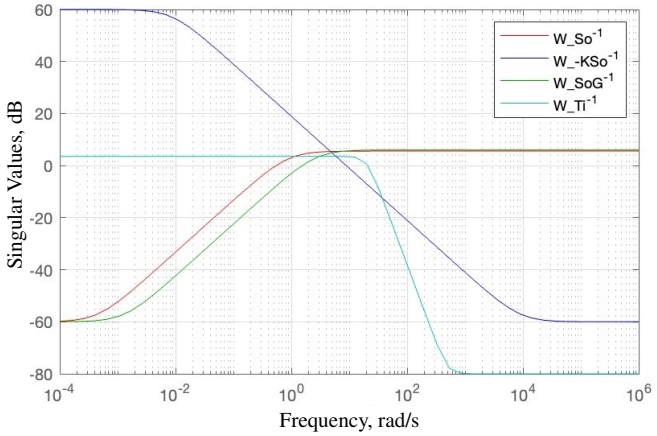


Fig. 9 Inverse weighting filters in frequency domain, nominal flight point

4. Reference tracking

The reference tracking behavior can be specified with a reference model. The $\mathcal{H}_\infty$ norm is then placed on the error between the actual model and the reference model responses. The reference model is a 2nd order transfer function with identified NMP zero, and has a form:

$$T_r = \frac{-\frac{\omega_{\text{ref}}^2}{z_{\text{nmp}}}s + \omega_{\text{ref}}^2}{s^2 + 2\omega_{\text{ref}}\zeta_{\text{ref}}s + \omega_{\text{ref}}^2} \tag{27}$$

The natural frequency and damping ratio are selected according to the short-period handling qualities criteria. A handling qualities analysis was performed on GHAME in [40] for all modes of motion, where GHAME was identified as a Class III aircraft - large, heavy, low-to-medium maneuverability. For a non-terminal flight condition of Category B (climb, cruise, descent) corresponding to the selected trim point, the short-period requirements are:

Level 1:	$0.46 \le \omega_{\text{ref}} \le 3.50$	$0.30 \le \zeta_{\text{ref}} \le 2.00$
Level 2:	$0.36 \le \omega_{\text{ref}} \le 6.00$	$0.20 \le \zeta_{\text{ref}} \le 2.00$
Level 3:	$0.36 \le \omega_{\text{ref}}$	$0.05 \le \zeta_{\text{ref}}$

Instead of guessing the optimal ω and ζ values, an optimization program $fmincon$ is used to compute them based on desired settling time and overshoot values. It is set to minimize the weighted sum of the squared errors in settling time

and overshoot between the set values and the achieved ones. The settling time is set to 4 seconds, overshoot to 1%. To meet Level 1 requirements, the bounds on optimal ω_{ref} are set between 0.5 and 3.5, and bounds on optimal ζ_{ref} are between 0.7 and 1.

The optimal values computed to be $\omega_{\text{ref}} = 1.0597$, and $\zeta_{\text{ref}} = 0.8279$, they are substituted into Eq. 27 along with the NMP zero in the n_z channel. The constraint is then set directly as $StepTracking$ tuning goal within $systune$ environment, which takes the reference model T_{ref} to optimize the transient response, and a relationship gap that is set to 15%. The input is r and the output is n_z, forming the final fifth performance channel. This concludes the process general constraint set-up for the nominal flight point model. All 5 performance specifications are passed as constraints of equal weight, and are summarized in Table 5. Finally, $systune$ function is called to compute the controllers.

5. Multi-modelling

In the multi-modelling extension, the design requirements identified earlier in this section now must be met for the entire model grid using a single controller configuration. Practically, the multi-modelling procedure is very similar to the one for the nominal flight point. Now, however, a model array is substituted into the G_{sp} block in Fig. 8. The array consists of the uncertain state space models from the grid in Table 4. Given a broad variety of conditions that the controller must satisfy, there are now two sets of tuning goals. One set contains the hard constraints that are applied on the uncertain model at the nominal flight point (model 1 from Table 4), whereas the second set contains soft constraints, which are applied on all the other uncertain models including those with fuel variation at the nominal trim point (models 2-11). Evidently from the name, the hard constraints are prioritized over the soft ones in the algorithm. The new hard and soft constraints cover the same performance channels as before, carry similar names, and are specified with the same logic. They are, however, generally more relaxed compared to designing for nominal flight point only, which is understandable since a single controller now has to satisfy the full grid of models. The final values for hard and soft constraints for multi-modelling are outlined in Table 5.

C. Results and analysis

The results are analysed in consecutive steps. First, the γ performance metric is given, one per tuning goal. This is a first indication if the constraints have been met successfully. The transfer functions are then plotted in frequency domain with their respective inverse filters to check how well they comply. The robustness is assessed with disk margins at the plant input and outputs. The open loops are then plotted on the Nichols chart. Each transfer function is then plotted in the time domain to assess performance wrt disturbance rejection and control effort.

The tuned controllers for the nominal FP design and for the full grid design are presented in Eq. 28.

$$\begin{aligned} &\text{Nominal FP:} && \text{Full Grid:} \\ C_i &= \frac{-3.7014(s+0.9311)}{(s+3.25)} & C_i &= \frac{-4.0888(s+0.8258)}{(s+2.434)} \\ C_q &= -24.587 & C_q &= -34.412 \\ C_{ff} &= \frac{0.80293(s+1.182)}{(s+0.9454)} & C_{ff} &= \frac{0.62306(s+1.466)}{(s+0.9135)} \end{aligned} \tag{28}$$

The resulting γ values for the nominal flight point design are:

$$\text{Nominal: } \gamma(S_O) = 0.9446, \quad \gamma(-KS_O) = 0.9964, \quad \gamma(S_OG) = 0.9968, \quad \gamma(T_i) = 0.9964, \quad \gamma(T_{\text{ref}}) = 0.9935 \tag{29}$$

The closer γ is to 1 from the lower side, the better performance is; a value larger than 1 indicates a constraint violation. The values from Eq. 29 can be related to the graphs in Fig. 10, where the respective transfer functions' singular values are plotted with their filter inverses. The γ values close to 1 from the lower side correspond to singular values from Fig. 10 (blue) being located close to the borders of the weighting filters (yellow), but not violating them. This indicates a successful constraint selection which uses full potential of the system. It is worth noting that γ for the reference tracking constraint is measured in the time domain according to how much the relationship gap is violated. It is therefore possible that the fifth γ is violated by one of the sampled models, but the average step tracking for all of them is generally in line with requirements, so it is better to assess the fifth tuning goal compliance visually. Notice also how the 3rd order weighting filter on T_i smoothly constraints its entire roll-off slope, whereas a lower order filter was found to be not steep enough, basically affecting T_i only at break-out frequency.

Table 5 Weighting filters for the nominal flight point (left) and for multi-modelling (right)

Constraint specification	Nominal flight point	Full grid, hard	Full grid, soft
$d_o \rightarrow n_z$:	S_O tuning goal, 1st order TF		
LF-gain	-60 dB	-60 dB	-60 dB
Bandwidth ω_B	0.4 rad/s	0.35 rad/s	0.3 rad/s
Gain at ω_B	-3.01 dB	-3.01 dB	-3.01 dB
HF-gain	1.9	1.9	1.9
$d_o \rightarrow \delta_{e,cmd}$:	$-KS_O$ tuning goal, 1st order TF		
LF-gain	60 dB	60 dB	60 dB
Bandwidth ω_B	50 rad/s	50 rad/s	50 rad/s
Gain at ω_B	-16 dB	-16 dB	-16 dB
HF-gain	-60 dB	-60 dB	-60 dB
$d_i \rightarrow n_z$:	S_OG tuning goal, 1st order TF		
LF-gain	-60 dB	-60 dB	-60 dB
Bandwidth ω_B	0.8 rad/s	1.4 rad/s	1.3 rad/s
Gain at ω_B	-3.01 dB	-3.01 dB	-3.01 dB
HF-gain	2	2	2
$d_i \rightarrow \delta_{e,cmd}$:	T_i tuning goal, 3rd order TF		
LF-gain	2	2	2
Bandwidth ω_B	50 rad/s	50 rad/s	50 rad/s
Gain at ω_B	-23 dB	-20 dB	-15 dB
HF-gain	-80 dB	-80 dB	-80 dB
$r \rightarrow n_z$:	T_{ref} tuning goal, using relationship gap		
ω_{ref}	1.0597	1.0597	1.0597
ζ_{ref}	0.8279	0.8279	0.8279
Relationship gap	15 %	15 %	20%

For the multi-modelling tuning with hard and soft constraints, two γ values are now given per performance channel: one for soft constraint and one for hard constraint, giving a total of 10 performance metrics. The respective γ values for the multi-modelling case are presented below:

$$\text{Hard: } \gamma(S_O) = 1.0000, \quad \gamma(-KS_O) = 1.0000, \quad \gamma(S_OG) = 1.0000, \quad \gamma(T_i) = 1.0000, \quad \gamma(T_{\text{ref}}) = 0.7156 \tag{30}$$

$$\text{Soft: } \gamma(S_O) = 1.0317, \quad \gamma(-KS_O) = 1.0505, \quad \gamma(S_OG) = 1.0009, \quad \gamma(T_i) = 1.0815, \quad \gamma(T_{\text{ref}}) = 1.2557 \tag{31}$$

Naturally, there are also 10 tuning goal compliance graphs, one per hard and one per soft constraint. They are shown Fig. 11, and follow similar relationship as in the nominal FP case. As can be seen from the γ values and the charts, the hard constraints are perfectly met, whereas some slight violation is present in the soft requirements. This is considered an acceptable violation, given the variety of trim conditions and the fact that less weight is put on the soft constraints than on the hard ones. The γ of reference tracking is understandably large for the soft case, but the average step pattern follows the desired one closely, and the models have approximately similar transient time, as can be seen in the same Fig. 11. Overall, the controller satisfies the tuning goals quite well for all 11 *uss* models. Additionally, the observed successful roll-off of T_i and $-KS_O$ at the actuator bandwidth frequency of 50 rad/s satisfies FCS requirement 7, whereas the successful fulfillment of step tracking goal fulfills FCS requirement 6.

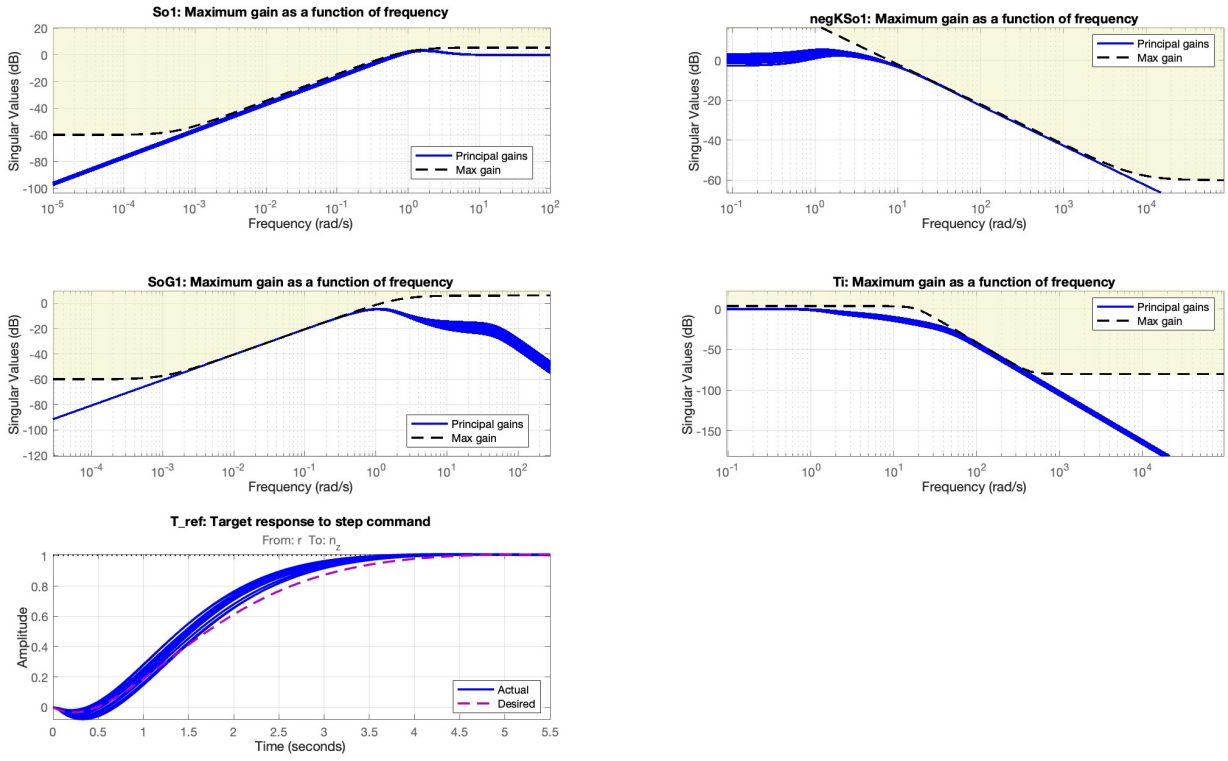


Fig. 10 Tuning goals results for nominal flight point model

For the SISO open-loop analysis, the uncertain state-space models are sampled at random values of uncertain parameters in C_{m_α}, $C_{m_{\delta_e}}$, C_{m_q} using $usample$ function in MATLAB. For illustration purposes, 20 state-space systems were randomly sampled per model in the grid. This amounts to 20 models for the nominal flight point design, and $11 \times 20 = 220$ models for the full grid design. The SISO open-loops were then computed with $loopsens$ MATLAB function which connects the model array G_{asp} to controller matrix K and automatically opens the loops at inputs and outputs. It also indicates if the closed-loops are stable and, to no surprise, confirmed the stability of all closed loops in the model arrays, thus satisfying FCS design requirement 1. The SISO open loops were then extracted, and the $diskmargin$ MATLAB function was called on all of the samples to compute the $S-T$ disk margin (DM) corresponding to Eq. 26 with $\sigma = 0$. This way the DM is symmetrical, with equal probability that gain and phase can vary in either direction. The smallest DM, disk gain margin (DGM), disk phase margin (DPM), and multi-loop input-output margin (MMIO) among all samples were extracted. It is worth noting that the multi-loop disk margin incorporates the structured singular value μ in the computation process [41].

The DM's were then used to plot the disks of the worst-case margins on the Nichols charts (orange), and the 6 dB DGM requirement was used to plot the exclusion regions (red). The opened SISO loops were then plotted on their respective Nichols charts, together with DGM (green) and DPM (purple) lines for better comprehension.

The opened loops at the plant input are presented in Fig. 12. Naturally, the margins for the worst case loop among those sampled at the nominal flight condition in Fig. 12a are significantly larger than the worst-case margins for the full grid samples in Fig. 12b. Nevertheless, both worst-case sampled scenarios pass the FCS design requirement 2 with flying colors. In contrast, the open loops at the plant output n_z, shown in Fig. 13, are significantly closer to the exclusions region. In a similar fashion, the margins for the full grid case in Fig. 13b are smaller than for the nominal point in Fig. 13a, but still pass the design requirement 2.

Although the n_z channel is the most important to analyze since that is the output to be controlled by the reference signal and that is the loop on which the output disturbance is acting, it is still necessary to consider all inputs and outputs in the system. The Nichols charts for the sampled SISO loops opened at plant output q are shown in Fig. 14, and follow the same trend as in the previous two charts with good DGM and DPM.

Therefore, a single set of controllers successfully handles the short-period dynamics under the specified parametric

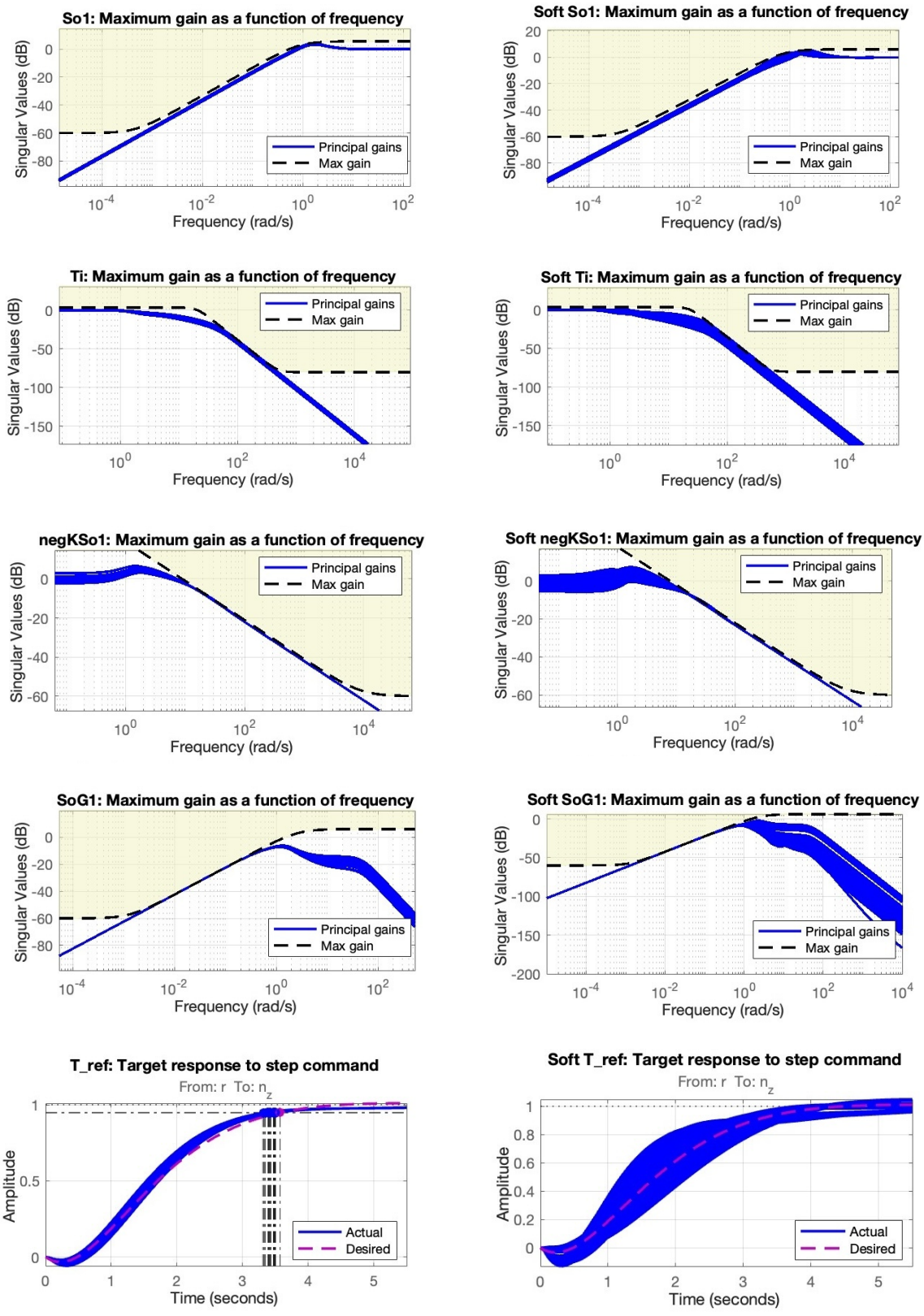
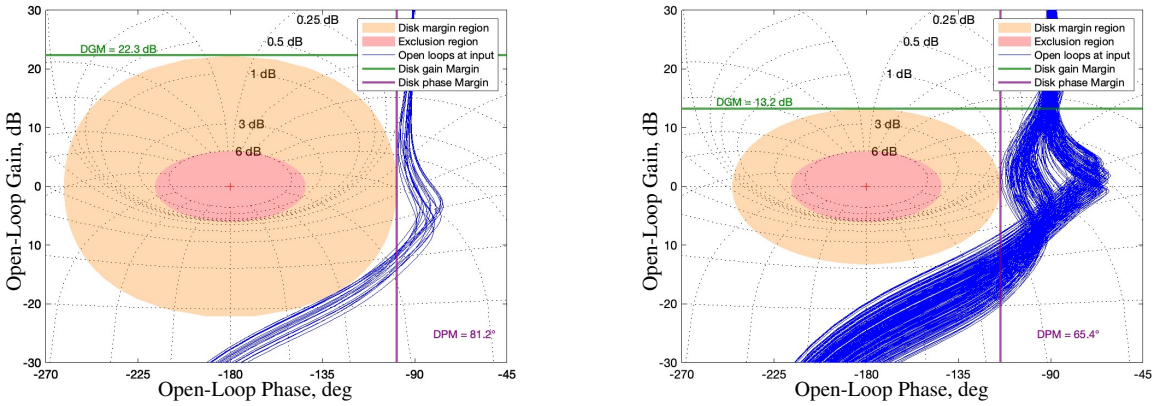
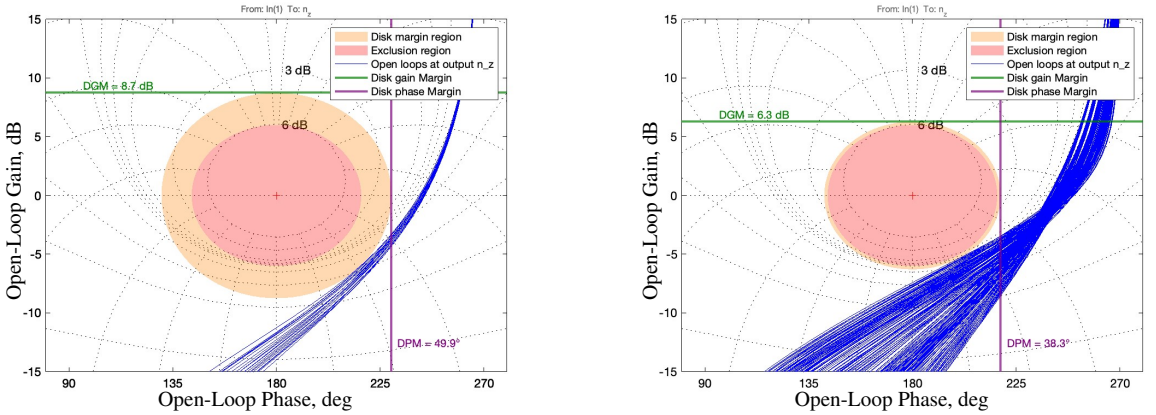


Fig. 11 Tuning goals results for full grid



(a) Nominal flight point **(b) Full grid**

Fig. 12 Nichols chart of sampled SISO loops opened at plant input

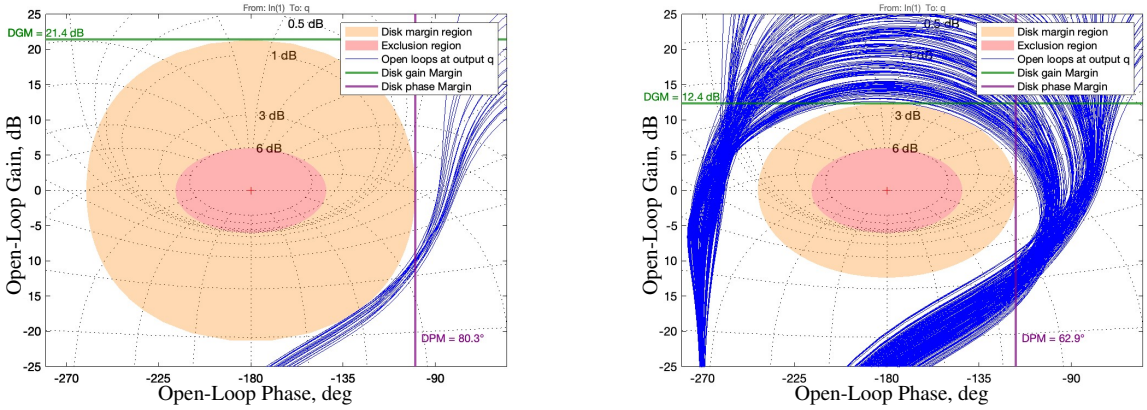


(a) Nominal flight point **(b) Full grid**

Fig. 13 Nichols chart of sampled SISO loops opened at plant output n_z

uncertainty of 20% in all C_m partial coefficients, and still meets the requirement of 6 dB and 35° at the inputs and outputs of the plant (FCS requirement 2 satisfied). This is especially a considerable achievement in the multi-modelling case, where 11 uncertain models from the grid are sampled into 20 each. The controller still guarantees robustness to unstructured uncertainty with larger margins than required, and that is the case for the worst of the 220 samples. To summarize the disk margin analysis, the DGM and DPM values from the Nichols charts are presented all together in Table 6, along with simultaneous MMIO gain and phase margins. The MMIO margins are indeed lower, to no surprise. However, those are, again, the worst case parametric uncertainty margins, and there are still robustness guarantees against simultaneous multi-loop changes at inputs and outputs.

Lastly, the CL performance of disturbance rejection and reference tracking were analysed in the time domain. In a similar fashion as for the SISO loops, the CL transfer functions were sampled in the parametric uncertainty domain Δ. The nominal flight point design is presented next to the full grid multi-modelling design. The step responses of the transfer functions are grouped together according to output disturbance rejection case, input disturbance rejection case, and the reference tracking case. In each of the plots, the thick red line indicates the nominal system response (no uncertainty) at the nominal operating point.



(a) Nominal flight point **(b) Full grid**

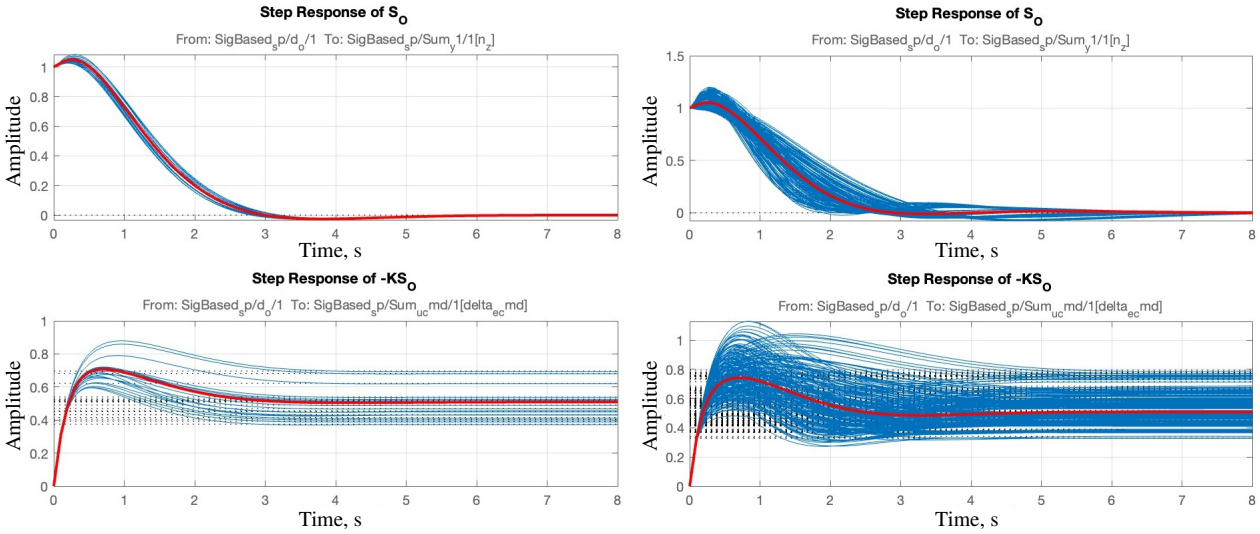
Fig. 14 Nichols chart of sampled SISO loops opened at plant output q

Table 6 Minimum achieved symmetrical disk margins under sampled parametric uncertainty

Margin type	Nominal flight point	Full grid	Units
Disk gain margin at input $\delta_{e,cmd}$	22.28	13.93	dB
Disk gain margin at output n_z	8.75	6.47	dB
Disk gain margin at output q	21.40	13.10	dB
Multi-loop input-output gain margin	3.90	2.88	dB
Disk phase margin at input $\delta_{e,cmd}$	81.20	67.25	deg
Disk phase margin at output n_z	49.87	39.22	deg
Disk phase margin at output q	80.28	65.04	deg
Multi-loop input-output phase margin	27.92	25.41	deg

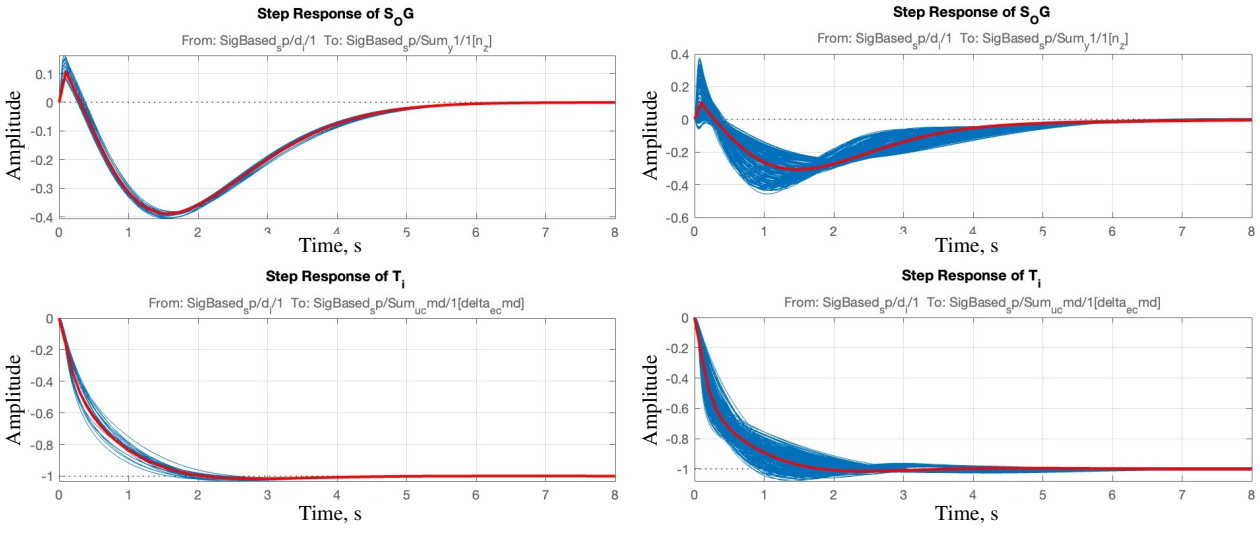
For output disturbance rejection, the step response of S_O indicating n_z response to d_o is plotted together with step response of $-KS_O$, which represents the commanded control response to reject d_o. The step responses are plotted in Fig. 15. It is evident that the disturbance is rejected efficiently in under about 3 seconds, thus fulfilling FCS design requirement 3. Both the overshoot and steady-state amplitudes of the $-KS_O$ plots exceed the actuator deflection limit of $20° = 0.3491$ rad. This indicates that the maximum achievable n_z value is significantly less than 1, which can not be accounted for in linear simulations directly. The overshoot, however, is contained to not exceed the steady-state value too far, implicitly avoiding potential actuator saturation and satisfying FCS requirement 8. The control step responses ($-KS_O$) for the full grid are especially vastly spread, which is expected. For example, look at the plots exactly at 2 seconds into the step in Fig. 15b. The slowly descending response with the highest amplitude at 2 seconds for both S_O and $-KS_O$ corresponds to the 9th model in the array (refer to grid in Table 4). G_sp_075_70_55 flies at the top altitude in the grid (smallest speed of sound), and at the smallest Mach in the grid, which together mean it has the smallest airspeed V among the entire model grid. Additionally, it has largest fuel fraction in the tank, thus having larger mass and MOI, so G_sp_075_70_55 has the least control authority in the grid, predictably making its response slow and required actuator deflection large.

In the same manner, the input disturbance rejection is assessed with $S_O G$ for output and T_i for control response in Fig. 16. Since the disturbance is acting on the commanded input, it is the actuator deflection that disturbs the output. Therefore, the n_z value oscillates around 0 until the input disturbance is completely rejected. As can be seen in the figures, the input disturbance is rejected smoothly without overshoot in the commanded deflection path, and n_z is brought back to zero for all models. The FCS requirement 4 is thus fulfilled.



(a) Nominal flight point model **(b) All models in the grid**

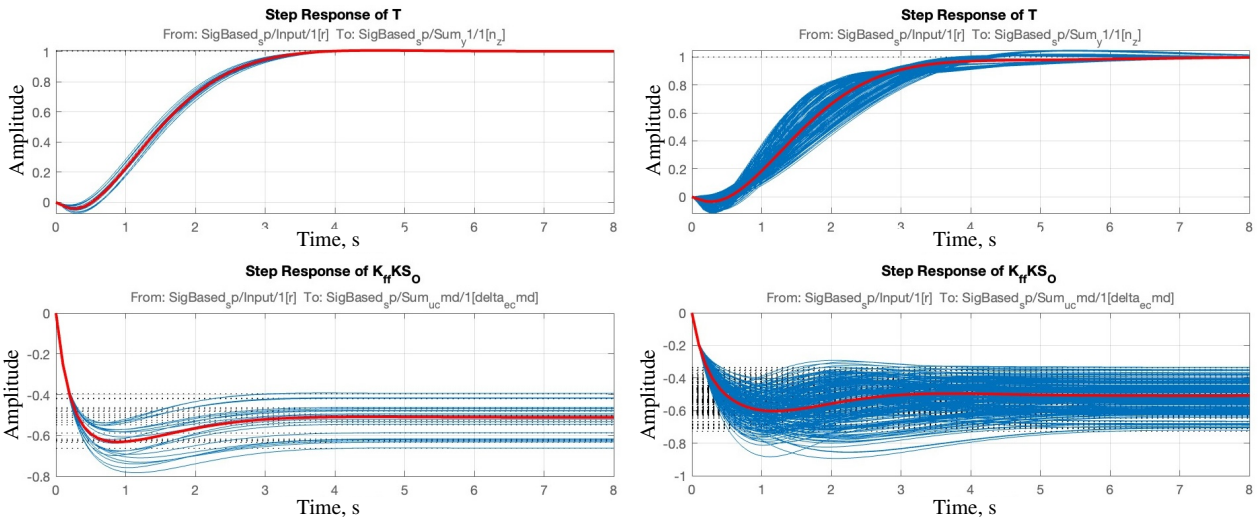
Fig. 15 Output disturbance rejection on linear system



(a) Nominal flight point model **(b) All models in the grid**

Fig. 16 Input disturbance rejection on linear system

Lastly, the reference n_z step tracking is presented in Fig. 17 with the transfer functions $T_{r\to n_z}$ and $[K_{ff}KS_O]_{r\to\delta_{e,cmd}}$. In the nominal flight point design in Fig. 17a, where the samples correspond only to previously defined parametric uncertainty, it can be observed for $K_{ff}KS_O$ how different the steady-state actuator deflections are. Needless to say that adding another 10 models to the nominal FP in Fig. 17b fills the chart completely. Nevertheless, that is the reason why the n_z tracking in the T charts is smoothly performed. Although it was possible to set a desired step tracking transient time 1 second faster than it is here, it resulted in severe actuator response overshoot (about double the steady state value), so a 1 second slower reference tracking response time was set to avoid actuator overload. Thus, the FCS requirements 5 and 8 are satisfied. As a result, the control system successfully meets all the design requirements specified back in Sec. IV.B, both for the nominal flight point design and for the multi-modelling design. The next step is to test it in the nonlinear simulation.



(a) Nominal flight point model **(b) All models in the grid**

Fig. 17 Reference step tracking on linear system

V. Non-linear implementation and simulation

A. Implementation

To test the synthesized controllers, a flight control system block is added to the Simulink model, as well as input and output disturbances. The complete nonlinear Simulink model layout is shown in Fig. 18. The *FCS* block is placed before the actuators, with output measurements fed back into it. The input disturbance is positioned to act simultaneously on both elevons in the *Actuators* block, and the output disturbance acts directly on the n_z component of the load factor in the *Sensors* block.

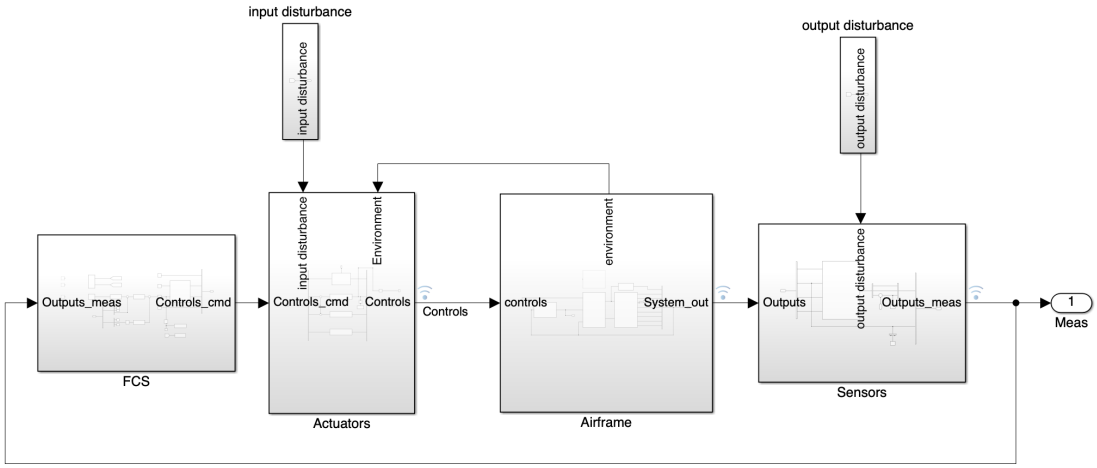


Fig. 18 Flight control system implementation into nonlinear model

The expanded *FCS* block is shown in Fig. 19. The n_z and q are extracted from the output measurements and fed into the familiar controller block structure. This time, however, $\delta_{e,cmd}$ must be initialized with the trim value, for which the integrator was moved past the summation point with C_q. It also contains saturation limits to avoid out-of-bounds commands. Since the q channel is now in the integrator path, it must be differentiated beforehand [42]. Therefore,

a derivative is placed before the C_q block. The time step of the simulation is set to 0.001 s, so the derivative block sampling is set to 0.01 s to have a smaller discrete time step relative to the simulation.

Commanded deflection $\delta_{e,cmd}$ is then fed into conversion block with $\delta_{a,cmd}$ to transform the commanded virtual elevator and aileron deflections into real elevon deflections. They are then added to the bus with trimmed throttle and trimmed rudder values and sent into the *Actuator* block. Lastly, in the top left corner of Fig. 19 there are reference and disturbance step inputs, each next to its destination: output disturbance and input disturbance into the $Goto$ blocks, and the reference signal fed directly to feed-forward controller.

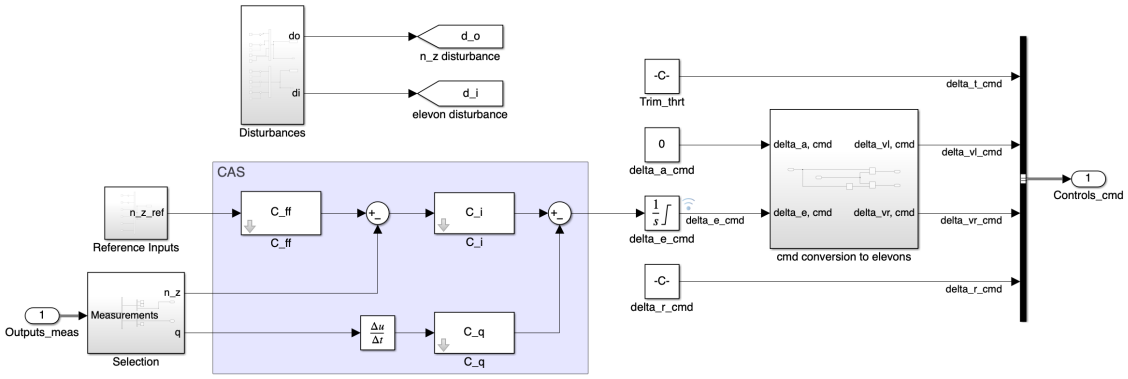


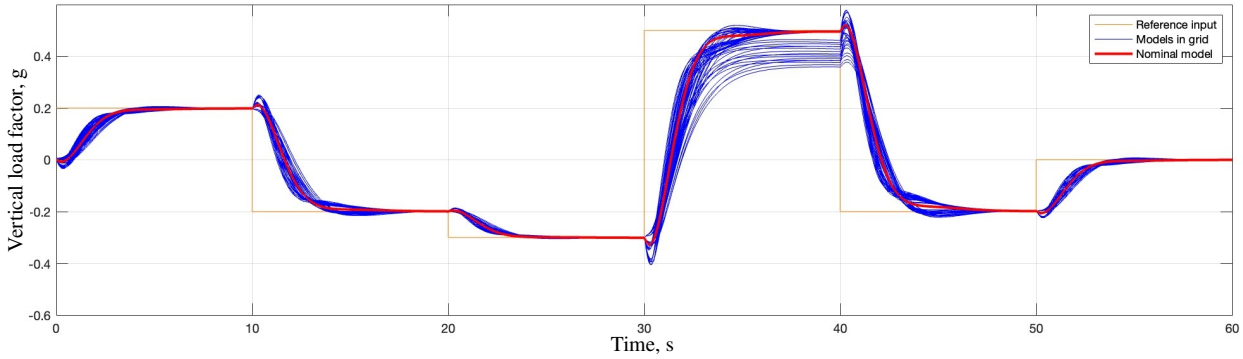
Fig. 19 Expanded flight control system block of the nonlinear model

B. Simulation

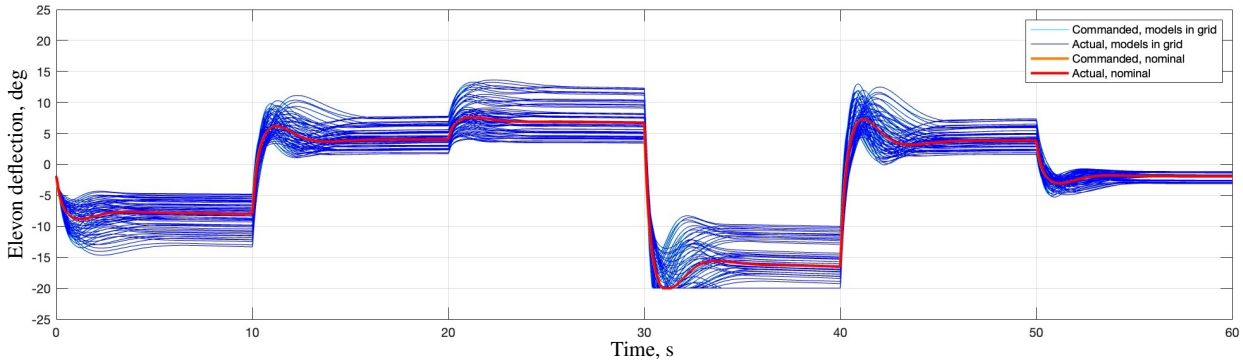
The controller blocks were set to the full-grid variant from Eq. 28. The model was initialized, trimmed, and simulated a total of 65 times to cover the operational window of 4500-5500 m, 0.7-0.8 Mach, 0.25-0.75 fuel fraction, and the 20% variations in the pitch moment partial coefficients. The latter were perturbed by directly multiplying the computed coefficient values with 0.8 or 1.2, depending on the iteration. The simulations were carried out separately for the reference tracking and the disturbances. As before, the nominal model response is highlighted with a thick red line.

The results of the step commands tracking are displayed in Fig. 20. The vertical load factor in Fig. 20a smoothly tracks the reference value for all models. The exception is for the command at 30 seconds into simulation, where the command was purposely set to a value which is unreachable for some of the models, in order to display the limitations of achievable load factor commands for GHAME vehicle at this flight point. Both commanded and actual elevon deflections are plotted in Fig. 20b, where the reached deflection limit is seen for the same reference step at 30 seconds. The respective actuator deflection rates are shown in Fig. 20c, they are far from saturating the deflection rate limits of 400 °/s. Lastly, the pitch rate is displayed in Fig. 20d. Note that the maximum achievable pitch rate with saturated elevons is only around 1.5 °/s at these conditions.

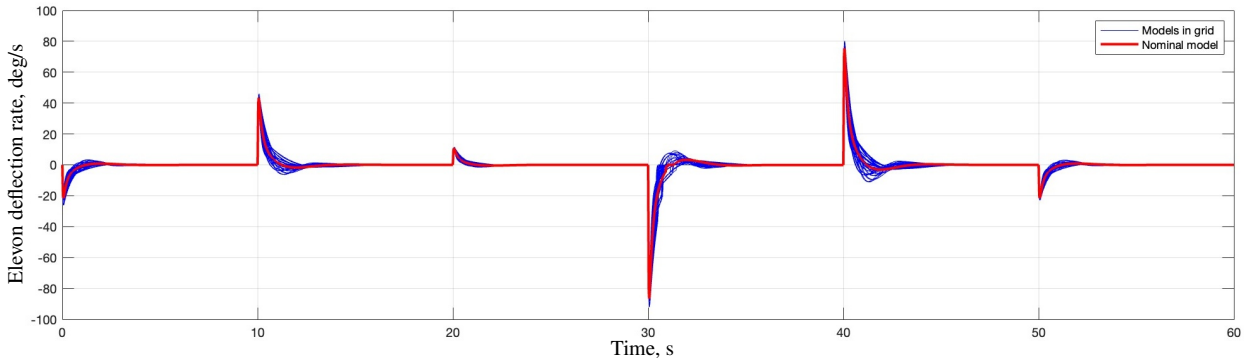
The disturbance rejection responses are shown in Fig. 21. The simulation starts with two consecutive input disturbances, then 3 consecutive output disturbances, and then a final joint input-output disturbance which superimposes for n_z. The input disturbance magnitudes can be read from the elevon deflection graph in Fig. 21b, where the deflection angle is instantaneously changed to disturbance value. The output disturbance magnitudes can be read from impulse changes in n_z in Fig. 21a. Note how the commanded deflection separates from the actual one due to input disturbances, yet controls the elevons well. The input disturbances are also easily spotted in the deflection rate graph in Fig. 21c - they appear as vertical lines that move past the boundaries of deflection rates. That is because disturbance step inputs act instantaneously on the deflection angle. Nevertheless, all disturbances are successfully rejected, which is clearly shown in Fig. 21a where the vertical load factor is brought to zero.



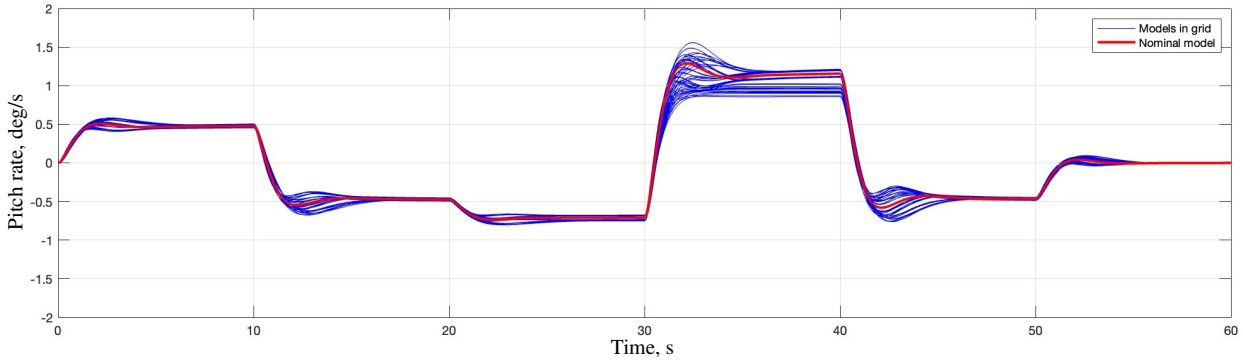
(a) Vertical load factor response



(b) Elevon deflection response

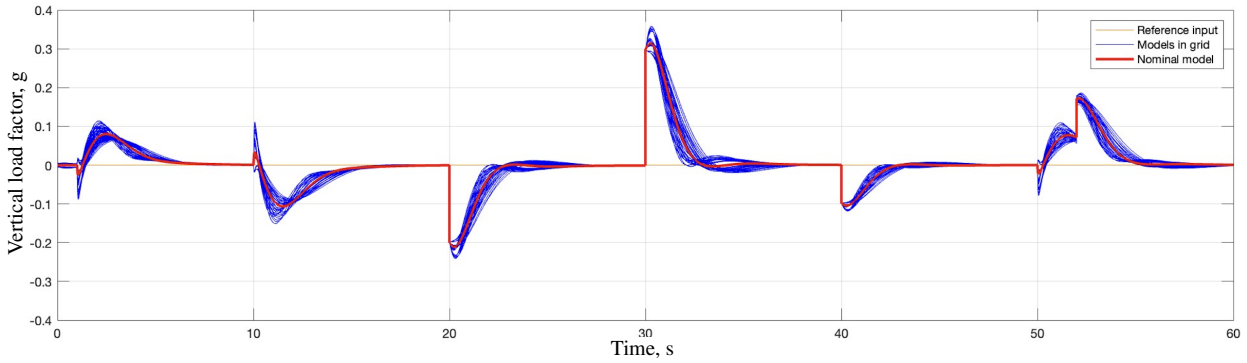


(c) Elevon deflection rate response

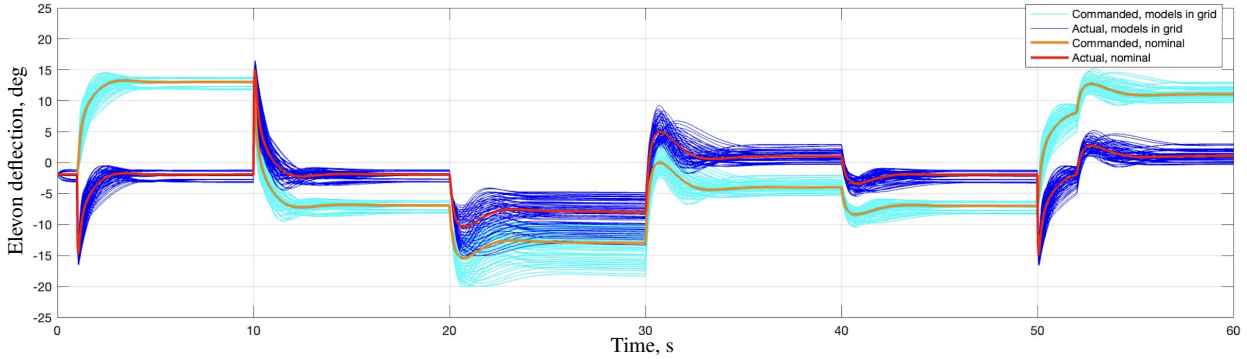


(d) Pitch rate response

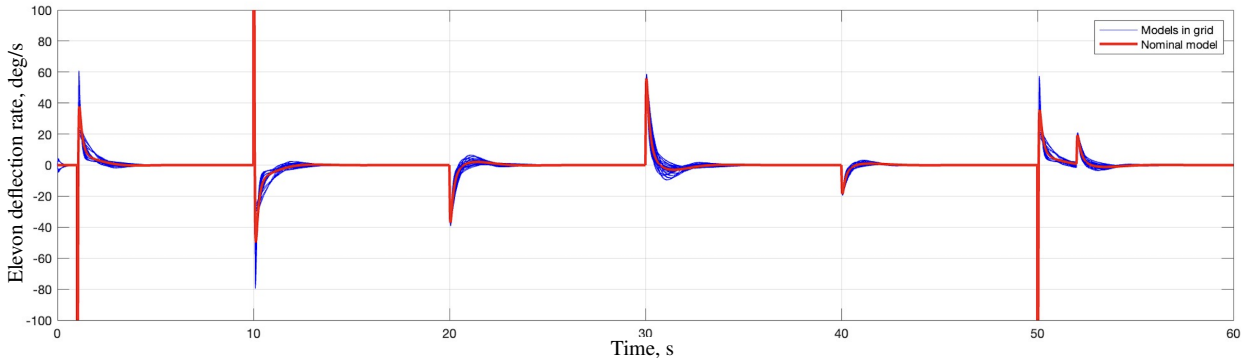
Fig. 20 Nonlinear simulation responses to reference load factor commands



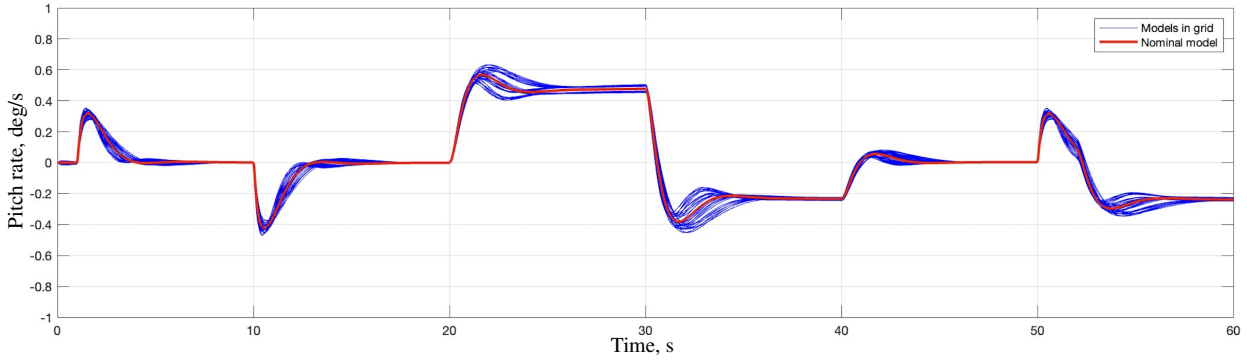
(a) Vertical load factor response



(b) Elevon deflection response



(c) Elevon deflection rate response



(d) Pitch rate response

Fig. 21 Nonlinear simulation responses to input and output disturbances

VI. Conclusion and Recommendations

This research focused on developing a robust flight control system for the GHAME hypersonic vehicle's subsonic short-period model under parametric uncertainties. A nonlinear GHAME model was integrated into Simulink, with a flexible subsystem-separated structure. During linearization, discrepancies in aerodynamic data were resolved by adjusting interpolation methods for aerodynamic coefficients. The model was trimmed and linearized at 11 operating points, incorporating 20% uncertainty in pitch moment partial coefficients, the short-period dynamics were extracted.

A fixed-structure flight control system was synthesized using multi-objective $\mathcal{H}_\infty$ mixed-sensitivity methods for disturbance rejection, high-frequency control attenuation, and reference tracking at the nominal flight point. This design was extended to a full grid of uncertain models, and the controller successfully stabilized all sampled models, meeting robustness margin requirements. The results showed effective disturbance rejection at the plant input and output at low frequencies, and smooth control signal roll-off at actuator bandwidth. The FCS was incorporated into the nonlinear model and successfully tested for disturbance rejection and reference tracking under parameter variations, demonstrating that a single fixed-structure controller can manage the GHAME model's subsonic dynamics under uncertainty.

Future work should address model limitations and control system enhancements, such as extending the framework to include phugoid and lateral dynamics, accounting for aeroelastic effects, incorporating engine dynamics, CG and in-flight mass variations. For future FCS designs, a broader flight envelope should be covered with super/hypersonic speeds and altitudes, potentially extending to gain-scheduled controllers. HV-specific scenarios like angle of attack hold during maneuvers should be considered. Furthermore, adding output disturbances for pitch rate, and incorporating gain and phase margin constraints can improve robustness of this research's particular design.

Appendix

Table 7 Mechanical and Control Constants

Name	Symbol	Value	Unit
Take-off gross vehicle mass	m_0	136077	kg
Total fuel mass	$m_{f_{\text{tot}}}$	81646	kg
Vehicle mass at burn-out	m_1	54431	kg
Moment of Inertia take-off (I11)	I_{11_0}	1.573×10^6	kg.m^2
Moment of Inertia take-off (I22)	I_{22_0}	31.6×10^6	kg.m^2
Moment of Inertia take-off (I33)	I_{33_0}	32.54×10^6	kg.m^2
Moment of Inertia take-off (I13)	I_{13_0}	0.38×10^6	kg.m^2
Moment of Inertia burn-out (I11)	I_{11_1}	1.18×10^6	kg.m^2
Moment of Inertia burn-out (I22)	I_{22_1}	19.25×10^6	kg.m^2
Moment of Inertia burn-out (I33)	I_{33_1}	20.2×10^6	kg.m^2
Moment of Inertia burn-out (I13)	I_{13_1}	0.38×10^6	kg.m^2
Wing area	S	557.42	m^2
Wing span	b	24.38	m
Wing chord	c	22.86	m
Engine cowl area factor	A_c	27.87	-
IMU position	s_{SB}	24.4	m
Standard gravity	g_0	9.80665	m/s^2
Maximum actuator deflection	$\delta_{v_{\max}}$	±20	°
Maximum actuator deflection rate	$\dot{\delta}_{v_{\max}}$	±400	°/s
Natural frequency of actuator	ω_v	50	rad/s
Damping ratio of actuator	ζ_v	0.7	-

The state-space matrices for the nominal short-period model at nominal trim condition are shown in Eq. 32.

$$A = \begin{bmatrix} -0.9091 & 1 \\ -0.03207 & -1.535 \end{bmatrix},\ B = \begin{bmatrix} -0.01765 \\ -0.1267 \end{bmatrix},\ C = \begin{bmatrix} 22.16 & -3.817 \\ 0 & 1 \end{bmatrix},\ D = \begin{bmatrix} 0.1166 \\ 0 \end{bmatrix} \tag{32}$$

Acknowledgments

The authors would like to thank Prof. Peter H. Zipfel for his contribution to this research and support with the GHAME aerodynamic model implementation.

References

[1] Sziroczak, D., and Smith, H., "A review of design issues specific to hypersonic flight vehicles," *Progress in Aerospace Sciences*, Vol. 84, 2016, pp. 1–28. https://doi.org/10.1016/j.paerosci.2016.04.001.

[2] Bolender, M. A., "An Overview on Dynamics and Controls Modelling of Hypersonic Vehicles," , 2009. https://doi.org/10.1109/ACC.2009.5159864.

[3] Erbland, P. J., "Current and Near-Term RLV/Hypersonic Vehicle Programs," *RTO/AVT VKI Lecture Series on "Critical Technologies for Hypersonic Vehicle Development", Brussels, Belgium*, 2004, pp. 10–14.

[4] Fidan, B., Mirmirani, M., and Ioannou, P., "Flight dynamics and control of air-breathing hypersonic vehicles: review and new directions," *12th AIAA international space planes and hypersonic systems and technologies*, 2003, p. 7081. https://doi.org/10.2514/6.2003-7081.

[5] Calise, A. J., "Research In Robust Control For Hypersonic Aircraft," Report NASA-CR-192127, NASA, 1993.

[6] Vu, P., and Biezad, D. J., "Longitudinal Control of Hypersonic Aircraft: an Alpha Follow-Up Scheme," , 1993. https://doi.org/10.1109/AEROCS.1993.720973.

[7] Gregory, I., McMinn, J., Shaughnessy, J., and Chowdhry, R., "Hypersonic vehicle control law development using H infinity and mu-synthesis," *AIAA 4th International Aerospace Planes Conference*, 1992. https://doi.org/https://doi.org/10.2514/6.1992-5010.

[8] Vu, P., and Biezad, D. J., "A pseudo-loop design strategy for the longitudinal control of hypersonic aircraft," , 1993. https://doi.org/10.2514/6.1993-3814.

[9] Shaughnessy, J. D., Pinckney, S. Z., McMinn, J. D., Cruz, C. I., and Kelley, M.-L., "Hypersonic Vehicle Simulation Model: Winged-Cone Configuration," Report NAS 1.15:102610, NASA, 1990.

[10] White, D., and Sofge, D., *Handbook of Intelligent Control*, Van Nostrand Reinhold, 1992. Ch. 11.

[11] Bolender, M., and Doman, D., "A Non-Linear Model for the Longitudinal Dynamics of a Hypersonic Air-breathing Vehicle," , 2005. https://doi.org/10.2514/6.2005-6255.

[12] Parker, J. T., Serrani, A., Yurkovich, S., Bolender, M. A., and Doman, D. B., "Control-Oriented Modeling of an Air-Breathing Hypersonic Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 3, 2007, pp. 856–869. https://doi.org/10.2514/1.27830.

[13] Groves, K., "Modelling, simulation, and control design of an air-breathing hypersonic vehicle," Thesis, 2005. URL http://rave.ohiolink.edu/etdc/view?acc_num=osu1302726196.

[14] Groves, K., Sigthorsson, D., Serrani, A., Yurkovich, S., Bolender, M., and Doman, D., "Reference Command Tracking for a Linearized Model of an Air-Breathing Hypersonic Vehicle," , 2005. https://doi.org/10.2514/6.2005-6144.

[15] Sigthorsson, D. O., Jankovsky, P., Serrani, A., Yurkovich, S., Bolender, M. A., and Doman, D. B., "Robust Linear Output Feedback Control of an Airbreathing Hypersonic Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1052–1066. https://doi.org/10.2514/1.32300.

[16] Anderson, M., Emami-Naeini, A., and Vincent, J., "Robust Cotrol Law Development for a Hypersonic Cruise Aircraft," *1991 American Control Conference*, IEEE, 1991, pp. 839–845. https://doi.org/10.23919/ACC.1991.4791490.

[17] Buschek, H., and Calise, A., "Fixed order robust control design for hypersonic vehicles," *Guidance, Navigation, and Control Conference*, AIAA, 1994, p. 3662. https://doi.org/10.2514/6.1994-3662.

[18] Buschek, H., and Calise, A. J., "Robust Control of Hypersonic Vehicles Considering Propulsive and Aeroelastic Effects," *Guidance, Navigation and Control Conference*, AIAA, 1993. https://doi.org/https://doi.org/10.2514/6.1993-3762.

[19] Buschek, H., and Calise, A., "Uncertainty Modeling and Fixed-Order Controller Design for a Hypersonic Vehicle Model," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 42–48. https://doi.org/10.2514/2.4031.

[20] Duan, H., and Li, P., "Progress in control approaches for hypersonic vehicle," *Science China Technological Sciences*, Vol. 55, 2012, p. 2965–2970. https://doi.org/https://doi.org/10.1007/s11431-012-5036-x.

[21] Mirmirani, M., Wu, C., Clark, A., Choi, S. B., and Colgren, R., "Modeling for control of a generic airbreathing hypersonic vehicle," *AIAA guidance, navigation, and control conference and exhibit*, 2005, p. 6256. https://doi.org/10.2514/6.2005-6256.

[22] Apkarian, P., and Noll, D., "The H_∞ Control Problem is Solved," *Aerospace Lab*, 2017. https://doi.org/10.12762/2017.AL13-01.

[23] Apkarian, P., and Noll, D., "Nonsmooth H_∞ synthesis," *IEEE Transactions on Automatic Control*, Vol. 51, No. 2, 2006, pp. 382–382. https://doi.org/10.1109/TAC.2005.860290.

[24] Apkarian, P., and Noll, D., "Nonsmooth optimization for multidisk H_∞ synthesis," *European Journal of Control*, Vol. 12, No. 3, 2006, pp. 229–244. https://doi.org/https://doi.org/10.3166/ejc.12.229-244.

[25] De Aguiar, R. S. D. S., Apkarian, P., and Noll, D., "Structured Robust Control Against Mixed Uncertainty," *IEEE Transactions on Control Systems Technology*, Vol. 26, No. 5, 2018, pp. 1771–1781. https://doi.org/10.1109/TCST.2017.2723864.

[26] MathWorks, "Robust Control Toolbox, version: 23.2 (R2023b)," , 2022. URL https://www.mathworks.com.

[27] Echols, J. A., Puttannaiah, K., Mondal, K., and Rodriguez, A., "Fundamental Control System Design Issues for Scramjet-Powered Hypersonic Vehicles," , 2015. https://doi.org/10.2514/6.2015-1760.

[28] Wu, N., and Yu, J., "Robust Controller Design of Hypersonic Vehicle in Uncertainty Models," *Proceedings of 3rd International Conference on Electromechanical Control Technology and Transportation (ICECT-2018)*, 2018, pp. 288–293.

[29] Zhu, Y., Shen, H., Liu, Y., Zhang, G., and Lu, Y., "Optimal control and analysis for aero-elastic model of hypersonic vehicle," *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, 2016, pp. 1911–1915. https://doi.org/10.1109/CGNCC.2016.7829081.

[30] Wenbiao, Z., Dong, L., Yong, M., and Yicheng, C., "Method of velocity controller design for an airbreathing hypersonic cruise vehicle," *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014, pp. 1817–1821. https://doi.org/10.1109/CGNCC.2014.7007458.

[31] Williams, J., and Vukelich, S., "The USAF stability and control digital DATCOM. Volume I. Users manual," , 1979.

[32] Alsuwian, T. M., Ordonez, R., and Jacobsen, L., "Adaptive Control for Longitudinal Dynamics of Hypersonic Vehicle at Subsonic Speeds," , 2017. https://doi.org/10.2514/6.2017-4009.

[33] MathWorks, "MATLAB, version: R2023b," , 2023. URL https://www.mathworks.com.

[34] MathWorks, "Simulink® Control Design, version: 23.2 (R2023b)," , 2022. URL https://www.mathworks.com.

[35] Zipfel, P. H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, 2nd ed., AIAA, 2000.

[36] Araki, J. J., "Reentry Dynamics and Handling Qualities of a Generic Hypersonic Vehicle," Thesis, Massachusetts Institute of Technology, 1992.

[37] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft control and simulation - dynamics, controls design, and autonomous systems*, John Wiley & Sons, 2015.

[38] Bates, D., and Postlethwaite, I., *Robust Multivariable Control of Aerospace Systems*, Vol. 8, IOS Press, 2002.

[39] Skogestad, S., and Postlethwaite, I., *Multivariable Feedback Control: Analysis and Design*, John Wiley & Sons, 2005.

[40] Choi, J., "Application of Hypersonic Vehicle Flying Qualities Criteria and Computational Considerations," Thesis, Massachusetts Institute of Technology, 1994.

[41] Seiler, P., Packard, A., and Gahinet, P., "An introduction to disk margins," *IEEE Control Systems Magazine*, Vol. 40, No. 5, 2020, pp. 78–95. https://doi.org/10.1109/MCS.2020.3005277.

[42] Kaminer, I., Pascoal, A. M., Khargonekar, P. P., and Coleman, E. E., "A velocity algorithm for the implementation of gain-scheduled controllers," *Automatica*, Vol. 31, No. 8, 1995, pp. 1185–1191. https://doi.org/https://doi.org/10.1016/0005-1098(95)00026-S.