

# Gain Scheduling for the Orion Launch Abort Vehicle Controller

Sara J. McNamara\*, Carolina I. Restrepo†, Jennifer M. Madsen‡,  
Edgar A. Medina§ and Ryan W. Proud¶, Ryan J. Whitley||

NASA Johnson Space Center, Houston, TX, 77058, USA

One of NASA's challenges for the Orion vehicle is the control system design for the Launch Abort Vehicle (LAV), which is required to abort safely at any time during the atmospheric ascent portion of flight. The focus of this paper is the gain design and scheduling process for a controller that covers the wide range of vehicle configurations and flight conditions experienced during the full envelope of potential abort trajectories from the pad to exo-atmospheric flight. Several factors are taken into account in the automation process for tuning the gains including the abort effectors, the environmental changes and the autopilot modes. Gain scheduling is accomplished using a linear quadratic regulator (LQR) approach for the decoupled, simplified linear model throughout the operational envelope in time, altitude and Mach number. The derived gains are then implemented into the full linear model for controller requirement validation. Finally, the gains are tested and evaluated in a non-linear simulation using the vehicle's flight software to ensure performance requirements are met. An overview of the LAV controller design and a description of the linear plant models are presented. Examples of the most significant challenges with the automation of the gain tuning process are then discussed. In conclusion, the paper will consider the lessons learned throughout the process, especially in regards to automation, and examine the usefulness of the gain scheduling tool and process developed as applicable to non-Orion vehicles.

## Nomenclature

$A$	State matrix	$I$	Identity matrix
$ACM$	Attitude Control Motor	$IMU$	Inertial Measurement Unit
$API$	Application Program Interface	$J$	Cost function
$B$	Control Matrix	$K$	Gain or stiffness matrix
$BMGM2$	Bending Mode Gain Margin beyond the first bending mode, $dB$	$LAS$	Launch Abort System
$BMPM$	Bending Mode Phase Margin, $deg$	$LAV$	Launch Abort Vehicle
$BWCL$	Bandwidth, Closed-Loop, $Hz$	$LFGM$	Low Frequency Gain Margin, $dB$
$C$	System output matrix	$LQR$	Linear Quadratic Regulator
$CFD$	Computational Fluid Dynamics	$M$	Mach number or mass matrix
$CM$	Crew Module	$PID$	Proportional Integral Derivative
$D$	Control output matrix	$Q$	State weight matrix
$DOF$	Degrees of Freedom	$R$	Control weight matrix
$G$	Plant	$RBPM$	Rigid Body Phase Margin, $deg$
$F$	Force matrix	$S$	Sensor selection matrix
$HFGM$	High Frequency Gain Margin, $dB$	$V$	Velocity, $\frac{ft}{s}$
		$Z$	Damping matrix

\*Aerospace Engineer, Integrated GN&C, NASA Johnson Space Center, sara.j.mcnamara@nasa.gov.

†Aerospace Engineer, Integrated GN&C, NASA Johnson Space Center, carolina.i.restrepo@nasa.gov, AIAA Student Member.

‡Aerospace Engineer, Integrated GN&C, NASA Johnson Space Center, jennifer.m.madsen@nasa.gov.

§Aerospace Engineer, Integrated GN&C, NASA Johnson Space Center, edgar.a.medina@nasa.gov, AIAA Senior Member.

¶Orion GN&C Ascent Abort System Manager, Integrated GN&C, NASA Johnson Space Center, ryan.w.proud@nasa.gov.

||Aerospace Engineer, Flight Mechanics & Trajectory Design, NASA Johnson Space Center, ryan.j.whitley@nasa.gov, AIAA Senior Member.

$p$	Roll rate, $\frac{deg}{s}$	$i\alpha$	Integral angle of attack, $deg \cdot s$
$q$	Pitch rate, $\frac{deg}{s}$	$\beta$	Angle of sideslip, $deg$
$r$	Yaw rate, $\frac{deg}{s}$	$i\beta$	Integral angle of sideslip, $deg \cdot s$
$u$	Control vector or actuator input	$\eta$	Control input
$x$	State vector	$\phi$	Roll angle, $deg$
$y$	Output vector or sensor output	$\delta$	Sensor deflection
$\Phi$	Flexible body modes	$\delta_{acm}$	Thrust delta on ACM, $lb_f$
$\alpha$	Angle of attack, $deg$		
<i>Subscript</i>			
$cmd$	Commanded	$i$	Iteration number
$err$	Error	$nav$	Navigated
$fp$	Pitch forward loop gain	$ref$	Reference
$fy$	Yaw forward loop gain	$T$	Target

## I. Introduction

The focus of this paper is the design process that starts from a design flight condition, creates a linearized plant, performs gain calculations, uses the full linear model to check requirements and then validates the results in a non-linear simulation that includes Monte Carlo verification. The challenge of this process is to cover the wide range of vehicle configurations and flight conditions experienced during the full envelope of potential abort trajectories from the pad to exo-atmospheric flight. Several factors are taken into account in the automation process for gain calculation including the abort effectors, the environmental changes and the autopilot modes. An overview of the LAV controller design and a description of the linear plant models are presented. In Figure 1 brief detail of a LAS abort timeline is shown with the available control effectors and guidance phases, which will be discussed in more detail in later sections of this paper.

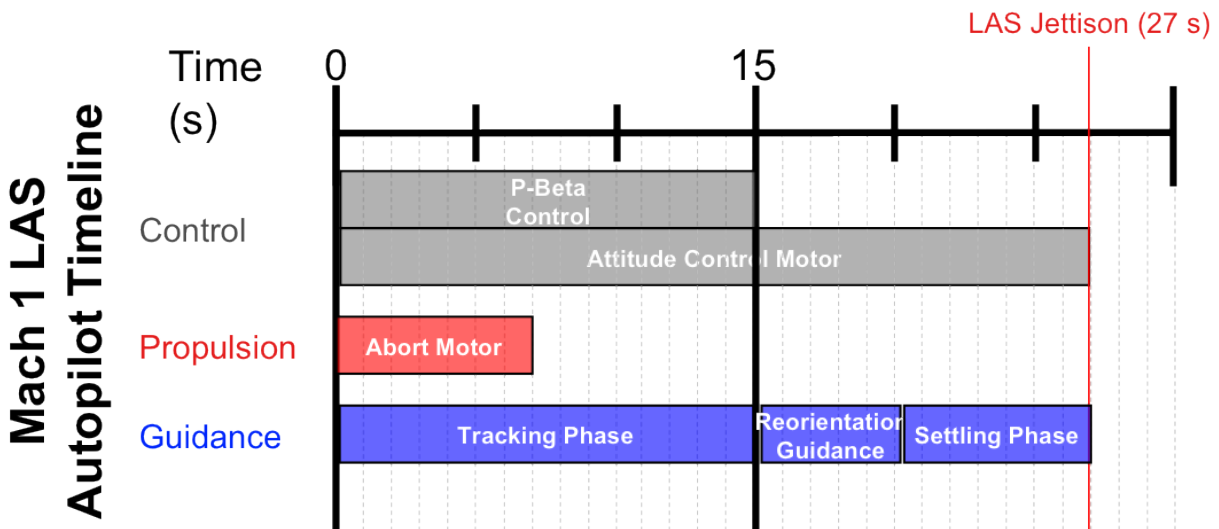


Figure 1: Example of LAS Autopilot Timeline during an Abort at Mach 1.

The gain scheduling process begins with a linearized rigid body model. The rigid body model is derived from a heritage model used on X-38 and earlier atmospheric spacecraft and is modified for use with the LAV and its effectors. The control algorithm is designed to tell the ACM how much thrust is needed and in which firing direction in order to meet constraining requirements. The aerodynamic characteristics of the LAV are captured in the rigid body model and are the main driver of the control system design. For the LAV the aerodynamics not only include the characteristics of the vehicle, but also the jet interactions between the abort motor, ACM and vehicle shell. The aerodynamics determine whether the vehicle is inherently stable or unstable, and in the case of the LAV, the vehicle is unstable for most of the abort flight. Thus, the gain scheduling process is further challenged by the instability of the vehicle. The control authority is tested

most during an abort in the transonic region of ascent, and as such this region is the most sensitive for gain scheduling.

Due to the complexity of the LAV flight, autopilot modes were developed each having different desired flight goals. There is a separation, tracking, reorientation and settling phase of the control algorithm. In the tracking phase, for example, the primary goal of the controller is to track zero angle of attack and angle of sideslip and the secondary goal is to null any rates on the vehicle. In the yaw channel this is accomplished through a  $p - \beta$  coupling that can be used due to the offset center of gravity (CG) location on the LAV. This advantage is called  $p - \beta$  because of the angle of sideslip and roll rate coupling on the axis symmetric vehicle with an offset CG. However, the  $p - \beta$  coupling also makes the yaw channel a non-tracking controller, which adds difficulty to determining the control goals of the yaw channel. All of these considerations to the flight objectives and design goals were used to create the controller topology for the pitch and yaw channels individually. Those topologies will be discussed in detail related to the gain scheduling process. After many attempts to design a controller suitable to meet all the desired performance goals, a phase-stabilized autopilot was chosen as opposed to a gain-stabilized autopilot. The history of the autopilot design will be discussed in short detail.

Gain scheduling is accomplished using a linear quadratic regulator (LQR) approach for the decoupled, simplified linear model throughout the operational envelope. Many different tuning methodologies were considered and LQR proved to be the simplest to use with the growing complexity of the aerodynamic characteristics of the vehicle. A simplified linear model is used to acquire the gains that meet the desired control goals and is compatible with the LQR methodology of a full feedback system. The longitudinal motion and lateral-directional motion are considered and tuned separately. LQR derived gains are scheduled across the full flight envelope in a three-dimensional table dependent on time since abort, altitude and Mach number. During the tuning process it is imperative to develop smooth gain transitions over time since the gain tables are used by interpolation. Breakpoints in the table are determined by the most critical flight regimes in time, altitude and Mach number. More breakpoints are added in the lower altitude abort region than in the higher altitudes due to the drastic differences in the environment between the two. Here also, examples of the most significant challenges with the automation of the gain tuning process will be discussed. Once the gains are finalized, they are then implemented into the full linear model for linear requirement validation. The full linear model includes the flexible body model and any latencies within the controller, signal processing or the actuator model of the ACM.

Finally, the gain tables are tested and evaluated in a non-linear simulation using the vehicles flight software to ensure the controller behaves as predicted and that performance requirements are met, such as preventing the vehicle from tumbling. Those results will be presented in brief detail to close out the gain tuning process. In conclusion, the paper will discuss the usefulness of the gain scheduling tool, future additions to the process and the generic gain scheduling process developed as applicable to non-Orion vehicles.

## A. Description of LAV

The LAV is comprised of the Crew Module (CM) and Launch Abort System (LAS) that, for the purposes of this paper, includes an abort motor and Abort Control Motor (ACM). The abort motor provides the large thrust to pull the LAV off of a failing booster rocket and is a solid rocket with non-vectorable thrust. The ACM is the effector of the LAV providing combinatorial side and normal force on the vehicle, which translate to a moment at the CG. The ACM is a solid propellant motor with one pressure chamber and eight pintle nozzles. The LAS controller provides the desired thrust and direction command to the ACM where it's own software converts the desired commands into appropriate configurations of the eight pintles.

## B. Mission Requirements

The LAS is tasked to abort from a failing booster rocket and safely return the crew in a water landing. In order to do this effectively, the system must control attitude and null any attitude rates imparted by the failure while performing a targeted maneuver to land safely. Requirements are imparted on the system to ensure success. The requirements that effect the LAS controller design include maintaining control of the vehicle, structural load limits and a targeted attitude at LAS jettison to ensure successful chute deploy conditions. There are also specific gain and phase margin requirements detailed in Table 1.

All requirements are verified through Monte Carlo analysis. While controllability is a necessity for success, the

Table 1: Linear Margin Requirements

Margin	Requirement
Phase	$\geq 20$ deg
Gain	$\geq 3$ dB

structural loads requirement competes with the application of control. Pitch profiles are carefully designed for low altitude abort trajectories with the limitation of not only the available control authority from the ACM, but also the limitation of the loads imparted on the LAV structure during a high pitch maneuver. Additionally, load limits are considered again during the reorientation phase to flip the LAV to heat shield forward flight and target a safe LAS jettison state. While the guidance design will not be discussed in this paper, the structural load limit is an example of one requirement limiting the control authority use.

### C. Trajectories

To survey the LAS abort flight regime and set the design space, un-dispersed trajectories aborting at important initial conditions throughout the booster rocket trajectory are generated from six-DOF simulation. These initial conditions include a dense sampling of the low altitude abort conditions, the region of flight most affected by aerodynamic contributions and dynamic pressure, flight through the transonic region and a more coarse sampling of higher altitude LAS aborts out to the time of nominal LAS jettison. Fro LAV flight, transonic aborts and flight during the maximum dynamic pressure region prove to be the most challenging conditions for controller design. The flight characteristics across the LAV flight envelope are used to verify the controller design for meeting linear design goals and also to create the controller tuning design cases that are scheduled in Mach number, altitude and time since abort. The flight parameters are either null were appropriate or scheduled on Mach number, altitude or time. Examples of appropriate null parameters include angle of attack and angle of sideslip. Abort motor thrust and LAV mass are scheduled versus time for the design cases. Once the controller design has been implemented in the six-DOF simulation, Monte Carlo analysis is also performed across the LAS abort flight regime to verify nonlinear requirements with the new controller design.

## II. LAV Linear Models

The linear models of the airframe, sensor, controller and actuator are required to create the design plants needed for the gain scheduling process. A flow chart of the linear models for the closed-loop response is presented in Figure 2. The airframe model is broken into a rigid body model and a flexible body model, which will each be discussed in more detail in the subsequent subsections. The Inertial Measurement Unit (IMU) sensor model takes in the true state outputs from the airframe model and outputs the sensed states for the controller model. The sensor is modeled as a simple second-order transfer function with a natural frequency at 50 Hz and a damping ratio of 0.707. Section III will give a detailed overview of the controller model. The linear controller model takes in the desired states and sensed states and outputs a command to the actuator model. The actuator model is also a simple second-order transfer function, but with a natural frequency at 15 Hz and 0.6 damping ratio. The actuator model simply converts the controller command output to an applied force for the airframe model input. The transfer function equation for the sensor model and actuator model is defined in 1. All latencies, such as data delay, command cycle sampling and rate estimation, are modeled together and applied prior to the actuator model with a second-order Pade formula. The four models can be configured for open-loop or closed-loop analysis.

$$G_{acm}(s) = \frac{K_{acm}^2 \omega_{nacm}^2}{s^2 + 2\zeta_{acm} \omega_{nacm} s + \omega_{nacm}^2} \quad (1)$$

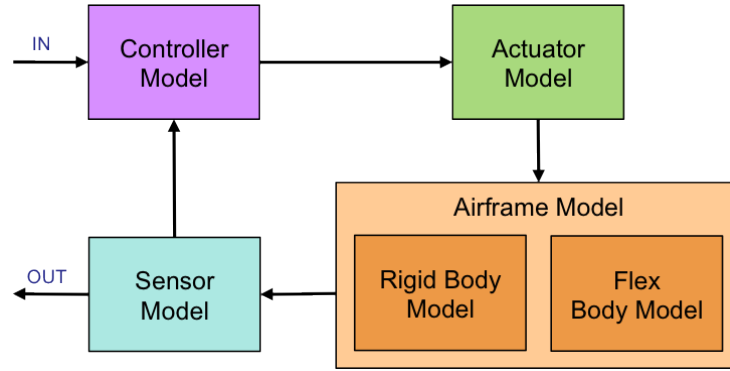


Figure 2: Linear closed loop model that includes an airframe, sensor, controller and actuator model.

### A. Rigid Body Model

The rigid body model is derived from a heritage model used on X-38 and earlier atmospheric spacecraft and is modified for use with the LAV and its effectors. The equations of motion are of the basic form shown in Equations 2 and 3, where the state vector,  $x$ , is made up of the vehicle’s position, velocity, body rates and rotation angles. The control vector,  $\eta$ , is vehicle specific and for the LAV, ACM side and normal force are the vehicle’s effectors.  $A$  and  $B$  are the system matrix and control matrix, respectively and  $C$  and  $D$  are the output system matrix and output control matrix, respectively.

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} \quad (2)$$

$$\vec{y} = C\vec{x} + D\vec{u} \quad (3)$$

The equations of motion are detailed in Reference XX and include few assumptions or simplifications. The only assumption made is that the abort motor thrust is not included in the model. Additionally, the derivatives of the abort motor thrust or induced moments with respect to the state inputs have been proven to be negligible. Since the model is input driven, simplifications can be applied in the input configuration as desired such as assuming zero body rates on the vehicle. A trim simplification is used for the aerodynamic inputs to the rigid body model where the available ACM control authority is used to ensure a null disturbance torque on the flight condition about to be linearized.

The aerodynamic characteristics of the LAV are the major component captured in the rigid body model. The aerodynamics are provided from an API to a database based on wind tunnel testing and CFD. The aerodynamics determine if the vehicle is stable or unstable in a particular flight regime and thus, determine the control authority necessary to maintain control of the vehicle. In the case of the LAV, the vehicle is inherently unstable for most of the abort flight regime during the abort motor burn. The stability derivatives are calculated from the aerodynamic database using central differencing of small perturbations. The perturbation for each parameter is given in Table 2.

From the input vehicle flight conditions and aerodynamic characteristics, the model provides the system matrix which is a twelve by twelve matrix filled with the stability derivatives with respect to the state vector. The model also provides the control matrix based on the effectors of the vehicle. The same principles applied for creating the LAV specific control matrix could be applied to any vehicle. Additionally, the system and control matrices can be uncoupled for independent lateral-directional or longitudinal motion analysis, which is often the desired in the LAS controller design analysis.

### B. Flexible Body Model

A flexible body model is provided in the form of a matrix of mode shapes and slopes at various nodes along the LAV and the frequencies of those modes. Six degrees of freedom are provided for each mode at each node. The standard state-space equations, Equations 2 and 3, are used for the flexible body effects also,

Table 2: Perturbations for Aerodynamic Derivatives

Parameter	Perturbation
V	$\pm 100 \frac{ft}{s}$
M	$a(V + \Delta v)$
$\alpha$	$\pm 1^\circ$
$\beta$	$\pm 1^\circ$
p	$\pm 1 \frac{\circ}{s}$
q	$\pm 1 \frac{\circ}{s}$
$\dot{\alpha}$	$\pm 1 \frac{\circ}{s}$
r	$\pm 1 \frac{\circ}{s}$
$\dot{\beta}$	$\pm 1 \frac{\circ}{s}$
$\delta_{acm}$	$\pm 100 lb_f$

where  $x$  represents the internal mode states,  $u$  is the actuator input and  $y$  is the sensor deflection output. The sensor deflections,  $\delta$ , are related to the mode states by the  $\Phi$  matrix,

$$\vec{\delta} = [\Phi] \vec{x} \quad (4)$$

The  $A$  and  $B$  matrices are derived from the standard differential equation of a dynamic system:

$$[M] \ddot{\vec{\delta}} + [Z] \dot{\vec{\delta}} + [K] \vec{\delta} = [F] \vec{u} \quad (5)$$

where  $M$  is the mass matrix,  $Z$  represents the damping matrix and  $K$  is the stiffness matrix. Using equations 4 and 5, the  $A$  and  $B$  matrices can finally be defined.

$$A = \begin{bmatrix} 0 & I \\ -[\Phi][K][\Phi] & -[\Phi][Z][\Phi] \end{bmatrix} \quad (6)$$

$$B = \begin{bmatrix} 0 \\ [\Phi][F] \end{bmatrix} \quad (7)$$

Considering the  $C$  and  $D$  matrix, the sensor deflection output can be related by:

$$y = \begin{bmatrix} S_1 \delta \\ S_2 \dot{\delta} \\ S_3 \ddot{\delta} \end{bmatrix} \quad (8)$$

where  $S$  represents a sensor selection matrix. Combining Equations 3, 6, 7 and 8,  $C$  and  $D$  are found to be:

$$C = \begin{bmatrix} [S_1][\Phi] & 0 \\ 0 & [S_2][\Phi] \\ -[S_3][\Phi][\Phi][K][\Phi] & -[S_3][\Phi][\Phi][Z][\Phi] \end{bmatrix} \quad (9)$$

$$D = \begin{bmatrix} 0 \\ 0 \\ [S_3][\Phi][\Phi][F] \end{bmatrix} \quad (10)$$

As seen in Equation 5, the flexible body model is also dependent on the vehicle mass, thus a selection flag is included in the model which is dependent on time in estimating the mass of the LAV. The flexible body model returns a state-space system for integration with the rigid body model to complete the linear airframe model.

### III. LAS Control System Overview

Throughout the development of the Orion project, different concepts for the LAS controller have taken shape. Initially, a gain stabilization approach was used since it was thought that actively damping flexible body modes would be very hard to accomplish with the large uncertainties in the early design cycles. Thus, low pass filters were used to shape the loop gain at the flexible body mode frequencies. Since that time, more instability has been introduced into the LAS plant due to rigid body modes and a phase-stabilized autopilot has been developed to phase stabilize the first bending mode and eliminate the instabilities at the rigid body modes. Also, a PID feedback controller was considered in a previous design cycle and today, the design stands as a proportional integral plus rate feedback controller. Because the body rate state is also fed back to the controller, the derivative control is there in the details, but not as a derivative gain directly. Notably, this type of controller topology integrates well with the LQR method for tuning gains since it can be reformatted to a non-zero set point controller, which will be discussed in Section IV. Additionally, in previous designs, the gains were computed from tables based on mass property data. Currently, the gains are calculated off-line and scheduled in a three-dimensional look-up table in time since abort, Mach number and altitude. This gain scheduling process is the topic of this paper, but first an introduction to the current LAS controller must be given.

#### A. Controller Topology

The LAS controller is split into a pitch channel autopilot and a yaw channel autopilot to control the two directions of motion, lateral-directional and longitudinal, uncoupled with their respective ACM command, a desired yaw force and a desired pitch force. Each channel has one output: a desired ACM command, that when considered together produce an ACM thrust and direction command to the actuator model. This gives two-axis control without direct roll control. Due to the offset z CG position on the axis-symmetric LAV, roll rate,  $p$ , and  $\beta$  are a cross-feed in the yaw channel that can be used to control the roll disturbance to the vehicle.

##### 1. Pitch Channel

The pitch channel autopilot, shown in Figure 3, requires the feedback of  $\alpha$  error,  $\alpha_{err}$ , and pitch rate error,  $q_{err}$ . The controller is also augmented with an  $\int \alpha$  state to further minimize  $\alpha_{err}$ . This gives three inner-loop gains on the pitch channel:  $K_\alpha$ ,  $K_{i\alpha}$  and  $K_q$ , that are scheduled in time, altitude and Mach number, which is the discussion topic of this paper. There are also two forward loop gains,  $K_{fp}$  and  $T_{ref}$ , but their purpose and tuning are not included in the topic of this paper.  $K_{fp}$  can be considered as a final gain on the entire system and  $T_{ref}$  is a gain to scale the command to a thrust value in  $lb_f$  as the input to the actuator model. Bending filter and lag-lead filters are also used to shape the stability of the system and limiters are used through out to either limit the amount of error introduced in the controller or to favor a particular state over another. The pitch channel controller tracks the  $\alpha$  state.

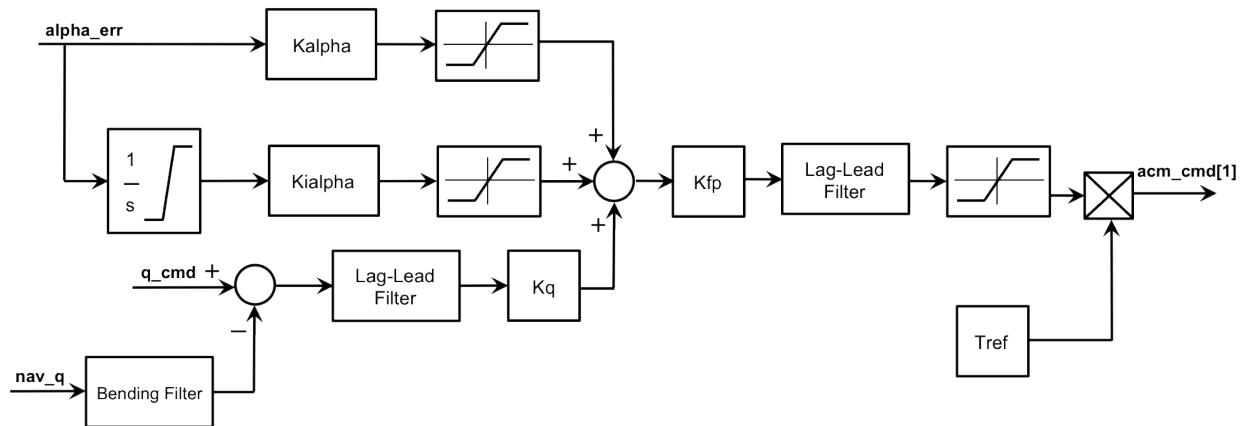


Figure 3: LAS Autopilot Pitch Topology.

## 2. Yaw Channel

The yaw channel autopilot, shown in Figure 4, requires the feedback of  $\beta$  error,  $\beta_{err}$ ; roll rate error,  $p_{err}$ ; and yaw rate error,  $r_{err}$ . The controller is also augmented with an  $\int \beta$  state and an  $\int p$  state to minimize the  $\beta_{err}$  and  $p_{err}$ , respectively. As shown in the yaw channel topology, the yaw channel uses the cross feed of  $\beta$  and  $p$  to also control the roll rates on the vehicle. This creates five inner-loop gains in the yaw channel:  $K_\beta$ ,  $K_{i\beta}$ ,  $K_r$ ,  $K_p$  and  $K_\phi$ . For clarification,  $K_\phi$  is equivalent to  $K_{ip}$ , but as will be shown in Subsection A of Section IV,  $\phi$  is the feedback state used to tune the  $K_\phi$  gain, so it is named as such. Similar to the pitch channel, the design of the feed forward gains, filters and limiters in the yaw channel will not be discussed in this paper. The yaw channel controller tracks the  $\beta$  state.

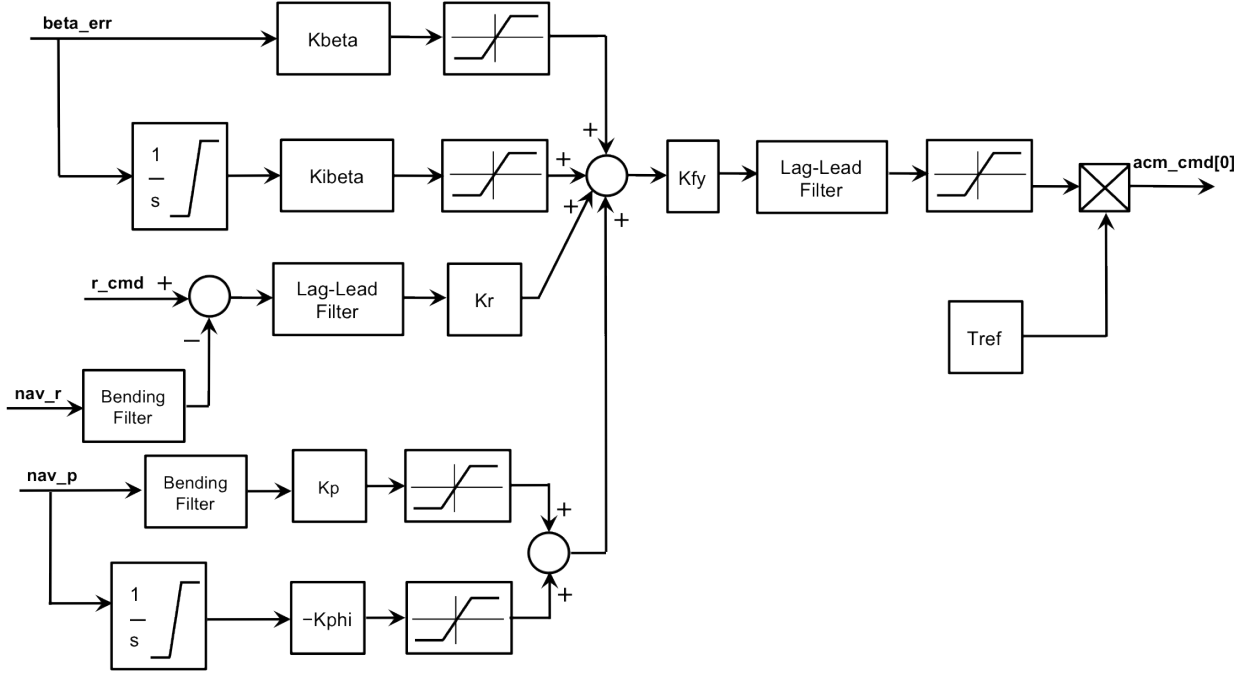


Figure 4: LAS Autopilot Yaw Topology.

## B. Development of Gains

There are two sets of gains in the LAS controller: a set in the inner-loop autopilot and a set in the feed forward path. Gain scheduling in time, Mach number and altitude is completed for the inner-loop PID gains of the pitch and yaw channels. The feed forward gains are currently not scheduled. In early designs, the gains for both channels were chosen from a pole-placement methodology to try to remove the instabilities in the system. However, with the complexity of the yaw-roll coupling in the yaw channel a more direct approach was desired where designers could have direct influence on the individual states of the system and emphasize one over another during different portions of flight. This led to both the gain scheduling approach and also using the LQR methodology to tune the controller. Investigating LQR, the yaw channel topology was reconfigured to the current PID controller. With the success of LQR in the yaw channel, the pitch channel was also easily reconfigured and the topology used with LQR tuning is what is presented in Figures 3 and 4. The gains tables are also separated into a tracking phase table and a settling phase table. As mentioned earlier, the autopilot tracking phase begins after the separation autopilot and the goal of the tracking phase is to track a zero  $\alpha$  and  $\beta$  command. The tracking phase ends at the start of the reorientation phase where the vehicle orientation is flipped from flying tower forward to heat shield forward to prepare for the descent and landing phase of CM flight. During the reorientation phase, the tracking gains are latched and held constant until the start of the settling phase. The settling phase begins at the end of reorientation and has the goal to target the CM trim conditions in  $\alpha$  and  $\beta$  in order to leave the CM in as stable of a configuration as possible post LAS jettison for free flight and chute deploy con-ops. The controller con-ops were shown



previously in Figure 1.

## IV. LQR Process

A brief introduction to LQR will be given here to familiarize the reader with its uses for the LAS controller tuning. A linear quadratic regulator (LQR) is a full-state feedback system whose gains are an optimal compromise between the control inputs and the states' deviations from zero. The name implies that the method works only for linear systems based on a quadratic cost function to regulate a signal to zero. The goal of LQR is to find the control law,  $u = -Kx$  such that the cost function:

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (11)$$

is minimized, where  $Q$  and  $R$  are the weights on the states and control inputs, respectively. LQR requires  $Q$  to be a positive semi-definite matrix and  $R$  to be a positive definite matrix for  $J$  to have a global minimum.  $Q$  and  $R$  are used to emphasize the corresponding state or control input in the cost function. Additionally, the ratios between the  $Q$  weights and the  $Q$  and  $R$  weights matter. For example, if the elements of  $R$  are larger than the elements of  $Q$ , then it costs more to minimize the control inputs than minimize the state error. This method also allows designers to have direct influence on the individual states of the system rather than the poles of the system in a pole-placement methodology. This has greater benefit in an unstable system where control authority is in high demand.

### A. Simplified Plant

To apply the LQR methodology to the LAS controller, a simplified plant is created from the PID controllers shown in Figures 3 and 4 that maintains a non-zero set point for the LQR method to regulate. The state-space equations outlined in Equations 2 and 3 must be rewritten for a tracking controller to minimize the state error, that is to minimize the actual command minus the desired command. To include full-state feedback control, the state-space equation must also be augmented with the integral state and rewritten as follows:

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{x}}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ D \end{bmatrix} u \quad (12)$$

where  $\tilde{x}$  is the error between the actual and commanded state and  $x_I$  is the error in the integral state. For full feedback control, the states of the pitch channel are  $V$ ,  $\alpha$ ,  $q$  and  $\theta$ . However, only  $\alpha$  and  $q$  are fed back in the actual LAS controller. Since  $\int \alpha$  is also a state of the pitch channel controller, the plant is augmented with an  $\int \alpha$  state to create the simplified plant used for LQR tuning. Thus, the LAS controller is simplified to a two-state feedback controller in  $\alpha$  and  $q$  only as shown in Figure 5.

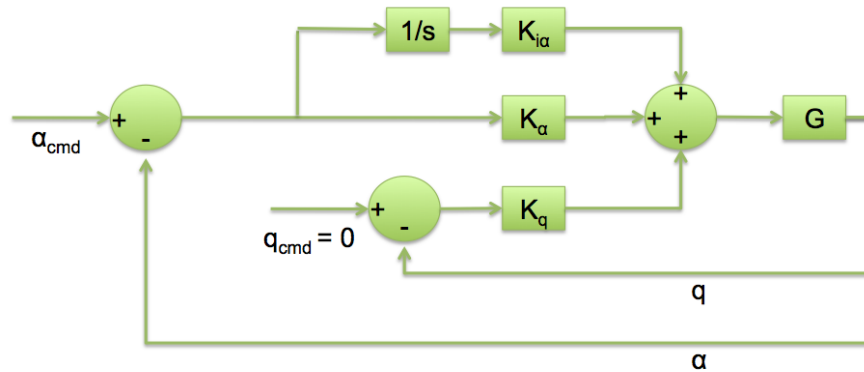


Figure 5: Simplified Pitch Plant for LQR Tuning.

Similarly for the yaw channel,  $\beta$ ,  $p$ ,  $r$  and  $\phi$  are available for feed back in the linear plant, so a simplified

plant with these four states for feedback is created for LQR tuning purposes. The yaw channel simplified plant is also augmented with  $\int \beta$  to maintain full state feedback. The  $\phi$  state is used in place of  $\int p$  for full state feedback and ensures full rank controllability for LQR. The plant used for LQR tuning in the yaw channel is shown in Figure 6.

In both channels, the simplified plants used in LQR tuning ignore the flexible modes and only include the rigid body modes.

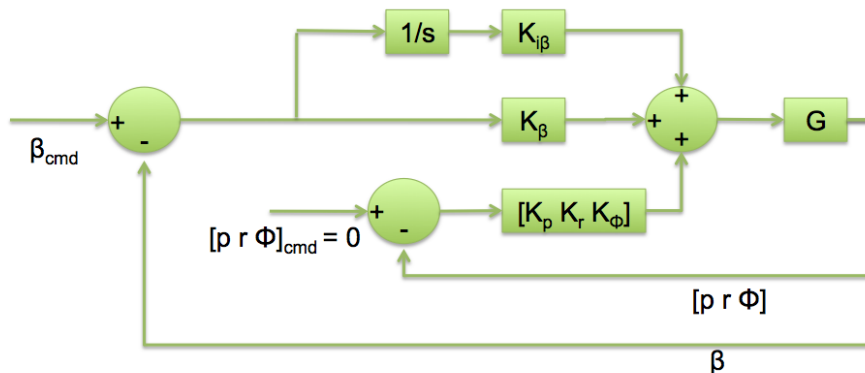


Figure 6: Simplified Yaw Plant for LQR Tuning.

## V. Gain Scheduling Tool

The gain scheduling tool incorporates three functionalities: linearization, the design space and LQR tuning. The linear models have been explained in brief detail in Section II and the LQR methodology was introduced in Section IV. The selection and details of the design space will first be introduced here before explaining the design approach in Subsection D. With these three functions to the tool, the gain scheduling process requires inputs of the design point flight condition and the design goals for the linear requirements. The tool then outputs the final gains of the particular design point for integration into the gain tables, which will be discussed in Subsection E.

### A. Design Space

Considering the tracking and settling phases of flight, meaningful breakpoints were created in the gain tables to define flight conditions requiring specific gain definitions. The tracking and settling phases are tuned individually, independent of one another. Since the tracking phase is a very dynamic phase of flight, breakpoints in time were created at every second from abort initiation to reorientation start. Additionally, Mach number breakpoints were defined consistent with the aerodynamic database to leverage on the changing dynamics due to aerodynamics. Similarly, five breakpoints in altitude were chosen to define the zero altitude, the most dynamic altitude at approximately the transonic abort region, an altitude at the end of the sensible atmosphere region, an upper altitude at the beginning of benign exo-atmospheric flight and then an end table altitude of 999,999 *ft* for interpolation to the nominal LAS jettison altitude. The settling table has fewer time breakpoints just marking the first available settling start time, the end of settling phase and a few breakpoints to ensure that interpolation for hold-last-value is held constant. The same Mach number breakpoints and altitude breakpoints as the tracking phase are used in the settling phase.

Benchmark Monte Carlo runs that include dispersions to mass properties, aerodynamics, atmosphere, etc. are created with the current nonlinear simulation and controller model to envelope the LAS abort flight regime. Considerations to the design space could also include different launch months, failure modes of the booster rocket, launch azimuth angles and guidance pitch profiles. These trajectories are then laid out in time, altitude and Mach number to define which areas of the gain tables are covered by the current flight envelope. The areas covered are then chosen for further investigation with the linear analysis tools. The addition of finer detail or the removal of breakpoints is contingent upon a number of factors including the flight envelope within the existing breakpoints and the need for finer detail in the gains due to rapidly changing

flight conditions. The latter may not be known until iterations begin on the gain tuning process.

## B. Design Points

Because it would be impossible to define a set of gains at a particular breakpoint that will satisfy every case that either may pass through that flight condition or interpolate from that flight condition, a set of design points within the areas of the gain tables covered by the LAS flight envelope are created for tuning the gains. The design points are flight conditions at the time, altitude and Mach number breakpoints with some parameters scheduled in time, such as mass properties. Other parameters are held constant based on known information, such as gravity, vehicle reference length and Euler angles. And another set of parameters are set to zero, such as body rates and attitude, to simplify the rigid body effects. The design points are then assessed in the linear analysis tool with the latest aerodynamic database, controller gains, latencies, etc. to determine which points meet the pass/fail criteria for redesign. Considering the dispersed requirements of 3 dB gain margin and 20 deg phase margin described in Subsection B of Section I, the design goals for the un-dispersed trajectories are 6 dB gain margin and 30 deg phase margin. This usually leads to the dispersed cases meeting the requirements. Also, a design goal is imposed on the closed-loop bandwidth to have some sort of tuning goal on performance impacts. Design points not meeting these goals are selected for redesign. Depending on the severity and location of failed design points, additional points in these areas may be considered to ensure smooth interpolation between the gains in the tables.

## C. Starting Weights

Normally, the  $Q$  and  $R$  weights to kickoff the controller tuning process would be gathered from the previous design cycle's ending weights that produced the gain tables. However, instances will arise where new breakpoints are adding and a starting point for iteration is needed. Or when initially starting the LQR process, the weights, or good choices on weights, are unknown. If the gains, or a good estimate of usable gains, is known than the starting weights for the gain tuning process can be derived from the LQR method in the reverse order. An iteration scheme using Newton's method was used to derive the weights given a set of controller gains as shown in Equation 13 where the subscript  $T$  represents the targeted gain and  $i$  represents the iteration number. Newton's method is used to target the gains by changing the weights and a first-order approximation is used to estimate the dependency of the gains on the weights. Note that the cost function used in LQR may have multiple minimums, so multiple weight combinations may exist. Also, as defined in the LQR theory, weights cannot be negative, which eliminates some resulting weight combinations. Additionally, since LQR is about the ratios between the weights, a simple scale factor can be applied to any result of this iteration scheme and the exact same set of gains will result. This was used once a solution was found to create more uniform starting numbers, such as 10 or 1. There is also a sensitivity to step size on the weights, so it is changed dynamically as the iterations converge to answer. The exit criteria for a solution is tolerance with the target gain value of  $10E - 2$ .

$$Gain_T = Gain_i + \frac{\partial Gain}{\partial Weight} \cdot Weight_i \quad (13)$$

For the LAS controller,  $\alpha$  and  $\beta$  tracking are the primary concern, so the  $Q_\alpha$  and  $Q_\beta$  weights are an order of magnitude larger than any other weight in their respective channels. Also, a dependency with altitude and time was found for the  $R$  weight since it has a larger direct impact on the system as a whole. An example for the yaw channel is shown in Figure 7 where the colors represent abort initiation altitudes starting at the pad (blue) to nominal LAS jettison (red). Thus, the starting  $R$  weight in the pitch and yaw channels were coarsely scheduled over time and altitude with hopes of better results in the tuning iteration to be discussed in Subsection D. In the yaw channel, the weights on  $i\beta$  in some altitude cases were on the order of the  $\beta$  weights to emphasize the  $p - \beta$  control. Also, due to the complexity and increased number of gains in the yaw channel,  $Q_{i\beta}$ ,  $Q_r$  and  $Q_\phi$  were all scheduled in a coarse altitude and time table, much like  $R$ , to start the tuning process.

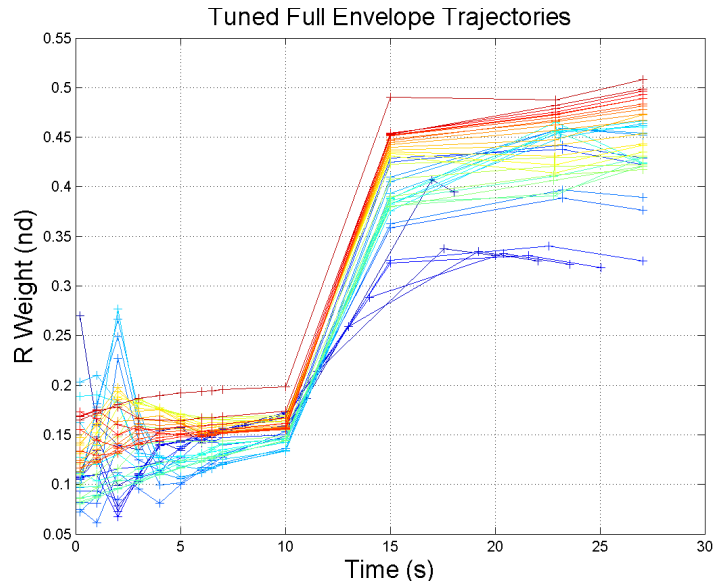


Figure 7: Example of R Starting Weight for Yaw Channel.

#### D. Design Approach

Multiple tuning approaches were considered with the LQR method. In the latest design cycle, 288 design points were selected for redesign in the gain tables, so a multiple designer approach was investigated. This method broke the gain tables into manageable chunks assuming each design point would require intensive iterative tuning. While acceptable gains were produced with this method, discontinuities existed in the resulting LQR weights between designers, which lead to discontinuities in the gain tables. Once applied in the six-DOF simulation to trajectories, the discontinuities would induce undesired interpolation results between the breakpoints of the table, which are not tested in the design point tuning approach. Thus, a "one designer" method was desired in which all design points could be considered at once and tuned by one designer without requiring a lot of time and cost to schedules.

Studying the trends between weights and gains and the trends between gains and margins, useable trends in the LQR weights were noted. For example, in most cases, when  $K_\alpha$  was increased, RBPM decreased. To increase the  $K_\alpha$  gain, the  $Q_\alpha$  weight was increased. Care must also be taken with the  $Q_\alpha$  weight since it also affects the  $K_q$  gain of the pitch channel; this effect is small in comparison to the effect on  $K_\alpha$ . Since the integral states were added to the simplified plant, they are the only case in which their gain is only affected by their weight. However, the vice versa is not true. Noting trends similar to this for all gains and weights, Newton's method was used to iterate on the LQR weights in a "one designer" approach. Iterations were completed until the linear margins for all design cases were met. In effect, Newton's method is the wrapper of the gain scheduling tool in which all design point can be considered at once. A simplified equation is shown below:

$$Weight_T = Weight_i + (Margin_T - Margin_i) \cdot \frac{Weight_{i+1} - Weight_i}{Margin_{i+1} - Margin_i} \quad (14)$$

where the subscript  $T$  represents the target or design goal and  $i$  represents the iteration number.

With the LQR method, the inputs are the weights which produce gains. Those gains are then used to assess the design space and determine if the targeted margin design goals have been met. If not, Newton's method is used to update the next weight "guess" such that the desired margin,  $Margin_T$ , is met. Since this approximation is linear,  $Margin_T$  may not be met by  $Weight_T$  and iterating is involved until acceptable margins are found. Only one margin, or design goal in the case of closed-loop bandwidth, and one weight can be considered at a time. Therefore, some iterating is also involved to find the balance in the weights that produce the desired margins. The trends learned earlier in the process help decide which weights to chose for each design goal. However, the iterations used in the "one designer" approach are less time consuming than iterating on each design point with multiple designers. From experience, the LQR space is highly nonlinear

and first derivative information is unknown. Thus, a small sensitivity analysis is performed to determine the appropriate step size in the weight for approximating the first derivative required in Newton's equation. Then, a slope approximation is used in place of the actual first derivative. After each iteration, the margins are assessed and only design points still failing to meet design goals are updated with Newton's equation. The gains of the design points in compliance are stored for integration into the gain tables.

Also, as the linear margins get closer to the target margins, the change to the weight with each iteration will be smaller and smaller. For example, one design point may come into compliance after one iteration. However, the whole design space may take an additional three iterations to meet the goals. Therefore, the design point in compliance within one iteration will still be considered in the additional three iterations even if little to no change is needed and applied.

The severity of the failures in the design goals will determine which design goal to start with. Additionally, the trends analysis will help determine which weight to alter to impact the design goal the most. In some cases, two weights are necessary to balance the negative and positive effects of tuning each gain. For example, as mentioned earlier,  $K_\alpha$  could be used to reduce *RBP*M. However,  $K_\alpha$  also reduces *HFGM* and could reduce *HFGM* beyond the design goal. In this case,  $K_{i\alpha}$  could then be used to bring both design goals into compliance.

## E. Table Creation

Once the tuning process is complete and all design points are compliant with all design goals, the final gain sets can be integrated into one table. This is a simple task since the breakpoints of the table have already been laid out in time, altitude and Mach number and due to the fact that gain tuning occurred at these specific breakpoints. Creation or removal of breakpoints in the gain tables would be warranted if the design space showed a need for finer or less detail and is already considered in tuning the design space. The following procedure occurs to update the gain tables.

1. Design points gains are updated in the tables.
2. Non-design breakpoints are updated to either:
  - a. Hold-last-value with the nearest neighbor if they are outside of the LAS flight envelope.
  - b. Keep the value of previous design cycle if they were compliant at the initiation of the design cycle.
3. Hold-last-value is used for first and last breakpoints of the gain tables in time, altitude and Mach number to ensure smooth interpolation, and not extrapolation.

To elaborate on the third step, in the tracking gain tables, the final altitude breakpoint in the LAS controller gain tables is at 999,999 *ft*, an infinity point, and the gains at the last exo-atmospheric altitude breakpoint populate the 999,999 *ft* breakpoints. The Mach zero breakpoints are mirrored from the Mach 0.2 data and the Mach 99 breakpoints (infinity points) are hold-last-value with the Mach 6 breakpoints. Likewise, the zero second breakpoints are mirrored with the one-second breakpoints and the 999-second breakpoints (infinity points) are mirrored with the time breakpoints at the end of the tracking phase.

As implied earlier, the gain tables are part of the vehicle's flight software and are used as a lookup table with linear interpolation between breakpoints.

## F. Advantages

There are many advantages to this gain scheduling process, some of which have already been mentioned throughout this paper. While LQR does not guarantee the actual LAS controller is stable and optimal since a simplified plant is used for LQR tuning, it does minimize the effects of dominant modes in the system, state errors in the time response and control usage all in one method. The LQR method also allows the designers to emphasize certain states through the weight selections as needed. In the end, the one designer approach provided a significant time savings. Through automation of the process to decide which design points to retune and the formulation of the linear analysis tools, a designer in the loop is only needed to decide which weights to adjust for meeting the desired design goals. In the latest design cycles, one day was needed for redesigning the LAS controller inner-loop gains for both the pitch and yaw channels. It takes approximately 30 minutes for each Newton iteration. This is significant when considering the interfacing and integration process needed with a multi-designer approach that could absorb days of a schedule.

In generic terms, this tool can be used for other atmospheric vehicles. The only changes required to the rigid body model would be in regards to the control effector. The flexible body model is data driven, which would be supplied with an alternate abort system design. Additionally, the actuator and sensor models can also be easily reconfigured for alternate vehicle needs. And obviously, the controller model would be specific for the control effector type. Larger modifications may be required for the simplified plants used in the LQR tuning method, but the simplified plants would easily be determined by reconfiguring the controller topology for full-state feedback. The gain scheduling tool wraps around all of these components and a starting point for weights and initial table breakpoints are all that are needed. Methods similar to the one discussed in Subsection C for determining the starting weights and in Subsection A for table breakpoints would be suitable for any vehicle or system.

## G. Issues

There are also issues with the gain scheduling process to point out. Since the purpose of this tool is to design gain schedules usable across many different flight conditions for one vehicle, simplifying assumptions are necessary to keep the gain tables in as simple of a form as possible, as a function of time, altitude and Mach number, for example. Essentially, the design points are not physical flight conditions and are created from the best assumptions applicable to the vehicle. This can cause some issues if the simplifications to the design points are not considered carefully. For example, it is safe to schedule the vehicle mass properties with respect to abort time since the large mass changes due to the abort motor burn are the same simulated undispersed in time. However, atmospheric parameters must be carefully considered since they are dependent on one another.

Additionally, in the gain tables, the time zero seconds, zero altitude and Mach zero is completely a factious breakpoint, but necessary for using the table with interpolation. This point is impossible to properly tune though. Therefore, a hold-last-value from the Mach 0.2, altitude 0 *ft* and one-second breakpoints are used. This is a critical point for low altitude aborts that could cause a stable or unstable time response for the initial first second of the abort. Again this flight regime is quickly flown through by the booster rocket and also the LAV once the abort motor is fired, so it is not a driver to controller design even if it causes unstable modes. Care must be taken to monitor the effects of this breakpoint and ensure that short period instabilities are not present in the time response.

Issues with the LQR process can also ensue if the plants are not fully vetted. For example, with the LAV there is a long period unstable roll mode present in some vehicle flight conditions that LQR, by design, will try to find the minimum solution to stabilize this unstable mode causing inconsistencies in the gains. While the unstable roll mode is truly present in LAV flight, this mode is of no consequence to control because it is on a much larger period than the dynamic LAV flight. But LQR will adjust the gains accordingly to stabilize the mode. Thus, the design points had to be biased in pitch angle and pitch rate to remove the long period unstable roll mode from the simplified plants. Similar issues could arise with any vehicle having unstable long period modes. Additionally, any bias introduced must also ensure that the plant is still full rank in order to use the LQR method for tuning.

A consistent step size for estimating the first derivative in Newton's method for LQR tuning has not been found yet. From experience, a delta between 0.5 and 0.25 on the weights seems to provide a good enough estimate of the slope changes in the LQR space. But some instances needing finer step sizes have been identified. This is a place in the process where some time, and automation for that matter, could be lost in tuning.

And finally, there is no exit criteria for the gain scheduling tool. Each iteration is initialized manually by the designer, so iterations can be completed until the designer deems the results acceptable.

## VI. Examples

Examples are presented in this section for tuning a set of breakpoints with the gain scheduling tool, performing a linear assessment with the linear models, which feeds back to gain tuning, and finally assessing the controller performance in the non-linear domain.

## A. Tuning

At the beginning of most design cycles, not all linear margins or design cases are failing. For example, with an update to the aerodynamic database, it would be expected that only the low frequency modes would be impacted and thus, only adjustments are needed to bring *LFGM*, *RBPM* and *BWCL* back into compliance with the design goals. Also, aerodynamic characteristics of the LAV beyond 65,000 *ft* are benign in comparison to denser atmospheric flight, so cases aborting higher than 65,000 *ft* may still be meeting design goals with the previous design cycle's gains. Hence, the metrics to be targeted in the tuning process and the design space has been identified with a simple baseline linear analysis.

An example will be shown here of a case that failed to meet the *RBPM* and closed-loop bandwidth design goals in the pitch channel with the previous design cycle's inner-loop gains. Luckily, both work with each other meaning that by bringing the closed-loop bandwidth back into compliance, the *RBPM* will most likely comply also. From the trend analysis,  $K_\alpha$  can be used to adjust both of these design goals, so the *R* weight is chosen to be iterated on with a 1 Hz closed-loop bandwidth as the target in Newton's equation, Equation 14. Consider the design point at 20,000 *ft*, Mach 0.9 and 1 second into an abort, which typically is an unstable vehicle condition for LAV flight. At the initialization of the design cycle, the design point was failing to meet the *RBPM*, *HFGM* and *BWCL* design goals, as shown in the *Init.* row of Table 3. Using the starting weights, the margins at this breakpoint are improved with only the closed-loop bandwidth, at 1.04 *Hz*, and the *RBPM*, at 29.92 deg, missing their respective design goal by a small margin. Having to update other areas of the design space also, the gain scheduling tool is able to find a more acceptable answer for this particular breakpoint. Initially, the *R* weight is set to 1 from the starting weights. *R* is unknown from the previous design. Table 3 shows each iteration of the gain scheduling tool and the results for all linear margins.

Table 3: Iterative Tuning with the Gain Scheduling Tool

Iteration	R	LFGM	RBPM	HFGM	BMPM	BMGM2	BWCL
Goal	-	$\geq 6$ dB	$\geq 30$ deg	$\geq 6$ dB	$\geq 30$ deg	$\geq 6$ dB	1.0 Hz
Init.	-	23.15 dB	25.15 deg	4.26 dB	56.38 deg	11.81 dB	1.496 Hz
0	1.0	20.13 dB	29.92 deg	6.88 dB	65.91 deg	14.36 dB	1.040 Hz
1	1.1449	19.81 dB	30.83 deg	7.21 dB	67.35 deg	14.59 dB	0.988 Hz
2	0.9643	19.89 dB	30.59 deg	7.12 dB	66.96 deg	14.53 dB	1.003 Hz
3	1.0091	19.87 dB	30.65 deg	7.14 dB	67.06 deg	14.55 dB	0.999 Hz

In the first iteration, the change to *R* is too much and iterations two and three work to bring *BWCL* back to 1 *Hz*. As can be seen in the table, only a change to *R* on the order of one-hundredth was necessary for compliance with all design goals and only three iterations were required. With the entire design space, it can take as much as 30 minutes to compute one iteration per channel. Considering the initial evaluation with the starting weights, approximately two hours were required to tune the complete design in the pitch channel, which is very acceptable.

The developed tuning approach also highlights the sensitivities in transonic abort region experienced by the LAV as these are the cases usually requiring additional iterations. Additionally, the last time point of the tracking phase is also a more difficult breakpoint to tune since performance bandwidth is allowed to decrease in the design goal. The time 2 seconds breakpoints require additional control authority use since this is around the time of peak thrust from the abort motor and thus, higher disturbance moments on the vehicle. In most cases, the *R* weight is significantly smaller at this time point to allow for more control use at low cost to the LQR scheme. From experience, the gains of the yaw channel are typically harder to tune than the pitch channel due to the number of gains and states; there are many weight ratios to keep in check. Typically, it takes longer and more finesse is needed to tune the yaw channel, but the gain scheduling tool has proven completely capable of doing so.

## B. Linear Performance Assessment

With the linear analysis tool incorporated into the gain scheduling tool, a before and after snapshot can be taken of the linear performance at the kickoff of a gain tuning design and at the completion of the tuning

cycle. Example plots of *RBPM*, *HFGM* and closed-loop bandwidth for an abort at Mach 1 are shown in Figure 8. These linear margins were out of compliance with the design goals, shown in black on the plots, at the beginning of the design cycle. With three iterations of the gain tuning tool and incorporating those gains into the linear analysis tool, all linear margins are met for a Mach 1 abort. The closed-loop bandwidth deviates at the 1 second point very slightly from the desired 1 Hz bandwidth during the first 10 seconds of flight. Since bandwidth is only a desired design goal and not a design requirement, this response was deemed acceptable.

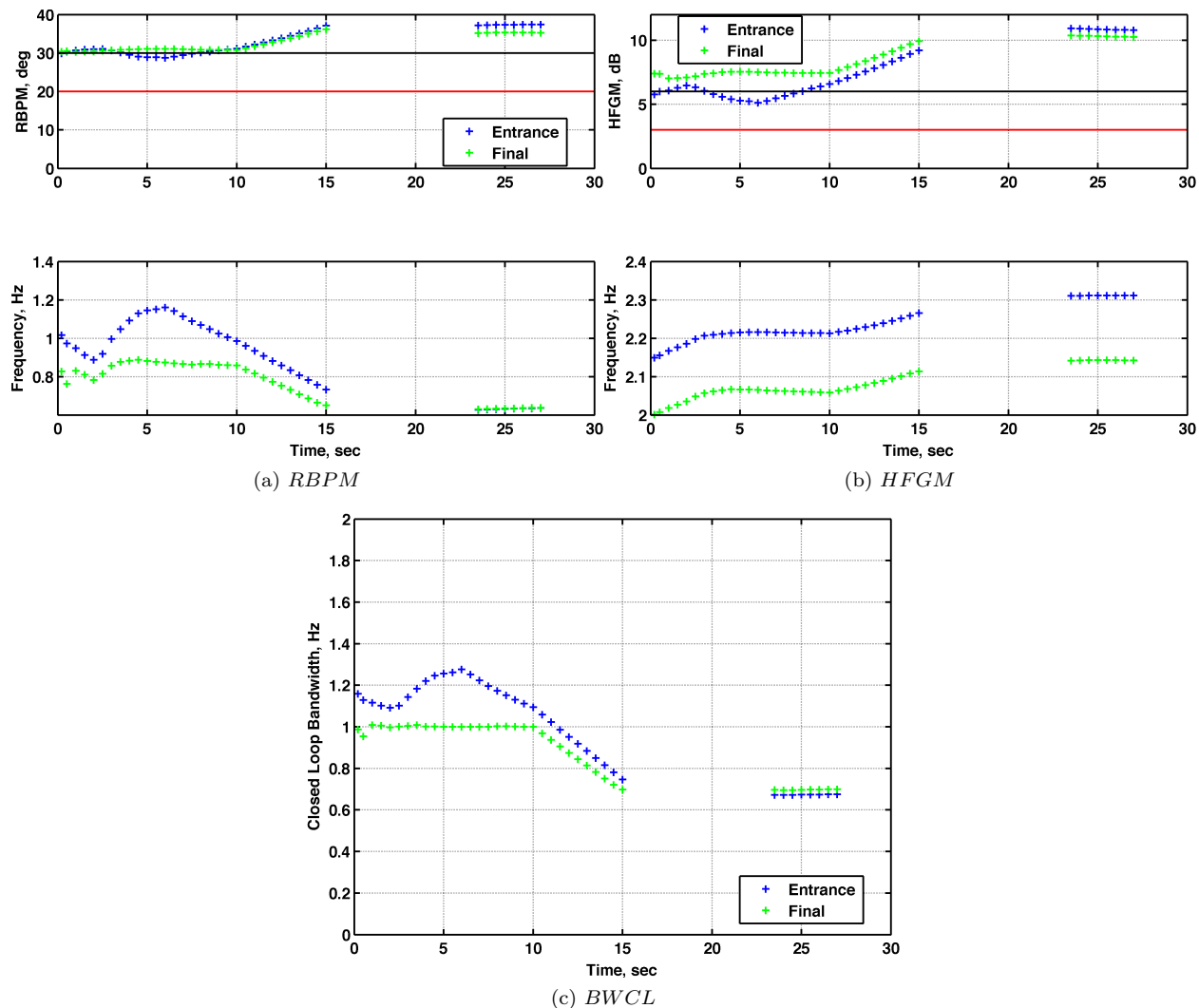


Figure 8: Example of Linear Margin Results in Pitch Channel for LAV Mach 1 Abort Trajectory.

Dispersed linear analysis is used to verify the requirements. There are approximately 40 parameters in the linear models ranging from flex mode shapes to aerodynamic slope characteristics that are dispersed for this analysis. To close out a design cycle, cases that are narrowly achieving the design goal in a critical phase of flight or are nominally missing the design goal are chosen for further consideration with dispersed analysis to ensure the requirements will be met with the new gains. Figure 9 presents the results of a linear dispersed analysis for a pad abort at 0.2 seconds after abort initiation. The green crosshair represents the nominal results and the blue crosshairs are from 100 dispersed cases. This particular point was chosen for further analysis since it interpolates almost exclusively from the altitude 0 ft, Mach 0, time 0 seconds gain table breakpoints where stability issues were discovered as previously stated in this paper. As shown in Figure 9, the requirements, solid red line, are met in the dispersed analysis for design validation. Similar results can also be produced across an entire trajectory. Much computation time is needed for dispersed analysis, so



selective verification is often used in the design phase of the program. Verification cycles with the final design will be completed before the close of vehicle for flight readiness.

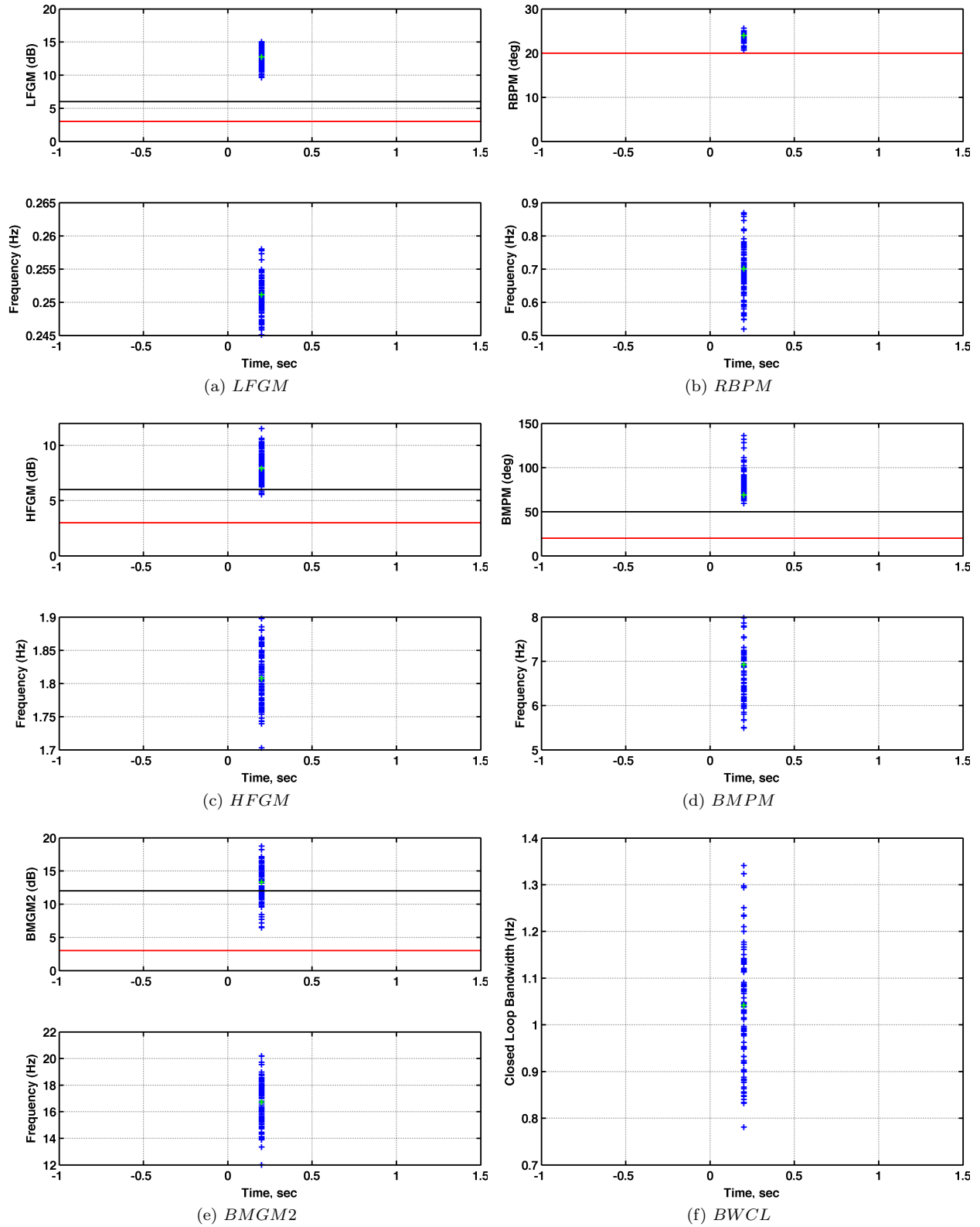


Figure 9: Example of Dispersed Linear Margin Analysis in Pitch Channel for LAV Pad Abort Trajectory.

## C. Non-linear Performance

As a final verification of the design, the gain tables are integrated into a six-DOF simulation and a full envelope analysis is performed to compare the trajectory results. Not only are the gains verified, but changes to the vehicle's flight are noted and compared with the expected response. This particular design cycle, de-gained the tracking phase in order to return stability to the controller and meet linear margin requirements. Additionally, closed-loop bandwidth was reduced across nearly the entire flight regime, which was expected to negatively impact non-linear performance. To verify this expectation, Monte Carlo analysis was performed and showed a slight increase to the loss of controllability metric, especially in the transonic abort region. The fact that more tumbling occurred in the areas of critical aerodynamic effects was expected and deemed acceptable in order to meet linear margin requirements. Since the gain tuning process is rather quick, once it is completed at the initialization of a design cycle, a redesign of the controller or autopilot modes can be examined if simply retuning the gains did not result in acceptable non-linear performance. Then, the gain scheduling tool can be used again at the end to finalize the gains of the new controller design. In the future, design goals on step responses may be beneficial to furthering the controllability of the LAV flight for non-linear performance.

## VII. Conclusion

The gain scheduling tool presented here successfully tunes the LAS controller in both the pitch and yaw channels across the LAS abort flight regime while not only minimizing steady state error in the system, but also schedule cost and required labor to the project. Even cases with an unstable plant were easily tuned with LQR and accounted for in the gain tables. The complexity of the yaw channel and yaw-roll cross feed is more manageable with the tool than with a pole placement method. The creation of the gain scheduling tool has verified the linear models used, created a PID controller compliant with the LQR process and allowed designers direct access to the stability and states of a given controller. In future design cycles, changes to the inner-loop PID gains of the LAS controller will be easily calculated with the gain scheduling tool.

This paper has introduced the LAS controller phases and topology and the linear models used for linear analysis. The creation of the simplified pitch channel and yaw channel plants for use with LQR were discussed. The iterations scheme for tuning the controller with the gain scheduling tool was detailed and demonstrated in an example. And finally, summarized results in linear performance and non-linear performance were also included.

Due to the generic nature of this tool that wraps around a linear model and simplified plant representation of a controller, this tool is usable for any future LAV design or alternate abort vehicle designs. As mentioned previously, the modeling of the control effector in the linear controller and rigid body models would need to be updated appropriately for an alternate vehicle. Modifications to specifically align the simplified plants with the controller topology would be needed for the LQR tuning method. Deriving suitable starting weights and table breakpoints were also discussed in the paper and are directly applicable to any atmospheric flight vehicle.

## Acknowledgments

The team would like to recognize and thank Ephram I. Rubin of Odyssey Space Research, LLC in Houston, Texas for the development and documentation of the flexible body model for the linear analysis tool and help in the creation of the gain scheduling tool.

## References

- <sup>1</sup>Proud, R., et. al., "Orion Guidance and Control Ascent Abort Algorithm Design and Performance Results," *Conference Papers – American Astronomical Society*, Xxxxxxx, XX, Aug. 2009.
- <sup>2</sup>Antoniewicz, R., et. al. "Derivation and Definition of a Linear Aircraft Model," Edwards, CA: NASA Ames Research Center and Dryden Flight Research Facility: National Aeronautics and Space Administration, 1988.