



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

Luke J. Miller
NASA Langley Research Center
Flight Dynamics Branch (D317)

AIAA SciTech Forum
January 12th, 2024



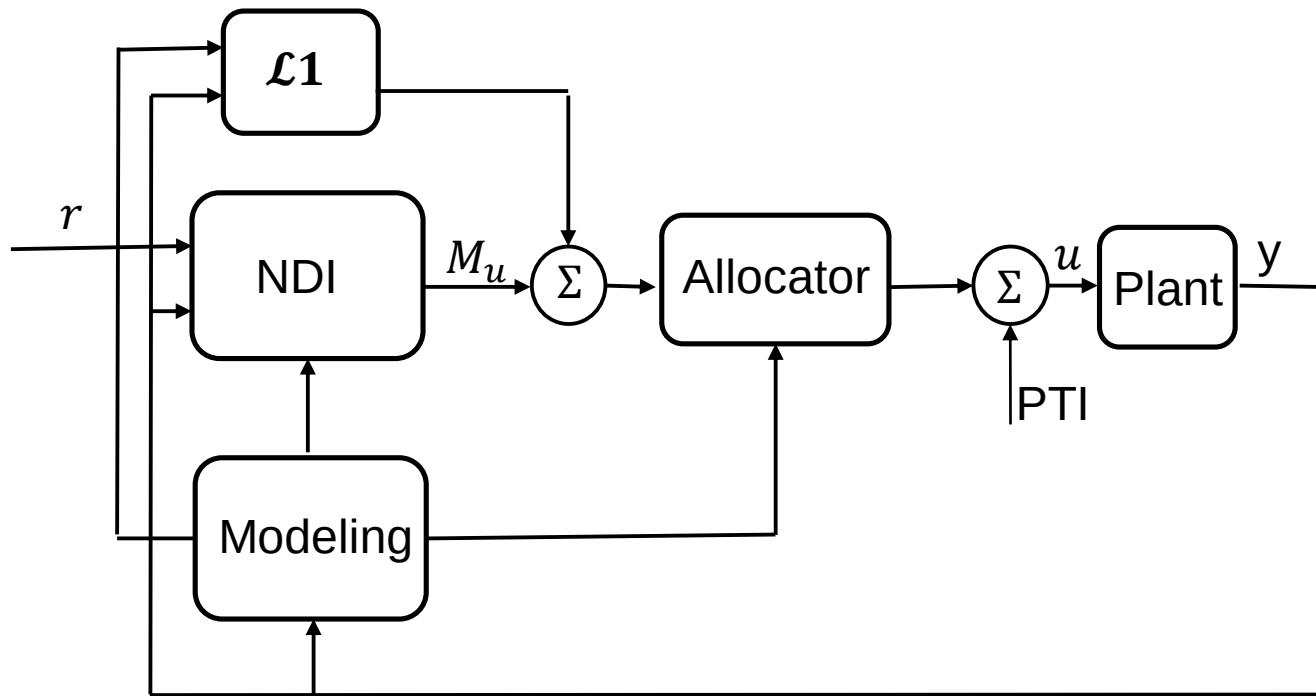
Background



- Performed under NASA Modeling and Control for Agile Aircraft Development (MCAAD) project.
- Aimed at reducing the time and cost of wind tunnel testing while developing new aircraft.
- Performed by simultaneous real-time modeling and real-time control design during flight.
- Previously performed with NDI-based control laws with adaptive $\mathcal{L}1$ augmentation on the NASA E1 aircraft under Learn To Fly (L2F) project.

NASA E1 Aircraft:





L2F Architecture:

- Nonlinear Dynamic Inversion (NDI) control law based on simplified equations of motion.
- Adaptive $\mathcal{L}1$ augmentation to assist the NDI and maintain stability for an aerodynamically unstable vehicle.
- Allocator commands control surface deflections to realize commanded moments.

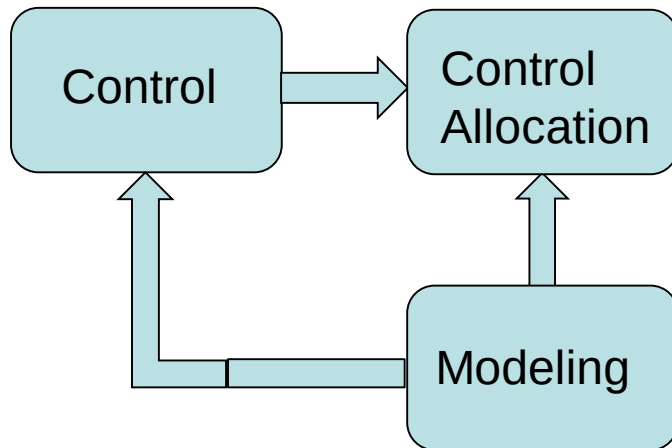


Control Principles



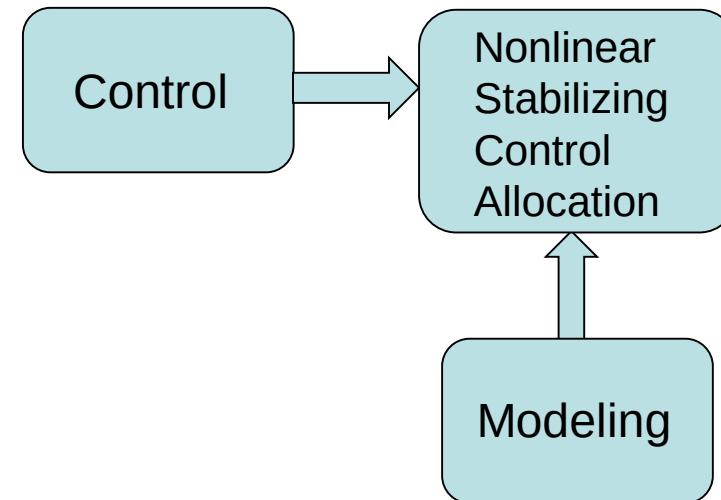
L2F Principles:

- Modeling influences both control and control allocation.
- Real time estimation of stability and control derivatives
- Stability derivatives adjust controller in real-time.
- Control derivatives adjust allocator in real-time.



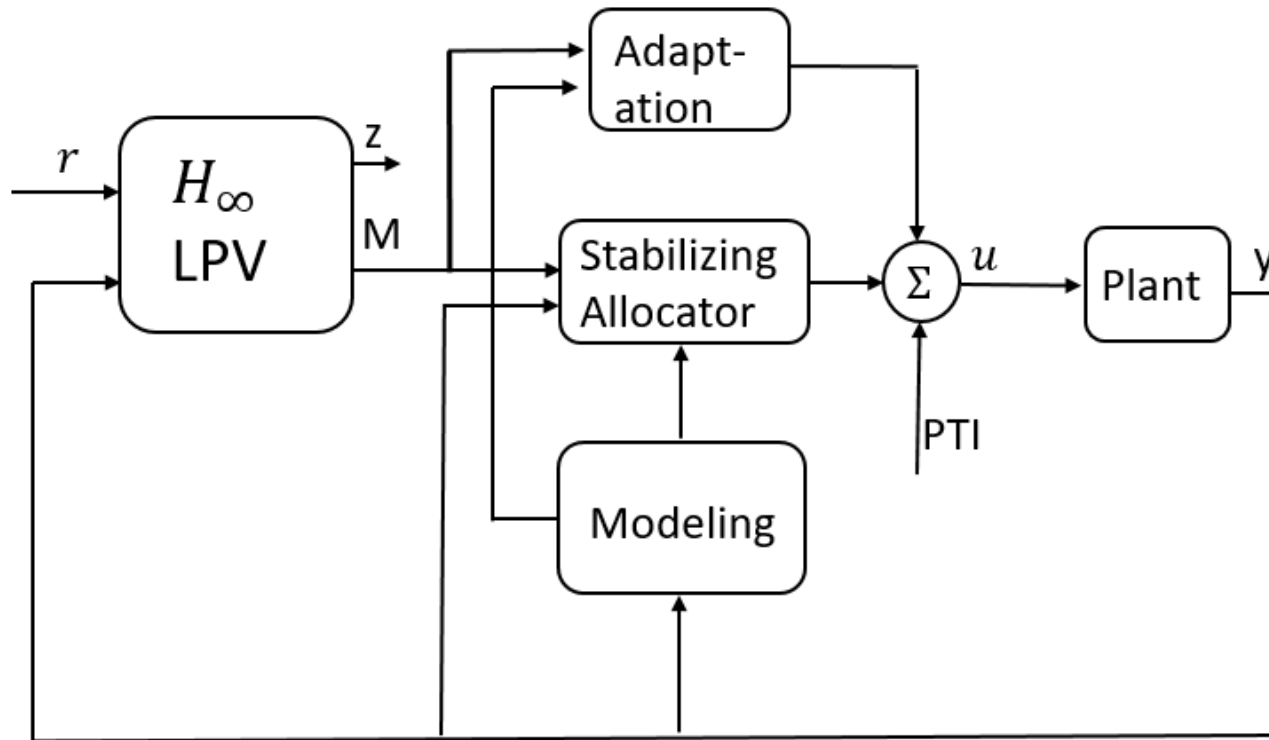
MCAAD Principles:

- Seeks to remove influence of modeling on control.
- Control is designed based on fully-known rigid body mechanics and designed offline.
- Stability derivatives and control derivatives are fed into stabilizing allocator and adjusted in real-time.



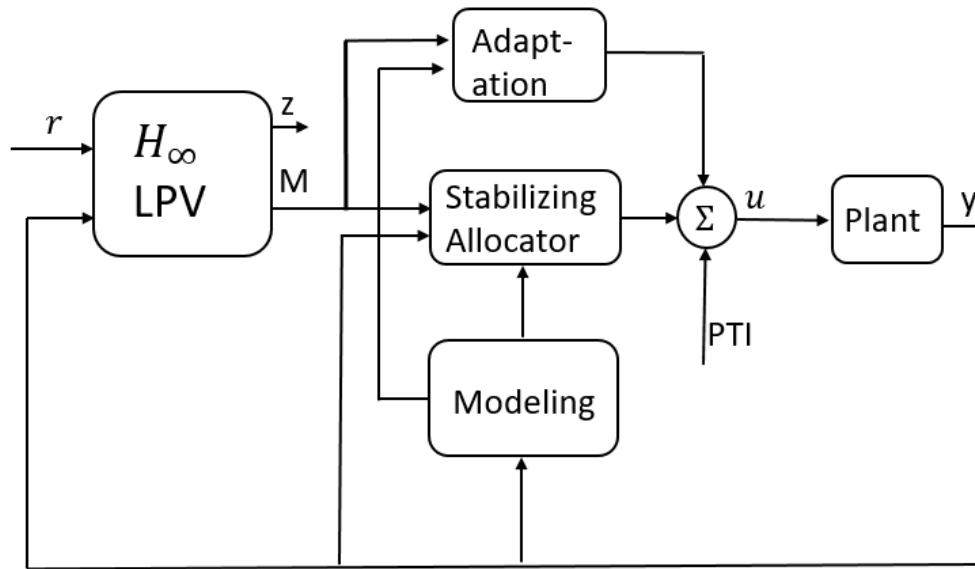
MCAAD Proposed Solution

- MCAAD (proposed full solution):



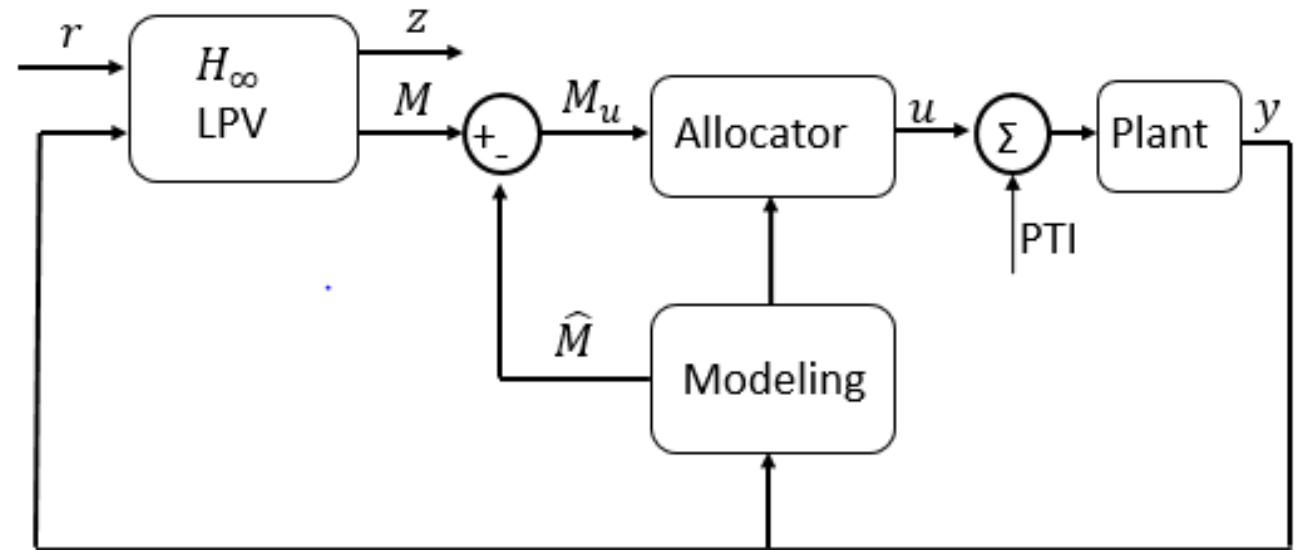
- H_∞ LPV receives a tracking command and feedback then calculates a *total* moment command. Total moment command could include moments due to:
 - Aerodynamics
 - Control surface deflections
 - Coupled aerodynamic and control surface deflections.
- Stabilizing Allocator ensures aerodynamic stability using feedback.
- H_∞ LPV globally asymptotically stabilizes the rigid-body dynamics, necessary for the development of stabilizing allocators.
- Full solution can handle unstable aircraft, this work is limited to stable aircraft.

- MCAAD (proposed full solution):



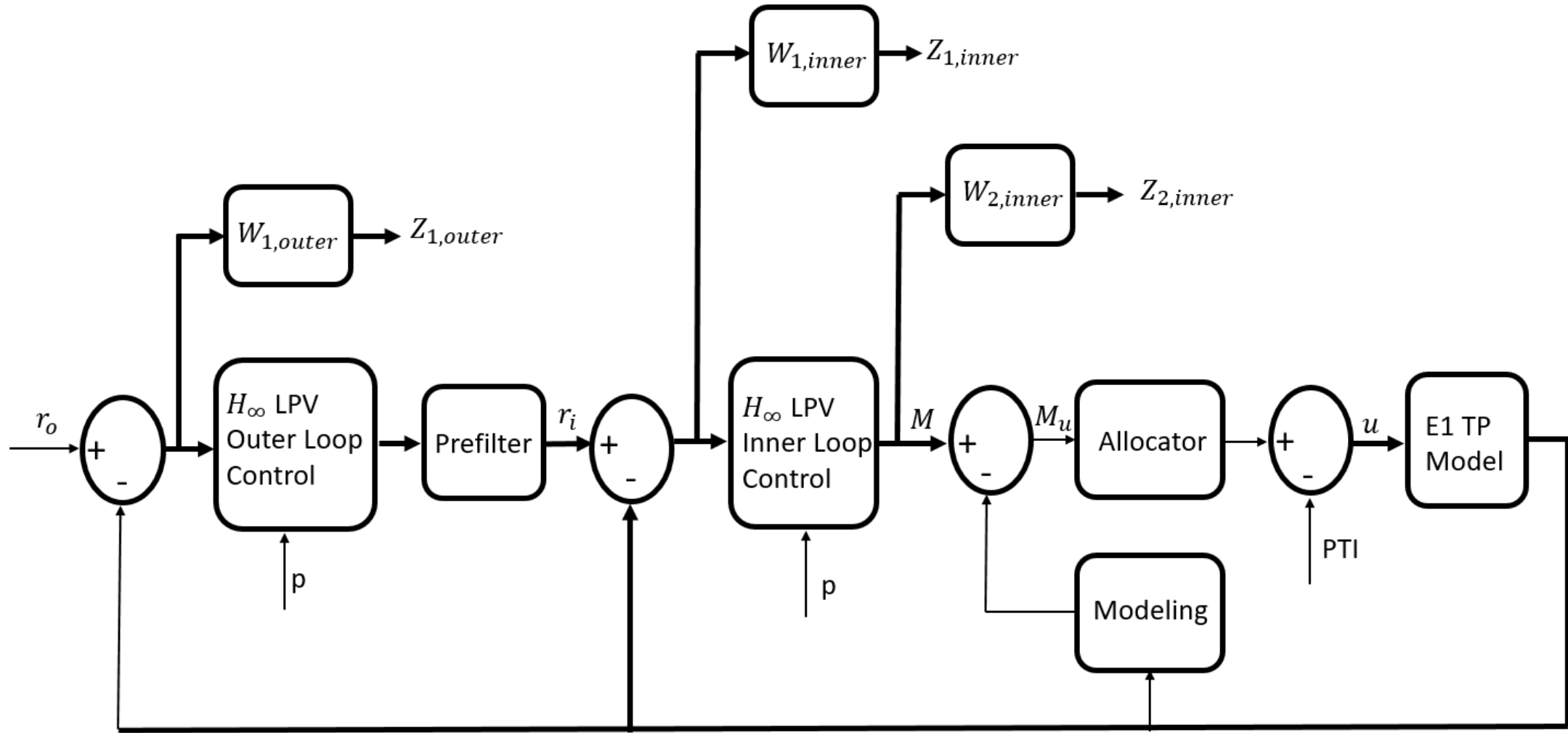
- Focus of this study (simplified solution using conventional control allocator):

- Ensures asymptotic stability.
- Will serve as an incremental step to the full solution
- This study is limited to stable aircraft.





LPV Control Design





LPV Model Derivation



LTI Systems:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- Traditional gain-scheduling control could be applied but has disadvantages.
 - No proof of global stability.
 - Performance deteriorates as system deviates from design point

LPV Systems:

- Capture nonlinearities by describing A, B, C, and D in terms of a parameter vector, $\mathbf{p}$.

$$\dot{x} = A(\mathbf{p})x + B(\mathbf{p})u$$

$$y = C(\mathbf{p})x + D(\mathbf{p})u$$

- Parameters are measurable quantities which are unknown in advance.
 - Examples: Angle of attack, attitude angles, or airspeed.

$$\dot{x}_K = A_K(\mathbf{p})x_K + B_K(\mathbf{p})e$$

$$u = C_K(\mathbf{p})x_K + D_K(\mathbf{p})e$$

- Can prove global asymptotic stability.



LPV Model Derivation



Affine Polytopic LPV systems:

- Most LPV systems do not have affine (linear) dependence on the parameter vector.
- Assume elements of $\mathbf{p}$ are bounded i.e. $p_i \in [\underline{p}_i \quad \overline{p}_i]$
- Assume $\mathbf{p}$ lies in a polytope, Θ with vertices, Θ_i
- Affine dependence allows controllers to be designed at vertices of the polytope.
- Non-Affine Systems must be converted to affine systems.
- Transformation into parameter dependent weights and parameter independent models.

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

Polytope:

- A geometric object with flat faces.



- Ziegler, Günter M. *Lectures on polytopes*. Vol. 152. Springer Science & Business Media, 2012.



Tensor Product (TP) Model Transformation:

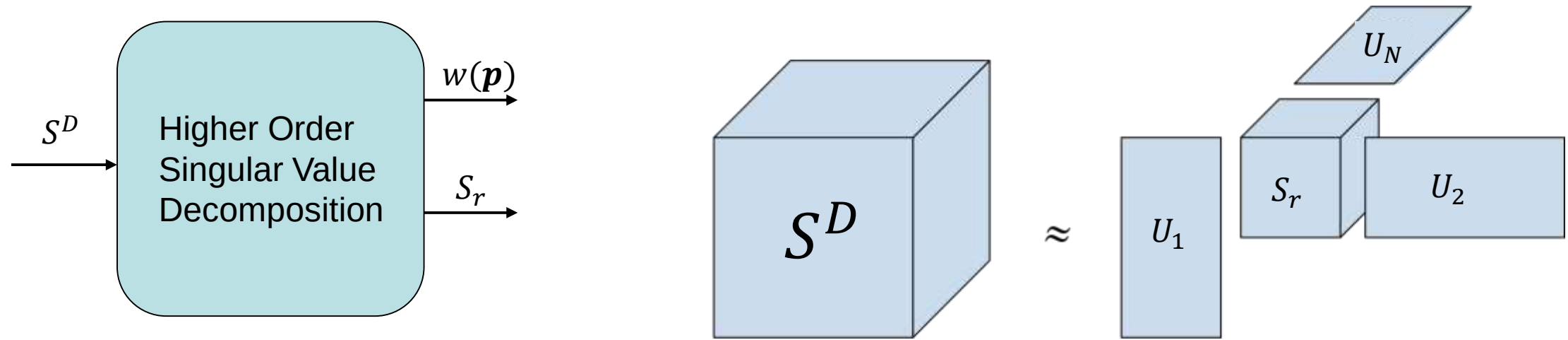
- TP Transformation transforms non-affine polytopic affine LPV model into affine LPV polytopic model.
- Write the LPV Model as $S(\mathbf{p}) = \begin{bmatrix} A(\mathbf{p}) & B(\mathbf{p}) \\ C(\mathbf{p}) & D(\mathbf{p}) \end{bmatrix} \in \mathbb{R}^{(n+q) \times (n+m)}$
 - n is the number of states
 - q is the number of outputs
 - m is the number of inputs
- LPV model is evaluated at grid points and stored as a tensor:
 - $S^D \in \mathbb{R}^{M_1 \times \dots \times M_p \times (n+q) \times (n+m)}$
 - $M_1, \dots, M_p$ are the dimensions of the parameter space

Table 2 Parameter bounds and points

Parameter	Min.	Max.	Unit	Grid Points
ϕ	-45	45	deg	50
θ	-20	20	deg	50
V	65	105	ft s ⁻¹	10



LPV Model Derivation





LPV Model Derivation



Outer Loop Equations of Motion:

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

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

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

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

Inner Loop Equations of Motion:

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

$$\dot{\theta} = \omega_2 \cos \phi - \omega_3 \sin \phi$$

$$\dot{\omega} = J^{-1}(M + M_u) + J^{-1}(\omega \times J\omega)$$



LPV Model Derivation



Simplifying Assumptions for Outer Loop:

$$\begin{aligned}\mu &\approx \phi \\ \cos \alpha &\approx \cos \gamma \approx 1 \\ Y &\approx 0 \\ \beta &\approx 0 \\ L \cos \phi &\approx mg \\ \sin \alpha \omega_1 &\approx 0\end{aligned}$$

Simplified Equations of Motion:

$$\begin{aligned}\dot{\alpha} &= \omega_2 - \frac{g}{V} \sin \phi \tan \phi \\ \dot{\beta} &= -\omega_3 + \frac{g}{V} \sin \phi \\ \dot{\gamma} &= 0 \\ \dot{\chi} &= \frac{g}{V} \tan \phi\end{aligned}$$

- Inner and Outer Loop Models are linearized about an equilibrium in straight and level flight.
- These assumptions are highly restrictive but have served to demonstrate the efficacy of the L2F controls project and will serve to demonstrate the efficacy of the MCAAD controls project.



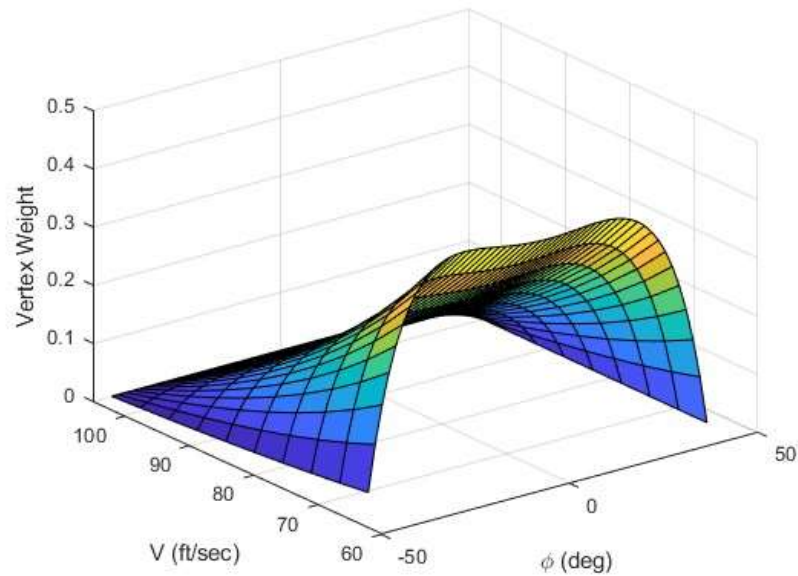
LPV Model Derivation



Outer Loop:

$$S(\mathbf{p}) = \sum_{i=1}^8 w_{o,r}(\mathbf{p}) S_{o,r}$$

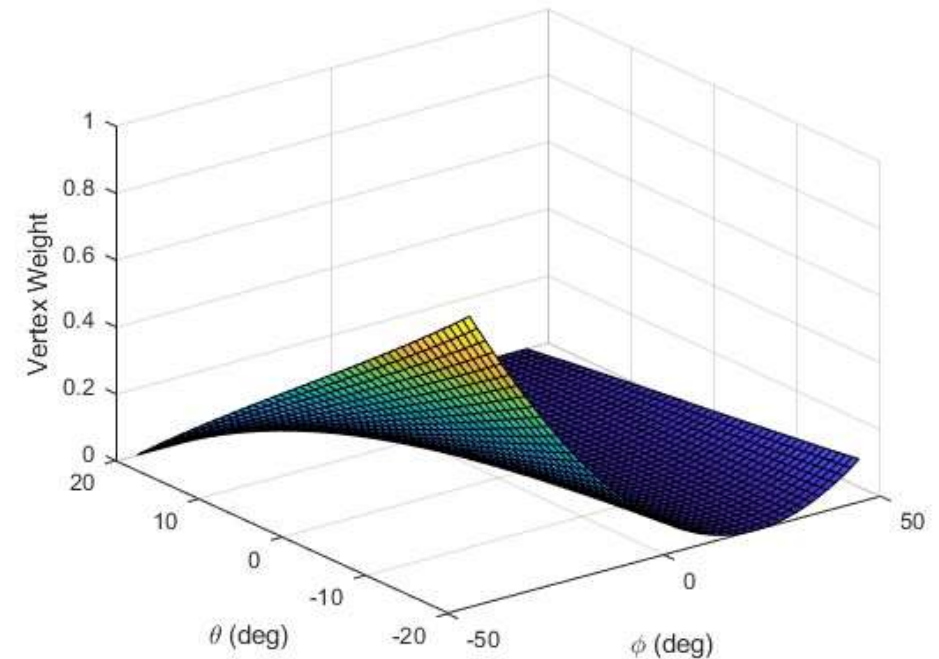
Weighting function $w_{o,2}$:



Inner Loop:

$$S(\mathbf{p}) = \sum_{i=1}^9 w_{i,r}(\mathbf{p}) S_{i,r}$$

Weighting function $w_{i,1}$:





Excitation for Real-Time Modeling

- Uses orthogonal multisine programmed test inputs (PTI) at the control surfaces.
- Real-time modeling observes the response due to the PTI at a wide range of explanatory variables across the flight envelope to accurately construct global aerodynamic models.
- Introduces a fundamental conflict. Good control mitigates disturbances which mitigates modeling effectiveness.

PTI FREQUENCY 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

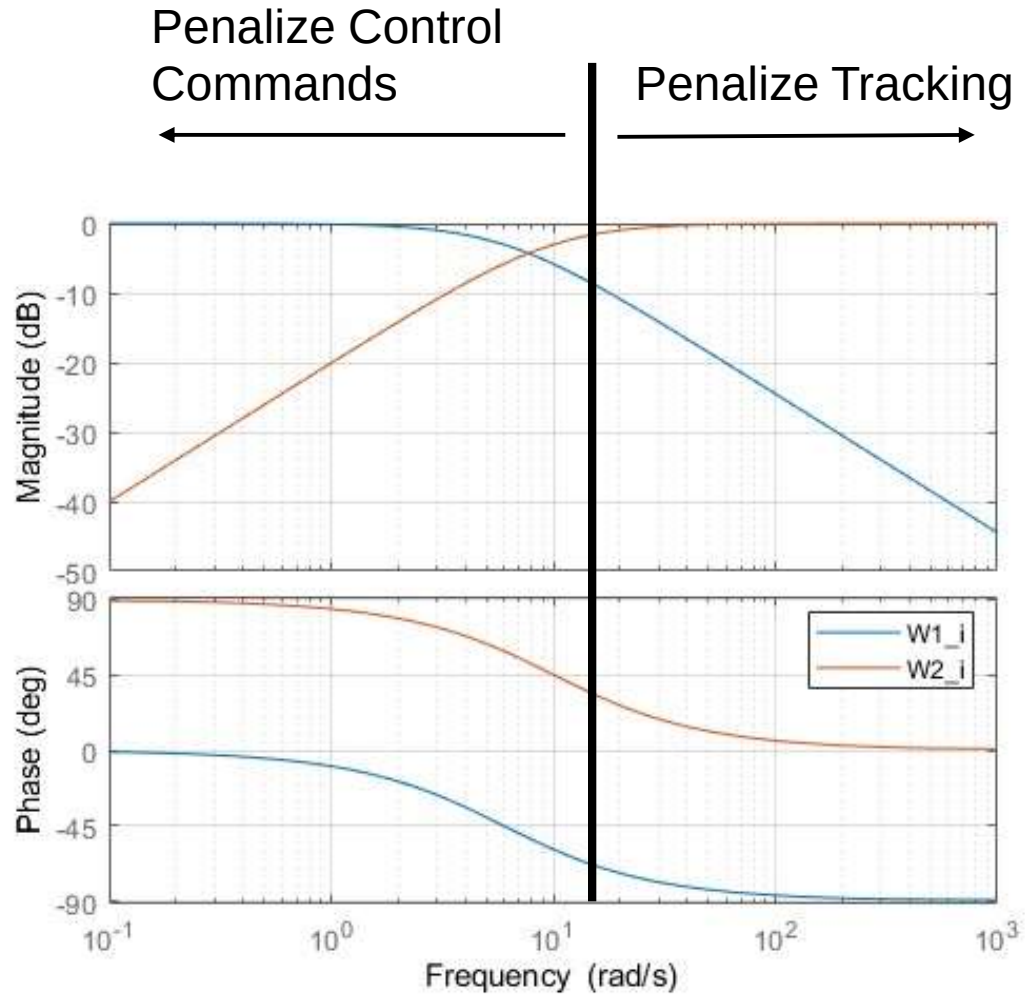
$$u_j = \sum_{k \in \{1,2,\dots,M\}} A_k \sin \left(\frac{2\pi kt}{T} + \phi_k \right)$$



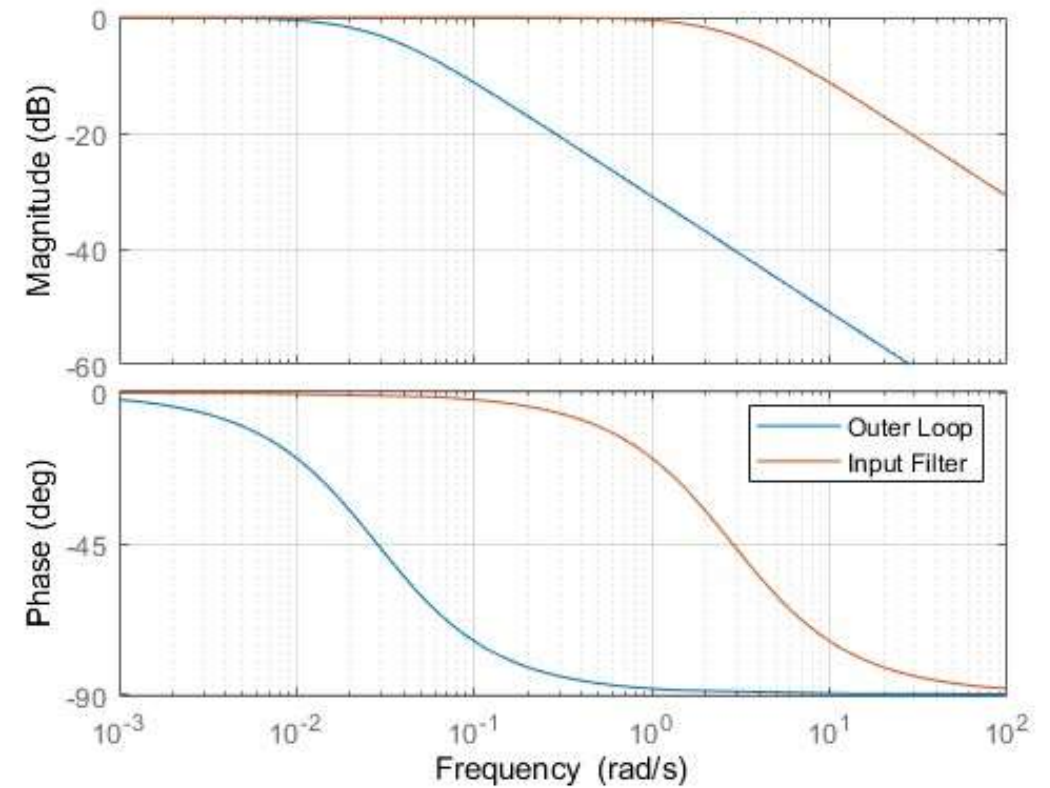
LPV Control Design



Inner Loop Bode Plots:



Outer Loop Bode Plots:





LPV Control Design

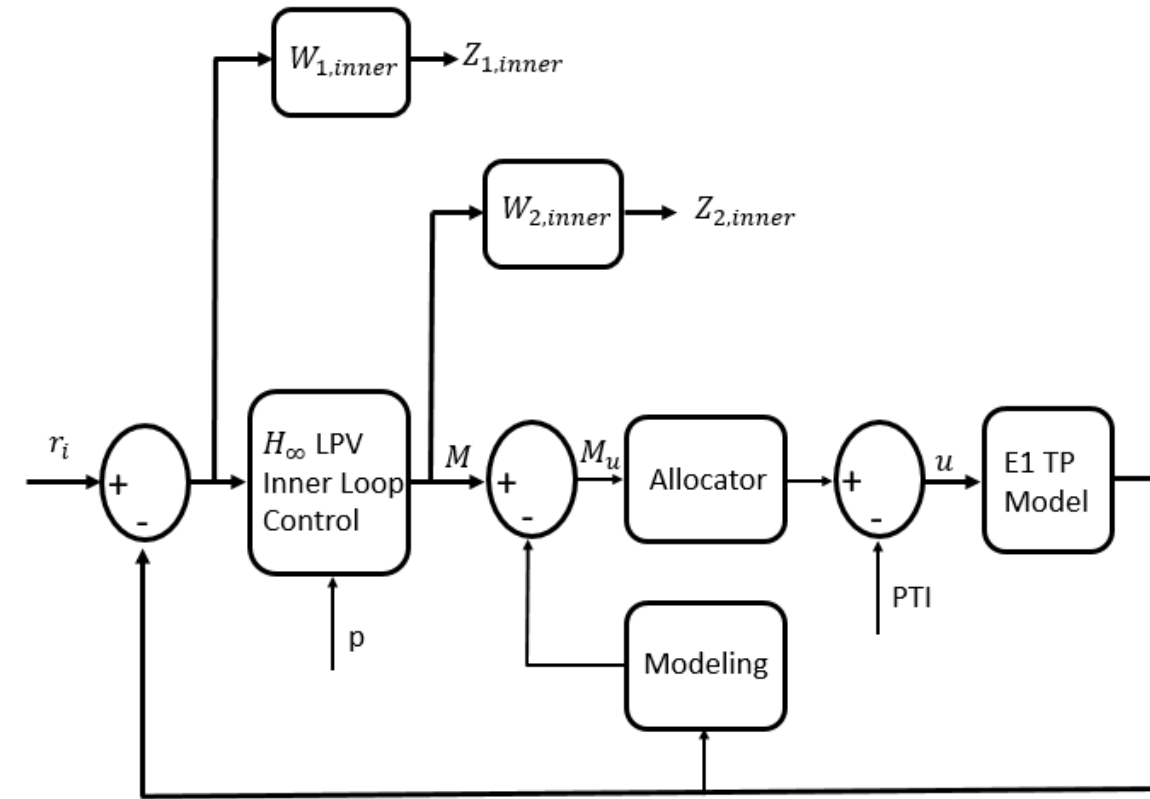


Inner Loop:

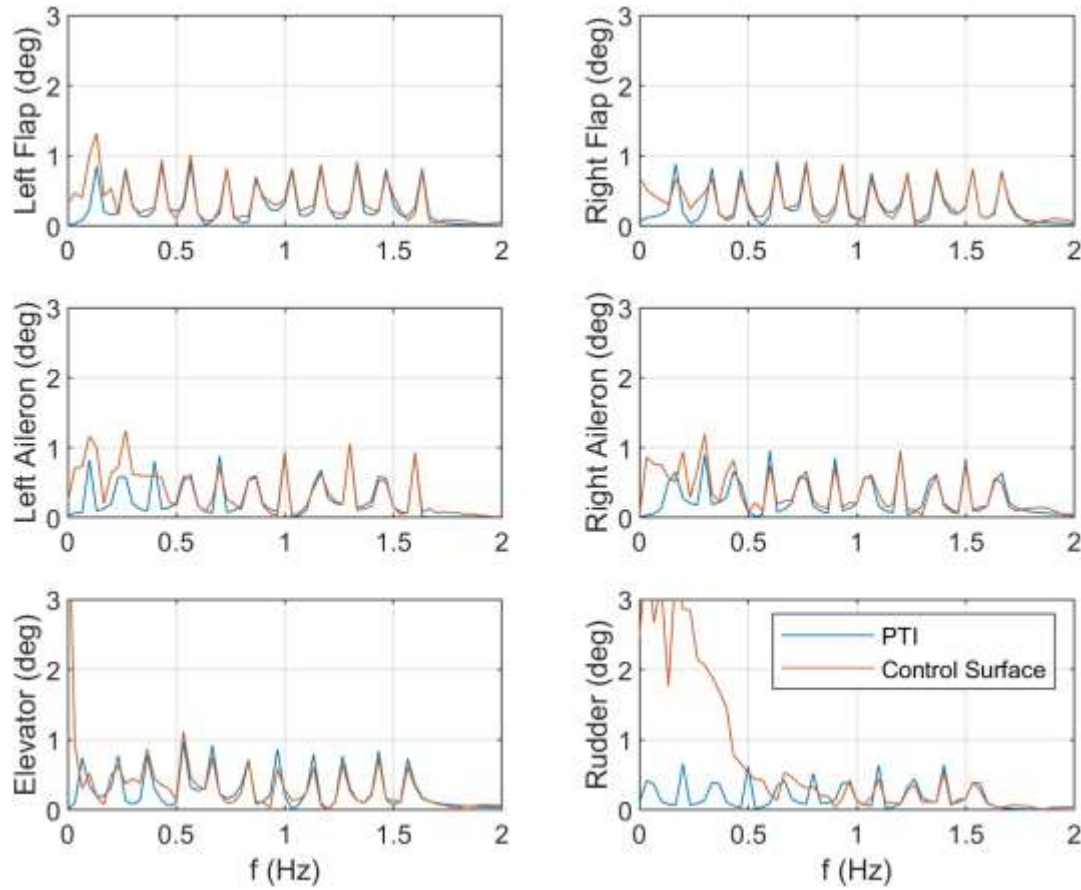
$$\dot{x}_{k,i} = \sum_{r=1}^9 w_{r,i}(\mathbf{p}) \left(A_{K,r,i} x_{K,i} + B_{K,r,i} \begin{bmatrix} \phi_{cmd,sat} \\ \theta_{cmd} \\ \omega_{1,cmd} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \end{bmatrix} \right)$$

$$M_{u,cmd} = \sum_{r=1}^9 w_{r,i}(\mathbf{p}) \left(C_{K,r,i} x_{K,i} + D_{K,r,i} \begin{bmatrix} \phi_{cmd,sat} \\ \theta_{cmd} \\ \omega_{1,cmd} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \end{bmatrix} \right) - \hat{M}$$

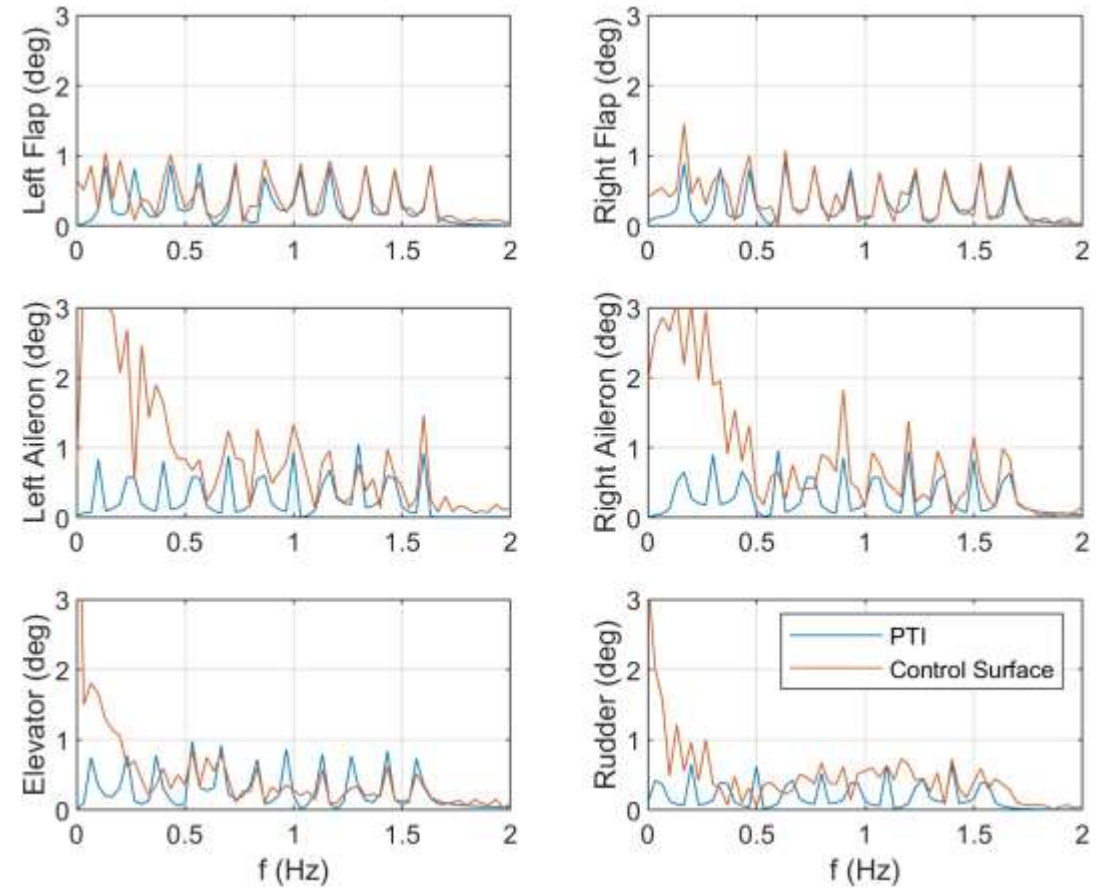
$\hat{M}$ is the estimated aerodynamic moment from the real-time modeling procedure.



LPV Frequency Domain Content:



NDI Frequency Domain Content:



- Tracking performance not shown here, shows some limitations but functional tracking.



Conclusions



- Demonstrated an incremental development of a complete MCAAD control architecture.
 - Satisfies design assumptions necessary to develop stabilizing control allocation.
 - Shapes control content better than NDI
- This study isolated the H_∞ LPV without the stabilizing allocator.
- Simulation results shows has less distortion of PTI which are required for real-time modeling, compared to previously developed NDI.
- Global exponential stability necessary for development of stabilizing allocator.
- Future work will focus on stabilizing control allocation, adaptive control allocation and will fully remove the aerodynamic dependency on the controller, relegating all aerodynamics to the allocator.



Backup Charts





Stabilizing Allocator Example



Unstable equations of motion single axis pitch example:

$$C_M = C_{M_0} + C_{M_\alpha} \alpha + C_{M_{\delta_e}} \delta_e + C_{M_q} \tilde{q}$$

Where:

$$C_{M_0} = -2.2029370 \times 10^{-2}$$

$$C_{M_\alpha} = 4.660702 \times 10^{-2}$$

$$C_{M_{\delta_e}} = -6.012308 \times 10^{-1}$$

$$C_{M_q} = -5.159153$$

From nonlinear F-16 Model, linearized at:

$$\alpha = 0, \delta_e = 0, \tilde{q} = 0.$$

Also, $x_{cg_{ref}} = x_{cg}$.

- Morelli, Eugene A. "Global nonlinear parametric modelling with application to F-16 aerodynamics." *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*. Vol. 2. IEEE, 1998.



Stabilizing Allocator Example



Control Design:

Equations of Motion:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 1/I_{yy} \end{bmatrix} M_2$$

$$I_{yy} = 55,814 \text{ slug-ft}^2$$

Controller (LQR):

$$M = -K \begin{bmatrix} \alpha \\ q \end{bmatrix}$$

$$K = [7 \times 10^6 \quad 7.38 \times 10^6]$$

Stabilizing Allocator:

$$C_M = C_{M_0} + C_{M_\alpha} \alpha + C_{M_{\delta_e}} \delta_e + C_{M_q} \tilde{q}$$

$$C_M = \frac{M}{\rho V^2 S \bar{c}}$$

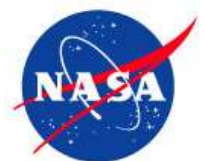
$$-25 \text{ deg} \leq \delta_e \leq 25 \text{ deg} \quad -150 \frac{\text{deg}}{\text{sec}} \leq \dot{\delta}_e \leq 150 \frac{\text{deg}}{\text{sec}}$$

Actuator Lag:

$$H = \frac{(40)(100)}{(s + 40)(s + 100)}$$

Flight Parameters:

$$\rho = .002 \frac{\text{slug}}{\text{ft}^3} \quad V = 880 \frac{\text{ft}}{\text{sec}} \quad S = 300 \text{ ft}^2 \quad \bar{c} = 11.32 \text{ ft}$$



Stabilizing Allocator Block Diagram

- Solution to Lagrange Optimization Problem:

$$\mathcal{L}(x, u, \lambda) = J(x, u) + \lambda^T (M_c - h(x, u))$$

- $J(x, u)$ is a cost function
- λ is a co-state
- M_c is a moment command
- $h(x, u)$ is the moment predicted by the model

- Design Based on Lyapunov Function:

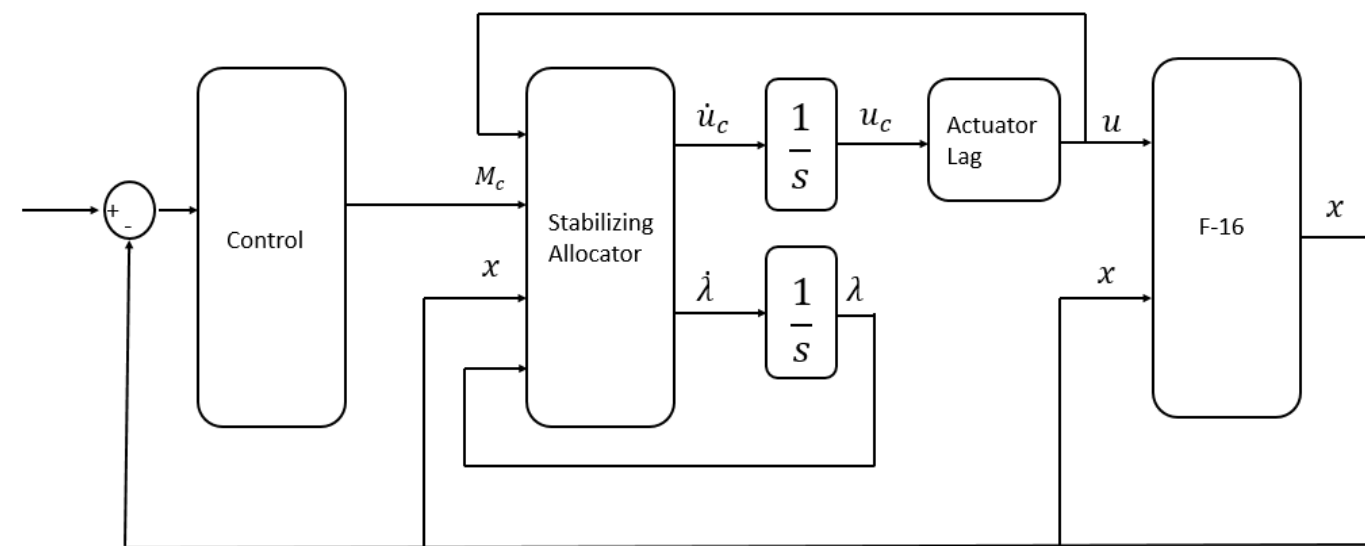
$$V(x, u, \lambda) \geq 0$$

$$\text{For stability, } \dot{V}(x, u, \lambda) < 0$$

- To ensure $\dot{V}(x, u, \lambda) < 0$ control laws are derived in the form of:

$$\dot{u} = f(x, u, \lambda, M_c)$$

$$\dot{\lambda} = g(x, u, \lambda, M_c)$$

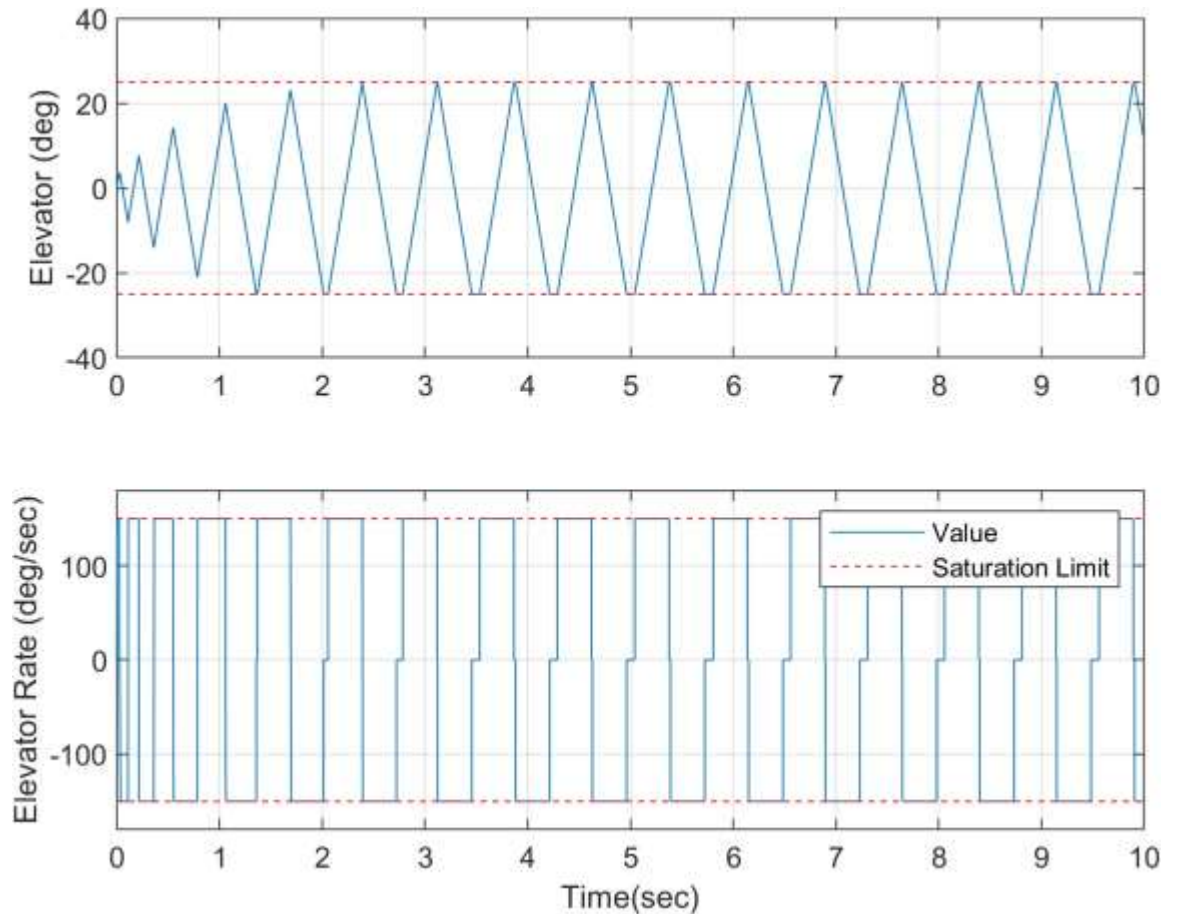
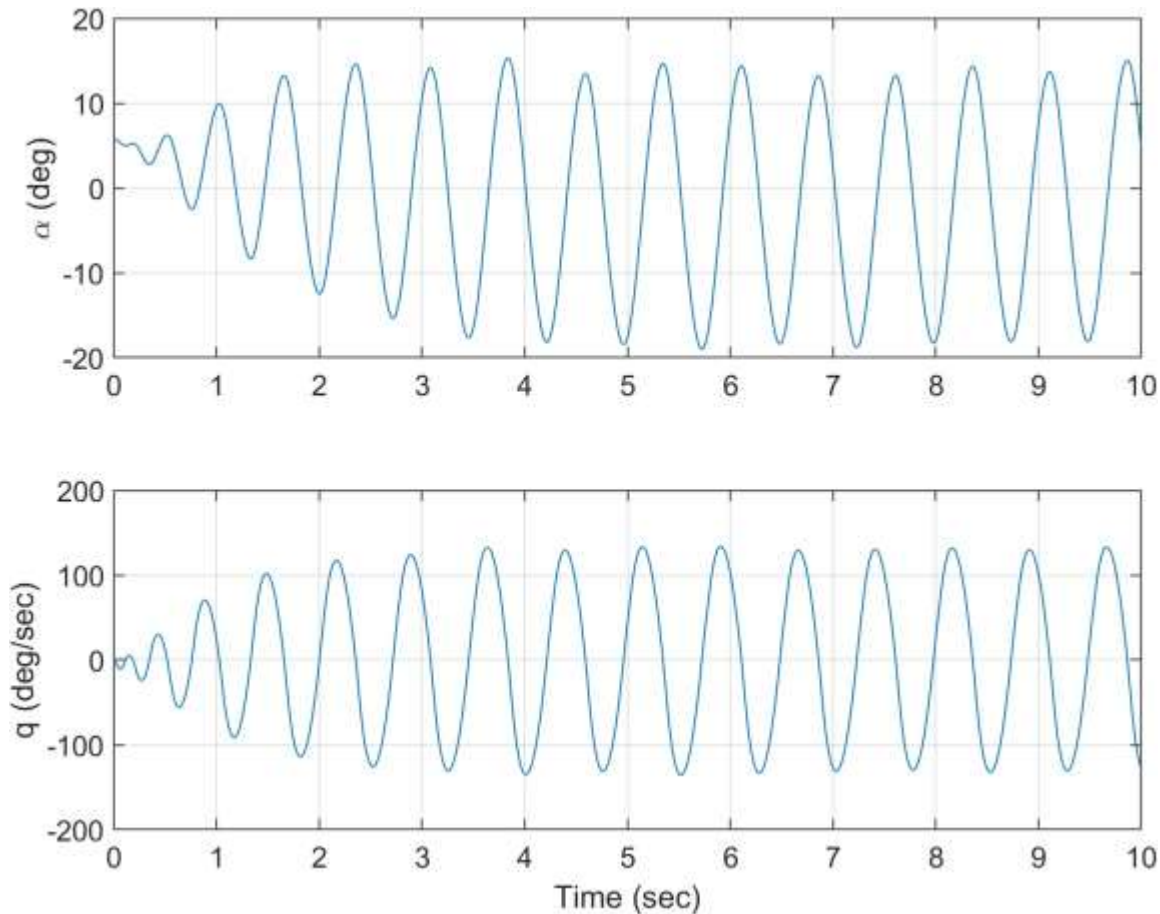




Stabilizing Allocator Simulation



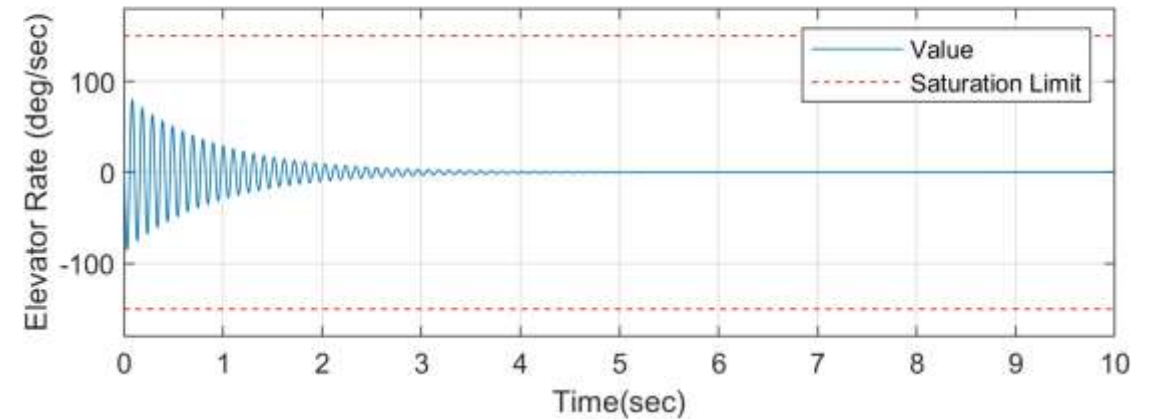
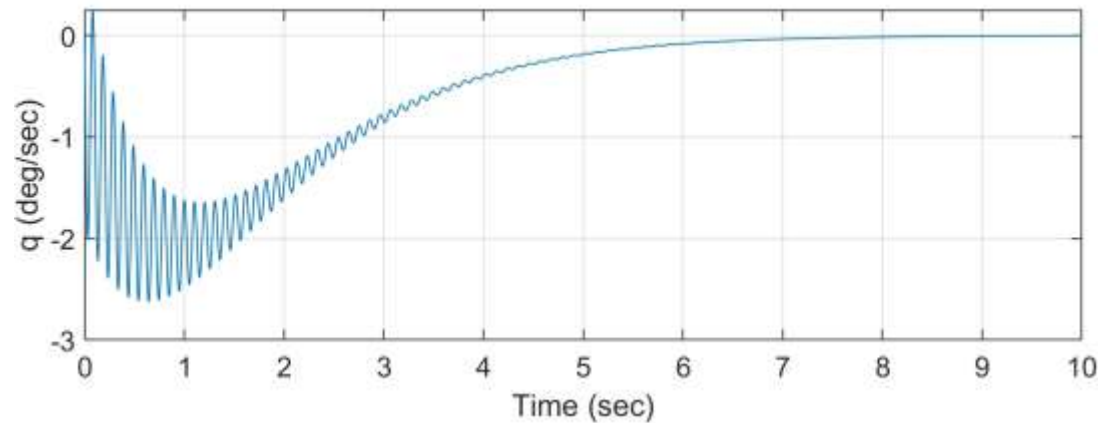
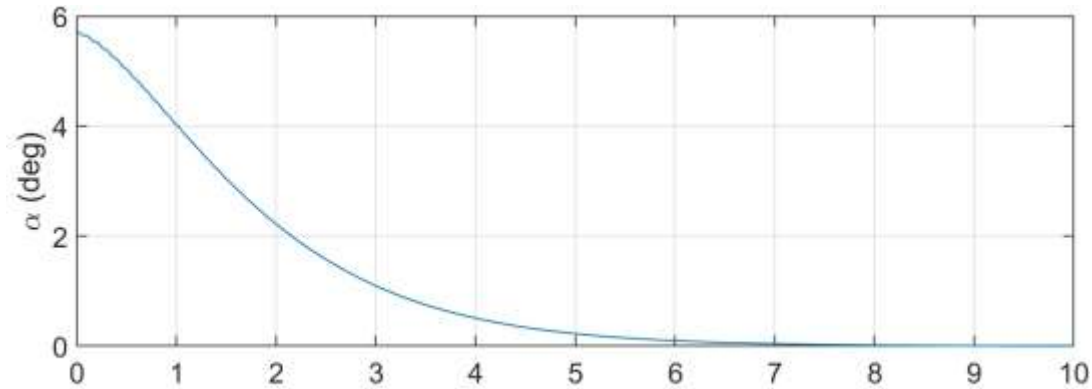
- Generalized inverse is fully determined (1 Moment and 1 Control Surface), but stabilizing allocator is fundamentally different than the generalized inverse thus the allocation solutions are not the same.
- Generalized Inverse:





Stabilizing Allocator Simulation

- Generalized inverse is fully determined (1 Moment and 1 Control Surface), but stabilizing allocator is fundamentally different than the generalized inverse thus the allocation solutions are not the same.
- Stabilizing Allocator:





Control Allocation



- Constrained Optimization Problem:

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

- B is determined from real time modeling algorithm
- u_p is the preferred control surface deflection
- Solved in real time using a simplex optimization algorithm.



Control Design by Bounded Real Lemma:

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

- Bounded Real Lemma established quadratic H_∞ performance, Γ .

$$\|y\| < \Gamma \|u\|$$

- Positive definite P solutions establish asymptotic stability with Lyapunov function, V .

$$V = x^T P x$$

Drawback:

- Infinite possible trajectories of $\mathbf{p}$, requires infinite solutions of Bounded Real Lemma.
- Assume elements of $\mathbf{p}$ are bounded i.e. $p_i \in [\underline{p}_i \quad \overline{p}_i]$



LPV Control Design



- $W_{1,outer}$ and $W_{1,inner}$ are low-pass filters, with cutoff frequencies:

$$\omega_{c,0} = .0045 \text{ Hz}$$

$$\omega_{c,1} = 1.6 \text{ Hz}$$

- $W_{2,inner}$ is a high-pass filter, with cutoff frequency at highest PTI frequency:

$$\omega_{c,2} = 1.6 \text{ Hz}$$

- $W_{prefilter}$ is a low-pass filter, with high cutoff frequency.
 - Removes parameter dependency on outer loop, B matrix.

$$\omega_{c,prefilter} = 0.5 \text{ Hz}$$



Outer Loop:

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

$$\phi_{cmd,sat} = Sat_{-45^\circ}^{45^\circ} \phi_{cmd}$$

$$\theta_{cmd} = \alpha_{cmd}$$

$$\omega_{1,cmd} = 0$$

Inner Loop:

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

$\hat{M}$ is the estimated aerodynamic moment from the real-time modeling procedure.



Outer Loop:

$$\begin{aligned}\dot{\mathbf{x}} &= A(\mathbf{p})\mathbf{x} + B_1(\mathbf{p})r + B_2e \\ \mathbf{z} &= C_1(\mathbf{p}) + D_{11}(\mathbf{p})r + D_{12}e \\ y &= C_2x + D_{21}r + D_{22}e\end{aligned}$$

$$\begin{aligned}r &= [\alpha_{cmd} \quad \beta_{cmd} \quad \chi_{cmd}] \\ \mathbf{x} &= [x_p^T \quad x_{1,o}^T \quad x_{prefilter}^T]^T \\ \mathbf{z} &= z_{1,outer}^T\end{aligned}$$

Inner Loop:

$$\begin{aligned}\dot{\mathbf{x}} &= A(\mathbf{p})\mathbf{x} + B_1(\mathbf{p})r + B_2M \\ \mathbf{z} &= C_1(\mathbf{p}) + D_{11}(\mathbf{p})r + D_{12}M \\ y &= C_2x + D_{21}r + D_{22}M\end{aligned}$$

$$\begin{aligned}r &= [\phi_{cmd} \quad \omega_{2,cmd} \quad \omega_{3,cmd}] \\ \mathbf{x} &= [x_p^T \quad x_{1,i}^T \quad x_{2,i}^T]^T \\ \mathbf{z} &= [z_{1,inner}^T \quad z_{2,inner}^T]^T\end{aligned}$$

- Inner and Outer Loop Controllers are built using HINFGS MATLAB command



LPV Model Derivation



Inner Loop LPV Model:

Trim and linearize about $\omega = [0 \ 0 \ 0]^T$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & 0 & 0 & \cos \phi & -\sin \phi \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \mathbf{M}$$

Elements of the parameter vector are variables with remain in linearized system:

$$\mathbf{p}_i = [\phi \ \theta]^T$$



LPV Model Derivation



Outer Loop LPV Model:

Trim and linearize about $\omega = [\omega_1 \quad \omega_2 \quad \omega_3]^T$

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\chi} \end{bmatrix} = 0_{3 \times 3} \begin{bmatrix} \alpha \\ \beta \\ \chi \end{bmatrix} + \begin{bmatrix} -\frac{g}{V} (\cos \phi \tan \phi + \sin \phi \sec^2 \phi) & 1 & 0 \\ \frac{g}{V} \cos \phi & 0 & -1 \\ \frac{g}{V} \sec^2 \phi & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

Elements of the parameter vector are variables with remain in linearized system:

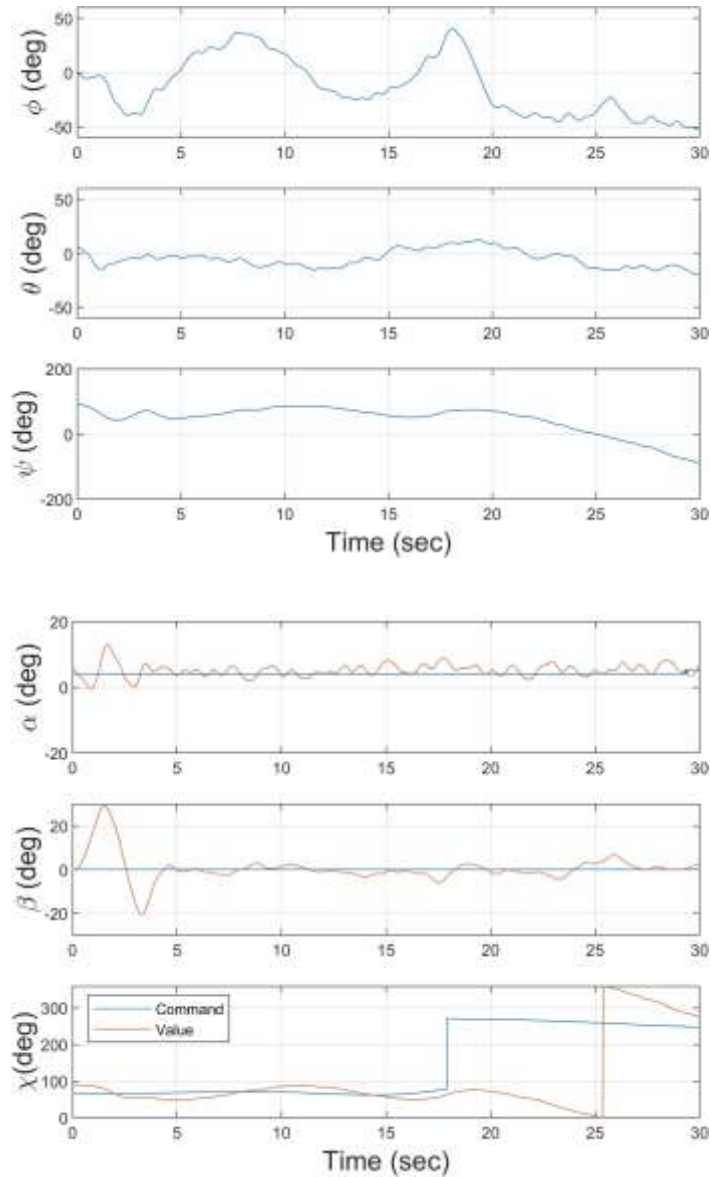
$$\mathbf{p}_o = [\phi \quad V]^T$$



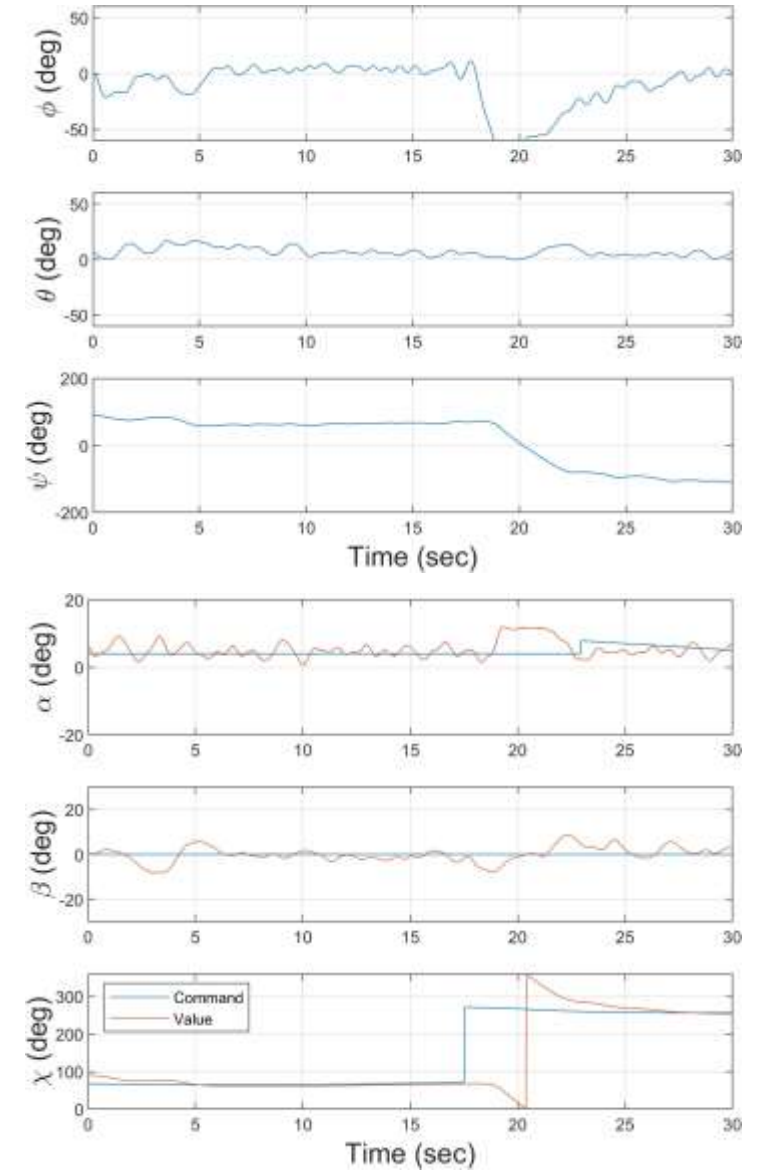
Simulation Results



LPV Control:



NDI Control:

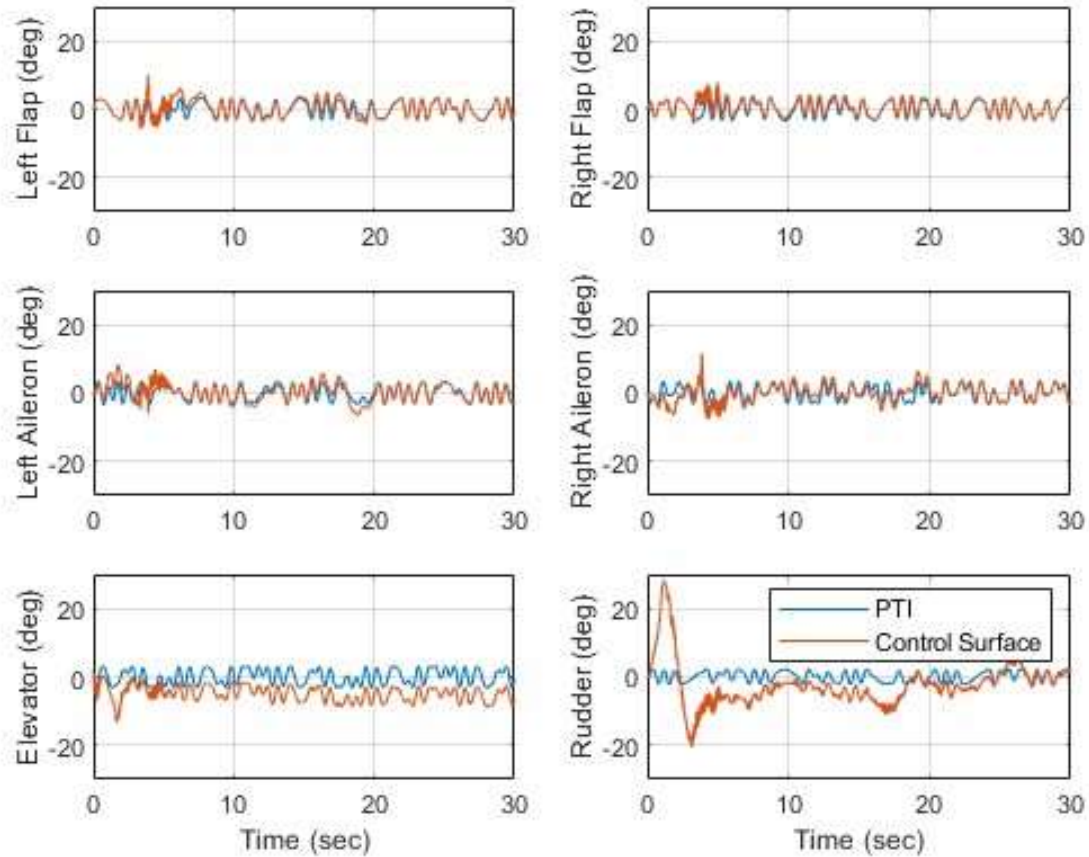




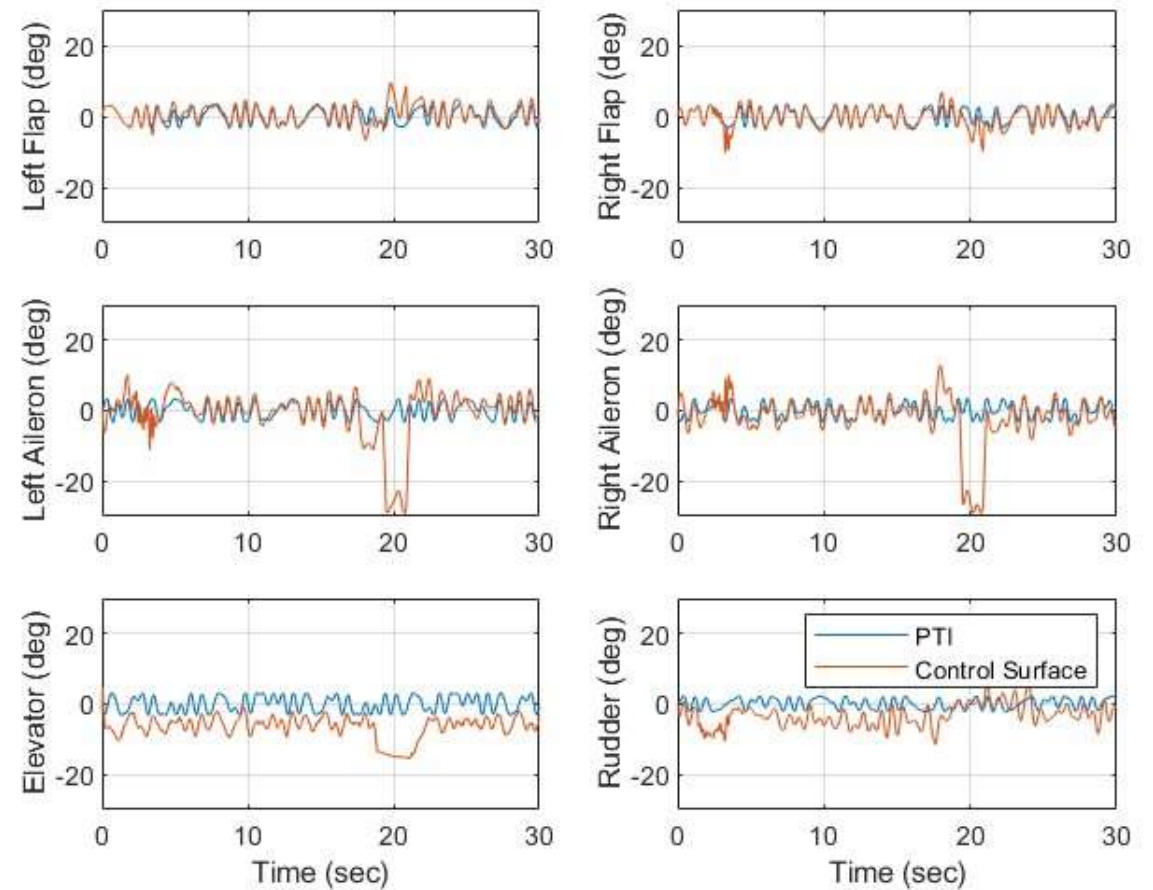
Simulation Results



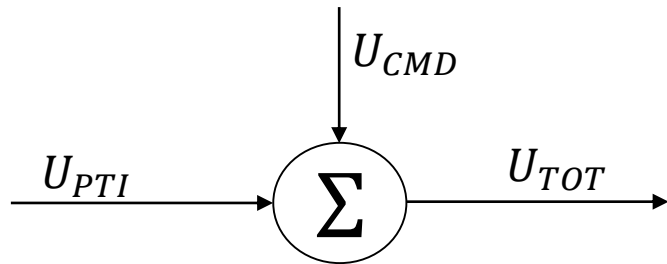
LPV Control Surface Deflections:



NDI Control Surface Deflections:



- Root-Mean-Squared error in time domain
- Used a similarity measure



RMSE	LPV (deg)	NDI (deg)
Left Flap	1.2451	1.8380
Right Flap	1.2052	1.5486
Left Aileron	1.6570	6.7097
Right Aileron	1.8307	6.6201
Elevator	5.0890	7.3421
Rudder	6.8781	4.5589