ROBUST NONLINEAR CONTROL DESIGN FOR A HYPERSONIC AIRCRAFT USING SUM–OF–SQUARES METHOD

Armin Ataei-Esfahani* and Qian Wang
Department of Mechanical and Nuclear Engineering
The Pennsylvania State University
University Park, Pennsylvania 16802
Email: axa943@psu.edu , quw6@psu.edu

ABSTRACT
In recent years, Sum–Of–Squares (SOS) method has attracted increasing interest as a new approach for stability analysis and controller design of nonlinear dynamic systems. This paper utilizes SOS method to design a robust nonlinear controller for longitudinal dynamics of a hypersonic aircraft model. Specifically, the searching of the nonlinear robust controller is reformulated as a robust SOS/robust LMI problem, and then solved via a stochastic iterative algorithm. As the simulation results show, the designed controller is capable of stabilizing the aircraft and following pilot commands in presence of parametric uncertainties in the aircraft model.

NOMENCLATURE
\( \alpha \) angle of attack, rad
\( \beta \) equivalence ratio, \( \in [0, 20] \)
\( \delta E \) elevator deflection, rad \( \in [-2\pi/9, 2\pi/9] \)
\( \gamma \) flight-path angle, rad
\( \mu \) gravitational constant, \( 3.94 \times 10^{14} \) m\(^3\)/s\(^2\)
\( \rho_0 \) density of air, kg/m\(^3\)

1 INTRODUCTION
One of the most challenging research problems associated with nonlinear control designs involves addressing robustness issues, especially robustness with respect to uncertain parameters in a dynamic system. Here, we take the aircraft flight control as an example. In the past several decades, Nonlinear Dynamic Inversion (NDI) has become a popular nonlinear control approach for flight control since it directly utilizes the nonlinear structure of the aircraft model without linearizing it as the traditional gain-scheduling approach, and hence it is capable of handling possibly large nonlinearities. However, due to the nature of NDI on canceling nonlinearities directly, the resulting control design may lack robustness since the mis-canceled nonlinear terms (due to parameter uncertainties) could be significant enough that even high-gain linear control, which is often appended to the NDI controller, might not be able to handle them.

Lyapunov redesign, sliding mode control, and backstepping techniques are often used to address the robustness problems in nonlinear control. Lyapunov redesign methods are restricted to handling matched uncertainty only. It is also well known that the sliding mode control may exhibit chattering phenomena. The controller complexity for backstepping designs usually increases dramatically with the increase of the backstepping steps; furthermore, the backstepping controller could become conservative to...
the extent that the controller effort may easily saturate the actuators [1].

In this paper, we propose and develop a nonlinear robust control design based on the sum-of-squares (SOS) method, and then apply it to a hypersonic aircraft simulation model subject to uncertain parameters in the aerodynamic coefficients. The SOS decomposition method, as shown by Parrilo [2], is a convex problem and can be efficiently solved using semidefinite programming algorithms. Lyapunov–based analysis of stability regions using the SOS method has been studied in [3, 4], mostly through low-order nonlinear systems. In [5], SOS method was used for the stability and performance analysis of a quasi-LPV model for an F/A-18 aircraft.

One limitation in applying the SOS technique to stability analysis and control design for general classes of nonlinear systems is that the SOS technique is only applicable to polynomial vector fields whereas most physical systems are not represented by polynomials. One way to overcome this limitation is to reformulate the given nonlinear system into a quasi linear-parameter-varying system that has polynomial dependence on the state/scheduling variables. Along this line of research, for a spacecraft application, SOS method was applied to the performance improvement of nonlinear control systems in terms of $H_{\infty}$ control and adaptive control [6]. Another promising direction is to derive algebraic transformations that can transform a given nonlinear system into polynomial vector fields, e.g., polynomial universal formats in [7], which essentially recast the original nonlinear system into a higher dimensional system and thus reducing the complexity of nonlinearities by increasing the system dimension. Along this line of research, [8] has applied the sum-of-squares method to design a nonlinear controller for a hypersonic aircraft. In this paper, we extend our research results in [8] to address parameter uncertainties and design a robust nonlinear controller for the hypersonic aircraft model subject to uncertain parameters in aerodynamic coefficients.

In general, a SOS problem can be recast into a Linear Matrix Inequality (LMI) problem. In this paper, we present an efficient way to construct a robust LMI problem corresponding to a SOS problem with parameter uncertainties. For an aircraft control application, we reformulate the design of the robust nonlinear control into a robust SOS problem and then reduce it to a robust LMI problem. This direction provides the SOS–based robust nonlinear control design a promising edge over NDI control design for aircraft applications.

The paper is organized as follows. Section 2 introduces the hypersonic aircraft model. In Section 3, preliminaries on SOS, converting SOS to LMI, as well as robust SOS and robust LMI are given. Controller design is presented in Section 4, which includes the algebraic recasting of the nonlinear aircraft model, time scale decomposition and parametrization of controllers. Simulation results are given in Section 5 and Conclusions are drawn in the end.

2 HYPERSONIC AIRCRAFT MODEL

The hypersonic aircraft under study is a conical aircraft designed by National Aero-Space Plane (NASP) Program for which Shaughnessy et al. [9] simulated force and moment coefficients for different Mach numbers, angles of attack and control surface deflections. Later, Marrison and Stengel [10] fitted aerodynamic functions to the data at the cruising condition of Mach number 15, altitude of 33528 m (110,000 ft), airspeed of 4590 m/sec and angle of attack of 0.0312 rad. The dynamics equations for airspeed, flight-path angle, altitude, angle of attack and pitch rate for the longitudinal dynamics of the aircraft are given by:

\[ V = (T \cos \alpha - D)/m - \mu \sin \gamma / r^2 \]  
\[ \dot{V} = (L + T \sin \alpha)/(mV) - \left[(\mu - V^2 r) \cos \gamma \right]/(V r^2) \]  
\[ \dot{h} = V \sin \gamma \]  
\[ \dot{\alpha} = q - \dot{\gamma} \]  
\[ \dot{\dot{\gamma}} = M_{yy}/I_{yy} \]

In the above equations $r = h + R_E$ and $L, D, T$ and $M$ denote lift, drag, thrust and pitching moment respectively and are defined as,

\[ L = \bar{q} S C_L \]  
\[ D = \bar{q} S C_D \]  
\[ T = \bar{q} C_T \]  
\[ M_{yy} = \bar{q} \bar{e} [C_M(\alpha) + C_M(\dot{\delta} E) + C_M(q)] \]

where $\bar{q}$ is the dynamic pressure defined as $\bar{q} = \frac{1}{2} \rho_a V^2$.

$C_L$, $C_D$, $C_T$ and $C_M$ in Eq. (6) – Eq. (9) are aerodynamic force and moment coefficients. These coefficients along with parameters used in Eq. (1) – Eq. (9) are defined as follows [10]:

\[ \rho_a = 1.2266 \exp \left( \frac{-h}{7315.2} \right) \]  
\[ C_L = v_1 \alpha (0.493 + 1.91/M) \]  
\[ C_D = v_2 \times (0.0012 M^2 - 0.054 M + 1) \]  
\[ C_T = \begin{cases} v_3 3.565 (1 + 17/M)(1 + 0.15 \beta), & \text{if } \beta < 1 \\ v_3 5.30 (1 + 17/M)(1 + 0.15 \beta), & \text{if } \beta \geq 1 \end{cases} \]  
\[ C_M(\alpha) = v_4 10^{-4} (0.06 - e^{M^{1/3}}) \times (-6565 \alpha^2 + 6875 \alpha - 1) \]

\[ \text{The thrust coefficient } (C_T) \text{ in this paper is slightly different from the one in } [10], \text{ where } C_T \text{ was modeled as an explicit function of the angle of attack. The thrust coefficient in Eq. (13) is defined based on equivalence ratio, } \beta = \frac{m_f}{m_a}, \text{ where } m_f \text{ is the rate of the mass of the fuel and } m_a \text{ is the rate of the mass of air entering the engine. Since the mass of air entering the engine is a function of angle of attack, the effect of changing angle of attack is implicitly included in } C_T \text{ in Eq. (13). For the very same reason, the original data reported by [9] is only a function of Mach number and equivalence ratio.} \]
\[ C_M(q) = \nu_5 \frac{\bar{c}}{2V} q (-0.025 M + 1.37) \times (-6.83 \alpha^2 + 0.303 \alpha - 0.23) \quad (15) \]

\[ C_M(\delta E) = \nu_6 0.0292 (\delta E - \alpha) \quad (16) \]

In the above equations, \( M \) is the Mach number defined as \( M = V/a \) where the speed of sound is given by,

\[ a = (2.9495 \times 10^{-3} h^2 - 9.16 \times 10^{-4} h + 303.58). \]

the parameters \( v_i (i = 1 \ldots 6) \) in the above equations are uncertainty parameters with the nominal values \( v_i = 1 \).

**Remark:** Note that the lifting effect of elevator control deflection is not included in the lift-coefficient model. Wang and Stengel in [11] have commented on the non–minimum–phase–zero effect due to the elevator lifting effect and have shown that any elevator control deflection derived by inverting the transfer function corresponding to the lifting effect is unstable. In this paper, however, the derivation of the elevator control does not include the inversion of the elevator lifting effect and thus will not cause instability due to the non–minimum–phase–zero. As a result, the only concern about using the simplified model of \( C_L \) in Eq. (11) is how it affects the aircraft’s stability and performance in simulations. A close examination of the lift–coefficient data in the NASA report [9] reveals that for the trimmed cruise condition \((M = 15, h = 33528 \text{ m})\), the lift–coefficient value contributed by elevator deflection is consistently less than 5\% of the lift–coefficient value contributed by the basic vehicle (which is given by the lift–coefficient model in this paper). Hence, we consider it safe to use the simplified lift–coefficient model without the elevator lifting effect.

For computational convenience, airspeed and altitude are scaled in the rest of this paper as follows:

\[ q' = 10^{-2} q \quad \text{and} \quad \mathcal{H} = 10^{-3} h. \quad (17) \]

At the trimmed cruise condition, where \( q' = 4.59 \text{ km/s} \), \( \mathcal{H} = 33.530 \text{ km} \), \( \alpha = 0.0310 \text{ rad} \), \( \beta = 0.3209 \) and \( \delta E = -0.00654 \text{ rad} \), the linearized model of the nominal system has an unstable short–period mode and an unstable height mode. The design goal is to stabilize the system and achieve good performance in both altitude and velocity command responses.

### 3.1 Preliminaries

**Definition 1.** A real polynomial \( p(x) = p(x_1, x_2, \ldots, x_n) \) is a sum of squares if there exists polynomials \( p_1(x), p_2(x), \ldots, p_m(x) \) such that

\[ p(x) = \sum_{i=1}^{m} p_i^2(x), \quad (18) \]

Equation (18) implies that the existence of a SOS decomposition is a sufficient condition for \( p(x) \geq 0 \) for all \( x \in \mathbb{R}^n \). Alternatively, the SOS decomposition in Eq. (18) can be written as

\[ p(x) = Z^T(x) Q Z(x), \quad (19) \]

where \( Q \) is a positive (semi)definite matrix and \( Z \) consists of the monomials in \( x \) whose degree is less or equal to half the degree of \( p(x) \). To show that Eq. (18) and Eq. (19) are equivalent, we take the Cholesky decomposition of \( Q = L^T L \), where \( L = \sqrt{K} U^T \) and \( A \) and \( U \), are eigenvalues and eigenvectors of \( Q \) respectively. Replacing \( Q = L^T L \) in Eq. (19), we get \( p(x) = \sum_i ([LZ(i)]_i^2) \), where \( (LZ)(i) \) corresponds to the \( i^{th} \) element of the \( LZ \) vector.

**Remark:** Note that the lifting effect of elevator control deflection is not included in the lift-coefficient model. Wang and Stengel in [11] have commented on the non–minimum–phase–zero effect due to the elevator lifting effect and have shown that any elevator control deflection derived by inverting the transfer function corresponding to the lifting effect is unstable. In this paper, however, the derivation of the elevator control does not include the inversion of the elevator lifting effect and thus will not cause instability due to the non–minimum–phase–zero. As a result, the only concern about using the simplified model of \( C_L \) in Eq. (11) is how it affects the aircraft’s stability and performance in simulations. A close examination of the lift–coefficient data in the NASA report [9] reveals that for the trimmed cruise condition \((M = 15, h = 33528 \text{ m})\), the lift–coefficient value contributed by elevator deflection is consistently less than 5\% of the lift–coefficient value contributed by the basic vehicle (which is given by the lift–coefficient model in this paper). Hence, we consider it safe to use the simplified lift–coefficient model without the elevator lifting effect.

For computational convenience, airspeed and altitude are scaled in the rest of this paper as follows:

\[ q' = 10^{-2} q \quad \text{and} \quad \mathcal{H} = 10^{-3} h. \quad (17) \]

At the trimmed cruise condition, where \( q' = 4.59 \text{ km/s} \), \( \mathcal{H} = 33.530 \text{ km} \), \( \alpha = 0.0310 \text{ rad} \), \( \beta = 0.3209 \) and \( \delta E = -0.00654 \text{ rad} \), the linearized model of the nominal system has an unstable short–period mode and an unstable height mode. The design goal is to stabilize the system and achieve good performance in both altitude and velocity command responses.

### 3.2 SOS Design Problem and Its LMI Formulation

The SOS analysis problem of Eqs. (18) and (19) can be extended to a SOS design problem by introducing decision variables \( x_c = [x_{c_1}, x_{c_2}, \ldots, x_{c_r}]^T \) into the polynomial \( p(x) \). In this case, the objective is to find \( x_c \), if it exists, such that \( p(x, x_c) \) is SOS. To this end, assuming \( p(x, x_c) \) is linear with respect to \( x_c \), the SOS decomposition of \( p(x, x_c) \) in Eq. (19) can be reformulated into a feasibility problem of the following SemiDefinite Programming (SDP) problem [2],

\[ A_{xdp}^T x_{dp} = b, \quad x_{dp} = [x_c, x_q]^T, \quad x_c \in \mathbb{R}^r, x_q \in \mathcal{K} \quad (21) \]

where \( x_q \) is a \( n^2 \times 1 \) vector of the elements of \( Q \) and \( \mathcal{K} \) denotes a semidefinite cone. Considering the symmetric property of the
matrix $Q$, a reduced form of the SDP problem in Eq. (21) can be derived by removing the symmetric elements of $Q$ from $x_{sdp}$ and rewriting Eq. (21) as,

$$A^T x = b, x = [x_c, \bar{x}_q]^T, x_c \in \mathbb{R}^r, \bar{x}_q \in \mathcal{K} \quad (22)$$

where $\bar{x}_q$ is a vector of length $n_q(n_q+1)/2$. In the event that there are multiple SOS conditions present in a SOS problem, we let $Q_{sos,i}$ denote the decomposition matrix $Q$ corresponding to the $i^{th}$ SOS condition, and then stack all the $Q_{sos,i}$ diagonally to form the overall $Q$ matrix in Eq. (19). In this case, the length of $x_q$ will be $\sum n_{sos,i}^2 < n_q^2$ where $n_{sos,i}$ is the size of the $Q_{sos,i}$ and $\bar{x}_q$ will be a vector of $\sum n_{sos,i}(n_{sos,i}+1)/2$ elements. In this paper, we use YALMIP [12], which is a MATLAB toolbox for convex optimization, to generate $A_{sdp}$ and $b_{sdp}$ matrices from the SOS problem formulation in Eq. (18).

The LMI formulation in Eq. (20) can be derived from the SDP problem in Eq. (22) using the following procedure. First, using the Row Reduced Echelon Form (RREF) of the augmented matrix $M = [A|b]$, we can identify the leading elements (pivots) in $x$ [13]. The remaining elements of $x_c$ and $\bar{x}_q$ then constitute the free variables ($\zeta$) in Eq. (20). To find $Q_0$ and $Q_i$ in Eq. (20), we first, remove the leading variables in the RREF, and then solve for elements of $Q_0$ and $Q_i$ using $\zeta$. In practice, there are far more variables than equations ($n >> m$) and therefore, for a given SOS problem, there exist either an infinite number of solutions or no solution at all (i.e. $p(x)$ is not SOS).

### 3.3 Robust SOS/LMIs

When the SOS problem includes parameter uncertainty, i.e. $p(x, \Delta) = \sum p_i^2(x, \Delta)$, the following robust LMI problem can be formed,

$$Q(\zeta, \Delta) = Q_0(\Delta) + \sum \zeta_i Q_i(\Delta). \quad (23)$$

where $\Delta$ denotes bounded uncertain parameters in the system. Furthermore, we assume that $Q_0(\Delta)$ and $Q_i(\Delta)$ have affine dependence on each of the uncertain parameters.

To form $Q(\zeta, \Delta)$, we first define the LMI corresponding to the nominal SOS problem as described in the previous section and find the nominal values of $Q_0$ and $Q_i$. Then, to account for the effect of uncertain parameters, we change one uncertain parameter at a time to a desired value (while leaving the rest at their nominal values) and record the new matrices $Q_0$ and $Q_i$ due to this change. By subtracting these new matrices $Q_0$ and $Q_i$ from their nominal counterparts and then dividing by the value of the uncertain parameter, we can find the effect of each uncertainty on the LMI problem of Eq. (23). Based on this information, we can then construct $Q_0(\Delta)$ and $Q_i(\Delta)$ for a given set of uncertainty values.

For a medium size robust SOS problem, solving the resulting robust LMI subject to uncertain parameters can be computationally very challenging for the existing LMI solvers. Therefore, in this paper, we propose to apply stochastic iterative algorithms, which allow to search the solution to uncertain LMIs in an iterative way rather than solving a large number of LMIs simultaneously [14]. By applying a stochastic gradient method such as ellipsoid algorithm, we can minimize the norm of the projection of $Q(\zeta, \Delta)$ on the non-positive definite cone [14], i.e. we choose the cost function to be

$$\Gamma(\zeta, \Delta) = \|\Pi^+ Q(\zeta, \Delta)\|_F, \quad (24)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\Pi^+$ denotes the projection on the non-positive cone, which can be computed by

$$\Pi^+ = U \text{diag}(\lambda^-_0, \ldots, \lambda^-_n) U^T.$$ 

In the above equation $\lambda^-_0 = \min(0, \lambda_0)$ and $\lambda_i$ and $U$ correspond to eigenvalues and eigenvectors of $Q$ respectively.

Note that the optimal solution to the cost function in Eq. (24) satisfies $\Gamma(\zeta, \Delta) = 0$, which implies that $Q(\zeta, \Delta) > 0$. One advantage of using ellipsoid algorithm over other stochastic optimization algorithms is that it is guaranteed to converge to a solution in a finite number of iterations with probability one. As we can see the LMI approach can be used to solve a nominal SOS problem as well as a robust one without adding much complexity to the control design process.

### 4 CONTROLLER DESIGN USING SOS METHOD

In this section, a robust nonlinear controller for the hypersonic aircraft model subject to uncertain parameters is developed using the robust SOS method. A nominal control design for the same model was given in our previous paper [8], where we parameterized the controller such that stabilization of the nonlinear aircraft model was cast as an SOS problem. For the robust design presented in this paper, we adopt a controller structure and a controller parameterization method similar to the nominal design in [8]. However, we search the controller parameters to minimize the cost function in Eq. (24) for a robust LMI which is derived from a robust SOS problem that has incorporated the uncertain parameters in the aerodynamic coefficients of the aircraft model.

#### 4.1 Dual to the Lyapunov Theorem

It is well known that for a general nonlinear system, $\dot{x} = f(x) + g(x)u$, it is difficult, if not impossible, to search for a controller $u(x)$ and Lyapunov function, $V(x)$, at the same time. This is mainly due to that the set $(V, u)$, which is required to satisfy the derivative condition in the Lyapunov’s stability theorem, \[ \frac{dV}{dx} (f(x) + g(x)u) \leq 0, \] is not convex. Prajna et al. [15] proposed a so-called density function $\rho(x)$, which can be interpreted as a
dual to the Lyapunov function and can transform the problem of searching for $\rho(x)$ and $u(x)$ into a convex problem.

**Theorem 1 [15].** Given the equation $\dot{x} = f(x) + g(x)u$, where $(f + gu)(x) \in C^1(\mathbb{R}^n, \mathbb{R}^n)$ and $(f + gu)(0) = 0$, suppose there exists a non–negative function $\rho(x) \in C^1(\mathbb{R}^n - \{0\}, \mathbb{R})$, such that $\rho(x)(f + gu)(x)/|x|$ is integrable on $x \in \mathbb{R}^n : |x| \geq 1$ and

$$\nabla \cdot [(f + gu)\rho](x) > 0 \quad \text{for almost all } x,$$

(25)

where $\nabla \cdot f = \sum_{i=1}^n \frac{\partial f_i}{\partial x_i}, f : \mathbb{R}^n \to \mathbb{R}^n$. Then, for almost all initial states, $x(t) \to 0$ as $t \to \infty$.

To jointly search for the density function $\rho(x)$ and the controller $u(x)$, consider the following parametrization,

$$\rho(x) = \frac{p(x)}{t(x)^2}, \quad u(x) = \frac{w(x)}{p(x)},$$

where $\rho(x)$, $t(x)$ and $w(x)$ are polynomials and $\tau$ is chosen to satisfy the integrability condition in Theorem 1. By choosing $t(x)$ to be positive, the condition Eq. (25) reduces to

$$r \nabla \cdot (fp + gw) - \tau \nabla t \cdot (fp + gw) > 0,$$

(26)

where $\nabla t = \left[ \frac{\partial t}{\partial x_1}, \ldots, \frac{\partial t}{\partial x_n} \right], t : \mathbb{R}^n \to \mathbb{R}$.

When $f(x)$ and $g(x)$ are polynomial vector fields, a sufficient condition for satisfying Eq. (26) can be derived by applying the SOS technique, i.e. we rewrite Eq. (26) as,

$$[r \nabla \cdot (fp + gw) - \tau \nabla t \cdot (fp + gw)] - \phi(x) \text{ is SOS},$$

(27)

where $\phi(x)$ is a SOS polynomial.

In addition, if the control system is subject to constraints $l_i(x) = 0, l_2(x) = 0, \ldots, l_m(x) = 0$, it follows from the application of the Positivstellensatz Theorem that these constraints can be incorporated into Eq. (27) as,

$$[r \nabla \cdot (fp + gw) - \tau \nabla t \cdot (fp + gw)] - \phi(x) + \sum_{i=1}^m \kappa_i(x)l_i(x) \text{ is SOS},$$

where $\kappa_i(x)$ ($i = 1 \ldots m$) are polynomials in $\mathbb{R}$. [2]

Considering an uncertain polynomial systems, $\dot{x} = f(x, \Delta) + g(x, \Delta)u$, and a common density function $\rho(x)$ for all the uncertainties (i.e., neither $p(x)$ nor $t(x)$ is considered to be a function of uncertainty $\Delta$), we can write the above equation as a robust SOS problem defined as:

$$[r \nabla \cdot (f(x, \Delta)p + g(x, \Delta)w) - \tau \nabla t \cdot (f(x, \Delta)p + g(x, \Delta)w)]$$

$$- \phi(x) + \sum_{i=1}^m \kappa_i(x)l_i(x, \Delta) \text{ is SOS}. \quad (28)$$

### 4.2 Two–Time Scale Decomposition

To reduce the computational burden, a two–time–scale decomposition of the aircraft model is used for the nonlinear control design. Through this time–scale decomposition, the system dynamics is broken into slow and fast time–scales, denoted by $x_s$ and $x_f$ respectively. When dealing with the slow dynamics, it is assumed that fast–dynamics states are at their equilibrium condition, and when dealing with the fast dynamics, it is assumed that slow–dynamics states remain constant (but not necessarily at the equilibrium condition), i.e. $x_s \equiv 0$. For each time scale, a controller is designed using the SOS method.

The two time–scales can be identified by looking at the eigenvalues and eigenvectors of the linearized system. For the aircraft system under study, we consider airspeed, flight–path angle and altitude to constitute the slow–dynamics since they are more dominant in phugoid and height modes. Angle of attack and pitch rate on the other hand play the major role in the short period mode and therefore are considered in the fast–dynamics time–scale. Note that in the design of the slow–dynamics controller, the angle of attack is treated as a control variable and it is denoted as the angle of attack command, $\alpha_c$.

Figure 1 shows the overall structure of the aircraft controller. The slow–dynamics controller, designed in Section 4.3, takes the altitude and airspeed command values ($h_c$ and $V_c$ respectively) and derives the engine equivalence ratio, $\beta$, as well as the angle of attack command value, $\alpha_c$. The commanded angle of attack is then used as input for the fast–dynamics controller (given in Section 4.4), which subsequently determines the elevator deflection angle, $\delta E$.

![Controller structure of the two-time scale controller](image)

**Figure 1.** Controller structure of the two-time scale controller

Following notations will be used throughout the rest of this paper. The superscript $*$ is used to denote the trim condition of a state or control variable and a (·) over a state or control variable represents the shift of that variable from its trim condition to the origin, e.g. $\bar{\alpha} = \alpha - \alpha^*$, where $\alpha^*$ is the trim angle of attack. The trim values for state and control variables are given in Section 2. Subscripts $s$ and $f$ are used to denote slow– and fast–dynamics time–scales respectively.

### 4.3 Slow–Dynamics Controller Design

Recall that the SOS control design technique of Section 4.1 can only be applied to polynomial systems. The aircraft model
however, includes some non–polynomial nonlinearities that must be dealt with before a controller can be designed. The following section specifies assumptions and approximations made for slow–dynamics system to transform it into a polynomial system.

4.3.1 Approximations and Assumptions As mentioned in Section 4.2, the control variables for slow dynamics consist of engine equivalence ratio, $\beta$, and angle of attack command, $\alpha$. To have an affine slow–dynamics system with respect to $\beta$ and $\alpha$, some aerodynamic coefficients need to be approximated. First, the term $C_{D,\alpha} = (171\alpha^2 + 1.15\alpha + 2)$ in Eq. (12) is replaced with a linear function of $\alpha$, which is calculated using the least squares method. Since generally small angles of attack are used throughout the flight, the least squares is computed such that the error is minimized over $\alpha \in [1, 3]$, which yields $C_{D,\alpha} = 13.0882\alpha + 1.809$.

Trigonometric terms $\sin \alpha$ and $\cos \alpha$ are also approximated with 0 and 1 respectively, assuming $\alpha$ remains small. Small angles of attack are enforced by adding a saturator, which bounds the angle of attack command between $-5^\circ$ and $15^\circ$. Since the flight–path angle also remains small throughout the flight, we approximate $\sin \gamma$ and $\cos \gamma$ with $\gamma$ and 1 respectively.

Next, the exponential term in the air density equation Eq. (10) is approximated using the Taylor series around the trim condition, and is given by,

$$\rho_a = 0.0125 - 0.0171(H - H^*) + 0.0117(H - H^*)^2,$$

Finally, the non–polynomial term $(1/r)$ is simplified by assuming that it is constant and is evaluated at the trim condition $(1/r^*)$. The error due to this assumption is small enough not to have any tangible impact on the overall dynamics of the system since $|1/r - 1/r^*| < 10^{-8}$ $\forall \eta < 10^8$ m.

4.3.2 Transforming the nonlinear aircraft model into polynomial vector fields The only remaining non–polynomial nonlinearity in the system is $1/\psi^r$, which must be recasted into polynomial form. To this end, we define an auxiliary state variable,

$$x_{aux} = 1/\psi^r.$$

The derivative of the above auxiliary variable is given by $\dot{x}_{aux} = -\dot{\psi}x_{aux}^2$, which in this case is a polynomial in terms of the original state variables and the new auxiliary variable. Had the derivative of the auxiliary variable included any additional non–polynomial terms, the same procedure could have been applied by defining those non–polynomial functions as new auxiliary states and repeating the same process until the overall augmented system was represented by polynomials only.

Applying the approximations of the previous section, the original slow–dynamics equations can now be written in terms of the scaled variables of Eq. (17) and the new augmented state variable $x_{aux}$ as,

$$\dot{x} = \frac{T - D}{10^{-2}} - \frac{\mu x_1}{10^{-2}r^2},$$

$$\dot{\gamma} = \frac{L}{m}x_3 - (\mu - 10^{-4}\gamma^2 r^2)x_2x_3,$$

$$\dot{\dot{\psi}} = 0.1 \dot{\psi}^2,$$

$$x_{aux} = -\dot{\psi}x_{aux}^2.$$

The above system can be represented in vector space as $\dot{x}_s = f_s(x_s) + g_s(x_s)u_s$, where

$$x_s = [\psi, \gamma, H, x_{aux}]^T, \quad u_s = [\alpha_c, \beta]^T,$$

and $f_s(x_s)$ and $g_s(x_s)$ are polynomial vector fields defined accordingly. Note that the above system is subject to the constraint $x_{aux} \psi^r - 1 = 0$, which is imposed by defining $l_1(x) = x_{aux} \psi^r - 1$ and incorporating it into the SOS problem as shown in Eq. (28).

4.3.3 Parametrization of the SOS Controller Since Eq. (28) only applies to systems with an equilibrium point at the origin, we shift all slow–dynamics variables from their trim conditions to the origin. Following the notations introduced in Section 4.2, these shifted state and control variables are denoted by $\hat{x}_s$ and $\hat{\psi}, \hat{\beta}, \hat{\alpha}_c$ respectively.

Next, we choose $t_s(\hat{x}_s)$ to be positive for all $\hat{x}_s \neq 0$. A good choice for such $t_s(\hat{x}_s)$ is the control Lyapunov function for the nominal system linearized at the trim condition. Since the linearized system has a polynomial form, it does not require any auxiliary variable and can be written as,

$$\dot{\hat{x}}_r = A_s\hat{x}_r + B_s\hat{u}_r, \quad \hat{x}_r = [\hat{\psi}, \hat{\gamma}, \hat{H}]^T, \quad \hat{u}_r = [\hat{\alpha}_c, \hat{\beta}]^T.$$

We now define $t_s(\hat{x}_s) = \hat{x}_r^T P_s \hat{x}_r$, where $P_s$, given below, is the solution to the Riccati equation of the linearized system $(A_s, B_s)$ with $Q = I_3$ and $R = I_2$ ($I_n$ denotes a $n \times n$ identity matrix)

$$P_s = \begin{bmatrix} 120.4521 & 32.8668 & -1.8330 \\ 32.8668 & 112.1825 & 20.5251 \\ -1.8330 & 20.5251 & 9.9068 \end{bmatrix}.$$

Next, we search for polynomials $p_s(\hat{x}_s)$ and $w_s(\hat{x}_s)$ in Eq. (28). We start by assuming that $p_s(\hat{x}_s)$ is a constant, and set $p_s(\hat{x}_s) = 1$. This will lead to a polynomial controller $u(\hat{x}_s)$ instead of a rational one. Since the system has two inputs, the input vector is denoted by $w_s(\hat{x}_s) = [w_1(\hat{x}_s), w_2(\hat{x}_s)]^T$, where $w_1(\hat{x}_s)$ and $w_2(\hat{x}_s)$ are fourth–degree polynomials in terms of the slow–dynamics state variables, $\hat{x}_s$. In Eq. (28), $\tau$ is set to 6, which is
large enough to satisfy the integrability condition in Theorem 1 and $\phi_s(x)$ is chosen to be a second-degree polynomial of slow-dynamics variables.

To design the robust controller, we plug the above variables into Eq. (28). Then, using the results of Section 3.2, we find the LMI problem corresponding to the nominal system ($v_1 = v_2 = v_3 = 1$). Following the procedure in Section 3.3, we can find how each uncertainty variable affect the nominal LMI and combine those effects to get the robust LMI in Eq. (23). For the slow-dynamics, the robust LMI problem has 2696 free variables and $Q_i(i = 0...2696)$ are $80 \times 80$ matrices. In the ellipsoid algorithm of Section 3.3, we start from the solution for the nominal system and assume that uncertainties are uniformly distributed over the range of 0.9 and 1.1. The resulting robust controller is then given by,

$$\dot{\alpha}_c = 0.76 \dot{p} - 5.25q - 0.72 \dot{H} + 0.12 \dot{\xi}_{aux} - 0.13 \dot{\gamma}^2 + 0.11 \dot{\psi} \dot{\gamma} - 0.01 \dot{\psi} \dot{\gamma}^2$$

Taking the absolute value of both commands and reached their command values. Figure 2 shows minimum and maximum trajectories (dotted lines) along with the trim condition and all simulations are performed in the same steps as in slow-dynamics, we can formulate the robust LMI corresponding to the fast-dynamics. By solving the resulting LMI problem using the ellipsoid algorithm, we get:

$$\delta E = -2.036\alpha - 1.98q + 0.59\alpha^2 + 0.01q - 1.997\alpha q - 0.025q^2 - 41.7\alpha^3 - 0.91\alpha q^2 - 0.15q^3$$

4.4 Fast-Dynamics Controller Design

As discussed in Section 4.2, the fast-dynamics time-scale includes dynamics of $\alpha$ and $q$. Recall that while working with fast-dynamics time-scale, all variables corresponding to slow-dynamics are assumed to be constant. Therefore, $\dot{\gamma}$ in Eq. (4) can be set to zero. Assuming that all other slow-dynamics variables are at their trim condition, we can write the fast-dynamics equations of motion as:

$$\dot{\alpha} = q$$

$$\dot{q} = 113.5891 \left[ 5.33 \times 10^{-6} (-6565\alpha^2 + 6875\alpha - 1) + 0.0026q(-6.83\alpha^2 + 0.303\alpha - 0.23) + 0.0292(\delta E - \alpha) \right]$$

which is a polynomial vector field and affine with respect to $\delta E$.

The above equations can be written in the vector space form $\dot{x}_f = f_f(x_f) + g_f(x_f)u_f$, where $x_f = [\alpha, q]^T$ and $u_f = \delta E$. Note that similar to slow-dynamics, all fast-dynamics variables must be shifted to the origin before Theorem 1 can be applied. For fast-dynamics however, $\alpha$ has to approach $\alpha_c$ and not its trim condition and thus $\dot{\alpha}$ is defined as $\dot{\alpha} = \alpha - \alpha_c$. Similar to the previous section, we define $f_f(x_f) = \hat{x}_f P_f \dot{x}_f$ where $P_f$ is the solution of the Riccati equation corresponding to the nominal linearized fast-dynamics system with $Q = \text{diag}([100, 1])$ and $R = 1$ and it is given by

$$P_f = \begin{bmatrix} 26.7282 & 3.0696 \\ 3.0696 & 0.7994 \end{bmatrix}.$$
remained stable and was still able to follow commands. However, on average it took slightly more time for the trajectories to converge to the pilot commands.

![Graphs](image1.png)

Figure 2. Stochastic performance envelope of the robust controller for pilot commands based on 10,000 Monte Carlo simulations of the uncertain system

6 CONCLUSION

In this paper, we formulate the robust nonlinear control of a hypersonic aircraft model subject to aerodynamic parametric uncertainties into a robust SOS problem. We then use the SOS problem to derive a set of robust LMIs, which can be solved using a stochastic iterative algorithm. As it is shown through simulations, the designed controller is perfectly able to stabilize the system and follow the pilot commands over the uncertainty range for which it was designed.

ACKNOWLEDGMENT

This work was partially supported by National Science Foundation under Grant 0751299.

REFERENCES


