

Linear Parameter Varying Control for the X-53 Active Aeroelastic Wing

Peter Seiler, Gary J. Balas and Andrew Packard

Abstract Fuel efficiency, endurance, and noise requirements are pushing modern aircraft to lighter, more flexible designs. This causes the structural modes to occur at lower frequencies increasing the coupling with the rigid body dynamics. The traditional approach to handle aeroservoelastic interaction is to design gain-scheduled flight control laws based on the rigid body dynamics and then use filters to avoid exciting the structural modes. This decoupled approach may not be possible in future, more flexible aircraft without reducing the flight control law bandwidth. Linear parameter varying (LPV) techniques provide a framework for modeling, analysis, and design of the control laws across the flight envelope. This chapter applies LPV techniques for the roll control of NASA Dryden's X-53 Active Aeroelastic Wing testbed. LPV techniques are first used to analyze a gain-scheduled classical controller. Gain scheduling is still the dominant design method in industrial flight control laws and LPV analysis tools can play an important role in certifying the performance of these systems. Next, an LPV controller is designed and its performance is compared against the gain-scheduled classical controller. All results are obtained with a set of LPV tools which makes use of object oriented programming to enable easy construction and manipulation of LPV models.

1 Introduction

Increased fuel efficiency and operational range are significant design drivers for modern commercial aircraft, e.g. the Boeing 787. Similar design objectives are also

Peter Seiler
MUSYN, Inc., Minneapolis MN, e-mail: peter.j.seiler@gmail.com

Gary J. Balas
MUSYN, Inc., Minneapolis MN, e-mail: balas@musyn.com

Andrew Packard
MUSYN, Inc., Berkeley, CA, e-mail: andrew.packard60@gmail.com

critical for future military aircraft, e.g. the SensorCraft concept aircraft [14, 18, 31]. In both cases, lighter aircraft are required to meet these objectives. The reduction in weight is typically achieved by reducing the structure in the wings and fuselage of the aircraft. This makes the aircraft more flexible and causes the structural modes to occur at lower frequencies. The main consequence is that lighter, more flexible aircraft have tight coupling between the rigid body and elastic structural modes. This increases the likelihood of adverse aeroservoelastic phenomena including flutter and control surface reversal.

The traditional approach to handle aeroservoelastic interaction is to design the flight control laws based on the rigid body dynamics and then use filters to avoid exciting the structural modes. The control laws are typically designed at various points in the flight envelope and then gain-scheduled by interpolating these point designs. This gain-scheduled approach may not be possible on future, more flexible aircraft for which the structural modes to occur at lower frequencies. The design will need to consider coupling between the rigid body dynamics, structural modes, and the time-varying gain-scheduled controller. Flexible aircraft would significantly benefit from an integrated aeroservoelastic and rigid body control system.

Several issues must be addressed to enable integrated active control to become a reality. First, the aeroelastic effects involve unsteady flows [16, 17, 15]. In addition, there can be nonlinear effects, e.g. nonlinear coupling between the structural modes and the aerodynamics [17, 5]. Advanced tools are needed to model these effects across the entire flight envelope. A second issue is that an integrated control design must account for the tight coupling between the rigid body and structural modes. This will likely require novel sensors that can measure, in real-time, the aerodynamic flow around the aircraft structures. Such sensors are currently being developed [16, 17, 15] and new control architectures may be required to take advantage of these novel measurements. A third issue is that analysis tools are required to certify that the designed feedback system meets structural load requirements and is free from aeroservoelastic instabilities. Existing approaches based on robust flutter margins [8, 13, 6] form a starting point but may need to be extended to handle the complexities introduced by the integrated design approach. To summarize, advanced tools are required for modeling, integrated controller synthesis, and analysis of flexible aircraft.

This chapter investigates the use of linear parameter varying (LPV) analysis and control techniques for flexible aircraft control. There are two main objectives of this chapter. The first is to introduce new software tools LPV modeling, analysis, and control synthesis. These tools implement existing analysis and synthesis conditions drawn from the large body of literature on LPV systems including [27, 20, 28, 33, 4, 1, 34, 19, 25, 24, 32]. Implementation of the LPV algorithms makes use of object oriented programming to enable easy construction and manipulation of LPV models. The second objective is to apply LPV techniques to NASA Dryden's X-53 Active Aeroelastic Wing (AAW) testbed [23, 21, 22]. The AAW is an experimental flight test capability for aeroservoelastic control research. This chapter will focus on roll rate control of the AAW in the supersonic regime.

The remainder of the chapter has the following outline. First, a brief review of the AAW program is given in Section 2. The LPV software data structures and their functionality are described in Section 3. Next, the AAW rigid body and aeroelastic dynamics are described in Section 4. A gain-scheduled classical control law is designed and analyzed in Section 5. Gain-scheduled classical control is the standard in industry for flight control design. LPV analysis tools can play an important role in certifying the performance of these systems and identifying potential issues due to fast variations in the gain scheduling parameters. The LPV analysis tools provide a useful complement to existing approaches, e.g. margin requirements at each flight condition or robust flutter margins [8, 13, 6]. Section 6 describes an LPV controller for the AAW and compares this design against the gain-scheduled classical design. Finally, conclusions are given in Section 7. Early versions of these results appear in [26].

2 Active Aeroelastic Wing

NASA Dryden's X-53 Active Aeroelastic Wing (AAW) [23, 21, 22] is an experimental flight test capability for aeroservoelastic control research. NASA and the USAF developed this test bed to investigate the use of aeroelastic flexibility for improved performance of high aspect ratio wings. The effectiveness of the conventional aircraft surfaces, e.g. ailerons and trailing edge flaps, is reduced at higher dynamic pressures due to the flexibility of the wing. This can lead to control reversal at sufficiently high dynamic pressures. The standard solution is to reduce wing flexibility by adding structure, and hence additional weight, to the wings.

The main objective of the AAW Flight Research program was to test an alternative concept that uses wing flexibility to improve control effectiveness. The AAW has inner and outer flaps on the leading edge of the wings. Small movements of these surfaces cause the wing to twist in the direction that increases the local angle of attack and induces a rolling moment on the aircraft. These flaps do not undergo a control reversal and, in fact, their effectiveness increases at higher dynamic pressures. Thus the wing flexibility and twist act in a direction beneficial for control.

To test this concept, the AAW wings were modified from the standard F/A-18 wings to reduce the torsional stiffness [22]. This increases the wing flexibility and reduces the frequency of the flexible modes. Advanced tools are required to model the aeroelastic effects because they involve unsteady flows [16, 17, 15] and there can be nonlinearities [17, 5]. For control design, linear models of the rigid body and aeroelastic dynamics are obtained at each flight condition via linearization. This naturally falls within the class of linear parameter varying (LPV) models that are scheduled as a function of the flight condition.

The flight-tested AAW control architecture is a modified version of the production F/A-18 control laws [11]. The basic architecture uses roll rate feedback to track roll rate commands from the pilot lateral stick inputs. The lateral controller commands the aileron, trailing edge flaps, inner leading edge flaps and outer leading

edge flaps. Each surface command is the sum of a proportional roll rate feedback term and a proportional roll rate tracking error term. The control gains were tuned to maximize roll rate performance while satisfying structural load and handling quality requirements [11]. The gains were tuned using the multi-objective optimization tool CONDUIT [30]. This tool performs nonlinear optimization incorporating results from a high fidelity simulation model.

The AAW control laws were designed and flight tested at separate design points in the flight envelope [10]. These control laws were tested during 34 Phase II test flights conducted from December 2004 through March 2005 [22]. The tests spanned the transonic and supersonic flight conditions and included 360° rolling maneuvers, 5g wind up turns, and 4g rolling pullouts. The flight test program validated the Active Aeroelastic Wing concept and was deemed a success. Additional details on the existing AAW flight control laws and flight tests can be found in [22, 11, 10].

3 Tools for LPV Analysis and Design

Linear Parameter Varying (LPV) models are time-varying, state-space models of the form:

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \quad (1)$$

where $A(\rho(t))$ is the state matrix, $B(\rho(t))$ is the input matrix, $C(\rho(t))$ is the output state matrix, $D(\rho(t))$ is the input transformation matrix, $\rho \in \mathbb{R}^{n_\rho}$ is a vector of measurable parameters, $x \in \mathbb{R}^{n_x}$ is the state, $y \in \mathbb{R}^{n_y}$ is the measurement, and $u \in \mathbb{R}^{n_u}$ is the control input. The dimensions of (A, B, C, D) are compatible with the signal dimensions.

LPV models arise in many contexts. In industrial settings, a finite collection of linear models is often used to describe the behavior of a system throughout an operating envelope. The linearized models describe the small signal behavior of the system at a specific operating point. The collection is parameterized by one or more physical variables whose values represent the operating point. For example, the LPV design model used in this chapter for the Active Aeroelastic Wing roll rate dynamics is based on a parameterized family of linearizations. The models are scheduled across the aircraft (Altitude, Mach) flight envelope. Alternatively LPV models can be constructed by considering state transformations on a class of nonlinear 'quasi-LPV' systems [27]. This method essentially ignores some nonlinear relationships in the system dynamics and hence introduces conservatism in the control design.

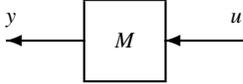
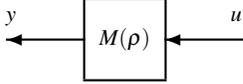
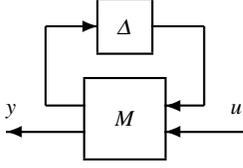
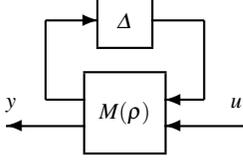
LPV software tools have been developed to enable the modeling, analysis, controller synthesis, and simulation for LPV systems. One issue is that several different methods have arisen for representing the parameter dependence in LPV models (Equation 1). These include linear fractional transformations [19, 20, 25, 24, 1], linearizations on a gridded domain [4, 33, 34], and polytopic (affine) dependence of

the state matrices on the parameters [29, 2, 9, 3]. Each of these different representations has benefits and drawbacks in terms of the modeling effort and model structure that can be exploited in developing computational algorithms. It would be useful for the controls community to have a single LPV toolbox that easily supports the different representations. Ideally this would include tools to transform an LPV model in one representation to another representation. This would enable easy comparison between the various LPV methods and with classical gain-scheduling approaches.

As a starting point, the initial implementation of the LPV algorithms focuses on models defined on gridded domains. This is motivated by aircraft aeroelastic control problems for which models are typically constructed as linearizations around various flight operating points. A key component of the software tools is the core LPV data structure object, referred to as a `pss` (denoting parameter-varying state space model). The LPV systems in Equation 1 are conceptually represented by a state-space system S that depends on a parameter vector ρ in some domain of \mathbb{R}^{n_p} . For general LPV systems this conceptual representation requires storing the state space system at an infinite number of points in the domain of ρ . The data structure object `pss` approximates this conceptual representation by storing the LPV system as a state space array defined on a finite, gridded domain. As a simple example, consider an LPV system $S(\rho)$ that depends on a single parameter ρ in the domain $\rho \in [a, b]$. The infrastructure requires the user to specify the domain with a finite grid, e.g. N points in the interval $[a, b]$. The software tools include an `rgrid` data object to facilitate the creation and manipulation of the multivariable parameter domains. The user must also specify the values of the state space system S at each point ρ in this gridded domain. The `pss` object stores the state-space array data using the standard MATLAB Control System Toolbox `ss` object. Thus the `pss` can be viewed as the parameter-varying extension of the standard `ss` object. To summarize, the LPV system $S(\rho)$ is represented by a `pss` data object which stores the gridded domain and the array that defines the state-space data at each point in the domain.

The notions of parameter-varying matrices and parameter-varying frequency responses arise naturally to complement the `pss` objects. LPV systems are time-varying and hence frequency responses can not be used to represent the system behavior as parameters vary. However frequency responses are useful to gain intuition about the system performance at fixed locations in the operating domain. The parameter varying matrices and frequency responses are represented by `pmat` and `pfrd` data objects, respectively. These two data objects are both stored as a data array defined on a gridded domain. Table 1 shows the relation between the LPV data objects (`pmat`, `pss`, `pfrd`) and existing MATLAB objects. The first row of the Table (“Nominal”) shows the basic MATLAB objects: matrices are `double` objects, state-space systems are `ss` objects, and frequency responses are `frd` objects. `double` objects are in the standard MATLAB release while the `ss` and `frd` objects are part of the Control System Toolbox. The second row of Table 1 (“Parameter Varying”) shows the core LPV objects. The LPV objects (`pmat`, `pss`, `pfrd`) can be viewed as parameter-varying extensions of the standard objects MATLAB and Control Toolbox objects (`double`, `ss`, `frd`).

Table 1: Relation between MATLAB objects

Object Type	Block	Matrix	System	Frequency Response
Nominal		double	ss	frd
Parameter Varying		pmat	pss	pfrd
Uncertain		umat	uss	ufrd
Uncertain Parameter Varying		pumat	puss	pufrd

The third row of the table (“Uncertain”) shows the equivalent objects used to represent uncertainty: uncertain matrices, state space systems, and frequency responses are represented by `umat`, `uss`, and `ufrd` objects, respectively. These objects are part of the MATLAB Robust Control Toolbox. The Robust Control Toolbox models the uncertainty as a perturbation Δ wrapped in feedback around a nominal part M , i.e. uncertainty is represented using a linear fractional transformation. Real parametric, complex parametric, and unmodeled dynamic uncertainties can be modeled. The last row of Table 1 (“Uncertain Parameter Varying”) shows the corresponding parameter-varying objects with uncertainty: uncertain parameter-varying matrices, state space systems, and frequency responses are represented by `pumat`, `puss`, and `pufrd` objects, respectively. The parameter-varying uncertain objects are not yet implemented but they are essential to developing robustness analysis and design tools for LPV systems. There is also a natural overlap with the linear fractional representation for LPV systems.

The LPV objects are being developed within MATLAB’s object-oriented programming framework. A benefit of object-oriented programming is that key operations can be overloaded to provide seamless, consistent functionality across a variety of objects. For example, if `A` and `B` are `double` objects then the syntax `A*B` simply

multiplies the matrices. If A and B are `pmat` objects then the syntax $A*B$ multiplies the parameter-varying matrices at each point in the parameter domain. The manipulation of parameter-varying objects is facilitated by this extension of the $*$ operation to a meaningful, intuitive operation for `pmat` objects. In addition, standard MATLAB syntaxes, e.g. $M(i, j)$ to index into the (i, j) element of an array, have been overloaded and extended for parameter-varying objects. Object-oriented programming enables this overloading of key functions and this enables meaningful, intuitive extensions for parameter-varying objects.

To summarize, development of object-oriented LPV software tools will help expand the functionality and tools for LTI systems, as developed in the MATLAB Control and Robust Control Toolboxes, to linear parameter-varying systems. SIMULINK blocks have also been developed to simulate LPV systems. There are several challenges and open issues going forward. These include the numerical stability and scalability of the algorithms, incorporation of uncertainty analysis, and development of the data infrastructure and tools to encompass the various LPV representations. The remainder of the chapter will demonstrate the application of LPV analysis and control design techniques to the AAW control design example.

4 AAW Roll Rate Model

The AAW rigid body roll rate dynamics are given by:

$$\dot{p} = L_p(h, M)p + L_\delta(h, M)\delta \quad (2)$$

where p is roll rate (deg/sec) and δ is the outer leading edge flap position (deg). The outer leading edge flap (OLEF) is effective across the supersonic flight envelope of interest. Thus only this surface is used for roll rate control design in this chapter. The rigid body LPV model from outer leading edge flap to roll rate (Equation 2) is denoted G_{rig} .

L_p and L_δ are defined on a grid of altitude h (ft) and Mach M (unitless) with values provided in Tables 2 and 3. This data was constructed from non-dimensional aerodynamic coefficients obtained from NASA Dryden [7]. The non-dimensional L_δ data was re-scaled to obtain a mean gain of 2 over the flight envelope. Hence the variations of the L_δ data in Table 3 across the flight envelope accurately represent the AAW OLEF gain but the absolute magnitude contains a scaling factor. This scaling will be discussed in the following paragraphs.

The AAW wings were modified for increased flexibility leading to flexible modes occurring at lower frequencies. Models of the AAW aeroelastic dynamics were obtained from NASA Dryden [7] on a grid of altitude, Mach, and remaining fuel. The dependence on remaining fuel is neglected and the models at 60% fuel are used for the design and analysis. The aeroelastic dynamics are defined on the same (h, M) grid used to define the rigid body dynamics, i.e. the aeroelastic dynamics are defined on the grid $h = \{10000, 15000, 20000, 25000\}$ ft and $M = \{1.1, 1.2, 1.3\}$. At

Table 2: AAW Rigid Body Data, L_p

	M=1.1	M=1.2	M=1.3
h=10k	-0.5652	-0.4614	-0.4009
h=15k	-0.5415	-0.4363	-0.3737
h=20k	-0.5165	-0.4128	-0.3606
h=25k	-0.5034	-0.3982	-0.3531

Table 3: AAW Rigid Body Data, L_δ

	M=1.1	M=1.2	M=1.3
h=10k	1.2916	1.3756	1.2425
h=15k	0.9305	1.0524	1.1958
h=20k	0.6032	0.7009	0.8326
h=25k	0.3056	0.4110	0.5258

each flight condition the aeroelastic dynamics are modeled as a state-space system with 164 states. This model includes 64 states for the first 32 flexible modes and another 100 states for aerodynamic lags. The aerodynamic lag states can be truncated at each point in flight envelope with minimal impact on the OLEF to roll rate dynamics. The resulting aeroelastic model, denoted G_{flex} , has 64 states at each (h, M) flight condition.

The aeroelastic dynamics are added in parallel to the rigid body dynamics to obtain the full 65 state model, $G_{full} = G_{rig} + G_{flex}$. Figure 1 shows the open-loop Bode plots of G_{full} from OLEF to roll rate at each point in the (h, M) domain. As noted above, the L_δ data was rescaled to give a mean gain of 2 across the flight envelope. This effectively increases the significance of the flexible modes relative to the rigid body dynamics. In particular, several of the flexible modes have magnitude exceeding 15dB at points in the flight envelope. The first cluster of flexible modes occur around 55-65 rad/sec. The OLEF actuator has a bandwidth of 75 rad/sec. This bandwidth is not fast enough to actively suppress these flexible modes. The original intent was to use LPV techniques to actively control the AAW flexible modes. The AAW aircraft does not require active attenuation of the flexible modes nor is the OLEF actuator sufficiently fast to suppress these modes. Thus any control law must roll-off to avoid exciting these modes. Since the OLEF actuator dynamics are substantially faster than the AAW roll subsidence mode (L_p) these dynamics will be neglected in most of the design and analysis in the subsequent sections.

Figure 2 shows the open-loop roll rate responses due to an OLEF step of magnitude $\delta = 1$ deg. The top plot shows the response over an 8 second time scale and the bottom plot zooms in on the transient response over the first second. In both plots the solid curves are the rigid body responses at each point in the (h, M) domain while the dashed curves are the responses of the full model with the flexible modes. The rigid body responses in the top plot show the variation in DC gain and time constant across the flight envelope. Based on the data in Table 2, the time constant varies from a minimum of 1.77 sec to a maximum of 2.83 sec. The responses of G_{full}

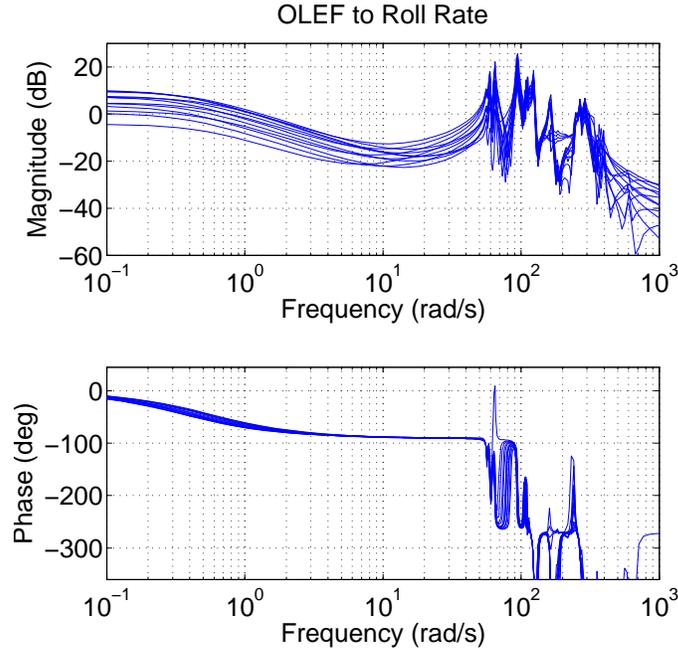


Fig. 1: Open loop Bode plots from OLEF to roll rate

show the transient effects of the flexible modes which are excited by the step input. The bottom plot more clearly shows this transient response. The flexible modes create a roll rate oscillation with an amplitude of approximately 0.6 deg/sec in the first 0.5 to 1.0 seconds of the step responses. These lightly-damped oscillations decay to a negligible amplitude after 2-3 seconds.

As described in Section 3, the core infrastructure for LPV modeling was developed using MATLAB object-oriented class programming. The overloading of key functions enables easy manipulation and analysis of these gridded models.

5 Gain Scheduled Classical Control

Gain scheduling via interpolation of point designs is still the predominant method used in industry to develop a full-envelope flight control law. LPV analysis can play an important role in certifying the performance of these control laws. Moreover, LPV analysis tools can uncover potential stability and performance degradations caused by rapid variations in the operating condition. This is especially important for systems with significant aeroelastic effects because flexible modes may be ex-

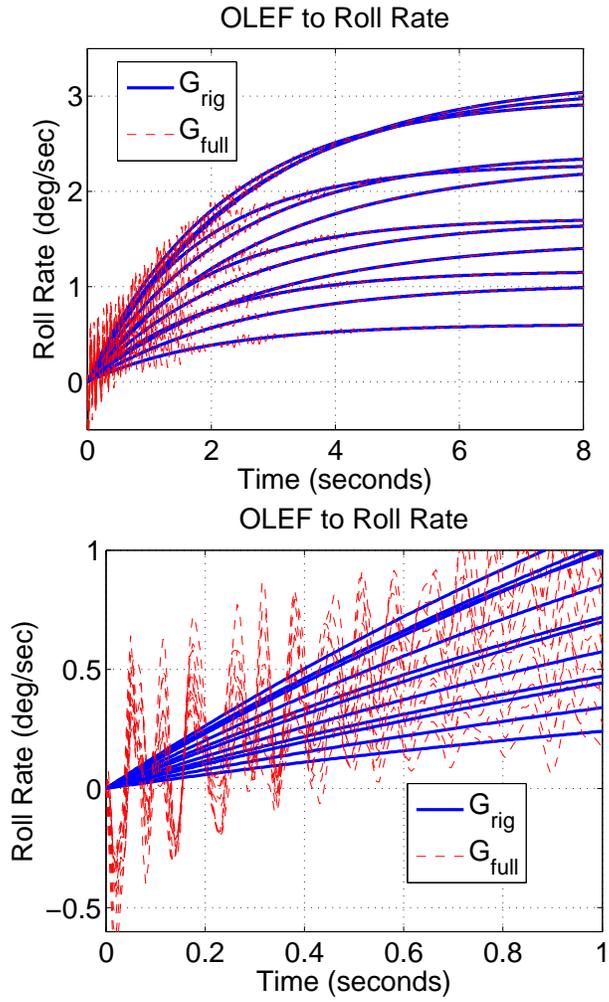


Fig. 2: Open loop step responses from OLEF to roll rate (Top: Long time scale, Bottom: Zoom on transient response)

cited during aircraft maneuvers. This section analyzes the performance of a gain scheduled classical roll rate control design using LPV techniques.

5.1 Control Design

The primary flight control objective is to design a feedback law to track roll rate commands. The variation in the speed of response and gain of the AAW rigid body dynamics across the (h, M) flight envelope (Tables 2 and 3) poses one issue for the control design. The roll subsidence mode varies from -0.56 rad/sec to -0.35 rad/sec and the input gain varies from 0.31 to 1.29. A gain scheduled controller is designed to achieve a consistent bandwidth of 1.25 rad/sec and zero steady state error due to step roll rate commands across the envelope. Another design issue is that the controller must be robust to the flexible modes. The gain scheduled controller is designed so that, in closed-loop, the flexible modes have magnitude less than -20dB at each point in the flight envelope. This is to ensure the flexible wing modes are not excited by the flight control system. In addition the gain scheduled controller should achieve 6dB of gain margin and 45deg of phase margin at each point in the envelope. These are standard margin requirements for military aircraft. The margin specifications at each point in the envelope essentially assume a quasi-steady approximation for (h, M) . LPV analysis tools will be used to determine the impact of variations in (h, M) on the closed-loop performance.

A classical gain-scheduled controller is designed to achieve these objectives. The basic idea is to invert the rigid body dynamics and replace them with a desired loop shape. In other words, the controller is given by $K_{cl} = G_{inv}G_{ls}$ where G_{inv} inverts the plant dynamics and G_{ls} is the desired loop shape. At each flight condition the AAW rigid body roll-rate dynamics are given by $\frac{L_{\delta}}{s+L_p}$. The rigid body-dynamics are, in general, time-varying due to the dependence on (h, M) and hence the transfer function representation is not correct. However, this representation will be used to provide the basic insight into the control design. The controller inverts the rigid body dynamics up to a bandwidth ω_{ro} to prevent exciting the flexible modes, $G_{inv} = \frac{s+L_p}{L_{\delta}} \frac{\omega_{ro}}{s+\omega_{ro}}$. The roll-off is chosen to be $\omega_{ro} = 12.5$ rad/sec. This is fast enough to have minimal impact on the roll-rate response but slow enough to avoid excessive excitation of the flex modes at 55-65 rad/sec. The desired loop shape is $G_{ls} = \frac{\omega_d^2}{s^2+2\zeta\omega_d}$. The values used in the control design are $\zeta = 0.8$ and $\omega_d = 1.25$ rad/sec. This desired loop shape provides a second order step response with small overshoot, zero steady state error, and a rise time of approximately 2.2 seconds.

As noted above, the plant and controller are both, in general, time-varying systems and transfer function representations are not meaningful. State-space representations should be used instead. The gain scheduled classical controller $K_{cl} = G_{inv}G_{ls}$ is given by the state-space representations:

$$\left[\begin{array}{c|c} A_{ls} & B_{ls} \\ \hline C_{ls} & D_{ls} \end{array} \right] := \left[\begin{array}{cc|c} -2\zeta\omega_d & 1 & 0 \\ 0 & 0 & \omega_d \\ \hline \omega_d & 0 & 0 \end{array} \right] \quad (3)$$

$$\left[\begin{array}{c|c} A_{inv} & B_{inv} \\ \hline C_{inv} & D_{inv} \end{array} \right] := \left[\begin{array}{c|c} -\omega_{ro} & \omega_{ro} \\ \hline \frac{L_p(h,M)+\omega_{ro}}{L_{\delta}(h,M)} & \frac{\omega_{ro}}{L_{\delta}(h,M)} \end{array} \right] \quad (4)$$

The dependence on (h, M) has been made explicit for clarity. The controller state matrices are defined on the 4×3 (h, M) grid for which plant data is available (see Tables 2 and 3). The classical controller is gain-scheduled by linearly interpolating the state matrices for G_{ls} and G_{inv} .

A simple, but important observation is that time-varying systems, in general, do not commute. Thus the closed-loop performance will be impacted by reordering the controller as $G_{ls}G_{inv}$ or using an alternative state-space realization for G_{inv} . The realization of G_{inv} in Equation 4 isolates all time-variations in the output and feedthrough matrices. This realization enables to the controller to instantly cancel variations in the plant dynamics. A drawback is that this realization of the controller will be sensitive to errors in the AAW gain-scheduled model.

5.2 LTI Point Analysis

Insight into the control design and feedback system can be obtained by studying the LTI performance at each point in the flight envelope. Figure 3 shows the Bode plots for the classical controller K_{cl} at each point in the flight envelope. The controller has proportional-integral action at low frequencies with a second order roll-off beyond ω_{ro} to avoid exciting the flexible modes. These Bode plots show an intuitive classical design at each point in the flight domain.

Figure 4 shows the Bode plots for the (open) loop function $G_{full}K_{cl}$ at each point in the flight envelope. The loop function again shows integral action at low frequencies. In addition, the loop function $G_{full}K_{cl}$ shows significant attenuation of the flexible modes in comparison with the open loop Bode G_{full} (Figure 1). The loop $G_{full}K_{cl}$ has all modes below -19dB at all points in the flight envelope. In closed loop the flexible modes are still well attenuated. The closed-loop response from roll rate command to roll rate has the flexible modes below -18.5dB at all points in the domain. The loop function $G_{full}K_{cl}$ has gain and phase margins exceeding 21.3dB and 65.9 degs at each point in the flight domain. Thus the classical controller has good gain and phase margins at each point in the flight envelope.

Figure 5 shows the closed-loop unit step responses to a 1 deg roll rate command at all points in the flight envelope. The full model with flexible modes G_{full} and gain-scheduled classical controller K_{cl} are used to generate these closed-loop responses. The top plot shows the roll rate response and the bottom plot shows the OLEF position. The blue solid curve in the top plot shows the ideal closed-loop response given by the open loop specification G_{ls} . The red dashed curves in both plots show the closed loop responses at each point in the (h, M) domain. The classical controller achieves consistent dynamic performance across the flight envelope with zero steady state error (top plot). The bottom plot shows the variation in the control actuation required to achieve this uniform tracking performance. In addition, the closed loop step responses (top plot of Figure 5) indicate only small oscillations in the initial transient due to the flexible modes. This is due to the high frequency roll-off of K_{cl} which attenuates the flexible modes. Finally, the closed-loop responses (top plot)

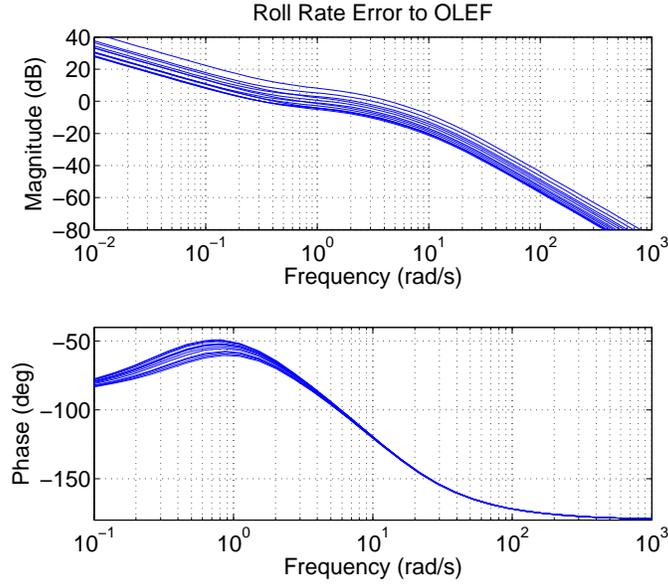


Fig. 3: Bode plots of classical controller

have a small deviation from the ideal response. This is due to the high-frequency roll-off in G_{inv} to avoid exciting the flexible modes.

The closed-loop performance can be evaluated using the induced L_2 norm of various closed-loop sensitivity functions in the feedback loop. Table 4a shows results computed for the feedback system consisting of the rigid body dynamics G_{rig} and the gain-scheduled classical controller K_{cl} . The rows labeled S_i , T_i , S_o , and T_o provide induced L_2 norm bounds for the input sensitivity, input complementary sensitivity, output sensitivity, and output complementary sensitivity. The second column, labeled Point H_∞ , is the maximum H_∞ norm over all points in the flight envelope. For the row labeled S_i , this was result was obtained by computing the H_∞ norm of the input sensitivity function at each point in the flight envelope and then maximizing over the flight envelope. The results in the other rows were obtained similarly. Each data point in this column took, on average, 0.21 seconds to compute on a laptop with a dual-core 2.16GHz Intel processor. At a fixed point in the flight envelope, the H_∞ norm for each sensitivity function is equal to its induced L_2 norm. This is a lower bound on the actual induced L_2 norm for the gain-scheduled system since it does not account for parameter variations. The results for the input and output functions are equal because the feedback loop is SISO and the point-wise H_∞ norm assumes time-invariant dynamics at each point in the domain. The remaining columns in Table 4a are discussed in the following subsection.

The pointwise H_∞ analysis can also be used to investigate the impact of the flexible modes at each point in the domain. The full model G_{full} with rigid body

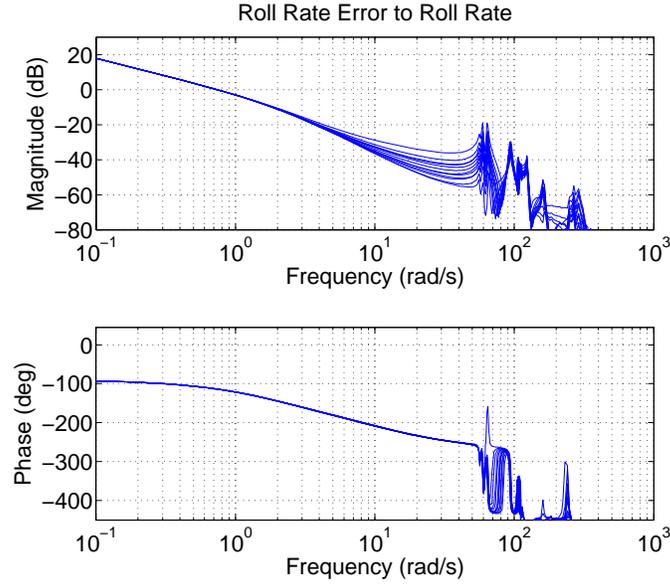


Fig. 4: Bode plots of loop function $G_{full}K_{cl}$

and aeroelastic dynamics has 65 states. For this analysis a reduced order model, G_{red} , that captures the first three flexible modes was constructed by residualizing the higher frequency flexible modes at all points in the flight envelope. G_{red} has a total of seven states: one state for the rigid body dynamics and six states for the first three flexible modes. The closed loop sensitivity functions are then formed with G_{red} and K_{cl} . Table 4b shows the various norms computed for the closed loop sensitivity functions with G_{red} and K_{cl} . The rows and columns of this table can be compared with the previous analysis for the closed loop with the rigid body dynamics (Table 4a). The maximum H_∞ norm over the flight envelope (column: Point H_∞) shows only minor differences with the results for the rigid body dynamics. Thus the H_∞ norms computed at each point in the flight domain indicate that the aeroelastic dynamics will have minimal impact on the gain scheduled performance.

5.3 LPV Analysis

The analysis of the gain-scheduled classical controller has, up to this point, focused on the performance at each point in the (h, M) flight envelope. This analysis neglects the impact of time variations in altitude and Mach. The induced L_2 norm for an LPV system is the maximum input/output gain over all inputs and a class of allowable parameter trajectories. A generalization of the Bounded Real condition leads to linear

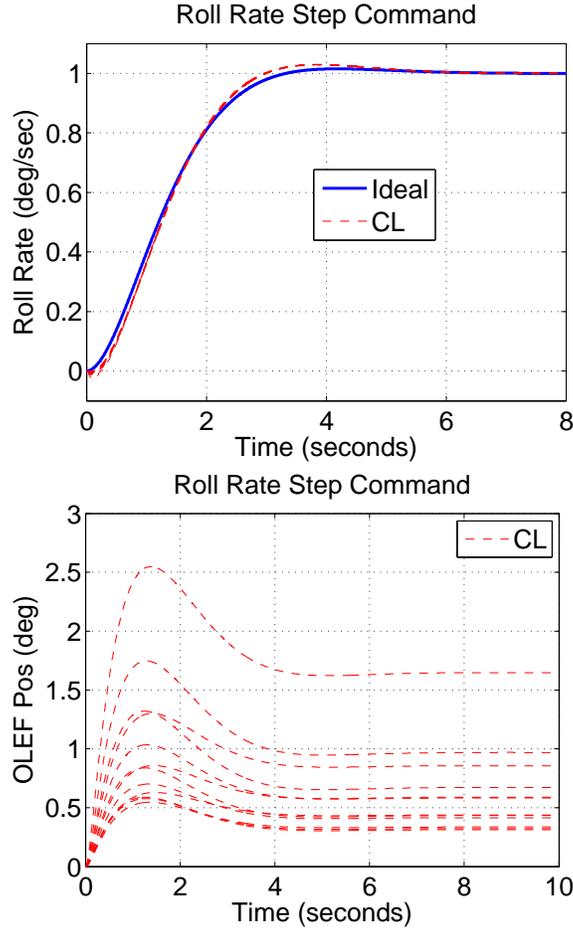


Fig. 5: Closed-loop responses to step roll rate command with K_{cl}

matrix inequality (LMI) conditions for computing bounds on the induced L_2 norm. A brief review is provided in Appendix 8.

Upper bounds on the induced L_2 norm of the various closed-loop sensitivity functions were computed to gain insight into the effect of variations in (h, \dot{M}) . The results for the closed loop with G_{rig} and K_{cl} are given in the columns LPV1, LPV2, and LPV3 of Table 4a. The results in these columns involve LPV induced L_2 norm upper bounds of various complexity. The column labeled LPV1 used constant Lyapunov matrices in the induced L_2 norm LMI conditions. This is equivalent to an analysis that assumes no knowledge of the parameter rates. The results in columns LPV2 and LPV3 both assume the rate bounds $|\dot{M}| \leq 0.02$ 1/sec and $|\dot{h}| \leq 1000$ ft/sec. LPV2 uses a parameter dependent Lyapunov function of the

Table 4: L_2 induced norms for closed-loop sensitivity functions with gain-scheduled classical controller

(a)					(b)						
Closed-loop with (G_{rig}, K_{cl})					Closed-loop with (G_{red}, K_{cl})						
	Point	H_∞	LPV1	LPV2	LPV3		Point	H_∞	LPV1	LPV2	LPV3
S_i		1.292	3.786	2.222	1.590	S_i		1.314	Inf	2.202	1.548
T_i		1.000	3.685	2.116	1.431	T_i		1.000	Inf	2.143	1.418
S_o		1.292	1.295	1.296	1.298	S_o		1.314	Inf	2.406	1.347
T_o		1.000	1.000	1.000	1.001	T_o		1.000	Inf	2.096	1.009

form $X(\rho) = X_0 + MX_1 + hX_2$ while LPV3 also includes quadratic bases functions, $X(\rho) = X_0 + MX_1 + hX_2 + M^2X_3 + MhX_4 + h^2X_5$. In theory the upper bounds on the induced L_2 norm should progressively decrease from analysis LPV1 to LPV2 to LPV3. The results for S_o and T_o in Table 4a do not demonstrate this theoretical trend but the results are within the stopping tolerances of the optimization solver.

There are several interesting aspects to the LPV analysis results in Table 4a. First, the output sensitivity functions, S_o and T_o , have an induced norm almost exactly equal to the pointwise H_∞ lower bound. This is because the classical controller perfectly cancels the rigid body dynamics for reference commands that enter at the plant output. However, the cancellation is not perfect for disturbances that enter at the plant input. Thus there is a gap between the upper bounds for S_i and T_i and the pointwise H_∞ lower bounds. The gap is reduced by including additional bases functions in the LPV analysis. The price paid for this improved upper bound is an increased computational complexity. Each LPV1, LPV2, and LPV3 analysis took 0.75, 10.1, and 37.6 seconds, on average. These results indicate that the variations in (h, M) are unlikely to affect the reference tracking but it may have some effect on disturbance rejection at the plant input.

It should also be re-emphasized that the performance depends on the state realization for G_{inv} . An alternative realization for G_{inv} is given by $B_{inv} = \frac{\omega_{ro}}{L_\delta(h, M)}$ and $C_{inv} = -(L_p(h, M) + \omega_{ro})$ with the same A_{inv} and D_{inv} given in Equation 4. The alternative realization is identical to the original realization for fixed flight conditions and hence it yields the same pointwise H_∞ norms as in Table 4a. However, the LPV3 analysis results yield norms of 3.433, 3.416, 1.365, and 1.107 for S_i , T_i , S_o , and T_o , respectively. These are significantly larger than the LPV3 results in Table 4a for the original state-space realization of G_{inv} in Equation 4. It should be noted that the alternative state-space realization for G_{inv} has parameter dependence in the input matrix. This parameter dependence must pass through the G_{inv} dynamics before it is able to cancel the variations in the plant dynamics. Thus one might have anticipated that this alternative realization leads to degraded performance when compared to the original realization in Equation 4. This analysis demonstrates that the LPV induced norm bounds can be used to aid the design engineer in selecting the best realization for gain scheduling.

The LPV analysis can also be used to investigate the impact of the flexible modes. As described in the previous subsection, a reduced order, 7-state aeroelastic model was obtained by retaining the first three flexible modes in the full model. This model reduction was mainly motivated by the computational complexity of the LPV analysis condition with respect to the plant state dimension. The 65-state model G_{full} is too large to be handled by current optimization solvers but the 7-state reduced order model can be analyzed in a reasonable amount of time. Columns LPV1, LPV2, and LPV3 in Table 4b shows the LPV upper bounds computed for the closed loop sensitivity functions with G_{red} and K_{cl} . The rows and columns of this table can be compared with the previous analysis for the closed loop with the rigid body dynamics (Table 4a). The LPV upper bounds computed with using linear and quadratic basis functions (column LPV3 in Table 4b) are roughly the same as the results obtained with the rigid body dynamics (column LPV3 in Table 4a). This indicates that the aeroelastic dynamics will have minor impact the gain-scheduled closed-loop performance. This agrees with the results obtained using pointwise H_∞ norms. The results labeled *Inf* in column LPV1 indicate that no provable upper bound on the induced L_2 norm can be obtained using constant, quadratic Lyapunov functions.

The LPV analysis (Tables 4a and 4b) has gaps between the pointwise H_∞ lower bounds and the LPV3 analysis upper bound. The user could continue adding bases functions and see if the LPV3 analysis bound can be reduced. However, there is a computational penalty to be paid for adding bases functions. Each Point H_∞ , LPV1, LPV2, and LPV3 analysis in Table 4b took 0.08, 0.58, 91.6 and 360.6 seconds, on average. An alternative is to compute improved lower bounds via simulation. This is discussed in the next subsection.

5.4 Worst-case Simulation

A SIMULINK block has been developed to enable easy simulation of LPV systems. This block is similar to the SIMULINK state space block but with an additional input to specify the parameter trajectory. The function `wcsim` has been developed to perform a worst-case LPV analysis directly on a SIMULINK model. “Worst-case” refers to maximizing or minimizing a user-specified cost over a set of parameter trajectories subject to user-specified constraint functions.

`wcsim` works directly on a SIMULINK model that contains feedback interconnections of LTI and/or LPV systems. The user specifies ρ to be a linear combination of piecewise continuous functions, i.e. $\rho_c(t) \in \mathcal{P} := \{\rho(t) = \sum_{i=1}^{n_b} c_i \rho_i(t)\}$ where $\{\rho_i\}$ are the user-specified basis functions. Let $y_c : [0, t_f] \rightarrow \mathbb{R}^{n_y}$ denote the vector of output signals from the SIMULINK diagram for a given choice of the parameter vector coefficients, $c \in \mathbb{R}^{n_b}$. The worst-case simulation problem is:

$$\begin{aligned} & \max_{c \in \mathbb{R}^{n_b}} G(y_c) \\ & \text{subject to: } \underline{C} \leq C(y_c) \leq \bar{C} \end{aligned} \tag{5}$$

$G : L_2^{n_y}[0, t_f] \rightarrow \mathbb{R}$ denotes the objective functional that quantifies the performance of the output by $G(y_c)$. $C : L_2^{n_y}[0, t_f] \rightarrow \mathbb{R}^m$ denotes a constraint function that defines a set of m constraints on the output. \underline{C} and $\bar{C} \in \mathbb{R}^m$ specify the lower and upper bounds on the constraint function. Maximization is without loss of generality. The objective and constraint functions for a worst-case simulation are specified with specialized SIMULINK Objective Function and Constraint Function blocks. These blocks are similar to a standard To Workspace block except their dialog boxes contain additional fields to specify the objective and constraint functions.

The `wcsim` function optimizes the objective function over the set of allowable parameter trajectories subject to the specified constraints. The basic syntax is `wcrho = wcsim('mdl')`. The input specifies the SIMULINK diagram with LPV blocks, Objective Function and (optionally) Constraint Function blocks. `wcsim` returns the worst-case parameter trajectory. No assumptions are made about the objective function G or constraint function C . As a result, the optimization is, in general not convex and it may have many local optima that are not global optima. `wcsim` simply uses gradient-based optimization to find a parameter trajectory that achieves a local maxima. While this does not necessarily find the global optima it does provide a means to improve upon 'bad' parameter trajectories found with other heuristic methods.

The gradient-based optimization is performed by `fmincon` and thus requires the MATLAB Optimization toolbox. At each iteration `fmincon` requires multiple evaluations of the objective and constraint functions in order to compute numerical gradients. An evaluation of all objective and constraint functions specified in a SIMULINK diagram requires one simulation of the model. At each iteration `fmincon` evaluates the objective function at the current value $c \in \mathbb{R}^{n_b}$ of the parameter trajectory coefficients well as at small perturbations along each coefficient direction. If the parameter trajectory contains n_b coefficients, `fmincon` will perform approximately $n_b + 1$ simulations at each iteration. Simulating the system is typically responsible for the bulk of the computation time to perform a worst-case simulation. Thus the total time for `wcsim` with no constraint blocks will be roughly $(n_c + 1)N_i\tau$ where τ is the computation time for one simulation and N_i is the number of iterations. If the model contains constraint blocks then `fmincon` will typically perform additional function evaluations, and hence additional simulations, per iteration.

Figure 6 shows the SIMULINK diagram used to investigate the AAW closed-loop roll-rate tracking performance. The diagram contains the controller K_{cl} in the LPV block labeled "Gain-Scheduled Controller". The LPV block labeled "LPV AAW" contains the full AAW model, G_{full} . In addition, the OLEF actuator dynamics have been included in the feedback loop. The diagram contains inputs for roll-rate command and plant step input disturbance. The LPV analysis in the previous section indicated that the input sensitivity functions were more likely to be affected by parameter variations. This section focuses on the OLEF actuator position response due to the step input disturbance. The roll rate command is set to zero.

The model was simulated at each point in the flight domain to investigate the closed-loop performance. The L_2 norm of the OLEF actuator position response to the input step disturbance varied from 2.720 to 2.723 across the flight domain. This indicates uniform disturbance rejection performance at each point in the flight envelope. The OLEF response due to the input step disturbance is shown as the solid line (‘Nominal’) in the top subplot of Figure 7. This simulation was performed at the center of the flight envelope, i.e. $(h, M) = (17500 \text{ ft}, 1.2)$. The response is well damped and shows no oscillations due to the flexible modes. In addition, the OLEF position cancels the disturbance in steady state. The average time to complete one simulation is 25.3 sec.

Next `wcsim` is used to perform a search for time-varying trajectories that degrade the disturbance rejection performance. For this example, `wcsim` maximizes the L_2 norm of the OLEF position by searching over parameter trajectories of the form $c_0 + c_1 t + c_2 \cos(t) + c_3 \sin(t)$ where c_i are constants to be optimized. Both the Mach and Altitude parameter trajectories are restricted to have this form and thus there are a total of eight coefficients to be optimized. The objective function is specified using the block labeled “Objective” in Figure 7. For this example, the `wcsim` objective is maximize the L_2 norm of the OLEF position response due to the input step disturbance.

Initially no constraints are placed on the parameter trajectory and parameter rates. The trajectories are projected to lie within the bounds of the (h, M) parameter domain at all times using a saturation block. After $N_i = 4$ gradient steps, `wcsim` finds a parameter trajectory that achieves an OLEF position L_2 norm of 3.096. The worst-case OLEF position response due to the step input disturbance is shown as the dashed curve in the top subplot of Figure 7. The worst-case Mach trajectory computed by `wcsim` is shown as the dashed curve in the bottom subplot of Figure 7. The worst-case altitude trajectory is not shown since it is a constant 17500 ft to within 0.25 ft. These results indicate that variations in altitude have a small impact on the closed-loop disturbance rejection. The expected computation for this `wcsim` is $(n_c + 1)N_i\tau \approx 910$ sec based on $n_c = 8$ coefficients, $N_i = 4$ iterations, and $\tau = 25.3$ seconds per simulation. The actual `wcsim` optimization ran 60 simulations and took 1735 sec to complete. The worst-case Mach trajectory (bottom subplot of Figure 7) shows fast variations with a peak rate of $|\dot{M}| \approx 0.156$ 1/sec. These are unrealistic Mach rates for the actual AAW aircraft.

Another `wcsim` optimization is run with the constraint $|\dot{M}| \leq 0.02$ 1/sec. This constraint is enforced in the “Parameter Rate Constraint” subsystem in Figure 6 using an approximate derivative and a `wcsim Constraint Function` block. The altitude was held constant at 17500 ft for this optimization since altitude variations appear to have minor impact on the OLEF position response. Thus `wcsim` only optimizes over the four coefficients in the Mach trajectory. After $N_i = 6$ gradient steps, `wcsim` finds a parameter trajectory that achieves an OLEF position L_2 norm of 2.948. The worst-case OLEF position response due to the step input disturbance is shown as the dash-dotted curve in the top subplot of Figure 7. The worst-case Mach trajectory computed by `wcsim` is shown as the dash-dotted curve in the bottom subplot of Figure 7. The worst case trajectory returned by `wcsim`

satisfies $|\dot{M}| \leq 0.202$ 1/sec. This slight violation of the enforced constraint is due to the restriction of $N_i = 6$. The Mach trajectory computed by `wcsim` simply slows the AAW down from its upper Mach limit to its lower limit over the ten second simulation. The expected computation for this `wcsim` is $(n_c + 1)N_i\tau \approx 759$ sec based on $n_c = 4$ coefficients, $N_i = 6$ iterations, and $\tau = 25.3$ seconds per simulation. The actual `wcsim` optimization ran 38 simulations and took 1154 sec to complete.

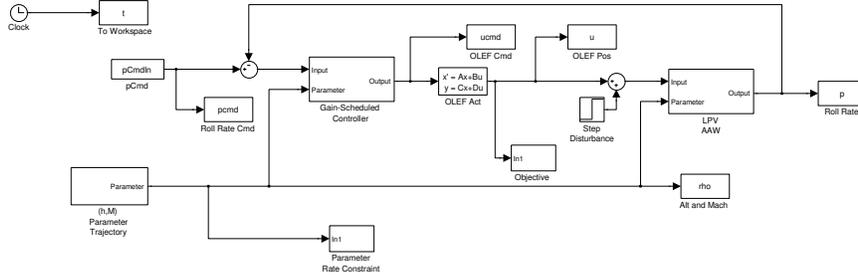


Fig. 6: AAW LPV SIMULINK diagram

6 LPV Control Design

This section describes an LPV control design that provides guaranteed performance with respect to time variations in the (h, M) parameters. The design approach is based on signal-weighted induced L_2 norms. Figure 8 shows the design interconnection used for the control synthesis. The performance objective is to minimize the induced L_2 norm from the design interconnection inputs to the outputs. The various I/O signals are weighted to obtain the desired trade-offs between reference tracking, disturbance/noise rejection, and actuation usage.

The rigid body AAW dynamic models G_{rig} are used in the design. The design interconnection formulates a model-matching problem. The key performance objective is for the closed-loop response from p_{cmd} to p to match the ideal response given by $G_{ideal} = \frac{\omega_d^2}{s^2 + 2\zeta\omega_d s + \omega_d^2}$. The ideal response natural frequency and damping are given by $\zeta = 0.8$ and $\omega_d = 1.25$ rad/sec. This is the same ideal response given by the loop shape G_{I_s} in the gain-scheduled classical design. The actuation and performance penalties are given by $W_a = \frac{100s+25}{s+2500}$ and $W_p = \frac{0.01s+12.5}{s+0.125}$. This emphasizes tracking of the ideal response up to ≈ 11.25 rad/sec and penalizes control usage at higher frequencies. The input disturbance and noise weights are chosen as $W_d = 0.1$ and $W_n = 0.01$. The small values are chosen to emphasize the actuation / performance trade-off. The ideal model and all weights are independent of the flight condition. Thus the performance objective aims to achieve similar perfor-

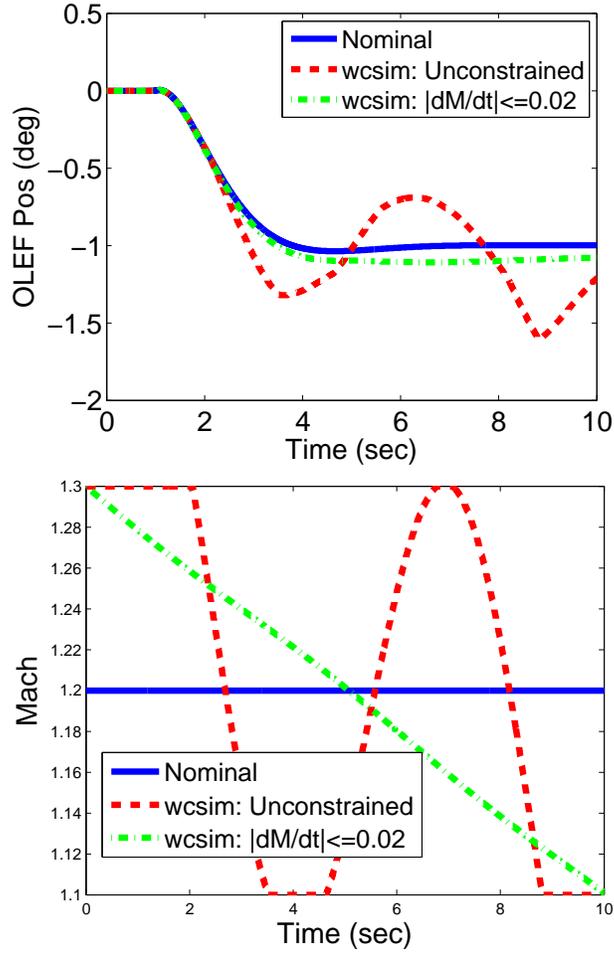


Fig. 7: Simulation results due to step input disturbance

mance across the flight envelope. The standard Robust Control Toolbox command `sysic` has been overloaded to allow interconnections of `pss` objects and standard MATLAB system objects. This overloaded `sysic` command was used to generate the LPV design interconnection shown in Figure 8.

To understand the limits of performance, the H_∞ optimal control problem specified by this design interconnection was solved at each point in the flight envelope. The optimal performance varied from a minimum of $\gamma = 1.008$ at $(h, M) = (25000ft, 1.1)$ to a maximum of $\gamma = 3.726$ at $(h, M) = (15000ft, 1.3)$. The optimal performance of any LPV design must be greater than or equal to the optimal H_∞ performance achieved at any point in the domain. Hence $\gamma = 3.726$ is

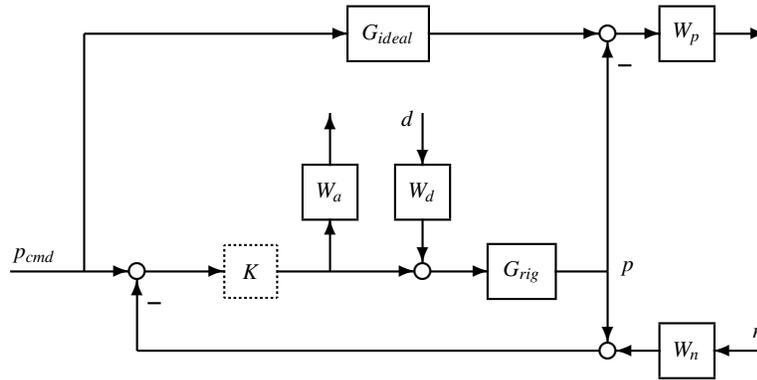


Fig. 8: Design interconnection AAW roll rate control

a lower bound on the achievable performance by the optimal LPV controller. For comparison, the gain-scheduled classical controller, K_{cl} , achieves a minimum gain of $\gamma = 1.000$ at $(h, M) = (25000ft, 1.1)$ and a maximum gain of $\gamma = 3.857$ at $(h, M) = (15000ft, 1.3)$.

Next, an LPV controller was synthesized without using knowledge about the rate variations of altitude and Mach. This control synthesis is performed using the function `lpvsyn` which solves a set of parameterized LMIs using a constant (non-parameter varying) Lyapunov function. The parameterized LMI conditions are based on results in [33]. The non-rate bounded design achieves an optimal gain of $\gamma_{lpv,nr} = 3.844$. This is very close to the performance lower bound computed based on the point-wise H_∞ designs. The design interconnection has a total of five states: one for the rigid body AAW dynamics, two for the ideal response model and one each for the actuation and performance weights. The non-rate bounded controller K_{lpv} has the same number of states as the design interconnection, i.e. K_{lpv} has five states. A rate-bounded LPV controller was also synthesized assuming $|\dot{M}| \leq 0.02$ and $|\dot{h}| \leq 1000$ ft/sec and using basis functions $\{1, M, h\}$. The rate bounded design achieved a gain of $\gamma_{lpv,rb} = 3.797$. This is a small improvement over the non-rate bounded design. Hence the bound on the rate of the parameter variation does not play a significant role in the AAW design. The remainder of the section will focus on the non-rate bounded control design.

Figure 9 shows the Bode plots for the the non-rate bounded LPV controller K_{lpv} at each point in the flight envelope. The controller has proportional-integral action at low frequencies and a second order roll-off beyond ω_{r0} to avoid exciting the flexible modes. Both these characteristics are similar to gain-scheduled classical design shown in Figure 3. One difference is that K_{lpv} has additional phase lead between 1 to 10 rad/sec. In addition, K_{lpv} has the same high frequency gain at all points in the flight envelope while the high frequency gain of the classical design K_{cl} varies with the flight condition. The point-wise Bode plots of K_{lpv} show an intuitive

classical design at each point in the flight domain and this provides confidence in the LPV design tools. One benefit of the LPV design tools, even for single-input single-output gain scheduling, is that the designer does not have to worry about the impact of the differing state space realizations. The impact of the state realization on the gain-scheduled performance is built into the design.

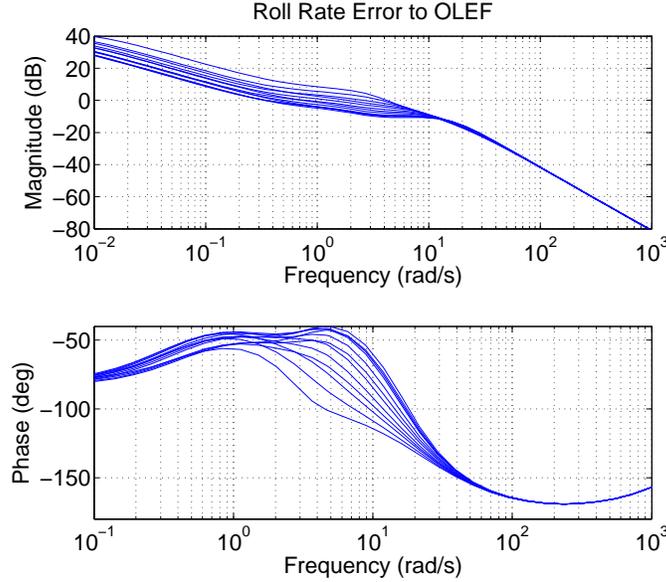


Fig. 9: Bode magnitude plots of LPV controller

Figure 10 shows the Bode plots for the (open) loop function $G_{full}K_{nr}$ at each point in the flight envelope. The loop function again shows integral action at low frequencies. The loop $G_{full}K_{nr}$ has all modes below -12.0dB at all points in the flight envelope. In closed-loop, the response from roll rate command to roll rate has the flexible modes below -12.4dB at all points in the domain. This is slightly less attenuation than achieved by the gain-scheduled classical controller. The additional phase lead in K_{nr} is evident from 1 to 10 rad/sec in the Bode plot of $G_{full}K_{nr}$. The loop function $G_{full}K_{cl}$ has gain and phase margins exceeding 18.8dB and 66.7 degs at each point in the flight domain. This a slightly smaller gain margin than K_{cl} but the phase margins of K_{cl} and K_{nr} are essentially the same.

Figure 11 shows the closed-loop unit step responses to a 1 deg roll rate command at all points in the flight envelope. The full model with flexible modes G_{full} and LPV controller K_{nr} are used to generate these closed-loop responses. The top plot shows the roll rate response and the bottom plot shows the OLEF position. The blue solid curve in the top plot shows the ideal closed-loop response specified by the ideal model G_{ideal} . The red dashed curves in both plots show the closed loop responses at

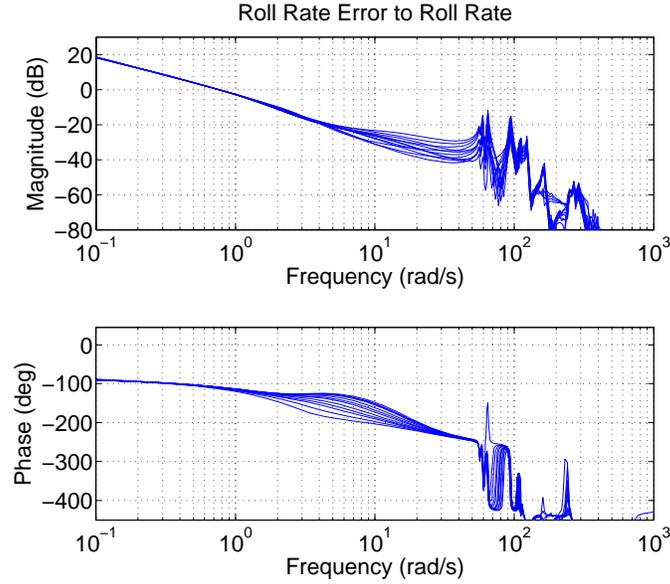


Fig. 10: Bode plots of loop function $G_{full}K_{nr}$

each point in the (h, M) domain. The LPV controller achieves consistent dynamic performance across the flight envelope with good attenuation of the flexible modes. The LPV controller provides slightly better tracking of the ideal response when compared to the classical design (Figure 5). Both K_{nr} and K_{cl} have similar actuator usage for the step roll rate command.

Tables 5a and 5b show the bounds on the LPV induced L_2 norms of the various closed-loop sensitivity functions. Table 5a was computed using G_{rig} and K_{lpv} while the results in Table 5b were computed with the reduced aeroelastic model G_{red} and K_{lpv} . These results can be compared with the gain scheduled classical controller performance K_{cl} in Tables 4a and 4b. The LPV controller has slightly achieves slightly better performance based on the rate-bounded upper bounds computed using the linear and quadratic basis functions (column LPV3).

7 Conclusions

This chapter investigated the use of LPV techniques for the roll control of NASA Dryden's X-53 Active Aeroelastic Wing (AAW) testbed. LPV analysis of a gain-scheduled classical controller indicated that variations in scheduling parameter would have minimal impact on reference tracking but may have some impact on disturbance rejection at the plant input. An LPV controller was also synthesized us-

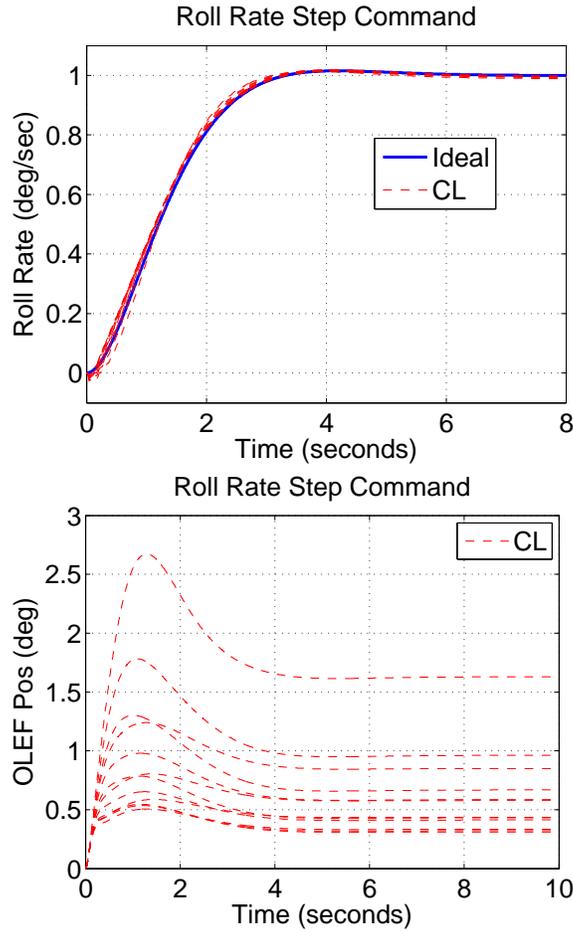


Fig. 11: Closed-loop responses to step roll rate command with K_{nr}

ing a model matching design. The LPV controller has an intuitive classical control characteristics and its performance was similar to the gain-scheduled classical design. All results were obtained using new LPV modeling, analysis and design software tools. Future work will consider the scalability of the numerical algorithms, incorporation of uncertainty analysis, and development of the data infrastructure and tools to encompass the various LPV representations.

Table 5: L_2 induced norms for closed-loop sensitivity functions with LPV controller

(a)					(b)						
Closed-loop with (G_{rig}, K_{lpv})					Closed-loop with (G_{red}, K_{lpv})						
	Point	H_∞	LPV1	LPV2	LPV3		Point	H_∞	LPV1	LPV2	LPV3
S_i		1.302	3.682	2.063	1.434	S_i		1.328	Inf	2.087	1.450
T_i		0.995	3.586	2.004	1.358	T_i		0.995	Inf	2.036	1.360
S_o		1.302	1.315	1.309	1.306	S_o		1.328	Inf	1.356	1.357
T_o		0.995	1.002	0.998	0.998	T_o		0.995	Inf	1.007	0.999

8 Acknowledgments

This research was supported under the NASA Dryden SBIR NNX11CD58P entitled “Adaptive Linear Parameter-Varying Control for Aeroelastic Suppression”. The technical contract monitor is Dr. Martin J. Brenner.

References

1. Apkarian, P., P.Gahinet: A convex characterization of gain-scheduled H_∞ controllers. IEEE Transactions on Automatic Control **40**(5), 853–864 (1995)
2. Balas, G., Bokor, J., Szabo, Z.: Invariant subspaces for LPV systems and their applications. IEEE Transactions on Automatic Control **48**(11), 2065–2069 (2003)
3. Baranyi, P., Yam, Y., Varlaki, P.: Tensor Product Model Transformation in Polytopic Model-Based Control. Taylor & Francis (2011)
4. Becker, G.: Quadratic stability and performance of linear parameter dependent systems. Ph.D. thesis, University of California, Berkeley (1993)
5. Boehm, B., Flick, P., Sanders, B., Pettit, C., Reichenbach, E., Zillmer, S.: Static aeroelastic response predictions of the active aeroelastic wing (AAW) flight research vehicle. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, AIAA 2001-1372 (2001)
6. Brenner, M.: Aeroservoelastic model uncertainty bound estimation from flight data. AIAA Journal of Guidance, Control, and Dynamics **25**(4), 748–754 (2002)
7. Brenner, M.: (2011). Personal Communication
8. Brenner, M., Lind, R.: Wavelet-processed flight data for robust aeroservoelastic stability margins. AIAA Journal of Guidance, Control, and Dynamics **21**(6), 823–829 (1998)
9. Bruzelius, F., Breitholtz, C.: Gain scheduling via affine linear parameter-varying systems and h_∞ synthesis. In: IEEE Conference on Decision and Control, pp. 713–718 (2001)
10. Clarke, R., Allen, M., Dibley, R., Gera, J., Hodgkinson, J.: Flight test of the F/A-18 active aeroelastic wing airplane. Tech. Rep. TM-2005-213664, NASA (2005)
11. Dibley, R., Allen, M., Clarke, R., Gera, J., Hodgkinson, J.: Development and testing of control laws for the active aeroelastic wing program. Tech. Rep. TM-2005-213666, NASA (2005)
12. Gahinet, P., Nemirovski, A., Laub, A., Chilali, M.: LMI control toolbox user’s guide. Tech. rep., The Mathworks (1995)
13. Lind, R., Brenner, M.: Analyzing aeroservoelastic stability margins using the μ method. In: AIAA Structures, Structural Dynamics, and Materials Conference, AIAA 98-1895, pp. 1672–1681 (1998)

14. Lucia, D.: The sensorcraft configurations: A non-linear aeroservoelastic challenge for aviation. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit, AIAA-2005-1943 (2005)
15. Mangalam, A., Mangalam, S., Flick, P.: Unsteady aerodynamic observable for gust load alleviation and flutter suppression. In: AIAA Atmospheric Flight Mechanics Conference, AIAA 2008-7187 (2008)
16. Mangalam, A., Moes, T.: Real-time unsteady loads measurements using hot-film sensors. In: AIAA Applied Aerodynamics Conference, AIAA 2004-5371 (2004)
17. Mangalam, S., Flick, P., Brenner, M.: Higher level aerodynamics input for aeroservoelastic control of flexible aircraft. In: AIAA Atmospheric Flight Mechanics Conference, AIAA 2007-6380 (2007)
18. Martinez, J., Flick, P., Perdsock, J., Dale, G., Davis, M.: An overview of sensorcraft capabilities and key enabling technologies. In: AIAA Applied Aerodynamics Conference, AIAA-2008-7185 (2008)
19. Packard, A.: Gain scheduling via linear fractional transformations. *Systems and Control Letters* **22**(2), 79–92 (1994)
20. Packard, A., Kantner, M.: Gain scheduling the LPV way. In: IEEE Conference on Decision and Control, pp. 3938–3941 (1996)
21. Pendleton, E., Bessette, D., Field, P., Miller, G., Griffin, K.: Active aeroelastic wing flight research program: Technical program and model analytical development. *Journal of Aircraft* **37**(4), 554–561 (2000)
22. Pendleton, E., Flick, P., Voracek, D., Reichenbach, E., Griffin, K., Paul, D.: The X-53: A summary of the active aeroelastic wing flight research program. In: AIAA Structures, Structural Dynamics, and Materials Conference (2007). Paper 2007-1855
23. Pendleton, E., Griffin, K., Kehoe, M., Perry, B.: A flight research program for active aeroelastic wing technology. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit (1996). Paper 96-1574-CP
24. Scherer, C.: LPV control and full block multipliers. *Automatica* **37**(3), 361–375 (2001)
25. Scherer, C., Weiland, S.: Linear Matrix Inequalities in Control. DISC Lecture Notes (2000)
26. Seiler, P., Balas, G., Packard, A.: Linear parameter varying control for the X-53 active aeroelastic wing. In: AIAA Guidance, Navigation, and Control Conference (2011)
27. Shamma, J., Athans, M.: Gain scheduling: potential hazards and possible remedies. In: American Control Conference, pp. 516–521 (1991)
28. Sharuz, S., Behtash, S.: Design of controllers for linear parameter-varying systems by the gain-scheduling technique. In: IEEE Conference on Decision and Control, pp. 2490–2491 (1990)
29. Sun, X., Postlethwaite, I.: Affine LPV modeling and its use in gain-scheduled helicopter control. In: UKACC International Conference on Control, pp. 1504–1509 (1998)
30. Tischler, M., Colbourne, J., Morel, M., Biezd, D., Levine, W., Moldoveanu, V.: CONDUIT: A new multidisciplinary integration environment for flight control development. In: AIAA Guidance, Navigation, and Control Conference, AIAA-1997-3773 (1997)
31. Vartio, E., Shimko, A., Tilman, C., Flick, P.: Structural mode control and gust alleviation for a sensorcraft concept. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit, AIAA-2005-1840 (2005)
32. Veenman, J., Scherer, C.: On robust LPV controller synthesis: A dynamic integral quadratic constraint based approach. In: IEEE Conference on Decision and Control, pp. 591–596 (2010)
33. Wu, F.: Control of linear parameter varying systems. Ph.D. thesis, University of California, Berkeley (1993)
34. Wu, F., Yang, X., Packard, A., Becker, G.: Induced l_2 norm control for LPV systems with bounded parameter variation rates. *International Journal of Nonlinear and Robust Control* **6**, 983–998 (1996)

Appendix: Induced L_2 analysis for LPV systems

This appendix briefly defines the L_2 norm for LPV systems and provides a set of conditions for computing a bound on this norm. The presentation essentially follows that given in [33].

Consider an LPV system of the form:

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \quad (6)$$

where (A, B, C, D) are continuous functions of ρ defined on a compact set $\mathcal{P} \subset \mathbb{R}^{n_\rho}$. The signal dimensions are $\rho \in \mathbb{R}^{n_\rho}$, $y \in \mathbb{R}^{n_y}$, $u \in \mathbb{R}^{n_u}$, and $x \in \mathbb{R}^{n_x}$. The dimensions of the state matrices are compatible with these signal dimensions.

ρ is a piecewise continuous function from \mathbb{R} to \mathbb{R}^{n_ρ} . It is assumed that $\rho(t) \in \mathcal{P} \forall t$. In addition, it is assumed that there exist $\{v_i\}_{i=1}^{n_\rho} \in \mathbb{R}$ such that $|\dot{\rho}_i| \leq v_i \forall t$. In other words, the $\{v_i\}_{i=1}^{n_\rho}$ are known rate bounds on the parameter trajectories. Let $\mathcal{F}_{\mathcal{P}, v}$ denote the set of piecewise continuous parameter trajectories that are restricted to \mathcal{P} and whose rates satisfy the bounds specified by $\{v_i\}_{i=1}^{n_\rho}$.

Let G denote the LPV system along with the set of allowable parameter trajectories $\mathcal{F}_{\mathcal{P}, v}$. The induced L_2 norm of G is defined as:

$$\|G\|_{i,2} := \sup_{\rho \in \mathcal{F}_{\mathcal{P}, v}} \sup_{\substack{u \in L_2 \\ \|u\|_2 \neq 0}} \frac{\|y\|_2}{\|u\|_2} \quad (7)$$

where $\|\cdot\|_2$ denotes the L_2 norm. In calculating this induced norm it is assumed that $x(0) = 0$. This norm is the maximum input/output gain over all inputs and all allowable parameter trajectories.

The following theorem, taken from [33], gives a condition for an upper bound on the induced L_2 norm.

Theorem 1. *If there exists a piecewise continuous function $X : \mathbb{R}^{n_\rho} \rightarrow \mathcal{S}^{n_x \times n_x}$ such that $X(\rho) > 0$ and*

$$\begin{bmatrix} A^T(\rho)W(\rho) + W(\rho)A(\rho) + \sum_{i=1}^{n_\rho} \beta_i \frac{\partial W}{\partial \rho_i} & W(\rho)B(\rho) & \gamma^{-1}C^T(\rho) \\ B^T(\rho)W(\rho) & -I_{n_u} & \gamma^{-1}D^T(\rho) \\ \gamma^{-1}C(\rho) & \gamma^{-1}D(\rho) & -I_{n_y} \end{bmatrix} < 0 \quad (8)$$

$\forall \rho \in \mathcal{P}$ and $|\beta_i| \leq v_i$ ($i = 1, \dots, n_\rho$) then:

1. the system G is parametrically-dependent stable over \mathcal{P} .
2. $\exists k$ with $0 \leq k < \gamma$ such that $\|G\|_{i,2} \leq k$.

Functions to compute an upper bound on the induced L_2 norm for an LPV system based on this result have been developed. There are two implementations of the induced LPV norm. The first computes an upper bound Γ on the induced L_2 norm of the pss G using a constant (parameter independent) Lyapunov matrix

$X(\rho) = X$. The use of a constant X is equivalent to computing the induced norm with no parameter rate bounds, i.e. a parameter-independent Lyapunov function can be used for rate unbounded analysis.

The second implementation computes a tighter (less conservative) bound on the induced L_2 norm by using a parameter dependent matrix Lyapunov $X(\rho)$ and bounds on the parameter rates of variation. $X(\rho)$ is assumed to be a linear combination of basis functions specified by

$$X(\rho) = \sum_{j=1}^{n_b} f_j(\rho) X_j \quad (9)$$

The functions $f_j : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}$ are piecewise continuous functions specified by the input `Xb`. The parameter rate bounds are specified by `RateUB` and `RateLB`. The toolbox searches for $\{X_j\}_{j=1}^{n_b}$ such that $X(\rho)$ satisfies the conditions in the theorem. The norm bound computed by this second syntax is less conservative at the expense of higher computational complexity.

The conditions in Theorem 1 are a parameterized set of linear matrix inequalities (LMIs) that must be verified for all $\rho \in \mathcal{P}$ and all $|\beta_i| \leq v_i$. The function `lpvnorm` approximately solves these conditions by enforcing the LMIs on a set of gridded points of \mathcal{P} . The terms involving parameter rates are handled by exploiting the fact that the β_i enter affinely in Equation 8. Specifically, if the LMIs hold for all combinations of $\beta_i = \pm v_i$ (a total of 2^{n_ρ} combinations) then they hold for all $|\beta_i| \leq v_i$.

The computational growth of these conditions is an issue. Let n_g denote the total number of grid points used to approximate \mathcal{P} . A rate bounded analysis must enforce the LMI conditions at all n_g grid points and for all 2^{n_ρ} combinations of $\beta_i = \pm v_i$. Thus there are a total of $n_g 2^{n_\rho}$ constraints each of dimension $(n_x + n_u + n_y)$. If there are n_b basis functions, then the Lyapunov matrix has n_b symmetric matrix decision variables $\{X_j\}_{j=1}^{n_b}$ each of dimension $n_x \times n_x$. This gives a total of $n_b \frac{n_x(n_x+1)}{2}$ individual decision variables in the rate bounded analysis. LMI optimization solvers have an asymptotic complexity that depends on both the number of decision variables and the number/dimension of LMI constraints. For example, `LMILab` has a floating point operation growth of $O(n_r n_v^3)$ where n_r is the total row dimension of the LMI conditions and n_v is the total number of decision variables [12]. This complexity assumes the default Cholesky factorization of the Hessian matrix is used to solve the least squares problem that arises in each iteration. Thus the complexity of solving the LPV analysis condition is roughly $O\left(n_g 2^{n_\rho} (n_x + n_u + n_y) (n_b n_x^2)^3\right)$. This growth limits the analysis to a modest number of parameters, grid points, and basis functions.