11-1-2007

Dynamic inversion of underactuated systems via squaring transformation matrix

Ryan Schkoda

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
Dynamic Inversion of Underactuated Systems Via Squaring Transformation Matrix

By

Ryan F. Schkoda

A Thesis Submitted in Partial Fulfillment of the Requirement for Master of Science in Mechanical Engineering

Approved by:

Dr. Agamemnon Crassidis – Thesis Advisor  
Department of Mechanical Engineering

Dr. Steve Weinstein  
Department of Mechanical Engineering

Dr. Mark Kempski  
Department of Mechanical Engineering

Dr. Edward Hensel  
Department Head of Mechanical Engineering
Dynamic Inversion of Underactuated Systems Via Squaring Transformation Matrix

I, RYAN F. SCHKODA, hereby grant permission to the Wallace Memorial Library of Rochester Institute of Technology to reproduce my thesis in the whole or part. Any reproduction will not be for commercial use or profit.

Date: ___________________________    Signature: ___________________________

November 2007
Abstract

In this thesis, a novel method for control of non-square dynamical systems using a model-following approach is developed. Control methodologies such as dynamic inversion and sliding mode control require an inversion of the input influence matrix. However, if the system input influence matrix is non-square direct inversion is not possible. Pseudo inversion of the input influence matrix may be performed for control allocation. However, pseudo inversion limits the control to states where the controller is directly applied. The pseudoinverse method does not permit the engineer to designate a particular state to control or track. When accurate tracking of states that are not directly controlled (“remaining states”) is required the pseudo inversion method is not useful. Current methods such as dynamic extension can be used to generate a square input influence matrix, essentially, creating an input influence matrix that is invertible. However, this method is tedious for large systems. In this work, a new transformation is applied to the original dynamical system model to develop an input influence matrix that is square. Assuming the system is controllable, the proposed transformation allows for accurate tracking of selectable states. Selection of the new transformation matrix is used to develop accurate tracking of certain states compared to the remaining states. A method based on optimal control theory is used to define the transformation matrix. The new approach is first applied to control a two mass system with simulation results presented showing the advantage of the proposed new control strategy. Finally, simulation results are presented for longitudinal control of an aircraft using one control input.
I would like to take this opportunity to thank Dr. Agamemnon Crassidis for his patience and willingness to have the same conversation more than once. Also, Dr. Steve Weinstein and Dr. Mark Kempski have been peers, colleagues and advisers in the development of this paper. Thank you all.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
</tr>
<tr>
<td>Acknowledgments</td>
</tr>
<tr>
<td>Table of Contents</td>
</tr>
<tr>
<td>List of Figures</td>
</tr>
<tr>
<td>List of Tables</td>
</tr>
<tr>
<td>Nomenclature</td>
</tr>
</tbody>
</table>

1 Introduction

1.1 Background ...................................................... 1
1.2 Current Work .................................................... 2
1.3 Overview and Motivation for Present Work ....................... 5

2 Theoretical Development

2.1 Lyapunov Theory ............................................... 7
   Preface .................................................................. 7
   2.1.1 Phase Portrait .............................................. 7
   2.1.2 Definitions of Stability .................................... 8
2.2 Sliding Mode Control ............................................. 11
   Preface .................................................................. 11
   2.2.1 Sliding Surface .............................................. 11
   2.2.2 Surface Stability ............................................ 12
   2.2.3 Controller Development .................................... 14
2.3 The Linear Quadratic Regulator ................................ 15
   Preface .................................................................. 15
   2.3.1 Final State Free Boundary Condition ...................... 15
   2.3.2 The Steady-State Solution .................................. 17
2.4 The Transform ..................................................... 17
2.5 The Pseudoinverse ............................................... 19
   Preface .................................................................. 19
   2.5.1 Relation to the True Inverse ............................... 20
   2.5.2 Square, Overactuated and Underactuated Systems ....... 20
2.5.3 The Moore-Penrose Pseudoinverse .................................... 22
2.5.4 Relationship Between the Transform and the Pseudoinverse ... 23

2.6 Dynamic Extension ............................................................. 23

2.7 The Solution ........................................................................... 26

2.7.1 Solution One: Use of the Moore-Penrose Pseudoinverse .... 26
2.7.2 Sliding Controller vs. Suboptimal Feedback .................... 26
2.7.3 Solution Two: Of the Form (TB)^{-1}T ................................ 27
2.7.4 Solution Three: Of the Form (T,B)^{-1}T .......................... 28
2.7.5 Summary ........................................................................... 32

2.8 Tracking Performance ............................................................ 32

2.9 Linearization and Model Replacement .................................... 34

2.9.1 Linearization ..................................................................... 34
2.9.2 Linear Model Validation and Replacement .................. 36

3 Results ...................................................................................... 41

3.1 Two Mass, Two Spring, Two Damper System ......................... 41

3.2 Longitudinal Aircraft Model ................................................... 48

3.2.1 Use of Moore-Penrose Pseudoinverse for Tracking ........ 48
3.2.2 Dominant Weighting of State One ................................. 49
3.2.3 Dominant Weighting of State Two ............................... 52
3.2.4 Dominant Weighting of State Four ............................... 54

4 Conclusion, Discussion and Future Work ............................... 57

4.1 Conclusion ............................................................................ 57
4.2 Discussion ............................................................................ 58
4.3 Future Work ......................................................................... 61

Bibliography .................................................................................. 63

A Dynamic Extension Example .................................................... 65

Preface .......................................................................................... 65
# List of Figures

2.1 Example of a 2-D phase portrait ........................................ 8
2.2 Concepts of stability ....................................................... 9
2.3 Sliding surface behavior ................................................. 14
2.4 Linear vs. Nonlinear Response: 0.1 Deg Deflection .................. 37
2.5 Linear vs. Nonlinear Response: 0.2 Deg Deflection .................. 38
2.6 Linear vs. Nonlinear Response: 0.3 Deg Deflection .................. 39
2.7 Linear vs. Nonlinear Response: 0.4 Deg Deflection .................. 39
2.8 Linear vs. Nonlinear Response: 0.5 Deg Deflection .................. 40
3.1 Two-Mass System Schematic ............................................. 42
3.2 Tracking Response: Moore-Penrose ..................................... 43
3.3 Tracking Response: State One .......................................... 44
3.4 Control effort expenditures: State One .................................. 45
3.5 Tracking Response: State Three ........................................ 46
3.6 Control effort expenditures: State Three ................................ 47
3.7 Tracking Response: Moore-Penrose ..................................... 49
3.8 Tracking Response: State One .......................................... 50
3.9 Control effort expenditures: State One .................................. 51
3.10 Tracking Response: State Two .......................................... 52
3.11 Control effort expenditures: State Two ................................ 53
3.12 Tracking Response: State Four ........................................ 55
3.13 Control effort expenditures: State Four .............................. 55
# List of Tables

2.1 Solution Summary ........................................... 32

3.1 Dynamic coefficients and eigenvalues: State One .................. 45
3.2 Dynamic coefficients and eigenvalues: State Three ............... 48
3.3 Dynamic coefficients and eigenvalues: State One .................. 51
3.4 Dynamic coefficients and eigenvalues: State Two ............... 54
3.5 Dynamic coefficients and eigenvalues: State Four ............... 56
Nomenclature

A  System Dynamic Matrix
B  Input Influence Matrix
B†  Moore-Penrose pseudoinverse of B
C  Output Matrix
x  State Vector
T  Transformation Matrix
T*  Alternate form of Transformation Matrix
Q  State Weighting Matrix
R  Input Weighting Matrix
s  Sliding Surface
λ  Positive Constant
λ  Lagrange Multiplier (Section 2.3)
u  Control Input
y  Transformed State Vector
J  Cost Function
K(t)  Solution to Linear Quadratic Regulator
K  Steady-State Solution to Linear Quadratic Regulator
Vt  True Velocity
α  Angle of Attack
p  Roll Rate
q  Pitch Rate
r  Yaw Rate
ϕ  Roll/Bank Angle
θ  Pitch Angle
ψ  Yaw/Heading Angle
V(x)  Lyapunov Function (of variable x)
x_d  Subscript (d) denotes desired value (i.e. desired value of x)
¯x  Difference between x and x_d (i.e. state error, x − x_d)
\(n\) System Order/Number of States

\(m\) Number of Inputs

\(p\) Number of Outputs

\(H\) Hamiltonian

\(S(t)\) Solution to the Riccati equation

\(S(\infty)\) Steady-State Solution to the Riccati equation

\(s_r\) Characteristic Root

\(j\) Unknown matrix that is to be determined

\(S(t_f)\) Final state weighting matrix for the LQR problem (Section 2.3.1)

\(F_{A_x}, F_{A_y} \& F_{A_z}\) Aerodynamic forces along the aircraft body-axis

\(F_{T_x}, F_{T_y} \& T_{A_z}\) Thrust forces along the aircraft body-axis

\(M_{e_x}, M_{e_y} \& M_{e_z}\) External applied moments about the aircraft body-axis

Note: Typical matrix notation is employed. A lowercase, italic variable indicates a scalar. A lowercase, bold variable indicates a vector. An uppercase, bold variable indicates a matrix.
Chapter 1

Introduction

1.1 Background

Classical linear control architectures such as proportional, proportional-integral (PI) and proportional-integral-derivative (PID) are powerful techniques for controlling a wide range of systems. However, with the advent of more complex and nonlinear systems there is a need for more advanced control schemes capable of controlling such systems.

The development of state-space methods is perhaps one of the greatest advances in the controls community in the past sixty years. The defining characteristic of the state-space form is the resulting mathematical system model is a system of first-order differential equations rather than higher-order differential equations characteristic of transfer function analysis. Individuals such as Professor Solomon Lefschetz, Professor J.R. Ragazzini, R.E. Kalman and J.E. Bertram among others as well as many scientists from the Soviet Union were responsible for bringing state-space methods to the forefront in the late 1950s and early 1960s [1].

Not only did state-space methods provide a different way of analyzing dynamical systems but numerical techniques were available to approximate the time history response of a system in state-space form. Numerical integration techniques such as the Euler and Runge-Kutta methods are particularly well suited to approximate the solution of a system in state-space form. Also, the introduction of digital computers further progressed calculation of these solutions. Without the aid of digital computers, solving a state-space model would be nothing more than an academic exercise.

One of a number of control schemes applicable to state-space models is sliding mode control (SMC). The advantage of SMC is the ability to account for parametric uncertainty in the system model as well as being applicable to nonlinear systems. SMC is the control scheme being investigated in this research.
1.2 Current Work

Non-square systems are common in nature as well as in engineering practice. If fact, when modeling systems in the “real world,” square systems may form the minority. For a system to be square the number of inputs must equal the system order. This is a completely arbitrary constraint and is rarely satisfied. Certain control architectures indirectly rely on a system being square. One such control architecture is sliding mode control. The development of a sliding controller requires an inversion of the system’s input influence matrix ($B$). If the system is non-square, the input influence matrix is singular and not invertible. There have been numerous attempts documented in literature to design viable control schemes for non-square systems. Most all of these schemes revolve around dynamic inversion based controllers.

After performing a literature review it became clear that topic of this thesis is a fairly novel idea. There is much literature concerning the control of non-square systems but a lack of literature related to the method being proposed in this thesis.

An automobile constitutes a non-square system. In a paper by Wang and Longoria [2] the problem of controlling vehicle chassis is investigated. Control inputs include wheel torque and steering actuation. A two stage control system is developed. Stage one involves developing a sliding controller to produce general forces and moments necessary and to control the vehicle and stage two utilizes a weighted pseudoinverse to determine control allocation. (For an definition and discussion on pseudoinverses and a weighted pseudoinverse see Section 2.5)

Non-square systems are also found in biological systems such as the muscular system designed to control and coordinate eye movement. A paper authored by Dean and Porrill [3] notes this problem of redundancy in both robotic and biological systems. For the control of oculomotor systems the authors use a PID controller utilizing a weighted pseudoinverse. Because eye movement is controlled by several thousand muscular actuators the basic control solution is not unique [3]. Once again a particular solution must be determined with the aid of the pseudoinverse.

Space and re-entry vehicles represent a class of non-square systems. In papers by Schierman et al [4] and Bolender et al [5] the problem of redundant control actuators is once again addressed. Both papers are concerned with implementing dynamic inversion based control laws for the control of redundantly actuated re-entry vehicles. The paper by Schierman et al determines a unique pseudoinverse by solving a mixed-optimization problem to minimize control effector displacement while avoiding rate or position saturation [4]. The paper by Bolender et al seeks to satisfy nearly identical constraints.
In the closely related field of unmanned or uninhabited air vehicles (UAVs) the problem of controlling non-square systems is addressed. Papers by Boskovic et al [6], Haitao & Jinyuan [7] and Boskovic & Mehra [8] all address the issue of tracking control for non-square aircraft models. All of these papers make use the weighted pseudoinverse to address the issue of control allocation. Boskovic et al seek to develop a weighting matrix such that it does not saturate the control surface actuators [6]. This weighting matrix is then used to form the pseudoinverse which in turn defines the control allocation. Haitao and Jinyuan proposed a control architecture utilizing traditional aircraft equations of motion, nonlinear mapping and a single hidden layer (SHL) neural network [7]. Once again the pseudoinverse is used to define control allocation. The pseudoinverse is defined such that it minimizes control energy. The paper by Boskovic and Mehra specifically addresses the issue of control allocation under position and rate limiting. The authors seek to find a more effective way of determine optimal and satisfactory control allocation.

In yet another application regime of underwater vehicles the problem of controlling non-square systems has been addressed. Papers by Omerdic et al [9] and Fossen & Johansen [10] specifically address the issue of control allocation. Both these papers assume the systems are overactuated and examine ways to efficiently control the craft based on certain criteria. Omerdic et al adapt a standard pseudoinverse method by combining it with a fixed-point iteration method. Fossen & Johansen produced a paper that was meant to be a survey of control allocation methods available for underwater vehicles. In section III there is specific reference to the generalized inverse and its relation to the Moore-Penrose pseudoinverse [10]. A derivation of Moore-Penrose using Lagrange multipliers is presented.

What is perhaps one of the most relevant discussions to this work is part of dissertation by Gordon G. Parker [11]. In Section 4.2.1 of the dissertation Parker outlines the usage of Sliding Mode Control (SMC) to control a general nonlinear system as an introduction to Section 4.2.2. In Section 4.2.2 Parker introduces a method he refers to as Augmented Sliding Mode Control (ASMC). In Section 4.2.1 Parker develops the sliding controller for the general nonlinear system shown in Eq. (1.1). The reader must note that to solve for the control law an inversion of the input influence matrix ($B$) is necessary. ASMC is presented as a way to form the sliding controller if the system is non-square and the input influence matrix is not invertible. The following is a paraphrasing of Section 4.2.2. (Note: Notation in this discussion has been adapted from the original text)
The general form of the nonlinear structural systems examined here is restricted to
\[ \ddot{x} = N(x, \dot{x}) + BU \] (1.1)
where \( x, \dot{x}, \ddot{x} \) are \( n \times 1 \) vectors of generalized coordinates and their first and second derivatives, respectively; \( N(x, \dot{x}) \) is an \( n \times 1 \) vector of nonlinear functions; \( B \) is an \( (m + 1) \times 1 \) matrix of control input weighting coefficients and \( U \) is a scalar.

The state vector is
\[ \bar{x}^T = [\theta, q_1, \ldots, q_m] \] (1.2)
where \( \theta \) is the hub rotation (Parker’s work dealt with mechanical rotation) and \( q_1 \) through \( q_m \) represent the remaining states. Therefore, there is one rotational equation of motion that can be extracted from Eq. (1.1) is
\[ \ddot{\theta} = N_1 + B_1 U \] (1.3)
where \( N_1 \) is the nonlinear rigid body equation of motion and \( B_1 \) is the input gain. Equation (1.3) is a scalar equation with \( U \) as the input and \( \ddot{\theta} \) as the output. However, \( N_1 \) is a nonlinear equation involving the remaining states.

Following from this the sliding surface may be chosen as
\[ s = w \left( \theta_e + \bar{w}^T \bar{q}_e \right) + \left( \dot{\theta}_e + \bar{w}^T \dot{\bar{q}}_e \right) = 0 \]
\[ \theta_e \equiv \theta - \theta_{ref} \]
\[ q_e \equiv q - q_{ref} \] (1.4)

Where \( w \) is a constant and \( \bar{w} \) is a \( 1 \times m \) vector of weighting coefficients. By defining the sliding surface as it has been defined in Eq. (1.4) it becomes a scalar equation where the dynamic variable is not state error (as it is in SMC) but a weighted sum of all the state errors. Now that both equations are in scalar form, the derivative of Eq. (1.4) may be taken and substituted into Eq. (1.3) to form the control law
\[ B_1 U = -w \left( \dot{\theta}_e + \bar{w}^T \dot{\bar{q}}_e \right) - N_1 + \ddot{\theta}_{ref} - \bar{w}^T \ddot{\bar{q}}_e \] (1.5)

The remaining problem is to determine the form of the constant \( w \) and the vector \( \bar{w} \). System performance will be defined by a nonlinear cost function \( J \). The parameters \( w \) and \( \bar{w} \) will be determined such that they minimize the cost function \( J \) as well as satisfy any other constraints.

In a paper by El Singaby [12] the “squaring” of a non-square system is addressed.
The author assumes the state-space model is as follows

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

(1.6)

It is desired to regulate the system with

\[ u = -Ex \]  

(1.7)

where \( E \) is a constant feedback gain matrix of the form

\[ E = (CB)^{-1}j \]  

(1.8)

Where \( j \) is an arbitrary diagonal matrix with specified eigenvalues. If the system outputs are defined as they are in Eq. (1.6) and the dimension of \( y \) is not equal to the dimension of \( x \) then the input-output relationship is non-square and the product \( CB \) in Eq. (1.8) cannot be inverted. El Singaby’s paper presents a method of redefining \( C \) as \( C_{sq} \) such that the product \( C_{sq}B \) is square and invertible.

The results reached by El Singaby are interesting because they are similar to the results developed in this thesis and reminiscent of the pseudoinverse. Throughout this thesis it is assumed that the states of the system are available or can be estimated with a high degree of certainty. This implies the \( C \) matrix is identity. The El Singaby paper does not make this assumption. Cases where there are fewer outputs than inputs \((p < m)\) and more outputs than inputs \((p > m)\) are addressed. In the case where \( p < m \) the system needs to be “squared up,” while for \( p > m \) the system needs to be “squared down.”

Although El Singaby’s paper proposes a matrix \( j \) responsible for closed loop pole placement there is no proposed method for choosing the matrix. Also, El Singaby’s paper does not attempt to track any desired states. The idea of squaring a system was investigated only for the purpose of the regulation, not tracking.

Additional methods for designing control schemes for overactuated models can be seen in Bakaric et al [13], Johansen [14] and Oppenheimer [15].

1.3 Overview and Motivation for Present Work

The following research is primarily concerned with the development of a sliding mode controller for a non-square, underactuated system. The biggest obstacle is determining what to do when the inverse of a singular matrix is requested. Currently a solution
may be developed by using what is known as a generalized inverse or pseudoinverse. However, the pseudoinverse is not unique and may not provide satisfactory results due to its form. By studying the problem and understanding the mathematics it may be possible to develop a satisfactory sliding controller despite the requirement of inverting a singular matrix.

The development and analysis of such a controller was done first by theoretical mathematical development then simulation in Simulink® and analysis in MATLAB®. The general structure for the research was as follows:

1. Develop a theoretical solution to overcome the difficulty of inverting a singular matrix
2. Simulate the dynamical response when the proposed control law is implemented
3. Analyze system properties of a closed-loop system

The developed methodology will be applied to the classical two-mass, two-spring, two-damper system model (Section 3.1) and then to a linearized longitudinal aircraft model (Section 3.2).
Chapter 2

Theoretical Development

2.1 Lyapunov Theory

Preface

When analyzing linear systems Laplace transforms and eigenstructure analysis can offer much insight into the stability of a system. However, Laplace transforms and eigenstructure analysis are not applicable when the system being analyzed is nonlinear. For nonlinear system analysis the theory developed by Aleksandr Mikhailovich Lyapunov is extremely useful. Some of Lyapunov’s work concerning nonlinear system stability can be more easily understood by examining the system’s phase portrait (For an example see Figure 2.1). Because each axis of a phase plane corresponds to a system state, the trace of the system states as the system propagates offers some insight as to the stability of the system. If for some initial condition $x(0)$ the state trajectories either remain in the vicinity of their initial location or tend toward the origin they are thought of as being well behaved, or stable. Conversely, if for some initial condition $x(0)$ the state trajectories tend toward infinity as the system propagates the system is unstable. The idea of phase plane behavior is used to characterize system stability.

2.1.1 Phase Portrait

The basic concept of stability analysis is to determine if, for a bounded input, the system output will remain bounded as well. For linear systems, eigenvalues directly show this character. Since nonlinear systems do not possess constant eigenvalues, it is necessary to develop another method for assessing system stability. Such methods make use of a phase portrait. The phase portrait is a method to simultaneously visualize all the state
trajectories of a system. Obviously, when physically constructing a phase portrait one is limited to three dimensions. However, a phase portrait may exist in any \( n \)-dimensional space.

Because the state variables define a dynamical system, their trajectories or behavior can offer insight to the character of the system. A phase portrait is a plot where each axis represents a state variable and the trace represents the time history response of the states.

Figure 2.1: Example of a 2-D phase portrait

Figure 2.1 displays an example of a 2-D phase portrait. The particular phase portrait describes a stable system where each state has some initial condition at \( x(0) \) and the final value of the states are at the origin, or 0. The trace defines the time history response of the state variable. The shape of the trace alone or the shape of the trace for various initial conditions may offer insight of system character.

### 2.1.2 Definitions of Stability

Consider the following form for a general nonlinear system

\[
\dot{x} = f(x, t)
\]  

(2.1)
where \( f \) is a \( n \times 1 \) nonlinear vector function, and \( x \) is the \( n \times 1 \) state vector. The definitions of stability in a Lyapunov sense are presented in Slotine [16]. Theory concerning sliding mode control relies heavily on an understanding of stability based on Lyapunov stability. The following definitions and discussion is meant to introduce the idea of Lyapunov stability.

**Definition 2.1.** The equilibrium state \( x = 0 \) is said to be *stable* if, for any \( R > 0 \), there exists \( r > 0 \), such that if \( \| x(0) \| < r \), then \( \| x(t) \| < R \) for all \( t \geq 0 \). Otherwise, the equilibrium point is *unstable*.

\[
\forall R > 0, \exists r > 0, \| x(0) \| < r \Rightarrow \forall t \geq 0, \| x(t) \| < R
\]

Definition 2.1 is perhaps the most basic definition concerning Lyapunov stability. Essentially, if the state trajectory is started arbitrarily close to an equilibrium point (defined by a ball with radius \( r \)), the state trajectory will stay in the vicinity of that equilibrium point (defined by a radius \( R \)). Figure 2.2 illustrates the idea of Lyapunov stability.

![Concepts of stability](image)

**Figure 2.2:** Concepts of stability

**Definition 2.2.** An equilibrium point \( 0 \) is *asymptotically stable* if it is stable, and if in addition there exists some \( r > 0 \) such that \( \| x(0) \| < r \) implies that \( x(t) \to 0 \) as \( t \to \infty \).

Definition 2.2 further restricts the concept of Lyapunov stability including the concept of asymptotic stability. Definition 2.1 makes no restriction on the limiting behavior of the state trajectory; only that the trajectory must remain arbitrarily close to the equilibrium point. If it can be shown that \( x(t) \to 0 \) as \( t \to \infty \) then the equilibrium point is said to be asymptotically stable. This is an important distinction. If it is known whether or not
a system is asymptotically stable there is some \textit{a priori} knowledge of how the systems trajectories will propagate.

**Definition 2.3.** An equilibrium point $0$ is \textit{exponentially stable} if there exists two strictly positive numbers $a$ and $b$ such that

$$\forall t > 0, \|x(t)\| \leq a\|x(0)\|e^{-bt}$$

in some ball $B_r$ around the origin.

Definition 2.3 restricts Definition 2.1 even further by defining not only whether $x(t) \to 0$ as $t \to \infty$ but how $x(t) \to 0$ as $t \to \infty$. If a system is determined to be exponentially stable its state trajectory is upper and lower bounded by $a\|x(0)\|e^{-bt}$.

**Definition 2.4.** If asymptotic (or exponential) stability holds for any initial states, the equilibrium point is said to be asymptotically (or exponentially) stable \textit{in the large}. It is also called \textit{globally} asymptotically (or exponentially) stable.

Definition 2.4 extends the definition of stability to include the entire surface for which the system is defined. The definitions presented prior to Definition 2.4 are all concerned with equilibrium points and assume $x(0)$ is arbitrarily close the the equilibrium point. If a system is determined to be globally asymptotically (or exponentially) stable then $x(t) \to 0$ as $t \to \infty$ for any and all $x(0)$.

The preceding definitions may now be used to define a \textit{Lyapunov function}.

**Definition 2.5.** If, in a ball $B_{R_0}$, the function $V(x)$ is positive definite and has continuous partial derivatives, and if its time derivative along any state trajectory of system (2.1) is negative semi-definite, i.e.

$$\dot{V}(x) \leq 0$$

then $V(x)$ is said to be a \textit{Lyapunov function} for the system (2.1).

**Theorem 2.6** (Local Stability). \textit{If, in a ball $B_{R_0}$, there exists a scalar function $V(x)$ with continuous first partial derivatives such that}

- $V(x)$ is positive definite (locally in $B_{R_0}$)
- $\dot{V}(x)$ is negative semi-definite (locally in $B_{R_0}$)

\textit{then the equilibrium point $0$ is stable. If, actually, the derivative $\dot{V}(x)$ is locally negative definite in $B_{R_0}$, then the stability is asymptotic.}
Theorem 2.6 may be extended to include the all possible state trajectories.

**Theorem 2.7 (Global Stability).** Assume that there exists a scalar function $V$ of the state $x$, with continuous first order derivatives such that

- $V(x)$ is positive definite
- $\dot{V}(x)$ is negative definite
- $V(x) \to \infty$ as $\|x\| \to \infty$

then the equilibrium at the origin is globally asymptotically stable.

The preceding definitions and theorems formalize the concept of stability from a Lyapunov standpoint. Linear systems have the advantage of possessing constant eigenvalues which offer information regarding system stability. Analysis based on Lyapunov theory allows for stability to be examined for nonlinear systems. The Lyapunov function $V(x)$ will be utilized in the development of what is called a sliding surface in Section 2.2.

### 2.2 Sliding Mode Control

**Preface**

The concept of sliding mode control seeks to reduce the dynamics of a general system to an asymptotically stable differential equation where the dynamic variable is tracking error. The control architecture is particularly useful because of its ability to control nonlinear systems and is robust to parametric uncertainty. The controller is designed to produce favorable state tracking. The following section introduces the fundamentals of sliding mode control as presented in Slotine [16].

#### 2.2.1 Sliding Surface

Sliding mode control centers around the concept of sliding surfaces. To illustrate the concept consider the following general system

$$x^{(n)} = f(x) + b(x)u \quad (2.2)$$

where the state vector $x$ is defined as $x = [x \; \dot{x} \; \cdots \; x^{(n-1)}]^T$, the $n^{th}$ derivative of the state vector is $x^{(n)}$, the system dynamics are defined by $f(x)$, the input influence matrix
is \( b(x) \) and the system input is \( u \). The goal of the control scheme is to track a desired, time-varying state vector defined as \( x_d = [x_d \dot{x}_d \cdots x_d^{(n-1)}]^T \).

Let the tracking error vector, \( \tilde{x} \), be defined as

\[
\tilde{x} = x - x_d = [\ddot{x} \dot{x} \cdots \ddot{x}^{(n-1)}]^T
\]

then define a time-varying surface, \( s(x; t) \), in the state-space, \( \mathbb{R}^{(n)} \), by

\[
s(x; t) = \left( \frac{d}{dt} + \lambda \right)^{(n-1)} \ddot{x}
\]

and \( \lambda \) is a strictly positive constant.

If, for instance, \( n = 2 \) the sliding surface would be defined as

\[
s(x; t) = \dot{\tilde{x}} + \lambda \tilde{x}
\]

or a weighted sum of all the state errors.

If the current state of the system satisfies \( s(x; t) = 0 \) then the error trajectories are said to be “on the sliding surface” and the error vector will approach zero according to the dynamics of the sliding surface. The situation is known as the “sliding phase.” Furthermore, if the condition

\[
x_d(0) = x(0)
\]

is satisfied then the error trajectories are at the origin and remain at the origin since the surface, \( s \), is constructed such that the origin is Lyapunov stable (See Section 2.2.2).

In the event \( s(x; t) \neq 0 \) the system is said to be in the “reaching phase.” If the error trajectories are not on the sliding surface it is necessary to show that they will tend toward the sliding surface. Section 2.2.2 addresses the solution to this potential problem utilizing Lyapunov stability.

Considering the preceding discussion it is clear that the sliding surface is both a place and a dynamic [16]. The method of sliding mode control develops a controller causing the closed loop dynamics to be that of the sliding surface; ensuring favorable tracking.

### 2.2.2 Surface Stability

The particular sliding surface used to develop a sliding controller is not unique. This means the engineer has some liberty in terms of what the form of the sliding surface will be. Regardless of the surface form it must be shown that its magnitude is stable from a Lyapunov standpoint. Because the objective of the sliding controller is to approach
$s(x; t) = 0$ when $s(x; t) \neq 0$, and to stay on $s(x; t) = 0$ once on it, the candidate Lyapunov function is chosen to be

$$V(x) = \frac{1}{2} s^2 \quad (2.6)$$

Conceptually, $s(x; t)$ defines a surface in the state-space. The desired location on the surface is $s(x; t) = 0$ because once on $s(x; t) = 0$ the tracking error, $\tilde{x}$, will tend toward zero according to Eq. (2.4). $V(x)$ is related to the distance from $s(x; t) = 0$. It is obvious that $V(x)$ is positive definite due to the square term.\(^1\) Since $V(x)$ is positive definite, whether or not the function is stable from a Lyapunov standpoint can be determined from Eq. (2.7)

$$\frac{1}{2} \frac{d}{dt} s^2 \leq 0 \quad (2.7)$$

Satisfying Eq. (2.7) implies that regardless of the magnitude of $s(x; t)$ it will not increase. However, this criteria is insufficient from a tracking standpoint. The magnitude of $s(x; t)$ is related to the magnitude of the function forcing the dynamic error equation (Eq. (2.4)). The left hand side of Eq. (2.7) equaling zero implies the magnitude of $s$ is neither increasing nor decreasing. In order for $\tilde{x}$ to be allowed to follow the dynamics associated with $s(x; t) = 0$ and to converge to zero the following condition is required [16].

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \quad (2.8)$$

Where $\eta$ is a small positive constant. Satisfying Eq. (2.8) means the forcing function will approach zero allowing $\tilde{x}$ to converge to zero. Notice, however, that $s(x; t)$ may be set equal to zero at $t = 0$ with proper selection of initial conditions. The simulations analyzed in this work made use of initial conditions that force $s(x; t)$ equal to zero at $t = 0$ and because of Eq. (2.7), $s(x; t)$ will never stray from zero.

The implications of Eq. (2.6) and Eq. (2.8) can be visualized in Figure 2.3. The error trajectories will always point toward $s(x; t) = 0$, and once at $s(x; t) = 0$, will tend toward the origin. What this means for sliding mode control is, once on the sliding surface, the overall state error will not increase. Furthermore, if there is no initial error (due to $x_d(0) = x(0)$) none is developed. Satisfying Eq. (2.6) and Eq. (2.8) is achieved through proper controller development which is discussed in Section 2.2.3.

\(^1\) $V(x)$ is actually positive semidefinite because $s$ can be zero but that case is not considered because $s(x; t) = 0$ is the goal. The analysis is done assuming $s(x; t) \neq 0$ to ensure that once $s(x; t)$ does equal zero it will remain at zero.
2.2.3 Controller Development

The objective of controller design is to develop a sliding controller so the closed loop system dynamics reduce to the sliding surface definition. The steps for designing a sliding controller are

1. Define a sliding surface

2. Set the sliding surface equal to zero and differentiate. Setting the surface equal to zero enforces the sliding condition; or, states the error trajectories will behave according to a stable, homogeneous differential equation. Differentiating the surface makes its order match that of the dynamic system so \( x^{(n)} \) may be eliminated.

3. Substitute the system dynamics into the equation of the differentiated sliding surface so as to eliminate \( x^{(n)} \)

4. Solve for the control input, \( u \)

By defining a “sliding surface” and developing a control law utilizing the sliding surface the dynamics of a general state-space model can be transformed to

\[
\dot{x} + \lambda \ddot{x} = 0
\]  

(2.9)
or a stable differential equation where the dynamic variable is tracking error, $\tilde{x}$. If $\lambda$ is a positive constant then Eq. (2.9) inherently defines a stable, homogeneous ordinary differential equation. Because of its form the tracking error will asymptotically approach zero. Development of a control law satisfying Eq. (2.9) requires dynamic inversion of the input influence matrix.

The term sliding surface is meant to be more of a physical interpretation of the state trajectory on the phase plane. The sliding surface uses the tracking error vector as its dynamic variable and results in a trajectory that leads to the origin, or zero error. If the error is “placed” on the sliding surface, it will “slide” toward the origin, or toward zero error. Whether or not the error trajectories are on the sliding surface depend on the initial conditions and uncertainties contained in the system.

### 2.3 The Linear Quadratic Regulator

#### Preface

The Linear Quadratic Regulator (LQR) is a well documented and classic control problem. The LQR problem is concerned with regulating a linear plant model subject to a quadratic cost function [17]. Generally, the form of the solution depends on the boundary conditions.

#### 2.3.1 Final State Free Boundary Condition

When seeking to regulate a state-space model there are two popular sets of boundary conditions. The initial conditions are defined by $x(0)$. The final state, $x(t_f)$, can either be fixed or free. The final state being fixed implies at $t_f$ the states will be exactly at a specified value. The optimal control problem develops a control law bringing the states to these specified values while minimizing the scalar cost function

$$J = \frac{1}{2} \int_{t_0}^{t_f} u^T R u \, dt$$

The final state being free implies the value of the states at $t_f$ is not predetermined. However, the final values are incorporated into the cost function. As before, the optimal control problem seeks the most efficient way to control the system. Consider the linear time-varying plant

$$\dot{x} = A(t)x + B(t)u$$
where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ with the associated quadratic cost function

$$J = \frac{1}{2} x^T(t_f)S(t_f)x(t_f) + \frac{1}{2} \int_{t_o}^{t_f} (x^TQx + u^TRu)dt$$

(2.10)

where $S(t_f) \geq 0$ is the final state weighting matrix, $Q \geq 0$ is the state weighting matrix and $R > 0$ is the input weighting matrix. These matrices are generally diagonal so each diagonal element is a weighting factor. Off diagonal elements imply cross-coupling among weights. Cross-coupled weights is a fairly nonintuitive concept and are atypical in practice.

The method for determining a control effort, $u$, minimizing the cost function $J$ is summarized as follows [17]:

1. Define the Hamiltonian, $H$

$$H(x, u, t) = (x^TQx + u^TRu) + \lambda^T(A(t)x + B(t)u - \dot{x})$$

where, in this section, $\lambda$, represents the *Lagrange multiplier*

2. Differentiate the Hamiltonian with respect to all variables, $\lambda$, $x$ and $u$ (resulting in the state equation, costate equation and stationary condition)

3. Enforce proper boundary conditions

4. Solve the *Riccati equation*

the result of this procedure is

$$u(t) = -R^{-1}B^T S(t)x(t)$$

where $S(t)$ is the solution to the Riccati equation. The *Kalman gain* is defined as

$$K(t) = R^{-1}B^T S(t)$$

so

$$u(t) = -K(t)x(t)$$

The optimal control for regulation of a linear state space model is time varying, full state feedback. Implementing this solution may be difficult do to time varying $K$ values. Section 2.3.2 presents an alternate solution.
2.3.2 The Steady-State Solution

As mentioned in section 2.3.1 the optimal control law for the LQR may be difficult to implement since $K(t)$ is time varying. It is possible, in some instances, to replace the time varying gain matrix $K(t)$ with a constant matrix $K$ without significant loss controller performance.

Because the Riccati equation is solved backwards in time the final value of the solution is the initial value of $K$. Furthermore, for a stabilizable plant, there is always a positive semidefinite limiting solution to the Riccati equation [16].

Let

$$u = -Kx$$  \hspace{1cm} (2.11)

Where

$$K = K(\infty) = R^{-1}B^TS(\infty)$$

Note that Eq. (2.11) is the optimal control law for the infinite horizon LQR problem whose performance index is

$$J_\infty = \frac{1}{2} \int_{t_0}^{t_f} (x^TQx + u^TRu) dt$$

Thus, as the final time approaches $\infty$ the use of the steady state control becomes more and more acceptable.

2.4 The Transform

The procedure developed in Section 2.2.3 requires an inversion of the input influence matrix. Depending on the system definition the input influence matrix may be singular and non-invertible. By introducing a transform and applying it to the original system it is possible to develop a sliding controller for the transformed system.

Consider the general State-Space model

$$\dot{x} = Ax + Bu$$  \hspace{1cm} (2.12)

where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. Define a sliding surface, $s$, as

$$s = \left( \frac{d}{dt} + \lambda \right)^n \left( \int_0^\tau \ddot{x} \, dr \right)$$  \hspace{1cm} (2.13)

Creating a surface for each equation in the state space system results in the system of
equations
\[ s = x - x_d + \lambda \int_0^t (x - x_d) \, dr \] (2.14)

Once the state trajectories are on the sliding surface no movement is ensured off the surface by taking the time derivative of \( s \) and setting it equal to zero. Applying Leibniz’s rule
\[ \dot{s} = \dot{x} - \dot{x}_d + \lambda \ddot{x} = 0 \] (2.15)

substituting Eq. (2.12) into Eq. (2.15) and re-arranging:
\[ u = B^{-1}[\dot{x}_d - Ax - \lambda \ddot{x}] \] (2.16)

If Eq. (2.16) is substituted into Eq. (2.12) the result is Eq. (2.9). As can be seen in Eq. (2.16) the control law developed via sliding mode control requires the inversion of the \( B \) matrix. If \( B \) is non-square and the pseudoinverse is used, certain dynamics may be lost and perfect tracking of all states may not be possible.

If \( B \) is invertible favorable control of all states is possible. If \( B \) is not invertible, however, then all the states may not be controlled simultaneously and the concept of selectable states arises. The term selectable states refers to selecting which states will have the most desirable behavior. The idea is to choose which states will be controlled most effectively at the expense of satisfactory control of remaining states. For instance, if \( B \) is not invertible and all the states cannot be controlled, is it possible to select some subset of states \( x_q \in x_n \) to be controlled more aggressively than the subset of remaining states \( x_{n-q} \in x_n \)?

We have noted the difficulties associated with controlling all states when \( B \) is not invertible. Instead, some combination of the states will be controlled. This is essentially a process of defining fictitious outputs where the mapping function is of “suitable” dimension as discussed in Section 6.4 of Friedland [1]. Define a mapping function, i.e.
\[ y = Tx \] (2.17)

where \( T \) is a constant, fully populated \( m \times n \) matrix. Differentiating Eq. (2.17) and substituting into Eq. (2.12) results in
\[ \dot{y} = TAx + TBu \] (2.18)
Define a new sliding surface, \( s \), as
\[
\begin{align*}
    s &= \left( \frac{d}{dt} + \lambda \right)^n \left( \int_0^t \ddot{y} \, dr \right) \\
    \text{(2.19)}
\end{align*}
\]

Creating a surface for each equation in the state space system results in the system of equations
\[
\begin{align*}
    s &= y - y_d + \lambda \int_0^t (y - y_d) \, dr \\
    \text{(2.20)}
\end{align*}
\]

As before, once the state trajectories are on the sliding surface we ensure no movement off the surface by taking the time derivative of \( s \) and setting it equal to zero. Applying Leibniz’s rule
\[
\dot{s} = \dot{y} - \dot{y}_d + \lambda \ddot{y} = 0 \\
\text{(2.21)}
\]

where
\[
\ddot{y} = y - y_d \\
\text{(2.22)}
\]

substituting Eq. (2.18) into Eq. (2.21) and re-arranging:
\[
\begin{align*}
    u &= (TB)^{-1} \left[ \dot{y}_d - TA \dot{x} - \lambda \ddot{y} \right] \\
    \text{(2.23)}
\end{align*}
\]

or
\[
\begin{align*}
    u &= (TB)^{-1} T \left[ \dot{x}_d - A \dot{x} - \lambda \ddot{x} \right] \\
    \text{(2.24)}
\end{align*}
\]

Note the term requiring inversion is no longer \( B \) but \( TB \). If \( T \) is chosen so that \( TB \) is non-singular an inversion of the resulting matrix is possible. If Eq. (2.23) is substituted into the system in Eq. (2.18) the closed loop system dynamics are similar to those in Eq. (2.9), but for the dynamic variable \( y \)
\[
\dot{\ddot{y}} + \lambda \ddot{y} = 0 \\
\text{(2.25)}
\]

This form allows for proper inversion but poses the investigator with a new problem; what is the form of \( T \)?

### 2.5 The Pseudoinverse

Preface

If a matrix \( B \) is non-square or singular, the true inverse (denoted as \( B^{-1} \)) does not exist. However, one may define a pseudoinverse (or generalized inverse). The pseudoinverse
may or may not contain properties associated with the true inverse. The pseudoinverse may be used in place of the true inverse at the expense of possibly unfavorable results.

### 2.5.1 Relation to the True Inverse

A pseudoinverse can be used to invert a singular matrix however, the results may be unfavorable. A non-singular matrix $B$ has a unique inverse, denoted by $B^{-1}$, such that

$$BB^{-1} = B^{-1}B = I$$  \hspace{1cm} (2.26)

Where $I$ is the identity matrix [18]. There are numerous properties of the inverse

$$(B^{-1})^{-1} = B$$

$$(B^T)^{-1} = (B^{-1})^T$$

$$(B^*)^{-1} = (B^{-1})^*$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

where $B^T$ and $B^*$ denote the transpose and conjugate transpose, respectively, of $B$. A matrix, $B^\diamond$, may be considered a pseudoinverse of a given matrix $B$ if it

1. Exists for a class of matrices larger than the class of nonsingular matrices.
2. Has some of the properties of the true inverse
3. Reduces to the true inverse when $B$ is non-singular

Because the definition of $B^\diamond$ is not extremely specific, a particular definition for a pseudoinverse may not be unique. And while $B^\diamond$ may be designed to have certain properties of the true inverse, Eq. (2.26) is usually the most sought after.

### 2.5.2 Square, Overactuated and Underactuated Systems

A state-space system of the form $\dot{x} = Ax + Bu$ has three distinct scenarios regarding the “shape” of the system. A matrix system is said to be square, overactuated or underactuated based on the dimensions of the system’s $B$ matrix. The $A$ matrix must be an $n \times n$ matrix due to the nature of the state-space formulation and therefore does not factor into the classification of the system. The $A$ matrix may be singular, but it will always be square.
Because the control architecture being investigated in this thesis requires full state feedback it is assumed the system outputs are the state variables. Therefore, the number of outputs equals the number of states (i.e. $p = n$) or $C = I_{n \times n}$.

**Definition 2.8.** A state-space model of the form $\dot{x} = Ax + Bu$ is said to be *square* if the number of inputs equals the number of outputs (i.e. $m = p$) and the columns of the $B$ matrix are linearly independent (i.e. $B$ is invertible).

The term “square,” when referring to the shape of a system, indicates there are as many inputs, $m$, as there are outputs, $p$, or in this case states. In this thesis, the term “square” also implies the $B$ matrix is invertible. The columns of $B$ being linearly independent implies each input has independent, unique influence on the system. Intuitively this means none of the controllers need to be “shared.”

**Definition 2.9.** A state-space model of the form $\dot{x} = Ax + Bu$ is said to be *overactuated* if the number of inputs is greater than the number of outputs (i.e. $m > p$) and the rank of $B$ is equal to $n$ (i.e. $R(B) = n$)

The condition of $m > p$ amounts to the $B$ matrix being “fat” or “wide.” The condition $R(B) = n$ implies after the matrix has been put in row echelon form it will be a square matrix. The $n$ independent columns of the $B$ matrix represent $n$ independent control inputs. All the columns eliminated via elementary row operations represent redundant or superfluous inputs. Because $m - n$ inputs are linearly dependent and may be removed without loss of control capability the system is considered overactuated.

It is common to encounter overactuated or redundantly actuated systems in practice. Any system that is responsible for human safety is typically overactuated. If a redundant control input is damaged or lost the system can still function properly. An example of such a system is the directional control of an aircraft. There are typically redundant control surfaces so that if one is damaged the aircraft may still be controlled.

**Definition 2.10.** A state-space model of the form $\dot{x} = Ax + Bu$ is said to be *underactuated* if the number of inputs is less than the number of outputs (i.e. $m < p$).

The condition of $m < p$ amounts to the $B$ matrix being “tall.” There is no condition of the rank of $B$ because if the columns are linearly dependent, the system will be underactuated so long as $m < p$. Underactuated systems represent a large portion of dynamical systems. Any single-input single-output (SISO) system with order $> 1$ and most single-input multiple-output (SIMO) systems are underactuated.

The shape of the system offers information as to whether or not $B^{-1}$ exists and to what type if properties its pseudoinverse may have. The shape of the system is an important system property and must be taken into consideration when designing a control system.
2.5.3 The Moore-Penrose Pseudoinverse

As noted in Section 2.5.1 the pseudoinverse of a given matrix is not unique and may be formed to posses certain properties. Consider, the linear system $Ax = b$. One would find the solution for $x$ by forming $x = A^{-1}b$ assuming $A^{-1}$ exists. In practice, however, one may encounter the situation where $A$ is non-square or singular. This should immediately inform the investigator the problem is poorly posed. Even so, it may be desired to form a solution $x$ that minimizes the residual, $r$, defined by $\|b - Ax\|^2$. The Moore-Penrose Pseudoinverse of $A$, denoted $A^\dagger$, accomplishes this task. The form of the Moore-Penrose pseudoinverse is as follows [19]

$$
B^\dagger = \begin{cases}
B^T(BB^T)^{-1} & \text{if } B \text{ has full row rank} \\
(BB^T)^{-1}B^T & \text{if } B \text{ has full column rank}
\end{cases}
$$

(2.27)

The following theorem and proof is from Cline [19].

**Theorem 2.11.** For any system of equations $Ax = b$ where $A$ has full column rank, $x = A^\dagger b$ is the unique vector with $\|b - Ax\|^2$ minimal.

**Proof.** If $A$ is square or if $m > n$ and $Ax = b$ is consistent, then with $A^\dagger = (A^TA)^{-1}A^T$ a left inverse of $A$ and $AA^\dagger b = b$, the vector $x = A^\dagger b$ is the unique solution with $\|b - Ax\|^2 = 0$. On the other hand, if $m > n$ and $Ax = b$ is inconsistent,

$$
\|b - Ax\|^2 = \|(I - AA^\dagger)b - A(x - A^\dagger b)\|^2 \\
= \|b - AA^\dagger b\|^2 + \|A(x - A^\dagger b)\|^2
$$

since $A^T(I - AA^\dagger) = 0$. Hence $\|b - Ax\|^2 \geq \|b - AA^\dagger b\|^2$ where equality holds if and only if $\|A(x - A^\dagger b)\|^2 = 0$. But $A$ with full column rank implies $\|Ay\|^2 \geq 0$ for any vector $y \neq 0$, in particular for $y = x - A^\dagger b$.

The Moore-Penrose pseudoinverse solves the least squares problem or minimizes the Euclidean norm of the vector $b - Ax$. If the problem is properly posed and there is a unique solution (i.e. a square system) then $\|b - Ax\|^2 = 0$. Furthermore, if the problem is improperly posed and there is more than one solution (i.e. an overactuated system) then $\|b - Ax\|^2 = 0$. However, when the problem is improperly posed and there is no solution (i.e. underactuated system) the use of the Moore-Penrose pseudoinverse produces a solution that is “as close to” the desirable solution, $\|b - Ax\|^2 = 0$, as possible.
2.5.4 Relationship Between the Transform and the Pseudoinverse

Recall the similarities between the original sliding mode control law in Eq. (2.16)

\[ u = B^{-1}[\dot{x}_d - Ax - \lambda \tilde{x}] \]

and the proposed sliding mode control law for the transformed system in Eq. (2.23)

\[ u = (TB)^{-1}T[\dot{x}_d - Ax - \lambda \tilde{x}] \]

The significant difference between Eq. (2.16) and Eq. (2.23) is \( B^{-1} \) appears to have been replaced by \( (TB)^{-1}T \). This is an interesting result since \( (TB)^{-1}T \) will produce a matrix whose dimensions are opposite that of \( B \). Furthermore, if \( T = B^T \) then \( (TB)^{-1}T \) is the Moore-Penrose inverse.

By defining fictitious outputs, or by applying a squaring transformation matrix to the original system we have proposed a problem that amounts to defining a new pseudoinverse. It is desired that this new pseudoinverse, defined by the matrix \( T \) will have the property that it maintains the sliding mode error equations for the fictitious outputs, as in Eq. (2.25), while minimizing the LQR cost function in terms of state trajectories and control effort.

2.6 Dynamic Extension

Preface

Dynamic extension is a method of redefining the system such that the input influence matrix is invertible [16]. The limitation to this technique is that it can be tedious for large systems. Also, dynamic extension produces a system that is higher than first order or no longer in state-space form. Because the system is no longer in state-space form many of the benefits associated with that form are lost. Dynamic extension is being considered herein because of its ability to transform the system into a purely square system. The results of using dynamic extension will not be compared to the use of the pseudoinverse or to the use of the transformation matrix being proposed in this paper. If dynamic extension is employed then perfect tracking will result. Poor tracking is not the reason dynamic extension is typically rejected. Dynamic extension is typically rejected because of the difficulty applying the technique to large scale systems and because the resulting
system is no longer in state-space form.

2.6.1 Concept

Dynamic Extension is a method for accommodating a non-square input influence matrix. By differentiating the system equations the derivative of the original input is present. By substituting the new set of equations into the original ones it is possible to, effectively, square the system [16]. Consider the following 2-state system where the \( A \) and \( B \) matrices are not functions of time.

\[
\dot{x} = A_{2\times2}x + B_{2\times1}u \tag{2.28}
\]

By performing dynamic extension, it is possible to obtain the system into the following form

\[
\ddot{x} = A'_{2\times2}x + B'_{2\times2}\begin{bmatrix} u \\ \dot{u} \end{bmatrix} \tag{2.29}
\]

where the prime symbols indicate that the numerical values of the matrix may have changed. Note, the system in Eq. (2.29) is now square and an inversion of \( B \) is possible. Also, the square system is no longer in standard state-space form; the second derivative of the state vector is on the left hand side of the equation rather than the first derivative.

This method has a number of shortcomings. While there is a general procedure for performing dynamic extension there is no general form of the solution. The system equations must be manipulated individually for each different system. While this is mainly differentiation and algebra, the implementation is tedious for systems with more than three states.

Results of this method will not be presented. Because the resulting system may be invertible, the sliding controller will be able to provide perfect tracking. The drawback to this method is not its inability to provide satisfactory tracking, it is the difficulty in performing the method. Appendix A shows the necessary work to transform a 4-state system with one input into a square system.

2.6.2 Effects of the Method

Notice dynamic extension is not a control methodology. It is merely a method to square a system. Once a system has has dynamic extension applied to it, the task of developing the control law remains.

In addition to squaring the system dynamic extension alters the basic form of the system. Section 2.6.1 presented a very brief example of a first order (state-space) model
being transformed into a second order (non state-space) model. The effect of this transformation is the addition of \( n \) poles at the origin.

Proof. Recall the general state-space model.

\[
\dot{x} = Ax + Bu \tag{2.30}
\]

Dynamic extension transforms the system to (see Section 2.6.1)

\[
\ddot{x} = A'x + B'u' \tag{2.31}
\]

The matrix \( A' \) contains the system character of the transformed system.

Reducing Eq. (2.31) to state-space form

Let \( Z_1 = x \) and \( Z_2 = \dot{x} \). Therefore

\[
\begin{align*}
\dot{Z}_1 &= \dot{x} = Ax + Bu = AZ_1 + Bu \\
\dot{Z}_2 &= \ddot{x} = A'x + B'u' = A'Z_1 + B'u'
\end{align*}
\]

In matrix form

\[
\begin{bmatrix} 
\dot{Z}_1 \\
\dot{Z}_2 
\end{bmatrix} = 
\begin{bmatrix} 
A & 0 \\
A' & 0 
\end{bmatrix} 
\begin{bmatrix} 
Z_1 \\
Z_2 
\end{bmatrix} + 
\begin{bmatrix} 
B & 0 \\
0 & B' 
\end{bmatrix} 
\begin{bmatrix} 
u \\
\end{bmatrix} \tag{2.32}
\]

or

\[
\dot{Z} = A_z Z + B_z U_z \tag{2.33}
\]

The characteristic equation of Eq. (2.33) will be the same as Eq. (2.31).

\[
|s_r I - A_z| = \left| s_r I \begin{bmatrix} 
A & 0 \\
0 & A' 
\end{bmatrix} - 
\begin{bmatrix} 
B & 0 \\
0 & B' 
\end{bmatrix} \right| = 0 \tag{2.34}
\]

Because of the form of \( A_z \), Eq. (2.34) may be written as

\[
s_r^n s_r I - A| = 0 \tag{2.35}
\]

where \( n \) is the number of states in either of the original systems (Eq. (2.30) or Eq. (2.31)).
Therefore, the roots of the transformed system are the same as the original system with the addition of \( n \) poles at the origin. This adds integrative character to the open loop system. The presence of these newly introduced poles must be taken into account in developing the control system. See Palm [20] for discussions on basic control system design.

2.7 The Solution

Preface

A number of analysis tools have been introduced previously. It is possible to use them together to solve the problem of developing a sliding controller when \( B \) is singular. First a sliding controller will be developed by making use of the Moore-Penrose pseudoinverse. Next, two sliding controllers will be developed for the system transformed by the transformation matrix \( T \). The problem of determining a suitable form for \( T \) will then be addressed. At the end of this section a summary and brief discussion of the proposed solutions will be provided.

2.7.1 Solution One: Use of the Moore-Penrose Pseudoinverse

A solution utilizing the Moore-Penrose pseudoinverse will be formed before a solution utilizing the transformation matrix \( T \) is constructed. Use of the Moore-Penrose pseudoinverse is typically the first technique used when faced with the problem of inverting a singular matrix. Because of this, a sliding controller making use of the Moore-Penrose pseudoinverse will serve as a baseline for comparison.

Recall the sliding controller from Eq. (2.16). Since \( B^{-1} \) does not exist it will be replaced with \( B^\dagger \), resulting in a sliding controller of the form

\[
    u = B^\dagger [\dot{x}_d - Ax - \lambda \tilde{x}]
\]  

(2.36)

The goal of the controllers being developed from this point on is to improve on the tracking performance provided by the controller in Eq. (2.36).

2.7.2 Sliding Controller vs. Suboptimal Feedback

The novel solution produced by the current research lends itself to an observed similarity between a sliding controller (when desired states are zero) and the suboptimal LQR
solution. By letting the sliding surface for the transformed system be defined as it is in Eq. (2.20)

\[ s = y - y_d + \lambda \int_0^t (y - y_d)dr \]

the controller is of the form in Eq. (2.23)

\[ u = (TB)^{-1}[\dot{y}_d - TAx - \lambda \dot{y}] \]

When the desired states, \( y_d \), are set equal to zero and the states, \( y \), are transformed back into the \( x \) domain

\[ u = -(TB)^{-1}T[A + \lambda I]x \]

the solution is similar to the form in Eq. (2.11).

\[ u = -Kx \]

If there were some way to satisfy

\[ K = (TB)^{-1}T[A + \lambda I] \]  

optimal character will have been successfully imparted onto the closed loop, unforced system. Satisfying Eq. (2.38) is the motivation for the following two solutions.

### 2.7.3 Solution Two: Of the Form \((TB)^{-1}T\)

Solving Eq. (2.38) explicitly for \( T \) may not always be possible. Rearranging Eq. (2.38) results in the following equality.

\[ T_{m \times n} (BK - [A + \gamma I])_{n \times n} = 0_{m \times n} \]  

Let the matrix \((BK - [A + \gamma I])_{n \times n} = G_{n \times n}\). Note the dimensions of the resulting matrices. This does not result in a square system. The product \(TG\) will result in \(m \cdot n\) equations that must all equal zero. Carrying out this multiplication, collecting terms and reforming a matrix equality results in

\[ \begin{bmatrix} G^T & 0 & \cdots & 0 \\ 0 & G^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G^T_m \end{bmatrix} \begin{bmatrix} T^T_{1 \times n} \\ T^T_{2 \times n} \\ \vdots \\ T^T_{m \times n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \]  

(2.40)
Where $T_{1 \times n}$ denotes the first row of the original $T$ matrix. Equation (2.40) is a homogeneous equation, meaning that if a solution does exist it is not unique. For a non-unique family of solutions to exist $G$ must be singular. In this case it is possible to select arbitrary values for certain elements of $T$. Because $G$ being singular is a restrictive constraint the solution developed in this section will not be used to produce any results.

2.7.4 Solution Three: Of the Form $(T_*B)^{-1}T$

If $G$ is nonsingular, the only solution to Eq. (2.40) is the trivial zero vector. In the case where $G$ is nonsingular it is necessary to use two different matrices, $T$ and $T_*$, in Eq. (2.38) and reformulate the problem. Introducing the second variable, $T_*$, allows the investigator to expand the range space of the pseudoinverse to include $K$. Consider the following equality.

$$K = (T_*B)^{-1}T[A + \gamma I] \quad (2.41)$$

The form in Eq. (2.41) is easily solved. Rearranging the equation yields

$$T = T_*BK[A + \gamma I]^{-1} \quad (2.42)$$

Notice there is one equation and two unknowns, meaning that there must be an arbitrary variable. The equation has been formed such that $T_*$ is the arbitrary variable. Because the product of $T_*B$ must be inverted, the product must be nonsingular. $T_*$ will be chosen to be $B^T$ which will always satisfy $T_*B$ being invertible.

Proof. Let $B$ be a $n > m$ rectangular matrix whose columns, $c$ & $d$, are linearly independent. This is a reasonable assumption since it implies each input has a unique effect on the system. If the inputs were not unique (i.e. linearly dependent column vectors) then it should be possible to combine the two inputs into one. Then

$$B = \begin{bmatrix} c & d \end{bmatrix}$$

and

$$B^T B = \begin{bmatrix} c^T \\ d^T \end{bmatrix} \begin{bmatrix} c & d \end{bmatrix} = \begin{bmatrix} \langle c|c \rangle & \langle c|d \rangle \\ \langle c|d \rangle & \langle d|d \rangle \end{bmatrix}$$

for the resulting matrix to be nonsingular the columns must be linearly inde-
pendent.

\[
\begin{bmatrix}
\langle c|c \rangle \\
\langle c|d \rangle
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
+ \begin{bmatrix}
\langle c|d \rangle \\
\langle d|d \rangle
\end{bmatrix}
= \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

The previous equation results in the following two equations

\[
a\langle c|c \rangle + b\langle c|d \rangle = 0
\]
\[
a\langle c|d \rangle + b\langle d|d \rangle = 0
\]

rearranging yields

\[
\langle c|ac + bd \rangle = 0
\]
\[
\langle d|ac + bd \rangle = 0
\]

Let the vector \(ac + bd = p\), resulting in

\[
\langle c|p \rangle = 0
\]
\[
\langle d|p \rangle = 0
\]

Note the only way in which both these relations can be true is if \(p\) is either the zero vector or a vector perpendicular to both \(c\) & \(d\). However, \(c\) & \(d\) have been defined as linearly independent vectors and in \(\mathbb{R}^2\) there can be only two linearly independent vectors.

\[
\therefore p = 0
\]

furthermore, \(p\) is a linear combination of two linearly independent vectors \((p = ac + bd)\).

\[
\therefore a = b = 0
\]

and finally

\[
\therefore B^TB \text{ is non singular}
\]

Furthermore \([A + \gamma I]\) must be nonsingular. So long as \(-\gamma\) is not chosen to be an eigenvalue of \(A\), this condition will be satisfied.

Proof. Recall the similarity transform, if an \(n \times n\) matrix \(A\) has a basis of
eigenvectors, then
\[ D = X^{-1}AX \]
is diagonal, with the eigenvalues of \( A \) as the entries on the main diagonal. Here \( X \) is the matrix with these eigenvectors as column vectors [21]. Applying this transform to \([A + \gamma I]\)
\[ X^{-1}[A + \gamma I]X \]
carrying out the multiplication
\[ [D + \gamma X^{-1}X] \]
\[ [D + \gamma I] \]
Here it can be seen that if \(-\gamma\) is equal to one of the diagonal elements of \( D \) (an eigenvalue of \( A \)), \([D + \gamma I]\) will be singular. In the case where \( G \) is nonsingular and \(-\gamma\) does not equal an eigenvalue of \( A \), two matrices, \( T \) and \( T^* \), can be found to satisfy Eq. (2.41).

The final form of the solution is
\[ u = (B^TB)^{-1}T[\dot{x}_d - Ax - \lambda \tilde{x}] \] (2.43)
subject to Eq. (2.42). The solution in Eq. (2.43) will have certain implications in terms of forming a sliding surface as well as evaluating the tracking effectiveness of the controller.

Recall from Section 2.7.3 that “solution two” resulted from applying the transform
\[ y = Tx \]
to the original state space model as well as to the definition of the sliding surface. The resulting controller is of the form
\[ u = (TB)^{-1}T[\dot{x}_d - Ax - \lambda \tilde{x}] \]

One may be inclined to ask the question, “How is solution three of a different form but was derived from the same system and transform?” Working backward from solution three to the transformed system shows that the transformed system would have to be in the form
\[ \dot{y} = TAx + T^*Bu \] (2.44)
The transform $y = Tx$ does not produce this system in Eq. (2.44). For this reason, solution three is not a rigorous solution to the tracking problem.

Solution three causes problems in terms of forming the sliding surface as well. Recall Eq. (2.16), the form of the sliding controller assuming the system is square.

$$u = B^{-1}[\dot{x}_d - Ax - \lambda \ddot{x}]$$

by substituting this sliding controller into the general state-space model

$$\dot{x} = Ax + BB^{-1}[\dot{x}_d - Ax - \lambda \ddot{x}]$$

collecting terms and moving them all to the left hand side of the equals sign

$$\dot{x} - \dot{x}_d + \lambda \ddot{x} - (Ax - Ax) = 0$$

results in the closed loop system dynamics observed in Eq. (2.9)

$$\dot{x} + \lambda \ddot{x} = 0$$

Assuming we are now concerned with control of the transformed states $y$ consider using solution two. By substituting the controller into the transformed system

$$\dot{y} = TAx + TB(TB)^{-1}[\dot{y}_d - TAx - \lambda \ddot{y}]$$

collecting terms and moving them all to the left hand side of the equals sign

$$\dot{y} - \dot{y}_d + \lambda \ddot{y} - (TAx - TAx) = 0$$

results in the closed loop system dynamics very similar to those observed in Eq. (2.9) but for variable $y$

$$\dot{y} + \lambda \ddot{y} = 0$$ (2.45)

The result indicates the sliding surface or sliding dynamics are properly formed and there will be guaranteed favorable tracking for $y$.

Consider using solution three. By substituting the controller into the transformed system

$$\dot{y} = TAx + TB(T_sB)^{-1}[\dot{y}_d - TAx - \lambda \ddot{y}]$$

\[ u = B^\dagger [\dot{x}_d - Ax - \lambda \tilde{x}] \quad \text{Eq. (2.36)} \]

Will be used to form baseline response. Developed solutions will seek to improve on this solution.

\[ u = (TB)^{-1} T[\dot{x}_d - Ax - \lambda \tilde{x}] \quad \text{Eq. (2.24)} \]

Will not be used to generate results because it is rarely applicable.

\[ u = (B^T B)^{-1} T[\dot{x}_d - Ax - \lambda \tilde{x}] \quad \text{Eq. (2.43)} \]

Will be used to generate results and will be compared directly to the solution utilizing the Moore-Penrose inverse. Solution #1.

<table>
<thead>
<tr>
<th>Number</th>
<th>Form</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>[ u = B^\dagger [\dot{x}_d - Ax - \lambda \tilde{x}] ] Eq. (2.36)</td>
<td>Will be used to form baseline response. Developed solutions will seek to improve on this solution.</td>
</tr>
<tr>
<td>#2</td>
<td>[ u = (TB)^{-1} T[\dot{x}_d - Ax - \lambda \tilde{x}] ] Eq. (2.24)</td>
<td>Will not be used to generate results because it is rarely applicable.</td>
</tr>
<tr>
<td>#3</td>
<td>[ u = (B^T B)^{-1} T[\dot{x}_d - Ax - \lambda \tilde{x}] ] Eq. (2.43)</td>
<td>Will be used to generate results and will be compared directly to the solution utilizing the Moore-Penrose inverse. Solution #1.</td>
</tr>
</tbody>
</table>

Table 2.1: Solution Summary

Collecting terms and moving them all to the left hand side of the equals sign

\[ \dot{y} - TB(T_s B)^{-1} \dot{y}_d + TB(T_s B)^{-1} \lambda \ddot{y} - [TAx - TB(T_s B)^{-1} TAx] = 0 \quad (2.46) \]

does not result in closed loop system dynamics similar to those observed in Eq. (2.9). By not being able to properly form the sliding surface it is not possible to guarantee favorable tracking.

2.7.5 Summary

Overall three solutions have been produced. One solution is based on the established Moore-Penrose pseudoinverse and the other two are based on a pseudoinverse related to the solution of the LQR problem. All three solutions are summarized in Table 2.1.

Solution one will be used to form the baseline results. Solution three will be used because of its ability to match exactly the solution of the LQR problem (for the unforced system). Solution two will not be used because of its inability to match the LQR solution.

2.8 Tracking Performance

Preface

The goal of the proposed control system is to develop satisfactory control of states originally having unsatisfactory control characteristics. A theoretical analysis must be employed to evaluate the ability of the proposed control system to favorably alter the closed
loop system dynamics. The closed loop dynamics associated with individual states is investigated to determine their ability to track a reference signal.

2.8.1 Dynamic Analysis

In order to evaluate the tracking characteristics of the states the closed loop dynamics between the state vector, \( x \), and the transformed vector, \( y \) will be evaluated. The original system is of the form

\[
\dot{x} = Ax + Bu
\]

where \( A \) represents the dynamic relationship between \( x \) and \( \dot{x} \). If the Moore-Penrose pseudoinverse solution is applied to the original system the closed loop system dynamics are developed as follows:

\[
\dot{x} = Ax + BB^\dagger[\dot{x}_d - Ax - \lambda \ddot{x}]
\]

\[
\dot{x} = Ax + BB^\dagger \dot{x}_d - BB^\dagger Ax - \lambda BB^\dagger x - \lambda BB^\dagger x_d
\]

\[
\dot{x} = [A - BB^\dagger A - \lambda BB^\dagger] x + [\lambda BB^\dagger BB^\dagger] \begin{bmatrix} x_d \\ \dot{x}_d \end{bmatrix}
\]

where \( A_{cl} \) is the closed loop system dynamic matrix. Applying the transform \( y = Tx \) the dynamic relationship becomes

\[
\dot{y} = TA_{cl} x
\]

The sliding surface is formed for \( y \) and \( y \) contains the dynamics of each state. Those states with faster dynamics (defined by \( TA_{cl} \)) will track more favorably. Or, the primary components of \( y \) will track better than the less substantial components.

The methodology is similar to the idea of model reduction. Consider the expression

\[
\dot{\mathbf{y}} = \begin{bmatrix} \alpha & \beta \\ \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

meaning \( \dot{\mathbf{y}} \) is influenced by two states, \( x_1 \) and \( x_2 \). Assuming \( \beta = 0 \), Eq. (2.47) becomes

\[
\dot{\mathbf{y}} = \alpha x_1
\]

and the solution to this differential equation is

\[
y = x(t = 0) e^{\alpha t}
\]
if $\alpha$ is a large negative number there are few dynamics associated with $x_1$ and in turn $y$, meaning favorable tracking.

Now assuming $\alpha = 0$, Eq. (2.47) becomes

$$\dot{y} = \beta x_2$$

and the solution to this differential equation is

$$y = x(t = 0) e^{\beta t}$$

if $\beta$ is a small negative number there are strong dynamics associated with $x_2$ and in turn $y$, meaning unfavorable tracking. The conclusion is the larger negative the state coefficient, the more favorable the tracking.

The closed loop system dynamic matrix in $y$ coordinates will be examined as part of the results to determine if the control law promotes favorable tracking of the selected state.

### 2.9 Linearization and Model Replacement

#### Preface

In order to use a linearized model in place of its nonlinear counterpart it must be determined if the linear model is valid over the proposed operating region. The nonlinear, longitudinal mode of an aircraft is simulated for various inputs. The linear and nonlinear system responses are observed to determine if the linear model is a valid replacement.

#### 2.9.1 Linearization

Consider the general nonlinear system

$$y = f(z, t) \quad (2.48)$$

As an engineer, one is interested in understanding an manipulating the time history response of such a system. If the the system is truly nonlinear this may be quite a formidable task. If, however, the system is linear there is an abundance of techniques available to control such a system. One may replace a nonlinear model with its linear counterpart [22]. Recall the Taylor series expansion of a function $f(z, t)$ about an
operating point $\bar{z}$

$$f(z, t) = f(\bar{z}, t) + \left( \frac{\partial f}{\partial z} \right)^T \bigg|_{z=\bar{z}} (z - \bar{z}) + \frac{1}{2} (z - \bar{z})^T \left( \frac{\partial^2 f}{\partial z^2} \right)^T \bigg|_{z=\bar{z}} (z - \bar{z}) + h$$

Where $h$ represents higher order terms. The first two terms in the expansion constitute a linear approximation of the nonlinear function $f(z, t)$.

Once a linearized version of the nonlinear system is realized it must be validated. More specifically, it must be determined for what region the linearization is considered valid. Since a nonlinear model is linearized about an operating point the model will be exact for that operating point and will be acceptable for some region around that operating point. If the original model is highly nonlinear the system response may diverge severely from its linear counterpart for only small perturbations of $z$. On the other hand, if the original model does not contain any highly nonlinear terms, the linear and nonlinear system responses may be almost indistinguishable for a relatively large region about the operating point.

Of course, what is considered “acceptable” must be determined by the engineer. Depending on the particular application one may be able to tolerate more model divergence. Being able to make this determination is something that comes only with experience.

The nonlinear flight dynamic equations requiring linearization are presented [23]. The Force Equations are

$$m(\dot{u} - vr + wq) = -mg \sin \theta + F_{Ax} + F_{Tx}$$
$$m(\dot{v} - ur + wp) = mg \sin \phi \cos \theta + F_{Ay} + F_{Ty}$$
$$m(\dot{w} - uq + vp) = mg \cos \phi \cos \theta + F_{Az} + F_{Tz}$$

The Moment Equations are

$$\dot{p}I_{xx} - \dot{q}I_{xy} - \dot{r}I_{xz} = qr(I_{yy} - I_{zz}) + (q^2 - r^2)I_{yz} - prI_{xy} + pqI_{xz} + M_{ez}$$
$$-\dot{p}I_{xy} + \dot{q}I_{yy} - \dot{r}I_{yz} = pr(I_{zz} - I_{xx}) + (r^2 - p^2)I_{xz} - pqI_{yz} + qrI_{xy} + M_{eu}$$
$$-\dot{p}I_{xz} - \dot{q}I_{yz} + \dot{r}I_{zz} = pq(I_{xx} - I_{yy}) + (p^2 - q^2)I_{xy} - qrI_{xz} + prI_{yz} + M_{es}$$
The Kinematic Equations are

\[
\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\
\dot{\theta} = q \cos \phi - r \sin \phi \\
\dot{\psi} = (q \sin \phi + r \cos \phi) \sec \theta
\]

where \( m \) is aircraft mass; \( p, q \) and \( r \) are aircraft body roll rate, pitch rate and yaw rate, respectively; \( F_{Ax}, F_{Ay} \) and \( F_{Az} \) are aerodynamic forces along the aircraft body-axis; \( F_{Tx}, F_{Ty} \) and \( T_{A_z} \) are thrust forces along the aircraft body-axis; \( M_{ex}, M_{ey} \) and \( M_{ez} \) are external applied moments about the aircraft body-axis (mostly aerodynamic but also may include thrust effects); and \( \phi, \theta \) and \( \psi \) are the Euler angles, i.e., bank angle, pitch angle and heading angle, respectively. The Force Equations in the stability axis

\[
\dot{\alpha} = q - (p \cos \alpha + r \sin \alpha) \tan \beta - \frac{LOM}{\sqrt{\alpha \cos \beta}} \\
\dot{\beta} = p \sin \alpha - r \cos \alpha + \frac{1}{\sqrt{\alpha}} (YOM \cos \beta + DOM \sin \beta) \\
\dot{V}_{T} = YOM \sin \beta - DOM \cos \beta \\
g[(\cos \theta \cos \phi \sin \alpha - \sin \theta \cos \alpha) \cos \beta + \cos \theta \sin \phi \sin \beta]
\]

where

\[
DOM = \frac{D - T \cos \alpha}{m}, \quad YOM = \frac{Y}{m}, \quad LOM = \frac{L + T \sin \alpha}{m}
\]

The preceding equations are used to simulate the fully nonlinear aircraft response. These same equations are numerically linearized to generate the linear response.

### 2.9.2 Linear Model Validation and Replacement

The control architecture being developed as part of this thesis will be applied to a linearized, longitudinal model of a high performance aircraft. The state-space form of the longitudinal aircraft model is shown in Eq. (2.49). The state vector \( \mathbf{x} \) is defined as

\[
\mathbf{x} = [V_t \alpha q \theta]^T
\]
Figure 2.4: Longitudinal linear verses nonlinear aircraft response for a horizontal tail deflection of 0.1 degrees. State trajectories are nearly identical.

where $V_t$ is true velocity, $\alpha$ is angle of attack, $q$ is pitch rate and $\theta$ is pitch angle. The single input, $u$, is horizontal tail deflection.

$$\begin{align*}
\dot{x} &= \begin{bmatrix}
-0.0140 & -0.0858 & -0.00329 & -0.561 \\
-0.00881 & -0.853 & 0.995 & 0 \\
0.0162 & -1.095 & -0.809 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} x + \begin{bmatrix}
0.0692 \\
-0.129 \\
-9.382 \\
0
\end{bmatrix} u \\
y &= \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} x + \begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix} u
\end{align*}$$

In order to determine whether or not it is acceptable to replace the nonlinear longitudinal model with the linearized longitudinal model, time history response curves are examined. Various horizontal tail deflections are used as inputs and nonlinear verses linear responses are evaluated. Trim condition for the aircraft is straight and level flight at an altitude of 15000 ft. and a velocity of 627 ft/sec.

Figure 2.4 compares the nonlinear verses linear time history responses of the longitudinal states for a horizontal tail deflection of 0.1 degrees. Clearly, the linearized model...
Figure 2.5: Longitudinal linear verses nonlinear aircraft response for a horizontal tail deflection of 0.2 degrees. State trajectories are nearly identical.

does a satisfactory job of approximating the nonlinear response.

Figure 2.5 compares the nonlinear verses linear time history responses of the longitudinal states for a horizontal tail deflection of 0.2 degrees. Again, the linearized model does a satisfactory job of approximating the nonlinear response.

Figure 2.6 compares the nonlinear verses linear time history responses of the longitudinal states for a horizontal tail deflection of 0.3 degrees. For a deflection of this magnitude divergence becomes evident, most notably of the true velocity and pitch angle. However, this amount of divergence is of no major concern.

Figure 2.7 compares the nonlinear verses linear time history responses of the longitudinal states for a horizontal tail deflection of 0.4 degrees. For a deflection of this magnitude divergence is evident in all states.

Figure 2.8 compares the nonlinear verses linear time history responses of the longitudinal states for a horizontal tail deflection of 0.5 degrees. Again, a deflection of this magnitude results in divergence of all state trajectories.

As is evident from the preceding analysis the linear model is an acceptable replacement for the nonlinear model for \( u \leq 0.4 \) degrees.
Figure 2.6: Longitudinal linear verses nonlinear aircraft response for a horizontal tail deflection of 0.3 degrees. Notice mild divergence.

Figure 2.7: Longitudinal linear verses nonlinear aircraft response for a horizontal tail deflection of 0.4 degrees. Notice divergence.
Figure 2.8: Longitudinal linear verses nonlinear aircraft response for a horizontal tail deflection of 0.5 degrees. Notice divergence.
Chapter 3

Results

3.1 Two Mass, Two Spring, Two Damper System

Preface

The first system under investigation is the classical two mass, two spring, two damper model. The system will first be controlled by the control law in Eq. (2.36) and then by the control law in Eq. (2.43) for various $Q$ & $R$ matrices. The closed loop system will be analyzed to validate the effectiveness of the controller.

3.1.1 Use of Moore-Penrose Pseudoinverse for Tracking

Figure 3.1 shows the two mass, two spring, two damper system schematic. The state-space system model is shown in Eq. (3.1) with system parameters defined in Eq. (3.2). The state vector $\mathbf{x}$ is defined as

$$\mathbf{x} = [x_1 \  \dot{x}_1 \ x_2 \ \dot{x}_2]^T$$

where $x_1$ is the position of mass one, $\dot{x}_1$ is the velocity of mass one, $x_2$ is the position of mass two and $\dot{x}_2$ is the velocity of mass two.
Figure 3.1: Two-Mass System Schematic

\[
\begin{align*}
\dot{x} &= \begin{bmatrix}
0 & 1 & 0 & 0 \\
\frac{k_1 + k_2}{m_1} & \frac{c_1 + c_2}{m_1} & 0 & 0 \\
0 & 0 & 0 & 1 \\
\frac{k_2}{m_2} & \frac{c_2}{m_2} & -\frac{k_2}{m_2} & -\frac{c_2}{m_2}
\end{bmatrix} x + \begin{bmatrix}
0 \\
0 \\
0 \\
\frac{1}{m_2}
\end{bmatrix} u \\
y &= \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} x + \begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix} u
\end{align*}
\]

(3.1)

\[
m_1 = 10, \quad c_1 = 5, \quad k_1 = 3 \\
m_2 = 20, \quad c_2 = 8, \quad k_2 = 7
\]

(3.2)

The desired tracking and \(\lambda\) values are defined by Eq. (3.3)

\[
x_{1d} = \sin(t); \quad \dot{x}_{1d} = \cos(t) \\
x_{2d} = \sin(2t); \quad \dot{x}_{2d} = 2\cos(2t) \\
\lambda = 20
\]

(3.3)

Figure 3.2 first shows the tracking result when the Moore-Penrose pseudoinverse is used to form the control law.
Figure 3.2: Tracking response when the Moore-Penrose pseudoinverse is used to form the control law

### 3.1.2 Dominant Weighting of State One

Figure 3.3 displays the state trajectories verses the desired state trajectories with $Q$ & $R$ weighting shown in Eq. (3.4).

$$
Q = \begin{bmatrix}
1000000000 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} ; \quad R = 10
$$

(3.4)

Choosing $T_\star = B^T$ and given Eq. (3.4), $T$ is determined from Eq. (2.42)

$$
T = \begin{bmatrix}
1.220 & 0.236 & 0.0206 & 0.0247
\end{bmatrix}
$$

It is important to take into consideration control effort expenditures. Unreasonable control effort makes the control system impractical. Figure 3.4 displays the control effort expenditures when the control law is developed based on the weighting factors, $Q$ & $R$, shown in Eq. (3.4). The control effort is not unreasonably large. In fact the average control effort is less than half of what is required for the control law utilizing the
Proposed Pseudoinverse Control Law Effectiveness

Figure 3.3: Tracking response of the proposed control law when state one is heavily weighted.

Analyzing the closed loop system will be done per the discussion in Section 2.8. The closed loop system dynamic matrix is

\[
A_{cl} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-1 & -1.3 & 0.7 & 0.8 \\
0 & 0 & 0 & 1 \\
-483 & -113 & -11.7 & -14.3 \\
\end{bmatrix}
\]

Applying the transform results in

\[
\dot{y} = TA_{cl}x = [-12.2 \quad -1.86 \quad -0.124 \quad -0.143]x
\]

Table 3.1 shows the dynamic coefficient associated with State 1 is the largest negative number. This agrees with the results observed Figure 3.3.

It is also good practice to analyze the closed loop eigenvalues. If any of the eigenvalues have a positive real part the system will be unstable. Table 3.1 also shows the closed loop eigenvalues. All real parts are negative, insuring system stability.
Figure 3.4: Control effort expenditures of the proposed control law when state one is heavily weighted.

<table>
<thead>
<tr>
<th>State</th>
<th>Dynamic Coefficients</th>
<th>Closed Loop Eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-12.2</td>
<td>-3.67 - 6.4i</td>
</tr>
<tr>
<td>2</td>
<td>-1.86</td>
<td>-3.67 + 6.4i</td>
</tr>
<tr>
<td>3</td>
<td>-0.124</td>
<td>-7.35</td>
</tr>
<tr>
<td>4</td>
<td>-0.143</td>
<td>-0.875</td>
</tr>
</tbody>
</table>

Table 3.1: Dynamic coefficients of each state when state one is weighted most heavily and closed loop eigenvalues
3.1.3 Dominant Weighting of State Three

Figure 3.5 displays the state trajectories verses the desired state trajectories with $Q$ & $R$ weighting shown in Eq. (3.5).

$$Q = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 100000000 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}; \quad R = 10 \quad (3.5)$$

Choosing $T_*=B^T$ and given Eq. (3.5), $T$ is determined from Eq. (2.42)

$$T = [0.000630 \quad 0.000697 \quad 1.249 \quad 0.0159]$$

Figure 3.5: Tracking response of the proposed control law when state three is heavily weighted.

It is important to take into consideration control effort expenditures. Unreasonable control effort makes the control system impractical. Figure 3.6 displays the control effort expenditures when the control law is developed based on the weighting factors, $Q$ & $R$, shown in Eq. (3.5). This control law mimics the control law utilizing the pseudoinverse.
Figure 3.6: Control effort expenditures of the proposed control law when state three is heavily weighted.

Because the proposed control law is attempting to track states the pseudoinverse control law tracks well it makes sense the control efforts are identical.

Analysis of the closed loop system will be done per the discussion in Section 2.8. The closed loop system dynamic matrix is

$$A_{cl} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-1 & -1.3 & 0.7 & 0.8 \\
0 & 0 & 0 & 1 \\
0.000657 & -0.000165 & -500 & -31.6
\end{bmatrix}$$

Applying the transform results in

$$\dot{y} = T A_{cl} x = [-0.000686 \quad -0.000278 \quad -7.94 \quad 0.748] x$$

Table 3.2 shows the dynamic coefficient associated with State 3 is the largest negative number. This agrees with the results observed Figure 3.5.

It is also good practice to analyze the closed loop eigenvalues. If any of the eigenvalues have a positive real part the system will be unstable. Table 3.2 also shows the closed
Dynamic Coefficients

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
<th>State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>−0.000686</td>
<td>−0.000278</td>
<td>−7.94</td>
<td>0.748</td>
</tr>
</tbody>
</table>

Closed Loop Eigenvalues

\[-0.65 - 0.76i \quad -0.65 + 0.76i \quad -15.8 - 15.8i \quad -15.8 + 15.8i\]

Table 3.2: Dynamic coefficients of each state when state three is weighted most heavily and closed loop eigenvalues. All real parts are negative, insuring system stability.

### 3.2 Longitudinal Aircraft Model

**Preface**

The second system under investigation is a longitudinal aircraft model. The system will first be controlled by the control law in Eq. (2.36) and then by the control law in Eq. (2.43) for various \(Q\) & \(R\) matrices. The closed loop system will be analyzed to validate the effectiveness of the controller.

#### 3.2.1 Use of Moore-Penrose Pseudoinverse for Tracking

The similar task was performed on a longitudinal aircraft model as was performed in Section 3.1. The state-space form of the longitudinal aircraft model is shown in Eq. (3.6). The state vector \(\mathbf{z}\) is defined as

\[
\mathbf{x} = [V_t \, \alpha \, q \, \theta]^T
\]

where \(V_t\) is true velocity, \(\alpha\) is angle of attack, \(q\) is pitch rate and \(\theta\) is pitch angle.

\[
\dot{\mathbf{x}} = 
\begin{bmatrix}
-0.0140 & -0.0858 & -0.00329 & -0.561 \\
-0.00881 & -0.853 & 0.995 & 0 \\
0.0162 & -1.095 & -0.809 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\mathbf{x} + 
\begin{bmatrix}
0.0692 \\
-0.129 \\
-9.382 \\
0
\end{bmatrix}
\mathbf{u}
\]

\[
y = 
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\mathbf{x} + 
\begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix}
\mathbf{u}
\]

\[\text{(3.6)}\]
Figure 3.7 first shows the tracking result when the Moore-Penrose pseudoinverse is used to form the control law.

![Figure 3.7: Tracking response when the Moore-Penrose pseudoinverse is used to form the control law](image)

### 3.2.2 Dominant Weighting of State One

Figure 3.8 displays the state trajectories verses the desired state trajectories with \( Q \) & \( R \) weighting shown in Eq. (3.7)

\[
Q = \begin{bmatrix}
1000000 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}; \quad R = 10
\]  \hspace{1cm} (3.7)

Choosing \( T_* = B^T \) and given Eq. (3.7), \( T \) is determined from Eq. (2.42)

\[
T = [1391.9 \quad -2.221 \quad -1.637 \quad -13.505]
\]

It is important to take into consideration control effort expenditures. Unreasonable control effort makes the control system impractical. Figure 3.9 displays the control effort
expenditures when the control law is developed based on the weighting factors, $Q$ & $R$, shown in Eq. (3.7). The control effort required by this control law is considerably higher than that required by the pseudoinverse control law, but not unreasonably higher.

Analysis of the closed loop system will be done per the discussion in Section 2.8. The closed loop system dynamic matrix is

$$A_{cl} = \begin{bmatrix} -21.9 & 0.0401 & 0.0373 & 0.265 \\ 41.1 & -1.09 & 0.919 & -1.55 \\ 2960 & -18.2 & -6.32 & -112 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Applying the transform results in

$$\dot{y} = TA_{cl}x = [-35400 \ 87.9 \ 46.8 \ 556]x$$

Table 3.3 shows the dynamic coefficient associated with State 1 is the largest negative number. This agrees with the results observed Figure 3.8.

It is also good practice to analyze the closed loop eigenvalues. If any of the eigenvalues
Figure 3.9: Control effort expenditures of the proposed control law when state one is heavily weighted.

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
<th>State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>−35400</td>
<td>87.9</td>
<td>46.8</td>
<td>556</td>
</tr>
</tbody>
</table>

**Closed Loop Eigenvalues**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>−25.1</td>
<td>−1.74 − 8.63i</td>
<td>−1.74 + 8.63i</td>
<td>−0.718</td>
</tr>
</tbody>
</table>

Table 3.3: Dynamic coefficients of each state when state one is weighted most heavily and closed loop eigenvalues
have a positive real part the system will be unstable. Table 3.3 also shows the closed loop eigenvalues. All real parts are negative, insuring system stability.

### 3.2.3 Dominant Weighting of State Two

Figure 3.10 displays the state trajectories verses the desired state trajectories with $Q$ & $R$ weighting shown in Eq. (3.8)

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 100000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad R = 10$$

(3.8)

Choosing $T_\ast = B^T$ and given Eq. (3.8), $T$ is determined from Eq. (2.42)

$$T = [-0.452 \quad -1429.4 \quad 52.034 \quad -2.519]$$

![Proposed Pseudoinverse Control Law Effectiveness](image)

**Figure 3.10:** Tracking response of the proposed control law when state two is heavily weighted.

It is important to take into consideration control effort expenditures. Unreasonable control effort makes the control system impractical. Figure 3.11 displays the control
Figure 3.11: Control effort expenditures of the proposed control law when state two is heavily weighted.

effort expenditures when the control law is developed based on the weighting factors, \( Q \) & \( R \), shown in Eq. (3.8). The control effort required by this control law is considerably higher than required by the pseudoinverse control law, but not unreasonably higher.

Analysis of the closed loop system will be done per the discussion in Section 2.8. The closed loop system dynamic matrix is

\[
A_{cl} = \begin{bmatrix}
-0.0175 & 21.5 & 0.332 & -0.522 \\
-0.00231 & -41.3 & 0.366 & -0.074 \\
0.486 & -2920 & -46.2 & -5.34 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

Applying the transform results in

\[
\dot{y} = T A_{cl} x = [28.6 \ -93000 \ -2930 \ -172] x
\]

Table 3.4 shows the dynamic coefficient associated with State 2 is the largest negative number. This agrees with the results observed Figure 3.10.

It is also good practice to analyze the closed loop eigenvalues. If any of the eigenvalues
Table 3.4: Dynamic coefficients of each state when state two is weighted most heavily and closed loop eigenvalues

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
<th>State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.6</td>
<td>-93000</td>
<td>-2930</td>
<td>-172</td>
</tr>
</tbody>
</table>

Closed Loop Eigenvalues

\[-43.8 - 32.7i -43.8 + 32.7i -0.00771 - 0.0708i -0.00771 + 0.0708i\]

have a positive real part the system will be unstable. Table 3.4 also shows the closed loop eigenvalues. All real parts are negative, insuring system stability.

3.2.4 Dominant Weighting of State Four

Figure 3.12 displays the state trajectories verses the desired state trajectories with \( Q \) & \( R \) weighting shown in Eq. (3.9)

\[
Q = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 100000
\end{bmatrix}; \quad R = 10
\]  

Choosing \( T_* = B^T \) and given Eq. (3.9), \( T \) is determined from Eq. (2.42)

\[
T = [-0.00472 \ 2.541 \ 35.118 \ -1392.3]
\]

It is important to take into consideration control effort expenditures. Unreasonable control effort makes the control system impractical. Figure 3.13 displays the control effort expenditures when the control law is developed based on the weighting factors, \( Q \) & \( R \), shown in Eq. (3.9). This control law mimics the control law utilizing the pseudoinverse. Because the proposed control law is attempting to track states the pseudoinverse control law tracks well it makes sense the control efforts are identical.

Analysis of the closed loop system will be done per the discussion in Section 2.8. The
Figure 3.12: Tracking response of the proposed control law when state four is heavily weighted.

Figure 3.13: Control effort expenditures of the proposed control law when state four is heavily weighted.
### Dynamic Coefficients

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
<th>State 3</th>
<th>State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.25</td>
<td>-2.46</td>
<td>-4100</td>
<td>-104000</td>
</tr>
</tbody>
</table>

### Closed Loop Eigenvalues

-38.5 - 38.5i, -38.5 + 38.5i, -0.0129, -0.839

Table 3.5: Dynamic coefficients of each state when state four is weighted most heavily and closed loop eigenvalues

Closed loop system dynamic matrix is

\[
A_{cl} = \begin{bmatrix}
-0.0144 & -0.0939 & 0.559 & 21.3 \\
-0.00814 & -0.838 & -0.0616 & -41.1 \\
0.0647 & -0.00932 & -77.1 & -2970 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

Applying the transform results in

\[
\dot{y} = T A_{cl} x = [2.25 \ -2.46 \ -4100 \ -104000] x
\]

Table 3.4 shows the dynamic coefficient associated with State 2 is the largest negative number. This agrees with the results observed Figure 3.10.

It is also good practice to analyze the closed loop eigenvalues. If any of the eigenvalues have a positive real part the system will be unstable. Table 3.5 also shows the closed loop eigenvalues. All real parts are negative, insuring system stability.
Chapter 4

Conclusion, Discussion and Future Work

4.1 Conclusion

A new control methodology was introduced for control of dynamic systems using a model-following approach. There is need for a novel method because certain control methodologies such as dynamic inversion and sliding mode control require an inversion of the input influence matrix. However, if the system’s input influence matrix is singular, inversion is not possible. The utility of the Moore-Penrose pseudoinverse was demonstrated. However, pseudo inversion limits control to states where the controller is directly applied. Therefore, the Moore-Penrose pseudoinverse is restricted by system structure and results in accurate tracking of only certain state variables. When accurate tracking of the remaining state variables is required the pseudo inversion method is not useful. The difficulty of using dynamic extension to square a relatively large system was presented. A new transformation was applied to a dynamical system model resulting in a square and invertible input influence matrix. In addition the transformation allows for the designer to select which state are to be controlled most effectively. A method based on optimal control theory was used to successfully define the proposed transformation matrix. The proposed control methodology was applied to a two mass, two spring, two damper system and to a longitudinal aircraft simulation model. In both cases the proposed control law allowed the designer to select which state was to be controlled most effectively with a fully invertible solution. Simulation results were presented for both example systems proving the validity of the newly proposed control method.
4.2 Discussion

This section provides a discussion and interpretation of the preceding work. Consider the two-mass problem shown in Figure 3.1. The input to the system is directly applied to the second mass. Figure 3.2 displays the linear position and velocity tracking results for the first and second mass using the Moore-Penrose pseudoinverse for the inverse of the input influence matrix. As shown in Figure 3.2 nearly perfect time-history tracking is observed for the second mass position and velocity. The result is not unexpected since the input is being applied directly to the second mass and the control law using the Moore-Penrose pseudoinverse inherently is successful in controlling the second masses two states. The explanation for the result is related to a property of $BB^\dagger$. As mentioned in Section 2.5.1, the normal inverse has the property

$$BB^{-1} = I$$

so that there are no rows of zero in $BB^{-1}$. There appears be a property for every row of zeros in $B$ there is a corresponding row of zeros in $BB^\dagger$. A row of zeros in $BB^\dagger$ indicates that Eq. (2.9) (the result of using a sliding mode controller) cannot be satisfied. However, Eq. (2.9) is a matrix equation where each row is an error function for a corresponding state. If the system equations resulting from the use of the sliding mode controller match the equations in the rows of Eq. (2.9) then the corresponding state will be successfully controlled.

For example, define

$$B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

(4.1)

then

$$BB^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(4.2)

In this case the true inverse does exist and Eq. (2.9) is fully satisfied. However if we define

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

(4.3)

then

$$BB^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

(4.4)

In this case Eq. (2.9) cannot be satisfied exactly for all the system tracking states. How-
ever, note the the first row of Eq. (4.2) and Eq. (4.4) are identical and there is a row of zeros in Eq. (4.4) corresponding to the row on zeros in Eq. (4.3). This implies the first row of Eq. (2.9) would be properly formed and perfect tracking would be achieved for the corresponding state. Also, the populated first row of Eq. (4.3) implies direct control of the state corresponding to the row. Therefore, proper formation of the first row in Eq. (2.9) implies perfect tracking for the states we have direct control over.

Next consider the two mass system under the newly proposed control methodology. Figure 3.3 displays the system tracking response under the new control methodology with $Q$ matrix defined in Eq. (3.4). Nearly perfect tracking is achieved for the position state corresponding to the first mass (and also the velocity state since the two states are kinematically linked). In this case, the LQR problem heavily weights cost associated with the first state and the result translates into a favorable tracking problem. The result is important in proving that the new control methodology can successfully be used to influence tracking of states where the input is not directly applied (assuming the system is controllable). Also, the new control methodology solves the input influence matrix inversion problem. Figure 3.5 displays the system tracking response with $Q$ matrix defined in Eq. (3.5). For this case, we expect excellent tracking for the third and fourth states and indeed is what is shown in Figure 3.5.

A similar type of observation is made for longitudinal control of the aircraft example with one key difference. The aircraft $B$ matrix has three nonzero rows and one input whereas the two mass example has one nonzero row in the $B$ matrix and one input. The number of nonzero rows not equaling the number of inputs implies the pseudoinverse does not correctly form the individual equations in Eq. (2.9). However, reasonable tracking is assured for a number of the states the control system has direct control.

The input to the longitudinal aircraft model is elevator deflection. The control surface is designed to control pitch rate and its integral, pitch angle. Although the elevator has direct control over three states (indicated by the three nonzero rows in the $B$ matrix) the relative magnitude corresponding to the pitch rate term indicates the elevator has more influence in controlling the pitch rate state compared to the velocity and flight path angle states. As shown in Figure 3.7 the control law based on the pseudoinverse provides excellent tracking for states three and four while providing relatively poor tracking for states one and two, as expected.

As shown in the two mass example problem, the proposed new transformation allows for tracking of selectable states using a LQR solution. The new methodology was applied to see if velocity and flight path tracking results can be improved with the $Q$ matrix as shown in Eq. (3.7) and Eq. (3.8). The improved velocity tracking resulting from the
new LQR solutions controller is shown in Figure 3.8. The results show the velocity state tracking has significantly increased relative compared to the results presented in Figure 3.7. A similar result when the LQR solution is setup to provide good tracking for the flight path state (shown in Figure 3.10). The pseudoinverse result (shown in Figure 3.7) can be duplicated by setting $Q$ to Eq. (3.9). The resulting control law produces the results in Figure 3.12 nearly identically.

A discussion of solutions two and three is necessary; first solution two. Recall solution two is of the form

$$\mathbf{u} = (\mathbf{T}\mathbf{B})^{-1}T[\dot{\mathbf{x}}_{\text{d}} - \mathbf{A}\mathbf{x} - \lambda\ddot{\mathbf{x}}]$$

where the term $(\mathbf{T}\mathbf{B})^{-1}T$ is replacing $\mathbf{B}^{\dagger}$ from solution one. A weighted pseudoinverse is of the form

$$\mathbf{B}_w^{\dagger} = (\mathbf{B}^T\mathbf{W}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}$$

and since

$$\mathbf{T} = \mathbf{B}^T\mathbf{W}$$

may be satisfied for a given $\mathbf{T}$ and $\mathbf{B}^T \neq 0$, solution two is a solution utilizing a weighted pseudoinverse rather than the Moore-Penrose pseudoinverse. This is not unlike the techniques use in literature to address the overactuated control allocation problem.

Solution three is again similar to solutions one and two but it makes use of the form $(\mathbf{T_s}\mathbf{B})^{-1}T$ to invert $\mathbf{B}$. This form is thought of as being a third type of pseudoinverse. However, it is not. Recall from Section 2.5.1 that a pseudoinverse shall “Reduces to the true inverse when $\mathbf{B}$ is non-singular.” If $\mathbf{B}$ were square and invertible then $(\mathbf{T_s}\mathbf{B})^{-1}T$ should reduce to the true inverse of $\mathbf{B}$. Notice

$$(\mathbf{T_s}\mathbf{B})^{-1}T = \mathbf{B}^{-1}\mathbf{T_s}^{-1}T \neq \mathbf{B}^{-1}$$

A problem arises because $\mathbf{T_s}^{-1}T$ does not reduce to identity. This result is closely related to the difficulties identified in Eq. (2.46).

Solution two is useful because it properly forms the sliding surface for $\mathbf{y}$ but is incapable of matching the performance of the LQR. Solution three is useful because it can match the performance of the LQR but introduces new difficulties because it is not a true pseudoinverse. These distinctions must be understood when deliberating about which solution to employ.
4.3 Future Work

The research done in this thesis is rather broad. As a result there are many unanswered questions. Answering these questions should be the subject of further research. The first area that should be investigated is Solution Two (Section 2.7.3). Solution Two has the benefit that, when implemented, it will properly form a sliding surface for the transformed variable $y$. Properly forming the sliding surface for $y$ will guarantee stable behavior of $\tilde{y}$. The difficulty encountered with the Solution Two is its inability to satisfy Eq (2.38). Further work should be concerned with determining a value of $T$ such that the closed loop system would minimize the LQR cost function. The problem statement would be as follows. Minimize

$$J = \frac{1}{2}x^T(t_f)S(t_f)x(t_f) + \frac{1}{2}\int_{t_0}^{t_f}(x^TQx + u^TRu)dt$$

subject to

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

$$u = -(TB)^{-1}T[Ax + \lambda x]$$

This problem differs from the classic LQR problem because of the second constraint.

Another area in which the current research could be extended is Multiple-Input Multiple-Output (MIMO) systems. Many of the derivations and equations are presented in matrix notation. This notation implies a MIMO system model. However, the systems that were investigated in Section 3 were Single-Input Multiple-Output (SIMO) systems. Considering the MIMO system would then make this work applicable to the control allocation problem. The form of the $R$ matrix determines how much control effort is expended by each control input. Altering the form of the $R$ matrix will reallocate control effort depending on the decision of the engineer. Further work should be concerned with the implementation of proposed control methodologies on MIMO systems.

Furthermore the systems investigated in this work were sterile in the sense that there was no noise present. In a physical system, state measurements are made with some type of sensor. No sensor is free of noise. This noise would propagate through the system and may have an effect on the controllers performance. It is necessary to determine the effects of noise on the proposed control methodology.

The discussion in Section 2.2.2 makes considerable effort to acknowledge the difference between a typical sliding controller and the controller used in this work. The effects
of satisfying equation Eq. (2.8) may be investigated. Satisfying Eq. (2.8) is typically accomplished by including $k \, \text{sgn}(s)$ in the control law [16]. The additional term is included to accommodate parametric uncertainty and inconsistent initial conditions.

Finally, one must recall the greatest utility of sliding mode control is its applicability to nonlinear systems. The current work concerned itself with only linear systems. Restricting the research to linear systems allowed the use of the LQR problem. Also, the application to linear systems was intended to be a “first pass” attempt at controlling a system with the proposed control methodology. The logic is if the proposed control architecture does not work for linear systems it is unlikely to work for nonlinear systems. It has been shown that the proposed control methodology has merit. The next step is to attempt implementation on more complex systems.
Bibliography


Appendix A

Dynamic Extension Example

Preface

It is discussed in Section 2.6 that performing dynamic extension may be tedious and require intense algebraic manipulation. Dynamic extension was performed on the Two Mass model from Figure 3.1. The following are the resulting equations as obtained from Maple®. This model has only four states. It should now be clear that dynamic extension can be an unreasonably involved process. Equations 1, 2, 3 & 4 are the original four state equations. Equations 37, 38, 39 & 40 are the resulting state equations after dynamic extension has been performed.

\[
\begin{align*}
eq1 &= \dot{Z1} = Z2; \\
\text{eq2} &= \dot{Z2} = \frac{-k1 \cdot Z1 - c1 \cdot Z2 + k2 \cdot Z3 - k2 \cdot Z1 + c2 \cdot Z4 - c2 \cdot Z2}{m1}; \\
\text{eq3} &= \dot{Z3} = Z4; \\
\text{eq4} &= \dot{Z4} = \frac{-k2 \cdot Z3 + k2 \cdot Z1 - c2 \cdot Z4 + c2 \cdot Z2 + fa}{m2}; \\
\text{eq5} &= \ddot{Z1} = Z2; \\
\text{eq6} &= \ddot{Z2} = \frac{-k1 \cdot \dot{Z1} - c1 \cdot \dot{Z2} + k2 \cdot \dot{Z3} - k2 \cdot \dot{Z1} + c2 \cdot \dot{Z4} - c2 \cdot \dot{Z2}}{m1}; \\
\text{eq7} &= \ddot{Z3} = Z4; \\
\text{eq8} &= \ddot{Z4} = \frac{-k2 \cdot \dot{Z3} + k2 \cdot \dot{Z1} - c2 \cdot \dot{Z4} + c2 \cdot \dot{Z2} + \dot{fa}}{m2}; \\
\text{eq9} &= \ddot{Z1} = Z2; \\
\text{eq10} &= \ddot{Z2} = \frac{-k1 \cdot \dot{Z1} - c1 \cdot \dot{Z2} + k2 \cdot \dot{Z3} - k2 \cdot \dot{Z1} + c2 \cdot \dot{Z4} - c2 \cdot \dot{Z2}}{m1}; \\
\text{eq11} &= \ddot{Z3} = Z4; \\
\text{eq12} &= \ddot{Z4} = \frac{-k2 \cdot \dot{Z3} + k2 \cdot \dot{Z1} - c2 \cdot \dot{Z4} + c2 \cdot \dot{Z2} + \dot{fa}}{m2};
\end{align*}
\]
$\text{eq7} := Z_{3\text{dotdot}} = Z_{4\text{dot}}$

$\text{eq8} := Z_{4\text{dotdot}} = \frac{-k_2 Z_{3\text{dot}} + k_2 Z_{1\text{dot}} - c_2 Z_{4\text{dot}} + c_2 Z_{2\text{dot}}}{m_2} + \text{fadot}$

$\text{eq9} := \text{algsubs(eq2,eq5)}$

$\text{eq10} := \text{algsubs(eq1,algsubs(eq2,algsubs(eq3,algsubs(eq4,eq6)))))}$

$\text{eq11} := \text{algsubs(eq4,eq7)}$

$\text{eq12} := \text{algsubs(eq1,algsubs(eq2,algsubs(eq3,algsubs(eq4,eq8))))}$

$\text{eq9} := Z_{1\text{dotdot}} = -\frac{k_1 Z_1 + c_1 Z_2 - k_2 Z_3 + k_2 Z_{1\text{dot}} - c_2 Z_4 + c_2 Z_2}{m_1}$

$\text{eq10} := Z_{2\text{dotdot}} = \frac{(-c_2 k_2 Z_3 m_1 - k_1 m_2 m_1 Z_2 - c_2^2 Z_4 m_1 + k_2 m_2 Z_4 m_1 + c_1 2 m_2 k_2 Z_3 - c_1 m_2 k_2 Z_1 - c_1 m_2 c_2 Z_4
+ 2 c_1 m_2 c_2 Z_2 + c_2 k_2 Z_1 m_1 - k_2 m_2 m_1 Z_2 + c_2 \text{fa} m_1 + c_2^2 Z_2 m_1
+ c_2 m_2 k_1 Z_1 - c_2 m_2 k_2 Z_3 + c_2 m_2 k_2 Z_1 - c_2^2 m_2 Z_4 + c_2^2 m_2 Z_2)}{(m_1^2 m_2)}$

$\text{eq11} := Z_{3\text{dotdot}} = -\frac{k_2 Z_3 + k_2 Z_1 - c_2 Z_4 + c_2 Z_2}{m_2} + \text{fadot}$

$\text{eq12} := Z_{4\text{dotdot}} = \frac{(-c_2 k_2 m_1 - k_2 m_2 Z_4 m_1 - c_2 k_2 Z_1 m_1 + c_2 k_2 Z_3 m_1 + c_2 m_1 m_2 Z_2 - c_2 m_2 k_1 Z_1 - c_2^2 Z_2 m_1 + c_2 m_2 k_2 Z_3 - c_2 \text{fa} m_1 + c_2^2 m_2 Z_4 - c_1 m_2 c_2 Z_2 + c_2^2 Z_4 m_1 - c_2 m_2 k_2 Z_1 - c_2^2 m_2 Z_2)}{(m_2^2 m_1)}$

$\text{eq13} := \text{collect(collect(collect(collect(collect(collect(collect(collect(eq9,Z_1),Z_2),Z_3),Z_4),fadot),fadotdot),fadotdotdot))}$

$\text{eq14} := \text{collect(collect(collect(collect(collect(collect(collect(collect(collect(eq10,Z_1),Z_2),Z_3),Z_4),fadot),fadotdot),fadotdotdot))}$

$\text{eq15} := \text{collect(collect(collect(collect(collect(collect(collect(collect(collect(eq11,Z_1),Z_2),Z_3),Z_4),fadot),fadotdot),fadotdotdot))}$

$\text{eq16} := \text{collect(collect(collect(collect(collect(collect(collect(collect(collect(eq12,Z_1),Z_2),Z_3),Z_4),fadot),fadotdot),fadotdotdot))}$

$\text{eq13} := Z_{1\text{dotdot}} = \frac{k_2 Z_3}{m_1} - \frac{(c_1 + c_2) Z_2}{m_1} - \frac{(k_2 + k_1) Z_1}{m_1} + \frac{c_2 Z_4}{m_1}$

$\text{eq14} := Z_{2\text{dotdot}} = \frac{(k_2 m_2 m_1 - c_2^2 m_2 - c_1 m_2 c_2 - c_2^2 m_1) Z_4}{m_1^2 m_2}
+ \frac{(-c_2 k_2 m_1 - c_1 m_2 c_2 - c_2^2 m_2 k_2) Z_3}{m_1^2 m_2}
+ \frac{(-k_1 m_2 m_1 + c_1^2 m_2 + 2 c_1 m_2 c_2 + c_2^2 m_2 - k_2 m_2 m_1 + c_2^2 m_1) Z_2}{m_1^2 m_2}
+ \frac{(c_2 k_2 m_1 + c_1 m_2 k_2 + c_2 m_2 k_2 + c_1 m_2 k_1 + c_2 m_2 k_1) Z_1}{m_1^2 m_2}
+ \frac{c_2 \text{fa}}{m_1 m_2}$

$\text{eq15} := Z_{3\text{dotdot}} = -\frac{c_2 Z_4}{m_2} - \frac{k_2 Z_3}{m_2} + \frac{c_2 Z_2}{m_2} + \frac{k_2 Z_1}{m_2} + \frac{\text{fa}}{m_2}$
\[ \text{eq16} := Z_{4\dot{\dot{\dot{}}} \dot{\dot{\dot{}}}} = -\frac{c_2 f a}{m^2} + \frac{(-k_2 m_2 m_1 + c_2^2 m_1 + c_2^2 m_2)}{m^2 m_1} Z_4 + \frac{(c_2 k_2 m_1 + c_2^2 m_2 k_2)}{m^2 m_1} Z_3 + \frac{(k_2 m_2 m_1 - c_2^2 m_2 - c_1 m_2 c_2 - c_2^2 m_1)}{m^2 m_1} Z_2 + \frac{(-c_2 k_2 m_1 - c_2^2 m_2 - c_2^2 m_1 m_1)}{m^2 m_1} Z_1 + f_{\dot{a}} \frac{m_2}{m_1} \]

\[ \text{eq17} := Z_{1\dot{\dot{\dot{}}} \dot{\dot{\dot{}}}} = \frac{k_2}{m_1} Z_3 + \frac{(c_1 + c_2)}{m_1} Z_{2\dot{\dot{\dot{}}}} - \frac{(k_2 + k_1)}{m_1} Z_{1\dot{\dot{\dot{}}}} + \frac{c_2}{m_1} Z_{4\dot{\dot{\dot{}}}} \]

\[ \text{eq18} := Z_{2\dot{\dot{\dot{}}} \dot{\dot{\dot{}}}} = \frac{(k_2 m_2 m_1 - c_2^2 m_2 - c_1 m_2 c_2 - c_2^2 m_1)}{m^1 m_2} Z_4 + \frac{(-c_2 k_2 m_1 - c_1^2 m_2)}{m^1 m_2} Z_3 + \frac{c_2}{m^1 m_2} Z_2 + \frac{(-k_1 m_2 m_1 + 2 c_1 m_2 c_2 + c_2^2 m_2 - k_2 m_2 m_1 + c_2^2 m_1)}{m^1 m_2} Z_1 \]

\[ \text{eq19} := Z_{3\dot{\dot{\dot{}}} \dot{\dot{\dot{}}}} = -\frac{c_2}{m_2} Z_{4\dot{\dot{\dot{}}}} - \frac{k_2}{m_2} Z_{3\dot{\dot{\dot{}}}} + \frac{c_2}{m_2} Z_{2\dot{\dot{\dot{}}}} + \frac{k_2}{m_2} Z_{1\dot{\dot{\dot{}}}} + \frac{f_{\dot{a}}}{m_2} \]

\[ \text{eq20} := Z_{4\dot{\dot{\dot{}}} \dot{\dot{\dot{}}}} = -\frac{c_2 f_{\dot{a}}}{m_2} + \frac{(-k_2 m_2 m_1 + c_2^2 m_1 + c_2^2 m_2)}{m^2 m_1} Z_4 + \frac{(c_2 k_2 m_1 + c_2^2 m_2 k_2)}{m^2 m_1} Z_3 + \frac{(-c_2 k_2 m_1 - c_2^2 m_2 - c_1 m_2 c_2 - c_2^2 m_1)}{m^2 m_1} Z_2 + \frac{(-c_2 k_2 m_1 - c_2^2 m_2 - c_2^2 m_1 m_1)}{m^2 m_1} Z_1 + f_{\dot{a}} \frac{m_2}{m_1} \]

\[ \text{eq21} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq17})))) \]

\[ \text{eq22} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq18})))) \]

\[ \text{eq23} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq19})))) \]

\[ \text{eq24} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq20})))) \]

\[ \text{eq21} := Z_{1\dot{\dot{\dot{}}} \dot{\dot{\dot{}}}} = (-c_2 k_2 Z_3 m_1 - k_1 m_2 m_1 Z_2 - c_2^2 Z_4 Z_1 + k_2 m_2 Z_4 m_1 + c_1 m_2 k_1 Z_1 + c_1^2 m_2 Z_2 - c_1 m_2 k_2 Z_3 + c_1 m_2 k_2 Z_1 - c_1 m_2 c_2 Z_4 + 2 c_1 m_2 c_2 Z_2 + c_2 f_{\dot{a}} m_1 + c_2^2 Z_2 m_1 + c_2 m_2 k_1 Z_1 - c_2 m_2 k_2 Z_3 + c_2 m_2 k_2 Z_1 - c_2^2 m_2 Z_4 + c_2^2 m_2 Z_2)/(m^1 m_2) \]
\[ \text{eq22} := Z2\text{dotdotdot} = (-c_2^3 m_1^2 f_a - c_2^2 m_1^2 Z_2 + c_2^3 m_1^2 Z_4 + c_2 f\text{dot} m_1^2 m_2 + c_2^3 m_1^2 k_2 Z_3 - c_2^2 m_1^2 k_2 Z_1 - k_2^2 m_2 m_1 Z_3 - 2 c_2^2 m_2 c_1 Z_2 m_1 + 2 k_2 m_2^2 m_1 k_2 Z_1 + 2 c_2 m_2^2 m_1 c_1 Z_2 - k_2^2 m_2^2 m_1 Z_3 + k_2^2 m_2^2 m_1 Z_1 + k_2^2 m_2 m_1^2 Z_1 + k_2 m_2 m_1^2 f_a - c_2^2 m_2 f_a m_1 - 2 c_2^3 m_2 Z_2 m_1 - m_2^2 c_1^3 Z_2 + m_2^2 c_2^3 Z_4 - m_2^2 c_3^2 Z_2 - 3 c_2 m_2^2 c_1^2 Z_2 + 2 c_2^2 m_2^2 c_1 Z_4 + 2 c_2^2 m_2^2 k_1 m_1 Z_2 - 2 c_2 m_2^2 c_1 k_1 Z_1 + 2 c_2 m_2^2 c_1 k_2 Z_3 - 2 c_2 m_2^2 c_1 k_2 Z_1 + 2 c_2 m_2^2 k_2 m_1 Z_2 + 2 c_2^3 m_2 Z_4 m_1 - 2 k_2 m_2 m_1^2 c_2 Z_4 + 2 k_2 m_2 m_1^2 c_2 Z_2 + 2 c_2 m_2^2 k_2 Z_3 m_1 - 2 c_2^2 m_2^2 k_2 Z_1 m_1 - 2 c_2 m_2^2 k_2 Z_4 m_1 + c_1 m_2 c_2^2 Z_4 m_1 - c_1 m_2 c_2 f a m_1 + c_1 m_2 c_2 k_2 Z_3 m_1 - c_1 m_2 c_2 k_1 Z_1 - k_2 m_2^2 c_1 Z_4 m_1 + 2 m_2^2 k_1 m_1 c_1 Z_2 - m_2^2 k_1 m_1 k_2 Z_3 - m_2^2 k_1 m_1 c_2 Z_4 - m_2^2 c_2^2 Z_1 Z_1 + m_2^2 c_2^2 c_1 Z_2 + 2 c_2^2 c_2^2 k_2 Z_3 - m_2^2 c_2^2 k_1 Z_1 + 2 c_2^2 c_2^2 k_1 Z_2 + 2 c_2^2 c_2^2 Z_2 + m_2^2 c_2^2 k_2 Z_3 - m_2^2 c_2^2 k_1 Z_1 + 2 m_2^2 k_1^2 m_1 Z_1)/(m_1^3 m_2^2) \]
\[ \text{eq23} := Z3\text{dotdotdot} = (f\text{dotdot} m_2 m_1 - k_3 m_2 Z_4 m_1 - c_2 k_2 Z_1 m_1 + c_2 k_2 Z_3 m_1 + k_2 m_2 m_1 Z_2 - c_2 m_2 k_1 Z_1 - c_2 Z_2 Z_1 + c_2 m_2 k_2 Z_3 - c_2 fa m_1 + c_2^2 m_2 Z_4 - c_1 m_2 c_2 Z_2 + c_2^2 Z_4 m_1 - c_2 m_2 k_3 Z_1 - c_2^2 m_2 Z_2)/(m_2^2 m_1) \]
\[ \text{eq24} := Z4\text{dotdotdot} = -(-f\text{dotdotdot} m_2^2 m_1^2 - c_2^2 m_1^2 f_a - c_2^3 m_1^2 Z_2 + c_2^3 m_1^2 Z_4 + c_2 f\text{dotdot} m_1^2 m_2 + c_2^2 m_1^2 k_2 Z_3 - c_2^2 m_1^2 k_2 Z_1 - k_2^2 m_2 m_1^2 Z_3 - c_2^2 m_2 c_1 Z_2 m_1 + k_1 m_2^2 m_1 k_2 Z_1 - c_2^2 m_1 m_2 k_1 Z_1 + k_2 m_2^2 m_1 c_1 Z_2 - k_2^2 m_2^2 m_1 Z_3 + k_2^2 m_2^2 m_1 Z_1 + k_2 m_2 m_1^2 f_a - c_2^2 m_2 f a m_1 - 2 c_2^3 m_2 Z_2 m_1 + m_2^2 c_2^3 Z_4 - m_2^2 c_2^3 Z_2 - c_2 m_2^2 c_1^2 Z_2 + c_2 m_2^2 c_1 Z_4 + c_2 m_2^2 k_1 m_1 Z_2 - c_2 m_2^2 c_1 k_1 Z_1 + c_2 m_2^2 c_1 k_2 Z_3 - c_2 m_2^2 c_1 k_2 Z_1 + 2 c_2 m_2^2 k_2 m_1 Z_2 + 2 c_2^3 m_2 Z_4 m_1 - 2 k_2 m_2 m_1^2 c_2 Z_4 + 2 k_2 m_2 m_1^2 c_2 Z_2 + 2 c_2^2 m_2 k_2 Z_3 m_1 - 2 c_2^2 m_2 k_2 Z_1 m_1 - 2 c_2 m_2^2 k_2 Z_4 m_1 - m_2^2 c_2^2 k_1 Z_1 - 2 m_2^2 c_2^2 c_1 Z_2 + m_2^2 c_2^2 k_2 Z_3 - m_2^2 c_2^2 k_2 Z_1)/(m_2^3 m_1^2) \]
eq25 := \[Z_{1 \text{ddot dot dot}} = \frac{(k_2 m_2 m_1 - c_2^2 m_2 - c_1 m_2 c_2 - c_2^2 m_1) Z_4}{m_1^2 m_2} + \frac{(-c_2 k_2 m_1 - c_1 m_2 k_2 - c_2 m_2 k_2)}{m_1^2 m_2} Z_3 + \frac{(-k_1 m_2 m_1 + c_1^2 m_2 + 2 c_1 m_2 c_2 + c_2^2 m_2 - k_2 m_2 m_1 + c_2^2 m_1)}{m_1^2 m_2} Z_2 + \frac{(c_2 k_2 m_1 + c_1 m_2 k_2 + c_2 m_2 k_2 + c_1 m_2 k_1 + c_2 m_2 k_1)}{m_1^2 m_2} Z_1 + \frac{c_2 fadot}{m_1 m_2}\]

\[eq26 := Z_{2 \text{ddot dot dot}} = \frac{(-c_2^2 m_1^2 + k_2 m_2 m_1^2 - c_2^2 m_2 m_1 - c_2 m_2 c_1 m_1) fa}{m_1^3 m_2} + (\frac{\text{other terms}}{m_1^3 m_2^2}) + (c_2 m_2 c_1 k_2 m_1 - k_2^2 m_2 m_1^2 + c_1^2 m_2 k_2 - k_1 m_2 m_1 k_2 + 2 c_2 m_2 c_1 k_2 + c_2 m_1 k_2 - k_2^2 m_2 m_1 + m_2^2 c_2^2 k_2 + 2 c_2^2 m_2 k_1 m_2) Z_3 / (m_1^3 m_2^2) + (\frac{\text{other terms}}{m_1^3 m_2}) + (+2 c_2^2 m_2 m_1 - c_1^2 m_2^2 - c_2^2 m_1^2 - 2 c_2 m_2 c_1 m_1 - 3 c_1^2 m_2^2 c_2 + 2 c_1 m_2^2 k_1 m_1 + 2 c_2 m_2^2 k_1 m_1 - m_2^2 c_3 + 2 c_2 m_2^2 k_2 m_1 + 2 c_1 m_2^2 k_2 m_1 - 3 c_2^2 m_2^2 c_1 + 2 c_2 k_2 m_1^2 m_2) Z_2 / (m_1^3 m_2^2) + (\frac{\text{other terms}}{m_1^3 m_2^2}) + (-2 c_2 m_2 c_1 k_1 - c_2^2 m_1^2 k_2 - c_2 m_1 m_2 k_1 + k_2^2 m_2 m_1 + m_2^2 c_2^2 k_2 - c_2 m_2 c_1 k_2 m_1 - c_1^2 m_2^2 k_1 - c_1^2 m_2^2 k_2 - 2 c_2^2 m_2 k_2 m_1 - m_2^2 c_2^2 k_1 - 2 c_2^2 c_1 k_2 + 2 c_1 m_2^2 c_1 k_2 + 2 k_1 m_2^2 m_1 m_2 - 2 c_2^2 c_1 k_2 + k_1^2 m_2^2 m_1) Z_1 / (m_1^3 m_2^2) + (\frac{c_2 fadot}{m_1 m_2})\]

\[eq27 := Z_{3 \text{ddot dot dot}} = \frac{-c_2 fadot}{m_2^2} + \frac{(-k_2 m_2 m_1 + c_2^2 m_1 + c_2^2 m_2)}{m_2^2 m_1} Z_4 + \frac{(c_2 k_2 m_1 + c_2 m_2 k_2)}{m_2^2 m_1} Z_3 + \frac{(-k_2 m_2 m_1 - c_2^2 m_2 m_1 - c_2 m_2 c_1 m_1)}{m_2^2 m_1} Z_2 + \frac{(-c_2 k_2 m_1 - c_2 m_2 k_2 - c_2 m_2 k_1)}{m_2^2 m_1} Z_1 + \frac{fadot}{m_2}\]

\[eq28 := Z_{4 \text{ddot dot dot}} = \frac{c_2 fadot}{m_2^2} - \frac{(-c_2^2 m_1^2 - c_2^2 m_2 m_1 + k_2 m_2 m_1^2)}{m_2^3 m_1^2} - (\frac{\text{other terms}}{m_2}) - (\frac{\text{other terms}}{m_2^3 m_1^2}) + \frac{fadot}{m_2}\]

\[\text{This represents a set of equations involving } Z_{1 \text{ddot dot dot}}, Z_{2 \text{ddot dot dot}}, Z_{3 \text{ddot dot dot}}, Z_{4 \text{ddot dot dot}}, \text{ and } fa, \text{ with various terms involving } k_1, k_2, m_1, \text{ and } m_2.\]
\begin{align*}
eq 29 := & \frac{Z_{1\dot{\ddot{\ddot{\ddot{\dot{}}}}}} = \left( -c_2 k_2 m_1 - c_1 m_2 k_2 - c_2 m_2 k_2 \right) Z_{3\dot{\dot{\dot{\dot{\dot{}}}}}}}{m_1^2 m_2} + \frac{\left( -k_1 m_2 m_1 + c_1^2 m_2 + 2 c_1 m_2 c_2 + c_2^2 m_2 - k_2 m_2 m_1 + c_2^2 m_1 \right) Z_{3\dot{\dot{\dot{\dot{\dot{}}}}}}}{m_1^2 m_2} + \frac{\left( c_2 k_2 m_1 + c_1 m_2 k_2 + c_2 m_2 k_2 + c_1 m_2 k_1 + c_2 m_2 k_1 \right) Z_{1\dot{\dot{\dot{\dot{\dot{}}}}}}}{m_1^2 m_2} + \frac{c_2 fadot}{m_1 m_2}.
\end{align*}
\begin{align*}
eq 30: & \quad Z_{4\text{dotdotdot}} = \frac{(-c_2^2 m_1^2 + k_2 m_2 m_1^2 - c_2^2 m_2 m_1 - c_2 m_2 c_1 m_1) \text{fadot}}{m_1^3 m_2^2} + (\nonumber \\
& -2 c_2 k_2 m_1^2 m_2 + c_2^3 m_1^2 + 2 c_2 m_2 m_1 + c_2^2 m_2 c_1 m_1 + m_2^2 c_2^3 
& - 2 c_2 m_2^2 k_2 m_1 + 2 c_2^2 m_2^2 c_1 - c_1 m_2^2 k_2 m_1 + c_1^2 m_2^2 c_2 - c_2 m_2^2 k_1 m_1) 
& Z_4\text{dot}/(m_1^3 m_2^2) + (c_2 m_2 c_1 k_2 m_1 - k_2 m_2 m_1^2 + c_1^2 m_2^2 k_2 - k_1 m_2^2 m_1 k_2 
& + 2 c_2 m_2^2 c_1 k_2 + c_2^2 m_1^2 k_2 - k_2 m_2^2 m_1 + m_2^2 c_2^2 k_2 + 2 c_2 m_2 k_2 m_1) 
& Z_3\text{dot}/(m_1^3 m_2^2) + (-2 c_3^2 m_2 m_1 - c_1^3 m_2^2 - c_3^3 m_1^2 - 2 c_2^2 m_2 c_1 m_1 
& - 3 c_1^2 m_2^2 c_2 + 2 c_2 m_2 k_1 m_1 + 2 c_2 m_2^2 k_1 m_1 - m_2^2 c_2^3 + 2 c_2 m_2^2 k_2 m_1 
& + 2 c_1 m_2^2 k_2 m_1 - 3 c_2^2 m_2^2 c_1 + 2 c_2 k_2 m_1^2 m_2) Z_2\text{dot}/(m_1^3 m_2^2) + (\nonumber \\
& -2 c_2 m_2^2 c_1 k_1 - c_2^2 m_1^2 k_2 - c_2^2 m_1 m_2 k_1 + k_2^2 m_2^2 m_1 + k_2^2 m_2 m_1^2 
& - c_2 m_2 c_1 k_2 m_1 - c_1^2 m_2^2 k_1 - c_1^2 m_2^2 k_2 - 2 c_2 m_2 k_2 m_1 - m_2^2 c_2^2 k_1 
& - 2 c_2 m_2^2 c_1 k_2 + 2 k_1 m_2^2 m_1 k_2 - m_2^2 c_2^2 k_2 + k_1^2 m_2^2 m_1) Z_1\text{dot}/(m_1^3 m_2^2) 
& + c_2 \text{fadot}/m_1 m_2 
& 
& \text{eq31} := Z_{3\text{dotdotdot}} = -\frac{c_2 \text{fadot}}{m_2^2} + \frac{(-k_2 m_2 m_1 + c_2^2 m_1 + c_2^2 m_2) Z_4\text{dot}}{m_2^2 m_1} 
& + \frac{(c_2 k_2 m_1 + c_2 m_2 k_2) Z_3\text{dot}}{m_2^2 m_1} 
& + \frac{(k_2 m_2 m_1 - c_2^2 m_2 - c_1 m_2 c_2 - c_2^2 m_1) Z_2\text{dot}}{m_2^2 m_1} 
& + \frac{(-c_2 k_2 m_1 - c_2 m_2 k_2 - c_2 m_2 k_1) Z_1\text{dot}}{m_2^2 m_1} + \frac{\text{fadot}/m_2}{m_2} 
& 
& \text{eq32} := Z_{4\text{dotdotdot}} = -\frac{c_2 \text{fadot}}{m_2^2} - \frac{(-c_2^2 m_1^2 - c_2^2 m_2 m_1 + k_2 m_2 m_1^2) \text{fadot}}{m_2^2 m_1} - (\nonumber \\
& c_2^3 m_1^2 + m_2^2 c_3^3 + 2 c_2^3 m_2 m_1 - 2 c_2 k_2 m_1^2 m_2 + c_2^2 m_2^2 c_1 - 2 c_2 m_2^2 k_2 m_1 
& ) Z_4\text{dot}/(m_2^3 m_1^2) - (-k_2^2 m_2 m_1^2 + c_2^2 m_1^2 k_2 + c_2 m_2^2 c_1 k_2 - k_2^2 m_2^2 m_1 
& + m_2^2 c_2^2 k_2 + 2 c_2^2 m_2 k_2 m_1) Z_3\text{dot}/(m_2^3 m_1^2) - (c_2 m_2^2 k_1 m_1 - c_2^3 m_1^2 
& - c_1^2 m_2^2 c_2 - 2 c_2^3 m_2 m_1 + c_1 m_2^2 k_2 m_1 - m_2^2 c_2^3 + 2 c_2 m_2^2 k_2 m_1 
& - c_2^2 m_2 c_1 m_1 - 2 c_2^2 m_2^2 c_1 + 2 c_2 k_2 m_1^2 m_2) Z_2\text{dot}/(m_2^3 m_1^2) - (\nonumber \\
& - c_2 m_2^2 c_1 k_1 + k_2^2 m_2^2 m_1 - c_2 m_2^2 k_1 k_2 - c_2^2 m_1 m_2 k_1 + 2 c_2^2 m_2 k_2 m_1 
& - c_2^2 m_2^2 k_1 - m_2^2 c_2^2 k_1 + k_2^2 m_2^2 m_1 + k_1 m_2^2 m_1 k_2 - m_2^2 c_2^2 k_2) Z_1\text{dot}/(m_2^3 m_1^2) + \frac{\text{fadot}/m_2}{m_2} 
& > \text{eq33} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq29})))) 
& > \text{eq34} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq30})))) 
& > \text{eq35} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq31})))) 
& > \text{eq36} := \text{algsubs}(\text{eq1}, \text{algsubs}(\text{eq2}, \text{algsubs}(\text{eq3}, \text{algsubs}(\text{eq4}, \text{eq32})))) 
& \end{align*}
eq33 := \(Z1dotdotdotdot = (-c2^2 m1^2 fa - c2^3 m1^2 Z2 + c2^3 m1^2 Z4 + c2 fadot m1^2 m2
+ c2^2 m1^2 k2 Z3 - c2^2 m1^2 k2 Z1 - k2^2 m2 m1^2 Z3 - 2 c2^2 m2 c1 Z2 m1
+ 2 k1 m2^2 m1 k2 Z1 - c2^2 m1 m2 k1 Z1 + 2 k2 m2^2 m1 c1 Z2 - k2^2 m2 m1 Z3
+ k2^2 m2^2 m1 Z1 + k2^2 m2 m1^2 Z1 + k2 m2 m1^2 fa - c2^2 m2 fa m1
- 2 c2^3 m2 Z2 m1 - m2^2 c1^3 Z2 + m2^2 c2^3 Z4 - m2^2 c2^3 Z2 - 3 c2 m2^2 c1^2 Z2
+ 2 c2^2 m2^2 c1 Z4 + 2 c2 m2^2 k1 m1 Z2 - 2 c2 m2^2 c1 k1 Z1 + 2 c2 m2^2 c1 k2 Z3
- 2 c2 m2^2 c1 k2 Z1 + 2 c2 m2^2 k2 m1 Z2 + 2 c2^3 m2 Z4 m1 - 2 k2 m2 m1^2 c2 Z4
+ 2 k2 m2 m1^2 c2 Z2 + 2 c2^2 m2^2 Z3 m1 - 2 c2^2 m2 k2 Z1 m1
- 2 c2 m2^2 k2 Z4 m1 + c1 m2 c2^2 Z4 m1 - c1 m2 c2 fa m1 + c1 m2 c2 k2 Z3 m1
- c1 m2 c2 k2 Z1 m1 - k2 m2^2 c1 Z4 m1 + 2 m2^2 k1 m1 c1 Z2 - m2^2 k1 m1 k2 Z3
- m2^2 k1 m1 c2 Z4 - m2^2 c1^2 k1 Z1 + m2^2 c1^2 k2 Z3 - m2^2 c1^2 k2 Z1
+ m2^2 c1^2 c2 Z4 - m2^2 c2^2 k1 Z1 - 3 m2^2 c2^2 c1 Z2 + m2^2 c2^2 k2 Z3
- m2^2 c2^2 k2 Z1 + m2^2 k1^2 m1 Z1)/(m1^3 m2^2)\)
\textbf{eq34} := Z2dotdotdotdot = \left(-fadot m^2 m^1 c_1 c_2 + c_2 m^2 k_1 m^1 k_2 Z^3 - 3 c_2 m^2 k_1 m^1 k_2 Z^1 + c_2^2 m^2 k_1 m^1 Z_4 - 2 c_2^2 m^2 k_1 m^1 Z_2 - 3 c_2^3 m_2 m_1^2 k_2 Z_3 + 3 c_2^3 m_2 m_1^2 k_2 Z_1 + c_1 m_2^3 k_2 m_1^2 Z_3 - c_1 m_2^2 k_2^2 m_1^2 Z_1 + 2 c_1 m_2^2 k_2 m_1^2 Z_4 + m_2 c^2 m_1^2 k_1 Z_1 + 2 m_2^2 c^2 m_1 k_1 Z_1 - 4 c_1 m_2^2 m_1^2 c_2 Z_2 + 4 c_2 m_2^2 k_2^2 m_1^2 Z_3 - 4 c_2 m_2^2 k_2^2 m_1^2 Z_4 + 6 c_2^2 m_2^2 k_2 m_1^2 Z_4 - 6 c_2^2 m_2^2 k_2 m_1^2 Z_2 - c^2 m_2 c_1 m_1^2 k_2 Z_3 + c_2^2 m_2 c_1 m_1^2 k_2 Z_1 - c_2^3 m_2 c_1 m_1^2 Z_4 + 2 c_3^2 m_2 c_1 m_1^2 Z_2 + 2 c_2 k_2^2 m_1^3 m_2 Z_3 - 2 c_2 k_2^2 m_1^3 m_2 Z_1 + 3 c_2^2 k_2 m_1^3 m_2 Z_4 - 3 c_2^2 k_2 m_1^3 m_2 Z_2 - 2 c_2 m_2^2 k_1 m_1^2 fa - c_1 m_2^2 k_2 m_1^2 fa - 2 c_2 m_2^2 k_2 m_1^2 fa + c_2^2 m_2 c_1 m_1^2 fa - 2 c_2 k_2 m_1^3 m_2 fa + 3 m_2^3 c_1^2 c_2 k_1 Z_1 - 3 m_2^3 c_1^2 c_2 k_2 Z_3 + 3 m_2^3 c_1^2 c_2 k_2 Z_1 + 2 m_2^2 c^2 c_1 m_1 k_1 Z_1 + 4 k_2 m_2^2 c_1 m_2 Z_4 m_1 + 3 k_2 m_2^3 c_2 Z_4 m_1 + k_2 m_2^2 c_1 Z_4 m_1 - c_1^2 m_2^2 c_2 Z_4 m_1 + 3 c_1^2 m_2^2 c_2 Z_2 m_1 - 3 m_2^2 c^2 k_2 Z_3 m_1 + 3 m_2^2 c^2 k_2 Z_1 m_1 - 4 c_2^3 m_2^2 c_1 Z_4 m_1 + 6 c_2^3 m_2^2 c_1 Z_2 m_1 + c_1^2 m_2^2 c_2 Z_2 m_1 + c_2^3 m_1^3 Z_4^2 + c_2^2 c_1 Z_2 m_1 + c_2^2 m_2^2 c_1 k_2 Z_3 m_1 + c_1^2 m_2^2 c_2 k_2 Z_1 m_1 - 4 c_2^2 m_2^2 c_1 k_2 Z_3 m_1 + 4 c_2^2 m_2^2 c_1 k_2 Z_1 m_1 + m_2^3 c_1^4 Z_2 - m_2^3 c_1^4 Z_4 + m_2^3 c_1^4 Z_2 - c_1^2 m_2^2 c_2 Z_3 m_1 + c_1^2 m_2^2 c_2 k_2 Z_1 m_1 - 4 c_2^2 m_2^2 c_1 k_2 Z_3 m_1 + 4 c_2^2 m_2^2 c_1 k_2 Z_1 m_1 + m_2^3 c_1^4 Z_2 - m_2^3 c_1^4 Z_4 + m_2^3 c_1^4 Z_2 - c_1^2 m_2^2 c_2 Z_1 m_1 - 3 m_2^3 c_2^2 k_2 m_1 Z_2 - 3 m_2^3 c_2^2 k_2 Z_1 m_1 - 6 m_2^2 c_1 k_2 m_1 Z_2 + 2 m_2^2 c_2 k_2 m_1 Z_3 - k_2^2 m_2^2 m_1^3 Z_4 - 2 m_2^2 c_2 k_2 m_1 Z_1 - 3 m_2^3 c_2^2 k_2 m_1 Z_2 - 2 m_2^3 c_1 k_1^2 m_1 Z_1 - 3 m_2^3 c_1^2 k_1 m_1 Z_2 + 2 m_2^3 c_1 k_1 m_2 Z_3 + c_2 fadotdotdotdot m_1^3 c_2^2 m_1^2 fa m_1 - 3 m_2^2 c_1^2 Z_4 m_1 + 3 m_2^2 c_1^2 Z_2 m_1 + 6 m_2^3 c_2^2 c_1^2 Z_2 - 3 m_2^3 c_2^2 c_1 Z_4 + 4 m_2^2 c_2^2 c_1 Z_2 + 2 m_2^2 c_1 k_1 m_1 c_2 Z_4 + 3 m_2^3 c_2^2 c_1 k_1 Z_1 - 3 m_2^3 c_2^2 c_1 k_2 Z_3 + 3 m_2^3 c_2^2 c_1 k_2 Z_1 + 4 m_2^3 c_1^3 c_2 Z_2 - 3 m_2^3 c_1^2 c_2 Z_4 + m_2^3 c_1^3 k_1 Z_1 - 2 m_2^3 c_2 k_2^2 m_1 Z_1 - 6 m_2^3 c_2 k_1 m_1 c_1 Z_2 + 2 m_2^3 c_2 k_1 m_1 k_2 Z_3 - 4 m_2^3 c_2 k_1 m_1 k_2 Z_1 + 2 m_2^3 c_2 k_1 m_1 Z_4 - 3 m_2^3 c_2^2 k_1 m_1 Z_2 + m_2^2 k_2^2 m_1^3 Z_2 - k_2 m_2^3 k_1 m_1^2 Z_4 + m_2^3 k_2^2 m_1^2 Z_2 + m_2^3 k_1^2 m_1^2 Z_2 - m_2^3 c_1^3 k_2 Z_3 - m_2^3 c_1^3 c_2 Z_4 + m_2^3 c_2^3 k_1 Z_1 - m_2^3 c_2^3 k_2 Z_3 + m_2^3 c_2^3 k_2 Z_1 + m_2^3 c_1^3 k_2 Z_1) / (m_1^4 m_2^3)
eq35 := Z3dotdotdotdot = \(-fadotdot m2^2 m1^2 - c2^2 m1^2 fa - c2^3 m1^2 Z2 + c2^3 m1^2 Z4 + c2 fadot m1^2 m2 + c2^2 m1^2 k2 Z3 - c2^2 m1^2 k2 Z1 - k2^2 m2 m1^2 Z3
- c2^2 m2 c1 Z2 Z1 + k1 m2^2 m1 k2 Z1 - c2^2 m1 m2 k1 Z1 + k2 m2^2 m1 c1 Z2
- k2^2 m2^2 m1 Z3 + k2^2 m2^2 m1 Z1 + k2^2 m2 m1^2 Z1 + k2 m2 m1^2 fa
- c2^2 m2 fa m1 - 2 c2^3 m2 Z2 Z1 + m2^2 c2^3 Z4 - m2^2 c2^3 Z2 - c2 m2^2 c1^2 Z2
+ c2^2 m2^2 c1 Z4 + c2 m2^2 k1 m1 Z2 - c2 m2^2 c1 k1 Z1 + c2 m2^2 c1 k2 Z3
- c2 m2^2 c1 k2 Z1 + 2 c2 m2^2 k2 m1 Z2 + 2 c2^3 m2 Z4 m1 - 2 k2 m2 m1^2 c2 Z4
+ 2 k2 m2 m1^2 c2 Z2 + 2 c2^2 m2 k2 Z3 m1 - 2 c2^2 m2 k2 Z1 m1
- 2 c2 m2^2 k2 Z4 m1 - m2^2 c2^2 c1 Z1 - 2 m2^2 c2^2 c1 Z2 + m2^2 c2^2 k2 Z3
- m2^2 c2^2 k2 Z1)/(m2^3 m1^2)

eq36 := Z4dotdotdotdot = \(-2 c2 m2^2 k1 m1^2 k2 Z1 - c2^2 m2^2 k1 m1^2 Z2
- 3 c2^3 m2 m1^2 k2 Z3 + 3 c2^3 m2 m1^2 k2 Z1 + m2 c2^3 m1^2 k1 Z1
+ 2 m2^2 c2^3 m1 k1 Z1 - 2 c1 m2^2 k2 m1^2 c2 Z2 + 4 c2 m2^2 k2^2 m1^2 Z3
- 4 c2 m2^3 k2^2 m1^2 Z1 + 6 c2^2 m2^2 k2 m1^2 Z4 - 6 c2^2 m2^2 k2 m1^2 Z2
+ c2^3 m2 c1 m1^2 Z2 + 2 c2 k2^2 m1^3 m2 Z3 - 2 c2 k2^2 m1^3 m2 Z1
+ 3 c2^2 m2 k1^2 m2 Z4 - 3 c2^2 k2^2 m1^3 m2 Z2 - 2 c2 m2^2 m2 k2 m1^2 fa
- 2 c2 k2 m1^3 m2 fa + m2^3 c1^2 c2 k1 Z1 - m2^3 c1^2 c2 k2 Z3 + m2^3 c1^2 c2 k2 Z1
+ m2^2 c2^2 c1 m1 k1 Z1 + 2 k2 m2^3 c1 c2 Z4 m1 + 3 k2 m2^3 c2^2 Z4 m1
+ c1^2 m2^2 c2^2 Z2 m1 - 3 m2^2 c2^3 k2 Z3 m1 + 3 m2^2 c2^3 k2 Z1 m1
- 2 c2^3 m2^2 c1 Z4 m1 + 4 c2^3 m2^2 c1 Z2 m1 + c2^3 m1^3 fa + c2^4 m1^3 Z2
+ c2^2 m2^2 c1 fa m1 - 2 c2^2 m2^2 c1 k2 Z3 m1 + 2 c2^2 m2^2 c1 k2 Z1 m1
- m2^3 c2^4 Z4 + m2^3 c2^4 Z2 - c2^4 m1^3 Z4 - fadotdotdotdot m2^3 m1^3
+ m2^3 k1 m1^2 k2 Z2 - k2^2 m2^3 m1^2 Z4 + fadot m2^2 m1^3 k2 - fadot m2^2 m1^2 c2^2
- fadot m2 m1^3 c2^2 - c2^3 m1^3 k2 Z3 + c2^3 m1^3 k2 Z1 - 3 c2^4 m2 m1^2 Z4
+ 3 c2^4 m2 m1^2 Z2 + 2 c2^3 m2 m1^2 fa - m2^3 c1 k2 m1 k1 Z1 - m2^3 c1^2 k2 m1 Z2
+ m2^3 c1 k2^2 m1 Z3 - m2^3 c1 k2^2 m1 Z1 - 4 m2^3 c1 k2 m1 c2 Z2
+ 2 m2^3 c2 k2^2 m1 Z3 - k2^2 m2^2 m1^3 Z4 - 2 m2^3 c2 k2^2 m1 Z1
- 3 m2^3 c2^2 k2 m1 Z2 + c2 fadotdot m1^3 m2^2 + m2^2 c2^3 fa m1 - 3 m2^2 c2^4 Z4 m1
+ 3 m2^2 c2^4 Z2 m1 + 3 m2^3 c2^2 c1^2 Z2 - 2 m2^3 c2^3 c1 Z4 + 3 m2^3 c2^3 c1 Z2
+ 2 m2^3 c2^2 c1 k1 Z1 - 2 m2^3 c2^2 c1 k2 Z3 + 2 m2^3 c2^2 c1 k2 Z1 + m2^3 c1^3 c2 Z2
- m2^3 c1^2 c2^2 Z4 - m2^3 c2 k1^2 m1 Z1 - 2 m2^3 c2 k1 m1 c1 Z2
+ m2^3 c2 k1 m1 k2 Z3 - 3 m2^3 c2 k1 m1 k2 Z1 + m2^3 c2 k1 m1 Z4
- 2 m2^3 c2^2 k1 m1 Z2 + m2^2 k2^2 m1^3 Z2 + m2^3 k2^2 m1^2 Z2 + m2^3 c2^3 k1 Z1
- m2^3 c2^3 k2 Z3 + m2^3 c2^3 k2 Z1)/(m2^4 m1^3)
\[
\text{eq37} := \frac{Z_1^{\cdot\cdot\cdot\cdot}}{m_1^3 m_2^2} + \frac{Z_4}{m_1^3 m_2^2} + \frac{Z_3}{m_1^3 m_2^2} + \frac{Z_2}{m_1^3 m_2^2} + \frac{Z_1}{m_1^3 m_2^2} + \frac{c_2 f_a}{m_1 m_2} + \left( -2 c_2 k_2 m_1^2 m_2 + c_2^3 m_1^2 + 2 c_2^3 m_2 m_1 + c_2^2 m_2 c_1 m_1 + m_2^2 c_2^3 \\
-2 c_2 m_2^2 k_2 m_1 + 2 c_2^2 m_2^2 c_1 - c_1 m_2^2 k_2 m_1 + c_1^2 m_2^2 c_2 - c_2 m_2^2 k_2 m_1 \\
\right) Z_2/(m_1^3 m_2^2) + \left( -2 c_2^3 m_2 m_1 - c_1^3 m_2^2 - c_2^3 m_1^2 - 2 c_2^2 m_2 c_1 m_1 - 3 c_1^2 m_2^2 c_2 \\
+ 2 c_1 m_2^2 k_1 m_1 + 2 c_2 m_2^2 k_1 m_1 - m_2^2 c_2^3 + 2 c_2 m_2^2 k_2 m_1 + 2 c_1 m_2^2 k_2 m_1 \\
- 3 c_2^2 m_2^2 c_1 + 2 c_2 k_2 m_1^2 m_2) Z_2/(m_1^3 m_2^2) + \left( -2 c_2 m_2^2 c_1 k_1 - c_2^2 m_1^2 k_2 \\
- c_2^2 m_1 m_2 k_1 + k_2^2 m_2^2 m_1 + k_2^2 m_2 m_1^2 - c_2 m_2 c_1 k_2 m_1 - c_1^2 m_2^2 k_2 \\
- c_1^2 m_2^2 k_2 - 2 c_2^2 m_2 k_2 m_1 - m_2^2 c_2^2 k_1 - 2 c_2 m_2^2 c_1 k_2 + 2 k_1 m_2^2 m_1 k_2 \\
- m_2^2 c_2^2 k_2 + k_1^2 m_2^2 m_1) Z_1/(m_1^3 m_2^2) + c_2 f_a \right) \]

eq38 := Z2\text{dotdotdot}\dot{=} \\
\frac{(-m^2 m^3 c_1 c_2 + m^2 m^3 k_2 - m^2 m^3 c_2^2 - m^2 m^3 c_2^2) \, \text{fadot}}{m_1 m_2^3} + (c_2^3 m_1^3) \\
- c_2 m^2 k_1 m_1^2 + c_1^2 m^2 c_2 m_1 - 2 m^2 c_2 k_2 m_1^2 + c_2^2 m_2 c_1 m_1^2 \\
- 2 c_2 k_2 m_1^3 m_2 - c_1 m_2^2 k_2 m_1^2 + m^2 c_2^3 m_1 + 2 m_2 c_2^3 m_1^2 \\
+ 2 m^2 c_2^2 c_1 m_1 f_a/(m_1 m_2^3) + (m^2 c_2^2 m_1^2 k_1 + 3 m_3 c_2^3 k_2 m_1 \\
- c_2^3 m_2 c_1 m_1^2 - c_1^2 m^2 c_2^2 m_1 + 2 m^2 c_2 c_1 k_2 m_1^2 - c_2^4 m_1^3 - m_3 c_2^2 m_1^2 \\
+ 3 m_2 c^2 m_1^3 k_2 - m_3 c_2^4 + 4 m^3 c_2 c_1 k_2 m_1 - 3 c_2^4 m_2 m_1^2 \\
+ m_2^3 c_1^2 k_2 m_1 - m_2^2 k_2 m_1^3 - 4 c_3^3 m_2^2 c_1 m_1 + 6 m_2^3 c_2^2 k_2 m_1^2 \\
- 3 m_2^2 c_4^3 m_1 - 3 m_3^3 c_3^3 c_1 + 2 m_2^3 c_2 c_1 k_1 m_1 - 3 m_3^3 c_2 c_1^2 \\
+ 2 m_2^3 c_2^2 k_1 m_1 - m_2^3 k_1 m_1^2 k_2 - m_2^3 c_2^3 c_1^2 Z_4 / (m_1 m_2^3) + (-c_2^3 m_1^3 k_2 \\
- 4 c_2^2 m^2 c_1 k_2 m_1 - 3 m_2^2 c_2^3 k_2 m_1 + 2 m_2^2 c_2 k_2^2 m_1 + c_2^2 m_2^2 k_1 m_1^2 k_2 \\
- 3 m_3^3 c_1^2 c_2 k_2 - 3 c_3^3 m_2 m_1^2 k_2 + c_1 m_2^2 k_2^2 m_1^2 + 2 m_2^3 c_1 k_2 m_1 k_1 \\
+ 4 c_2 m_2^2 k_2^2 m_1^2 - 3 m_2^3 c_2^2 c_1 k_2 - c_2^2 m_2 c_1 m_1^2 k_2 + 2 m_3^3 c_2 c_1 m_1 k_2 \\
+ 2 c_2 k_2^2 m_1^3 m_2 + 2 m_2^3 c_1 k_2^2 m_1 - m_2^3 c_1^3 k_2 - c_1^2 m_2^2 c_2 k_2 m_1 \\
- m_2^3 c_3^3 k_2) Z_3 / (m_1 m_2^3) + (-6 m_2^3 c_2 c_1 k_2 m_1 - 4 m_2^2 c^2 c_2 c_1 k_2 m_1^2 \\
- 6 m_2^3 c_2 c_1 k_1 m_1 + c_2^4 m_1^3 + m_2^3 c_1^4 + m_2^3 c_2^4 + 3 c_2^4 m_2 m_1^2 \\
+ 3 m_2^3 c_2^4 m_1 + 6 m_2^3 c_2^2 c_1^2 + 4 m_2^3 c_2^3 c_1 + 4 m_2^3 c_1^3 c_2 + 2 c_2^3 c_2 m_2 c_1 m_1^2 \\
+ 3 c_1^2 m_2^2 c_2^2 m_1 + 6 c_2^3 m_2^2 c_1 m_1 - 3 m_2 c_2^2 m_1^3 k_2 + m_2^3 c_2^2 m_1^2 \\
+ m_2^3 k_2^2 m_1^3 + m_2^3 k_1^2 m_1^2 - 3 m_2^3 c_1^2 k_2 m_1 - 2 m_2^2 c_2^2 m_1^2 k_1 \\
- 6 m_2^3 c_2^2 k_2 m_1^2 + 2 m_2^3 k_1 m_1^2 k_2 - 3 m_2^3 c_1^2 k_2 m_1 - 3 m_2^3 c_2^2 k_1 m_1 \\
- 3 m_2^3 c_2^2 k_2 m_1 Z_2 / (m_1 m_2^3) + (-4 m_2^3 c_2 c_1 k_1 m_2 + c_2^3 m_1^3 k_2 \\
+ m_2^3 c_1^3 k_1 + m_2^3 c_2^3 k_2 + m_2^3 c_3^3 k_2 + m_2^3 c_2^3 k_2 - 3 c_2 m_2^2 k_1 m_1^2 k_2 \\
+ c_2^2 m_2 c_1 m_1^2 k_2 + 2 m_2^2 c_2^2 c_1 m_1 k_1 + c_2^2 m_2 c_2 k_2 m_1 \\
+ 4 c_2^2 m^2 c_1 k_2 m_1 - 4 m_2^3 c_1 k_2 m_1 + 3 c_2^3 m_2 m_1^2 k_2 - c_1 m_2^2 k_2^2 m_1^2 \\
+ m_2 c_2^3 m_1^2 k_1 + 2 m_2^2 c_2^3 m_1 k_1 - 4 c_2 m_2^2 k_2^2 m_1^2 - 2 c_2 k_2^2 m_1^3 m_2 \\
+ 3 m_2^3 c_1^2 c_2 k_1 + 3 m_2^3 c_1^2 c_2 k_2 + 3 m_2^2 c_2^3 k_2 m_1 - 2 m_2^3 c_1 k_2^2 m_1 \\
- 2 m_2^3 c_2 k_2 m_1 - 2 m_2^3 c_1 k_2 k_1 + 3 m_2^3 c_2 c_2 c_1 k_1 + 3 m_2^3 c_2^2 c_1 k_2 \\
- 2 m_2^3 c_2 k_1^2 m_1 Z_1 / (m_1 m_2^3) + \frac{c_2 \text{fadotdotdot}}{m_1 m_2}$
\[ eq39 := Z_3 \dot{f} \dot{a} = \frac{-c_2 f \dot{a}}{m_2^2} - \frac{(-c_2^2 m_1^2 - c_2^2 m_2 m_1 + k_2 m_2 m_1^2) f a}{m_2^3 m_1^2} \]
\[ (c_2^3 m_1^2 + m_2^2 c_2^3 + 2 c_2^3 m_2 m_1 - 2 c_2 k_2 m_2 m_2 + c_2^2 m_2^2 c_1 - 2 c_2 m_2^2 k_2 m_1) Z_4 \]
\[ - (-k_2^2 m_2 m_1^2 + c_2^2 m_1^2 k_2 + c_2 m_2^2 c_1 k_2 - k_2^2 m_2^2 m_1 + m_2^2 c_2^2 k_2 + 2 c_2^2 m_2 k_2 m_1) Z_3/(m_2^3 m_1^2) - (c_2 m_2^2 k_1 m_1 - c_2^3 m_1^2 - c_1^2 m_2^2 c_2 \]
\[ - 2 c_2^3 m_2 m_1 + c_1 m_1 m_2^2 - m_2^2 m_1 - m_2^3 m_2 c_3 + 2 c_2 m_2^2 k_2 m_1 - c_2^2 m_2 c_1 m_1 \]
\[ - 2 c_2^3 m_2 m_1 + 2 c_2 k_2 m_1^2 m_2) Z_2/(m_2^3 m_1^2) - (-c_2 m_2^2 c_1 k_1 + k_2^2 m_2 m_1^2 \]
\[ - c_2 m_2^2 c_1 k_2 - c_2^2 m_1 m_1 k_1 - 2 c_2^2 m_2^2 m_2 k_1 - c_2^2 m_2^2 k_2 - m_2^2 m_2^2 k_1 \]
\[ + k_2^2 m_2^2 m_1 + k_1 m_2^2 k_1 k_2 - m_2^2 c_2^2 k_2) Z_1/(m_2^3 m_1^2) + \frac{f \dot{a} \dot{d} \dot{t}}{m_2^2} \]

\[ eq40 := Z_4 \dot{f} \dot{a} \dot{d} = \frac{-c_2 f \dot{a} \dot{d}}{m_2^2} - \frac{(m_2^2 m_1^3 k_2 - m_2^2 m_1^2 c_2^2 - m_2 m_1^3 c_2^2) f \dot{a} \dot{d}}{m_2^4 m_1^3} \]
\[ - 2 m_2^2 c_2 k_2 m_1^2 + m_2^2 c_2^2 c_1 m_1 + 2 m_2 c_2^3 m_1^2 + m_2^2 c_2^3 m_1 + c_2^3 m_1^3) f a/(m_2^4 m_1^3) \]
\[ + 3 m_2 c_2^3 m_1^3 k_2 - m_2^2 c_2^3 m_1^2 + 2 m_2^3 c_2 c_1 k_2 m_1 - m_2^3 c_2 k_2 m_1^3 \]
\[ - 2 c_2^3 m_2^2 c_1 m_1 - 3 c_2^4 m_2 m_1^2 - 3 m_2^2 c_2^4 m_1 - 2 m_2^3 c_2^3 c_1 - m_2^3 c_2^2 c_1^2 \]
\[ + 3 m_2^3 c_2^2 k_1 m_1) Z_4/(m_2^4 m_1^3) - (-c_2^3 m_2 m_1^2 k_2 + m_2^3 c_1 k_2 m_1^2 \]
\[ + 4 c_2 c_2^2 m_2^2 m_2^2 m_2 m_2 + 2 m_2^3 c_2 k_2^2 m_1 \]
\[ - 2 c_2 c_2^2 c_1 k_2 m_1 - 2 m_2^3 c_2^2 c_1 k_2 - 2 m_2^3 c_1 k_2) Z_3/(m_2^4 m_1^3) - (c_2^3 m_2 c_1 m_1^2 c_2^2 c_2 \]
\[ - m_2 c_1^2 k_2 m_1 + m_2 c_2 c_1^2 + m_2^3 c_1^2 k_2 + c_2^2 m_2^2 c_2 \]
\[ - 2 m_2^2 c_2 c_1 k_2 m_1^2 - 6 m_2^2 c_2^2 k_2 m_1^2 + 3 m_2^2 c_2^4 m_1 + 3 m_2^3 c_2^2 c_1^2 \]
\[ + 3 c_2^4 m_2 m_1^2 + 3 m_2^3 c_2^3 c_1 - 3 m_2 c_2^2 m_1^3 k_2 + m_2^3 c_1^3 c_2 \]
\[ - 4 m_2^3 c_2 c_1 k_2 m_1 - 2 m_2^3 c_2 c_1 k_2 m_1 - m_2^2 c_2^2 m_1^2 k_1 - 3 m_2^3 c_2^2 k_2 m_1 \]
\[ - 2 m_2^3 c_2^2 k_1 m_1 + m_2^2 c_2^2 m_1^3 + m_2^3 c_2^2 m_1^2 + 4 c_2^3 m_2^2 c_1 m_1) Z_2/(m_2^4 m_1^3) \]
\[ - (-2 c_2 m_2^2 k_2 m_1^2 k_2 + 3 c_2^3 m_2 m_1^2 k_2 - 2 c_2 k_2^2 m_1^3 m_2 + 2 m_2^2 c_2^3 m_1 k_1 \]
\[ + m_2 c_1^2 k_2 m_1 + m_2^2 c_1^2 c_2 k_2 - m_2^3 c_2 c_1 k_2 m_1 + 2 m_2^3 c_2^2 c_1 k_2 \]
\[ - 4 c_2 m_2^2 k_2^2 m_1^2 + 2 m_2^3 c_2^2 c_1 k_1 - m_2^3 c_2 k_1 m_1 + c_2^3 m_1^3 m_2 \]
\[ - m_2^3 c_1 k_2^2 m_1 - 2 m_2^3 c_2 k_2^2 m_1 + 2 c_2^2 m_2^2 c_1 k_2 m_1 + m_2^2 c_2^2 c_1 k_1 m_1 k_1 \]
\[ + m_2^3 c_1^2 c_2 k_1 - 3 m_2^3 c_2 k_1 m_1 k_2 + c_2^3 c_3 k_1 + 3 m_2^2 c_2^3 k_2 m_1 \]
\[ + m_2^3 c_2^3 k_2) Z_1/(m_2^4 m_1^3) + \frac{f \dot{a} \dot{d} \dot{t}}{m_2^2} \]