2021

Development of a Robust and Tunable Aircraft Guidance Algorithm

Jacob R. Spangenberg
Wright State University

DEVELOPMENT OF A ROBUST AND TUNABLE AIRCRAFT GUIDANCE ALGORITHM

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mechanical Engineering

by

JACOB R. SPANGENBERG
B.S.M.E., Wright State University, 2019

2021
Wright State University

Wright State University
GRADUATE SCHOOL

January 10, 2022

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY JACOB R. SPANGENBERG ENTITLED Development of a Robust and Tunable Aircraft Guidance Algorithm BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Mechanical Engineering.

Mitch Wolff, Ph.D.
Thesis Director

Raghu Srinivasan, Ph.D., P.E.
Chair, Department of Mechanical &
Materials Engineering

Committee on
Final Examination

Mitch Wolff, Ph.D.

James Menart, Ph.D.

Scott Thomas, Ph.D.

Barry Milligan, Ph.D.
Vice Provost for Academic Affairs
Dean of the Graduate School

ABSTRACT

Spangenberg, Jacob R. M.S.M.E., Department of Mechanical and Materials Engineering, Wright State University, 2021. *Development of a Robust and Tunable Aircraft Guidance Algorithm.*

A set of guidance control laws is developed for application to a reduced order dynamic aircraft model. A feedback control formulation utilizing a linear quadratic regulator (LQR) is developed, together with methods for easing the design burden associated with gain tuning. Metrics are developed to assess the stability margin of the controller over the full flight envelope of a notional unmanned aerial vehicle (UAV) model. A feedforward control path is then added to the architecture. The performance of the guidance control laws is assessed through time domain step response metrics as well as through execution of a design mission. The thesis closes with a discussion of possible improvements regarding gain optimality and run-time performance of the model.

Contents

List of Figures

List of Tables

List of Symbols

Roman Symbols

A	state matrix	
b	wing span	ft
B	control matrix	
C	output state matrix	
C_D	drag coefficient	
C_{D_0}	parasite drag coefficient	
C_{D_V}	minimum drag coefficient	
C_L	lift coefficient	
C_{L_0}	lift coefficient at minimum drag	
$C_{L_{\delta a}}$	non-dimensional roll moment derivative with respect to aileron deflection	$1/rad$
C_{L_α}	lift coefficient derivative with respect to α	$1/rad$
C_{L_p}	non-dimensional roll moment derivative with respect to roll rate	$1/rad$
C_{L_V}	lift coefficient at minimum drag	
C_M	controllability matrix	
D	drag; output control matrix	lb
e	tracking error	
g	local acceleration due to gravity	ft/s^2
G	state space reference command matrix	ft/s^2
h	altitude	ft
I	identity matrix	
I_{xx}	moment of inertia about x axis	$slug \cdot ft^2$
J	optimal control cost functional	
k	induced drag coefficient	
K	gain matrix	
L	lift;roll moment	lb; $ft \cdot lb$
$L_{\delta a}$	roll moment derivative with respect to aileron deflection	$slug \cdot ft/s$
L_p	roll moment derivative with respect to roll rate	$slug \cdot ft/s^2$
$L(s)$	loop transfer matrix	
m	mass	$slug$
M	Mach number	ft
N	state-control interaction weighting matrix	
n_z	load factor	
O_M	observability matrix	
p	roll rate	
P	solution to the algebraic Riccati equation	

Q	state weighting matrix	
$\bar{q}$	dynamic pressure	lb/ft^2
r	reference command	
R	control weighting matrix	
$\vec{r}$	point in a local frame	ft
s	frequency domain	
S	wing area	ft^2
T	thrust	lb
t	time	s
u	control vector	
V	velocity; plant disturbance	ft/s
w	sensor disturbance	
W	weight	lb
x	state vector	

Greek Symbols

α	angle of attack; cost functional design weights	$^\circ$
α_σ	minimum singular value of the return difference matrix	
β_σ	minimum singular value of the stability robustness matrix	
ϵ	compensator integral state	
ω_{cg}	crossover frequency need to find gain margin	rad/s
ω_{cp}	crossover frequency need to find phase margin	rad/s
γ	climb angle	$^\circ$
ϕ	roll angle; characteristic polynomial	$^\circ$
ψ	heading angle	$^\circ$
ρ	air density	$slug/ft^3$
τ_r	roll time constant	s
θ	pitch angle	$^\circ$

List of Acronyms

1DOF	One Degree-of-Freedom
6DOF	Six Degree-of-Freedom
CFD	Computational Fluid Dynamics
GM	Gain Margin
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MIMO	Multi Input Multi Output
NED	North East Down
PM	Phase Margin
PID	Proportional Integral Derivative
SISO	Single Input Single Output
UAV	Unmanned Aerial Vehicle

List of Subscripts and Superscripts

0	initial condition
cl	closed loop
in	inertial frame
max	maximum value
ol	open loop

Acknowledgment

I would like to take this opportunity to extend my thanks to Dr. Mitch Wolff, Dr. James Menart, and Dr. Scott Thomas for their guidance throughout the effort.

Dedicated to

My parents

1. Introduction

The introductory section of the 1965 report *Apollo Guidance and Navigation* [1] opens with the following guidance definition:

> Guidance is the process of collecting and applying information for the purpose of generating maneuver commands to control vehicle movements. In effect, this process represents closure of the essential control feedback branch that has to be associated with structure and propulsion in order for any vehicle system to operate successfully.

Although Draper's report [1] was focused on guidance and navigation of the Apollo spacecraft, his definition is also perfectly suited for this thesis, which concerns development of guidance control algorithms for fixed-wing aircraft. Most often associated with problems governing the gross motion of the aircraft, guidance control is fundamentally concerned with tracking problems. Outer-loop autopilot functions are usually operating at the guidance and navigation level, with inner loops then responsible for the aircraft attitude control. The most basic such guidance functions are probably those associated with maintaining cruise conditions for speed, altitude and heading. Such controllers are regulators principally written to provide disturbance rejection, but can be considered tracking algorithms designed to track constant reference inputs. More generally, guidance controllers are designed to provide vehicle control commands to track a dynamically changing set of input reference signals. Examples include trackers that intercept other aircraft, maintain relative position in formation flight, monitor and track surface vehicles and ships, or provide services for trajectory following, path following and waypoint following.

In this thesis, the focus is on development of guidance algorithms that provide tracking control for the aircraft. The input reference signals are assumed to be in the form of prescribed trajectory data for aircraft speed, altitude and heading. The problems associated with generating such inputs are usually termed *path planning* problems, even though reference trajectories are often the sought-after inputs.[1] Although path planning is an extremely interesting and important discipline – especially with the emergence of machine learning and artificial intelligence capabilities – it is not a focus area for this thesis. We can think of the trajectory data feeding our guidance algorithms as telling us where we want the aircraft to go; the guidance algorithms then determine how the aircraft is to achieve the desired state.

In the following section, the basic loop structure for the guidance control laws is presented, with an emphasis on distinguishing the feedforward and feedback portions of the control signal. The chapter concludes with an overview of the objectives, structure and scope of this thesis.

[1] We adopt here the usual technical distinction between paths and trajectories: paths are (three dimensional) curves connecting two points in space; trajectories are paths that associate with each of its interior points a time at which the aircraft is to be at that point. Velocity specification is therefore implicit with trajectory tracking.

1.1 Guidance Control

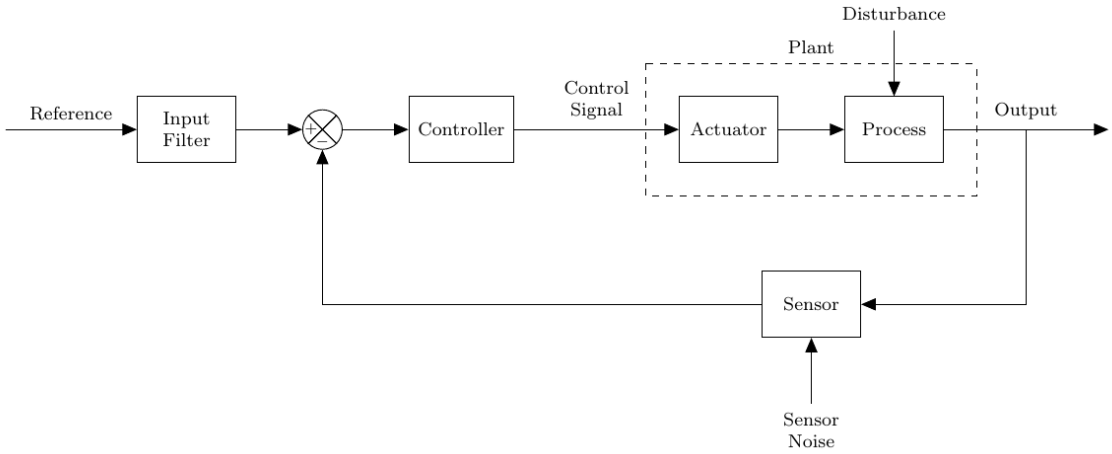


Figure 1.1: Elementary feedback control architecture [2]

Figure 1.1, which has been reproduced from [2], is a top-level representation of a classical closed-loop negative feedback system. The diagram's components, consisting of the input filter, controller, plant model and sensor package, are the four key elements that are to be modeled, characterized or developed for closed-loop control of any dynamic system. With respect to application to aircraft control, the process disturbance could, for example, represent winds aloft or atmospheric gusts and turbulence. Sensor noise will be present in the measurements of critical signals such as the airspeed, angle of attack, sideslip angle, as well as attitude state data along all three body axes. For real-world applications, the controller designs must exhibit sufficient robustness and stability margins to perform in the presence of process disturbances and sensor noise.

The negative feedback system shown in Figure 1.1 is the foundational architecture for closed-loop control. By feeding back the process output and comparing it to the desired reference state, an error signal is created. The controller is designed to drive this error signal to zero, which tends to move the system output towards the desired reference state. One drawback to the negative feedback architecture is that it is a reactive controller, meaning that no anticipatory control action is present unless such dynamics are built into the input

filter. The controller only acts to drive the error signal to zero and therefore must "wait" for non-zero errors to occur before control commands are issued to the plant actuators.[2]

Many systems achieve better response dynamics when the negative feedback control action is complemented by a feedfoward control component. For example, the top-level aircraft control architecture shown in Figure 1.2 [3] exhibits the blending of a feedforward control signal with each of the primary feedback loops. While the feedback controllers operate on error signals, the feedforward controllers operate directly on the incoming reference commands. The feedforward signal therefore provides an open-loop control, which, when properly designed, can carry a significant majority of the overall control action to the system. In such designs, the feedback controller may be contributing relatively minor corrections; nevertheless, they are critically important to maintaining high precision tracking fidelity.

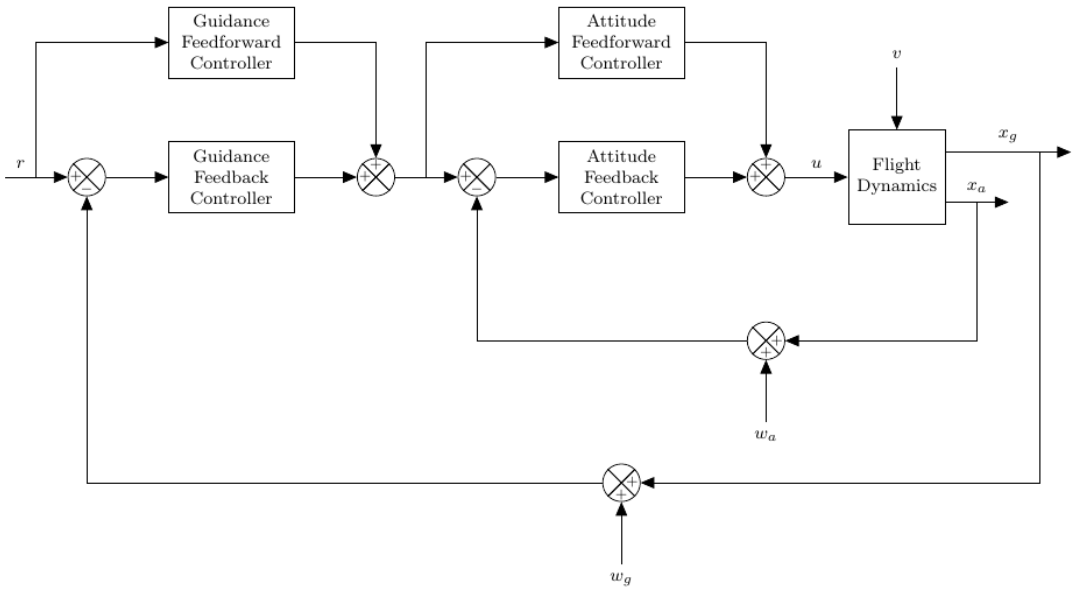


Figure 1.2: Typical aircraft flight control system top-level diagram [3]

In addition to featuring the feedforward control signals typical of most aircraft flight

[2]Derivative control, such as one encounters with classical PID controllers, can improve the system's responsiveness to sudden changes in the reference inputs. However, too much derivative control action can be de-stabilizing, especially when operating in the presence of signal noise.

control systems, Figure 1.2 also illustrates the nested loop topology employed for control of systems with time scale separation. The inner loop, indicated here as the loop controlling the aircraft's attitude dynamics, often executes at a higher frame rate – typically an order of magnitude faster than the outer loop. This is consistent with the higher frequency dynamics of the aircraft's rotational response compared to its linear (translational) response. The inner feedback circuit is conventionally referred to as the aircraft's control loop and the outer feedback circuit is the guidance loop.

The reference input signal r in Figure 1.2, is aircraft state data in the form of the aircraft's desired position and linear velocity. For a conventional six degree-of-freedom aircraft system, the outputs of the guidance loop consist of commanded attitude angles and angular rates. The aircraft control laws are then designed to produce a set of commands for the engine(s) and control surfaces to achieve the desired attitude state. Development of the inner-loop control laws is a non-trivial task. Traditional linear control approaches require knowledge of the aircraft's stability and control derivatives, which characterize it's dynamic response to control inputs. The derivatives are dependent on the aerodynamic and inertial properties of the aircraft, which in turn are dependent on the flight condition and gross weight. Due to the (nonlinear) dependence of the stability and control derivatives on the operating condition, it is usually the case that gain scheduling is required to achieve the desired flight dynamics throughout the aircraft flight envelope. Gain scheduling refers to the process of tuning the control gains at a collection of operating points in the flight envelope and then interpolating the gains as a function of, say, gross weight and dynamic pressure at all off-design flight conditions. The design point gain tuning process can itself be a complex, labor-intensive task. Guaranteeing robust and stable closed-loop performance over the full flight envelope is an additional challenge.

There are many applications where the fidelity of a full 6DOF aircraft model is not required. Indeed there are many applications where 6DOF aircraft modeling is not desired. Putting aside the heavy burden associated with assembling and verifying the plant and con-

troller models, a 6DOF model executes much more slowly than a point mass model. This is not just due to the higher number of integrators, but to the higher frequency dynamics associated with its moment equations of motion. Resolving the (closed-loop) aircraft attitude dynamics imposes an upper bound on the integrator time-step – generally it is the short period pitch dynamics that can become numerically unstable if the integration time-step is too large. Point-mass aircraft models, by contrast, are far easier to create in terms of plant modeling and can execute far more rapidly for constructive simulation studies.

Constructive simulation generally refers to modeling and simulation for faster than real-time execution. Such simulation efforts include a very large number of applications. In the context of aircraft simulation, three important examples include:

Campaign modeling: Focuses on quantifying the effectiveness of an Air Force in accomplishing a strategic objective. Campaign models often include a large number of interacting, dynamic components, such as an ownship, other friendly aircraft, enemy aircraft, communication and support systems, weapons systems, etc. Since the number of model elements is high, and the number of dynamic simulation scenarios can reach well into the thousands, it is important to balance the run-time performance of the models against the required fidelity. Often it is the case that an aircraft model whose performance is limited by purely kinematic constraints is not adequate, but that a full 6DOF fidelity is not needed.

Optimization and design studies: Multi-disciplinary aircraft design optimization often requires assembly of the gradient of an objective function with respect to its constituent design variables. Assembly of the gradient may require time domain simulation to evaluate the functional. Again, the trade-off between run-time performance and model fidelity is a key element, as the optimization process will not scale well if the cycle time per iteration gets too large.

Aircraft subsystem studies: Time domain modeling of aircraft subsystems, such as for

electrical power and thermal management, is strongly dependent on accurately modeling the engine states as a function of flight condition. A point mass model can reasonably approximate the required net thrust (or power) without the overhead associated with 6DOF aircraft modeling.

Development of an air vehicle model that enables constructive simulation activities such as these is the focus of this thesis. In 2017, Shimmin [4] formulated equations of motion for a point mass air vehicle model, together with an outer-loop guidance control based on dynamic inversion (feedback linearization). In 2019, Brendlinger [5] expanded the applicability of the model by developing additional guidance modes; they included a trajectory tracking mode, a waypoint following mode and a mode that allowed for arbitrary step input commands in speed, altitude and heading. For the trajectory tracking mode, a PI feedback control was implemented to assure tight tracking performance. While straightforward in design and implementation, introduction of the feedback controller raised the issue of gain tuning, as well as potential gain scheduling as discussed earlier.

The ideal guidance controller for the reduced order (point mass) air vehicle model is one that *requires* no user tuning for suitable closed-loop performance over the full flight envelope, but also one that *allows* user tuning if specific closed-loop performance metrics are to be met. In this context, the possibility of employing a linear quadratic regulator (LQR) formulation for the closed-loop feedback suggests itself. LQR control provides a number of very desirable features for this application, including:

- The gain matrix resulting from an LQR formulation is static; no additional control states are created, thus the order of the closed-loop system is not increased.
- Loop closures for multi-input multi-output (MIMO) systems such as this one are managed simultaneously.
- The theoretical foundation for LQR control is very well established and solution algorithms are widely available.

- LQR control guarantees a 60° phase margin; this is a very attractive feature from a stability and robustness perspective.

The typical downsides to implementing LQR control are twofold. First, LQR requires full state feedback. In many real-world applications, full state knowledge is not possible, generally leading to state estimation through Kalman filtering. When dynamic state estimators are implemented, one loses the advantage of LQR control not increasing the system order. Second, the designer of an LQR control implementation must specify matrix elements in its quadratic cost functional. Selecting these elements is tantamount to gain tuning traditional PID controllers. As the order of the system increases, the number of design parameters in the cost functional increases geometrically. Thus, while preferable to gain tuning nested sets of PID controllers, LQR design can still be a complex and iterative process.

As the primary application of this model is to support simulation activities, the concern regarding full state knowledge becomes moot. Within the simulation environment we have full knowledge of all states, including both the plant states as well as any compensator states built in the guidance controller. Thus, the primary challenge is to develop a methodology for selecting matrix elements in the LQR cost functional. This is essentially the LQR gain tuning problem. Ideally, an automated tuning process could be developed. If such an algorithm were implemented, the user could simply input the plant data needed to define the performance characteristics of the aircraft being modeled. The tuning algorithm would then automatically derive the LQR gain matrix and implement it in the guidance feedback controller. Short of realizing a fully automatic tuning methodology, the goal would be to 1) reduce the number elements needed to generate the LQR gain matrix, and 2) to provide the designer with physical insight into the meaning and effect of each design parameter.

1.2 Goals and Objectives

The top-level goal is to provide a point mass air vehicle model that is well suited for the types of constructive simulation studies described earlier. Developing and implementing a robust and tunable guidance controller is the primary technical need in realizing this objective, and is therefore the focus of this thesis.

By comparison to Figure 1.2, the simplified architecture depicted below more accurately represents the model framework for the system of interest. External disturbances and sensor noise are not a priority and have therefore been dropped from the diagram. The guidance controller will consist of both feedforward (open loop) and feedback components, with the feedback portion consisting of an LQR tracking formulation.

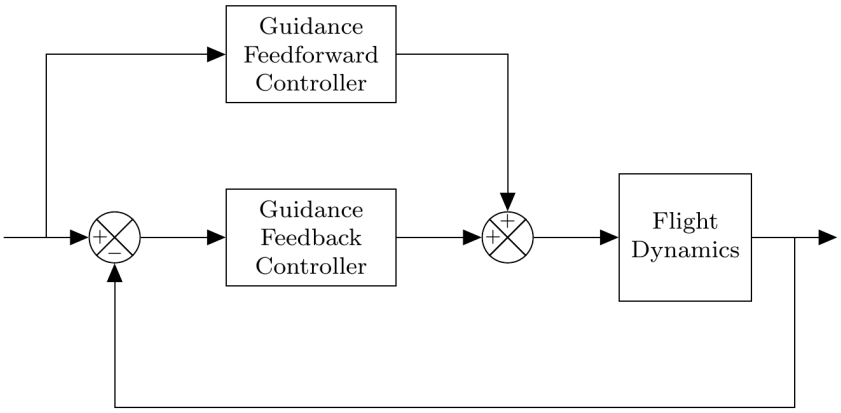


Figure 1.3: Top-level guidance control architecture

The framework for developing the formulation and presenting the results of this thesis is as follows:

Chapter 2 provides a brief review of the fundamentals of LQR tracking control and presents an example application in the form of a simplified aircraft roll dynamics problem. A traditional PID formulation is also presented. This example case is intended to demonstrate the superior performance of the LQR controller at off-design points. A methodology for measuring gain and phase margins for MIMO systems is also demonstrated; it is later

used to quantify the closed-loop system performance of the point mass air vehicle model.

Chapter 3 provides the equations of motion for the point mass aircraft model, together with the approximate plant equations from which expressions for the feedforward guidance terms are derived. A sample air vehicle model is also introduced as a case study for presenting simulation results.

Chapter 4 then provides the results of the thesis. The full LQR state space model is developed; this includes linearization of the vehicle equations of motion as well as inclusion of compensator states. A methodology for dramatically reducing the number of LQR design parameters appearing in the cost functional is then presented. Controller tuning on the reduced set of design gains is demonstrated, and the system stability over the full flight envelope is investigated. A methodology for including the feedforward control action is then developed and demonstrated. The combined feedforward/feedback control architecture is suggested as a means by which to reduce the control effort required in the feedback loop. This is an important consideration in determining whether gain scheduling is required to achieve acceptable closed-loop performance throughout the flight envelope. Finally, the analytic expressions developed for the state space model enable automatic re-evaluation of the LQR gain matrix without the need to manually adjust the design parameters. While not a fully automated process, the design burden is significantly reduced as compared to traditional guidance control algorithms.

Chapter 5 concludes the thesis with a summary of important results and ideas for future improvements.

2. Linear Quadratic Regulator

Optimal control theory includes two separate but complementary methodologies. The first is a dynamic programming approach; in practice this often leads to formulation of the Hamilton-Jacobi-Bellman equation, for which a general solution does not exist. The second is a direct functional minimization approach through application of the calculus of variations. Classic textbooks on the subject such as Athans and Falb [6] and Bryson and Ho [7] develop and explain both approaches, along with example problems illustrating their relative advantages. There is an important class of problems, however, where either approach is adequate; namely, for linear systems subject to optimization of a quadratic cost functional. These problems are classified as linear quadratic regulator (LQR) problems. Kirk [8] provides solutions to the LQR problem using both the dynamic programming approach and the variational calculus approach and notes that "...in linear regulator problems all routes lead to the same destination." For LQR problems, that destination is the algebraic Riccati equation.

As there are many exhaustive treatments of the LQR optimal control problem, including both the development and solution of the Riccati equation, this chapter does not recapitulate those results. Instead, the basic framework of the LQR formulation is summarized and a review of existing work in the area of LQR for guidance control is provided. The chapter then introduces an algorithm for quantifying the controller robustness for MIMO systems. The LQR design methodology, including problem formulation, gain tuning and closed-loop performance assessment is then presented in the context of an example problem

– roll control for a fixed wing aircraft. The example provides an opportunity to compare and contrast the LQR and PID design methodologies, as well as illustrating the variation in plant dynamics that one should expect for air vehicle models operating at different flight points.

2.1 LQR Control Policy

The linear quadratic regulator control methodology requires that the plant dynamics and its output variables can be expressed as a linear, time invariant system of the form

$$\dot{x} = Ax + Bu \tag{2.1}$$

$$y = Cx + Du \tag{2.2}$$

These equations are known, respectively, as the state and output equations. State space form satisfies the LTI requirement because the equations consist of the state x and control u vectors multiplied by constants. LQR uses the state space model as a constraint on its goal of minimizing its cost functional J, which is written

$$J = \int_0^\infty (x^T Q x + u^T R u + x^T N u) dt \tag{2.3}$$

where $Q \geq 0$ and $R > 0$ are symmetric matrices. The symmetric matrix Q must be positive semi-definite, meaning the number $x^T Q x$ must be non-negative for every real column vector x. The symmetric matrix R must be positive-definite, meaning the number $u^T R u$ must be positive for every real column vector u. The LQR control problem also requires that the state space model is controllable and the controlled outputs are observable. The

system is controllable if the controllability matrix

$$C_M = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{2.4}$$

has full row rank, where n is the number of states modeled. The observability condition is satisfied if the observability matrix

$$O_M = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{2.5}$$

has full row rank. Again, n is the number of states included in the state space model. With these conditions satisfied, the LQR controller will be able to control the desired states while minimizing the cost function J. The cost function contains three terms to penalize different parts of the system. The first term x^TQx penalizes a non-zero state vector. The second term penalizes a non-zero control vector. The last term penalizes the interaction between the state and control vectors. This means the control system design must determine the balance between penalizing states and controls using the Q, N, and R weighting matrices. Once a selection has been made, the algebraic Riccati equation is used to determine the optimal control policy K that is used to minimize the cost function.

Solution of the algebraic Riccati equation satisfies the necessary and sufficient conditions for minimization of the cost functional, equation 2.3. The Riccati equation can be written in the following form

$$A^TP + PA - (PB + N)\, R^{-1} \left(B^TP + N^T\right) + Q = 0 \tag{2.6}$$

where one notes that in addition to the state and control matrices A and B as well as the weighting matrices Q, N, and R, the Riccati equation includes an unknown matrix P. If a P can be computed such that it solves the Riccati equation, then we have minimized the quadratic cost functional via the optimal control policy:

$$u = -R^{-1}\left(B^T P + N^T\right) x = -Kx \tag{2.7}$$

Carrying out the solution of the Riccati equation and computing the LQR optimal gain matrix K is accomplished by utilizing MATLAB's numerical **LQR** algorithm [9].

Looking at dimensionality, the gain matrix will have columns equal to the number of states and rows equal to the number of controls. The optimal control law $u = -Kx$ is used to close the feedback loop on the provided LTI state space model.

2.2 Frequency Domain Robustness

The control system designer is often interested in the gain and phase margin, also known as stability margins, of a closed-loop system to understand its stability properties. One of the principal results of linear systems theory is that the output of a SISO LTI system is a combination of gain and phase imparted on the input signal. Gain is the change in amplitude of the signal and phase is the change in phase shift of the signal. The gain and phase of the system's outputs generally change with the frequency of the input signal. Bode diagrams are a standard way to present the gain and phase of a system over the entire frequency range of interest.

The gain and phase *margins* represent the amount of gain and phase available before the system reaches the instability point. To illustrate the point, consider the unity gain negative feedback system depicted in Figure 2.1.

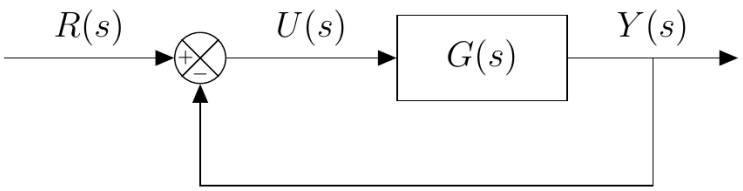


Figure 2.1: Closed-loop negative feedback architecture

The closed-loop transfer function can be written

$$\frac{Y(s)}{R(s)} = \frac{U(s)G(s)}{U(s) + U(s)G(s)} = \frac{G(s)}{1 + G(s)} \tag{2.8}$$

A characteristic of unstable systems is infinite gain, which is produced in the closed-loop system when the denominator of the transfer function goes to zero. For the simple negative feedback system shown in Figure 2.1, the open-loop transfer function must be $G(s) = -1$ to create an unstable system. The gain of this open-loop system is one (0 decibels) since the input is not scaled. The phase of the open-loop system is -180 degrees since the sign of the input signal is flipped. Therefore, the point 0 dB gain and -180 degrees phase causes the entire system to be unstable, even if the point exists at only one frequency.

Figure 2.2 [9] is an example Bode diagram provided by the Matlab **Bode** documentation that also shows measurement of gain and phase margin.

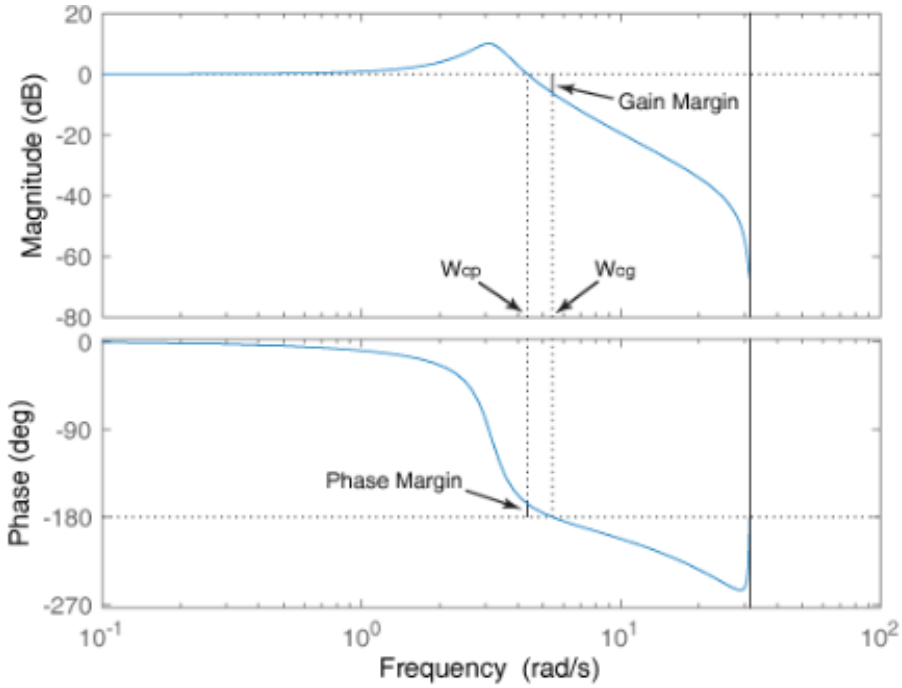


Figure 2.2: Example Bode diagram noting gain and phase margin [9]

The crossover frequencies ω_{cp} and ω_{cg} denote the frequencies at which the open-loop system reaches 0 dB gain and -180 degrees phase, respectively. This means the gain margin is the margin from 0 dB when the system phase is -180 degrees. The phase margin is the margin from -180 degrees when the system gain is 0 dB. Systems with lower stability margins are considered less stable because small changes in the open-loop system can cause instability. From a modeling and controls perspective, higher phase margin means greater robustness in the face of inevitable discrepancies between the plant model and a real-world system.

2.2.1 MIMO Systems

The stability margins for SISO systems are easily obtained by analyzing Bode diagrams. But, the process gets more complicated for multi-input multi-output (MIMO) systems because the single transfer function becomes a matrix of transfer functions. This is one of the many reasons that a state space formulation is preferred for MIMO systems: the state ma-

trix eigendata can provide insight into the stability of the system regardless of the system dimensions.

The state space model of a closed-loop negative feedback can be converted to a transfer function using the Laplace Transform [10]. The following equation is the resulting loop transfer matrix, which is the loop transfer function expressed in matrix form for a MIMO system.

$$L(s) = K(sI - A)^{-1}B \tag{2.9}$$

The term $(sI - A)^{-1}$ is the state transition matrix. This term is used to describe the state vector once the Laplace Transform is applied to the model, $\dot{x} = Ax + Bu$. With this term defined, the loop transfer matrix looks very similar to the state feedback control law, $u = Kx$.

The loop transfer matrix is an important indicator of the relationship between the closed-loop system and the open-loop system. Using matrix determinant identities, Latrevsky shows the following relationship between the open-loop and closed-loop characteristic polynomials [11].

$$\det\left[I + L(s)\right] = \frac{\phi_{cl}}{\phi_{ol}} \tag{2.10}$$

The quantity within the brackets is known as the return difference matrix. This is also used to relate the plant input to the controller output to show the effects of the closed-loop controller on the input signal. Previous sections showed that instability is directly associated with an undefined denominator of the closed-loop transfer function for a SISO system. Latrevsky uses the return difference matrix and the stability robustness matrix $(I + L^{-1})$ to evaluate the stability of MIMO systems through characterization of modeling errors as additive and multiplicative uncertainties. The system is said to be unstable when these matrices approach singularity. Singular value decomposition is used to determine the near singularity of the matrices of interest. Ridgley and Banda have also investigated MIMO stability robustness and have provided an instructive guide to singular value decomposition

[12]. They state that a square, complex matrix can be put in the form

$$M = U\Sigma V^H \tag{2.11}$$

where U and V are unitary matrices and V^H refers to the complex conjugate transpose. A unitary matrix is a square complex matrix where its inverse is equal to its conjugate transpose. The conjugate transpose is comprised of the transpose of a matrix, followed by the complex conjugate (flipping the sign of the imaginary component). The matrix Σ is a diagonal matrix that arranges the singular values of M in descending order along the diagonal. The minimum singular value indicates the near singularity of the matrix. A value closer to zero means M is closer to being a singular matrix. Singular value decomposition is very useful for the stability robustness problem. Latrevsky uses this operation to find two important indicators of the stability margins of MIMO systems [11].

$$\alpha_\sigma = \min_\omega \underline{\sigma}\left(I + L\right) \tag{2.12}$$

$$\beta_\sigma = \min_\omega \underline{\sigma}\left(I + L^{-1}\right) \tag{2.13}$$

The two above equations represent the minimum singular value of the return difference and stability robustness matrices across the frequency spectrum. These values are then used to compute gain margin and phase margin for each characterization of modeling uncertainty.

$$GM_{I+L} = \left[\frac{1}{1+\alpha_\sigma}, \frac{1}{1-\alpha_\sigma}\right] \tag{2.14}$$

$$GM_{I+L^{-1}} = [1-\beta_\sigma, 1+\beta_\sigma] \tag{2.15}$$

$$PM_{I+L} = \pm 2\sin^{-1}\frac{\alpha_\sigma}{2} \tag{2.16}$$

$$PM_{I+L^{-1}} = \pm 2\sin^{-1}\frac{\beta_\sigma}{2} \tag{2.17}$$

The gain margin extracted for the SISO system from the Bode plot represents the amount that the system gain may be increased without destabilizing the system. The MIMO gain margin that is output from this method contains two values: upper and lower gain margin. These values represent the limit on system gain without destabilizing the system, where the negative number represents the limit on system gain reduction.

The independent gain and phase margin for MIMO systems is represented by the union of the two modeling uncertainty characterizations.

$$GM = GM_{I+L} \cup GM_{I+L^{-1}}, \quad PM = PM_{I+L} \cup PM_{I+L^{-1}} \tag{2.18}$$

The independent gain margin for MIMO systems can be understood as the amount of gain that each feedback loop can independently vary without causing the system to become unstable, while the phase is held constant. Independent phase margins are the amount of phase that each feedback loop can vary without destabilizing the system, holding gain constant. The best achievable gain and phase margins occur when $\alpha_\sigma = \beta_\sigma = 1$, stating that the matrices of interest are strictly non-singular. For this scenario, the highest achievable gain and phase margin are $+\infty$ and ± 60 degrees, respectively. This method is widely used in the multivariable control field to capture stability margins [12], [13]. It also utilized throughout this thesis to make claims about the stability robustness of LQR closed-loop systems.

2.3 Time Domain Performance Metrics

The performance of a closed-loop system responsible for tracking a reference command is measured by examining the response of the system to a non-zero error signal. A natural example of this performance characterization is to measure the system response to a step input. Common design goals are to create a controller that exhibits sufficient rise time,

reduces unnecessary overshoot, and eliminates steady-state error. These metrics are time domain step response performance metrics used to design closed-loop systems. There exists a function in Matlab, *stepinfo*, that extracts time domain step response metrics from time history data of the commanded and actual values. The data extracted by Matlab is shown graphically in Figure 2.3 [9].

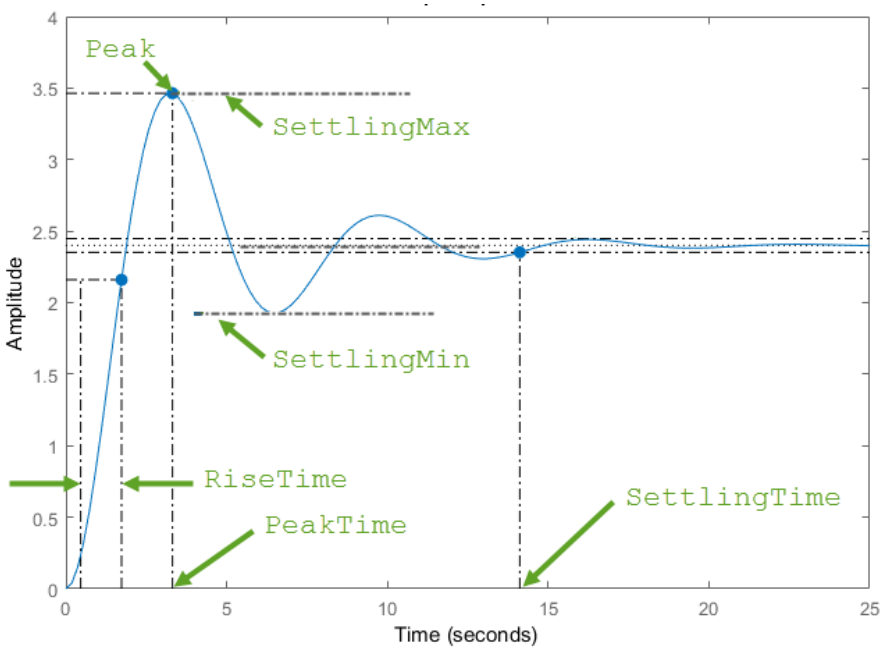


Figure 2.3: Time domain step response metrics from Matlab [9]

Rise time, peak time, settling time, and overshoot percentage give the user sufficient data to analyze the system response. Rise time is the time required for the response to move from 10% to 90% of the steady-state value. Peak time is the time associated with the maximum value of the response. Settling time is the time for $e = |y - y_{ss}|$ to fall below 2% of the steady-state value. Overshoot percentage is the percent difference that the response surpasses the steady-state value. The two settling extrema represent the bounds within which the system settles from the peak value to the steady-state value. The time domain performance metrics provided by Matlab's *stepinfo* function give the control system designer different ways to quantify the performance of the system.

2.4 Roll Dynamics: A Single DOF Model

Aircraft are often represented by a six degree of freedom (6DOF) model comprised of three translational and three rotational degrees of freedom. Attitude control is governed by aerodynamic moments, which are resolved about the aircraft body axis system. A simplified example of this model can be used to demonstrate the process required to design an LQR closed-loop negative feedback control law. For this example, pure rolling motion will be examined which has only a single degree of freedom. Kinematic equations from a 6DOF model express the aircraft roll rate as a function of Euler rates as

$$p = \dot{\phi} - \dot{\psi} \sin \gamma \tag{2.19}$$

The second term in the equation represents effects from a climbing turn, which is not possible in the simplified roll model. Thus, for a model of pure rolling motion, the aircraft roll rate is

$$p = \dot{\phi} \tag{2.20}$$

This equation can be integrated with respect to time to determine the aircraft bank angle. With the kinematics of the model established, the dynamics are now modeled, as shown in Figure 2.4 [11]. Ailerons are typically deflected in coordination with the rudder to change the aircraft heading. For a pure roll, the ailerons are deflected in a differential manner to affect the aircraft lift distribution and induce a non-zero roll moment L.

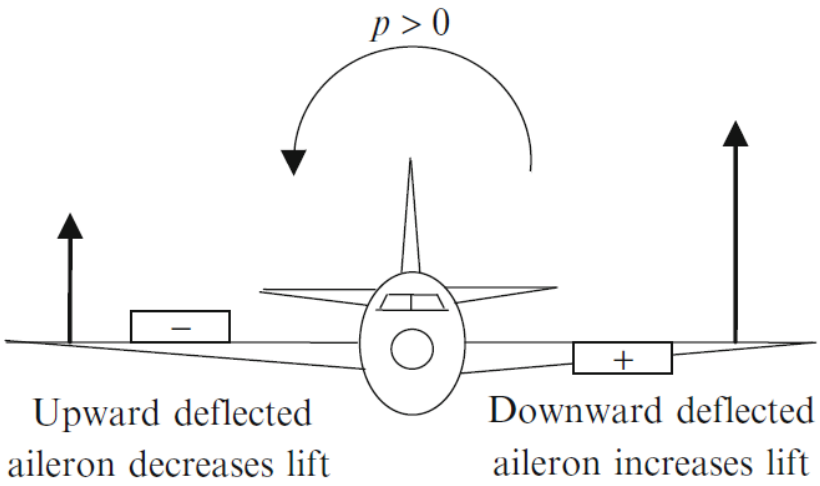


Figure 2.4: Roll dynamics diagram [11]

A summation of roll moments is used to create an equation of motion for the problem shown in Figure 2.4. The equation of motion for pure rolling motion is

$$I_{xx}\dot{p} = L_p p + L_{\delta_a}\delta_a \tag{2.21}$$

where I_{xx} is the resistance to rotation about the x axis due to aircraft roll inertia, L_p is the dimensional roll moment derivative with respect to roll rate, and L_{δ_a} is the dimensional roll moment derivative with respect to differential aileron deflection. Although L_p has the potential to be positive on a stalled wing, it is always assumed negative in this model, meaning it is the roll rate damping term. An important indicator of the rolling response is the time constant of the system given by

$$\tau_r = -\frac{I_{xx}}{L_p} \tag{2.22}$$

As L_p is always taken to be a negative number, tau_r is a positive time constant measured in seconds. The time constant indicates the speed at which the system will settle to its steady-state roll rate after being disturbed. This can be thought of as the amount of damp-

ing present in the system. A small time constant results from a heavily damped system, conversely, a large value indicates a lightly damped system.

Data obtained from Nelson for the roll dynamic problem is displayed in Table 2.1 for a collection of aircraft at multiple flight conditions [14]. The collection consists of a general aviation aircraft (NAVION), two fighters (F104-A and A4-D), one business jet (Jetstar), and two transport jets (Convair 880 and Boeing 747).

	W (lb)	I_{xx} (slug-ft^2)	S (ft^2)	b (ft)	Altitude (ft)	Mach	C_{L_p} (rad)$^{-1}$	$C_{L_{\delta_a}}$ (rad)$^{-1}$	τ_r (sec)
NAVION	2,750	1,048	184	33.4	0	0.158	-0.410	-0.134	0.119
F104-A	16,300	3,549	196	21.9	0	0.257	-0.285	0.039	0.774
	16,300	3,549	196	21.9	55,000	1.80	-0.270	0.017	1.123
A4-D	17,578	8,090	260	27.5	0	0.40	-0.260	0.08	0.597
	17,578	8,090	260	27.5	35,000	0.80	-0.240	0.072	1.196
Jetstar	38,200	118,773	542.5	53.75	0	0.20	-0.370	0.054	1.544
	38,200	118,773	542.5	53.75	40,000	0.80	-0.420	0.06	1.593
Convair 880	155,000	1,510,000	2,000	120.0	0	0.25	-0.381	-0.038	0.830
	155,000	1,510,000	2,000	120.0	35,000	0.80	-0.312	-0.050	1.172
Boeing 747	636,600	18,200,000	5,500	195.7	0	0.25	-0.450	0.0461	1.158
	636,600	18,200,000	5,500	195.7	40,000	0.90	-0.300	0.014	2.260

Table 2.1: Important roll dynamics parameters for a collection of aircraft flight conditions

The data presented shows the change in non-dimensional derivatives and roll time constant as a function of flight condition. Mass properties of the aircraft are held constant, so the changes in time constant are affected by airspeed and the non-zero density gradient. Roll time constant, shown in the rightmost column, nearly doubles for some of the aircraft at the cruise condition. The change would be even greater if the mass were variable due to fuel burn. This demonstrates the need for a controller that provides good performance at a wide variety of flight conditions.

The non-dimensional data from the table must be converted to dimensional form to be used in the roll dynamics equation of motion. The dimensional roll moment with respect to roll rate is

$$L_p = \frac{\bar{q}Sb^2C_{L_p}}{2V} \tag{2.23}$$

where $\bar{q}$ refers to the dynamic pressure, b is the span of the wings, and S is the wing

reference area. Likewise, the dimensional roll moment with respect to differential aileron deflection is

$$L_{\delta_a} = C_{L_{\delta a}} \bar{q} S b \tag{2.24}$$

The aircraft selected for analysis is the A4-D fighter. It exhibits large changes in roll time constant between flight conditions and has a much smaller moment of inertia meaning that it will require smaller controller gains to track a reference signal. The top and side views of the A-4D are shown in Figure 2.5.

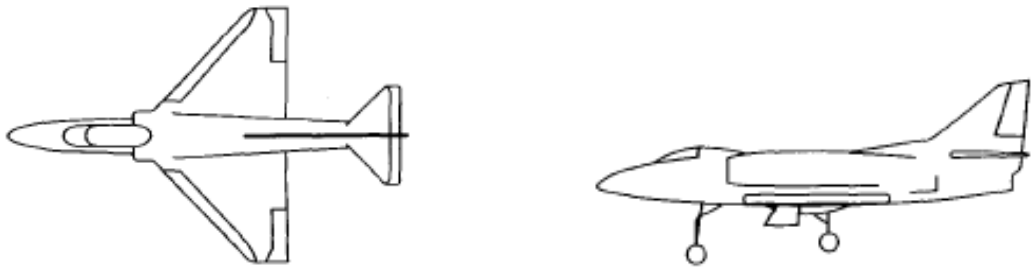


Figure 2.5: Top and side view of A-4D fighter [14]

The A4-D fighter is a single engine, subsonic fighter jet with a similar weight to the General Dynamics F-16. The pure rolling motion model uses aircraft data for the A4-D and the equation of motion presented previously. Typical trajectory data does not include roll rate as an option for the closed-loop controller. Therefore, a single equation of motion describing roll rate is not sufficient. Equation 2.20 can be included in the model to kinematically relate aircraft bank angle to roll rate, a parameter which is sometimes included in trajectory data to direct the shape of a turn. This set of equations can be put into state space form,

$$\dot{x} = Ax + Bu \tag{2.25}$$

using linearization about an equilibrium point. Luckily, the equations of motion for the single degree of freedom model are both linear first order ODEs, making the conversion to state space form trivial. The equilibrium point is also zero roll rate, so the state space

model includes the total states, not a perturbation from the equilibrium. The state space form of the roll dynamics problem is

$$\begin{Bmatrix} \dot{\phi} \\ \dot{p} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & L_p \end{bmatrix} \begin{Bmatrix} \phi \\ p \end{Bmatrix} + \begin{bmatrix} 0 \\ L_{\delta_A} \end{bmatrix} \begin{Bmatrix} \delta_A \end{Bmatrix} \tag{2.26}$$

As discussed previously, trajectory data sometimes includes bank angle to direct the aircraft to fly a turn in a specific way. It makes sense to design the model to track a reference command in bank angle using a closed-loop controller. The following sections will demonstrate traditional PID control design and LQR control design.

2.4.1 PID Design

PID control is a common choice for a closed-loop negative feedback controller. Proportional, integral, and derivative terms are easily understood by engineers and do not require expertise in control theory. The classical PID architecture in parallel form is

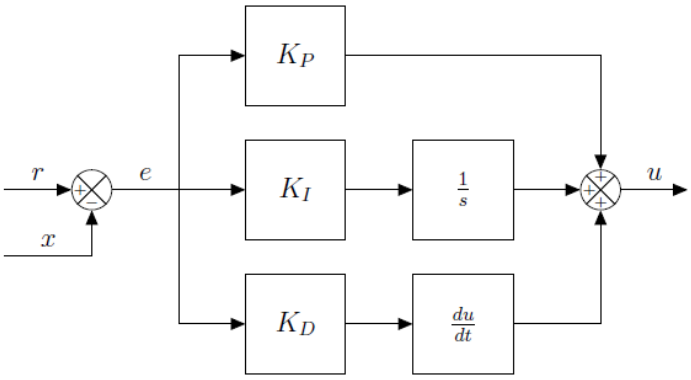


Figure 2.6: PID control architecture

All three gains work on the tracking error signal, which is bank angle error for this model. The proportional term K_P typically makes the controller respond faster to a non-zero error signal, but will not eliminate steady-state error for a type zero system. The

integral term K_I eliminates the steady-state error, but can make the system oscillatory, or even unstable if set too high. The derivative term K_D serves to reduce overshoot. A general understanding of the closed-loop response as the PID gains are increased can be obtained from the following table.

	Rise Time	**Overshoot**	**Settling Time**	**Steady-State Error**
K_P	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Decrease
K_D	Small Change	Decrease	Decrease	No Change

Table 2.2: Change in time domain step response metrics for increased PID gains

The rules presented in Table 2.2 [15] are not true for every system, but they serve as a general guide to aid the control system designer. PID controllers are typically designed with time domain step response metrics in mind. A set of gains is selected as the baseline, and then a trial-and-error approach is used to capture the desired step response.

2.4.2 LQR Design

LQR controllers in their most basic form are used to regulate all the state variables in the model to zero. The design goal for this roll dynamics model is to track a reference command in bank angle. The bank angle state should be driven to the reference value instead of zero. From knowledge of PID controllers, it is known that integrators are required in the controller to eliminate steady state error. The number of integrators required in the controller is a function of the type of signal to be tracked. Figure 2.7 shows an LQR controller that requires three integrators to eliminate steady-state tracking error from the system.

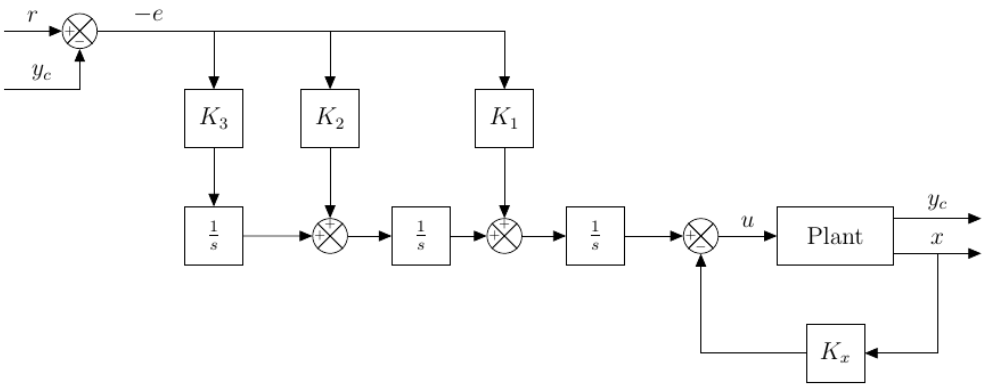


Figure 2.7: LQR control architecture

The block diagram shows the integral control component that exists to eliminate steady-state tracking error. It also shows a state feedback component that is derived from the basic form of the LQR controller. The state feedback component works to drive the states to the equilibrium point (zero in this model). This distributed effort works to change the state of interest to track the reference command while regulating the other states to zero. Applying this to the roll dynamics model results in bank angle tracking a reference command while roll rate is driven to zero.

For type zero systems, the number of integrators required to track the reference signal with zero steady-state error is directly tied to the nature of the reference signal. A constant command such as a step command requires one integrator. A ramp input requires two integrators. The system shown in Figure 2.7 must have a signal with a non-zero second derivative since three integrators are required. The reference signal that drives the roll dynamics model is designed to track only step command for simplicity. Therefore, the LQR controller only requires one integrator and one error equation.

The error equation that must be added to the state space model to form a signal to be integrated to drive the closed-loop LQR controller is

$$\dot{\epsilon}_\phi = \phi - r_\phi \tag{2.27}$$

where ϵ_ϕ is the integrator state used by the LQR gain matrix. A closed-loop state space model is of the form

$$\dot{x} = Ax + Bu + Gr \tag{2.28}$$

to include the reference signal in the model. Including the new error equation in the previously derived state space model gives

$$\begin{Bmatrix} \dot{\epsilon}_\phi \\ \dot{\phi} \\ \dot{p} \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & L_p \end{bmatrix} \begin{Bmatrix} \epsilon_\phi \\ \phi \\ p \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ L_{\delta_A} \end{bmatrix} \left\{ \delta_A \right\} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \left\{ r_\phi \right\} \tag{2.29}$$

Matlab's LQR function uses the A and B state space matrices to solve the continuous time algebraic Riccati equation, with a set of user selected weighting matrices Q, N, and R. The selection of these matrices is made simple for this roll dynamics model. The N matrix is set to zero and R is set to one. The Q matrix, representing state weighting, is only non-zero in the first element. This element is used to tune the weight on the bank angle error integrator state to meet the design goals of the closed-loop system. The output gain matrix of Matlab's LQR function K is a [1x3] matrix. The first element is used to weight the integral action of the tracking loop. The last two elements are the state feedback component used to keep the closed-loop system stable. The step response performance of the two closed-loop systems is discussed in the following section.

2.4.3 Step Response Performance

The controllers developed in the two preceding sections are tuned and evaluated using step inputs in the reference command. Design goals for this system are limited to achieving a rise time of one second at the design flight condition. Data for the A4-D fighter aircraft is given at two flight conditions in Table 2.1. The takeoff flight condition at sea-level and Mach 0.4 is used as the design flight condition. The off-design flight condition, Mach

0.8 and 35000 feet, is used to investigate controller performance across the flight envelope of the aircraft. Each closed-loop system is subjected to a five degree step input in the reference bank angle command. The simulation results shown in this section are achieved using Matlab/Simulink. A Simulink model of the 1DOF roll dynamics model has been created. The step response data provided utilizes a 0.01 second time step using a fixed step integration scheme. The first controller investigated is the PID controller.

PID

The PID design methodology that was discussed in this chapter was used to tune the controller responsible for tracking a step input in bank angle. The controller takes the input error signal in bank angle and outputs a control input to the plant in the form of differential aileron deflection. Figure 2.8 shows the tracking history and control effort required to eliminate the steady-state error introduced with the step command.

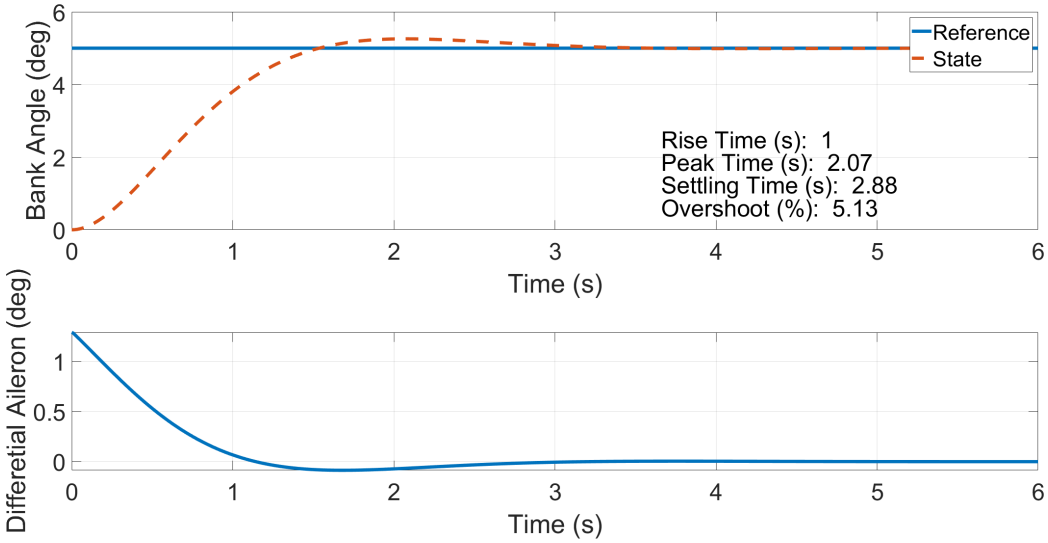


Figure 2.8: Step response and control effort of the PID controller at the design flight condition: $M = 0.4$, $h = 0$ feet

The closed-loop response satisfies the time domain design goal. The rise time of the signal equals one second and the response overshoots the steady-state values by 5%. To determine the effects of flight condition on the closed-loop response, the flight condition

is moved from the takeoff values to the cruise values. The control effort plot shows an instantaneous differential aileron deflection at the onset of the step input. This could be potentially destabilize the system if large enough step commands are input to the controller. Figure 2.9 shows the effects of flight condition on the roll dynamics model using a PID control strategy.

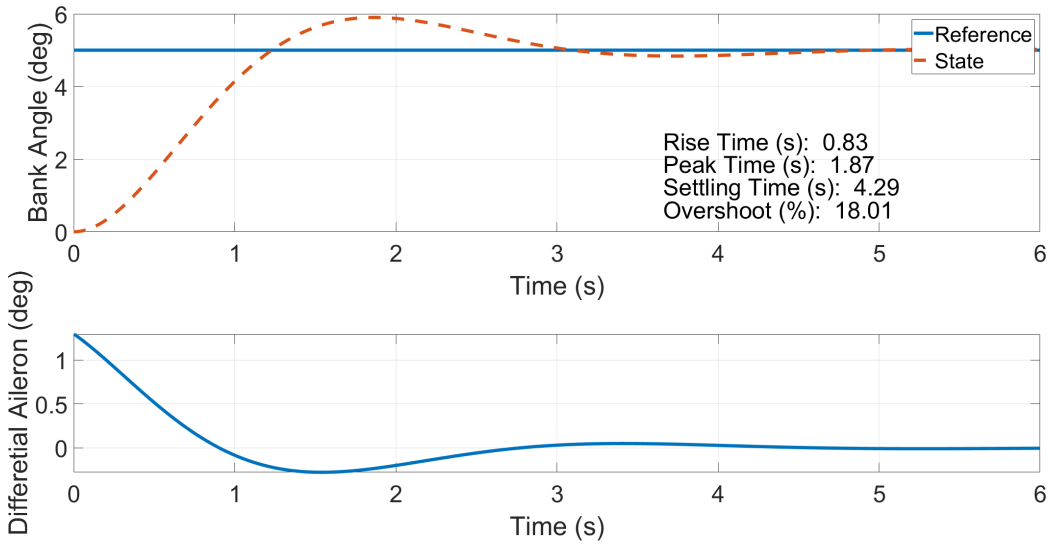


Figure 2.9: Step response of the PID controller at the off-design flight condition: $M = 0.8$, $h = 35000$ feet

The closed-loop response at the off-design flight condition is significantly different than Figure 2.8. The rise time is reduced by nearly 20% meaning the system is more sensitive to the change in reference command. With the faster response comes a greater overshoot, which is shown to more than triple. It is probably not acceptable for a trajectory tracking controller to overshoot the reference command to this degree. This change in performance can be traced back to the non-dimensional roll moment parameters since the aircraft weight and geometry data is constant. The roll time constant nearly doubles at the off-design condition, which means less damping in the system and a more oscillatory response. The LQR controller is subjected to the same step input test as the PID system.

LQR

The LQR controller was designed with the same rise time goal in mind. Figure 2.10 shows the tracking history and control effort required to eliminate the steady-state error introduced with the step command.

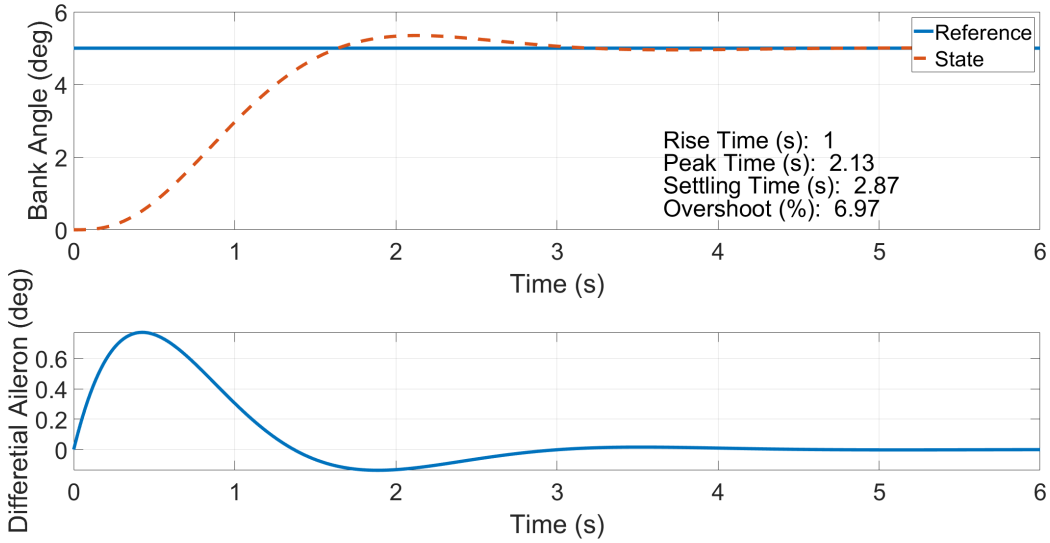


Figure 2.10: Step response and control effort of the LQR controller at the design flight condition: $M = 0.4$, $h = 0$ feet

The step response using the LQR controller satisfies the rise time design goal. The other time domain performance metrics are similar to the PID system, with overshoot percentage having the largest difference. Another difference in the step response is the shape of the control effort plot. The LQR controller naturally drives the state and control vectors to zero. Integrated bank angle error is the only component used to track the reference command. This architecture matches the response as the control effort is shown to ramp up from zero instead of instantaneously deflecting the ailerons. Figure 2.11 shows the change in step response with the off-design flight condition.

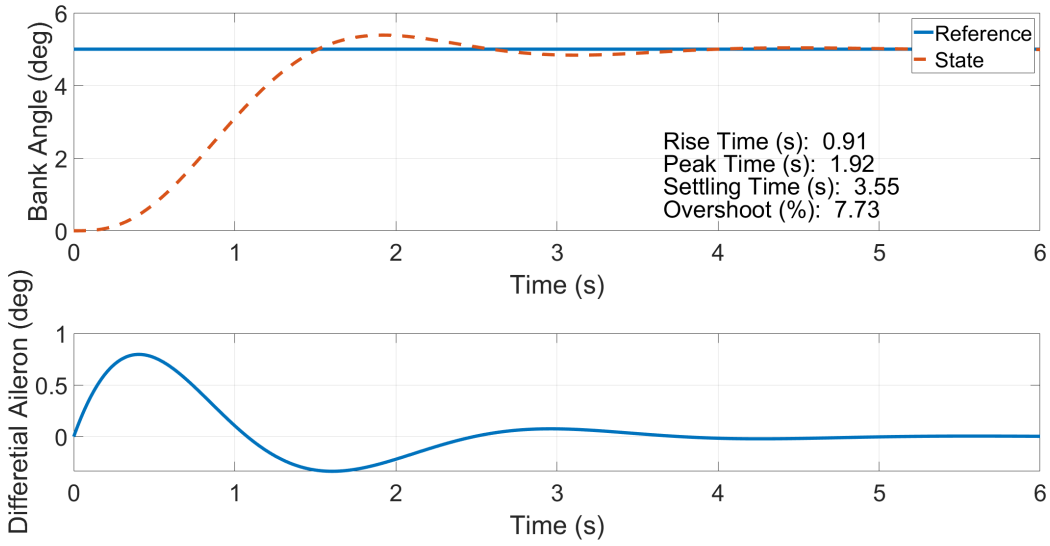


Figure 2.11: Step response of the LQR controller at the off-design flight condition: $M = 0.8$, $h = 35000$ feet

Like the PID controller, the closed-loop LQR system shows a higher sensitivity to the reference command and a more oscillatory response which is dictated by the physics that make up the roll time constant. The main difference from PID control is the magnitude of the changes between flight conditions. The PID controller shows an overshoot increase from 5% to 18%. The LQR controller shows a significantly smaller change of 7% to 8%. The large increase in oscillatory nature for the PID controller has a potential to destabilize the system. This problem is typically addressed using gain scheduling, where the control system designer chooses multiple design points to tune the system. The system then interpolates a table of gains depending on flight condition. This can become a laborious process for an aircraft with a large flight envelope or a multi degree of freedom model with coupled dynamic equations. This process has the potential to be avoided using LQR control, which is shown to have greater performance robustness.

2.4.4 Stability Analysis

Stability robustness data can be extracted from the LQR closed-loop model using the MIMO singular value methods discussed in Section 2.2. Table 2.3 shows the gain and

phase margin for the closed-loop system at the design point that the gain matrix was constructed and at the off-design point.

	Gain Margin (dB)	**Phase Margin (deg)**
Design Point	$[-19.0, +\infty]$	60.0
Off-design Point	$[-20.6, 29.6]$	57.8

Table 2.3: LQR closed-loop stability margins

The stability margins at the design point show infinite positive gain margin and 60 degrees phase margin. These are identical to the theoretical guaranteed stability margins for LQR controllers. Highlighting upper gain margin, the system gain can be multiplied by any value greater than one and remain stable. The closed-loop system is very stable at the design point. While lower than the theoretical guaranteed values, the stability margins are still large at the off-design point with gain margin $[-20.6, 29.6]$. The system gain can be increased significantly before the system destabilizes. This analysis shown here for a simple MIMO system can also be applied to a more complex system such as a reduced order aircraft model.

3. Reduced Order Air Vehicle Modeling

Aircraft performance assessments based on point mass modeling of the air vehicle have a history that dates back to the very earliest days of aviation [16]. Questions concerning the aircraft's maximum rate of climb, speed and altitude for best range and endurance, minimum turning radius, distances required for take-off and landing, and minimum time to climb are all examples of performance metrics that can be estimated with point mass analyses. To do so, the concept of the aircraft drag polar is of central importance. It is through the drag polar that we estimate the power required to sustain a particular flight condition or maneuver. Then with an estimate of the power available – through a model of the (installed) propulsion system – we can quantify the instantaneous *excess power*. At a very fundamental level, it is an aircraft's maximum aerodynamic lift capacity and its excess power margin that determine its flight performance.

The reduced order air vehicle model described in this chapter has been developed to require the same level of model input data as are required for traditional point mass performance models. The capabilities and limitations of the model are discussed, and the formulation of the feedforward guidance commands is presented. In the final sections of the chapter an example aircraft model is presented, along with an example design mission. These are leveraged through the remainder of the thesis to demonstrate the guidance control laws that are developed.

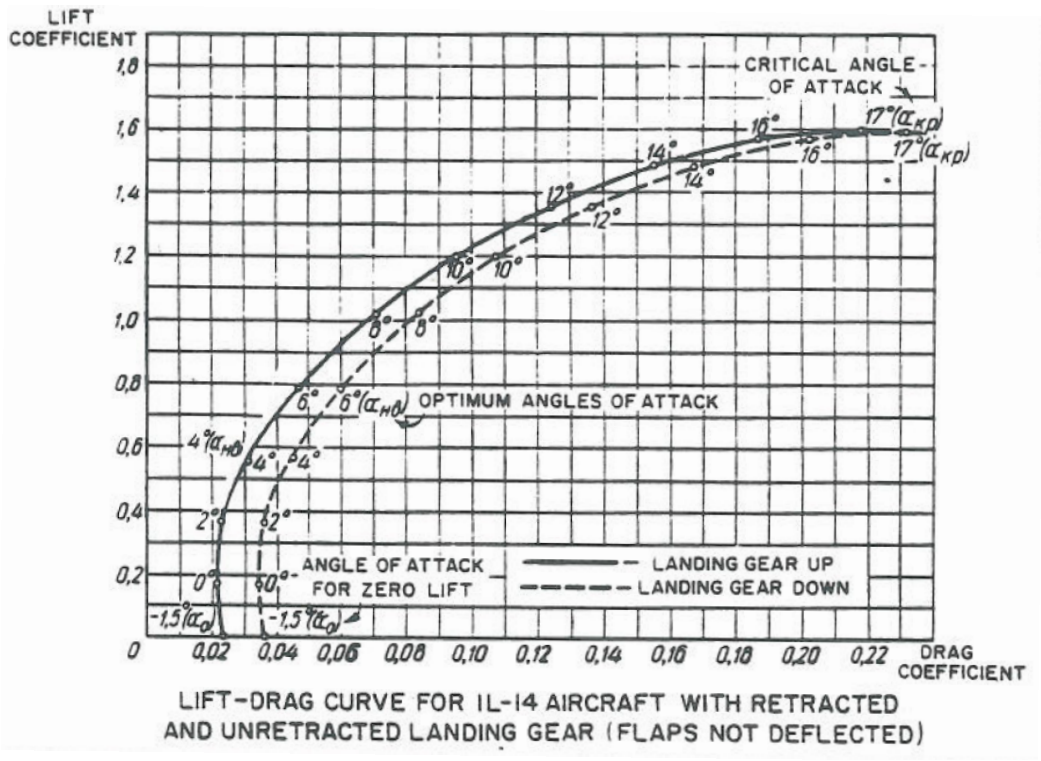
3.1 Model Requirements

Any aircraft performance model requires inputs to characterize three fundamental entities: its mass properties, its propulsion, and its lift and drag. Regarding the mass properties, moments and products of inertia are not required as there are no moment equations associated with a point mass model. Gross weight is the only relevant mass property.

Propulsion models spanning a wide range of complexity and capability can be integrated with the air vehicle model. Of paramount importance, however, is to constrain the maximum and minimum net thrust as a function of speed and altitude. These are the limits that define the max or min power available, and hence limit the excess power margins. Most often, the propulsion system is characterized by providing static tabular engine data. Commonly, a so-called five-column engine table is utilized, consisting of three independent input variables and two dependent outputs. The inputs are the Mach number, altitude and engine throttle setting, and the outputs are the net thrust and the fuel burn rate.[1] If latencies associated with the engine dynamics are of engineering interest, the engine tables can be accompanied by filters. Often a first-order lag is adequate. In principle, higher order filters can be used, or even high fidelity dynamic engine models can be directly integrated. However, the computational overhead of including higher fidelity propulsion models is often at odds with the intended use of the point mass air vehicle model.

The aerodynamic characteristics of the aircraft are represented through just two equations: a parabolic drag polar and a linear lift curve. By way of illustration, data for the Ilyushin IL-14, a cold war era Russian cargo aircraft, is presented. Figure 3.1 is a reproduction of the drag polar image from a 1957 article in *Aviation Week* [17].

[1] For prop-driven aircraft, power, not thrust, is desired output.



LIFT-DRAG CURVE FOR IL-14 AIRCRAFT WITH RETRACTED AND UNRETRACTED LANDING GEAR (FLAPS NOT DEFLECTED)

Figure 3.1: IL-14 drag polar data [17]

The IL-14 drag polar depicted in Figure 3.1 is a standard representation, with the dependent variable C_D plotted along the horizontal axis. Data for two configurations is shown: gear-up and gear-down. For each configuration, data points along the curves are annotated with the angle of attack at which the (C_L, C_D) pair has been measured. Many standard aircraft performance formulations assume that the parabolic lift-drag relationship can be written

$$C_D = C_{D_0} + k\,C_L^2 \tag{3.1}$$

In such a case, the two coefficients C_{D_0} and k define the curve for a particular aircraft configuration. This implies that C_{D_0} is the zero-lift drag on the aircraft. However, as evident from Figure 3.1, it is not necessarily the case that the minimum drag occurs at a zero-lift configuration. Therefore, a slight modification of equation 3.1 is often made so that the vertex of the parabola can be moved off the zero-lift axis. With this modification,

we write

$$C_D = C_{D_V} + k\,(C_L - C_{L_V})^2 \tag{3.2}$$

so that now there are three parameters needed to define the drag polar: C_{D_V}, C_{L_V} and k. The point (C_{L_V}, C_{D_V}) represents the minimum drag flight condition for the aircraft. The subscript "V" is used to indicate that this is the vertex of the parabola. This notation is adopted to avoid confusing the lift at zero angle of attack, C_{L_0}, with the lift for minimum drag.

Determination of the drag polar coefficients is best accomplished through a simple curve-fitting exercise. This requires that a set of (C_L, C_D) data points is available – there are a number of ways that such data can be acquired. Best, of course, is flight test data if one is modeling an existing aircraft. Short of this, CFD methods can be employed. A series of CFD runs can be executed to estimate the lift and drag coefficients at various angles of attack, and the drag polar coefficients are then easily identified. Short of this, empirical and semi-empirical methods can be employed. Many aircraft design books [18],[19],[20],[21], provide methodologies for build-up of drag polar coefficients. For a fully empirical approach, digital DATCOM [22] is probably the industry standard. MATLAB/Simulink, for example, includes tools for directly importing data files output from digital DATCOM. As with any of these methods for assembling the lift-drag data or for building up estimates of the coefficients directly, the key point is that it must represent the net lift and drag for the entire aircraft.

It must be noted that compressibility effects require that a *family* of drag polar coefficients are needed for aircraft that can fly faster than Mach 0.3. Anderson [20], for example, provides case studies illustrating the variation of the drag polar coefficients as as function of Mach number. If the aircraft is capable of supersonic flight, wave drag dominates and the shape of the curve is significantly "flattened" relative to subsonic flight. Nevertheless, parabolic representations of the lift-drag relationship remain valid, and in practice it is a simple matter to implement 1-D vectors of drag polar coefficients that are dependent on

Mach number.

A linear lift curve expressing the lift coefficient as a function of angle of attack is also required. Extracting points from the IL-14 data shown on Figure 3.1, we can produce Figure 3.2, where the linear curve-fit excludes the two data points for which $\alpha > 14°$. The

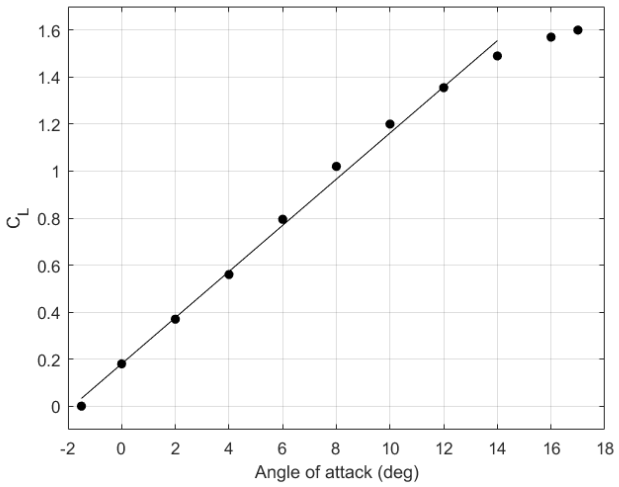


Figure 3.2: IL-14 lift curve data

standard linear lift curve is specified in terms of its lift at zero angle of attack and the lift curve slope, C_{L_α}:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha \tag{3.3}$$

Here again, the coefficients C_{L_0} and C_{L_α} must be representative of the whole aircraft and must generally be scheduled as a function of Mach number. For the lift curve slope the Prandtl-Glauert compressibility correction is often employed, though higher order corrections or CFD data can be used if operation in the transonic regime is an important flight condition for the air vehicle of interest. Post-stall aerodynamics is not intended to be a part of the air vehicle modeling.

3.2 Equations of Motion

Derivation of the point mass equations of motion proceeds along the lines traditionally used to develop rigid-body 6 degree-of-freedom equations of motion [21], [23]. This includes defining coordinate rotations to orient a set of aircraft body-axis coordinates in inertial space, expressing the local angular velocity vector in terms of Euler rates, and supplementing the force equations with kinematic expressions that resolve aircraft motion back onto the inertial reference frame. However, there are two notable departures and simplifications: 1) the climb angle γ takes the place of the pitch angle θ in the Euler sequence, and 2) there are no moment equations!

The derivation follows very closely the work by Weitz with the MITRE corporation [24]. That model, however, does not include resolution of the angle of attack. For applications of the model that require low flight speeds – such as for terminal area operations, during take-off and final approach – the inclusion of angle of attack is important for constraining the max lift capability.

3.2.1 Reference Frames

A traditional "North-East-Down" (NED) reference frame is introduced as the inertial coordinate frame for the purpose of applying Newton's 2^{nd} law. The NED coordinates are a right-handed Cartesian system whose origin is located on the earth's surface, directly below the aircraft. Thus the north-east plane is tangent to the earth's surface, with the "down" axis pointing to the center of the earth. Adopting an NED reference frame as our inertial coordinate system means that accelerations due to the earth's rotation are not considered in this model. This is a standard assumption for a large number of aircraft 6DOF models.

Denoting unit vectors along the NED axes as $\hat{n}$, $\hat{e}$, $\hat{d}$, and employing the traditional aerospace euler sequence (yaw rotation, then pitch rotation, then roll rotation), we arrive at equation 3.4. Note, however, that the climb angle γ is used in place of the pitch angle. This

choice is motivated by the fact that for guidance level control it is more natural to directly model the climb angle than an attitude angle.

$$\begin{Bmatrix} \hat{i}_3 \\ \hat{j}_3 \\ \hat{k}_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{n} \\ \hat{e} \\ \hat{d} \end{Bmatrix}$$
$$= \begin{bmatrix} c\gamma\, c\psi & c\gamma\, s\psi & -s\gamma \\ -c\phi\, s\psi + s\phi\, s\gamma\, c\psi & c\phi\, c\psi + s\phi\, s\gamma\, s\psi & s\phi\, c\gamma \\ s\phi\, s\psi + c\phi\, s\gamma\, c\psi & -s\phi\, c\psi + c\phi\, s\gamma\, s\psi & c\phi\, c\gamma \end{bmatrix} \begin{Bmatrix} \hat{n} \\ \hat{e} \\ \hat{d} \end{Bmatrix} \tag{3.4}$$

Consistent with the use of γ instead of θ, the unit vectors $\hat{i}_3,\ \hat{j}_3,\ \hat{k}_3$ in equation 3.4 represent a traditional aircraft stability axis system. This is because the $\hat{i}_3$ unit vector is positive in the direction of the aircraft velocity. By contrast, for 6DOF models, the aerospace convention is to define a set of body axes with unit vectors $\hat{i}_b,\ \hat{j}_b,\ \hat{k}_b$ such that the origin is at the aircraft cg, with $\hat{i}_b$ pointing forward toward the nose, $\hat{j}_b$ pointing to the pilot's right, and $\hat{k}_b$ pointing downward to complete the right-handed triad. The key difference between body axes and stability axes is that the nose of the aircraft is not generally aligned with its instantaneous velocity. For this model, sideslip is always taken to be zero; therefore, the angle separating the vectors $\hat{i}_b$ and $\hat{i}_3$ is the angle of attack. [2] Of course for a point mass model, there is no "nose" of the aircraft. However, by resolving the angle of attack in this way we can align the net thrust vector along the nominal longitudinal axis of the aircraft instead of simply making it co-linear to the drag vector.

We note that as the product of three rotation matrices, the final transformation matrix in equation 3.4 is orthogonal. Therefore, $A^{-1} = A^T$ or

[2] This assumes that $\hat{i}_b$ is on the aircraft zero-lift line. This is not always true, but correcting the geometry is simple in principle if zero-lift line data is available.

$$\begin{Bmatrix} \hat{n} \\ \hat{e} \\ \hat{d} \end{Bmatrix} = \begin{bmatrix} c\gamma c\psi & -c\phi s\psi + s\phi\, s\gamma c\psi & s\phi s\psi + c\phi s\gamma c\psi \\ c\gamma s\psi & c\phi c\psi + s\phi s\gamma s\psi & -s\phi c\psi + c\phi s\gamma s\psi \\ -s\gamma & s\phi c\gamma & c\phi c\gamma \end{bmatrix} \begin{Bmatrix} \hat{i}_3 \\ \hat{j}_3 \\ \hat{k}_3 \end{Bmatrix} \tag{3.5}$$

3.2.2 Kinematics

The position vector to the aircraft in NED coordinates is written

$$\vec{r} = x\hat{n} + y\hat{e} + z\hat{d} \tag{3.6}$$

with associated inertial velocity

$$\vec{v} = \dot{x}\hat{n} + \dot{y}\hat{e} + \dot{z}\hat{d} \tag{3.7}$$

With equation 3.5, resolving the aircraft velocity $V\,\hat{i}_3$ in the NED coordinate frame is straightforward. The first column of the transformation matrix provides our three first-order kinematic expressions:

$$\dot{x} = V\cos\gamma\cos\psi \tag{3.8a}$$

$$\dot{y} = V\cos\gamma\sin\psi \tag{3.8b}$$

$$\dot{z} = -V\sin\gamma \tag{3.8c}$$

Writing the point mass equations of motion requires knowledge of the inertial acceleration $\dot{\vec{v}}_{in}$ resolved onto the $\hat{i}_3$, $\hat{j}_3$, $\hat{k}_3$ coordinate frame. This is accomplished through an

application of the transport theorem for differentiating rotating vectors. Therefore we write

$$\dot{\vec{v}}_{in} = \dot{\vec{V}} + \vec{\omega} \times \vec{V} \tag{3.9}$$

where the angular velocity vector $\vec{\omega}$ is built up by resolving the euler rates onto the local $\hat{i}_3$, $\hat{j}_3$, $\hat{k}_3$ axes. Thus,

$$\vec{\omega} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \left(\begin{bmatrix} 0 \\ \dot{\gamma} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \tag{3.10}$$

which can be expanded as

$$\begin{aligned} \vec{\omega} = &\left(\dot{\phi} - \dot{\psi}\sin\gamma\right)\hat{i}_3 + \left(\dot{\gamma}\cos\phi + \dot{\psi}\cos\gamma\sin\phi\right)\hat{j}_3 \\ &+ \left(-\dot{\gamma}\sin\phi + \dot{\psi}\cos\gamma\cos\phi\right)\hat{k}_3 \end{aligned} \tag{3.11}$$

Substituting equation 3.11 into equation 3.9 and performing the indicated differentiation and cross product, we find

$$\dot{\vec{v}}_{in} = \dot{V}\hat{i}_3 + V\left(-\dot{\gamma}\sin\phi + \dot{\psi}\cos\gamma\cos\phi\right)\hat{j}_3 - V\left(\dot{\gamma}\cos\phi + \dot{\psi}\cos\gamma\sin\phi\right)\hat{k}_3 \tag{3.12}$$

The components of equation 3.12 are the needed kinematic expressions for application of Newton's law in the local aircraft stability axes.

3.2.3 Dynamics

Figures 3.3 and 3.4 provide the force diagrams for applying Newton's law. We observe in Figure 3.3 that the thrust and drag vectors are not co-linear and that the aircraft pitch angle,

though not included as an explicit state variable in the model, can be constructed via

$$\theta = \gamma + \alpha \tag{3.13}$$

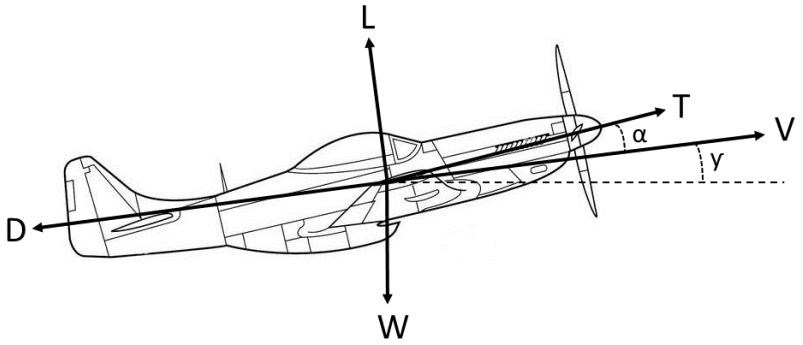


Figure 3.3: Side view free body diagram for air vehicle equations of motion

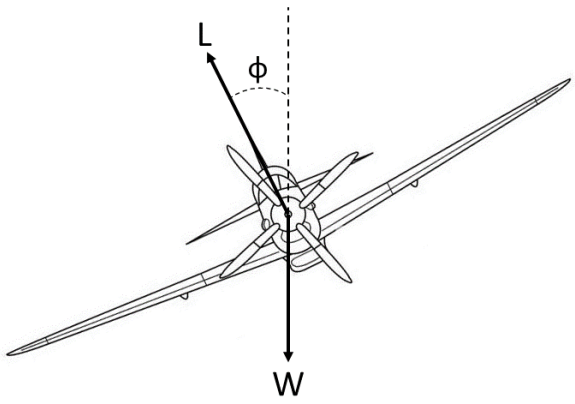


Figure 3.4: Front view free body diagram for air vehicle equations of motion

Components of the forces depicted in Figures 3.3 and 3.4 can be collected to show

$$\begin{aligned}\sum \vec{F} = &\left(T\cos\alpha - D - mg\sin\gamma\right)\hat{i}_3 + \left(mg\cos\gamma\sin\phi\right)\hat{j}_3 \\ &+ \left(mg\cos\gamma\cos\phi - L - T\sin\alpha\right)\hat{k}_3\end{aligned} \tag{3.14}$$

Together with the acceleration components from equation 3.12, application of Newton's 2^{nd} law provides equations of motion resolved into the local $\hat{i}_3$, $\hat{j}_3$, $\hat{k}_3$ axes. These are equations 3.15.

$$m\dot{V} = T\cos\alpha - D - mg\sin\gamma \tag{3.15a}$$

$$mV\left(-\dot{\gamma}\sin\phi + \dot{\psi}\cos\gamma\cos\phi\right) = mg\cos\gamma\sin\phi \tag{3.15b}$$

$$mV\left(\dot{\gamma}\cos\phi + \dot{\psi}\cos\gamma\sin\phi\right) = L + T\sin\alpha - mg\cos\gamma\cos\phi \tag{3.15c}$$

We observe in equations 3.15b and 3.15c that there is an inertial coupling involving the $\dot{\gamma}$ and $\dot{\psi}$ terms. A decoupling of the inertial terms results in a further coupling on the control side, i.e., the $(L + T\sin\alpha)$ term that only appears in equation 3.15c appears in both of the decoupled expressions. Nevertheless, the equations take on a cleaner form when decoupled. They are also easier to solve and allow for easier physical interpretation in decoupled form. Application of the Pythagorean trig identity and some algebraic manipulation produces equations 3.16.

$$m\dot{V} = T\cos\alpha - D - mg\sin\gamma \tag{3.16a}$$

$$mV\dot{\gamma} = \left(T\sin\alpha + L\right)\cos\phi - mg\cos\gamma \tag{3.16b}$$

$$mV\dot{\psi}\cos\gamma = \left(T\sin\alpha + L\right)\sin\phi \tag{3.16c}$$

Taken together, the kinematic equations 3.8 and the dynamic equations 3.16 constitute a set of six first-order differential equations. These equations are the core of the reduced-order dynamic aircraft model.

3.3 Open Loop Control Laws

In principle, the open loop thrust, lift and bank angle commands can be extracted directly from the equations of motion through an algebraic inversion of equations 3.16. If successful, the thrust, lift and bank angle consistent with the instantaneous maneuvering state of the aircraft would result. However, closed form inversion of equations 3.16 is not possible. This is due to the nonlinear terms in α, together with the implicit dependence on α in the lift and drag terms. Given that the air vehicle model is only valid for linear (pre-stall) aerodynamics, it seems reasonable to linearize α terms in equations 3.16. This provides

$$m\dot{V} = T - D - mg\sin\gamma \tag{3.17a}$$

$$mV\dot{\gamma} = (T\alpha + L)\cos\phi - mg\cos\gamma \tag{3.17b}$$

$$mV\dot{\psi}\cos\gamma = (T\alpha + L)\sin\phi \tag{3.17c}$$

Unfortunately, closed form inversion of equations 3.17 is also not available. When attempted using MAPLE's symbolic processing engine, the computer eventually crashed due to a memory access violation. The next logical simplification is to exclude the α terms entirely, i.e., let $\alpha = 0$. The resulting system equations are now equivalent to the point mass model put forward by MITRE in their 2015 paper [24]. Equations 3.18 represent the simplified dynamic system used to produce feedforward control signals. Their algebraic inversions, equations 3.19, provide open loop commands as a function of state data and the accelerations $\dot{V}$, $V\dot{\gamma}, V\dot{\psi}$.

Simplified plant model for open loop control

$$m\dot{V} = T - D - mg\sin\gamma \tag{3.18a}$$

$$mV\dot{\gamma} = L\cos\phi - mg\cos\gamma \tag{3.18b}$$

$$mV\dot{\psi}\cos\gamma = L\sin\phi \tag{3.18c}$$

Analytic control inversions

$$T = m\dot{V} + D + mg\sin\gamma \tag{3.19a}$$

$$L = mg\cos\gamma\cos\phi + mV\dot{\gamma}\cos\phi + mV\dot{\psi}\cos\gamma\sin\phi \tag{3.19b}$$

$$\phi = \tan^{-1}\left[\frac{V\dot{\psi}\cos\gamma}{V|\dot{\gamma}| + g\cos\gamma}\right] \tag{3.19c}$$

3.4 Reference Platform: UAV 3100

Demonstration of the point mass model and the guidance controllers developed in this thesis is carried out using a fictitious UAV with a nominal gross weight of $3{,}100\ lb$. The model is referred to as the UAV 3100. It has no real-world counterpart and has been assembled for this academic purpose. The aerodynamic data has been collected using digital DATCOM; the propulsion data is directly from the Mathworks Generic Turbofan model available with the Aerospace Blockset. Thus the aerodynamic and engine performance parameters are meant to be realistic, representative data, but with no pedigree to any specific models or hardware.

The drag polar data is limited to two sub-sonic speeds: M = 0.3 and M = 0.925. Coefficient data is linearly interpolated for intermediate speeds and capped for operation outside the interval. For flight below 0.3M the compressibility effects are negligible.

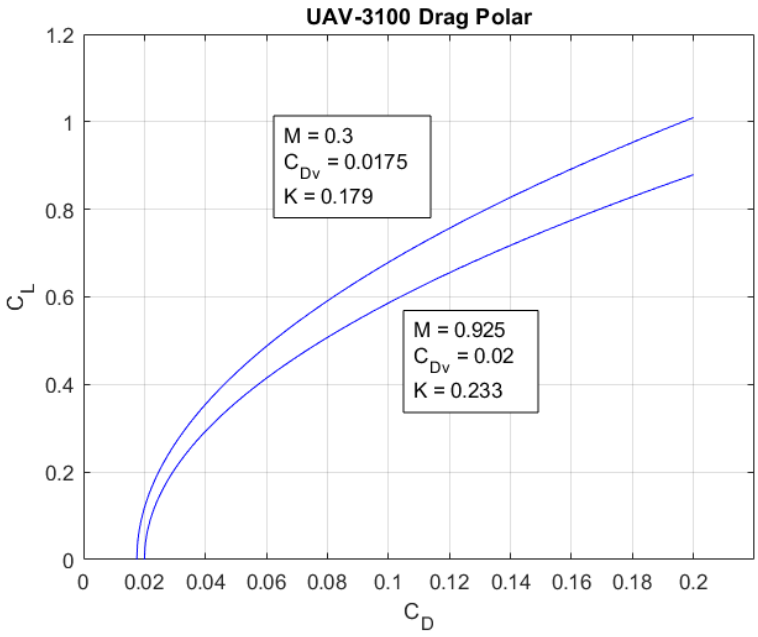


Figure 3.5: Drag polar curve fit

For the coefficient data listed in Figure 3.5, the C_{L_V} data is taken to be zero for both flight points. Thus the drag model is more in the spirit of equation 3.1 than equation 3.2. For the lift curve, the lift at zero angle of attack is assumed to be zero, with the lift slopes $C_{L_\alpha} = 4.154\,\mathtt{rad}^{-1}$ at 0.3M and $C_{L_\alpha} = 5.184\,\mathtt{rad}^{-1}$ at 0.925M. These data are also linearly interpolated (a better interpolation model would certainly be employed for a more realistic implementation).

The engine data simply caps the maximum net thrust as function of Mach and altitude. The sea-level static thrust rating was selected so as to achieve a flight envelope that extends into the high subsonic speed regime, but not in excess of 0.925M. Figure 3.6 displays the maximum installed thrust data within the flight envelope for the UAV 3100. The flight envelope is generated by running a trim sweep over a set of speed and altitude flight conditions. The trim routine uses Newton iterations to converge on the required lift and thrust values for sustaining a steady-level cruise at the operating point. With values for the required lift and thrust, the trim algorithm then checks against available lift and available power to determine whether the aircraft can achieve the desired operating cruise condi-

tion. Most of the boundary points along the flight envelope perimeter are defined by the power condition; however, the linear portion on the left hand side of the figure is due to aerodynamics.

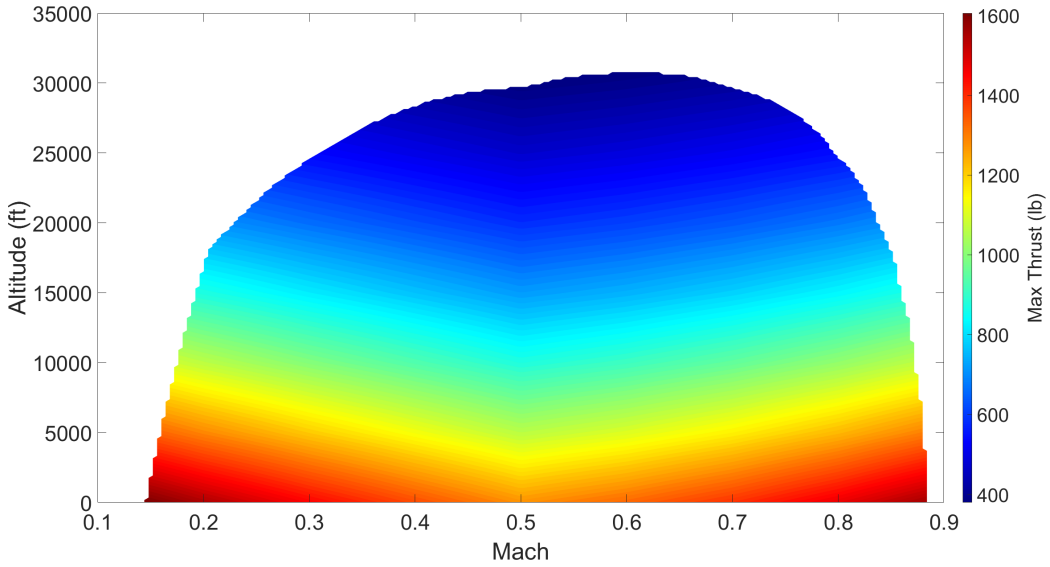


Figure 3.6: Maximum installed thrust as a function of flight condition

For all altitudes, we observe in Figure 3.6 a minimum point in max thrust at 0.5M. This is responsible for the small cleft that is visible at the top of the flight envelope at this Mach number.

The flight envelope plots provide a convenient means to convey information about the aircraft performance. Figures 3.7 and 3.8 are included as additional examples. In Chapter 4, plots of this kind are used to display information on robustness of the guidance controller.

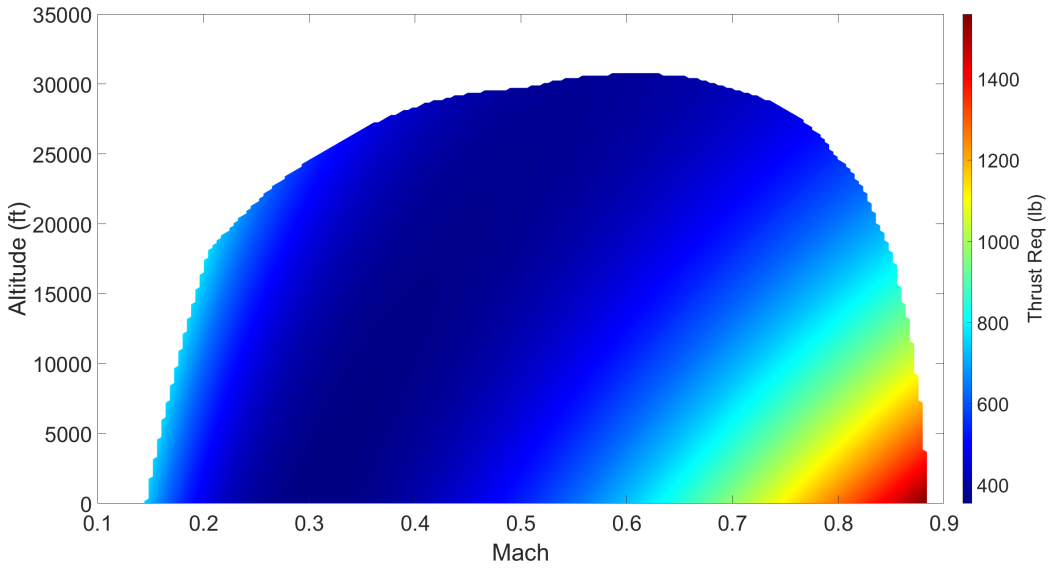


Figure 3.7: Thrust required to trim the aircraft

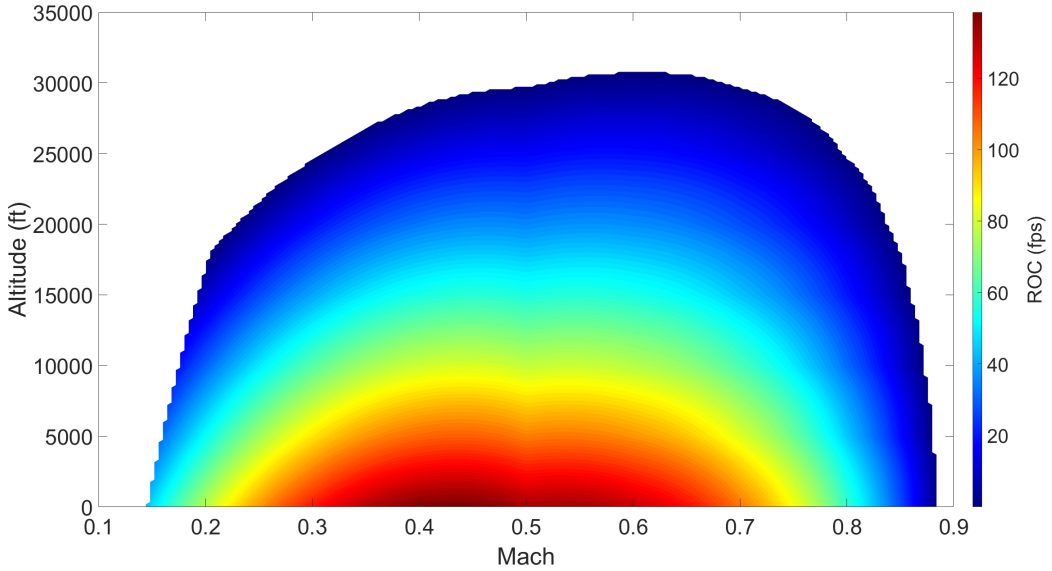


Figure 3.8: Maximum rate of climb

3.5 Mission Design

To evaluate the effectiveness of the guidance controllers, a set of five reference flight points has been created. They are listed out in Table 3.1 and shown graphically on the excess thrust diagram, Figure 3.9.

Design Point	**Mach**	**Altitude (ft)**
Design cruise condition	0.70	20,000
Low-speed sea-level	0.20	0
High-speed sea-level	0.85	0
Low-speed cruise	0.30	20,000
High-altitude cruise	0.60	30,000

Table 3.1: Reference flight conditions for UAV 3100

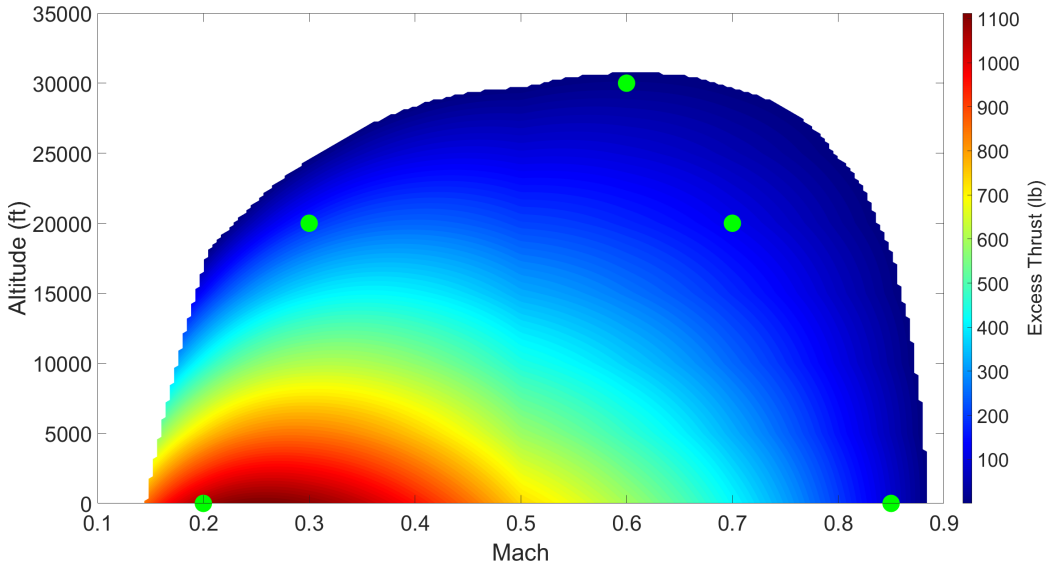


Figure 3.9: Aircraft reference points (green) shown over the flight envelope

The reference points have been selected to represent a nominal design flight condition for the UAV 3100 and four additional operating points at extreme positions in the flight envelope. The design point is a 0.7M cruise at 20,000 ft altitude. This is the point at which all of the baseline LQR analysis is conducted. The four remaining points allow us to assess the performance of the controller at off-design conditions. Guaranteeing desirable (and stable) performance at *all* points in the flight envelope is a subject that is addressed in chapter 4. However, the continuity of the plant data and of the control formulation tend to give the designer confidence that if suitable performance can be demonstrated at the reference points, the design is likely to be acceptable throughout the flight envelope. This philosophy is commonly adopted by aircraft control designers.

The controller performance is tested further by creating a hypothetical mission that exercises some of the design points and maneuvers between them. The phases of the design mission are listed in Table 3.2 and shown graphically in Figure 3.10.

Phase	**Description**
0	Takeoff at sea-level and Mach 0.2
1	Accelerating climb to 20000 ft and Mach 0.3
2	90 degree turn to the right
3	Acceleration to Mach 0.7
4	270 degree turn to the left
5	Constant Mach descent to sea-level
6	Constant altitude deceleration to Mach 0.2

Table 3.2: Phases of the design mission for UAV 3100

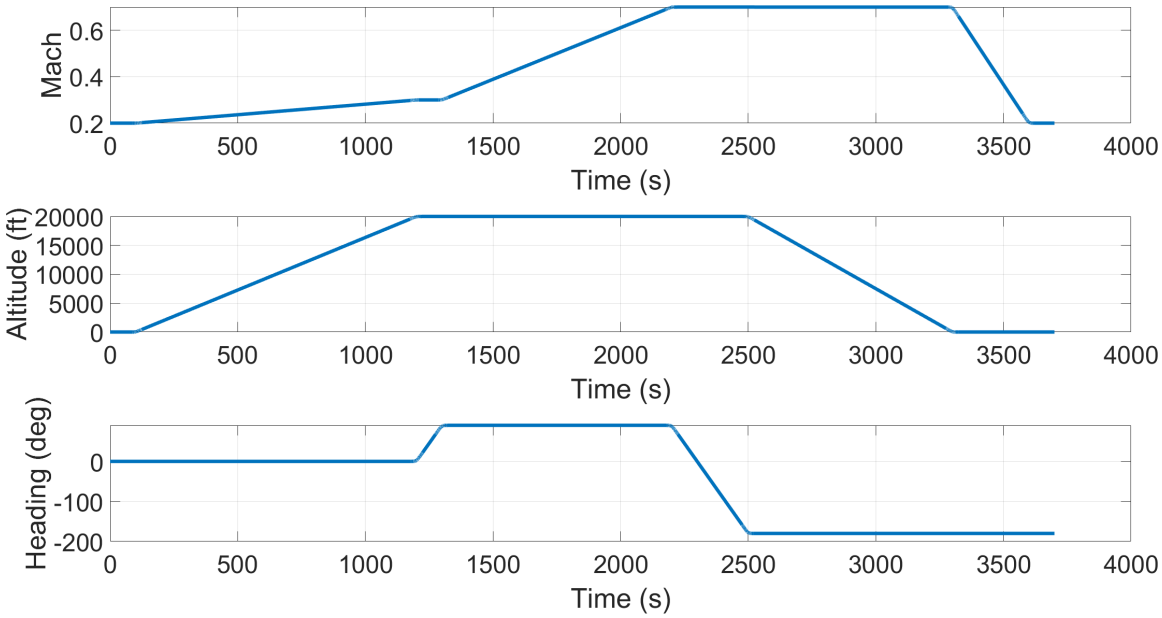


Figure 3.10: Design mission time vectors of Mach, altitude, and heading

The mission described in Table 3.2 is specified to the model as a time stamped command vector for Mach, altitude, and heading. While not at all comprehensive, the design mission is a common tool in demonstrating acceptable performance through a significant portion of the flight envelope using time domain simulation.

It should be noted that the corner points for the command vectors are all "rounded" using Bezier curves to assure smoothness. This has been implemented through a preprocessing algorithm because the open loop command signals require derivatives of the trajectory data.

4. LQR Control of the Reduced Order Aircraft Model

Chapter 2 provided a detailed review of LQR control design with a simple example aircraft model. In this chapter we design an LQR feedback controller for the reduced order aircraft model introduced in Chapter 3. First, an LTI state space model of the system is created by assuming steady-level flight at any prescribed airspeed and altitude. A method is then developed to design the LQR weighting matrices with a goal of reducing user design burden. The performance and stability robustness of the closed-loop system are quantified using time domain step response metrics and MIMO stability analysis, respectively. Then, the controller's ability to track the design mission is investigated, noting the impact of incorporating feedforward control in the system. Finally, the efficacy of an automated method used to improve performance at off-design point is analyzed.

4.1 Controller Model Design

This section explores the application of LQR control to the reduced order aircraft model described in Chapter 3. The main goal of the controller is to track reference command data while eliminating steady-state error. Other design goals are to provide stable system performance "out of the box", without requiring case-specific tuning, while at the same

time allowing the user to tune the system to meet time domain response metrics should that be of importance.

Creating a closed-loop system that tracks with zero steady-state error requires development of a compensator. This is not surprising, as most systems whose dynamics are describable through application of Newton's law are of type zero. Compensation is introduced to add additional integral control action to the closed-loop system [11]. Figure 4.1 shows the block diagram representation of the LQR controller applied to the reduced order aircraft model with compensation.

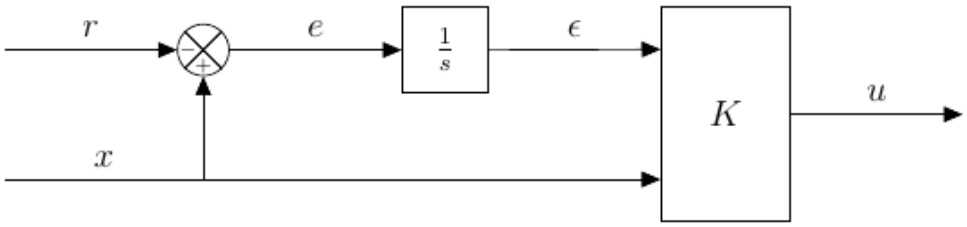


Figure 4.1: LQR control architecture with compensation

The signals listed on the diagram correspond to the state vector x, the reference command r, the control input to the plant u, the tracking error e, and integrator state ϵ. The product of compensation is the addition of three closed-loop control equations, one for each tracking loop (h, V, ψ). Using the altitude tracking loop as an example, the rate of change in the integral state is dictated by the instantaneous tracking error $\dot{\epsilon}_h = h - r_h$. The integral state ϵ_h is used to eliminate steady-state error in the altitude tracking loop. With a general understanding of the controller model, the development process of the controller can begin.

LQR control requires a model expressed in state space form and a set of weighting matrices that allow the user to define the preference between command tracking and control effort. The first step, generating a state space model from a set of nonlinear equations of motion, is discussed in detail in the following pages.

4.1.1 State Space Modeling

The state space model is used to represent the equations of motion for the model in linear form. The simplified dynamic equations of motion for the controller development are repeated here as equations 4.1. While there are three kinematic equations, shown in equations 4.2, the design goals of the system are not concerned with tracking the x and y locations in NED space. Altitude (obtained by integrating equation 4.2c) is the only position important to the tracking problem. The last three equations in the model are created to add integral action in the three tracking loops. This closed-loop system is designed to track reference commands in altitude, velocity, and heading. The full equation set is the following:

Dynamic equations

$$m\dot{V} = T - D - mg\sin\gamma \tag{4.1a}$$

$$mV\dot{\gamma} = L\cos\phi - mg\cos\gamma \tag{4.1b}$$

$$mV\dot{\psi}\cos\gamma = L\sin\phi \tag{4.1c}$$

Kinematic equations

$$\dot{x} = V\cos\psi\cos\gamma \tag{4.2a}$$

$$\dot{y} = V\sin\psi\cos\gamma \tag{4.2b}$$

$$\dot{h} = V\sin\gamma \tag{4.2c}$$

Closed loop compensator equations

$$\dot{\epsilon}_h = h - r_h \tag{4.3a}$$

$$\dot{\epsilon}_V = V - r_V \tag{4.3b}$$

$$\dot{\epsilon}_\psi = \psi - r_\psi \tag{4.3c}$$

Associated open loop commands

$$T = m\dot{V} + D + mg\sin\gamma \tag{4.4a}$$

$$L = mg\cos\gamma\cos\phi + mV\dot{\gamma}\cos\phi + mV\dot{\psi}\cos\gamma\sin\phi \tag{4.4b}$$

$$\phi = \tan^{-1}\left[\frac{V\dot{\psi}\cos\gamma}{V|\dot{\gamma}| + g\cos\gamma}\right] \tag{4.4c}$$

State space linearization

The state equations of this model consist of three dynamic equations, one kinematic equation, and three reference command tracking equations. Questions immediately arise regarding the inclusion of altitude as an explicit state given the relationship between altitude and gamma. A design goal of the closed-loop system is to have the ability to track altitude explicitly. The flight path angle allows for tracking of altitude rate, but not altitude. Therefore, the kinematic equation for altitude rate is included in the model.

Development of an LQR controller for the reduced order aircraft model requires an LTI system represented in state space form. The equations presented for the aircraft model are non-linear for both the state data and the controls, and vary with time due to the dependence on flight condition. Linearization is required to create a state space model of the non-linear system, which is used to determine the LQR gain matrix.

The first step of linearization is choosing an equilibrium point $f(x_0, u_0)$ such that

$\dot{x}_0 = 0$. This is typically a cruise condition where the aircraft response is most important, since this is the design point of the system. Next, a first order Taylor series expansion is created about the equilibrium point

$$f(x,u) \approx f(x_0,u_0) + \frac{\partial}{\partial x} f(x_0,u_0)\,(x - x_0) + \frac{\partial}{\partial u} f(x_0,u_0)\,(u - u_0) \tag{4.5}$$

where $\delta x = x - x_0$ and $\delta u = u - u_0$. The developed state space model describes the behavior of the system with respect to the chosen equilibrium point. Linearization about an equilibrium point is required to convert the set of nonlinear equations to state space form. This process can become rather laborious for aerospace applications where much of the aircraft derivative data is a function of flight condition. The state, control, and reference vectors for the state space model are

$$x = \begin{Bmatrix} h \\ V \\ \gamma \\ \psi \end{Bmatrix}$$

$$u = \begin{Bmatrix} T \\ L \\ \phi \end{Bmatrix}$$

$$r = \begin{Bmatrix} r_h \\ r_V \\ r_\psi \end{Bmatrix}$$

The state space model is derived by applying the first order Taylor series expansion to the state equations. Stevens and Lewis provide a compact method using four Jacobians

to determine the state space matrices required to define the system [23]. These Jacobian derivatives represent the rate that state equations change as each of the variables move away from the chosen equilibrium point, as denoted by the δ notation. The state space equation provided by Stevens and Lewis is

$$\frac{\partial f}{\partial \dot{x}}\delta\dot{x} + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial u}\delta u + \frac{\partial f}{\partial r}\delta r = 0 \tag{4.6}$$

Applying linearization of the form shown above requires the non-linear state equations be rearranged into homogeneous form $f(x, \dot{x}, u, r) = 0$. The Jacobian matrices are extracted using this homogeneous form of the state equations

$$f_1(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -\dot{\epsilon_h} + h - r_h = 0 \tag{4.7a}$$

$$f_2(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -\dot{\epsilon_V} + V - r_V = 0 \tag{4.7b}$$

$$f_3(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -\dot{\epsilon_\psi} + \psi - r_\psi = 0 \tag{4.7c}$$

$$f_4(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -\dot{h} + V\sin\gamma = 0 \tag{4.7d}$$

$$f_5(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -m\dot{V} + T - D - mg\sin\gamma = 0 \tag{4.7e}$$

$$f_6(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -mV\dot{\gamma} + L\cos\phi - mg\cos\gamma = 0 \tag{4.7f}$$

$$f_7(x, \dot{x}, u, r) = 0 \quad \rightarrow \quad -mV\dot{\psi}\cos\gamma + L\sin\phi = 0 \tag{4.7g}$$

The state space equation shown in 4.6 can be solved for $\delta\dot{x}$ to obtain the canonical form of a state space model

$$\dot{x} = Ax + Bu + Gr \tag{4.8}$$

Rearranging terms in 4.6 gives

$$\delta\dot{x} = -\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1}\left[\frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial u}\delta u + \frac{\partial f}{\partial r}\delta r\right] \tag{4.9}$$

Using the above equation, the state space matrices A, B, and G are easily identified and extracted. A deconstructed solution for each state space matrix is shown as

$$A = -\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1}\frac{\partial f}{\partial x} \tag{4.10}$$

$$B = -\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1}\frac{\partial f}{\partial u} \tag{4.11}$$

$$G = -\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1}\frac{\partial f}{\partial r} \tag{4.12}$$

The equations for the state space matrices show that four Jacobians must be computed: $-\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1}$, $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial u}$, and $\frac{\partial f}{\partial r}$. The first Jacobian is evaluated by differentiating the seven homogeneous model equations, shown in equation 4.7 with respect to $\dot{x}$.

$$\frac{\partial f}{\partial \dot{x}} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -m & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -mV & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -mV\cos\gamma \end{bmatrix} \tag{4.13}$$

The first Jacobian matrix is diagonal and can be inverted simply by taking the inverse of

each diagonal element. Including the minus sign in the matrix gives

$$-\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{mV} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{mV\cos\gamma} \end{bmatrix} \tag{4.14}$$

It is important to note that the last term in $-\left(\frac{\partial f}{\partial \dot{x}}\right)^{-1}$ is undefined if the flight path angle is equal to $\pm\frac{\pi}{2}$. This is known as the gimbal lock scenario for a three gimbal system, such as the Euler sequence used in this aircraft model. This phenomenon occurs when two gimbals are parallel, therefore a degree of freedom is lost in the system. Gimbal lock shows up in this model when the aircraft is flying straight up or down. The azimuth angle ψ is not well defined because, in this condition, the aircraft is not flying in the north-east plane at all. The impact of gimbal lock in this model is not an important restriction.

Taking derivatives of the homogeneous model equations with respect to the state vec-

tor results in

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \sin\gamma & V\cos\gamma & 0 \\ 0 & 0 & 0 & -\frac{\partial D}{\partial h} & -\frac{\partial D}{\partial V} & -mg\cos\gamma & 0 \\ 0 & 0 & 0 & \frac{\partial L}{\partial h}\cos\phi & -m\dot{\gamma} + \frac{\partial L}{\partial V}\cos\phi & mg\sin\gamma & 0 \\ 0 & 0 & 0 & \frac{\partial L}{\partial h}\sin\phi & -m\cos\gamma\dot{\psi} + \frac{\partial L}{\partial V}\sin\phi & mV\dot{\psi}\sin\gamma & 0 \end{bmatrix} \tag{4.15}$$

Within the matrix, $\frac{\partial f}{\partial x}$, four partial derivatives are left to be expanded. The non-zero density gradient present in the atmosphere $\frac{\partial \rho}{\partial h}$ creates the terms $\frac{\partial D}{\partial h}$ and $\frac{\partial L}{\partial h}$. However, the density gradient is assumed to be small enough that the partial derivatives with respect to altitude can be neglected. The same assumption has already been made to address the change in gravity with respect to altitude $\frac{\partial g}{\partial h}$. The acceleration due to gravity changes with altitude, but this change is sufficiently small that the effects due to this gradient can be neglected from the model. To address the two remaining partial derivatives $\frac{\partial D}{\partial V}$ and $\frac{\partial L}{\partial V}$, the expressions for aircraft lift and drag must be examined. Lift is obtained from the lift coefficient by multiplying by the dynamic pressure and the aircraft wing reference area.

$$L = C_L \cdot \frac{1}{2}\rho V^2 S \tag{4.16}$$

Differentiating the aircraft lift equation with respect to velocity gives

$$\frac{\partial L}{\partial V} = \frac{\partial C_L}{\partial V} \cdot \frac{1}{2}\rho V^2 S + C_L \cdot \rho V S \tag{4.17}$$

The aerodynamic model used to relate aircraft lift to drag is a drag polar, which is included in the model using a parabolic curve fit. The term $\frac{\partial C_L}{\partial V}$ is likely nonzero due to its dependence on Mach number and Reynolds number. But, the aerodynamic model used lacks the detail required to approximate the effects of Mach and Reynolds number on the lift coefficient. Therefore, the partial derivative is neglected from the state space model.

Aircraft drag is obtained from the non-dimensional drag coefficient C_D. Multiplying by dynamic pressure and the aircraft wing reference area allows the dimensional aircraft drag to be obtained as

$$D = C_D \cdot \frac{1}{2}\rho V^2 S \tag{4.18}$$

Differentiating the aircraft drag equation with respect to velocity gives

$$\frac{\partial D}{\partial V} = \frac{\partial C_D}{\partial V} \cdot \frac{1}{2}\rho V^2 S + C_D \cdot \rho V S \tag{4.19}$$

Knowing that C_D is computed using parabolic curve fit of the drag polar relationship $C_D = C_{D_0} + kC_L^2$, the dependence of drag coefficient on velocity can be expanded as

$$\frac{\partial C_D}{\partial V} = \frac{\partial C_{D_0}}{\partial V} + \frac{\partial k}{\partial V} C_L^2 + 2k\frac{\partial C_L}{\partial V} \tag{4.20}$$

The terms $\frac{\partial C_{D_0}}{\partial V}$ and $\frac{\partial k}{\partial V}$ are neglected using the same logic as $\frac{\partial C_L}{\partial V}$. The aerodynamic model does not contain the detailed data required to approximate the partial derivative. These reductions allow the Jacobian with respect to the state vector to be significantly simplified.

The state matrix A is obtained with equation 4.10 using the two Jacobians that have

been computed.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \sin\gamma & V\cos\gamma & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{m}\frac{\partial D}{\partial V} & -g\cos\gamma & 0 \\ 0 & 0 & 0 & 0 & -\frac{\dot{\gamma}}{V} + \frac{\partial L}{\partial V}\frac{\cos\phi}{mV} & \frac{g}{V}\sin\gamma & 0 \\ 0 & 0 & 0 & 0 & -\frac{\dot{\psi}}{V} + \frac{\partial L}{\partial V}\frac{\sin\phi}{mV\cos\gamma} & \dot{\psi}\tan\gamma & 0 \end{bmatrix} \tag{4.21}$$

This aircraft model will always be initialized in trimmed flight. The trimmed flight condition is defined as a flight point where rotations about the aircraft center of gravity are zero. For the point mass aircraft model, this condition has the implication that $\dot{\gamma}$ and $\dot{\psi}$ are zero. The aircraft is also initialized in a steady flight condition meaning no instantaneous speed change. The last descriptor that can be applied to the model initialization is the aircraft starts wings level to the ground (in a flat Earth sense), making γ and ϕ equal to zero. This fully defines the initialization point for the aircraft model as steady, level, trimmed flight. Applying the simplifications introduced by the initial condition, the state matrix reduces to

$$A_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & V & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{m}\frac{\partial D}{\partial V} & -g & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{mV}\frac{\partial L}{\partial V} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.22}$$

The initial condition used as the equilibrium point in the state space linearization significantly reduces the complexity of the state matrix A. The reduced state matrix is only a function of speed, mass, gravity, and two aerodynamic derivatives. To derive the control matrix B, a third Jacobian matrix is required. This matrix is obtained by differentiating the homogeneous model equations with respect to the control vector u.

$$\frac{\partial f}{\partial u} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & -\frac{\partial D}{\partial L} & 0 \\ 0 & \cos\phi & -L\sin\phi \\ 0 & \sin\phi & L\cos\phi \end{bmatrix} \tag{4.23}$$

There exists one partial derivative left to be expanded within the above Jacobian, $\frac{\partial D}{\partial L}$. Solv-

ing for C_L in equation 4.16 and substituting the expression into the drag polar equation creates drag as a function of lift.

$$D = \frac{1}{2}\rho V S \left(C_{D_0} + k \left(\frac{2L}{\rho V^2 S} \right)^2 \right) \tag{4.24}$$

Differentiating the drag equation with respect to lift results in the partial derivative

$$\frac{\partial D}{\partial L} = \frac{4kL}{\rho V^2 S} \tag{4.25}$$

Using equation 4.11, the state space control matrix becomes

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & -\frac{4kL}{m\rho V^2 S} & 0 \\ 0 & \frac{\cos\phi}{mV} & -\frac{L\sin\phi}{mV} \\ 0 & \frac{\sin\phi}{mV\cos\gamma} & \frac{L\cos\phi}{mV\cos\gamma} \end{bmatrix} \tag{4.26}$$

Applying the steady, level, trimmed initial condition allows the control matrix to be sim-

plified as

$$B_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & -\frac{4kL}{m\rho V^2 S} & 0 \\ 0 & \frac{1}{mV} & 0 \\ 0 & 0 & \frac{L}{mV} \end{bmatrix} \tag{4.27}$$

The reduced control matrix is now purely a function of lift, velocity, mass, and aerodynamic parameters. The last state space matrix left to be computed is the reference command matrix G. This matrix is included to fully define the controller state equations in state space form. The last Jacobian required to create the reference command matrix is obtained by differentiating with respect to the reference command vector r. The resulting matrix contains only three terms. These terms are constant at every flight condition, so the steady, level, trimmed flight condition applies no simplification to the matrix.

$$G = \frac{\partial f}{\partial r} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.28}$$

The state space representation of the reduced order aircraft model is now fully defined. The nonlinear homogeneous differential equations have been linearized with respect to the initial condition that will start a simulation. The state space model is a requirement for the development of many optimal control laws, such as LQR controllers. The development of the LQR control law can now continue by designing the weighting matrices that define the LQR cost function.

4.1.2 Cost Function Design

The Matlab LQR function used to generate the LQR gain matrix requires the state space model A and B matrices. It also requires three weighting matrices Q, N, and R that originate from the LQR cost function, equation 2.3. The Q matrix penalizes a non-zero state vector. The R matrix imposes a penalty on control effort. The N matrix is used to create a cross-coupling penalty between states and controls.

To create analytic expressions for the cost functional, all seven state space equations are squared so as to create quadratic forms. To keep the following equations readable, matrix element notation is used to denote the element row and column. This means the a and b terms in these equations refer to elements of the state space A and B matrices. The

expressions appear as

$$\dot{\epsilon}_h^2 = h^2 a_{1,4}^2 + 2hr_V a_{1,4} g_{1,1} + r_V^2 g_{1,1}^2 \tag{4.29}$$

$$\dot{\epsilon}_V^2 = V^2 a_{2,5}^2 + 2Vr_\gamma a_{2,5} g_{2,2} + r_\gamma^2 g_{2,2}^2 \tag{4.30}$$

$$\dot{\epsilon}_\psi^2 = \psi^2 a_{3,7}^2 + 2\psi r_\psi a_{3,7} g_{3,3} + r_\psi^2 g_{3,3}^2 \tag{4.31}$$

$$\dot{h}^2 = a_{4,6}^2 \gamma^2 \tag{4.32}$$

$$\begin{aligned} \dot{V}^2 = {} & L^2 b_{5,2}^2 + 2LTb_{5,1} b_{5,2} + 2LV a_{5,5} b_{5,2} + 2L\gamma a_{5,6} b_{5,2} + T^2 b_{5,1}^2 \\ & + 2TV a_{5,5} b_{5,1} + 2T\gamma a_{5,6} b_{5,1} + V^2 a_{5,5}^2 + 2V\gamma a_{5,5} a_{5,6} + \gamma^2 a_{5,6}^2 \end{aligned} \tag{4.33}$$

$$\dot{\gamma}^2 = L^2 b_{6,2}^2 + 2LV a_{6,5} b_{6,2} + V^2 a_{6,5}^2 \tag{4.34}$$

$$\dot{\psi}^2 = b_{7,3}^2 \phi^2 \tag{4.35}$$

We observe from these expressions, that due to the large number of zeroes in the A and B matrices (equations 4.22, 4.27) we can identify elements in the Q, N and R matrices that must be zero. This is accomplished through execution of a MAPLE script that uses symbolic algebra to perform a term-wise comparison of an expansion of the cost kernel

$$x^T Q x + u^T R u + x^T N u \tag{4.36}$$

By identifying common terms involving the state and reference data, analytic expressions for non-zero terms in the Q, N and R matrices are created. This reduces the number of potential design parameters from 55 down to 13.

The LQR cost function shown in Section 2.1 does not include the reference command vector, so the terms that contain r in these equations are eliminated. Before constructing

the penalty weighting matrices, a set of tunable gains are introduced so the user has some ability to change the response of the controller. The α_{1-13} gains are used to weight all the equations included in the LQR cost function. A description of the weight assignment is shown in Table 4.1.

Weighting Expression	**Description**
$\alpha_1\epsilon_h^2 \quad \alpha_2\epsilon_V^2 \quad \alpha_3\epsilon_\psi^2$	Penalize tracking error integral states
$\alpha_4\dot{\epsilon_h}^2 \quad \alpha_5\dot{\epsilon_V}^2 \quad \alpha_6\dot{\epsilon_\psi}^2 \quad \alpha_7\dot{h}^2$ $\alpha_8\dot{V}^2 \quad \alpha_9\dot{\gamma}^2 \quad \alpha_{10}\dot{\psi}^2$	Penalize changes in compensator and state variables
$\alpha_{11}T^2 \quad \alpha_{12}L^2 \quad \alpha_{13}\phi^2$	Penalize control expenditure

Table 4.1: User specified cost function weighting terms

Weighting defined by α_{1-13} is included in the cost function by placing consistent coefficients in the weighting matrices. The three weighting matrices are now defined as

$$Q = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_4 a_{1,4}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{2,5}^2\alpha_5 + a_{5,5}^2\alpha_8 + a_{6,5}^2\alpha_9 & \alpha_8 a_{5,5} a_{5,6} & 0 \\ 0 & 0 & 0 & 0 & \alpha_8 a_{5,5} a_{5,6} & a_{4,6}^2\alpha_7 + a_{5,6}^2\alpha_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha 6 a_{3,7}^2 \end{bmatrix} \tag{4.37}$$

$$N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \alpha_8 a_{5,5} b_{5,1} & \alpha_8 a_{5,5} b_{5,2} + \alpha_9 a_{6,5} b_{6,2} & 0 \\ \alpha_8 a_{5,5} b_{5,1} & \alpha_8 a_{5,6} b_{5,2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.38}$$

$$R = \begin{bmatrix} \alpha_8 b_{5,1}^2 & \alpha_8 b_{5,1} b_{5,2} & 0 \\ \alpha_8 b_{5,1} b_{5,2} & \alpha_8 b_{5,2}^2 + \alpha_9 b_{6,2}^2 & 0 \\ 0 & 0 & \alpha_{10} b_{7,3}^2 \end{bmatrix} \tag{4.39}$$

The term by term comparison method coupled with the reduced set of alpha gains has reduced the number of weights the user must select from 55 (assuming symmetric Q and R matrices) to 13 weights. While this is a large improvement, there are still a large number of weights that will change the controller response. The fact that these gains don't have the same units makes this problem more challenging. The problem of unit inconsistency can be addressed using normalization. Nelson suggests using an ideal maximum value for all penalty variables in the cost function to non-dimensionalize the cost function entirely [14]. The first alpha weight can be used as a general example of normalization. α_1 shows up in the LQR cost function because the term $\alpha_1 \epsilon_h^2$ is included. ϵ_h has units of ft $\cdot$ s. Therefore, the ideal maximum value with units of feet can be used to normalize $\alpha_1 \epsilon_h^2$ by

$$\alpha_1 = \frac{1}{\Delta h_{\text{max}} \tau}^2 \tag{4.40}$$

The variable τ is introduced to non-dimensionalize the time component of the integral state ϵ_h. This variable τ can be thought of as the time to regulate error. To reduce the dimensionality of the user selected design parameters, τ can be set to one second for all alpha terms, so that it weights all three tracking loops equally. Applying this normalization to all of the alpha weights gives

$$\alpha = \frac{1}{\Delta h_{\max}\tau}^2, \frac{1}{\Delta V_{\max}\tau}^2, \frac{1}{\Delta \psi_{\max}\tau}^2, \frac{1}{\dot{\epsilon}_{h\max}}^2, \frac{1}{\dot{\epsilon}_{V\max}}^2, \frac{1}{\dot{\epsilon}_{\psi\max}}^2, \frac{1}{\dot{h}_{\max}}^2, \frac{1}{\dot{V}_{\max}}^2, \frac{1}{\dot{\gamma}_{\max}}^2, \frac{1}{\dot{\psi}_{\max}}^2, \frac{1}{T_{\max}}^2, \frac{1}{L_{\max}}^2, \frac{1}{\phi_{\max}}^2 \tag{4.41}$$

Significant user design burden remains even with the gains now normalized. An optimization routine could be used to determine the best set of gains to evoke the intended response performance from the controller. Instead, a user driven trial and error method is used to alter the gains to achieve the desired response. The set of 13 gains must be reduced further using engineering judgment to simplify the design process. Ideally, all 13 gains would be retained to give the user the greatest ability to shape the closed loop system response. This solution is not feasible for constructive simulation because controller design would take a large amount of time due to the high number of gains. Therefore, engineering judgment, in this application, means that control over the system response is sacrificed to make the design process easier for a user. The claim is not made that this reduced set of gains is the best solution. This reduced set makes the controller design process easier and has provided good results in practice. The final set of gains used to design the closed loop system are obtained through three reductions.

The first set of gains to be examined are associated with $\dot{\epsilon}$, the rate of change of the controller integrator states. They differ from ϵ by a single integration. If only one set of gains were to be chosen, the ones that include integral action are preferred to retain the ability to regulate steady-state error. The state feedback component of the control law will stop the closed-loop system from becoming oscillatory, which is typical for integral only

control systems. With this reasoning, α_{4-6} can be set to zero.

The second reduction of gains regards the redundancy in penalizing the control vector explicitly. The equations of motion ($\dot{V}, \dot{\gamma}, \dot{\psi}$) contain terms that include the control vector, which creates an implicit penalty on the controls. Therefore, it makes sense to penalize non-zero accelerations, which will create a driving force in the cost function pushing the system towards a stable flight condition. Applying this to the cost function results in α_{11-13} being eliminated.

The final gain reduction regards the inclusion of $\dot{h}$ in the cost function. A penalty exists for $\dot{\gamma}$, which penalizes accelerations in the vertical maneuver band. It seems unnecessary to also penalize altitude rate, since accelerations are more closely linked to control effort. Therefore, α_7 is set to zero for this design.

The reduction of gains in this section has reduced the set from 55 to 13 using term by term comparison and 13 to 6 using further engineering judgment. Having only six gains allows the user to better understand the implications of each gain. The reduction to six gains is also important because there are two gains for each tracking loop (h, V, ψ), which can be used to balance tracking performance and aircraft control effort. With expressions created for the Q, N, and R penalty weighting matrices, Matlab's LQR function can be used to create a gain matrix K that will provide tracking of the reference command vector.

4.2 Controller Tuning

The LQR closed-loop controller is tuned with step response behavior in mind. The user has six gains available to change the response of the closed-loop system. In this section, the tuning methodology is demonstrated to achieve the desired response according to time domain step response metrics.

Time domain step response performance metrics were introduced in Section 2.3 and give the control system designer a variety of metrics from which to design the system's

closed-loop performance. These time domain metrics are examined for a step command of five feet of altitude, one foot per second of speed, and one degree of heading at the cruise flight condition. It is important to note that the LQR controller isn't the only control signal affecting the aircraft. Since the controller is constructed using a perturbation state space model, the control solution required to stabilize the aircraft at the steady, level, trimmed flight condition is used as an open-loop control signal to the aircraft equations of motion. This control solution is obtained using the zero angle of attack equations of motion because expressions for the required control vector are analytically solvable. The initial condition values added to the LQR signal are constants so they will not affect the step response.

Figures 4.2 - 4.4 use unity values for the six normalization values that create the six α weights to show a baseline behavior that will be improved upon. All step response data is obtained using the model that includes the zero angle of attack assumption. This eliminates the initial stabilization that occurs due to the discrepancy between the control solution and the nonlinear angle of attack model. Therefore, the response shown in the following figures is attributed to the step inputs only. If the nonlinear angle of attack plant were used, a numerical solver would need to be implemented to determine the stabilizing control vector at the initial condition. This would be important so the transients that arise due to inconsistency of the initial control vector and the plant are eliminated.

The cruise flight condition for the UAV 3100 is the design point for the closed-loop system, so the step commands input to the model represent small deviations from the cruise point. Like the roll dynamics model, simulation results shown in this section are achieved using Matlab/Simulink. The step response data from the reduced order aircraft model is obtained utilizing a 0.01 second time step using a fixed step integration scheme.

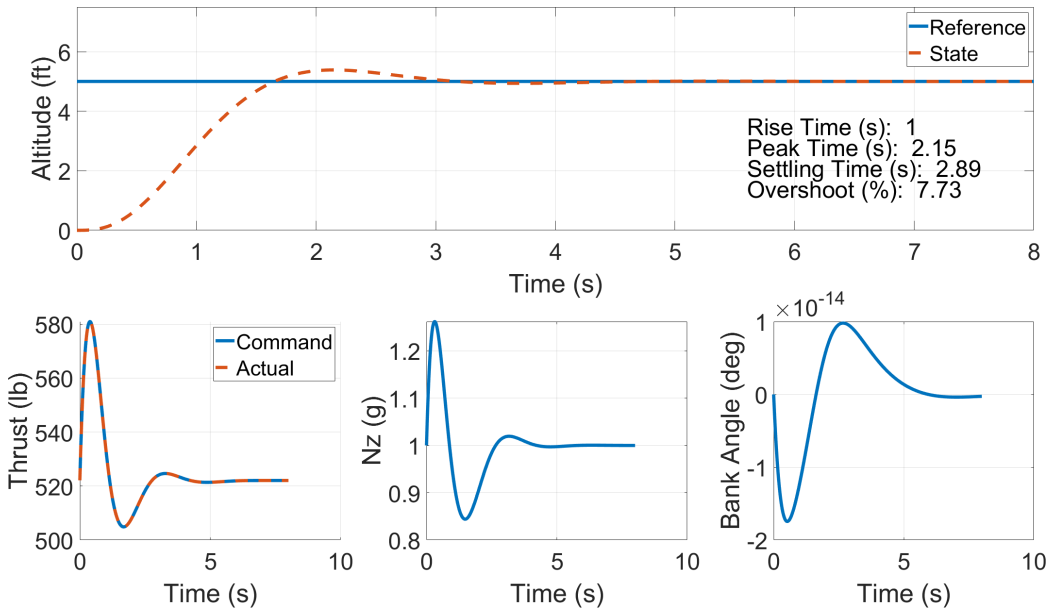


Figure 4.2: Altitude step response and control effort with unity gains and initial condition: $M = 0.7$, $h = 20000$ feet

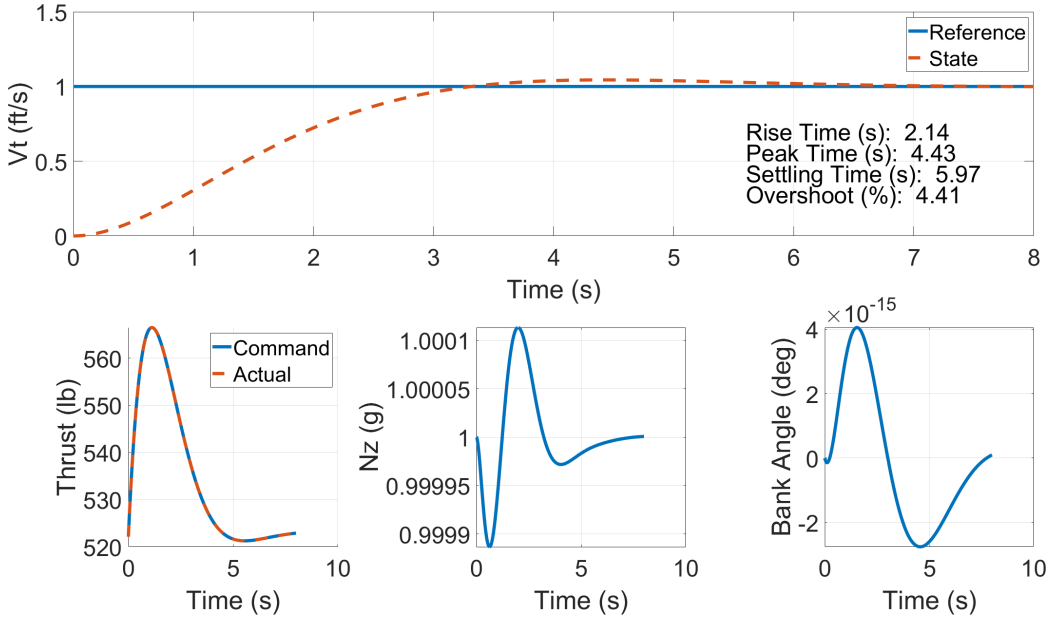


Figure 4.3: Velocity step response and control effort with unity gains and initial condition: $M = 0.7$, $h = 20000$ feet

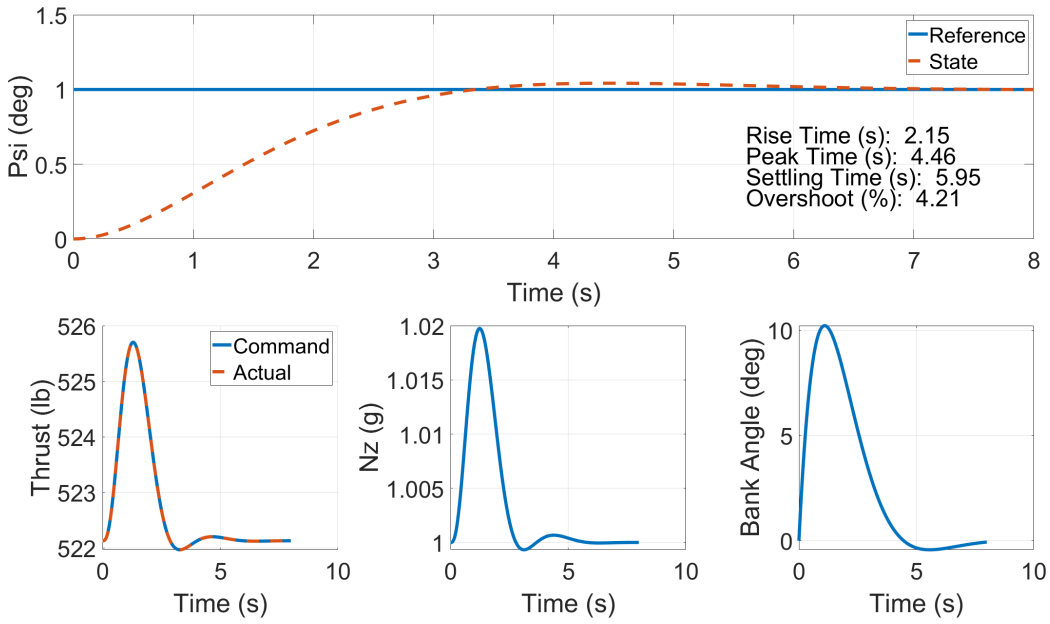


Figure 4.4: Heading step response and control effort with unity gains and initial condition: $M = 0.7$, $h = 20000$ feet

It can be seen that the response in the altitude loop is much faster than the speed or heading loops. All three loops reach the steady-state value in under 6 seconds. While the response maybe not be suitable for every user, the controller does track the reference command with zero user tuning (unity gains). The thrust command plot shows that the commanded and actual signals are always equal, meaning the controller never issues a command that exceeds the engine limits of the aircraft. This simple test shows the controller provides accurate tracking performance using the simplest set of design gains and provides the user with a good baseline.

One issue that has potential to cause problems is the small magnitude of the R matrix. A small R matrix means the controls have a very small penalty which can cause high gains in the K matrix. The weights on $\dot{V}$, $\dot{\gamma}$, and $\dot{\psi}$ can be tuned, keeping the diagonal elements of the R matrix much closer to one. Another reason to keep the R matrix close to diagonal with MIMO systems is that stability methods show that stability can degrade for an R matrix with off-diagonal terms. Therefore, tuning the weighting matrices to get closer to an identity matrix is a reasonable goal.

The tuning process is continued by establishing a design goal. The magnitudes of the step inputs were selected as a command that the closed-loop controller should quickly track. The rise time shown in Figure 4.2 is coincidentally equal to one second. Perhaps a user would conclude that $n_{z_{max}} = 1.25$ is too large for the small five foot altitude step. In this case, the rise time of the system should be increased to make the closed-loop response slower. The work to address the small R matrix leaves the user with three gains to change the response of the three loops. The result of the tuning process is shown in Figures 4.5 - 4.7.

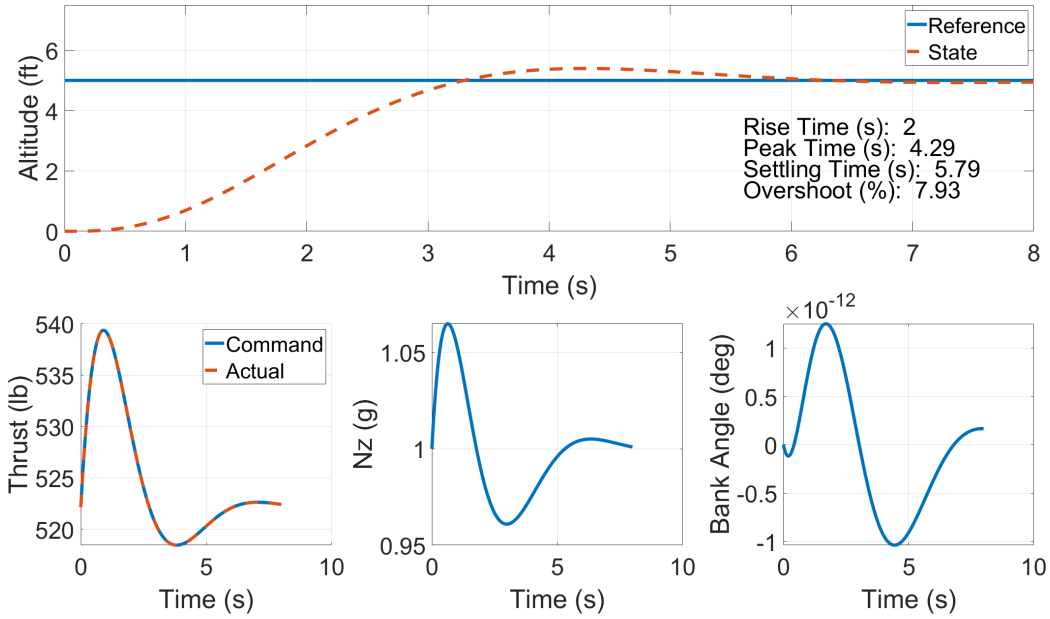


Figure 4.5: Altitude step response and control effort with initial condition: $M = 0.7$, $h = 20000$ feet

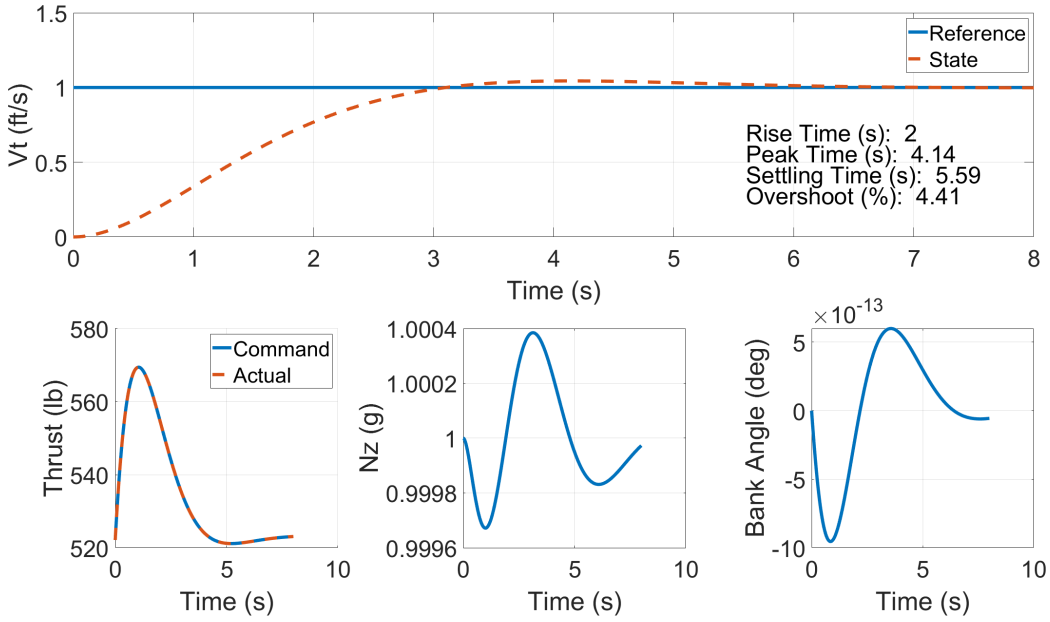


Figure 4.6: Velocity step response and control effort with initial condition: $M = 0.7$, $h = 20000$ feet

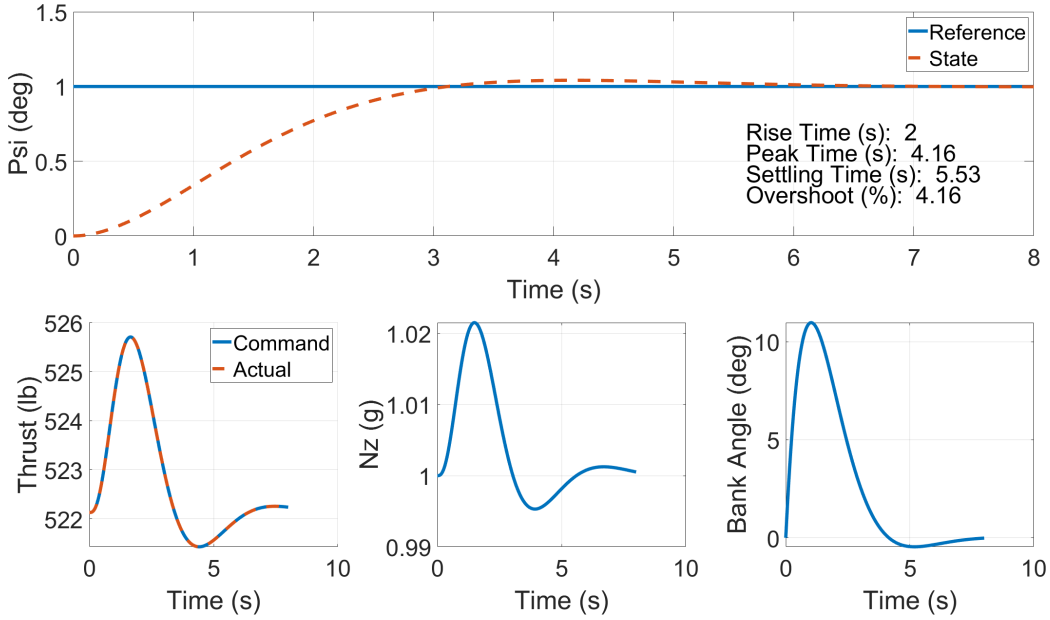


Figure 4.7: Heading step response and control effort with initial condition: $M = 0.7$, $h = 20000$ feet

To create a slower response to the step commands, a design goal for rise time of two seconds was selected for all three tracking loops. The step responses shown are not pushing performance limits of the aircraft as $n_{z_{\max}} = 1.07$. This is to be expected since it

would not be ideal for the aircraft to perform an aggressive maneuver for small changes in reference command. The purpose of this tuning exercise is to demonstrate that it is possible to achieve a desired step response using the reduced set of weights. The example of matching rise time is a very simple design problem for this controller. It is possible that the user wants to match all four time domain criteria which could require more weights to be retained and use of an optimization script. However, the user might also care more about overall mission tracking than step response performance, where the unity gains may work well enough. The main point is that the design goals could vary so widely that narrowing the scope to matching one time domain metric is still useful.

4.2.1 Stability Analysis

One of the main reasons that LQR controllers are desired is the guaranteed closed-loop stability. Stability is only guaranteed at the gain matrix design point, which is the cruise flight condition for this model. The other design points in the flight envelope must be examined to have confidence that the gain matrix will not destabilize the aircraft at any off-design condition. The gain and phase margin for the closed-loop system at the UAV 3100 design points are shown in Table 4.2. The stability margins are calculated using the same MIMO technique as is described in Section 2.2. The LQR gain matrix used in these calculations is calculated at the cruise flight condition.

Mach	Altitude (ft)	Gain Margin (dB)	Phase Margin (deg)
0.7	20000	[-9.4, 47.6]	59.7
0.6	30000	[-10.5, 48.7]	59.8
0.85	0	[-8.6, 38.8]	59.2
0.2	0	[-12.5, 57.9]	59.9
0.3	20000	[-12.6, 53.3]	59.9

Table 4.2: Gain and phase margin using the designed LQR gain matrix at the reference flight conditions for the UAV 3100

The stability margins for the closed-loop system vary as the flight condition gets further away from the cruise point. Gain margin varies significantly more than phase margin. LQR control theory claims that the closed loop system will have a gain margin that approaches positive infinity and a phase margin of 60 degrees. This guarantee, however, hinges on a diagonal R matrix. The design of the weighting matrices using term by term comparison created off-diagonal elements, meaning these guarantees can no longer be made [25]. Evidence of this can be seen in the reduction of gain margin in Table 4.2.

It is typical for the control system designer to have ideal stability margins for the closed-loop system. Typical values for aircraft actuation control loops state that minimum gain and phase margin should be 6 dB and 30 degrees, respectively [23]. Applying these design goals to the current system gives a great amount of confidence in the stability robustness of the system. The closed-loop system easily satisfies these stability goals for every design point for the UAV 3100.

The data presented in Table 4.2 serves as a quick check of the stability of the closed-loop system throughout the flight envelope. However, it does not guarantee that there are no regions within the flight envelope that degrade stability. Bode's gain-phase relationship states that there is a link between phase margin and the damping ratio of the system [2]. Lower phase margin results in a more oscillatory response, while higher phase margin

provides a highly damped response. A phase margin of 60 degrees is a balance point between tracking performance and oscillatory nature. Therefore, it is worth investigating phase margin on a more detailed level to determine if undue oscillations arise between the design points. This can be investigated by computing the phase margins at a sweep of flight conditions within the flight envelope, shown in 4.8. Again, only one gain matrix is used in the analysis. Calculating the gain matrix at every point would be appropriate if a gain scheduling approach were to be studied.

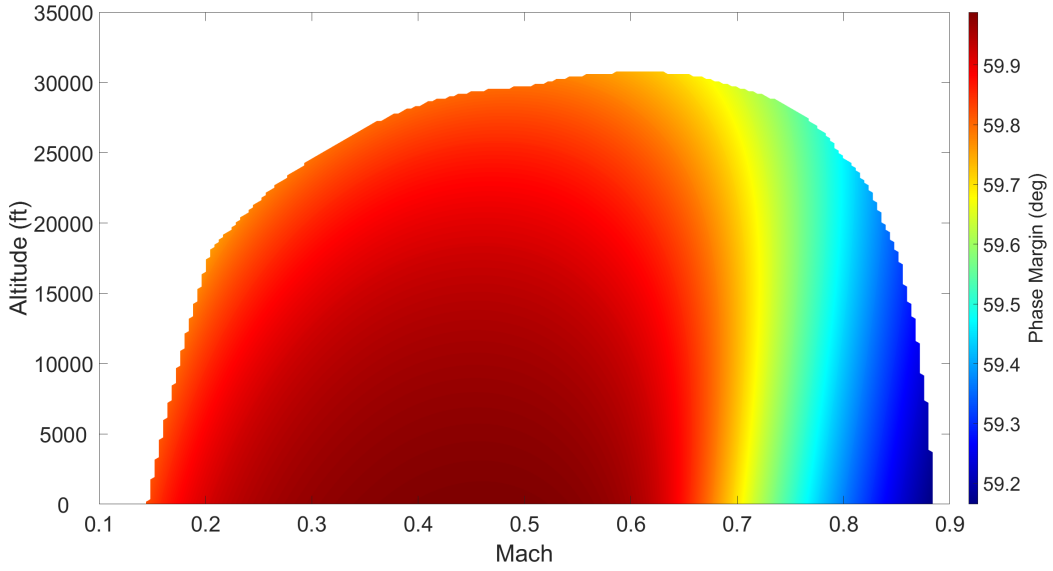


Figure 4.8: Phase margin within the flight envelope using the cruise condition gain matrix

The flight condition sweep shows phase margin does not deteriorate significantly as the aircraft moves away from the design point. The maximum reduction in phase margin over the flight envelope is one degree. The vast majority of the flight envelope has a phase margin of nearly 60 degrees, only degrading slightly at higher Mach numbers. This is very close to the guaranteed phase margin suggested by LQR control theory. The analysis of this section shows that gain scheduling is not required to maintain phase margin across the flight envelope. This, however, does not guarantee that closed-loop response will also be preserved in the same way. The tracking performance of the designed LQR controller is analyzed in the following section with respect to the UAV 3100 design mission.

4.3 Feedback Tracking of the Design Mission

The closed-loop system designed in preceding sections exhibits ideal step response behavior at the cruise point and has sufficient stability margins at the off-design points. Another goal is to have the controller track the design mission specified in Section 3.5. Figures 4.9 - 4.11 show the tracking performance of the LQR controller in the altitude, speed, and heading loops while highlighting the tracking error throughout the mission. The control signal of the LQR controller is added to the trim values for the control vector because the system is modeled as a perturbation off the initial steady, level, trimmed flight condition. The equations of motion used in this test include nonlinear angle of attack, therefore this test will show if the zero angle of attack assumption used to create the gain matrix is good enough to provide accurate tracking.

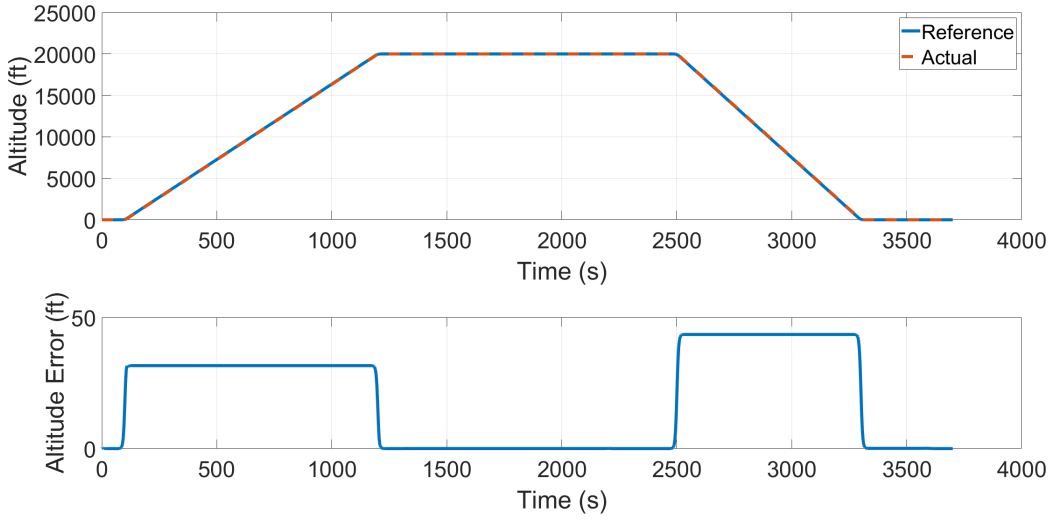


Figure 4.9: LQR altitude tracking and tracking error for the design mission

The altitude plot shows an initial climb of 20000 feet and a descent near the end of a mission. The controller provides accurate tracking throughout the mission. By reducing the error to zero near the middle of the mission, the controller shows its capability to track a constant command with zero steady-state error. Small errors of 35 to 45 feet arise when the controller is fed a ramp reference command. Errors would be smaller for a more casual maneuver, but this accelerating climb maneuver pushes the engine limits of the aircraft. The

creation of small errors during the ramp section is to be expected because the design goal of this controller was to track a constant command with zero steady-state error. Additional work would be required to design a controller that is able to track a ramp command, mainly adding a set of integrators to the compensator. The speed tracking loop is shown in the following figure using Mach number so the speed command changes due to speed of sound are not visible.

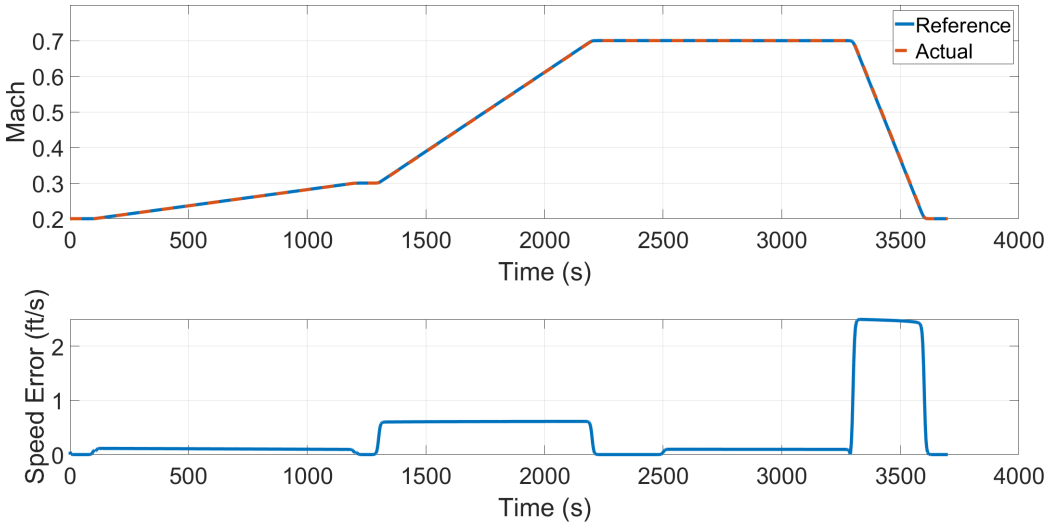


Figure 4.10: LQR Mach number tracking and tracking error for the design mission

Like the altitude plot, the controller provides accurate tracking of Mach number. One important observation is the non-zero speed error from 2500 to 3300 seconds. At first glance, this might be attributed to poor controller performance, but this flight segment is a constant speed descent. The small speed error arises due to the coupling of the $\dot{V}$ and $\dot{\gamma}$ equations of motion. Heading tracking performance is displayed in the final plot.

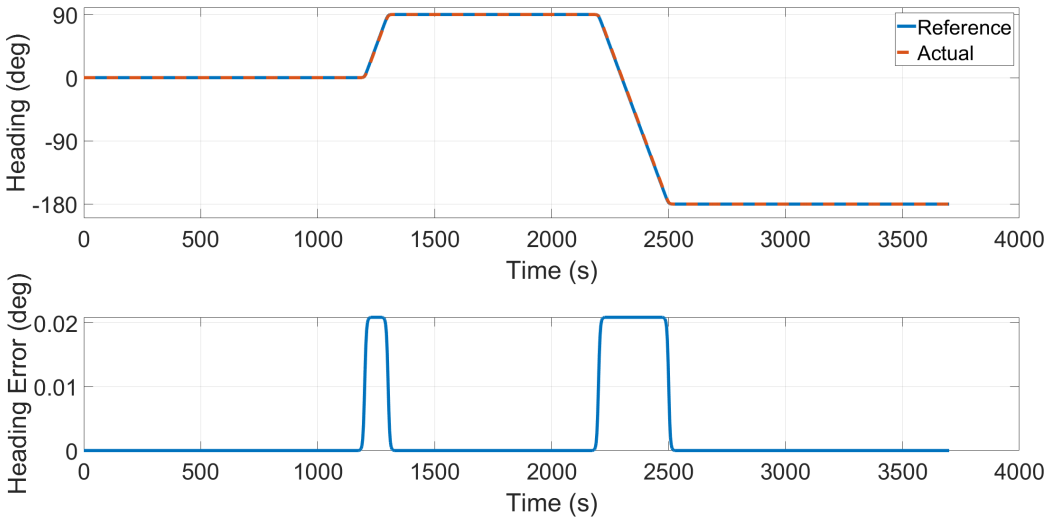


Figure 4.11: LQR heading tracking and tracking error for the design mission

Again, the tracking performance in the heading loop is excellent. The maneuvers in the heading direction are less aggressive than the altitude and speed directions. This can be seen in the very small heading errors that arise during the ramp commands. This design mission shows that the magnitude of the steady-state error for a ramp command is directly a function of the maneuver aggressiveness.

The results shown in this section rely on the closed-loop LQR controller to close out any tracking errors. The next section investigates the expanded use of the open-loop equations to reduce the control burden placed on the LQR controller.

4.4 Feedforward Control

The previous sections have discussed the inclusion of the trim control vector to allow steady, level, trimmed initialization at any point in the flight envelope. This means the LQR controller only provides control action when the the aircraft deviates from the initialization point. For the design mission, the aircraft spends most of its time away from the initial point, with a max deviation of 20000 feet and Mach 0.5. The LQR controller is responsible for the entire difference. Typically control designers want the feedback con-

troller to close small tracking errors instead of the entire error. Feedforward control is used to achieve this behavior. Open-loop control equations exist for this aircraft model and can be used to provide an estimate of the control vector required to track the reference command vector. The interaction between the feedforward and feedback controllers is shown in block diagram form as

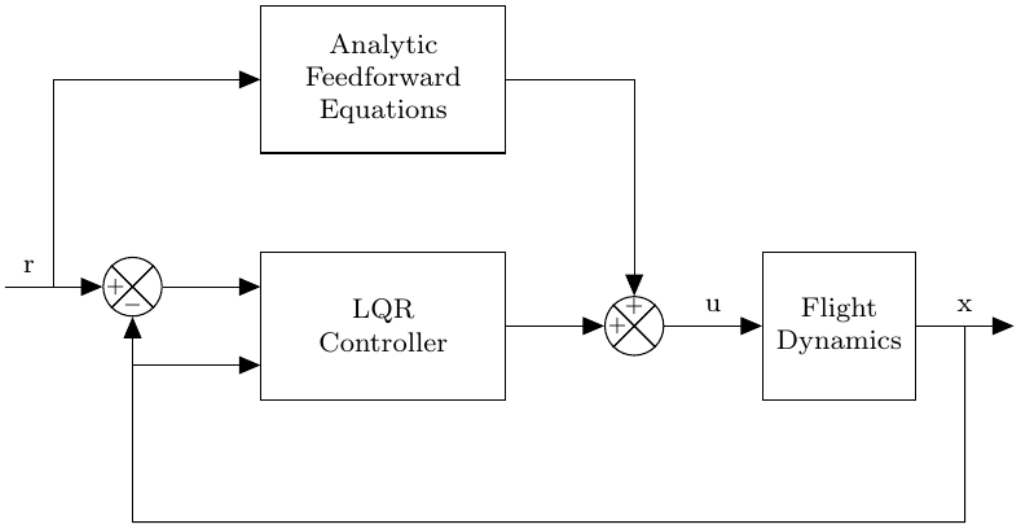


Figure 4.12: Combined feedforward, feedback control architecture

The dynamic equations of motion without angle of attack can be analytically solved for open-loop control equations as a function of aircraft state and $\dot{V}, \dot{\gamma}, \dot{\psi}$. The equations 4.4 allow for the creation of a feedforward control signal that will drive the aircraft near the intended reference value. By adding the feedforward command to the LQR control, the LQR controller should only work on the small modeling discrepancies that arise due to the zero angle of attack assumption present in the analytic control equations. In practice, the two controllers track the reference command, but exhibit a fighting behavior. This is not an acceptable response for the controller and more work must be done for the controllers to work together to track the reference command signal.

4.4.1 State Projection

The open-loop commands generated from equations 4.4 are a good estimate of the control vector that is required to reach the desired state. Since this estimate is available, it can be used to minimize the effort of the closed-loop controller. The remaining problem to be addressed is how to create a control architecture where the closed-loop component works in conjunction with the open-loop estimate. The LQR controller has no knowledge of the open-loop component, so it not surprising that disturbances in the control vector cause problems in the controller response. A solution proposed for this design problem is called state projection. The general idea is to project the effect the open-loop control vector will have on the aircraft model states and use that projection to reduce the error vector that is input to LQR for regulation.

State projection is applied to the model by running the open-loop control vector through the zero angle of attack equations of motion. The outputs of this operation are the projected aircraft states assuming the open-loop control vector is held constant during one model time step. The projected states can then be used to feed the LQR controller to clean up any errors between the projection and the reference vector. Finally, the open and closed-loop control vectors are added to create the vector of T, L, and ϕ that is sent to the aircraft equations of motion that include non-linear angle of attack dynamics. The combined controller is a form of cascade control, where the outputs of one controller affect the behavior of another controller. The control architecture exists in block diagram form as

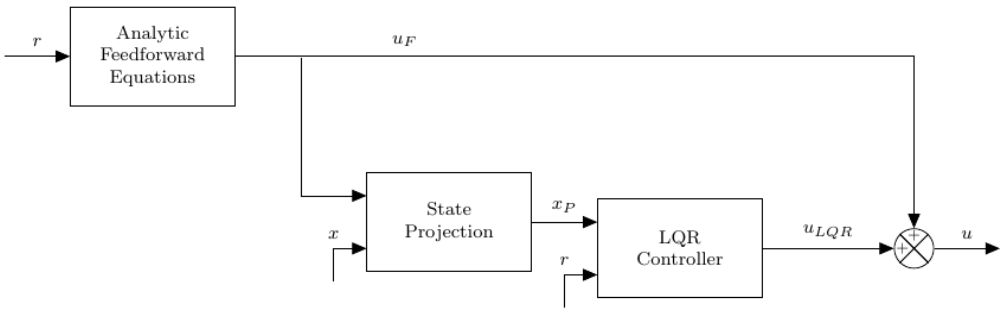


Figure 4.13: Control architecture with state projection

The effects of feedforward control with state projection can be examined by comparing the control effort exercised by the feedforward and closed-loop controllers over the course of the design mission. The following sections investigate the differences in tracking performance using the feedforward controller with state projection compared with the LQR controller alone. The distribution of control effort will also be compared to show the reduced control burden placed on the closed-loop controller.

4.5 Feedforward, Feedback Tracking of the Design Mission

The UAV 3100 design mission is used to determine the effectiveness of the inclusion of feedforward terms. The plots are of a slightly different layout as those in the LQR only section. Instead of comparing the actual aircraft state with the commanded reference value, they compare the state history for each control strategy: LQR control and LQR + feedforward control. This exercise is meant to investigate the differences caused in tracking performance when feedforward control is introduced. Figures 4.14 - 4.16 display the tracking performance along with instantaneous tracking error throughout the design mission. The model used in this exercise includes the same nonlinear angle of attack dynamics as the previous tracking plots, so the only difference in the model is the inclusion of the open-loop equations and state projection. The change in error displayed in the plots refers to the difference in instantaneous tracking error with the addition of feedforward control. A positive value means the combined controller increased tracking error and negative values mean it decreased tracking error.

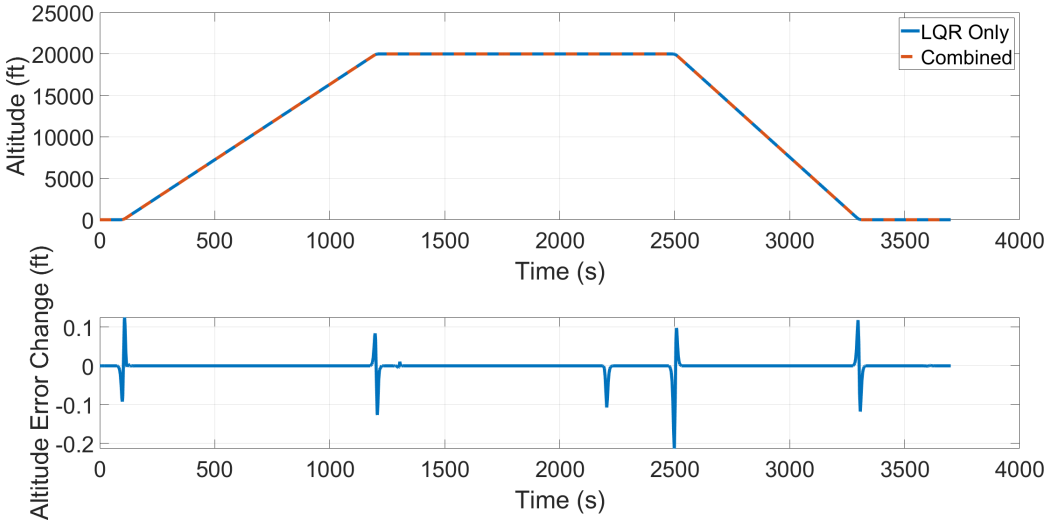


Figure 4.14: LQR and feedforward altitude tracking and tracking error for the design mission

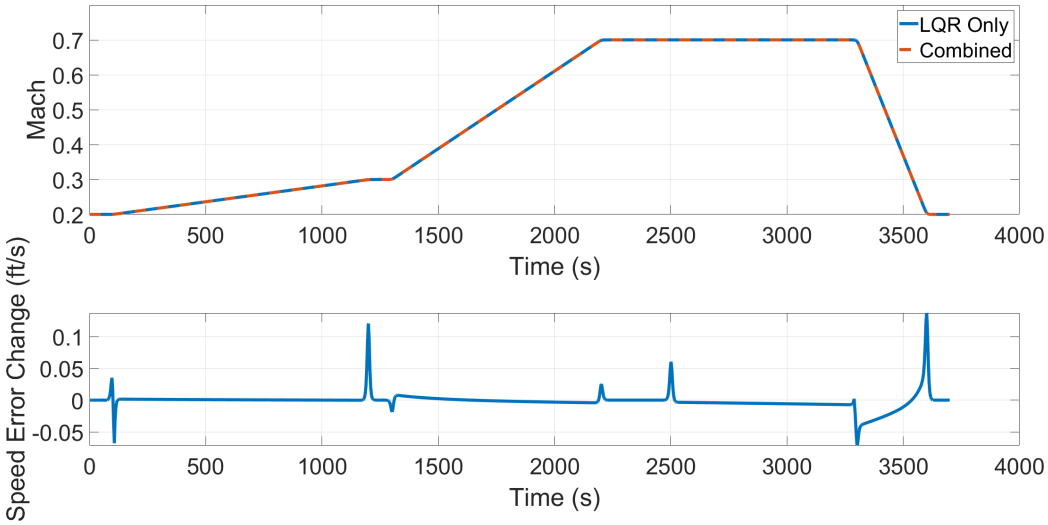


Figure 4.15: LQR and feedforward Mach number tracking and tracking error for the design mission

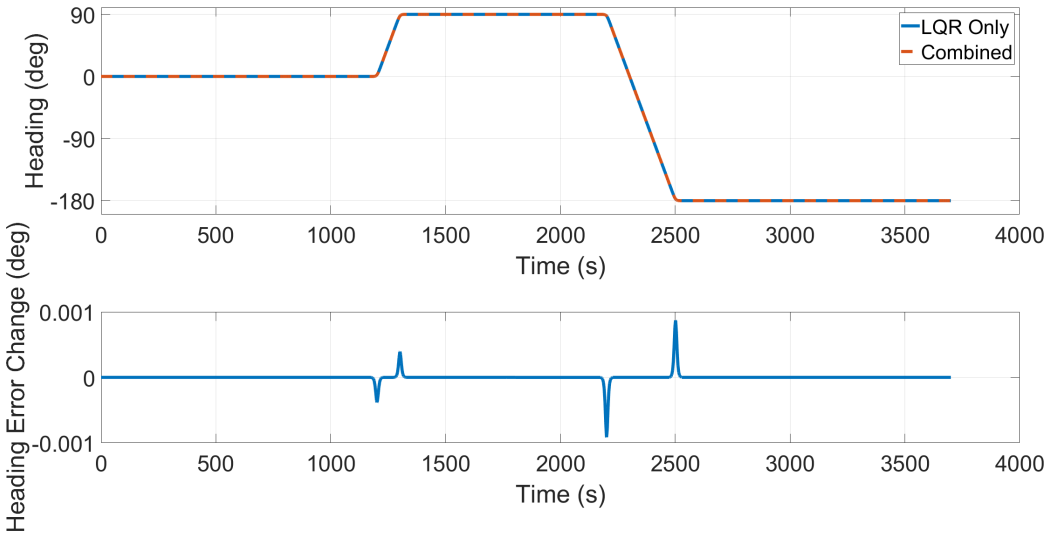


Figure 4.16: LQR and feedforward heading tracking and tracking error for the design mission

All three tracking loops show that the addition of feedforward terms does not significantly affect the tracking performance of the controller. This is not a poor result as the track performance was accurate using LQR alone. The purpose of adding feedforward control is to minimize the control effort expended by the LQR controller.

Tracking performance may be improved using open-loop equations that include angle of attack, since these are the equations of motion used in the dynamic model. However, these equations are not able to be obtained analytically, which is the reason for the zero angle of attack assumption. A numerical solver would need to be implemented to determine the control vector associated with the reference commands for the non-linear angle of attack equations.

4.5.1 Control Effort Comparison

The control effort issued by the LQR controller is predicted to decrease with the addition of feedforward control. The feedforward component should get much closer to the solution than only using the initial condition control solution. Figures 4.17 - 4.18 display the control distribution used to track the design mission using only the closed-loop LQR controller and

the trim control vector.

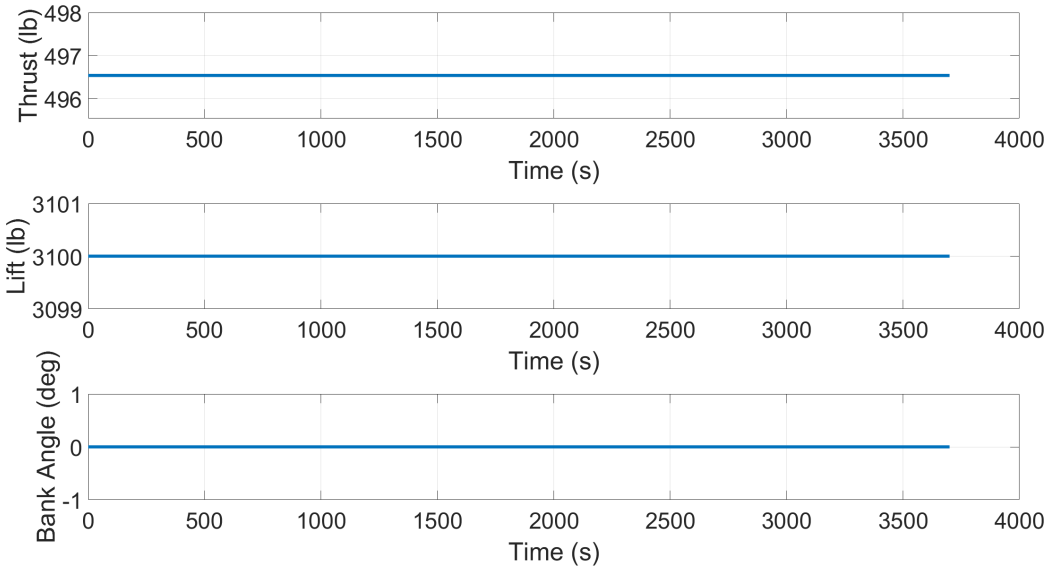


Figure 4.17: Open-loop control vector for the design mission without feedforward control

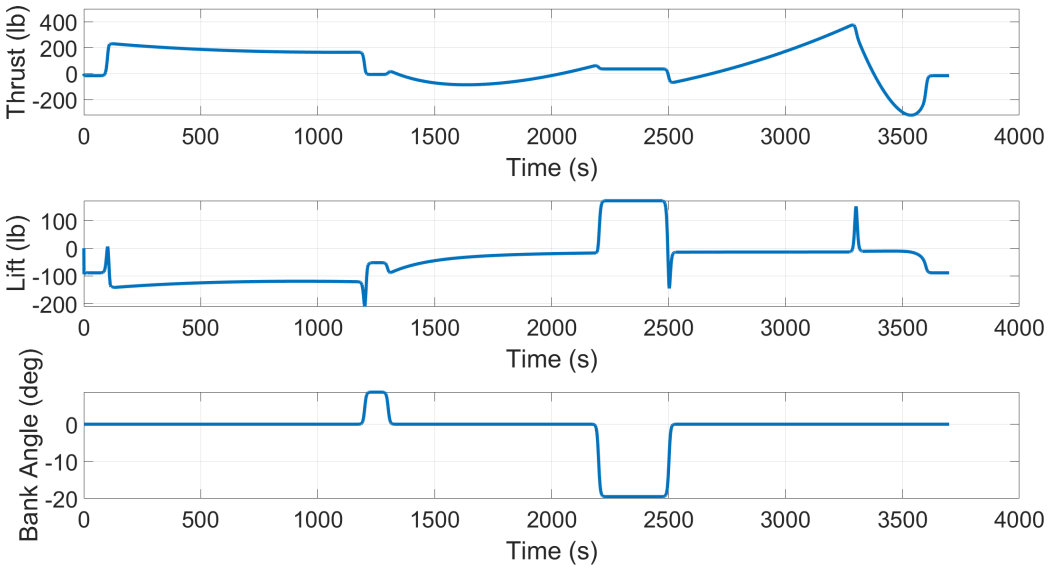


Figure 4.18: Closed-loop control vector for the design mission without feedforward control

The open-loop control values do not change because they are a constant control vector used to initialize the aircraft at a steady, level, trimmed flight condition. The closed-loop LQR controller starts by issuing a control vector of zeros at the initialization point, but quickly has to subtract from thrust and lift to adapt to the nonlinear angle of attack equations of motion. The LQR control effort used to track the design mission has a maximum

magnitude of roughly 400 pounds of thrust, 200 pounds of lift, and 20 degrees of bank angle. Figures 4.19 - 4.20 display the control distribution used to track the design mission using the combined LQR and feedforward controller. The control vector defined by the open-loop component will now change over time due to the addition of analytic feedforward equations.

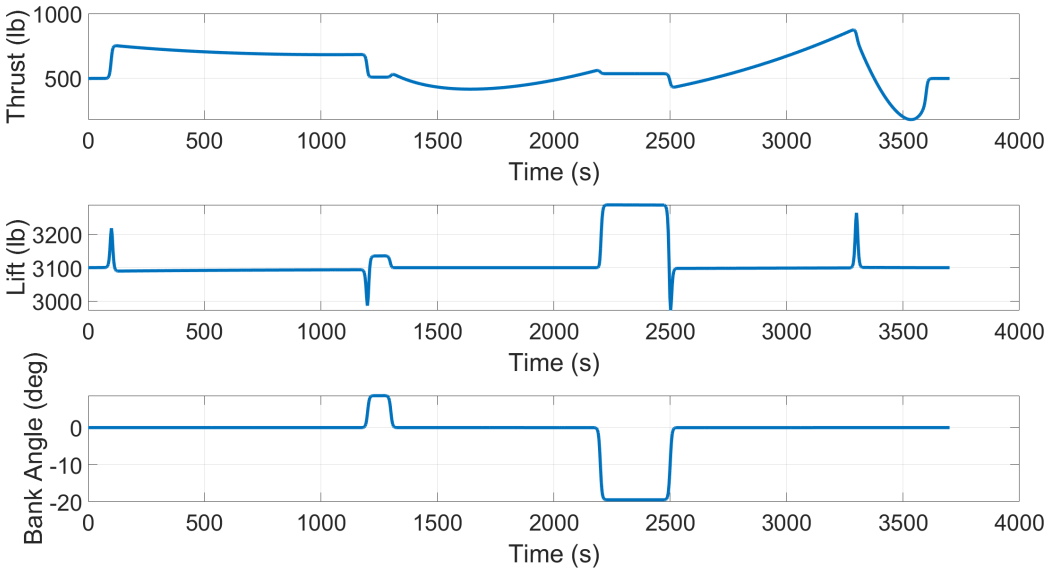


Figure 4.19: Open-loop control vector for the design mission with feedforward control

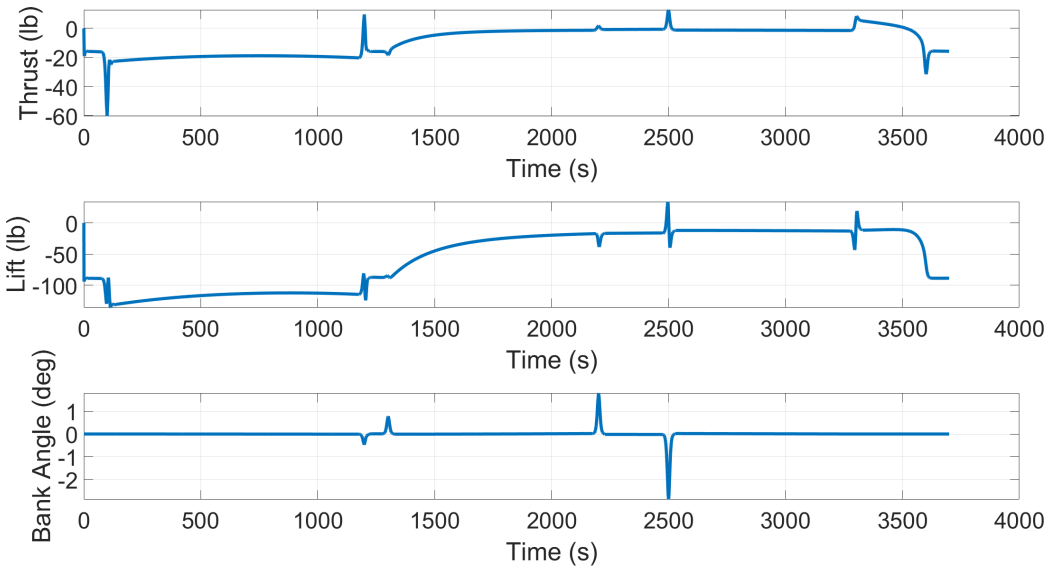


Figure 4.20: Closed-loop control vector for the design mission with feedforward control

The component of the control vector issued by the LQR controller is smaller when

feedforward terms are used. The maximum magnitudes of thrust and bank angle have been reduced to 60 pounds and three degrees, respectively. The lift command does not show as much of a decrease as the other two controls. The time history plot of lift shows that it initially decreases by 100 pounds, then only deviates from that number by about 50 pounds. The is due to the feedforward control equations using the zero angle of attack assumption. The feedforward component is almost always overestimating the control vector required to reach the reference command. This again brings up the idea of using a numerical solver to determine the control vector required for the non-linear angle of attack equations. But, without the added complexity of requiring a numerical solver, the control architecture presented in the section provides accurate tracking of the design mission and the option to include feedforward control to reduce the control burden placed on the closed-loop LQR component.

4.6 Automated Gain Scheduling

Traditional control techniques, such as PID control, use gain scheduling in the presence of a wide array of design points. The UAV 3100 would certainly require gain scheduling to achieve desired performance over the full flight envelope using a nested PID control strategy. The gain scheduling process involves tuning the controller at a set of design points and using interpolation to find an intermediary solution. The set of five design points used for the UAV 3100 could be sufficient to achieve desired performance across the flight envelope. The results shown in Sections 4.3 and 4.5 showed the design mission was tracked accurately using a single set of gains, making gain scheduling unnecessary. However, the control system designer may desire consistent time domain response characteristics over the flight envelope, where gain scheduling is likely required. Gain scheduling can be added to the LQR control methodology developed in this thesis with no additional design burden. Automated gain scheduling is a good name for this technique because the gain matrix is

simply recomputed at a variety of flight conditions. The change in the state space matrices adjust the gain matrix to account for flight condition. There is no limit to the number of flight conditions at which the gain matrix is recomputed, so there is potential for a higher density mesh. As an example, the gain matrix is computed at the five design points to show how the elements change. Figure 4.21 shows the change in the non-zero elements of the LQR gain matrix (the zero elements remain equal to zero). This plot is understood as a change in magnitude because it is obtained by dividing each element by its value at the cruise design point.

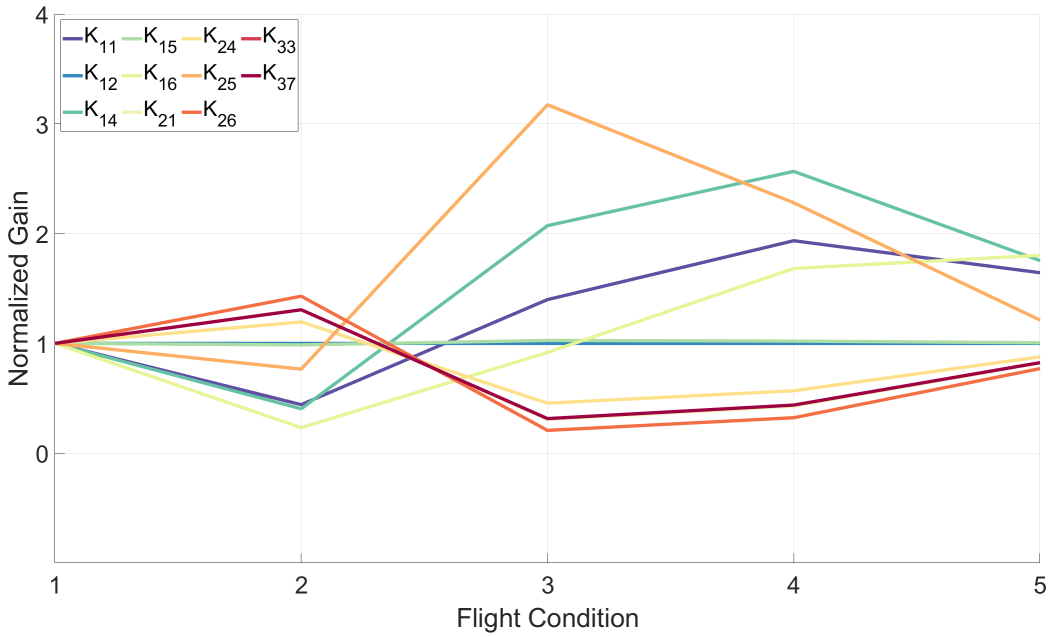


Figure 4.21: Variation of LQR gain matrix for five flight conditions points

The 11 non-zero elements of the gain matrix are shown to change over the five flight conditions in Figure 4.21. The flight conditions shown in this plot are obtained from Table 3.1. The gain matrix elements both increase and decrease in magnitude over the set of design points. This finding makes sense because the cruise condition, at which the gain matrix was designed lies in the central region of the flight envelope. There are design points at both higher and lower speeds. An interesting finding is that none of the gains change sign at the off-design flight conditions. This makes physical sense because the only

variables that should change sign are climb angle and heading. Climb angle is set to zero due to the level initial condition. Heading does not show up in the state space matrices because the only effect that heading has on the model is the change in north-east position for the aircraft. A single gain matrix element is investigated for a much larger set of flight conditions, the full flight envelope, in Figure 4.22.

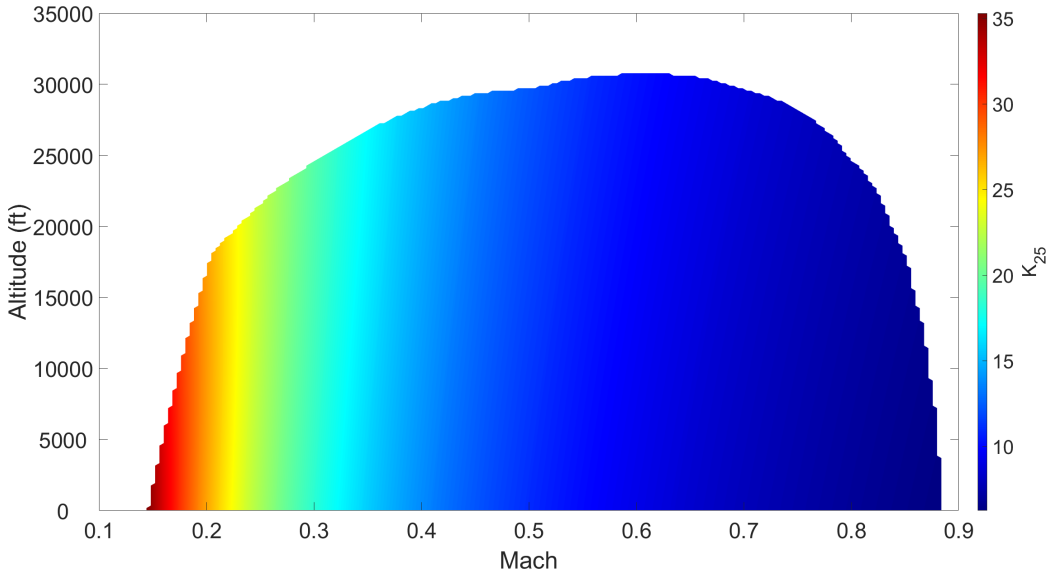


Figure 4.22: Variation of $K_{2,5}$ over the aircraft flight envelope

The gain matrix element $K_{2,5}$ varies from 5 to 35 over the flight envelope. This element was investigated because it showed the most significant change in magnitude over the set of flight conditions. $K_{2,5}$ is used to weight the effect of the aircraft velocity on the lift command of the LQR controller. The contour plot shows that the change is almost entirely a function of Mach number, with the largest changes occurring from Mach 0.1 to 0.3. It is difficult to determine the exact reasoning for the behavior due to the method that generates the gain matrix, solving the algebraic Riccati equation. The gain matrix requires the state matrix, the control matrix, and the three cost function weighting matrices. The effects of each on the gain matrix is not easily understood. However, it is reasonable that the lift command should vary with velocity due to the aerodynamic model, the drag polar. Lift is expressed as lift coefficient C_L in this model, which picks up a V^2 term upon obtaining

dimensional lift.

The phase margin achieved using automated gain scheduling is shown in Figure 4.23. The difference between this plot and Figure 4.8 is the flight condition where the gain matrix is computed. This plot is obtained using the gain matrix consistent with the flight condition, opposed to using the gain matrix computed at the cruise design point.

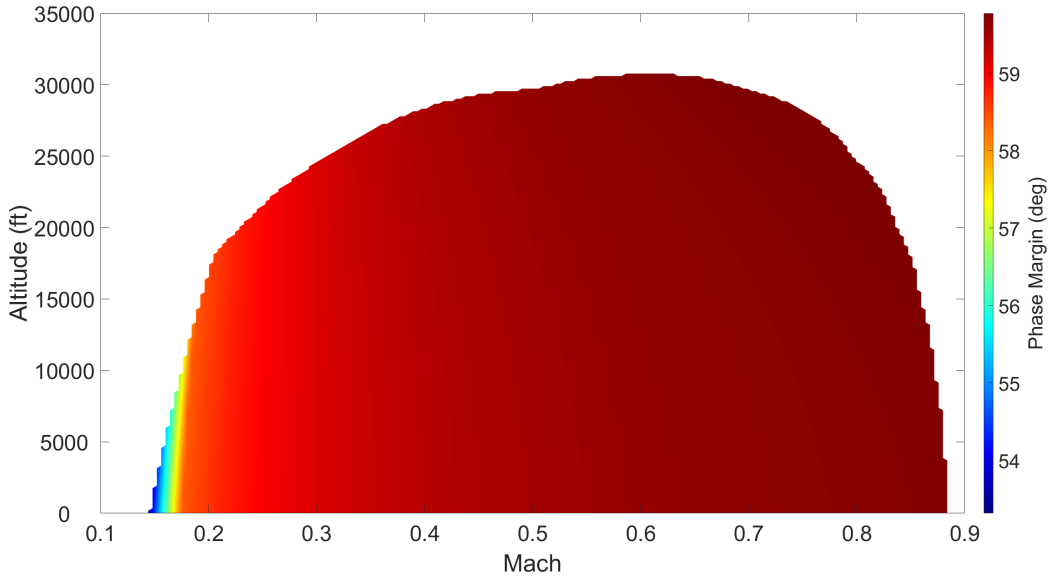


Figure 4.23: Variation of phase margin using automated gain scheduling

Phase margin is near the theoretical value of 60 degrees of phase using automated gain scheduling. There was not much room for improvement since the lowest phase margin using a single gain matrix was 59.2 degrees. One important difference was introduced using automated gain scheduling opposed to a single gain matrix. Phase margin degrades slightly at lower Mach numbers. This degradation is not severe with a minimum value of 54.5 degrees, but it is still important to understand. A probable explanation involves the character of the R matrix. In preceding sections, it was said that the guaranteed stability margins for LQR were only guaranteed for a diagonal R matrix. Although one of the design goals was to achieve a near diagonal matrix, term by term comparison introduces off-diagonal terms. The change in flight condition has likely made these terms sizable due to their dependence on the aircraft states and control vector. While this degradation is

important to discuss, this figure confirms that the closed-loop system will remain stable and non-oscillatory in the presence of phase shift introduced through modeling uncertainty.

The reason automated gain scheduling is being investigated is to create more consistent response characteristics at the off-design flight conditions. Therefore, it is important to investigate the change in step response caused by the updated gain matrix. Figures 4.24 - 4.26 show the step response in altitude, speed, and heading at a takeoff flight condition using the gain matrix designed for the cruise point.

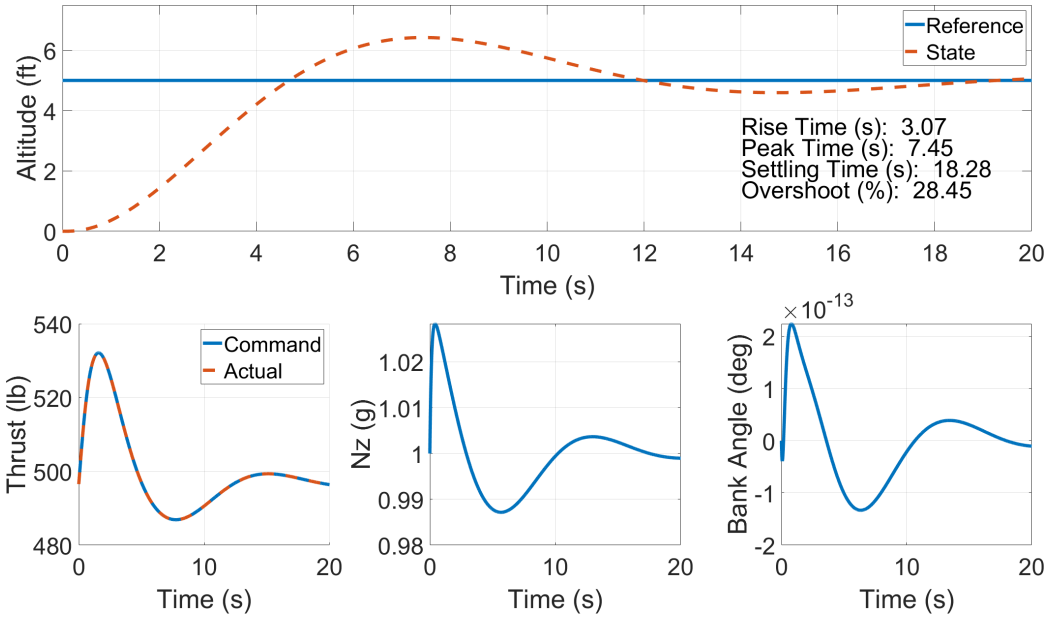


Figure 4.24: Altitude step response and control effort with cruise gain matrix and initial condition: $M = 0.2$, $h = 0$ feet

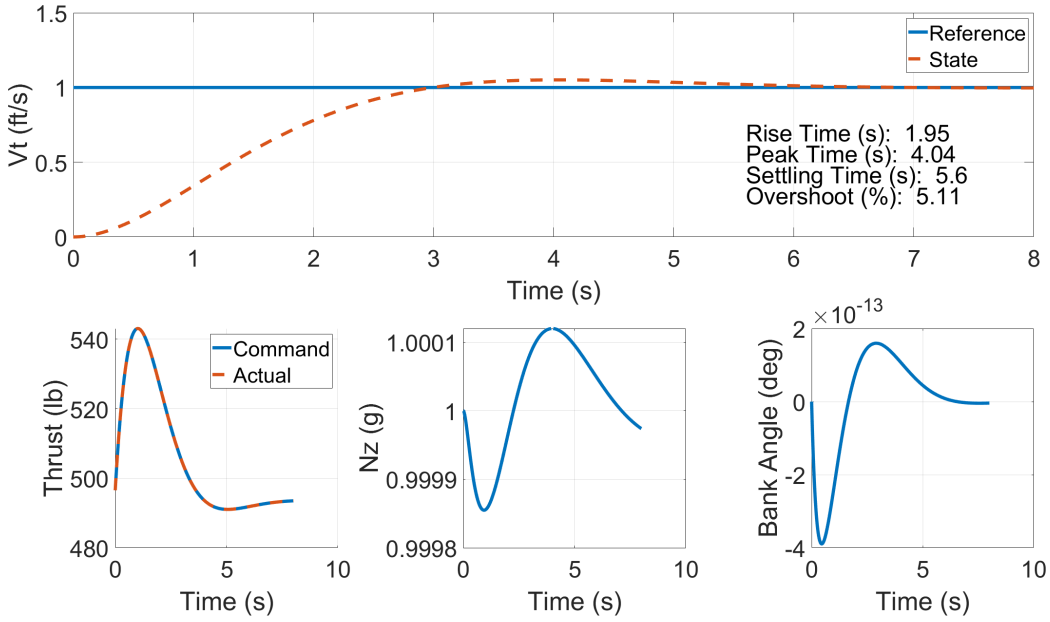


Figure 4.25: Velocity step response and control effort with cruise gain matrix and initial condition: $M = 0.2$, $h = 0$ feet

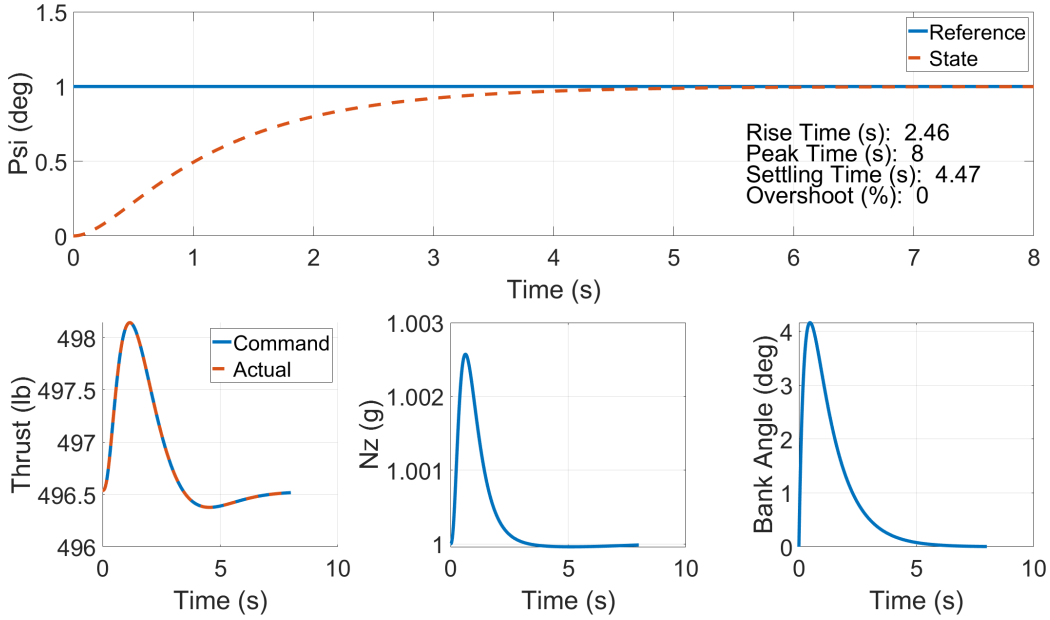


Figure 4.26: Heading step response and control effort with cruise gain matrix and initial condition: $M = 0.2$, $h = 0$ feet

One of the three tracking loops shows good performance using the cruise point gain matrix at the off-design point. The original design goal of the closed-loop system was to achieve a rise time of two seconds. The velocity tracking loop has a rise time of 1.95

seconds, which is nearly identical to the cruise point. The heading tracking loops shows undershoot instead of overshoot in the response to the step command, which is a characteristic of an overdamped response. The altitude tracking loop shows the opposite performance as the heading tracking loop. It has a response that is far too oscillatory, with an overshoot of nearly 30 percent. This underdamped altitude response has a settling time that went beyond the scope of the original step input test, so the end time had to be increased to 20 seconds. The response of the system using the cruise point gain matrix is fine for tracking the design mission, shown in Section 4.3. However, the step response is quite different than the original design goal for the system. The effects of automated gain scheduling are analyzed in Figures 4.27 - 4.29 by using the gain matrix consistent with the flight condition, instead of the cruise point gain matrix.

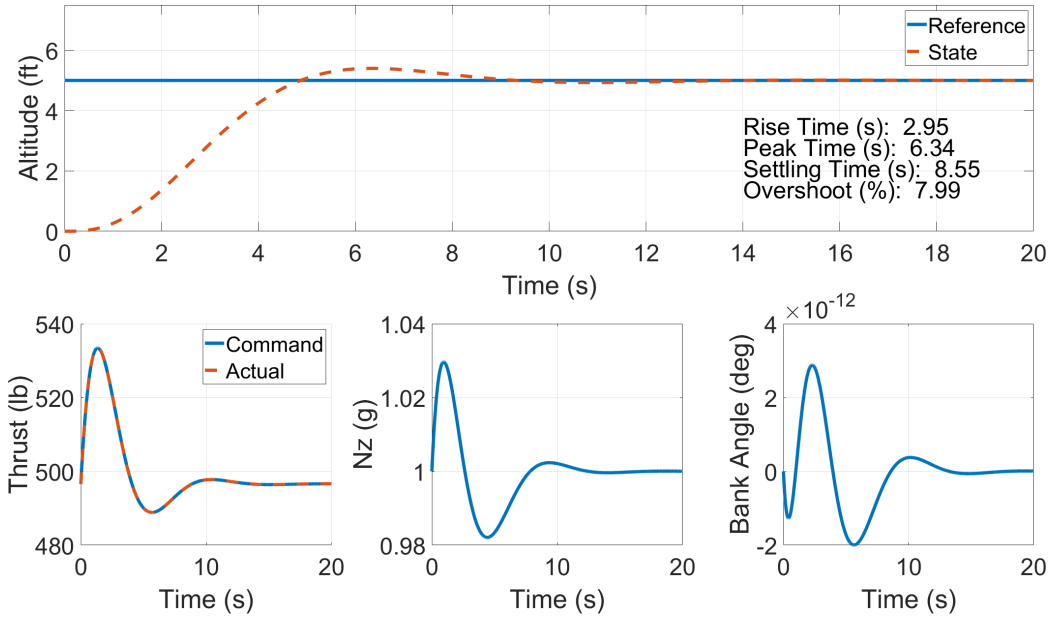


Figure 4.27: Altitude step response and control effort with cruise automated gain scheduling and initial condition: $M = 0.2$, $h = 0$ feet

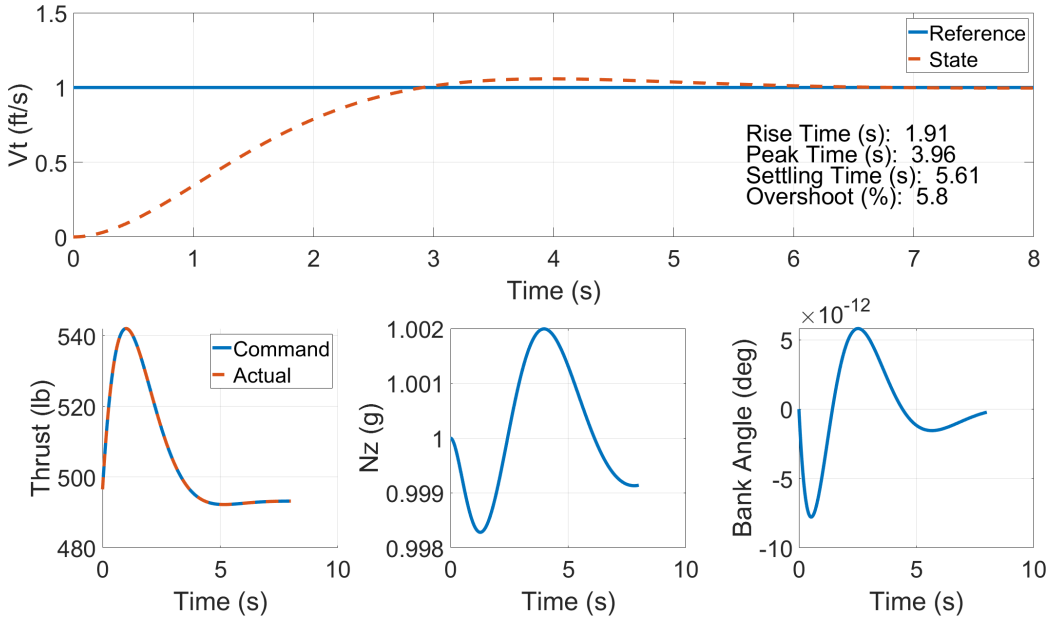


Figure 4.28: Velocity step response and control effort with automated gain scheduling and initial condition: $M = 0.2$, $h = 0$ feet

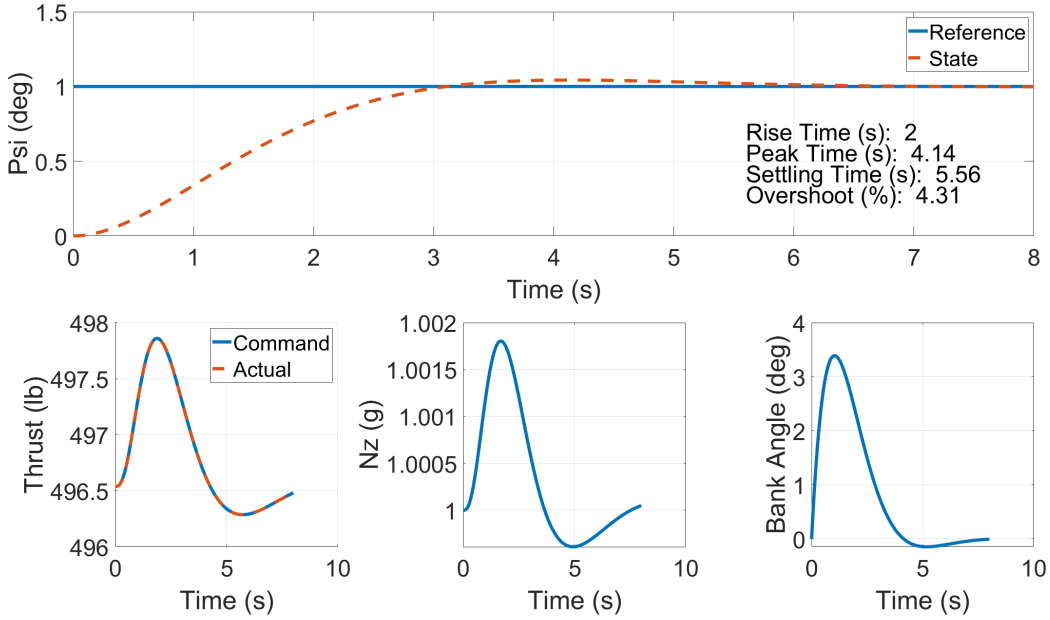


Figure 4.29: Heading step response and control effort with automated gain scheduling and initial condition: $M = 0.2$, $h = 0$ feet

The use of automated gain scheduling improves the step response performance in the altitude and heading tracking loops. The velocity response is very close to using the cruise point gain matrix. The heading response is changed from the overdamped response shown

in Figure 4.26 to a slightly underdamped response showing around four percent overshoot. The altitude response shows the most improvement of the three tracking loops. While the rise time is still higher than the design goal of two seconds, the response is significantly less oscillatory with an overshoot of about eight percent. This improvement in response shows that, while automated gain scheduling may not guarantee that the step response is identical everywhere in the flight envelope, it does a good job making the response more consistent considering the system was only tuned at one design point.

The control system designer could gain schedule, in the more traditional sense of the method, by tuning the controller via the α weights at the set of five reference flight conditions. The tuning could then be paired with this automated method by interpolating the set of α gains depending on flight condition. This process would still have a significantly lower design burden than tuning a set of nested PID control loops. Since limiting user design burden is one of the main goals of this thesis, the automated method presented here is valuable because it shows a fairly consistent response with no user design requirement and allows for fine tuning by implementing traditional gain scheduling techniques on a small set of gains.

5. Conclusions and Recommendations for Future Work

5.1 Conclusions

The focus of this project was to develop a closed-loop guidance controller tailored to constructive simulation applications. The reduced order aircraft model is the target of this work as it is well suited to the constructive simulation studies discussed in Chapter 1, weighing a balance of modeling fidelity and run-time performance. The ideal solution for the guidance controller is to develop a controller that requires no user tuning to achieve sufficient tracking performance of the design mission, but allows user tuning if finer control over the response is desired. The LQR guidance controller developed satisfies this requirement. The controller was shown to track the reference command in all three tracking loops using unity gains for the α terms. Later, the controller was tuned using a trial-and-error process to meet the time domain step response metric, rise time. An additional goal for the controller was to guarantee the closed-loop system remains stable throughout the flight envelope. Widely used MIMO stability margin methods were taken from the literature and applied to the reduced order aircraft model. Though this analysis, stability was maintained over the flight envelope using a single gain matrix obtained at the cruise point. This means the closed-loop system is both robust and tunable, which are the two most critical objectives of this work.

The development of a controller that satisfies these objectives can be categorized into four categories,

1. State Space Modeling
2. Cost Function Design
3. Inclusion of Feedforward Control
4. Automated Gain Scheduling

The creation of a state space model is a necessary condition for implementing an LQR control strategy. For the nonlinear equations of motion for the reduced order aircraft model, the equations were linearized about a cruise condition to allow for state space representation. The linearization was achieved using a first order Taylor series expansion of the equations of motion. Engineering judgment was used to eliminate terms in the state space model based on the data available and the physical meaning of the terms. The steady, level, trimmed flight condition was then applied to create a relatively compact state space model that is easily understood by the user.

A methodology was developed to reduce the complexity of the LQR cost function. Automated term-by-term comparison was used to implicitly define the cost function weights as opposed to directly designing the elements of the LQR weighting matrices. This method created a systematic way to include specific equations in the cost function and weighting them on a per equation basis by way of the α weights. Further engineering judgment was used to reduce the number of weights from thirteen to six – two for each tracking loop – making the cost function weighting tractable. It also allowed for tuning the controller using a trial-and-error method based on time domain step response metrics.

The high distribution of control effort placed on the LQR controller was addressed by including the analytic open-loop control equations in the form of a feedforward controller. This created a feedforward, feedback control architecture that placed the bulk of

control effort on the the feedforward component. The errors accrued due to the modeling assumptions were left to the LQR controller. The combined system tracked the reference commands with zero steady-state error.

Finally, to address concerns of unintended controller response at off-design flight conditions, a method to automatically schedule gains according to flight conditions was developed. The dependence of the LQR controller on the state space model allows the gain matrix to be updated according to flight conditions. This is called automated gain scheduling because it is provided at no additional design burden to the user. The results showed that using automated gain scheduling performed much closer to the time domain design goals at off-design points than using a single gain matrix.

5.2 Future Work

The controller developed in this thesis could currently be deployed to the constructive simulation applications highlighted in its design goals. However, multiple refinements have emerged during this research and have potential to make improvements on the controller. If work were to be continued on this topic, six important points of interest would be:

1. Compensator Design: One design goal for the developed LQR controller was to track the reference command vector with zero steady-state error for a constant command. The design mission tracking results show that the controller achieves this design goal, only accruing error as the reference command vector changes. While the accrued error is small and bounded for a ramp command, the user may desire tracking the ramp commands of the design mission with zero steady-state error. Achieving this new design goal requires development of the compensator of the controller. Currently, the compensator contains a set of three integrators responsible for closing steady-state error on a constant command, one for each tracking loop. Tracking the ramp command with zero steady-state error would

require some adjustment of the compensator design: the addition of three more integrators. Figure 5.1 shows the updated controller architecture required to track a ramp command.

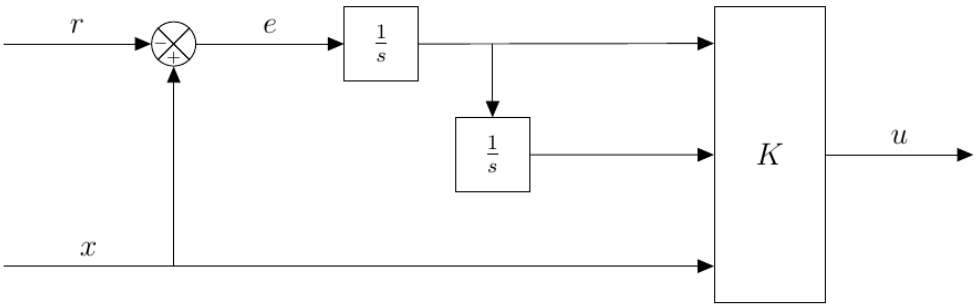


Figure 5.1: LQR controller architecture with ramp compensation

The addition of three integrators in the compensator would also show up as three new control equations in the closed loop state space model. The cost function design methodology does not currently account for these extra integrator states. It is probable that the number of α weights rise to 19 with this change. Although this proposed design change has not been extensively tested, some results do exist as a proof of concept. Figures 5.2 and 5.3 show the ramp command tracking improvement as a result of the updated compensation. The example application is the altitude tracking loop of the reduced order aircraft model. The controller is tuned to have a slower response to highlight the improvement of the updated compensator.

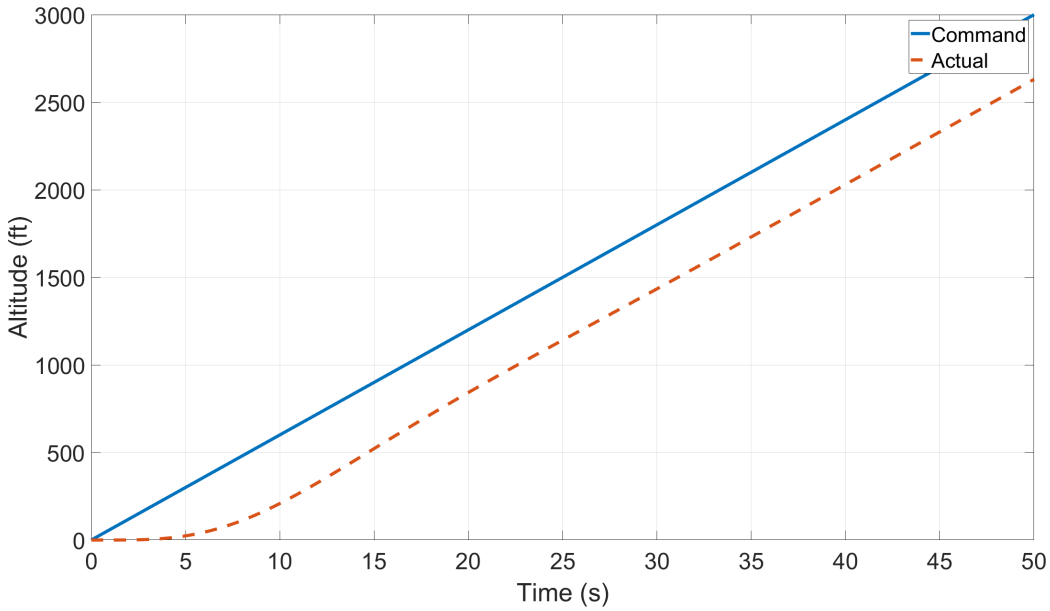


Figure 5.2: Tracking performance using original compensator

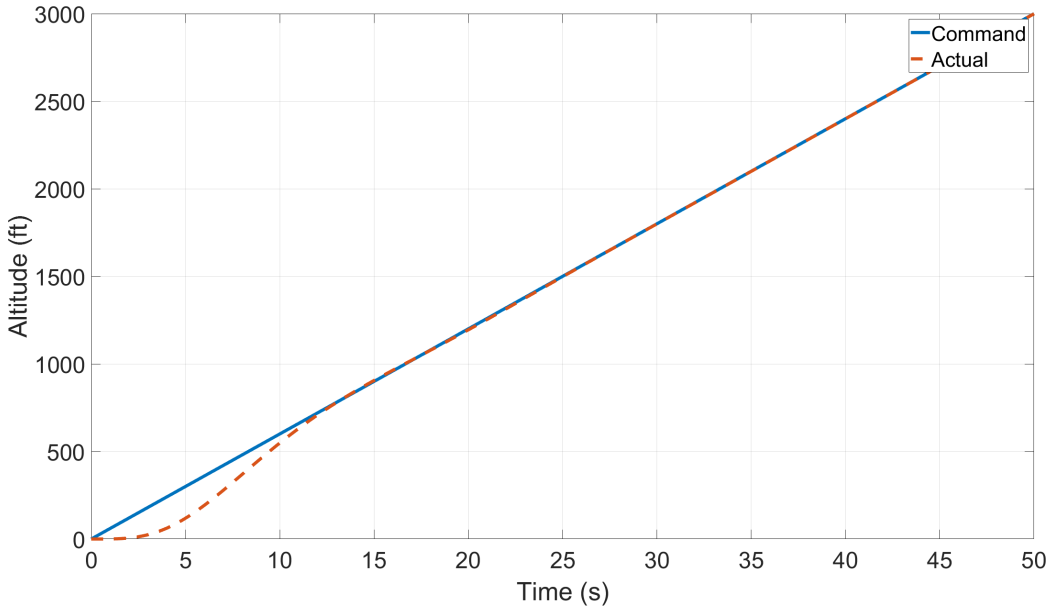


Figure 5.3: Tracking performance using new compensator

As expected, the steady-state error for the ramp command is regulated to zero with the addition of three compensator integral states. The design mission tracking performance would surely be improved with this change. But, the design mission is tracked accurately without these integral states. Therefore, the control system designer would have to decide

if the improved ramp tracking is worth increasing the order of the closed-loop system by three.

2. Feedforward Controller Development: The feedforward controller used for the reduced order aircraft model utilize the control equations for the zero angle of attack assumption. These equations were used because they are an analytic expression for the control vector, meaning no numerical solver is required. However, the more accurate equations of motion, the ones used in the model plant, include angle of attack. This difference in modeling assumption creates an inconsistency in the model. The implication of this inconsistency emerged as the control distribution was analyzed for the combined feedforward, feedback system. The feedforward control vector consistently overestimated the thrust and lift required to achieve the reference command vector. An improvement to the closed-loop system would be to use the feedforward control vector consistent with the equations of motion in the plant of the model. The drawback of this approach is the associated control equations for the equations of motion with angle of attack are not analytically solvable. Therefore, a numerical solver must be implemented to compute the consistent control vector. The Newton-Raphson method is a suitable choice of an iterator that will converge on the correct control vector. This choice is made because the method requires knowledge of the function and its derivatives, both of which are readily available. The Newton-Raphson method is known to converge very quickly, so it may be enough to run a single iteration for each time step of the model. The problem with using a numerical solver is the solution may diverge. This work has shown that the zero angle of attack analytic equations are good enough to track the design mission accurately. Therefore, the control vector should be reset using those analytic equation in the case of divergence. The use of the equations of motion with angle of attack, together with the Newton-Raphson numerical solver, should provide a much better approximation of the required feedforward control vector. This improvement has potential to further decrease the control effort required by the closed-loop

LQR controller. It is also likely that for application of the model in the terminal area, the resolving the correct angle of attack could be of greater importance than for up-and-away flight simulation.

3. Integrator Wind-up: The compensator designed for the closed-loop system included three integrators to ensure zero steady-state error in the three tracking loops. Whenever integrators are included in the controller, a discussion on integrator wind-up must occur. Near the design points of the controller, integrator wind-up is not likely an issue because the performance limits of the aircraft are not tested. At off-design points or in aggressive maneuvers, the controller may issue a control vector that is not achievable due to aerodynamic or engine performance limits. The implication is the error will not be closed and the integrator state will keep rising (wind-up), which can eventually destabilize the system. Three possible solutions to the integral wind-up problem are limiting the controller output, limiting the integral state to produce a reasonable output, or resetting the integral state during control saturation. A controller output limit makes sense for the reduced order aircraft model because the engine power limits are available via a table lookup and the aerodynamic limits are available through the drag polar and angle of attack curve fits. Adding protection against integral wind-up would give further merit to the stability robustness claims made in this thesis.

4. Additional Guidance Modes: The design mission for the reduced order aircraft model consisted of smoothed time vectors of Mach, altitude, and heading. This guidance mode was created to simplify the reference commands to only include one position command (altitude). It is certainly feasible to re-design the controller to track speed and position data required for trajectory tracking or waypoint following. Another approach would be to process the trajectory or waypoint data to generate equivalent speed, altitude, and heading commands to ensure accurate tracking. With either method, adding more guidance modes

that the controller can accept would make this guidance controller better suited for the constructive simulation studies that motivated its creation.

5. Gain Optimization: The response design goal for the closed-loop system was to achieve a rise time of two seconds in response to a step command. This design goal is one of the four time domain step response performance metrics discussed; the others are peak time, settling time, and overshoot percentage. It is not possible to satisfy design goals of multiple performance metrics using only two gains for each feedback loop. Introducing more gains, however, makes the problem less tractable for the system designer. A solution to this problem would be the use of an optimization routine. One example method would be a gradient based method. There is no function that can be evaluated in this case, meaning the values would be experimentally determined using the step response of the system. The gradient would then be numerically calculated with respect to each gain using a finite difference scheme. This method would make the tuning process easier on the user because they would only need to define the time domain performance metrics, instead of tuning the gains via trial-and-error. Writing an optimization scheme to automatically select all thirteen α parameters is a very interesting possibility that may be pursued in the future.

6. Case Studies: The results provided in this thesis were obtained for a fictitious fixed wing UAV with a 3100 pound gross weight. One example platform does not begin to capture the variations of aircraft size and performance present within the class of fixed wing aircraft. Case studies investigating propeller driven transport aircraft (Ilyushin IL-14) or agile fighter aircraft capable of supersonic flight (General Dynamics F-16) would begin to broaden the scope of this research to a wider range of fixed wing aircraft.

The main design goals for the controllers developed in this thesis were to meet the needs of constructive simulation; this generally translates to maximizing the run-time per-

formance while minimizing required user input to instantiate a model. Refinements such as inclusion of the feedforward controller appear to reduce the required user input while marginally hindering run-time performance. However, including the feedforward signal means that the LQR feedback controller works on a much smaller error, making the gain scheduling less important for tracking so long as stability is maintained. Improvements, such as this one, have the potential to improve the design goals for constructive simulation activities and overall tracking performance of the system.

Bibliography

[1] C. S. Draper and W. Wigley, *Apollo Guidance and Navigation*. Massachusetts Institute of Technology, June 1965.

[2] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th ed. Pearson, 2015.

[3] A. Tewari, *Advanced Control of Aircraft, Spacecraft and Rockets*. John Wiley & Sons, Ltd., 2011.

[4] K. Shimmin, "An architecture for rapid modeling and simulation of an air-vehicle system," Master's thesis, University of Dayton, 2016.

[5] J. Brendlinger, "Development of guidance laws for a reduced order dynamic aircraft model," Master's thesis, Wright State University, 2017.

[6] M. Athans and P. Falb, OPTIMAL CONTROL, *An Introduction to the Theory and Its Applications*. Dover Publications, Inc., 2007 [originally published in 1966].

[7] A. E. Bryson and Y.-C. Ho, APPLIED OPTIMAL CONTROL, *Optimization, Estimation and Control*. John Wiley & Sons, 1973.

[8] D. E. Kirk, OPTIMAL CONTROL THEORY *An Introduction*. Dover Publications, Inc., 1998 [originally published in 1970].

[9] *Control System Toolbox: User's Guide*, MathWorks, 2021.

[10] E. Cheever, *Transformation: Transfer Function - State Space*, Swarthmore College.

[11] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control with Aerospace Applications*. Springer-Verlag, 2013.

[12] D. B. Ridgely and S. S. Banda, "Introduction to robust multivariable control," Air Force Wright Aeronautical Laboratories, Tech. Rep., 1986.

[13] J. D. Robinson, "A linear quadratic regulator weight selection algorithm for robust pole assignment," Master's thesis, Air Force Institute of Technology, 1990.

[14] R. C. Nelson, *Flight Stability and Automatic Control*. McGraw-Hill Book Company, 1989.

[15] D. Tilbury and B. Messner, *Control Tutorials for Matlab and Simulink*, University of Michigan, National Science Foundation, 1996.

[16] J. D. Anderson, *Introduction to Flight*, 2nd ed. McGraw-Hill, 1985.

[17] V. Bragin, "New IL-14 much improved over IL-12," *Aviation Week*, April 22, 1957.

[18] J. Roskam, *Aircraft Design, Parts I-VIII*. Design, Analysis and Research Corporation, 2015.

[19] D. P. Raymer, *Aircraft Design: A conceptual Approach*, 6^{th} ed. AIAA Education Series, 2018.

[20] J. D. Anderson, *Aircraft Performance and Design*. McGraw-Hill, 1999.

[21] W. F. Phillips, *Mechanics of Flight*, 2nd ed. John Wiley & Sons, Inc., 2010.

[22] J. E. Williams and S. R. Vukelich, "The USAF stability and control digital DATCOM," AFRL, Tech. Rep., November 1979.

[23] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2nd ed. John Wiley & Sons, Inc., 2003.

[24] L. A. Weitz, "Derivation of a point-mass aircraft model used for fast-time simulation," MITRE, Tech. Rep., 2015.

[25] C. Chen and A. Holohan, "A revisit of the stability robustness of linear quadratic regulators," in *25th Irish Signals and Systems Conference*, 2014.