

Linear Parameter-Varying Control Compatible with Excitation for Real-Time Modeling

Luke J. Miller *

NASA Langley Research Center Hampton, Virginia, 23681

The Modeling and Control for Agile Aircraft Development project (MCAAD) seeks to reduce the time spent on wind tunnel testing and computational fluid dynamic modeling when developing control laws for new commercial and military aircraft. MCAAD uses advanced dynamic modeling procedures coupled with real-time control design which can reduce the time and cost spent on traditional aircraft development methods. This is done in-flight by using a real-time modeling procedure where programmed test inputs (PTI) deflect the aircraft control surfaces, and the responses are measured in order to create a dynamic model. This paper presents an H_∞ -based linear parameter-varying (LPV) control law designed on a nonlinear rigid-body dynamic model which commands control surface deflections at frequencies above the PTI frequency band. Thus the controller will not suppress the response due to the PTI. The LPV controller is compared with a previously designed Nonlinear Dynamic Inversion (NDI) controller and the coherence between the commanded control surface deflections and the PTI is plotted for both controllers. The LPV controller is shown to distort the PTI less than the NDI controller.

I. Nomenclature

b	=	wing span, ft
$\bar{c}$	=	wing mean aerodynamic chord, ft
C_l, C_m, C_n	=	body-axis nondimensional aerodynamic moment coefficients
g	=	acceleration due to gravity, ft s^{-2}
$I_{xx}, I_{yy}, I_{zz}, I_{xz}$	=	inertia tensor elements, slug ft^2
K	=	control gain
m	=	mass, slug
M	=	vector of aerodynamic moments, lb ft
M_u	=	vector of aerodynamic moments due to control surface deflection, lb ft
$\omega_1, \omega_2, \omega_3$	=	body-axis roll, pitch, and yaw rates, rad s^{-1}
$\bar{q}$	=	dynamic pressure, lb ft^{-2}
S	=	wing reference area, ft^2
V	=	true airspeed, ft s^{-1}
V_{cas}	=	calibrated airspeed, ft s^{-1}
Th	=	Throttle, lb
α	=	angle of attack, rad
β	=	sideslip angle, rad
r	=	control system tracking command
u	=	control surface deflection, rad
ζ	=	damping ratio
μ, γ, χ	=	bank, flight path, and track angles, rad
ϕ, θ, ψ	=	Euler roll, pitch, and yaw angles, rad
ω	=	body-axis angular rate, rad s^{-1}
ω_n	=	natural frequency, rad s^{-1}
y	=	plant output
z	=	output of weighting filter

*Research Aerospace Engineer, Flight Dynamics Branch, AIAA Member

Subscripts

cmd	=	command
err	=	error
K	=	controller
lf, rf	=	left and right flap
la, ra	=	left and right aileron
e	=	elevator
ru	=	rudder
i	=	inner loop controller
o	=	outer loop controller
p	=	prefilter

Superscripts

$\hat{}$ = estimate

II. Introduction

Stability and reference tracking are paramount concerns to a control system designer. However, in order to ensure stability and tracking performance, an accurate model of the aircraft system must be available. One of the goals of the NASA Modeling and Control for Agile Aircraft Development (MCAAD) project is to develop an aircraft model and aircraft control system in real time when there is no prior knowledge of the aircraft dynamics. This real-time modeling and control design procedure has the potential to greatly reduce the development time and cost of commercial and military aircraft.

Real-time model identification and real-time control design has been considered before to study actuator failure cases, as flown on the VISTA/F-16 aircraft [1]. Additionally, identification schemes using prior information to enhance parameter estimates in a recursive least-squares formulation have been used [2]. Reconfigurable control allocation in the presence of actuator failure has also been well studied [3] [4]. The NASA Learn-to-Fly (L2F) project explored simultaneous real-time control design and real-time aerodynamic modeling aimed at reducing time spent on traditional aircraft development methods. The Learn-to-Fly project used baseline Nonlinear Dynamic Inversion (NDI) control laws, with adaptive $\mathcal{L}1$ augmentation which was developed by Snyder et al. [5], [6], [7]. The control architecture used in L2F is depicted in fig. 1. The real-time aerodynamic modeling under L2F was developed by Morelli [8].

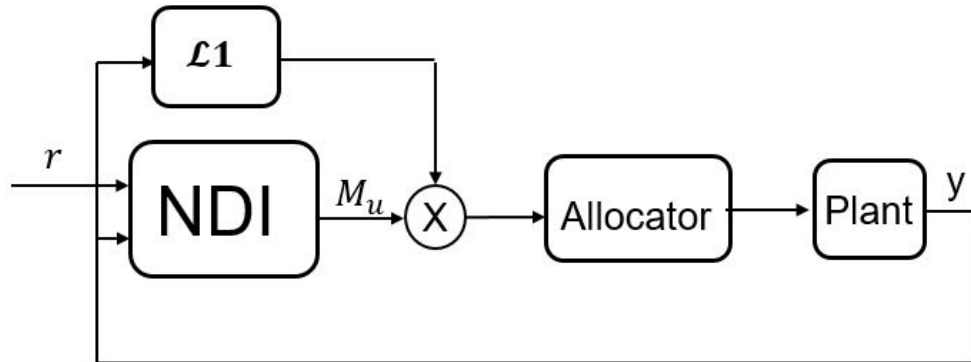


Fig. 1 Control Architecture used in the NASA Learn-to-Fly project.

The L2F project successfully demonstrated the feasibility of simultaneous control design and real-time aerodynamic modeling. However, the NDI control law incorporated the aerodynamic modeling parameters which are learned in

real-time and are subject to large initial error. This could cause the aircraft to behave in undesirable and unpredictable ways.

This paper proposes a framework of an alternative solution to the L2F problem, performed under the NASA MCAAD project and presents initial results, shown to be more compatible with excitation necessary for real-time modeling. This solution seeks to remove the influence of the modeling results on the control system by incorporating the real-time aerodynamic modeling results exclusively in the control allocator. Figure 2 shows the influence of the modeling on the control system, experienced under the L2F project. Figure 3 shows the proposed removal of the influence of the modeling on the control under the MCAAD project.

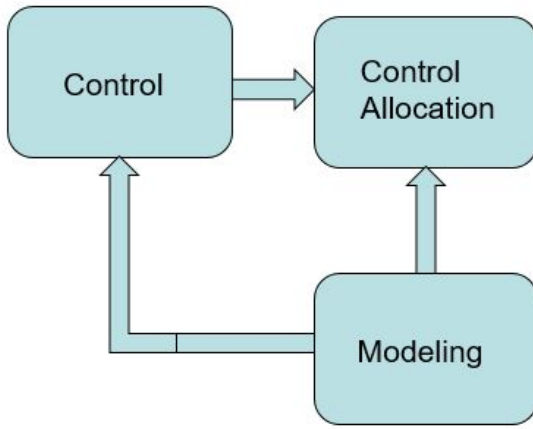


Fig. 2 Real-time modeling influence on real-time control design under L2F.

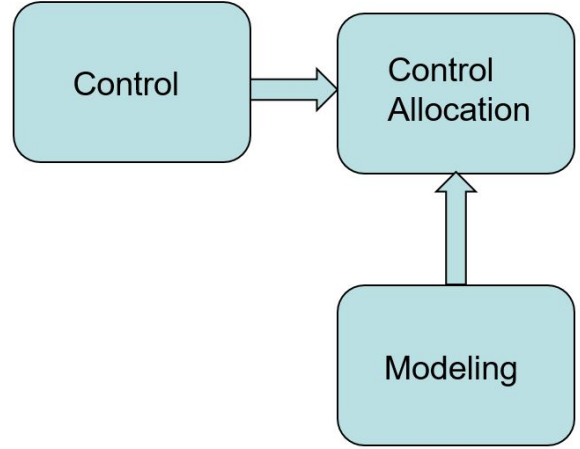


Fig. 3 Proposed removal of real-time modeling influence on real-time control design under MCAAD.

The full extent of the proposed control architecture is shown in fig. 4. Successfully removing the influence of the aerodynamics on the control system will allow the control designer to only consider rigid-body equations of motion, which will greatly simplify the controls problem. This paper shows that an H_∞ -based linear parameter-varying (LPV) control law can be used to shape the response of the rigid-body equations of motion into a more compatible response with excitation for real-time modeling than that of the NDI control law used in the L2F project while, establishing uniform global exponential stability of the rigid body dynamics.

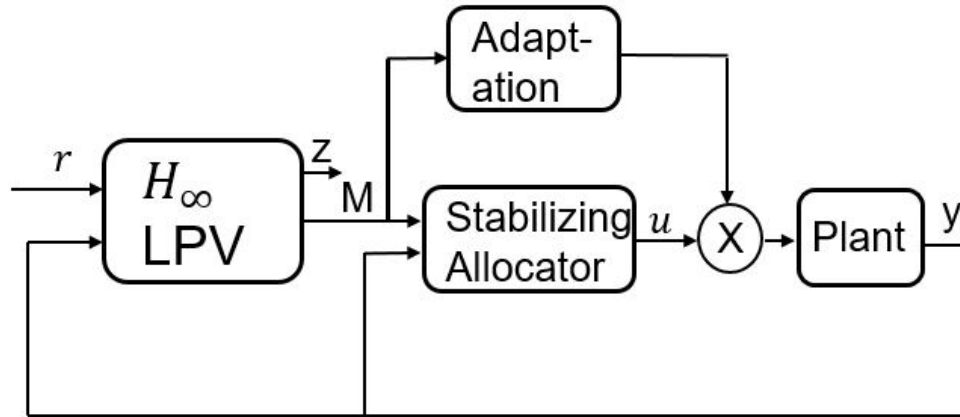


Fig. 4 Proposed comprehensive control architecture to be developed under NASA MCAAD project.

Successful development of the control architecture shown in fig. 4 will entail the implementation of closed-loop, exponentially stabilizing control allocation laws. Johansen showed that this was possible through a control-Lyapunov

design approach which derived an optimal control allocation algorithm which guarantees uniform boundedness and uniform global exponential convergence in conjunction with an exponentially stable trajectory-tracking controller [9]. The H_∞ -based LPV controller presented in this paper can serve as the exponentially stable trajectory tracking controller necessary for the development of closed-loop stabilizing control allocators.

Under the proposed control architecture in fig. 4 the control allocation is designed based on aerodynamic modeling parameters, which are learned in real-time and are subject to large initial error. Due to this error, the stabilizing allocator might not realize the exact moment commanded by the LPV controller. This could cause undesirable and unpredictable behavior in the aircraft. To address this problem, an adaptive control allocation element will be used to adjust control surface deflection commands. A control-Lyapunov approach can be used to derive these adaptive control allocation laws and is shown to prove closed-loop stability [10].

In this paper an LPV control design methodology is presented based on rigid-body equations of motion and is shown to be more compatible with real-time modeling methods than the NDI control developed under the L2F project. Additionally, this control design using a rigid-body model ensures robustness and exponential stability [11] which Johansen shows is necessary for the development of any future stabilizing allocators [9]. This technique could also be applied to aircraft models subject to more complex dynamic effects, such as slosh or flex modes. The LPV technique is an alternative to the traditional gain scheduling process. The dynamic model and controller are still linear, but they are dependent on time-varying parameters whose values can be measured, but are unknown in advance. A traditional gain-scheduling approach involves linearizing dynamic models at many trim points and then designing linear time invariant (LTI) controllers at each trim point. The global controller is then estimated by interpolating the gains of the local controllers. This approach has been successfully implemented in many engineering applications but there is no guarantee of exponential stability [12]. The LPV control strategy can show exponential stability for all possible values of the parameters and can be implemented like a traditionally gain-scheduled control system [11] [13].

The LPV control strategy requires an affine dynamic system that must be converted into convex polytopic form [11]. The rigid-body aircraft model is not affine and will result in an infinite number of Linear Matrix Inequalities (LMIs) to determine the controller [11]. This difficulty is overcome using the Tensor Product (TP) model transformation to convert the nonaffine system into convex polytopic form. There is then a linear control allocator generated from a linearized system model produced by the real-time modeling procedure. The control allocator commands control surface deflections such that the moment commanded by the controller is realized. The newly developed LPV control law was tested on a nonlinear simulation of the NASA E1 aircraft, depicted in fig. 5.



Fig. 5 NASA E1 Aircraft

The E1 aircraft is a commercially-available 40-percent scale Edge 330 SC remotely-piloted fixed-wing airplane. Control surfaces are conventional ailerons and trailing-edge flaps on the wings, along with a conventional rudder and a split elevator. E1 is powered by an electric motor driving a fixed-pitch tractor propeller. More information about E1 can be found in [8, 14, 15].

III. Excitation for Real-Time Modeling

In order for the real-time modeling procedure to obtain a nonlinear global aircraft model, the flight data must include a wide range of values for the explanatory variables. The programmed test inputs developed and applied as described in [8] simultaneously command each of the aircraft control surfaces using inputs with wide-band frequency content. Each perturbation u_j applied to the j^{th} individual control surface is a sum of selected harmonic sinusoids with individual phase shifts, ϕ_k . The frequencies of these harmonic sinusoids are listed in Table 1.

$$u_j = \sum_{k \in \{1, 2, \dots, M\}} A_k \sin\left(\frac{2\pi k t}{T} + \phi_k\right) \quad (1)$$

The integer M is the number of available harmonically-related frequencies. T is the time length of excitation, A_k are the amplitudes for the sinusoidal components, and t is time. The design of each multisine input is determined by methods shown in [8].

Table 1 PTI frequencies for each effector (Hz)

L. Flap	R. Flap	L. Ail.	R. Ail.	Elev.	Rud.
.1250	.1750	.1000	.1500	.0750	.0500
.2750	.3250	.2500	.3000	.2250	.2000
.4250	.4750	.4000	.4500	.3750	.3500
.5750	.6250	.5500	.6000	.5250	.5000
.7250	.7750	.7000	.7500	.6750	.6500
.8750	.9250	.8500	.9000	.8250	.8000
1.0250	1.0750	1.0000	1.0500	.9750	.9500
1.1750	1.2250	1.1500	1.2000	1.1250	1.1000
1.3250	1.3750	1.3000	1.3500	1.2750	1.2500
1.4750	1.5250	1.4500	1.5000	1.4250	1.4000
1.6250	1.6750	1.6000	1.6500	1.5750	1.5500

The real-time modeling procedure is also described in detail in [8]. The PTI excitations, which are necessary for real-time modeling, produce aircraft responses which the control system sees as disturbances that must be suppressed. This creates a fundamental conflict. Good control tends to mitigate disturbances, such as those due to the PTI. However, real time modeling needs to observe the aircraft response to the PTI in order to accurately construct global aerodynamic models.

IV. LPV Model Derivation

The MCAAD project presents a challenge in real-time control design: the aircraft must be stabilized and controlled without a predefined model. However, a real-time modeling procedure can be performed to generate a global nonlinear aerodynamic model. The real-time modeling procedure uses PTI to excite the aircraft response to produce informative data. Thus the controller must be effective enough to stabilize and ensure safety of the aircraft, but not suppress the response of the aircraft due to the PTI. This is accomplished through a multi-loop control architecture design. The outer loop receives a guidance command $r = [\alpha_{cmd} \ \beta_{cmd} \ \chi_{cmd}]^T$ and calculates an attitude command state $(\phi_{cmd}, \omega_{2,cmd}, \omega_{3,cmd})$ by the outer loop of the LPV control method described later in this paper, where $\omega = [\omega_1 \ \omega_2 \ \omega_3]^T$ are the body-axis angular rates. Additionally $\theta_{cmd} = \alpha_{cmd}$ and $\omega_{1,cmd} = 0$ are added to the attitude command state. The inner loop then receives the attitude command state and calculates a moment command (M_{cmd}) to be passed to the control allocator. In this section the LPV models used for inner- and outer- loop control design are derived.

A state-space representation of an LTI system is:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{2}$$

Where A , B , C , and D denote the system, input, output, and direct-transmission matrices respectively. In general, aircraft dynamics are nonlinear and the LTI representation in eq. 2 is obtained by linearization of a nonlinear model at a specific point in the operating range. Linear control methods can be applied to stabilize an LTI system, but the performance of the linear controller will deteriorate as the system deviates from the design point. To capture nonlinear effects in the linear system, matrices A , B , C , and D can be described in terms of a parameter vector $\mathbf{p}$ which is unknown in advance, but can be measured in real-time. An example of an element in the parameter vector might be angle of attack, attitude angles, or airspeed. This yields the LPV model of the system as [13]:

$$\begin{aligned}\dot{x} &= A(\mathbf{p})x + B(\mathbf{p})u \\ y &= C(\mathbf{p})x + D(\mathbf{p})u\end{aligned}\tag{3}$$

The parameter-dependent LPV controller is defined by:

$$\begin{aligned}\dot{x}_K &= A_K(\mathbf{p})x_K + B_K(\mathbf{p})e \\ u &= C_K(\mathbf{p})x_K + D_K(\mathbf{p})e\end{aligned}\tag{4}$$

The value, e , is the error signal that is the difference between the command and the output. The LPV dynamic model and the controller are parameter dependent. Therefore, both the LPV dynamic model and the controller are automatically updated based on the measured values for $\mathbf{p}$.

The LPV controllers are derived using Bounded Real Lemma with quadratic H_∞ performance Γ [13]. With closed-loop state-space matrices, $A_{cl}(\mathbf{p})$, $B_{cl}(\mathbf{p})$, $C_{cl}(\mathbf{p})$, and $D_{cl}(\mathbf{p})$ the systems have quadratic H_∞ performance Γ if and only if there exists a single positive definite matrix, P such that,

$$\begin{bmatrix} A_{cl}^T(\mathbf{p})P + PA_{cl}(\mathbf{p}) & PB_{cl}(\mathbf{p}) & C_{cl}^T(\mathbf{p}) \\ B_{cl}^T(\mathbf{p})P & -\Gamma I & D_{cl}^T(\mathbf{p}) \\ C_{cl}(\mathbf{p}) & D_{cl}(\mathbf{p}) & -\Gamma I \end{bmatrix} < 0\tag{5}$$

for all possible values of $\mathbf{p}$. The Lyapunov function, $V(x) = x^T Px$ then establishes asymptotic stability and the $\mathcal{L}_2$ gain between the input and the output is bounded by $\|y\| < \Gamma\|u\|$. Quadratic H_∞ performance requires a single fixed quadratic Lyapunov function for the entire operating range [13]. Despite the controllers being designed by solving the linear matrix inequality (LMIs) given in eq. 5 there are an infinite number of solutions, corresponding to the infinite possible trajectories of $\mathbf{p}$. The concept of an Affine Polytopic LPV model has been introduced to overcome this difficulty [12] and will be discussed in section IV.A.

The aircraft model used in both LPV control loops is a rigid-body dynamics model which are fully-known across all flight conditions. Use of this rigid-body model renders the controller independent of the real-time modeling algorithm which will help mitigate interactions between the real-time modeling and control. The control allocation algorithm, described later, uses the results of the real-time aerodynamic modeling algorithm to realize the commanded moment by the LPV control loops.

To discuss the development of the LPV models, it is assumed that the aircraft dynamics are governed by the standard rigid-body equations of motion for gliding flight:

Inner Loop Equations of Motion:

$$\dot{\phi} = \omega_1 + \tan(\theta)(\omega_2 \sin(\phi) + \omega_3 \cos(\phi))\tag{6a}$$

$$\dot{\theta} = \omega_2 \cos(\theta) - \omega_3 \sin(\theta)\tag{6b}$$

$$\dot{\omega} = I^{-1}(M + M_u) - I^{-1}(\omega \times I\omega)\tag{6c}$$

Outer Loop Equations of Motion:

$$\dot{\alpha} = \omega_2 - \tan(\beta)(\cos(\alpha)\omega_1 + \sin(\alpha)\omega_1) + \frac{1}{mV \cos(\beta)}(-L + mg \cos(\gamma) \cos(\mu)) \quad (7a)$$

$$\dot{\beta} = -\cos(\alpha)\omega_3 + \sin(\alpha)\omega_2 + \frac{1}{mV}(Y \cos(\beta) + mg \cos(\gamma) \sin(\mu)) \quad (7b)$$

$$\dot{\gamma} = \frac{1}{mV}(L \cos \mu - Y \sin(\mu) \cos(\beta) - mg \cos(\gamma)) \quad (7c)$$

$$\dot{\chi} = \frac{1}{mV \cos(\gamma)}(L \sin(\mu) + Y \cos(\mu) \cos(\beta)) \quad (7d)$$

Under the following simplifying assumptions:

- $\mu \approx \phi$,
- $\cos(\alpha) \approx \cos(\gamma) \approx 1$,
- $Y \approx 0$,
- $\beta \approx 0$,
- $L \cos(\phi) \approx mg$,
- $\sin(\alpha)\omega_1 \approx 0$,

The outer loop equations of motion become:

$$\dot{\alpha} = \omega_2 - \frac{g}{V} \sin(\phi) \tan(\phi) \quad (8a)$$

$$\dot{\beta} = -\omega_3 + \frac{g}{V} \sin(\phi) \quad (8b)$$

$$\dot{\gamma} = 0 \quad (8c)$$

$$\dot{\chi} = \frac{g}{V} \tan(\phi) \quad (8d)$$

The simplifying assumptions, albeit highly restrictive, simplify the controls problem to a tractable problem for demonstration of the efficacy of the LPV control law.

Now separate the LPV model into inner and outer control loops. The outer loop receives a guidance command, $(\alpha_{cmd}, \beta_{cmd}, \chi_{cmd})$ and calculates an attitude state command, $(\phi_{cmd}, \omega_{2,cmd}, \omega_{3,cmd})$. Equation 6 and 8 are now linearized about an equilibrium point in flight, $[\omega_1 \ \omega_2 \ \omega_3]^T = [0 \ 0 \ 0]^T$ and put into LPV form for both the inner and outer loops:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\chi} \end{bmatrix} = A_o \begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} + B_o \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = A_i \begin{bmatrix} \phi \\ \theta \\ \omega \end{bmatrix} + B_i M \quad (10)$$

where,

$$A_o = 0_{3 \times 3} \quad B_o = \begin{bmatrix} -\frac{g}{V}(\cos(\phi) \tan(\phi) + \sin(\phi) \sec^2(\phi)) & 1 & 0 \\ \frac{g}{V} \cos(\phi) & 0 & -1 \\ \frac{g}{V} \sec^2(\phi) & 0 & 0 \end{bmatrix}$$

Assuming $I_{xz} = 0$:

$$A_i = \begin{bmatrix} 0 & 0 & 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & 0 & 0 & \cos(\phi) & -\sin(\phi) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

Defining the parameter vector as the variables that remain in the linearized system, the parameter vectors for the outer loop and inner loop LPV controllers are:

$$\mathbf{p}_o = [\phi \quad V]^T \quad (11a)$$

$$\mathbf{p}_i = [\phi \quad \theta]^T \quad (11b)$$

A. Affine Polytopic LPV Systems

Assume that each element of the parameter vector, $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_N] \in \mathbb{R}^N$ lies between extreme values $\underline{p}_i$ and $\bar{p}_i$ and $\mathbf{p}$ lies in a polytope $\mathfrak{T}$ with vertices $\mathfrak{T}_1, \mathfrak{T}_2, \dots, \mathfrak{T}_m$. With $p \in \mathfrak{T}$ and $m = 2^N$ [16]. In an affine LPV model the system matrices, $S(\mathbf{p})$ are linear functions of the time-varying parameter vector and can be written as a convex combination of the vertex matrices of the parameter polytope, $\mathfrak{T}$ [13].

$$S(\mathbf{p}) = \begin{bmatrix} A(\mathbf{p}) & B(\mathbf{p}) \\ C(\mathbf{p}) & D(\mathbf{p}) \end{bmatrix} = \sum_{i=1}^m w_i(\mathbf{p}) \begin{bmatrix} A(\mathfrak{T}_i) & B(\mathfrak{T}_i) \\ C(\mathfrak{T}_i) & D(\mathfrak{T}_i) \end{bmatrix} \quad (12)$$

$$\sum_{i=1}^m w_i(\mathbf{p}) = 1$$

It can be shown that eq. 5 holds for all possible trajectories of $\mathbf{p}$ if, and only if, it holds at the vertices, $\mathfrak{T}_i$ for all $i = 1, \dots, m$ [13]. This result allows for an infinite number of LMIs to reduce to a finite number of LMIs due to the convexity of the affine LPV system. However, eqs. 9 and 10 are not linear functions of the parameter vector, thus the LPV model is not affine. The Tensor Product model transformation is used to generate an affine polytopic LPV system [11].

B. Tensor Product (TP) Transformation

Many control problems can be formulated into the affine LPV form, however the aircraft model in the previous section is in nonaffine form and has to be converted into affine form to obtain a LPV controller with a finite number of LMIs. This conversion process is done through the Tensor Product model transformation based on the Higher Order Singular Value Decomposition of tensors (HOSVD). The TP transformation is given in [16] and reviewed here. Begin by writing the LPV model as follows:

$$S(\mathbf{p}(t)) = \begin{bmatrix} A(\mathbf{p}) & B(\mathbf{p}) \\ C(\mathbf{p}) & D(\mathbf{p}) \end{bmatrix} \in \mathbb{R}^{(n+q) \times (n+m)} \quad (13)$$

The integers n , q , and m are the number of states, inputs, and outputs respectively. Given bounds on the limits of parameter variations, the parameter space, P is,

$$P = [\underline{p}_1, \bar{p}_1] \times [\underline{p}_2, \bar{p}_2] \times \dots \times [\underline{p}_N, \bar{p}_N] \quad (14)$$

The LPV model is evaluated at grid points in the parameter space and is stored as a tensor $\mathbb{S}^D \in \mathbb{R}^{M_1 \times M_2 \times \dots \times M_N \times (n+q) \times (n+m)}$ where $M_1, M_2, \dots, M_N$ are the number of grid lines defined in each dimension of the grid. The next step is to apply the HOSVD on the first N dimensions of the of the tensor $\mathbb{S}^D$ and discard the zero or small singular values, σ_k and their corresponding singular vectors in all dimensions $1, 2, \dots, N$. This is performed using the tool given in [17]. The HOSVD results in a core tensor $\mathbb{S}$ and set of matrices $U_1, U_2, \dots, U_N$ such that $\mathbb{S}^D \approx \mathbb{S} \times U_1 \times U_2 \times \dots \times U_N$. The matrices $U_1, U_2, \dots, U_N$ contain the continuous weighting functions for each

bounded variable. For example the matrix $U_n \in \mathbb{R}$ is made of weighting functions for the n^{th} variable, $w_{n,j}(p_n) \in \mathbb{R}^{M_n}$, $j = 1, \dots, I_n$ and I_n is the number of weighting functions used in the n^{th} dimension of the parameter vector p_n . The weighting functions at each vertex can now be defined as,

$$w_r(\mathbf{p}) = \prod_{n=1}^N w_{n,j}(p_n) \quad (15)$$

The purpose of the TP model transformation is to find a LTI vertex system S_r and weighting functions, $w_r(\mathbf{p})$ such that the system can be written by the following:

$$S(\mathbf{p}) = \sum_{r=1}^R w_r(\mathbf{p}) S_r \quad (16)$$

Convex combinations of the S_r -LTI vertex system are guaranteed by the following conditions [16]:

$$\forall n \in [1, N], j \in [1, I_n], p_n : w_{n,j}(p_n) \in [0, 1] \quad (17)$$

$$\forall n \in [1, N], p_n : \sum_{j=1}^{I_n} w_{n,j}(p_n) = 1 \quad (18)$$

$$\forall r \in [1, R], p : w_r(\mathbf{p}) \in [0, 1] \quad (19)$$

$$\forall \mathbf{p} : \sum_{r=1}^R w_r(\mathbf{p}) = 1 \quad (20)$$

The TP method will transform an LPV system, $S(\mathbf{p}(t))$ into a convex hull of LTI vertex systems, S_r . This is performed for both the inner and outer loops such that, $S_i(\mathbf{p}(t))$ and $S_o(\mathbf{p}(t))$ are transformed into $S_{i,r}$ and $S_{o,r}$ respectively. The TP convex-polytopic system representation is an approximation and the error, ϵ is given as [16]:

$$\left\| S(\mathbf{p}) - \sum_{r=1}^R w_r(\mathbf{p}) S_r \right\| = \epsilon \quad (21)$$

During the HOSVD of the inner loop, three singular values were found along the ϕ dimension of $S_i(\mathbf{p}(t))$. Three singular values were also found along the θ dimension of $S_i(\mathbf{p}(t))$. This results in nine vertices in the inner loop. The HOSVD of the outer loop found four singular values along the V dimension of $S_o(\mathbf{p}(t))$ and two singular values along the ϕ dimension of $S_o(\mathbf{p}(t))$. This results in eight vertices. The original non-affine LPV model can now be summed across all vertices and be rewritten in TP-convex polytopic form as:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\chi} \end{bmatrix} = \sum_{r=1}^8 w_{o,r}(\mathbf{p}) B_{o,r} \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} \phi \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \sum_{r=1}^9 w_{i,r}(\mathbf{p}) \left(A_{i,r} \begin{bmatrix} \phi \\ \theta \\ \omega \end{bmatrix} + B_{i,r} M \right) \quad (23)$$

Where $A_{i,r}$, $B_{i,r}$ are extracted from the LTI vertices from the inner loop, $S_{i,r}$. $B_{o,r}$ is extracted from the LTI vertices of the outer loop, $S_{o,r}$. Several weighting functions, $w_{i,1}$ and $w_{o,2}$ are depicted in fig. 6 and fig. 7. The boundaries of the parameter space are determined by iteratively increasing them until the control design procedure described in section V is unable to find a global optimal solution and they are listed in Table 2. Lastly, both TP-convex polytopic models were compared with the original LPV models over 1,000 randomly selected points in the parameter space the mean and maximum error are calculated by eq. 21 and tabulated in Table 3. It is shown that the error is small and thus both the TP-convex polytopic models are good representations of the LPV models.

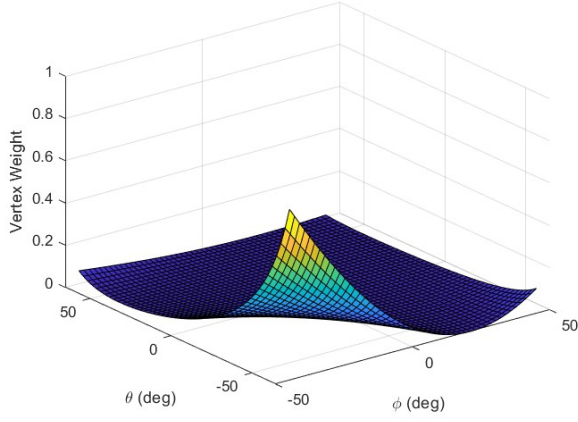


Fig. 6 Weighting function: $w_{i,1}$

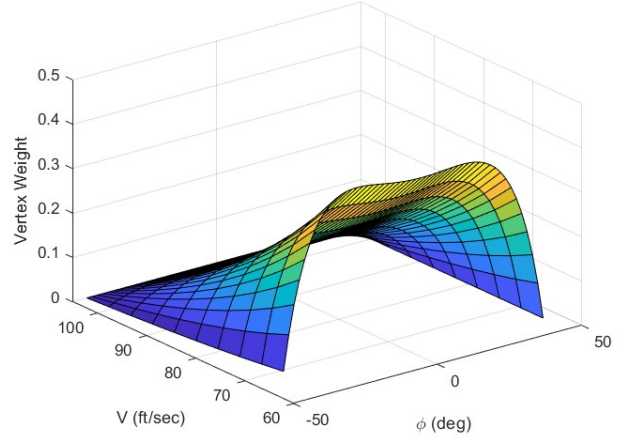


Fig. 7 Weighting function: $w_{o,2}$

Table 2 Parameter bounds and points

Parameter	Min.	Max.	Unit	Grid Points
ϕ	-45	45	deg	50
θ	-65	65	deg	50
V	65	105	ft s ⁻¹	10

Table 3 Error Parameters

Model	Mean Error (%)	Max. Error (%)
Inner Loop	3.2055×10^{-5}	6.7593×10^{-5}
Outer Loop	5.3495×10^{-5}	9.4605×10^{-5}

V. Linear Parameter-Varying Control Design

It is desired that the controller track the guidance commands, but also not suppress the response of the aircraft to the PTI excitations. The guidance algorithms are described in detail in [18]. To design the LPV control loops to achieve adequate tracking performance, there are two penalty filters in the inner loop, $W_{1,inner}$ and $W_{2,inner}$, and one penalty filter, $W_{1,outer}$ in the outer loop. The control architecture is built using the HINFGS MATLAB routine and shown in fig. 8.

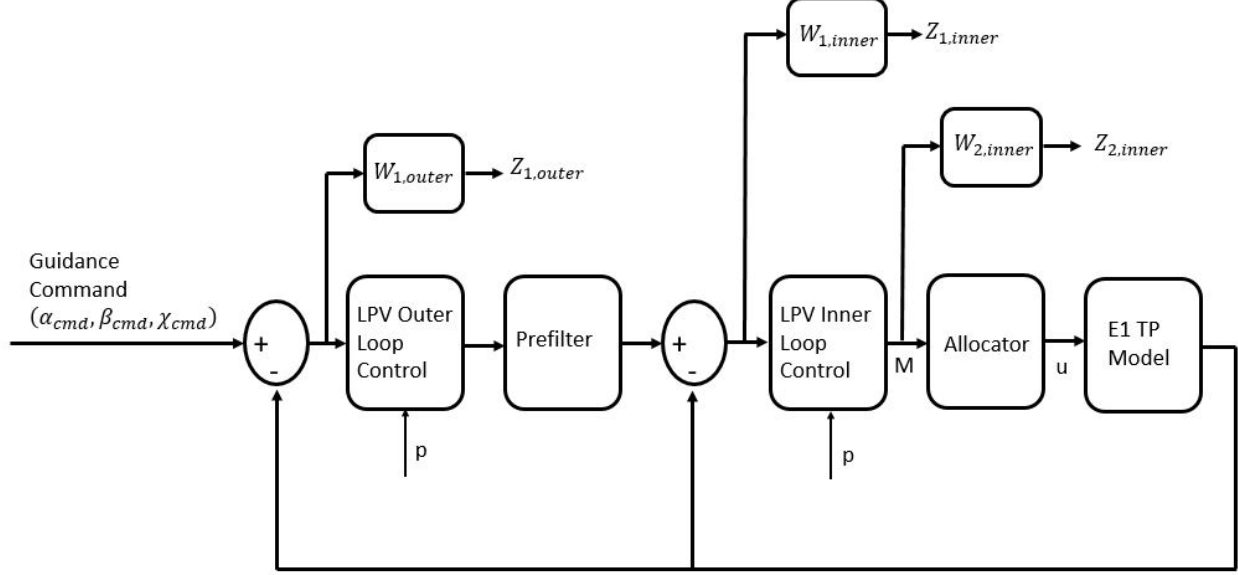


Fig. 8 Detailed H_∞ LPV Control Architecture

The inner loop controller is designed to penalize moment commands inside the PTI bandwidth, through minimization of the output of $W_{2,inner}$, $Z_{2,inner}$. $W_{2,inner}$ is chosen as a first-order high-pass Bessel filter with cutoff frequency, $\omega_{c,2}$ to penalize control signals within the PTI bandwidth and force the control signals above the PTI bandwidth. The cutoff frequency was chosen by iteratively increasing the frequency until HINFGS can no longer find a global-optimal controller.

$$\omega_{c,2} = 1.6\text{Hz} \quad (24)$$

The state-space of the first order, high-pass, Bessel filter is:

$$(A_{2,i}, B_{2,i}, C_{2,i}, D_{2,i}) = (-10.0531, 1, -10.0531, 1) \quad (25)$$

The moment vector is a three-dimensional vector and thus the first-order, high-pass, Bessel filter is concatenated into a three-input, three-output filter to form $W_{2,inner}$.

$$W_{2,inner} \begin{cases} \dot{x}_{2,i} = \begin{bmatrix} A_{2,i} & 0 & 0 \\ 0 & A_{2,i} & 0 \\ 0 & 0 & A_{2,i} \end{bmatrix} x_{2,i} + \begin{bmatrix} B_{2,i} & 0 & 0 \\ 0 & B_{2,i} & 0 \\ 0 & 0 & B_{2,i} \end{bmatrix} M \\ Z_{2,inner} = \begin{bmatrix} C_{2,i} & 0 & 0 \\ 0 & C_{2,i} & 0 \\ 0 & 0 & C_{2,i} \end{bmatrix} x_{2,i} + \begin{bmatrix} D_{2,i} & 0 & 0 \\ 0 & D_{2,i} & 0 \\ 0 & 0 & D_{2,i} \end{bmatrix} M \end{cases} \quad (26)$$

To ensure adequate response speed, a first-order, low-pass, Bessel filter is put on the tracking commands in the inner-loop. The cutoff frequency, $\omega_{c,1}$ is used and the filter is concatenated into a five-input, five-output filter to form, $W_{1,inner}$.

$$\omega_{c,1} = 1.6\text{Hz} \quad (27)$$

The state space of the first-order, low-pass, Bessel filter is:

$$(A_{1,i}, B_{1,i}, C_{1,i}, D_{1,i}) = (-10.0531, 1, -10.0531, 1) \quad (28)$$

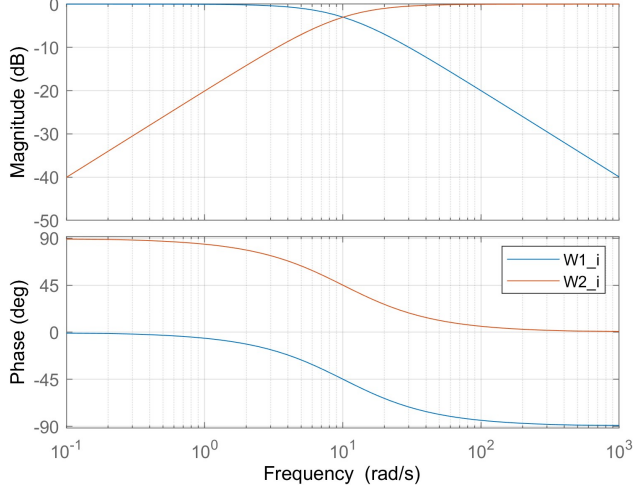


Fig. 9 Bode Plots of $W_{1,inner}$ and $W_{2,inner}$

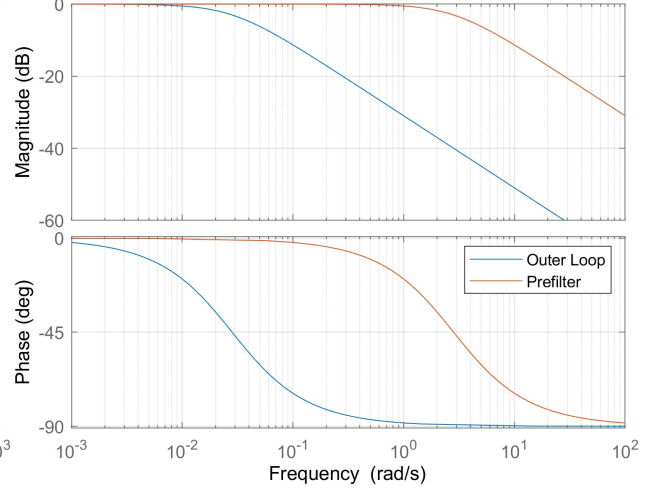


Fig. 10 Bode Plots of $W_{1,outer}$ and $W_{prefilter}$

The vector $e_i = [\phi_{err} \ \theta_{err} \ \omega_{err}^T]^T$ is five-dimensional and thus the filter is concatenated into a five-input, five-output filter to form $W_{1,inner}$.

$$W_{1,inner} \begin{cases} \dot{x}_{1,i} = \begin{bmatrix} A_{1,i} & 0 & \dots & 0 \\ 0 & A_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{1,i} \end{bmatrix} x_{1,i} + \begin{bmatrix} B_{1,i} & 0 & \dots & 0 \\ 0 & B_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B_{1,i} \end{bmatrix} e_i \\ Z_{1,inner} = \begin{bmatrix} C_{1,i} & 0 & \dots & 0 \\ 0 & C_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & C_{1,i} \end{bmatrix} x_{1,i} + \begin{bmatrix} D_{1,i} & 0 & \dots & 0 \\ 0 & D_{1,i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_{1,i} \end{bmatrix} e_i \end{cases} \quad (29)$$

To ensure adequate response speed in the outer-loop controller, a penalty filter on the guidance tracking error, $W_{1,outer}$ is added; which is also a first-order, low-pass, Bessel filter with cutoff frequency $\omega_{c,outer}$.

$$\omega_{c,outer} = 0.0045\text{Hz} \quad (30)$$

The state space of the first-order, low-pass, Bessel filter is:

$$(A_{1,o}, B_{1,o}, C_{1,o}, D_{1,o}) = (-.0283, 1, .0283, 0) \quad (31)$$

The vector $e_o = [\alpha_{err} \ \beta_{err} \ \chi_{err}]^T$ is three-dimensional and thus the filter is concatenated into a three-input, three-input filter.

$$W_{1,outer} \begin{cases} \dot{x}_{1,o} = \begin{bmatrix} A_{1,o} & 0 & 0 \\ 0 & A_{1,o} & 0 \\ 0 & 0 & A_{1,o} \end{bmatrix} x_{1,o} + \begin{bmatrix} B_{1,o} & 0 & 0 \\ 0 & B_{1,o} & 0 \\ 0 & 0 & B_{1,o} \end{bmatrix} e_o \\ Z_{1,outer} = \begin{bmatrix} C_{1,o} & 0 & 0 \\ 0 & C_{1,o} & 0 \\ 0 & 0 & C_{1,o} \end{bmatrix} x_{1,o} + \begin{bmatrix} D_{1,o} & 0 & 0 \\ 0 & D_{1,o} & 0 \\ 0 & 0 & D_{1,o} \end{bmatrix} e_o \end{cases} \quad (32)$$

Lastly, a prefilter has been added to remove the parameter dependency on the matrix, B_o [13]. This prefilter is also chosen as a first-order Bessel filter with high-cutoff frequency as to not modify the aircraft dynamics.

$$\omega_{c,p} = 0.5\text{Hz} \quad (33)$$

$$(A_{1,p}, B_{1,p}, C_{1,p}, D_{1,p}) = (-3.1416, 1, 3.1416, 0) \quad (34)$$

The tracking error command is three-dimensional, thus the prefilter is concatenated into a three-input, three-output filter.

$$W_p \begin{cases} \dot{x}_{1,p} = \begin{bmatrix} A_{1,p} & 0 & 0 \\ 0 & A_{1,p} & 0 \\ 0 & 0 & A_{1,p} \end{bmatrix} x_{1,p} + \begin{bmatrix} B_{1,p} & 0 & 0 \\ 0 & B_{1,p} & 0 \\ 0 & 0 & B_{1,p} \end{bmatrix} \begin{bmatrix} \alpha_{err} \\ \beta_{err} \\ \chi_{err} \end{bmatrix} \\ \begin{bmatrix} \alpha_{err,p} \\ \beta_{err,p} \\ \chi_{err,p} \end{bmatrix} = \begin{bmatrix} C_{1,p} & 0 & 0 \\ 0 & C_{1,p} & 0 \\ 0 & 0 & C_{1,p} \end{bmatrix} x_{1,p} + \begin{bmatrix} D_{1,p} & 0 & 0 \\ 0 & D_{1,p} & 0 \\ 0 & 0 & D_{1,p} \end{bmatrix} \begin{bmatrix} \alpha_{err} \\ \beta_{err} \\ \chi_{err} \end{bmatrix} \end{cases} \quad (35)$$

Bode plots of the inner-loop and outer loop filters are given in fig. 9 and 10 respectively. TP-convex polytopic models can now be concatenated with the state space models of the filters and put into H_∞ control form to minimize the output of the weighting filters, $\mathbf{z}_o$ and $\mathbf{z}_i$ for both inner and outer loops.

$$\begin{aligned} \dot{\mathbf{x}}_o &= \mathcal{A}_o(\mathbf{p})\mathbf{x}_o + B_{1,o}(\mathbf{p})r + B_{2,o}u_o \\ \mathbf{z}_o &= C_{1,o}(\mathbf{p})\mathbf{x}_o + D_{11,o}(\mathbf{p})r + D_{12,o}u_o \\ y_o &= C_{2,o}\mathbf{x}_o + D_{21,o}r + D_{22,o}u_o \end{aligned} \quad (36)$$

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathcal{A}_i(\mathbf{p})\mathbf{x}_i + B_{1,i}(\mathbf{p})u_o + B_{2,i}M \\ \mathbf{z}_i &= C_{1,i}(\mathbf{p})\mathbf{x}_i + D_{11,i}(\mathbf{p})u_o + D_{12,i}M \\ y_i &= C_{2,i}\mathbf{x}_i + D_{21,i}u_o + D_{22,i}M \end{aligned} \quad (37)$$

The values $\mathbf{x}_o$ and $\mathbf{x}_i$ represent the concatenated state vector for both loops. x represents the states of the TP-convex polytopic models. $\mathbf{z}_o$ and $\mathbf{z}_i$ represent the concatenated output of the weighting filters for both loops. y represents the values of the TP-convex polytopic models. r represents the guidance command $r = [\alpha_{cmd} \ \beta_{cmd} \ \chi_{cmd}]^T$ and u_o represents the control command $u_o = [\phi_{cmd} \ \omega_{2,cmd} \ \omega_{3,cmd}]^T$.

$$\mathbf{x}_o = [x_{p,o}^T \ x_{1,o}^T \ x_{prefilter}^T]^T \quad (38)$$

$$\mathbf{x}_i = [x_{p,i}^T \ x_{1,i}^T \ x_{2,i}^T]^T \quad (39)$$

$$\mathbf{z}_o = Z_{1,outer} \quad (40)$$

$$\mathbf{z}_i = [Z_{1,inner}^T \ Z_{2,inner}^T]^T \quad (41)$$

The inner and outer loop controllers were built in MATLAB using the `HINFGS` MATLAB routine. The quadratic H_∞ performance was found to be $\Gamma_i = 1.301217$ for the inner loop and $\Gamma_o = 1.000021$ for the outer loop.

A. NDI Review

The LPV control algorithm is compared with an existing Nonlinear Dynamic Inversion control algorithm and the control surface deflection commands from each controller are compared. The NDI control law is given in [5] and reviewed here. To start, the outer-most loop of the NDI converts the χ guidance commands to attitude commands, ϕ :

$$\phi_{cmd} = \text{Sat}_{-45^\circ}^{45^\circ} \left(\arctan \left(\frac{V}{g} K_\chi (\chi_{cmd} - \chi) \right) \right) \quad (42)$$

The next inversion step treats ω_1 , ω_2 , and ω_3 as inputs to the dynamics governing ϕ , α , and β respectively. Based on the simplified dynamics in eq. 8:

$$\omega_{1,cmd} = K_\phi(\phi_{cmd} - \phi) - \tan(\theta)(q \sin(\phi) + r \cos(\phi)) \quad (43a)$$

$$\omega_{2,cmd} = K_\alpha(\alpha_{cmd} - \alpha) + \frac{g}{V} \sin(\phi) \tan(\phi) \quad (43b)$$

$$\omega_{3,cmd} = -K_\beta(\beta_{cmd} - \beta) - \frac{g}{V} \sin(\phi) \quad (43c)$$

Table 4 Desired frequency calculations

Roll	$\omega_n = \sqrt{ \frac{\bar{q}Sb}{2I_{xx}}C_{l\delta_r} }$
Pitch	$\omega_n = \sqrt{ \frac{\bar{q}S\bar{c}}{I_{yy}}C_{m\alpha} }$
Yaw	$\omega_n = \sqrt{ \frac{\bar{q}Sb}{I_{zz}}C_{n\beta} }$

The fastest inner loop inverts the angular rate commands to produce a desired angular acceleration proportional to the angular rate error.

$$M_{u,cmd} = IK_\omega(\omega_{cmd} - \omega) - (\hat{M} - \omega \times I\omega) \quad (44)$$

$\hat{M}$ is the estimate of the current moment on the aircraft, which is determined from real-time modeling results. In order to determine the proportionality constants it is assumed the dynamic inversion commands were able to remove the nonlinear effects and only the desired linear portions remain. As an example, examine the linearized short-period dynamics:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_q K_\alpha & -K_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ K_q K_\alpha \end{bmatrix} \alpha_{cmd} \quad (45)$$

The control parameters are written in terms of the natural frequency and damping of the second order system.

$$K_q = 2\zeta\omega_n \quad (46)$$

$$K_\alpha = \omega_n/(2\zeta) \quad (47)$$

The desired natural frequency was determined based on the modeling results using the relationship in Table 4 and the desired damping ratio was fixed at $\zeta = 0.8$. The control parameters for the roll channel (K_ϕ, K_p) and the yaw channel (K_β, K_r) are determined by the same procedure. The control parameter for the tracking angle, χ was fixed at $K_\chi = 0.375$.

B. AutoThrottle

Both the LPV and the NDI controllers use the same auto throttle controller to regulate the calibrated airspeed of the aircraft. A proportional-integral (PI) controller was taken from [5] and used here.

$$Th = K_p(V_{CAS,cmd} - V_{CAS}) + K_I \int_0^t (V_{CAS,cmd} - V_{CAS}) d\tau \quad (48)$$

C. Control Allocation

Control allocation algorithms are tasked with determining aircraft control surface deflections to achieve the desired moments commanded by the controller. Control allocation is generally required for aircraft where the number of control effectors exceeds the number of moments (over-actuated). Many researchers have formulated the control allocation problem as a constrained-optimization problem [19–25]. The allocator in this paper is taken from the L2F project, [5] and is used for both the LPV and NDI control laws.

The real-time modeling procedure gives a matrix $B \in \mathbb{R}^{3 \times n_d}$ with dimensional control derivatives defining the partial derivatives with respect to the n_d control surface deflections. Given a moment and acceleration command the control allocation problem is stated as follows [6]. Find the deflection vector, u such that,

$$J = \|Bu - M_{cmd}\|_1 + \epsilon \|u - u_p\|_1 \quad (49)$$

is minimized, subject to $u_{min} \leq u \leq u_{max}$ and u_p is the current control surface deflection. In MCAAD the modeling determines B, whereas the online control specifies M_u . This formulation of the control allocation problem is a mixed optimization problem that combines the error and control minimization problems into a single problem with parameter, ϵ . Minimizing the moment error $\|Bu - M_{cmd}\|_1$ is the primary objective of the control allocation designer, but for an over-actuated aircraft configuration there are infinitely many solutions. Thus the second term, $\epsilon \|u - u_p\|_1$ is added to the

optimization problem to determine a particular solution. In this research the metric used to define J is the l_1 norm which is the sum of the absolute values within the norm:

$$||x||_1 = \sum_{i=1}^n |x_i| \quad (50)$$

The control allocation problem in eq. 49 is solved in real time using an efficient algorithm formulated by Bodson using linear programming approaches, providing guaranteed convergence in an acceptable period of time [26]. The algorithm was based on the revised simplex method [27] with additional refinements, such as anti-cycling, as described in detail in [26].

VI. Simulation

As previously noted, the architecture shown in fig. 8 is built in SIMULINK[®] and the LPV controllers are designed using the HINFGS MATLAB routine. The gains from the two-loops LPV controller gains are now determined by:

$$K_{p,i} = \sum_{r=1}^9 w_{r,i}(\mathbf{p}) K_{r,i} \quad (51)$$

$$K_{p,o} = \sum_{r=1}^8 w_{r,o}(\mathbf{p}) K_{r,o} \quad (52)$$

Where:

$$K_r = \begin{bmatrix} A_{K,r} & B_{K,r} \\ C_{K,r} & D_{K,r} \end{bmatrix} \quad (53)$$

The implementation of the two-loop LPV control architecture is described in the following subsections.

A. Outer Loop Implementation

$$\begin{aligned} \dot{x}_{K,o} &= \sum_{r=1}^8 w_{r,o}(\mathbf{p}) \left(A_{K,r,o} x_{K,o} + B_{K,r,o} \begin{bmatrix} \alpha_{cmd} \\ \beta_{cmd} \\ \chi_{cmd} \end{bmatrix} \right) \\ \begin{bmatrix} \phi_{cmd} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \end{bmatrix} &= \sum_{r=1}^8 w_{r,o}(\mathbf{p}) \left(C_{K,r,o} x_{K,o} + D_{K,r,o} \begin{bmatrix} \alpha_{cmd} \\ \beta_{cmd} \\ \chi_{cmd} \end{bmatrix} \right) \end{aligned} \quad (54)$$

The commands calculated by the outer loop controller ($\phi_{cmd}, \omega_{2,cmd}, \omega_{3,cmd}$) are then passed through the prefilter described in sec. V to produce ($\phi_{cmd,p}, \omega_{2,cmd,p}, \omega_{3,cmd,p}$). There is also a saturation function to prevent the outer loop from commanding excessive roll angle.

$$\phi_{cmd,sat} = \text{Sat}_{-45^\circ}^{45^\circ}(\phi_{cmd,p}) \quad (55)$$

In order to command a zero flight path angle, pitch angle command is set to angle of attack command.

$$\theta_{cmd} = \alpha_{cmd} \quad (56)$$

Body-axis roll rate command, $\omega_{1,cmd}$ is set to zero.

$$\omega_{1,cmd} = 0 \text{ rad s}^{-1} \quad (57)$$

B. Inner Loop Implementation

$$\begin{aligned} \dot{x}_{K,i} &= \sum_{r=1}^9 w_{r,i}(\mathbf{p}) \left(A_{K,r,i} x_{K,i} + B_{K,r,i} \begin{bmatrix} \phi_{cmd,sat} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \end{bmatrix} \right) \\ M_{u,cmd} &= \sum_{r=1}^9 w_{r,i}(\mathbf{p}) \left(A_{K,r,i} x_{K,i} + B_{K,r,i} \begin{bmatrix} \phi_{cmd,sat} \\ \theta_{cmd} \\ \omega_{1,cmd} \\ \omega_{2,cmd,p} \\ \omega_{3,cmd,p} \end{bmatrix} \right) - \hat{M} \end{aligned} \quad (58)$$

The two-loop LPV controller has been tested on a SIMULINK simulation of the NASA E1 aircraft and compared with the results of the NDI controller.

It can be seen from fig. 11 that there is a large initial transient until 5s and then the LPV controller is able to track (α, β) commands well, with an error about $\pm 5^\circ$. The initial transient is caused by the controller's response to the large initial error in χ . When tracking χ commands, the LPV controller exhibits worse tracking performance than the NDI controller with a tracking error about $\pm 15^\circ$.

The NDI controller exhibits a better response to the initial χ error, causing less of an initial transient than the LPV controller, as shown in fig. 13. Additionally the guidance commands a step in α command at 23s which is not seen in the LPV simulation. This step is commanded because of a switch in the guidance block being triggered by high altitude. After the initial transient is settled, around 5s, the NDI and LPV exhibit similar tracking characteristics when with respect to α and β . The NDI consistently tracks χ better than the LPV controller, but the NDI has a very large α excursion after the χ step change at 17s while the LPV does not. Figure 12 and fig. 14 show the attitudes realized by both control laws. The LPV controller exhibits a large attitude response in roll angle, ϕ which is caused by the large initial χ error. Around 17s a large change in χ is commanded with results in a large change in roll angle, as realized by both the NDI and LPV controllers. Both controllers exhibit similar pitch angle excursions, with a maximum pitch angle about $+20^\circ$. The yaw angle, ψ follows the relationship $\psi = \chi - \beta$ and the resulting change in ψ stems from the commanded change in χ .

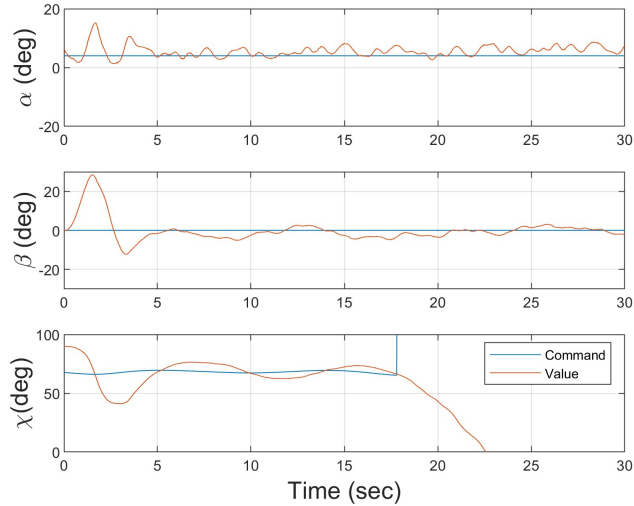


Fig. 11 Guidance tracking of the LPV controller

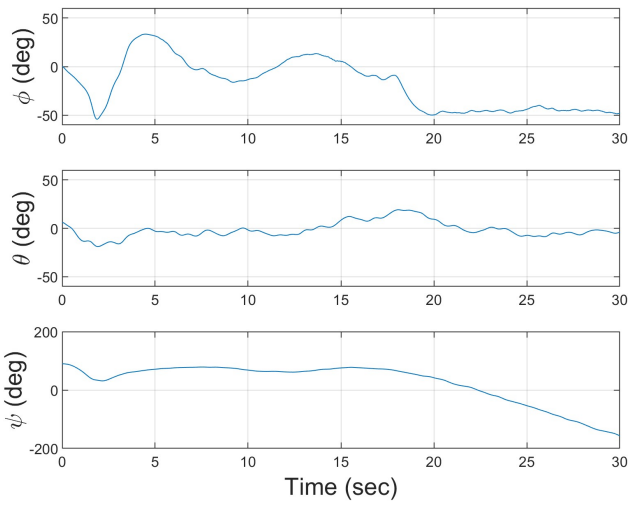


Fig. 12 Attitude angles using LPV controller

Additionally, the control surface deflections were recorded and they were compared to the commanded PTI control surface deflections in both the time domain and the frequency domain.

Figure 15 shows the control surface deflections commanded by the NDI controller compared to that of the PTI. The deflections correlate well for both left and right flaps. Additionally the deflections correlate well for both left and right ailerons until about 17s when there is an aggressive change in χ_{cmd} which results in an aggressive change in aileron command. The elevator and rudder are offset by about 5° from the PTI throughout the simulation.

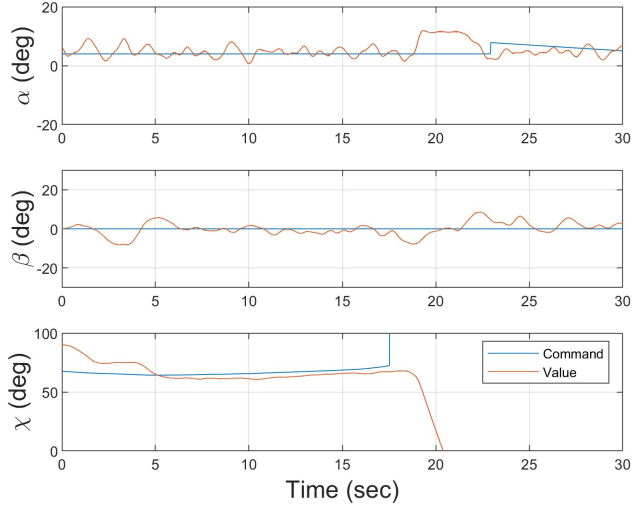


Fig. 13 Guidance tracking of the NDI controller

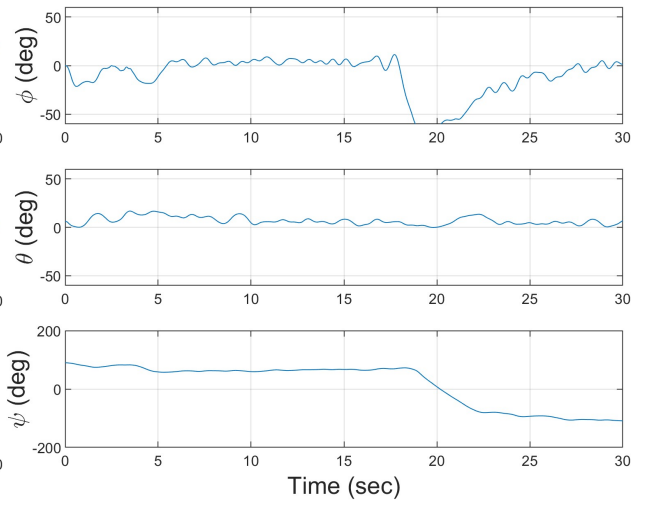


Fig. 14 Attitude angles using NDI controller

Figure 16 shows the control surface deflections commanded by the LPV controller compared to that of the PTI. The deflections correlate well for both left and right flaps and the left and right ailerons. Around 15s high frequency control surface content can be seen, which is expected for the LPV controller. Similar to the NDI controller, the LPV commands for elevator and rudder deflection offset about 5° .

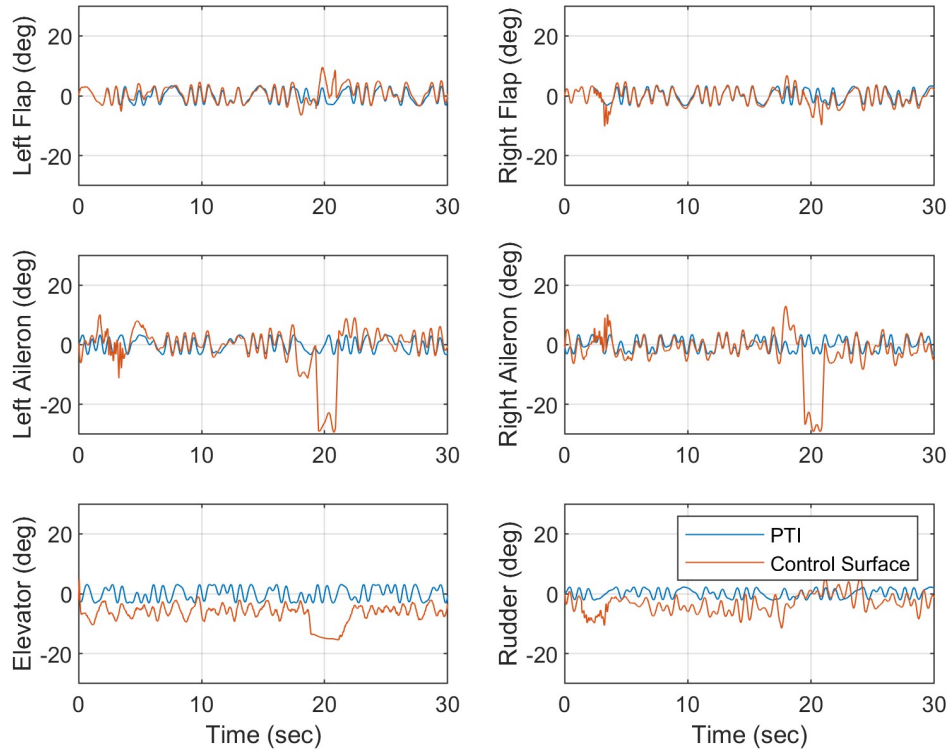


Fig. 15 Control Surface Deflections with NDI Control

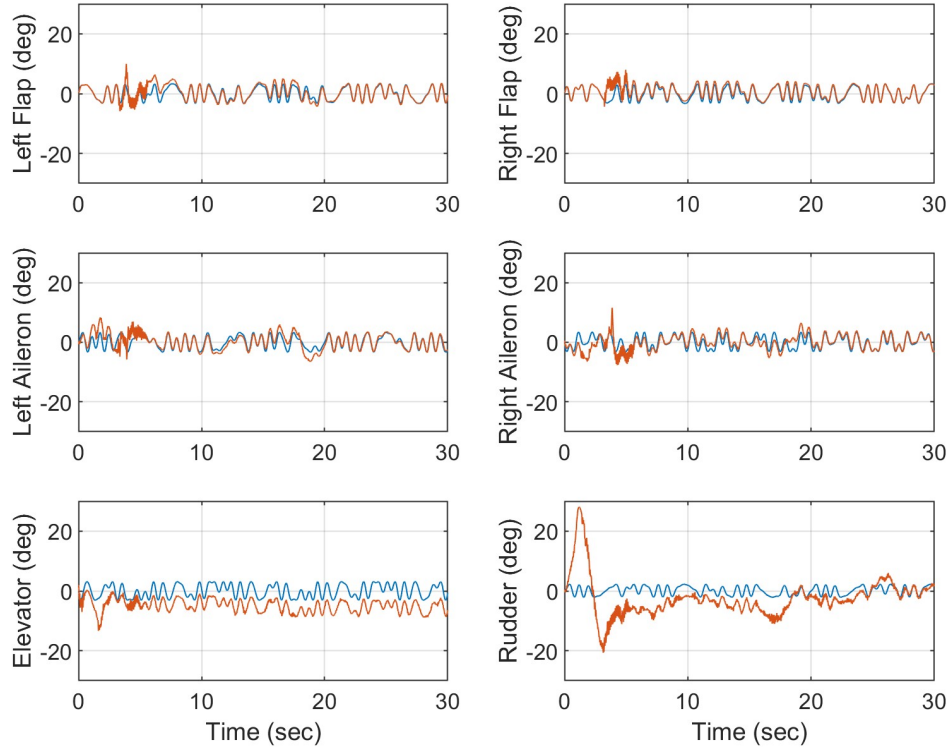


Fig. 16 Control Surface Deflections with LPV Control

Figure 17 shows the frequency domain content of the PTI and the control surface deflections commanded by the NDI control law. The left and right flap correlate well with the peaks of the PTI. The left and right aileron show poor correlation at frequencies below 0.5Hz, however at frequencies above 1Hz the control surface deflections start to align with the PTI. Similarly, the elevator and aileron do not align with the PTI well at frequencies below 0.5Hz but align better at frequencies above 1Hz.

Figure 18 shows the frequency domain content of the PTI and the control surface deflections commanded by the LPV control law. For all six effectors, there is little distortion in magnitude at frequencies below 0.5Hz and almost none at frequencies above 1Hz. It is important to note that the LPV control law commands control surface deflections at a higher frequency than the PTI, which allows for minimal distortion.

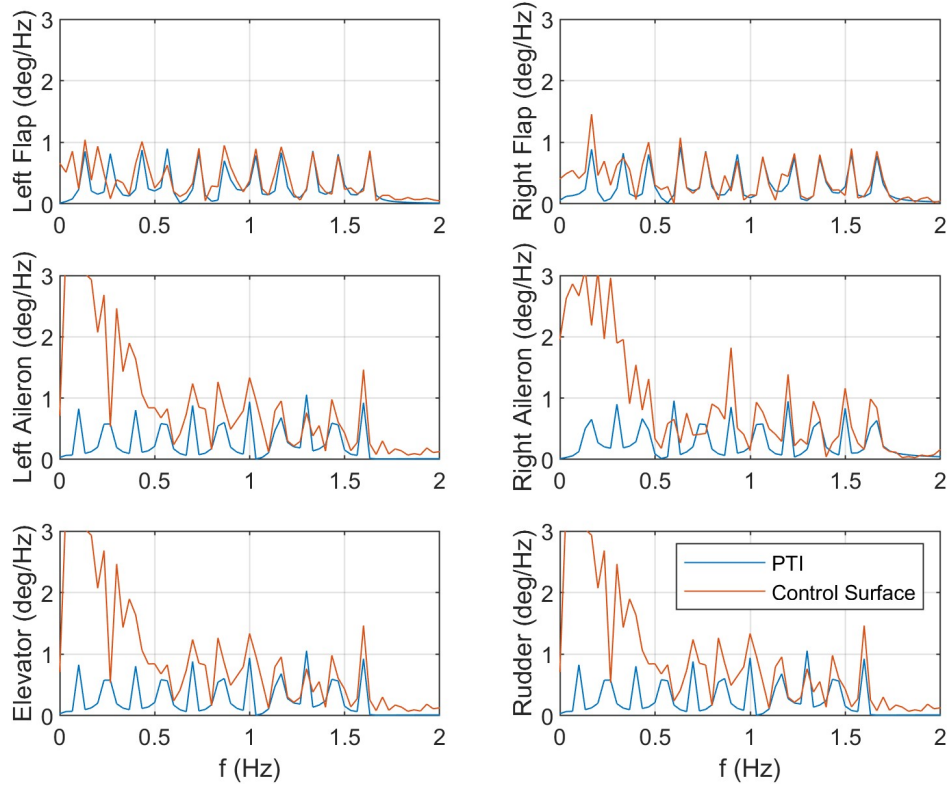


Fig. 17 Control Surface Frequency Content with NDI Control

To help quantify the similarity between control surface deflection and commanded PTI deflection in the frequency domain with both control laws, the magnitude squared coherence estimate was plotted for each control surface, for both control laws. The magnitude squared coherence estimate is a function of frequency which outputs values between 0 and 1. A value of 1 would indicate perfect coherence (linear dependency), and a value of 0 would indicate no coherence [28]. When considering coherence between the PTI and the control surface deflection, a value of 1 would indicate that 100% of the control surface deflection is due to PTI. A value of 0 would indicate that none of the control surface deflection is due to PTI.

$$C_{xy}(f) = \frac{|P_{xy}(f)|^2}{P_{xx}(f)P_{yy}(f)} \quad (59)$$

Equation 59 shows how the coherence between two sample signals is calculated. Where, P_{xy} is the cross-spectral density between x and y. P_{xx} and P_{yy} are the spectral densities of x and y respectively. Figure 19 shows the coherence between the control surface command due to PTI and the control surface command due to the controller.

From fig. 19 the flaps show similar coherence between the control surface commands and the PTI for all frequencies. For both the left and right ailerons the control surface commands from the LPV controller show consistently better coherence with the PTI until roughly 1.75Hz where coherence decreases for both NDI and LPV controllers. Lastly the coherence of the PTI with the LPV control surface commands is consistently better than that of the NDI control surface commands for the elevator and rudder.

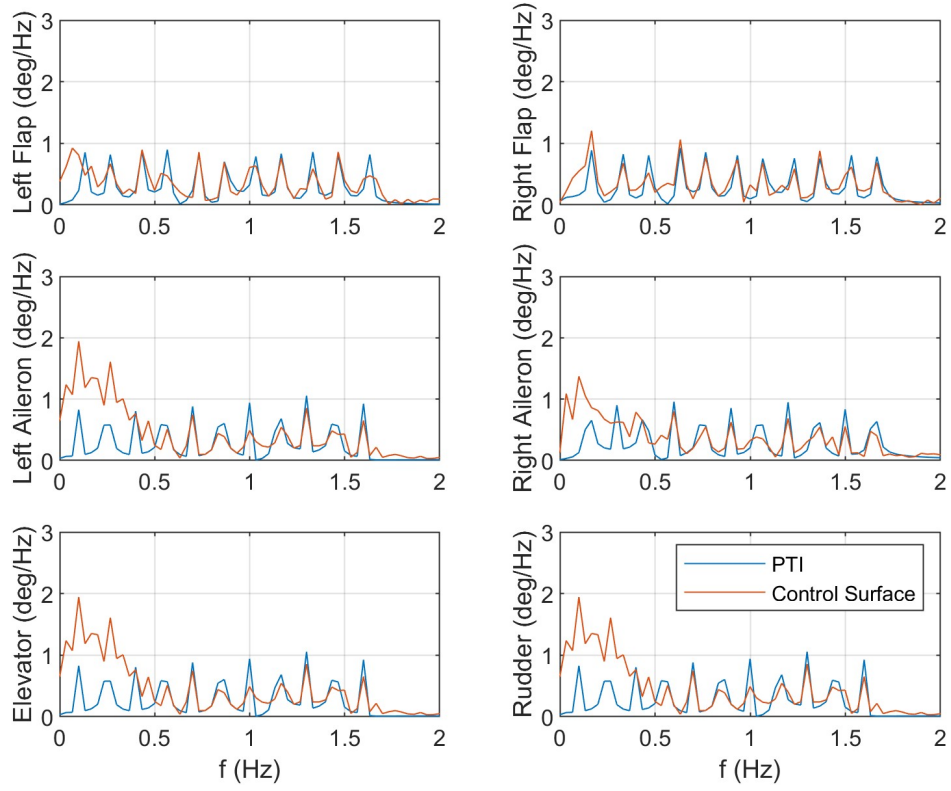


Fig. 18 Control Surface Frequency Content with LPV Control

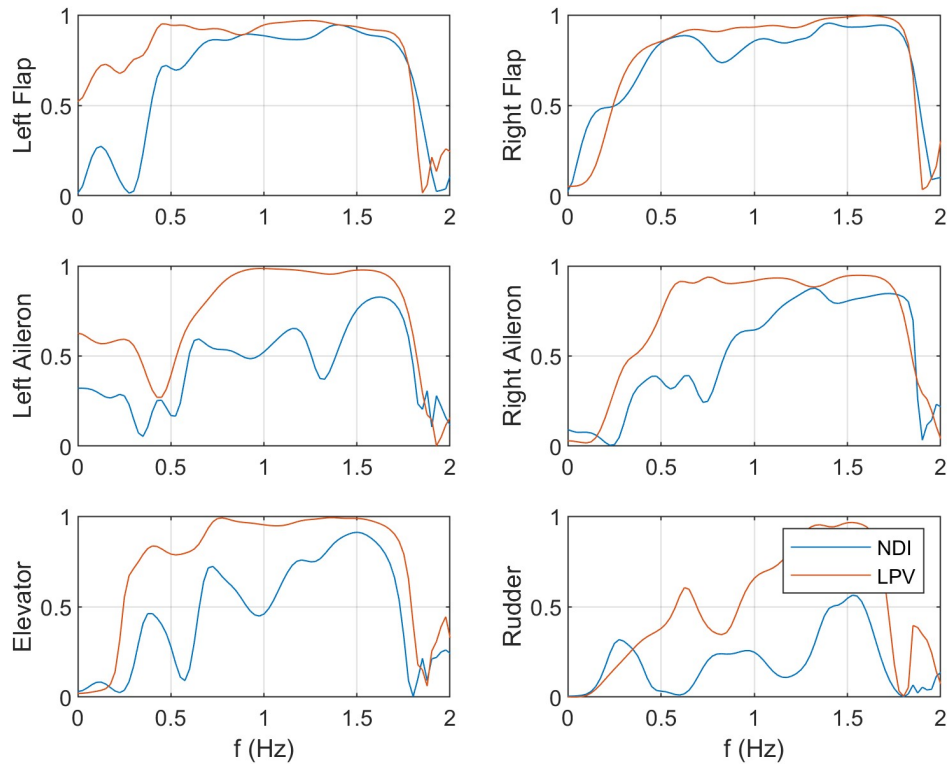


Fig. 19 Coherence between control surface commands and PTI

VII. Conclusions

In this work a two-loop LPV control law was designed for the NASA E1 aircraft. The synthesis of both control loops requires an affine LPV model, which was generated from the non-affine kinematic-based E1 model using the TP method. The derived two-loop LPV controller is tested on a SIMULINK® simulation of the NASA E1 aircraft, which compared the performance of the LPV controller to a previously designed NDI controller. It was found that the LPV controller is able to track guidance commands, but could exhibit slightly worse tracking performance than the NDI controller. Despite the slightly worse tracking performance, the LPV controller exhibits desirable characteristics such as its invariance of the real-time modeling parameters and its guarantee of exponential stability. Additionally, by commanding higher frequency control surface commands, the LPV control law distorts the PTI less, which will minimize interaction between real-time modeling and real-time control design.

Future work will entail development of nonlinear-stabilizing control allocation algorithms to and adaptive control allocation algorithms.

Acknowledgments

The author would like to thank Dr. Eugene Morelli and Dr. Steven Snyder at NASA Langley Research Center for their thorough guidance and peer review.

References

- [1] Ward, D., Monaco, J., Barron, R., Bird, R., Virnig, J., and Landers, T., "Self-Designing Controller: Design, Simulation, and Flight Test Evaluation," *SBIR Phase II Final Technical Report Contract F49620-94-C-0087, Barron Associates, Inc.*, November 1996.
- [2] Chandler, P., Mears, M., and Pachter, M., "On-line optimizing networks for reconfigurable control," *Proceedings of 32nd IEEE Conference on Decision and Control*, IEEE, San Antonio, TX, 1993, pp. 2272–2277.
- [3] Durham, W. C., and Bordignon, K. A., "Multiple control effector rate limiting," *Journal of guidance, control, and dynamics*, Vol. 19, No. 1, 1996, pp. 30–37.
- [4] Bolling, J. G., "Implementation of constrained control allocation techniques using an aerodynamic model of an f-15 aircraft," Master's thesis, Dept. of Aerospace and Ocean Engineering, Virginia Polytechnic Inst. and State Univ., May 1997.
- [5] Snyder, S., Bacon, B., Morelli, E. A., Frost, S., Teubert, C., and Okolo, W., "Online control design for learn-to-fly," *AIAA Atmospheric Flight Mechanics Conference*, AIAA, June 2018, p. 3311.
- [6] Snyder, S., "L1 adaptive control within learn-to-fly," Ph.D. thesis, University of Illinois at Urbana-Champaign, May 2019.
- [7] Snyder, S., "Autopilot Design with Learn-to-Fly," *AIAA Scitech 2020 Forum*, AIAA, Orlando, Florida, January 2020, p. 0763.
- [8] Morelli, E. A., "Practical aspects of real-time modeling for the learn-to-fly concept," *2018 Atmospheric Flight Mechanics Conference*, AIAA, June 2018, p. 3309.
- [9] Johansen, T. A., "Optimizing nonlinear control allocation," *43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, Vol. 4, IEEE, Atlantis, Paradise Island, Bahamas, 2004, pp. 3435–3440.
- [10] Tjønnås, J., and Johansen, T. A., "Adaptive control allocation," *Automatica*, Vol. 44, No. 11, 2008, pp. 2754–2765.
- [11] Rangajeeva, S. L., and Whidborne, J. F., "Linear parameter varying control of a quadrotor," *2011 6th International Conference on Industrial and Information Systems*, IEEE, 2011, pp. 483–488.
- [12] Chumalee, S., and Whidborne, J. F., "Gain-scheduled H-infinity control via parameter-dependent Lyapunov functions," *International Journal of Systems Science*, Vol. 46, No. 1, 2015, pp. 125–138.
- [13] Apkarian, P., Gahinet, P., and Becker, G., "Self-scheduled H infinity control of linear parameter-varying systems: a design example," *Automatica*, Vol. 31, No. 9, 1995, pp. 1251–1261.
- [14] Heim, E. H., Viken, E., Brandon, J. M., and Croom, M. A., "NASA's Learn-to-Fly Project Overview," *2018 Atmospheric Flight Mechanics Conference*, 2018, p. 3307.
- [15] Riddick, S. E., Busan, R. C., Cox, D. E., and Laughter, S. A., "Learn to fly test setup and concept of operations," *2018 Atmospheric Flight Mechanics Conference*, 2018, p. 3308.

- [16] Petres, Z., "Polytopic decomposition of linear parameter-varying models by tensor-product model transformation," Ph.D. thesis, Budapest University of Technology and Economics, May 2006.
- [17] Nagy, S., Petres, Z., and Baranyi, P., "TP Tool - A Matlab toolbox for TP model transformation," *Proc. of 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, Budapest, Hungary, 2007, pp. 483–495.
- [18] Foster, J. V., "Autonomous guidance algorithms for NASA learn-to-fly technology development," *2018 Atmospheric Flight Mechanics Conference*, 2018, p. 3310.
- [19] Petersen, J. A., and Bodson, M., "Constrained quadratic programming techniques for control allocation," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 1, 2005, pp. 91–98.
- [20] Durham, W. C., "Attainable moments for the constrained control allocation problem," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1371–1373.
- [21] Enns, D., "Control allocation approaches," *Guidance, navigation, and control conference and exhibit*, 1998, p. 4109.
- [22] Buffington, J. M., "Modular control law design for the innovative control effectors (ICE) tailless fighter aircraft configuration 101-3," *Ohio: Airforce Research Laboratory*, 1999.
- [23] Burken, J. J., Lu, P., Wu, Z., and Bahm, C., "Two reconfigurable flight-control design methods: Robust servomechanism and control allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001, pp. 482–493.
- [24] Petersen, J. A., and Bodson, M., "Interior-point algorithms for control allocation," *Journal of guidance, control, and dynamics*, Vol. 28, No. 3, 2005, pp. 471–480.
- [25] Harkegard, O., "Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation," *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002., Vol. 2, IEEE, 2002, pp. 1295–1300.
- [26] Bodson, M., "Evaluation of optimization methods for control allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711.
- [27] Luenberger, D. G., Ye, Y., et al., *Linear and nonlinear programming*, Vol. 2, Reading, MA: Addison-Wesley, 1984.
- [28] González, A. G., Rodríguez, J., Sagartzazu, X., Schumacher, A., and Isasa, I., "Multiple coherence method in time domain for the analysis of the transmission paths of noise and vibrations with non stationary signals," *Proceedings of ISMA 2010*, 2010.