Adaptive Nonlinear Control for Autonomous Ground Vehicles

William Spencer Black

Purdue University, wsblack1@gmail.com

Follow this and additional works at: http://docs.lib.purdue.edu/open_access_theses

Part of the Aerospace Engineering Commons, Mechanical Engineering Commons, and the Robotics Commons

Recommended Citation

This is to certify that the thesis/dissertation prepared

By William S. Black

Entitled
Adaptive Nonlinear Control for Autonomous Ground Vehicles

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:
Kartik B. Ariyur
Chair
Dengfeng Sun

Douglas E. Adams

To the best of my knowledge and as understood by the student in the Research Integrity and Copyright Disclaimer (Graduate School Form 20), this thesis/dissertation adheres to the provisions of Purdue University’s “Policy on Integrity in Research” and the use of copyrighted material.

Approved by Major Professor(s): Kartik B. Ariyur

Approved by: David C. Anderson 12/04/2013
Head of the Graduate Program Date
ADAPTIVE NONLINEAR CONTROL FOR AUTONOMOUS GROUND VEHICLES

A Thesis
Submitted to the Faculty
of
Purdue University
by
William S. Black

In Partial Fulfillment of the Requirements for the Degree
of
Master of Science in Mechanical Engineering

December 2013
Purdue University
West Lafayette, Indiana
I would like to dedicate this thesis to my family and friends for their support.
ACKNOWLEDGMENTS

My family has always been there for me and supported me in my pursuits. They understand that I need to explore things for myself, and helped me pursue my interests as much as possible. My friend and officemate Poorya Haghi is an invaluable resource and a true pleasure to be around. He is one of the most brilliant people I have ever met. You would be hard-pressed to find someone with even half of his talents as a mathematician. I really cannot thank him enough for all of the help I received from him while learning control theory. I would also like to thank Yann Delorme for joining me in crazy antics (e.g. mini-golf, pool, basketball, ’rhino-ball’) and always keeping things interesting. He was always able to make me laugh in any situation, and I enjoyed being around him. He has an exceptional mind and I know he will accomplish great things. Prof. Doug Adams is a truly inspirational person, always bringing an unparalleled sense of enthusiasm and passion. I hope that I can bring the same level of passion to everything I do for the rest of my life. Prof. Kartik Ariyur, my advisor and friend, has been instrumental in my academic development. He has always accepted who I am as a person, and enjoys when I challenge his beliefs. He always pushes me beyond my limits and gets me to think about problems from new perspectives. I have a tremendous amount of respect for him not only as an academic, but as a person as well. I want to especially thank him for convincing me to switch from fluid dynamics to systems, measurement, and control. I am a much happier person now that I can work on so many different interesting problems.
## TABLE OF CONTENTS

| LIST OF TABLES                          | vi       |
| LIST OF FIGURES                        | vii      |
| ABSTRACT                               | xii      |
| 1. INTRODUCTION                        | 1        |
| 1.1 Background on Modern Controls      | 1        |
| 1.2 Background and Motivation for Autonomy | 4        |
| 1.3 Challenges                         | 9        |
| 1.4 Trajectory Generation              | 13       |
| 2. SYSTEM MODELING AND DESIGN EXAMPLES | 15       |
| 2.1 Background on Modeling             | 15       |
| 2.2 Introduction to the Vehicle Design | 17       |
| 2.3 Rigid Body Dynamics                | 20       |
| 2.4 Kinematics and Attitude Propagation| 21       |
| 2.5 Navigation                         | 23       |
| 2.6 Stability Coordinate System Equations | 23      |
| 2.7 Drivetrain                          | 24       |
| 2.8 Active-Passive Suspension           | 25       |
| 2.9 Equations of Motion                | 28       |
| 3. NONLINEAR SYSTEMS AND CONTROL       | 32       |
| 3.1 Differential Equations on Differentiable Manifolds | 33 |
| 3.2 Observability                      | 35       |
| 3.3 Controllability                    | 37       |
| 3.4 Coordinate Transformations and Zero Dynamics | 38 |
| 3.5 Feedback Linearization             | 39       |
| 3.6 Backstepping                       | 40       |
| 3.7 Linear Parameterization            | 41       |
| 3.8 Filtered Derivatives               | 41       |
| 4. ADAPTIVE CONTROL METHODS            | 43       |
| 4.1 Model Reference Adaptive Control (MRAC) | 43     |
| 4.2 Adaptive Sliding Mode Control (ASMC)| 50     |
| 4.3 Extremum Seeking Model Reference Adaptive Control (ES-MRAC) | 57     |
| 5. ADAPTIVE CONTROL OF GROUND VEHICLE  | 65       |
5.1 Controllability ........................................ 65
5.2 Zero Dynamics ........................................... 66
5.3 Cascade Normal Form and Block Backstepping ........ 68
5.4 Model Reference Adaptive Control with Backstepping ... 69
5.5 Model Reference Adaptive Control with Feedback Linearization ... 78
5.6 Adaptive Sliding Mode Control with Backstepping .......... 84
5.7 Adaptive Sliding Mode Control with Feedback Linearization ... 92
5.8 Extremum Seeking Model Reference Adaptive Control .... 99

6. NONLINEAR DISTURBANCE OBSERVERS ................. 105
6.1 Single-Input-Single-Output Disturbance Observer Design .... 105
6.2 First Order Nonlinear Example .......................... 107
6.3 Multi-Input-Multi-Output Disturbance Observer Design (LMI) ... 111
6.4 Ground Vehicle Disturbance Observer ..................... 113

7. SUMMARY ................................................. 119

LIST OF REFERENCES ......................................... 120
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>60</td>
</tr>
</tbody>
</table>

1.1 Mineral composition and abrasiveness of regolith [12].
4.1 Parameters for model reference adaptive control.
4.2 Parameters for adaptive sliding mode control.
4.3 Parameters for ESMRAC.
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Feedback controller structure.</td>
<td>1</td>
</tr>
<tr>
<td>1.2 PID control structure.</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Robust control problem structure.</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Adaptive control structure.</td>
<td>3</td>
</tr>
<tr>
<td>1.5 Lunar landing site proposals [10].</td>
<td>6</td>
</tr>
<tr>
<td>1.6 Titanium distribution on the near and far sides (left &amp; right respectively) [11].</td>
<td>7</td>
</tr>
<tr>
<td>1.7 Composition of rock from lunar highlands [11].</td>
<td>7</td>
</tr>
<tr>
<td>1.8 Lunar rock distribution highlights [12].</td>
<td>8</td>
</tr>
<tr>
<td>1.9 Lunar topography [12].</td>
<td>11</td>
</tr>
<tr>
<td>1.10 Lunar gravitational field [12].</td>
<td>12</td>
</tr>
<tr>
<td>1.11 Lunar geoid [12].</td>
<td>12</td>
</tr>
<tr>
<td>1.12 Nodes and cells for Dijkstra’s algorithm.</td>
<td>13</td>
</tr>
<tr>
<td>2.1 Anscombe’s quartet [17].</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Skid steering ground vehicle.</td>
<td>18</td>
</tr>
<tr>
<td>2.3 Front steering ground vehicle.</td>
<td>19</td>
</tr>
<tr>
<td>2.4 Front and rear steering ground vehicle.</td>
<td>19</td>
</tr>
<tr>
<td>2.5 Euler angles [13].</td>
<td>21</td>
</tr>
<tr>
<td>2.6 DC motor, gears, and load of the drivetrain [22].</td>
<td>24</td>
</tr>
<tr>
<td>2.7 Active suspension system (quarter).</td>
<td>26</td>
</tr>
<tr>
<td>3.1 Tangent space of a sphere.</td>
<td>33</td>
</tr>
<tr>
<td>3.2 Tangent bundle of a circle.</td>
<td>34</td>
</tr>
<tr>
<td>3.3 Input output feedback linearization structure.</td>
<td>39</td>
</tr>
<tr>
<td>3.4 Frequency response for butterworth filters.</td>
<td>42</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.1</td>
<td>Model reference adaptive control structure.</td>
</tr>
<tr>
<td>4.2</td>
<td>MRAC step response.</td>
</tr>
<tr>
<td>4.3</td>
<td>MRAC error in step response.</td>
</tr>
<tr>
<td>4.4</td>
<td>MRAC control input for step.</td>
</tr>
<tr>
<td>4.5</td>
<td>MRAC parameter convergence for step.</td>
</tr>
<tr>
<td>4.6</td>
<td>MRAC pulse response.</td>
</tr>
<tr>
<td>4.7</td>
<td>MRAC error in pulse response.</td>
</tr>
<tr>
<td>4.8</td>
<td>MRAC control input for pulse.</td>
</tr>
<tr>
<td>4.9</td>
<td>MRAC parameter convergence for pulse.</td>
</tr>
<tr>
<td>4.10</td>
<td>Adaptive sliding mode control structure.</td>
</tr>
<tr>
<td>4.11</td>
<td>Sliding surface phase portrait.</td>
</tr>
<tr>
<td>4.12</td>
<td>ASMC step response.</td>
</tr>
<tr>
<td>4.13</td>
<td>ASMC error in step response.</td>
</tr>
<tr>
<td>4.14</td>
<td>ASMC control input for step.</td>
</tr>
<tr>
<td>4.15</td>
<td>ASMC parameter convergence for step.</td>
</tr>
<tr>
<td>4.16</td>
<td>ASMC pulse response.</td>
</tr>
<tr>
<td>4.17</td>
<td>ASMC error in pulse response.</td>
</tr>
<tr>
<td>4.18</td>
<td>ASMC control input for pulse.</td>
</tr>
<tr>
<td>4.19</td>
<td>ASMC parameter convergence for pulse.</td>
</tr>
<tr>
<td>4.20</td>
<td>Extremum-seeking model reference adaptive control structure [33].</td>
</tr>
<tr>
<td>4.21</td>
<td>Adaptation in extremum seeking model reference adaptive control [33].</td>
</tr>
<tr>
<td>4.22</td>
<td>ESMRAC step response.</td>
</tr>
<tr>
<td>4.23</td>
<td>ESMRAC control input for step.</td>
</tr>
<tr>
<td>4.24</td>
<td>ESMRAC error in step response.</td>
</tr>
<tr>
<td>4.25</td>
<td>ESMRAC parameter convergence for step.</td>
</tr>
<tr>
<td>4.26</td>
<td>ESMRAC pulse response.</td>
</tr>
<tr>
<td>4.27</td>
<td>ESMRAC control input for pulse.</td>
</tr>
<tr>
<td>4.28</td>
<td>ESMRAC error in pulse response.</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>4.29 ESMRAC parameter convergence for pulse.</td>
<td>64</td>
</tr>
<tr>
<td>5.1 BSMRAC circle path response.</td>
<td>72</td>
</tr>
<tr>
<td>5.2 BSMRAC control input for circle path.</td>
<td>72</td>
</tr>
<tr>
<td>5.3 BSMRAC trajectory errors for circle path.</td>
<td>73</td>
</tr>
<tr>
<td>5.4 BSMRAC velocity errors for circle path.</td>
<td>73</td>
</tr>
<tr>
<td>5.5 BSMRAC parameter convergence for circle path.</td>
<td>74</td>
</tr>
<tr>
<td>5.6 BSMRAC $p_4$ adaptation for circle path.</td>
<td>74</td>
</tr>
<tr>
<td>5.7 BSMRAC looped path response.</td>
<td>75</td>
</tr>
<tr>
<td>5.8 BSMRAC control input for looped path.</td>
<td>75</td>
</tr>
<tr>
<td>5.9 BSMRAC trajectory errors for looped path.</td>
<td>76</td>
</tr>
<tr>
<td>5.10 BSMRAC velocity errors for looped path.</td>
<td>76</td>
</tr>
<tr>
<td>5.11 BSMRAC parameter convergence for looped path.</td>
<td>77</td>
</tr>
<tr>
<td>5.12 BSMRAC $p_4$ parameter adaptation for looped path.</td>
<td>77</td>
</tr>
<tr>
<td>5.13 FLMRAC circle path response.</td>
<td>80</td>
</tr>
<tr>
<td>5.14 FLMRAC control inputs for circle path.</td>
<td>81</td>
</tr>
<tr>
<td>5.15 FLMRAC errors for circle path.</td>
<td>81</td>
</tr>
<tr>
<td>5.16 FLMRAC parameter convergence for circle path.</td>
<td>82</td>
</tr>
<tr>
<td>5.17 FLMRAC looped path response.</td>
<td>82</td>
</tr>
<tr>
<td>5.18 FLMRAC control inputs for looped path.</td>
<td>83</td>
</tr>
<tr>
<td>5.19 FLMRAC errors for looped path.</td>
<td>83</td>
</tr>
<tr>
<td>5.20 FLMRAC parameter convergence for looped path.</td>
<td>84</td>
</tr>
<tr>
<td>5.21 BSASMC circle path response.</td>
<td>86</td>
</tr>
<tr>
<td>5.22 BSASMC control inputs for circle path.</td>
<td>87</td>
</tr>
<tr>
<td>5.23 BSASMC trajectory errors for circle path.</td>
<td>87</td>
</tr>
<tr>
<td>5.24 BSASMC velocity errors for circle path.</td>
<td>88</td>
</tr>
<tr>
<td>5.25 BSASMC parameter convergence for circle path.</td>
<td>88</td>
</tr>
<tr>
<td>5.26 BSASMC $p_4$ adaptation for circle path.</td>
<td>89</td>
</tr>
<tr>
<td>5.27 BSASMC looped path response.</td>
<td>89</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>5.28 BSASMC control inputs for looped path.</td>
<td>90</td>
</tr>
<tr>
<td>5.29 BSASMC trajectory errors for looped path.</td>
<td>90</td>
</tr>
<tr>
<td>5.30 BSASMC velocity errors for looped path.</td>
<td>91</td>
</tr>
<tr>
<td>5.31 BSASMC parameter convergence for looped path.</td>
<td>91</td>
</tr>
<tr>
<td>5.32 BSASMC $p_4$ adaptation for looped path.</td>
<td>92</td>
</tr>
<tr>
<td>5.33 FLASMC circle path response.</td>
<td>95</td>
</tr>
<tr>
<td>5.34 FLASMC control inputs for circle path.</td>
<td>95</td>
</tr>
<tr>
<td>5.35 FLASMC errors for circle path.</td>
<td>96</td>
</tr>
<tr>
<td>5.36 FLASMC parameter convergence for circle path.</td>
<td>96</td>
</tr>
<tr>
<td>5.37 FLASMC looped path response.</td>
<td>97</td>
</tr>
<tr>
<td>5.38 FLASMC control inputs for looped path.</td>
<td>97</td>
</tr>
<tr>
<td>5.39 FLASMC errors for looped path.</td>
<td>98</td>
</tr>
<tr>
<td>5.40 FLASMC parameter convergence for looped path.</td>
<td>98</td>
</tr>
<tr>
<td>5.41 BSESMRAC circle path response.</td>
<td>100</td>
</tr>
<tr>
<td>5.42 BSESMRAC control inputs for circle path.</td>
<td>101</td>
</tr>
<tr>
<td>5.43 BSESMRAC trajectory errors for circle path.</td>
<td>101</td>
</tr>
<tr>
<td>5.44 BSESMRAC velocity errors for circle path.</td>
<td>102</td>
</tr>
<tr>
<td>5.45 BSESMRAC parameter convergence for circle path.</td>
<td>102</td>
</tr>
<tr>
<td>5.46 FLESMRAC circle path response.</td>
<td>103</td>
</tr>
<tr>
<td>5.47 FLESMRAC control inputs for circle path.</td>
<td>103</td>
</tr>
<tr>
<td>5.48 FLESMRAC errors for circle path.</td>
<td>104</td>
</tr>
<tr>
<td>5.49 FLESMRAC parameter convergence for circle path.</td>
<td>104</td>
</tr>
<tr>
<td>6.1 Response with disturbance.</td>
<td>109</td>
</tr>
<tr>
<td>6.2 Observer tracking error.</td>
<td>110</td>
</tr>
<tr>
<td>6.3 Observer estimate of disturbance.</td>
<td>110</td>
</tr>
<tr>
<td>6.4 BSMRC response with disturbances.</td>
<td>116</td>
</tr>
<tr>
<td>6.5 Observer error for disturbance in $u$.</td>
<td>116</td>
</tr>
<tr>
<td>6.6 Observer error for disturbance in $\omega$.</td>
<td>117</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>6.7</td>
<td>Observer estimate of disturbance in $u$.</td>
</tr>
<tr>
<td>6.8</td>
<td>Observer estimate of disturbance in $\omega$.</td>
</tr>
</tbody>
</table>
ABSTRACT

Black, William S. M.S.M.E., Purdue University, December 2013. Adaptive Nonlinear Control for Autonomous Ground Vehicles. Major Professor: Kartik B. Ariyur, School of Mechanical Engineering.

We present the background and motivation for ground vehicle autonomy, and focus on uses for space-exploration. Using a simple design example of an autonomous ground vehicle we derive the equations of motion. After providing the mathematical background for nonlinear systems and control we present two common methods for exactly linearizing nonlinear systems, feedback linearization and backstepping. We use these in combination with three adaptive control methods: model reference adaptive control, adaptive sliding mode control, and extremum-seeking model reference adaptive control. We show the performances of each combination through several simulation results. We then consider disturbances in the system, and design nonlinear disturbance observers for both single-input-single-output and multi-input-multi-output systems. Finally, we show the performance of these observers with simulation results.
1. INTRODUCTION

1.1 Background on Modern Controls

In order to achieve autonomy in a ground vehicle, we must talk about feedback control. Fig. 1.1 shows a common representation of a simple feedback control loop.

![Feedback controller structure](image)

Figure 1.1. Feedback controller structure.

We start by giving a reference command signal to the controller, which sends a control signal to the system plant. The system plant’s performance is measured by sensors in a feedback loop, which is then given back to the controller. In the iterative process, the controller compares the feedback and reference signals, trying to minimize the error between them. Design methods such as PID (Proportional-Integral-Derivative) control, shown in Fig. 1.2 and lead-lag compensation fit into the generalized feedback control theory, but we have not discussed large disturbances and uncertainties. To handle various types of disturbances and uncertainties, modern control theories beyond basic feedback control are placed in two categories: robust and adaptive.
Robust control is generally used when significant model uncertainties and large disturbances are expected. Several common methods of robust control are $H_\infty$, $H_2$, and variable structure methods such as sliding mode control. Linear systems are often modified by linear fractional transformations (LFTs) to give a standard interconnection structure associated with robust control problems as seen in Fig. 1.3.
In Fig. 1.3, \( d \) represents various external signals (i.e. commands, disturbances, noise), \( e \) represents tracking errors, \( y \) contains the available signals for control, with \( u \) as the control inputs, and \( v, z \) are signals associated with uncertainty and perturbations. The end structure is

\[
\begin{bmatrix}
  z \\
  e \\
  y
\end{bmatrix} = P
\begin{bmatrix}
  v \\
  d \\
  u
\end{bmatrix},
\] (1.1)

where \( P = C(sI - A)^{-1}B + D \).

Adaptive control is generally used when significant parametric uncertainties are present, but expected disturbances and model uncertainties are small. Adaptive control uses a negative gradient algorithm based on system performance error to modify parameter estimates. We show the basic structure for adaptive control in Fig. 1.4.

![Figure 1.4. Adaptive control structure.](image)
Robust-Adaptive [1] and Adaptive-Robust [2] control techniques (they are different) have also become available in the last few decades, with Robust-Adaptive control adding robustness to adaptive control, and Adaptive-Robust control adding adaptation to robust control techniques. While we will eventually use one ‘Robust-Adaptive’ technique, the theoretical pursuits of these methods is beyond the scope of the current work as we want to focus only on adaptive control.

1.2 Background and Motivation for Autonomy

Autonomous vehicles have been a research topic for quite some time, with autopilot controllers on aircraft being an early low level example. The focus has increased in the last 20 to 30 years from the significant advances in semiconductors and computer technology, allowing for real time implementation of more sophisticated and computationally intensive algorithms. Unmanned aerial vehicles have successfully completed missions in the civilian and military sectors, receiving heavy media focus in the Iraq and Afghanistan wars. Google’s 'Driver-less Car' has also received a lot of attention recently after several states passed laws allowing autonomous vehicles to be tested and used on public roads. Autonomous vehicles provide much greater safety in the air and on the road, as well as efficiency in traffic flows and fuel economy. However, the industry that relies on autonomy the most (for now) continues to be the space industry.

The last decade has seen a revolutionary growth in robotic space exploration. The United States and Russia are no longer the only countries with successful space programs – it is now a global endeavor. In the last ten years we have seen seven remote-sensing satellites launch to gather further information about the Moon: Europe’s SMART-1 (2004), Japan’s Kaguya (2007), China’s Chang’e-1 (2007) and Chang’e-2 (2010), India’s Chandrayaan-1 (2008), and the LRO (2009) [3], the LCROSS (2009) [4] and GRAIL (2012) [5] of the United States. The LRO-LCROSS orbiters in 2009 helped NASA discover large amounts of water in several craters of the moon after de-
liberately sending the LCROSS to impact inside one of them [6]. The evidence came after a prediction in 1998 that the moon contained water [7]. This is important due to the fact that water is essential to life. The presence of water on the moon allows us to consider building permanent stations where water can be purified for drinking as well as split into hydrogen and oxygen for other purposes. Astronauts would be able to stay on the moon for extended periods to perform more research and set up mining operations for precious metals. The mining of metals from lunar materials is an important goal for NASA as well as other countries. Oxygen production also allows for very significant cost savings in fuel oxidant for rockets or even the ground vehicles mining. The structure of the lunar interior also provides fundamental information on the evolution of differentiated planetary bodies. Since the Moon lacks plate tectonics, its crust and mantle have remained relatively intact and isolated since its creation. The interior should contain evidence of early planetary processes that other planets have since lost.

These new discoveries are not only limited to the moon, they also exists on Mars. NASA released confirmation in 2009 that large amounts of methane had been discovered in the Martian atmosphere which may indicate life [8], [9]. Small amounts of methane would be destroyed quickly in the atmosphere of Mars, but scientists are detecting fairly large amounts which they believe indicate that the planet is still somewhat alive in a geological sense. Geological processes such as the oxidation of iron release methane, but so do certain types of bacteria and biological processes [8], [9]. We need more robots that can look at more than just the surface, and analyze what is really in the soil. This comes at an important time when space exploration is facing massive budget cuts, and its future is uncertain. Most people don’t realize the long term benefits that these programs provide. Many modern conveniences came directly from the Apollo missions, which we are still receiving economic benefits from even after more than 40 years.
These programs that push the outer envelope always have a very high return, and we need to rekindle public interest in fundamental research again. In response to NASA’s announcement to mine for various elements off planet, scientists and engineers have already started planning mission trajectories and landing sites as seen in Fig. 1.5.

![Figure 1.5. Lunar landing site proposals [10].](image)

We mentioned earlier that we are interested in mining precious metals along with other elements and we’ve shown proposed landing sites, but we haven’t discussed why we want to go to these specific locations. It turns out that the elements we are most interested in are highly concentrated in specific areas. The red, yellow, and blue coloring in Fig 1.6 refer to 'high', 'low', and 'negligible' Titanium concentrations in the regolith.
Figure 1.6. Titanium distribution on the near and far sides (left & right respectively) [11].

We can see that there is a significant concentration of Titanium on the near side of the moon, and we show samples of the concentrations obtained from various Apollo missions in Fig. 1.7.

Figure 1.7. Composition of rock from lunar highlands [11].
We may also note that there are also significant amounts of Silicon and Aluminum in the regolith. It is clear that the lunar regolith has high oxygen content as we mentioned before. Fig. 1.8 on page 8 shows a surface map of Iron-Oxide concentration levels by weight percent, as well as trace amount of Thorium which may be used for nuclear power plants.

![Figure 1.8. Lunar rock distribution highlights [12].](image)

Now that we know where we are landing, and why we are landing there, we need to discuss how exactly we are going to extract the elements we are interested in. There are numerous chemical reactions and processes that may be applied to regolith to extract compounds like water, titanium oxide, methane, etc. We list several of these reactions below [11]:

- Reaction A
- Reaction B
- Reaction C
- Reaction D
- Reaction E
• Reduction with Hydrogen, Methane, or Sulfuric acid

• Electrolysis of solid or molten regolith

• Vapor phase pyrolysis

Each reaction has its own advantages and disadvantages. Some of these processes require extremely high temperatures which may not be feasible, or they may have too many steps to provide a high enough yield in the end. The process or processes used should be chosen on a case by case basis and tailored to fit specific mission profiles.

1.3 Challenges

Unlike vehicles bound to terrafirma, minimizing vehicle weight is a top priority for space missions. The relationship of payload to rocket fuel weight is nonlinear. Not only must the rocket have enough fuel for the mission payload weight, but also has to have enough fuel for the added weight of the fuel used for the mission payload weight. This unfortunate relationship is shown by the Tsiolkovsky rocket equation [13],

$$\Delta v = v_e \ln \frac{m_0}{m_1}. \quad (1.2)$$

While this equation alone cannot be used to accurately calculate propellant masses required for an entire mission, it is useful for calculating the propellant required for short maneuvers which still shows the importance of payload weight.

Lunar regolith is known as being extremely abrasive, along with having micron sized dust particles. On average the minerals in lunar regolith have a Mohs rating of about 6, but some other minerals of significant contribution have Mohs ratings of 7 and 8, which are considered extremely abrasive. Table 1.1 shows some of the common minerals found in lunar regolith along with their Mohs ratings and abundance.

By comparison, we often build the structures for machines out of materials like aluminum and titanium which have Mohs ratings of 2.75 and 6.0 respectively. Abrasion occurs between materials of different Mohs ratings. The larger the difference, the
Table 1.1. Mineral composition and abrasiveness of regolith [12].

<table>
<thead>
<tr>
<th>Mineral</th>
<th>Mohs</th>
<th>Amount</th>
<th>Chemical Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anorthite</td>
<td>6</td>
<td>Abundant</td>
<td>(CaAl_2Si_2O_8)</td>
</tr>
<tr>
<td>Bytownite</td>
<td>6.0-6.5</td>
<td>Major</td>
<td>((CaNa)(SiAl)_4O_8)</td>
</tr>
<tr>
<td>Labradorite</td>
<td>7</td>
<td>Major</td>
<td>((CaNa)(SiAl)_4O_8)</td>
</tr>
<tr>
<td>Olivine</td>
<td>6.5-7.0</td>
<td>Major</td>
<td>((MgFe)_2SiO_4)</td>
</tr>
<tr>
<td>Fayalite</td>
<td>6.5-7.0</td>
<td>-</td>
<td>(Fe_2SiO_4)</td>
</tr>
<tr>
<td>Forsterite</td>
<td>6.5-7.0</td>
<td>-</td>
<td>(Mg_2SiO_4)</td>
</tr>
<tr>
<td>Clinoenstatite</td>
<td>5.0-6.0</td>
<td>Major</td>
<td>(Mg_2[Si_2O_6])</td>
</tr>
<tr>
<td>Pigeonite</td>
<td>6</td>
<td>Major</td>
<td>((Mg, Fe^{+2}, Ca)_2[Si_2O_6])</td>
</tr>
<tr>
<td>Hedenbergite</td>
<td>6</td>
<td>Major</td>
<td>(CaFe^{+2}[Si_2O_6])</td>
</tr>
<tr>
<td>Augite</td>
<td>5.5-6.0</td>
<td>Major</td>
<td>((Ca, Na)(Mg, Fe, Al, Ti)[(Si, Al)_2O_6])</td>
</tr>
<tr>
<td>Enstatite</td>
<td>5.0-6.0</td>
<td>Abundant</td>
<td>(Mg_2[Si_2O_6])</td>
</tr>
<tr>
<td>Spinel</td>
<td>7.5-8.0</td>
<td>Minor</td>
<td>(MgAl_2O_4)</td>
</tr>
<tr>
<td>Hercynite</td>
<td>7.5-8.0</td>
<td>Minor</td>
<td>(Fe^{+2}Al_2O_4)</td>
</tr>
<tr>
<td>Ulvospinel</td>
<td>5.5-6.0</td>
<td>Minor</td>
<td>(TiFe^{+2}_2O_4)</td>
</tr>
<tr>
<td>Chromite</td>
<td>5.5</td>
<td>Minor</td>
<td>(Fe^{+2}Cr_2O_4)</td>
</tr>
<tr>
<td>Troilite</td>
<td>4</td>
<td>Trace</td>
<td>(FeS)</td>
</tr>
<tr>
<td>Whilockite</td>
<td>5</td>
<td>Trace</td>
<td>(Ca_9(Mg, Fe^{+2})(PO_4)_6(PO_3OH))</td>
</tr>
<tr>
<td>Apatite</td>
<td>5</td>
<td>Trace</td>
<td>(Ca_5(PO_4)_3(OH, F, Cl))</td>
</tr>
<tr>
<td>Ilmenite</td>
<td>5.5</td>
<td>Minor</td>
<td>(Fe^{+2}TiO_3)</td>
</tr>
<tr>
<td>Native Iron</td>
<td>4.5</td>
<td>Trace</td>
<td>(Fe)</td>
</tr>
</tbody>
</table>

more abrasion, and the material with the lower Mohs rating incurs the damage. The difference between Mohs ratings of our build materials and the lunar environment just go to show how severe the lunar environment is and how we cannot expect to have any one machine operating for extended periods of time (without having a massive budget). The Mohs rating is very difficult to standardize and apply when considering
alloys, but regardless of the improvements gained by using alloys the lunar regolith remains extremely abrasive. We also try to seal our machines from dust particles by using various types of polymers, but these are even more susceptible to abrasion, not to mention the difficulty of sealing off the system from micron sized particles.

The only property of lunar regolith we can use to our advantage is the fact that the dust carries a static charge [14]. Since the moon has no atmosphere, the solar wind (ionized particles) bombard the dust and actually knock off electrons, which in turn ionizes the dust. This means that in order to ‘dust-proof’ our vehicle, we can utilize weak electric fields to protect our vehicle from this destructive dust.

One of the most important problems of extra-terrestrial missions is navigation. On Earth we typically use GPS for accurate position and velocity measurements, and during blackout periods we rely on inertial navigation. However, we obviously do not have GPS satellites on the moon, which clearly makes GPS impossible. The problem with inertial navigation is that even with extremely expensive military-grade equipment, a few minutes of blackout navigation could put our position off by several meters. In the case of inexpensive equipment, we could be off by kilometers within a minute or less. We show the Moon’s topography in Fig. 1.9, which is quite interesting in itself because of the non-uniformity, but also shows how quickly the terrain may change as we drive across.

Figure 1.9. Lunar topography [12].
We might now consider using some sort of magnetometer, magnetic compass, or fluxgate compass for navigation. The geological activity in the cores of planets causes molten iron alloys to move about, which creates the planetary magnetic field. Unfortunately the Moon no longer has any strong geological activity in its core, leading to an extremely weak magnetic field. The weak field that does exist does not have any significant differentiation, also making it quite useless for navigation. We show the lunar gravitational field and geoid in Fig. 1.10 and Fig. 1.11.

Figure 1.10. Lunar gravitational field [12].

Figure 1.11. Lunar geoid [12].
For global navigation, the only method we have left is the gyrocompass or fibre-optic gyrocompass. The gyrocompass uses gyroscopic precession to detect ‘true north’ on any planetary body. While fast changes in speed and direction can introduce errors into a mechanical gyrocompass, requiring time for it to adjust, the fibre-optic gyrocompass has much better performance because it relies on the Sagnac effect [15] rather than fluid viscosities, etc. For local navigation and obstacle avoidance, cameras and/or RADAR/LIDAR systems may be used to detect obstacles and allow the control system to take the necessary countermeasures.

1.4 Trajectory Generation

To accomplish trajectory following control, which is a basic component of autonomy, we need to generate a trajectory for our system to follow. There are many ways to do this and three of the most common methods used today are the A* search algorithm, simultaneous localization and mapping algorithm (SLAM), and potential field path planning (a.k.a ’Laplacian path planning’). SLAM and A* are variants of Dijkstra’s algorithm (application of Voronoi diagrams), which is a graph search algorithm that analyzes various nodes and the distances between them to find the shortest path.

![Figure 1.12. Nodes and cells for Dijkstra’s algorithm.](image-url)
The robot is typically preprogrammed with a grid representing the domain or configuration space in which it operates, and given coordinates that represent its desired location. It uses various sensors (video, radar, etc.) to detect objects (called 'landmarks') and track them. The robot continues to populate its internal map of the domain, and update its own position by triangulation based on the perceived movement of landmarks.

Potential field planning is also a popular method. We use Laplace’s equation

$$\nabla^2 \phi = \sum_{i=1}^{n} \frac{\partial^2 \phi}{\partial x_i^2} = 0,$$

(1.3)

for static maps; and the wave equation

$$\frac{\partial^2 \phi}{\partial t^2} - \nabla^2 \phi = 0,$$

(1.4)

for dynamic maps. The idea here is that we consider the robot to be a particle (single rigid body) of some size, moving in a flow field around some obstacles. The region of operation is discretized using as much a priori information as possible. Most likely the size of the domain will be known (i.e. room, city, planet, etc.) but obstacles within that domain may not be. The robot will first localize itself in the domain, and then attempt to reach some desired coordinate. Before we start gathering sensor data the path planning algorithm would look like a very trivial case of source to sink flow, and the robot will start out on the optimal path using a gradient descent method. Once an obstacle is detected by the sensors (i.e. cameras, radar, lidar), a steep positive gradient is assigned to it (proportional to its size) such that the robot will avoid it and choose a new path. This produces a smooth potential field without any local minima, ensuring that the robot will continue smoothly to its goal.
2. SYSTEM MODELING AND DESIGN EXAMPLES

2.1 Background on Modeling

In general there are two types of models: ‘grey box’ and ‘black box’. Black box modeling indicates that hardly anything is known about the system except perhaps the output signal given some input. We have no idea what is inside the black box, but we may be able to figure it out based on the input/output relation. Grey box modeling is a situation where some a-priori information about the system is known and can be used. For example, knowing the design of the system and being able to use Newton’s laws of motions to understand the dominating dynamics would be considered grey box modeling. Even though we may know a lot about the system, there is no way we can know everything about it, hence the ‘grey’ designation. Uncertainty in our models and parameters are fundamental problems that we must unfortunately deal with in the real world.

When we model systems we always make some fundamental assumptions about the system itself. For physical systems we typically make the following assumptions [16]:

- Time is one-dimensional
- Space is three-dimensional and Euclidean
- The initial state of a system (the totality of positions and velocities of its points at some moment of time) uniquely determines all of its motion.
We also make assumptions about the coordinate systems we operate in. We assume there exist coordinate systems called 'inertial', possessing the following properties [16]:

- All the laws of nature at all moments of time are the same in all inertial coordinate systems
- All coordinate systems in uniform rectilinear motion with respect to an inertial one are themselves inertial

We almost always use physics laws (e.g. Newton, Euler, Kirchhoff, etc.) to model the system that we have designed, but as mentioned earlier there is always uncertainty in the system. This often leads to the field of machine learning, the process of machines 'learning' from data. This boils down to approximating a relationship (function) between data points, and then hopefully achieving correct predictions based on its function approximation if it is to be used in control. There are many methods associated with machine learning (e.g. Neural Networks, Bayesian Networks, etc.), but we should be very careful in how we implement these algorithms. There are situations where using machine learning algorithms provide a significant advantage to the user, and situations where it would be disadvantageous.

An example of a system where machine learning would be advantageous is a highly articulated robot such as a humanoid robot. A system of this type has too many states to handle in practice, not to mention complexities due to nonlinearities and coupling, so a machine learning algorithm would be very advantageous here. However, since the algorithm can only improve the function approximation by analyzing data, this approach obviously requires a lot of 'learning' time as well as computational resources. Therefore the system (or its data) has to be available to be tested in a controlled environment which isn’t always possible (consider a hypersonic vehicle). The other disadvantage to certain types of machine learning algorithms is known as Anscombe’s Quartet, shown in Fig. 2.1.
Anscombe’s Quartet was a famous data set given by Francis Anscombe to show the limitations of regression analysis without graphing the data. Each of the significantly different data sets has the exact same mean, variance, correlation, and linear regression line. This shows that there will always be outliers that manipulate the system, but also that we should attempt to have a universal function identifier if we cannot narrow down the distributions of data we will see. There are machine learning algorithms that are universal function identifiers (see the Cybenko Theorem), but these methods will always be susceptible to outliers as shown, which further illustrates the importance of the ‘learning’ period as well as using as much a-priori information as possible when it is available.

2.2 Introduction to the Vehicle Design

We first consider three separate vehicle steering configurations that cover most ground vehicle designs [18], [19], [20]: skid steering (Fig. 2.2), front wheel steering (Fig. 2.3), and front and rear wheel steering (Fig. 2.4). Skid or differential steering...
works by varying the rotational speed between the wheels on each side of the vehicle. If the wheels on the right side of the vehicle are rotating faster than the left, the vehicle will turn to the right, and vice versa. The greater the difference in rotation, the faster the vehicle will turn. Skid steering also allows zero point turns, which entails one set of wheels rotating clockwise while the others rotate counterclockwise.

Vehicles with front wheel steering move exactly like cars. The back wheels on each side rotate at the same speed, and the front wheels are controlled by a steering mechanism. A vehicle with front and rear wheel steering have much more robust control capability, and can perform near-zero point turns.
Figure 2.3. Front steering ground vehicle.

Figure 2.4. Front and rear steering ground vehicle.
2.3 Rigid Body Dynamics

We start by considering the velocity and acceleration equations for a rigid body

\[ V_B = \frac{dr}{dt} + \omega_B \times r, \quad (2.1) \]

and

\[ A_B = \frac{dV_B}{dt} + \omega_B \times V_B. \quad (2.2) \]

A more common representation for the acceleration equation of a rigid body is shown below [21],

\[ \dot{V}_B = \frac{1}{m} F_B - (\omega_B + B \omega_M) \times V_B + B [g - \omega_M \times (\omega_M \times p)]. \quad (2.3) \]

Since the moon has negligible rotation due to its tidal locking orbit we may simplify to

\[ \dot{V}_b = \frac{1}{m} F - \omega \times V_b. \quad (2.4) \]

We consider a rigid body that is free to rotate about each axis (for now), and these angular accelerations are found by using Euler’s equation of motion

\[ I \ddot{\omega} = -\omega \times I \omega + \tau, \quad (2.5) \]

where \( I \) is the inertia tensor, \( \omega \) is the rotational vector in body coordinates, and \( \tau \) is the torque on the body.
2.4 Kinematics and Attitude Propagation

A point on a rigid body can be defined in terms of body-fixed axes $x, y, z$. To determine the orientation of the body itself, we introduce Euler’s angles $\psi, \phi, \theta$ which are three independent quantities capable of defining the position of the $x, y, z$ body axes relative to the inertial $X, Y, Z$ axes as in Figure 2.5. Euler angles [13].

$$
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} =
\begin{bmatrix}
1 & \tan(\theta) \sin(\phi) & \tan(\theta) \cos(\phi) \\
0 & \cos(\phi) & -\sin(\phi) \\
0 & \sin(\phi) / \cos(\theta) & \cos(\phi) / \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix}.
$$

(2.6)

The Euler angle equation has several disadvantages associated with it. When we consider the bottom row of the rotation matrix, we see that we are dividing the last two terms by $\cos \theta$. This means that if we have a pitch angle of plus/minus 90, we will have a division by zero which results in large numerical errors as well as a loss in the degrees of freedom. Even if the pitching angle is close to 90 or 270, the matrix will become ill-conditioned. A third problem with this calculation is that the integration for the angles can lead to values outside ‘normal’ ranges, called ‘wrap-around’ [21]. The last problem is that the equation is linear in $\omega$, but nonlinear in the Euler angles. We can use quaternions to fix these problems.
Quaternions are a product of applying Euler’s rotation theorem to the rotation of a rigid body about a point. The scalar rotation angle along with a unit vector defining the axis of rotation are plugged into Euler’s formula to give the quaternion. Multiple rotations may also be considered, but require matrix operations. The rotation equations using quaternions are

\[
\begin{bmatrix}
\dot{q}_1 \\
\dot{q}_2 \\
\dot{q}_3 \\
\dot{q}_4 
\end{bmatrix} =
\begin{bmatrix}
0 & -r & -q & -p \\
r & 0 & -p & q \\
qu & pq & 0 & -r \\
p & -q & r & 0
\end{bmatrix}
\begin{bmatrix}
q_1 \\
q_2 \\
q_3 \\
q_4
\end{bmatrix},
\]

(2.7)

where each quaternion is defined as:

\[
q_1 = \cos\left(\frac{\psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right), 
\]

(2.8)

\[
q_2 = \sin\left(\frac{\psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) - \cos\left(\frac{\psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right), 
\]

(2.9)

\[
q_3 = \cos\left(\frac{\psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right), 
\]

(2.10)

and finally

\[
q_4 = \cos\left(\frac{\psi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right) - \sin\left(\frac{\psi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right).
\]

(2.11)

The quaternion form removes the previous complications with Euler angle calculations, but now we have a new problem in that we have an attitude propagation relation that is non-minimal. Non-minimal system representations can introduce non-physical dynamics. Fortunately ground vehicles do not experience the gimbal lock problem as they do not pitch at large angles.
2.5 Navigation

The navigation equations are found by multiplying the velocity vectors by a rotation matrix that transforms between the body and global coordinates [21] to get:

\[ \dot{x}_b = u \cos(\theta) \cos(\psi) + v(- \cos(\phi) \sin(\psi) + \sin(\phi) \sin(\theta) \cos(\psi)) \\
+ w(\sin(\phi) \sin(\psi) + \cos(\phi) \sin(\theta) \cos(\psi)), \quad (2.12) \]

\[ \dot{y}_b = u \cos(\theta) \sin(\psi) + v(\cos(\phi) \cos(\psi) + \sin(\phi) \sin(\theta) \sin(\psi)) \\
+ w(- \sin(\phi) \cos(\psi) + \cos(\phi) \sin(\theta) \sin(\psi)), \quad (2.13) \]

and lastly,

\[ \dot{z}_b = u \sin(\theta) - v \sin(\phi) \cos(\theta) - w \cos(\phi) \cos(\theta). \quad (2.14) \]

These simplify significantly when we make assumptions such as planar motion \((w = 0, \theta = 0, \phi = 0, \dot{z}_b = 0)\) which result in:

\[ \dot{x} = u \cos(\psi) - v \sin(\psi), \quad (2.15) \]

and

\[ \dot{y} = u \sin(\psi) + v \cos(\psi). \quad (2.16) \]

2.6 Stability Coordinate System Equations

For systems with steering mechanisms or vehicles where the velocity is not always collinear with a specific body axis, it is useful to convert the equations of motion into relations of speed, angle of attack, and sideslip angles:

\[ \dot{V} = \frac{1}{m} \left[ \sum F_x \cos(\alpha) \cos(\beta) + \sum F_y \sin(\beta) + \sum F_z \sin(\alpha) \cos(\beta) \right], \quad (2.17) \]

\[ \dot{\alpha} = \frac{1}{mV_r \cos(\beta)} \left[ \sum F_z \cos(\alpha) - \sum F_x \sin(\alpha) \right] + q - p \cos(\alpha) \tan(\beta) \\
+ r \sin(\alpha) \tan(\beta), \quad (2.18) \]
and
\[
\dot{\beta} = \frac{1}{mV_r} \left[ - \sum F_x \cos(\alpha) \sin(\beta) + \sum F_y \cos(\beta) - \sum F_z \sin(\alpha) \sin(\beta) \right] \\
- p \sin(\alpha) - r \cos(\alpha).
\] (2.19)

Systems using these equations in their representation are sometimes said to be in a ’stability coordinate system’. They give relations that rely more on values that may be directly measured, requiring fewer observers or indirect measurements, which is very useful for many systems. Examples of sensors used in these equations would be accelerometers, rate-gyros, encoders and/or cameras, and level sensors.

2.7 Drivetrain

Using DC motors in the drivetrain can satisfy a number of autonomous vehicle applications, so we will consider the simple DC motor drivetrain model is shown in Fig. 2.6.

![Diagram of DC motor, gears, and load of the drivetrain](image)

We first solve for the rotational equations on both the load as well as the motor itself resulting in:

\[
J_L \ddot{\omega} + B_L \dot{\omega} = \tau_L - F R_t,
\] (2.20)

and

\[
J_A \ddot{\Omega} + B_A \dot{\Omega} = \tau_m - \frac{N_1}{N_2} \tau_L.
\] (2.21)
Then we combine the equations for the motor armature/driveshaft, gear box, and wheel to get

$$\left( J_L + J_A \left( \frac{N_2}{N_1} \right)^2 \right) \ddot{\omega}_i + \left( B_L + B_A \left( \frac{N_2}{N_1} \right)^2 \right) \omega_i = \frac{N_2}{N_1} K_T i_A - F_i R_i. \quad (2.22)$$

For now we will assume that we are dealing with smaller robots or vehicles that can use DC motors for driving the wheels, but $\tau_m$ could be a function of something other than an electrical circuit’s state. To find $\tau_m$ we use Kirchhoff’s voltage law to sum the voltages around the circuit and obtain

$$E_{emf} = E_i - R_A i_A - L_A \frac{di_A}{dt}. \quad (2.23)$$

Since it is a DC motor, we can neglect the $\frac{di}{dt}$ term, and use the voltage rating of the motor for $E_i$. Then plugging in the motor voltage constant relation $E_{emf} = K_b \Omega$ we can solve for the current

$$i = \frac{V - K_b \Omega}{R_A} = \frac{V - K_B \frac{N_2}{N_1} \omega}{R_A}. \quad (2.24)$$

2.8 Active-Passive Suspension

To suppress vertical oscillations as well as pitch and roll oscillations to help stabilize the system we may want to add an active and/or passive suspension system. This helps make our planar motion assumption a little more realistic. First, let us consider the suspension system in Fig. 2.7.

The spring $k_2$ and damper $b_2$ model the tire or tread on the road, the spring $k_1$ and damper $b_1$ model the passive portion of the suspension system, and $u$ is the active portion. The active damping system could be magnetorheological, magnetic, etc. For off-planet applications a fully magnetic system would be preferred as a hydraulic system’s seals would quickly succumb to the abrasive and micron sized dust particles, not to mention the extreme temperature variations.
Performing a force analysis on the system gives the equation for the body

\begin{equation}
M_1 \ddot{x}_1 = -b_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2) + u,
\end{equation}

and for the suspension

\begin{equation}
M_2 \ddot{x}_2 = b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2) + b_2(\dot{w} - \dot{x}_2) + k_2(w - x_2) - u.
\end{equation}

We are looking at suppressing disturbances from the road profile \( w \) which is a stabilization, or regulation problem. We need to define some output that we would like to keep in a small neighborhood of zero. We decide that we would like the distance between the vehicle and suspension to remain a constant, so we choose our output to be the difference

\begin{equation}
y \triangleq x_1 - x_2.
\end{equation}
We can shift this problem to the origin based on whatever we want our distance to be. Following from this we try to find our input to output map using:

\[ M_1 \ddot{x}_1 = -b_1 \dot{y} - ky + u, \quad (2.28) \]

and

\[
\ddot{y} = -\left( \frac{b_1}{M_1} + \frac{b_1}{M_2} + \frac{b_2}{M_2} \right) \dot{y} - \left( \frac{k_1}{M_1} + \frac{k_1}{M_2} + \frac{k_2}{M_2} \right) y + \frac{b_2}{M_2} \dot{x}_1 + \frac{k_2}{M_2} x_1 \\
- \frac{b_2}{M_2} \dot{w} - \frac{k_2}{M_2} w + \left( \frac{1}{M_1} + \frac{1}{M_2} \right) u. \quad (2.29)
\]

Obviously \( y \) will not work as a state, because state equations cannot have input derivatives (\( \dot{w} \)). Usually we can model tires as just a spring \( k_2 \) which removes this problem, but we might also be curious about some other type of system where we do need \( b_2 \). As long as we have a \( \dot{w} \) term, we need to formulate a new state. We start by defining new states:

\[ q_1 \triangleq x_1, \quad (2.30) \]
\[ q_2 \triangleq \dot{x}_1, \quad (2.31) \]
\[ q_3 \triangleq y - \beta_0 w, \quad (2.32) \]
\[ q_4 \triangleq \dot{y} - \beta_0 \dot{w} - \beta_1 w = \dot{q}_3 - \beta_1 w, \quad (2.33) \]

and then look at the dynamics of \( q_4 \),

\[ \dot{q}_4 = \ddot{y} - \beta_0 \ddot{w} - \beta_1 \dot{w}. \quad (2.34) \]

Ultimately what we are attempting to do here is figure out a clever new state representation that only uses \( w \), but directly removes the \( \dot{w} \) term. Replacing our \( y \), \( \dot{y} \), and \( \ddot{y} \) terms with our new coordinates we get the new representation

\[
\dot{q}_4 = -\left( \frac{b_1}{M_1} + \frac{b_1}{M_2} + \frac{b_2}{M_2} \right) (q_4 + \beta_0 \dot{w} + \beta_1 w) - \left( \frac{k_1}{M_1} + \frac{k_1}{M_2} + \frac{k_2}{M_2} \right) (q_3 + \beta_0 w) \\
+ \frac{b_2}{M_2} q_2 + \frac{k_2}{M_2} q_1 - \frac{b_2}{M_2} \dot{w} - \frac{k_2}{M_2} w - \beta_0 \ddot{w} - \beta_1 \dot{w} + \left( \frac{1}{M_1} + \frac{1}{M_2} \right) u. \quad (2.35)
\]
We use the $\beta_i$ terms in our new states to cancel out our unwanted terms. There is no $\dot{w}$, so $\beta_0$ must be zero, but there is a $\dot{w}$ term, which must be set to $-b_2/M_2$. We must also make the necessary changes to Eqn. 2.28 since it now depends on $y$.

We finally get a state space representation of our suspension system in our new coordinates, and can use a control method of our choosing to force $y \rightarrow y_d$. The state-space representation is

$$
\begin{bmatrix}
\dot{q}_1 \\
\dot{q}_2 \\
\dot{q}_3 \\
\dot{q}_4
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
-b_1b_2/M_1M_2 & 0 & -b_1/M_1 & 0 \\
b_2/M_2 & 0 & 0 & 1 \\
\Gamma_1 & 0 & \Gamma_2 & \Gamma_3
\end{bmatrix}
\begin{bmatrix}
q_1 \\
q_2 \\
q_3 \\
q_4
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & 0 & 0 \\
\frac{1}{M_1} & \frac{b_1b_2}{M_1M_2} & 0 & 0 \\
0 & 0 & -\frac{b_2}{M_2} & 0 \\
\left(\frac{1}{M_1} + \frac{1}{M_2}\right) & -\Gamma_1
\end{bmatrix}
\begin{bmatrix}
u \\
w
\end{bmatrix}
$$

(2.36)

and the parameters $\Gamma_1$, $\Gamma_2$, and $\Gamma_3$ are:

$$
\Gamma_1 = \left(\frac{k_2}{M_2} - \frac{b_2}{M_2}\Gamma_3\right), \quad \Gamma_2 = -\left(\frac{k_1}{M_1} + \frac{k_1}{M_2} + \frac{k_2}{M_2}\right), \quad \text{and} \quad \Gamma_3 = -\left(\frac{b_1}{M_1} + \frac{b_1}{M_2} + \frac{b_2}{M_2}\right).
$$

2.9 Equations of Motion

First considering the vehicle with front and rear steering, we develop the kinematic equations below:

$$
\dot{x} = V \cos(\psi + \beta),
$$

(2.37)

$$
\dot{y} = V \sin(\psi + \beta),
$$

(2.38)

with the heading angle

$$
\dot{\psi} = \omega.
$$

(2.39)

Then we sum the forces in the $x$ and $y$ directions:

$$
\sum F_x = (F_{fl} + F_{fr}) \cos(\delta_f) + (F_{fl}^\perp - F_{fr}^\perp) \sin(\delta_f) \\
+ (F_{bl} + F_{br}) \cos(\delta_r) + (F_{bl}^\perp + F_{br}^\perp) \sin(\delta_r),
$$

(2.40)

$$
\sum F_y = (F_{fl} + F_{fr}) \sin(\delta_f) + (F_{fl}^\perp - F_{fr}^\perp) \cos(\delta_f) \\
+ (F_{bl} + F_{br}) \sin(\delta_r) + (F_{bl}^\perp + F_{br}^\perp) \cos(\delta_r),
$$

(2.41)
and take the z-moment about the center of mass,

$$\sum M_z = \frac{d}{2} \sum F_x + b \sum F_y.$$  \hspace{1cm} (2.42)

For a front steering vehicle $\delta_r = 0$, and we might also only drive one set of wheels, so either the front or back traction forces would then be zero. For a skid steering robot, both $\delta_f$ and $\delta_r$ are zero. From here on we will consider the skid steering configuration only, and leave the other two configurations for future work. Eqn. 2.43-2.45 show the final kinematic representations for location and heading. These include the condition when the point we’re tracking is not collocated with the center of mass, where:

$$\dot{x} = u \cos(\psi) - a\omega \sin(\psi),$$  \hspace{1cm} (2.43)

$$\dot{y} = u \sin(\psi) + a\omega \cos(\psi),$$  \hspace{1cm} (2.44)

and the heading angle remains the same

$$\dot{\psi} = \omega.$$  \hspace{1cm} (2.45)

We show the kinematic relations between the wheel rotations and the vehicle velocities $u$, $\omega$, and $v$:

$$u = \frac{r}{4}(\omega_{fr} + \omega_{fl} + \omega_{bl} + \omega_{br}),$$  \hspace{1cm} (2.46)

$$\omega = \frac{r}{d}(\omega_{fr} + \omega_{br} - \omega_{fl} - \omega_{bl}),$$  \hspace{1cm} (2.47)

and

$$v = \frac{br}{d}(\omega_{fr} + \omega_{br} - \omega_{fl} - \omega_{bl}) = b\omega,$$  \hspace{1cm} (2.48)

where each $\omega_i$ represents the angular velocity of a specific wheel.

Next we consider the force equations for the skid steering robot. After zeroing the steering angles, our equations simplify to:

$$\sum F_x = m(\dot{u} - v\omega) = F_{fl} + F_{fr} + F_{bl} + F_{br},$$  \hspace{1cm} (2.49)

$$\sum F_y = m(\dot{v} + u\omega) = -F_{fl}^\perp - F_{fr}^\perp + F_{bl}^\perp + F_{br}^\perp,$$  \hspace{1cm} (2.50)
\[ \sum M_z = I_z \dot{\omega} = \frac{d}{2}(F_{br} + F_{fr} - F_{bl} - F_{fl}) + b(F_{fl} + F_{fr} - F_{bl} - F_{br}). \] (2.51)

Next we solve for the translational dynamics \( \dot{u} \), and plug in all of our known terms and forces to get

\[
m(\dot{u} - v \omega) = \frac{K_T N_2}{N_1 R A R_t} (V_r + V_l) - \frac{K_T K_B}{R A R_t} \left( \frac{N_2}{N_1} \right)^2 (\omega_{fr} + \omega_{br} + \omega_{fl} + \omega_{bl}) \\
- \frac{B}{R_t} (\omega_{fr} + \omega_{br} + \omega_{fl} + \omega_{bl}) - \frac{J}{R_t} (\dot{\omega}_{fr} + \dot{\omega}_{br} + \dot{\omega}_{fl} + \dot{\omega}_{bl}). \] (2.52)

Then we may substitute the dynamics for a PD DC motor controller [23]

\[ V_r + V_l = K_{PT} (u_{ref} - u) - K_{DT} \dot{u}, \] (2.53)

to get the full dynamics

\[
\left( m + \frac{N_2 K_T K_{DT}}{N_1 R_t} + \frac{4J}{R_t^2} \right) \dot{u} = m b \omega^2 + \frac{N_2 K_T K_{PT}}{N_1 R A R_t} u_{ref} \\
- \left( \frac{N_2 K_T K_{PT}}{N_1 R A R_t} + \frac{4K_T K_B}{R A R_t^2} \left( \frac{N_2}{N_1} \right)^2 + 4B \right) u. \] (2.54)

We are finally left with a nonlinear equation for the translational dynamics

\[ \dot{u} = b_1 \omega^2 - b_2 u + b_3 u_{ref}. \] (2.55)

Now considering the moment equation about the z-axis, we see that we can replace the second term by the left hand side of Eqn. 2.50 and follow the same procedure as with \( \dot{u} \). Starting with

\[ I_z \dot{\omega} = \frac{d}{2}(F_{br} + F_{fr} - F_{bl} - F_{fl}) + b(m \dot{v} + mu \omega), \] (2.56)

and the PD motor controller dynamics

\[ V_r - V_l = K_{PR} (\omega_{ref} - \omega) - K_{DR} \dot{\omega}, \] (2.57)
we get

\[
\left( I_z + \frac{dK_T K_{DR} N_2}{2R_A R_t} \frac{1}{N_1} + \frac{d^2 J}{2R_t^2} + mb^2 \right) \dot{\omega} = -mbu\omega + \frac{dK_T K_{PR} N_2}{2R_A R_t} \frac{1}{N_1} \omega_{ref}
\]

\[- \left( \frac{dK_T K_{PR} N_2}{2R_A R_t} \frac{1}{N_1} + \frac{d^2 K_T K_B}{2R_A R_t^2} \left( \frac{N_2}{N_1} \right)^2 + \frac{d^2 B}{2R_t^2} \right) \omega. \]

(2.58)

After simplifying the parameters and terms we get the nonlinear angular acceleration equation

\[
\dot{\omega} = -b_4 u\omega - b_5 \omega + b_6 \omega_{ref}. \quad (2.59)
\]

Collecting all of our equations together into one system we finally have:

\[
\dot{x} = u \cos(\psi) - a\omega \sin(\psi), \quad (2.60)
\]

\[
\dot{y} = u \sin(\psi) + a\omega \cos(\psi), \quad (2.61)
\]

\[
\dot{\psi} = \omega, \quad (2.62)
\]

\[
\dot{u} = b_1 \omega^2 - b_2 u + b_3 u_{ref}, \quad (2.63)
\]

\[
\dot{\omega} = -b_4 u\omega - b_5 \omega + b_6 \omega_{ref}. \quad (2.64)
\]
3. NONLINEAR SYSTEMS AND CONTROL

Nonlinear systems are systems that do not satisfy the superposition principle. Most real systems are nonlinear, and many classifications based on structure have been made. The difficulty with nonlinear systems is that there aren’t any general design methods to handle all of the various structures we know of. Some examples of nonlinearities include:

- Chaos
- Nonlinear Damping
- Solitons
- Aperiodic Oscillations

The general nonlinear system description has the input and output relations:

\[
\dot{x} = f(x) + g(x)u + p(x)w, \quad (3.1) \\
y = h(x), \quad (3.2)
\]

where \( f(x), g(x), p(x), \) and \( h(x) \) are nonlinear functions describing the functions of state, input, disturbance, and output respectively. The different classifications mentioned earlier refer to the structures of the vector-fields \( f(x), g(x), p(x), \) and \( h(x) \).

Now that we have a nonlinear system, we might be interested to know whether we have existence and uniqueness of solutions. Considering a general nonlinear O.D.E \([24]\)

\[
\frac{dx}{dt} = f(x, t), \quad (3.3)
\]

we assume \( f \) is bounded, continuous, and differentiable at least once. The basic requirement for existence and uniqueness is for \( f \) to satisfy a Lipschitz condition in \( x \)

\[
|| f(x, t) - f(y, t) || \leq L \ || x - y \ ||. \quad (3.4)
\]

Basically, this is a bound on how fast the function can change.
3.1 Differential Equations on Differentiable Manifolds

$M$ is a differentiable manifold if $M$ is equipped with a finite or countable set of charts, such that every point is represented in at least one chart. A chart is a an open set $U$ along with a bijective mapping from $U$ to some subset of $M$, $\phi : U \to \phi(U) \subset M$ [25]. If two different points with two different charts have the same image in $M$, there is a mapping between the neighborhoods of both points. If these charts are differentiable, then they are compatible. An atlas is a union of compatible charts, and two atlases are equivalent if their union is also an atlas. A differentiable manifold is a class of equivalent atlases, whose dimension $n$ is the number of independent variables used in the charts. [16]

A tangent vector may be defined as [25]

$$\dot{x} = \lim_{\Delta t \to 0} \frac{\phi(t) - \phi(0)}{\Delta t},$$

(3.5)

where $\phi(0) = x$ and $\phi(t) \in M$. The set of vectors tangent to $M$ at $x$ is called the tangent space to $M$ at $x$, shown in Fig. 3.1. This tangent space, $T_x M$, has the same dimension as the manifold $M$ [26], [27].

![Figure 3.1. Tangent space of a sphere.](image)
For functions and vector fields on manifolds, we often use the Lie derivative notation. Given a function $f : M \to \mathbb{R}$ and a vector field $X$ defined on $M$, the Lie derivative $L_X f$ of a function $f$ along a vector field $X$ can be interpreted as the directional derivative of $f$ along $X$ [25]. At a point $p \in M$ we have

$$ (L_X f)(p) \triangleq X_p(f). \quad (3.6) $$

We may also take the union of various tangent spaces, which gives another differential manifold structure called the tangent bundle and is written as $TM$. The dimension of the tangent bundle $TM$ is twice that of the manifold $M$ due to the theorem of cartesian products of manifolds [25]. This says that the cartesian product of two manifolds $X$ and $Y$ is another manifold with the dimension specified by

$$ \dim(X \times Y) = \dim(X) + \dim(Y). \quad (3.7) $$

![Figure 3.2. Tangent bundle of a circle.](image)

If we have a mapping from the tangent bundle to the manifold, $p : TM \to M$, this is called the natural projection (naturally). The inverse image $p^{-1}(x)$ is the tangent space at $x$, $T_x M$. This is often called the fiber of the tangent bundle [25].
Considering the general nonlinear system equations again, we now know that the collection of admissible sets $x$ can be seen as a manifold $M$. The dynamic equations for $x$ are now clearly on the tangent bundle $TM$, $f(x), g(x)u \in TM$. The vector field $f(x)$ is often called the drift of the manifold.

Interestingly enough, some of the most common problems in engineering control are not in $\mathbb{R}^n$. For example, the location of a rigid body may be in $\mathbb{R}^3$ but the orientation of a rigid body is not. The orientation actually belongs to the special orthogonal group $SO(3)$. The vector fields describing the dynamics of the location, pitch, roll, and yaw about a fixed point would then be in the tangent bundle $TSO(3)$, which is equal to $\mathbb{R}^3 \times SO(3)$ [24].

### 3.2 Observability

First consider the LTI system:

\[ \dot{x} = Ax + Bu, \quad (3.8) \]
\[ y = Cx + Du. \quad (3.9) \]

For observability, we want to see if we find the initial conditions based on our outputs which is represented by

\[
\begin{bmatrix}
  y \\
  \dot{y} \\
  \ddot{y} \\
  \vdots \\
  y^{(n-1)}
\end{bmatrix} = \begin{bmatrix} u \\ \dot{u} \\ \ddot{u} \\ \vdots \\ u^{(n-1)} \end{bmatrix}, \quad (3.10)
\]

where $O$ is the observability matrix we are solving for, and $T$ is the lower triangular matrix

\[
T = \begin{bmatrix}
  D & 0 & \ldots & 0 \\
  CB & D & \ldots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  CA^{(n-2)}B & \ldots & CB & D
\end{bmatrix}. \quad (3.11)
\]
The observability matrix for this system is then

\[ O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}. \]  

(3.12)

The observability matrix must be full rank for the system to be fully observable, that is

\[ \text{rank}(O) = n. \]  

(3.13)

This is a corollary to saying the system dynamics are injective, or a one-to-one mapping. This states that if the function \( f \) is injective for all \( a \) and \( b \) in its domain, then \( f(a) = f(b) \) implies \( a = b \). More intuitively, for linear systems this means that if the rows are linearly independent, each state is observable through linear combinations of the output. We can also notice that we can get each column vector in \( O \) by taking the first \( n - 1 \) derivatives of the output \( y \). So in order to extend this to nonlinear systems, we can just use the Lie derivative. For nonlinear systems, the observation space \( O_s \) is defined as the space of all repeated Lie derivatives of the covector \( h(x) \) [28], written as

\[ O = \begin{bmatrix} L^0_f h_1 & \cdots & L^0_f h_p \\ \vdots & \ddots & \vdots \\ L^{n-1}_f h_1 & \cdots & L^{n-1}_f h_p \end{bmatrix}. \]

The system is said to be observable if

\[ \text{dim}(dO_s) = n. \]  

(3.14)

We can see that this is true if we substitute \( Cx \) for \( h(x) \), and \( Ax \) for \( f(x) \). The collection of Lie derivatives will then produce the observability matrix for the LTI system.
3.3 Controllability

Consider the same LTI system dynamics

\[ \dot{x} = Ax + Bu. \] (3.15)

The reachable subspace for the LTI system by the Cayley-Hamilton theorem is

\[ R = \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \] (3.16)

Expanding out the matrix exponential we get the controllability matrix

\[ C = [B \mid AB \mid \ldots \mid A^{n-1}B]. \] (3.17)

As with the observability matrix we say the system is controllable if this matrix is full rank, shown as

\[ \text{rank}(C) = n. \] (3.18)

This is a corollary to saying the system dynamics are surjective, a mapping that is onto. The function \( f \) is said to be surjective if for every \( y \) in its range there exists at least one \( x \) in its domain such that \( f(x) = y \). For linear systems this implies that the columns are linearly independent. It is also important to note that in linear systems, controllability and observability are related through transposition.

We want to find the controllability or reachability for the nonlinear system, so we look at the reachability equation above. After expanding the matrix exponential, we will get a collection of matrix multiplications of \( A \) and \( B \). For nonlinear systems, we can multiply vector fields by using the Lie bracket. Thus the controllability matrix is the collection of Lie brackets on the vector fields of the system [28], shown as

\[ C = [g_1, \ldots, g_m, [g_i, g_j], \ldots, [ad_{g_i}^k, g_j], \ldots, [f, g_i], \ldots, [ad_{f}^k, g_i], \ldots]. \] (3.19)

Just like the observability problem, we can easily check this formulation and see that it even works for linear systems by substituting \( Ax \) for \( f(x) \) and \( B \) for \( g(x) \).
3.4 Coordinate Transformations and Zero Dynamics

Consider the following example of system and output dynamics [24]:

\[
\begin{align*}
\dot{x}_1 &= x_3 - x_2^2, \\
\dot{x}_2 &= -x_2 - u, \tag{3.20} \\
\dot{x}_3 &= x_1^2 - x_3 + u, \tag{3.22} \\
y &= x_1. \tag{3.23}
\end{align*}
\]

In order to form a nonlinear coordinate transformation, we need to find a global diffeomorphism for the system in consideration. We know that the Lie derivative is defined on all manifolds, and the inverse function theorem will allow us to form a transformation using the output and its \(n-1\) Lie derivatives. We consider the transformations

\[
z_1 = x_1, \tag{3.24}
\]

and

\[
z_2 = x_3 - x_2^2, \tag{3.25}
\]

to construct our diffeomorphism for the example system. However, we still do not have enough functions for a coordinate transformation. The remaining transformation is often referred to as the 'zero dynamics' or the 'internal dynamics' of the system. In order to find a global diffeomorphism, we need to determine the final coordinate transformation \(z_3\) such that the Lie derivative with respect to \(g\) is zero (as in the first two coordinate changes), shown as

\[
\frac{\partial z_3}{\partial x} g(x) = \frac{\partial z_3}{\partial x_1}(0) + \frac{\partial z_3}{\partial x_2}(-1) + \frac{\partial z_3}{\partial x_3}(1) = 0. \tag{3.26}
\]

We can see that an easy solution is \(z_3 = x_2 + x_3\). Now we check the Jacobian to see if it is regular (no critical points) for all \(x\) with

\[
\frac{\partial \phi}{\partial x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3x_2^2 & 0 \\ 0 & 1 & 1 \end{bmatrix}. \tag{3.27}
\]

Indeed, the Jacobian is regular for all \(x\) and invertible, thus it is a global diffeomorphism.
3.5 Feedback Linearization

Feedback linearization is a method in which a nonlinear coordinate transformation between the input and output is found such that the transformed system is linear along all trajectories. The first $r$-derivatives of the output are the coordinate transformations. We then design the input such that the $r^{th}$ output derivative is equivalent to some desired dynamics, $\nu$, and all nonlinearities are canceled \[29\].

![Figure 3.3. Input output feedback linearization structure.](image)

We require the nonlinear change of coordinates to be a diffeomorphism (as stated before), so it must be invertible and smooth. If we consider the transformation

$$z = \Phi(x),$$ \hspace{1cm} (3.28)

we require the left inversion to be smooth and satisfy

$$\Phi^{-1}(\Phi(x)) = x,$$ \hspace{1cm} (3.29)

and the right inversion to also be smooth and satisfy

$$\Phi(\Phi^{-1}(x)) = x.$$ \hspace{1cm} (3.30)

We may use the previous section’s results to determine our diffeomorphism.
3.6 Backstepping

Backstepping is the other common method for nonlinear control apart from feedback linearization. Consider a common nonlinear system structure:

\[ \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2, \quad (3.31) \]
\[ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3, \quad (3.32) \]
\[ \vdots \quad (3.33) \]
\[ \dot{x}_{n-1} = f_{n-1}(x_1, \ldots, x_{n-1}) + g_{n-1}(x_1, \ldots, x_{n-1})x_n, \quad (3.34) \]
\[ \dot{x}_n = f_n(x_1, \ldots, x_n) + g_n(x_1, \ldots, x_n)u. \quad (3.35) \]

We view each state equation in this structure as its own subsystem, where the term coupled to the next state equation is viewed as a virtual control signal. An ideal value for signal is constructed, and the difference between the ideal and actual values is constructed such that the error is exponentially stable by Lyapunov.

We define the error between the system and model as

\[ e_1 = x_1 - x_m, \quad (3.36) \]

and choose \( \alpha_1 \) as the ideal value for virtual control \( x_2 \) such that \( e_2 \) becomes \( e_2 = x_2 - \alpha_1 \).

Then we consider a Lyapunov candidate for the first subsystem

\[ V_1 = \frac{1}{2} e_1^2. \quad (3.37) \]

We will be able to prove that the term containing \( e_1 \) is negative definite (guaranteeing stability), but we will be left with an \( e_2 \) from substituting in \( x_2 = e_2 + \alpha \). This leads us to design a virtual control for the next subsystem in the same fashion, and then combining the Lyapunov candidates to get

\[ V_2 = V_1 + \frac{1}{2} e_2^2. \quad (3.38) \]

This continues on with \( \alpha_{i-1} \) as the ideal value for \( x_i \) and their corresponding errors, until we reach the real control input \( u \). The final Lyapunov candidate function is

\[ V_n = V_1 + \ldots + V_{n-1} + \frac{1}{2} e_n^2. \quad (3.39) \]

One of the main advantages to backstepping is that we may leave helpful nonlinear terms in the equations. In feedback linearization, we have to cancel out all of the nonlinearities
using the control input and various integrators. This makes backstepping much more robust than feedback linearization, and also allows us to use nonlinear damping [30].

3.7 Linear Parameterization

Following from feedback linearization or backstepping, we obtain a control law that may be linearly parametrized as [31]

\[ u = \psi(x) + \phi^T(x)p + \Theta J(x) \nu, \]  

(3.40)

where \( \psi(x) \) are the known functions of state, \( \phi^T(x)p \) are known functions of state multiplied by unknown parameters, and \( \Theta J(x) \nu \) are state coefficients of \( \nu \) multiplied by parameter coefficients and \( \nu \). We also assume that \( p \) may be partitioned by sets \{\theta_j\} such that each element of the linearly parametrized equation may be written as

\[ u_k = \psi_k(x) + \phi_k^T(x)\theta_k + p\theta_k \sum_{j=1}^{m} J_{kj} \nu_j. \]  

(3.41)

If we want to make our control schemes adaptive, then we replace the parameters in the control law with their estimates. That is,

\[ u = \psi(x) + \phi^T(x)\hat{p} + \hat{\Theta} J(x) \nu, \]  

(3.42)

where \( \hat{p} = p - \bar{p} \).

3.8 Filtered Derivatives

It is clear from the previous chapter that feedback linearization and backstepping may require derivatives of desired trajectories. In order to get stable derivatives of the desired trajectories, we use filters whose order corresponds to the highest derivatives needed. There are numerous stable filters in existence that give varying outputs, so it is up to the designer to choose the filter most suited to the application.

The Butterworth filter provides a flat frequency response, and is often referred to as a 'maximally flat magnitude filter'. The great attribute of this filter is that it doesn’t produce any ripple effect around the cutoff frequency, but at a cost of not having as steep
of a roll-off as other filters. The bode plots are shown in Fig. 3.4 and the polynomials for the denominators of the first, second, and third order filters are

\[ P_1(s) = s + 1, \quad (3.43) \]

\[ P_2(s) = s^2 + 1.4142s + 1, \quad (3.44) \]

and lastly,

\[ P_3(s) = (s + 1)(s^2 + s + 1). \quad (3.45) \]

![Bode Diagram](image)

Figure 3.4. Frequency response for butterworth filters.

There are many other types of filters that may be used including: the Chebyshev, Bessel, Legendre, and Elliptic filters.
4. ADAPTIVE CONTROL METHODS

4.1 Model Reference Adaptive Control (MRAC)

Model Reference Adaptive Control, or MRAC, is a control system structure in which the desired performance of the system is expressed in terms of a reference model that gives a desired response to a command signal. The command signal is fed to both the model as well as the actual system, and the controller adapts the gains such that the output errors are minimized and the actual system responds like the model (desired) system.

\[ \dot{x} = -ax + bu, \tag{4.1} \]

where \(a\) and \(b\) are unknown but constant parameters. We now define a stable reference model that represents the performance we want our unknown system to have for some trajectory \(r\)

\[ \dot{x}_m = -a_m x_m + b_m r. \tag{4.2} \]
Now we match the equations with $x$ in place of $x_m$

$$-ax + bu = -a_m x + b_m r,$$  \hspace{1cm} (4.3)

and define our control law as

$$u = \frac{1}{b} [(a - a_m)x + b_m r].$$  \hspace{1cm} (4.4)

Defining the parameters $p_1$ and $p_2$, we replace them with their estimates in the control law (equivalency control)

$$u = \hat{p}_1 x + \hat{p}_2 r.$$  \hspace{1cm} (4.5)

We define the output error as the difference between the system and the reference model, that is

$$e \triangleq x - x_m.$$  \hspace{1cm} (4.6)

We substitute the control law into the error dynamics equation, and attempt to find a solution such that the error will be driven to zero, and parameter errors will go to zero as well. The error dynamics are written as

$$\dot{e} = -ax + b(p_1 x + \hat{p}_2 r) + a_m x_m - b_m r.$$  \hspace{1cm} (4.7)

Using the relation $\hat{p} = p - \tilde{p}$, we cancel out all of the terms with the exact parameter values that we don’t know to get

$$\dot{e} = -ax + b(p_1 - \tilde{p}_1)x + b(p_2 - \tilde{p}_2)r + a_m x_m - b_m r.$$  \hspace{1cm} (4.8)

Finally we get a representation that only relies on the parameter errors,

$$\dot{e} = -a_m e + b \Phi^T \tilde{p}.$$  \hspace{1cm} (4.9)

We construct a Lyapunov candidate

$$V = \frac{1}{2} e^2 + \frac{1}{2} \tilde{p}^T \Gamma^{-1} \tilde{p},$$  \hspace{1cm} (4.10)

and take the derivative

$$\dot{V} = e(-a_m e + b \Phi^T \tilde{p}) + \tilde{p}^T \Gamma^{-1} \dot{\tilde{p}},$$  \hspace{1cm} (4.11)

to prove its stability by showing each term is negative definite.
We can see that the first error term will be stable, and the entire system will be stable if we can force the other terms to be zero. We then simplify the expression and attempt to solve for the parameter adaptation. It is important to note here that since \( b \) is a constant and \( \Gamma \) is a gain matrix that we design, \( b \) can easily be ‘absorbed’ by \( \Gamma \). The final representation of the Lyapunov analysis is shown as

\[
\dot{V} = -a_m e^2 + \tilde{p}^T (\Gamma^{-1} \dot{\hat{p}} + \Phi e) .
\]  

(4.12)

The parameter adaptation law is, as we might think, a negative gradient descent relation that is a function of the output error of the system,

\[
\dot{\hat{p}} = -\Gamma \Phi e.
\]  

(4.13)

We now consider a system with the parameters listed in Table 4.1. We show results for step inputs in Fig. 4.2-4.5, and results for pulse inputs in Fig. 4.6-4.9.

<table>
<thead>
<tr>
<th>Table 4.1. Parameters for model reference adaptive control.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a = 0.75 ) \quad ( a_m = 2 ) \quad ( \hat{p}_1 = 0.8 ) \quad ( \gamma_1 = 5000 )</td>
</tr>
<tr>
<td>( b = 0.75 ) \quad ( b_m = 2 ) \quad ( \hat{p}_2 = 0.5 ) \quad ( \gamma_2 = 5000 )</td>
</tr>
</tbody>
</table>
Figure 4.2. MRAC step response.

Figure 4.3. MRAC error in step response.
Figure 4.4. MRAC control input for step.

Figure 4.5. MRAC parameter convergence for step.
Figure 4.6. MRAC pulse response.

Figure 4.7. MRAC error in pulse response.
Figure 4.8. MRAC control input for pulse.

Figure 4.9. MRAC parameter convergence for pulse.
4.2 Adaptive Sliding Mode Control (ASMC)

Adaptive Sliding Mode Control, or ASMC, is a variable structure control method that specifies a manifold or surface along with the system will operate or ‘slide’. When the performance deviates from the manifold, the controller provides an input in the direction back towards the manifold to force the system back to the desired output. ASMC has been shown to be much more robust to noise, uncertainty, and disturbances than MRAC, but requires larger input signals.

![Figure 4.10. Adaptive sliding mode control structure.](image)

![Figure 4.11. Sliding surface phase portrait.](image)
The sliding surface $s$ in Fig. 4.11 is considered a stable manifold that we want our trajectories to be driven towards as shown in the figure. We do this by guaranteeing the sliding dynamics are always driven towards zero, that is

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -k\|s\|.$$  \hspace{1cm} (4.14)

The above simplifies to

$$s \dot{s} = -k\|s\|,$$  \hspace{1cm} (4.15)

and finally gives the non-equivalent portion of the controller, which is

$$\dot{s} = -k \text{sgn}(s).$$  \hspace{1cm} (4.16)

However, we can see that if $s$ is close to zero the controller will chatter, which can damage the system. A fix for this is to introduce a ‘boundary layer’ around the sliding surface in the form of a saturation function, to smooth out the controller response and remove the chatter. The saturation function is defined as

$$\text{sat}\left(\frac{s}{\phi}\right) = \begin{cases} \frac{s}{\phi}, & s \leq \phi \\ \text{sgn}(s), & s > \phi \end{cases}$$  \hspace{1cm} (4.17)

where $\phi$ is the ‘boundary layer’. We now start in the same way as before by defining our system and the performance we would like our system to have with:

$$\dot{x} = -ax + bu,$$  \hspace{1cm} (4.18)

and

$$\dot{x}_m = -a_mx_m + b_mr.$$  \hspace{1cm} (4.19)

Then we define some stable manifold we want our system to ‘slide’ along, and determine its dynamics:

$$s \Delta \left( \frac{d}{dt} + \lambda \right) \int_0^t e \, d\tau,$$  \hspace{1cm} (4.20)

and

$$\dot{s} = \dot{e} + \lambda e.$$  \hspace{1cm} (4.21)

We replace the surface dynamics with the discrete term that represents the trajectory motion towards the manifold which is

$$-k \text{sat}(s/\phi) = -ax + bu + a_mx_m - b_mr + \lambda e.$$  \hspace{1cm} (4.22)
Solving for \( u \) we obtain the control law

\[
u = \frac{1}{b} \left[ -ksat(s/\phi) + ax - a_m x_m + b_m r - \lambda e \right]. \tag{4.23}\]

Then replacing the parameters with their estimates, we attempt to substitute \( u \) into \( \dot{s} \) and obtain an equation for \( \dot{s} \) that is stable and includes \( \tilde{p} \) which is shown by:

\[
u = \hat{p}_1 \left[ -ksat(s/\phi) - a_m x_m + b_m r - \lambda e \right] + \hat{p}_2 x, \tag{4.24}\]

\[
\dot{s} = -ax + b(p_1 - \tilde{p}_1) \left[ -ksat(s/\phi) - a_m x_m + b_m r - \lambda e \right] + b(p_2 - \tilde{p}_2) x + a_m x_m - b_m r + \lambda e, \tag{4.25}\]

\[
\dot{s} = -b \tilde{p}_1 \left[ -ksat(s/\phi) - a_m x_m + b_m r - \lambda e \right] - b \tilde{p}_2 x - ksat(s/\phi), \tag{4.26}\]

and

\[
\dot{s} = b \Phi^T \tilde{p} - ksat(s/\phi). \tag{4.27}\]

In order to solve for an adaptation law that forces the error to go to zero, we consider the Lyapunov candidate

\[
V = \frac{1}{2} s^2 + \frac{1}{2} \tilde{p}^T \Gamma^{-1} \tilde{p}, \tag{4.28}\]

and take its derivative

\[
\dot{V} = s(b \Phi^T \tilde{p} - ksat(s/\phi)) + \tilde{p}^T \Gamma^{-1} \tilde{p}, \tag{4.29}\]

to determine the stability by showing each term is negative definite. We isolate the \( \tilde{p} \) terms in

\[
\dot{V} = -sksat(s/\phi) + \tilde{p}^T (\Gamma^{-1} \tilde{p} + b \Phi s), \tag{4.30}\]

and solve for the adaptation law

\[
\dot{\tilde{p}} = -\Gamma \Phi s, \tag{4.31}\]

which will guarantee the sliding surface is stable. Notice that we still get a gradient descent relationship, which is dependent upon \( s \) instead of \( e \). We now show some step and pulse input results for a system given by Table 4.2 in Fig. 4.12-4.15 and Fig. 4.16-4.19 respectively.
Table 4.2. Parameters for adaptive sliding mode control.

\[
\begin{align*}
    a &= 0.75 & a_m &= 2 & \hat{p}_1 &= 0.8 & \gamma_1 &= 5000 \\
    b &= 0.75 & b_m &= 2 & \hat{p}_2 &= 0.5 & \gamma_2 &= 5000 \\
    \phi &= 0.1 & k &= 10 & \lambda &= 5 & \beta &= 0 
\end{align*}
\]

Figure 4.12. ASMC step response.
Figure 4.13. ASMC error in step response.

Figure 4.14. ASMC control input for step.
Figure 4.15. ASMC parameter convergence for step.

Figure 4.16. ASMC pulse response.
Figure 4.17. ASMC error in pulse response.

Figure 4.18. ASMC control input for pulse.
4.3 Extremum Seeking Model Reference Adaptive Control (ES-MRAC)

Extremum Seeking Model Reference Adaptive Control, or ESMRAC, is a control method that uses Extremum Seeking [32] for parameter adaptation rather than deriving a law from Lyapunov stability analysis. We present the control structure for ESMRAC in Fig. 4.20.

For ESMRAC we start with the same controller structure that we had in the MRAC subsection,

\[ u = \ddot{p}_1 x + \ddot{p}_2 r. \]  

(4.32)

However, we have a different structure for the parameter estimates because we are using Extremum-Seeking for adaptation. The adaptation structure is shown in Fig. 4.21.

Following from Fig. 4.21, the new parameter estimates are given by

\[ \ddot{p}_i \triangleq \dot{p}_i + c_i \sin(\omega_i t), \]  

(4.33)

where \( \dot{p}_i \triangleq p_i - \ddot{p}_i \). Now solving for the error dynamics we get

\[ \dot{e} = -ax + b[(\dot{p}_1 + c_1 \sin(\omega_i t))x + (\dot{p}_2 + c_2 \sin(\omega_i t))r] + a_m x_m - b_m r, \]  

(4.34)
where its expansion results in

$$\dot{e} = -ax + bp_1 x - b\hat{p}_1 x + bc_1 \sin(\omega_1 t)x + bp_2 r - b\hat{p}_2 r + bc_2 \sin(\omega_2 t)r + a_m x_m - b_m r.$$ (4.35)

After canceling terms this simplifies to

$$\dot{e} = -a_m e + b\phi^T [CS - \hat{p}].$$ (4.36)

A Lyapunov analysis would be performed after finding the error dynamics, but the proof of stability is extremely long, so we will just refer to [34].
We will look at three different compensator structures: a compensator that handles general signals, a compensator with enhanced robustness, and a compensator that handles persistently exciting signals. These are respectively shown by

\[ C_i = -\frac{g_i(1 + d_i s)}{s}, \]  

\[ C_i = -\frac{g_i(1 + d_i s)}{s + \sigma_i}, \]  

and

\[ C_i = -\frac{g_i}{s}. \]

The compensator chosen also restricts the choice of the cost function \( J \) (somewhat). For the first two compensator choices, we will get an adaptation law dependent upon \( \dot{J} \), which means that it cannot be dependent upon \( \dot{e} \). The third compensator will not have a \( \dot{J} \) term in the adaptation law, which means it can depend on both \( e \) and \( \dot{e} \) if we so choose.

Parameter estimates using the first compensator are described as

\[ \hat{p}_i = C_i J \sin(\omega_i t), \]  

\[ = -g_i \frac{1 + d_i s}{s} J \sin(\omega_i t). \]  

Expanding the previous result we get

\[ \dot{\hat{p}}_i - \dot{\hat{p}}_i = -g_i J \sin(\omega_i t) - g_i d_i \left( \dot{J} \sin(\omega_i t) + J \omega_i \cos(\omega_i t) \right). \]  

After plugging in the cost function \( J = \frac{1}{2} \dot{e}^2 \) and simplifying we arrive at

\[ \dot{\hat{p}}_i = \frac{1}{2} \dot{e}^2 g_i \sin(\omega_i t) + g_i d_i \left( \dot{e} \dot{\sin}(\omega_i t) + \frac{1}{2} \dot{e}^2 \omega_i \cos(\omega_i t) \right), \]

where we would plug in our error dynamics for \( \dot{e} \) if we have them, or use a Butterworth filter to create an \( \dot{e} \). Parameter estimates using the second compensator are found using

\[ \hat{p}_i = C_i J \sin(\omega_i t), \]  

\[ = -g_i \frac{1 + d_i s}{s + \sigma_i} J \sin(\omega_i t). \]
After expanding the previous result we get

\[ \dot{\hat{p}}_i - \hat{p}_i + \sigma_i(p_i - \hat{p}_i) = -g_i J \sin(\omega_i t) - g_id_i \left( \dot{J} \sin(\omega_i t) + J\omega_i \cos(\omega_i t) \right). \]  

(4.46)

Finally we arrive at the parameter adaptation law

\[ \dot{\hat{p}}_i = \frac{1}{2} e^2 g_i \sin(\omega_i t) + g_id_i \left( e\dot{e} \sin(\omega_i t) + \frac{1}{2} e^2 \omega_i \cos(\omega_i t) \right) + \sigma_i(p_i - \hat{p}_i). \]  

(4.47)

Parameter estimates using the third compensator are found using

\[
\begin{align*}
\hat{p}_i &= C_i J \sin(\omega_i t), \\
&= -\frac{g_i}{s} J \sin(\omega_i t).
\end{align*}
\]

(4.48)  

(4.49)

This third compensator allows us to come up with more creative cost functions, so we can choose \( J = \frac{1}{2} (qe + \dot{e})^2 \), and solve for \( \dot{\hat{p}} \) to get

\[ \dot{\hat{p}}_i = \frac{1}{2} (qe + \dot{e})^2 g_i \sin(\omega_i t). \]  

(4.50)

It is clear that some of these adaptation laws may not be directly calculated, but the beautiful part is that we don’t need to directly calculate these as Extremum-Seeking does this for us.

We now consider the system and parameters given in Table 4.3. We also present results to step and pulse reference inputs in Fig. 4.22-4.25 and Fig. 4.26-4.29 respectively.

Table 4.3. Parameters for ESMRAC.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.75</td>
<td>a_m</td>
<td>2</td>
<td>( \hat{p}_1 )</td>
</tr>
<tr>
<td>b</td>
<td>0.75</td>
<td>b_m</td>
<td>2</td>
<td>( \hat{p}_2 )</td>
</tr>
<tr>
<td>c_1</td>
<td>0.007</td>
<td>g_1</td>
<td>50</td>
<td>d_1</td>
</tr>
<tr>
<td>c_2</td>
<td>0.006</td>
<td>g_2</td>
<td>40</td>
<td>d_2</td>
</tr>
</tbody>
</table>
Figure 4.22. ESMRAC step response.

Figure 4.23. ESMRAC control input for step.
Figure 4.24. ESMRAC error in step response.

Figure 4.25. ESMRAC parameter convergence for step.
Figure 4.26. ESMRAC pulse response.

Figure 4.27. ESMRAC control input for pulse.
Figure 4.28. ESMRAC error in pulse response.

Figure 4.29. ESMRAC parameter convergence for pulse.
5. ADAPTIVE CONTROL OF GROUND VEHICLE

5.1 Controllability

We start by analyzing the controllability of the ground vehicle system using the method from Chapter 3. Our system has one drift vector field $f(x)$ and two input vector fields $g_1(x)$ and $g_2(x)$, so we construct the controllability matrix

$$ C = \begin{bmatrix} g_1 & g_2 & [g_1, g_2] & [f, g_1] & [f, g_2] \end{bmatrix}, $$

(5.1)

where we use the definition of the Lie bracket for the last three terms. These Lie brackets are

$$ [g_1, g_2] = \frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2, $$

(5.2)

$$ [f, g_1] = \frac{\partial g_1}{\partial x} f - \frac{\partial f}{\partial x} g_1, $$

(5.3)

and

$$ [f, g_2] = \frac{\partial g_2}{\partial x} f - \frac{\partial f}{\partial x} g_2. $$

(5.4)

We use the same vector fields $f$, $g_1$, and $g_2$ for the system as before, where

$$ f = \begin{bmatrix} u \cos(\psi) - a \omega \sin(\psi) \\ u \sin(\psi) + a \omega \cos(\psi) \\ \omega \\ b_1 \omega^2 - b_2 u \\ -b_4 u \omega - b_5 \omega \end{bmatrix}, $$

(5.5)

$$ g_1 = \begin{bmatrix} 0 & 0 & 0 & b_3 & 0 \end{bmatrix}^T, $$

(5.6)

and

$$ g_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & b_6 \end{bmatrix}^T. $$

(5.7)
Next we calculate the Jacobian matrices for each of the vector fields for use in the Lie products. The Jacobian of $f$ is
\[
\frac{\partial f}{\partial x} = \begin{bmatrix}
0 & 0 & -u \sin(\psi) - a \omega \cos(\psi) & \cos(\psi) & -a \sin(\psi) \\
0 & 0 & u \cos(\psi) - a \omega \sin(\psi) & \sin(\psi) & a \cos(\psi) \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -b_2 & 2b_1 \omega \\
0 & 0 & 0 & -b_4 \omega & -b_5
\end{bmatrix},
\] (5.8)

and the Jacobians of $g_1$ and $g_2$ are zero matrices since they only contain constant parameters. Finally, we may construct our controllability matrix for the ground vehicle system which is
\[
C = \begin{bmatrix}
0 & 0 & 0 & -b_3 \cos(\psi) & ab_6 \sin(\psi) \\
0 & 0 & 0 & -b_3 \sin(\psi) & -ab_6 \cos(\psi) \\
0 & 0 & 0 & 0 & b_6 \\
b_3 & 0 & 0 & b_2 b_3 & -2b_1 b_6 \omega \\
0 & b_6 & 0 & b_3 b_4 \omega & b_5 b_6
\end{bmatrix}.
\] (5.9)

Unfortunately the Lie product $[g_1, g_2]$ does not produce a new vector field, so our system is rank deficient for controllability and has internal or 'zero' dynamics.

### 5.2 Zero Dynamics

Consider the coordinate transformation [18]
\[
\Phi(X) = \begin{bmatrix}
x \\
y \\
u \cos(\psi) - a \omega \sin(\psi) \\
u \sin(\psi) + a \omega \cos(\psi) \\
z_5
\end{bmatrix},
\] (5.10)

where $z_5$ is the transformation representing the zero dynamics of the system. Once again we want to find the suitable $z_5$ such that $L_g z_5 = 0$. Based on our previous controllability calculations, as well as insight, we know that the zero dynamics must be $\psi$. Checking
the lie derivative requirement, we find that $L_g \psi$ is in fact equal to zero. Expressing our
diffeomorphism in terms of our state variables we have

$$Z = \Phi(X) = \begin{bmatrix} x \\ y \\ u \cos(\psi) - a\omega \sin(\psi) \\ u \sin(\psi) + a\omega \cos(\psi) \\ \psi \end{bmatrix}.$$  \hfill (5.11)

Now we must find the inverse mapping of this diffeomorphism. The inverse mapping is
given in terms of our transformed coordinates $z$ and is shown as

$$X = \Phi^{-1}(Z) = \begin{bmatrix} z_1 \\ z_2 \\ z_5 \\ z_3 \cos(z_5) + z_4 \sin(z_5) \\ \frac{1}{a}(-z_3 \sin(z_5) + z_4 \cos(z_5)) \end{bmatrix}. \hfill (5.12)$$

We already know that our heading angle $\psi$ ($z_5$) does not need to be bounded since it
is always between zero and $2\pi$. However we do need to prove the boundedness of the zero
dynamics $\dot{z}_5$. We start by taking the $L^p$ norm of the zero dynamics, and use the Minkowski
Inequality to separate the terms \cite{35} as shown below in

$$\|\dot{z}_5\|_p = \|\frac{1}{a}(-z_3 \sin(z_5) + z_4 \cos(z_5))\|_p,$$  \hfill (5.13)

and

$$\|\dot{z}_5\|_p \leq \|\frac{1}{a} z_3 \sin(z_5)\|_p + \|\frac{1}{a} z_4 \cos(z_5)\|_p.$$  \hfill (5.14)

The sine and cosine terms are maximally bounded by one, so we can further simplify to

$$\|\dot{z}_5\|_p \leq \|\frac{1}{a} z_3\|_p (\|z_3\|_p + \|z_4\|_p).$$  \hfill (5.15)

Our control law will be designed such that $\dot{x}$ and $\dot{y}$ approach their desired reference
trajectories, which we design to be bounded \cite{18}. This means that the errors will approach
a neighborhood of zero. Using this knowledge we can write the previous equation as

$$\|\dot{z}_5\|_p \leq \|\frac{1}{a} (\|\dot{x} - \dot{x}_d\|_p + \|\dot{x}_d\|_p + \|\dot{y} - \dot{y}_d\|_p + \|\dot{y}_d\|_p).$$  \hfill (5.16)
Finally, we use the knowledge that the neighborhoods are bounded to arrive at
\[
\| \dot{z}_5 \|_p \leq \| \frac{1}{a} \|_p \left( C_{e_x} + C_{\dot{x}_d} + C_{e_y} + C_{\dot{y}_d} \right),
\] (5.17)
which proves that the zero-dynamics are stable and bounded under all \( p \)-norms.

### 5.3 Cascade Normal Form and Block Backstepping

Before we can apply backstepping to our system, we need to ensure that we transform it into the so-called “cascade normal form.” We restate our system equations below:

\[
\begin{align*}
\dot{x} &= u \cos(\psi) - a\omega \sin(\psi), \\
\dot{y} &= u \sin(\psi) + a\omega \cos(\psi), \\
\dot{\psi} &= \omega, \\
\dot{u} &= b_1\omega^2 - b_2u + b_3u_{ref}, \\
\dot{\omega} &= -b_4u\omega - b_5\omega + b_6\omega_{ref}.
\end{align*}
\] (5.18-5.22)

Cascade normal form requires that the system conform to the following structure [30]:

\[
\begin{align*}
\dot{x}_1 &= f(x_1) + g(x_1)x_2, \\
\dot{x}_2 &= f(x_1, x_2) + g(x_1, x_2)x_3, \\
\dot{x}_3 &= f(x_1, x_2, x_3) + g(x_1, x_2, x_3)u.
\end{align*}
\] (5.23-5.25)

It is fairly obvious to see that if we define new vectors \( x_1 = [x \ y \ \psi]^T \) and \( x_2 = [u \ \omega]^T \) we can transform our system into the correct form:

\[
\begin{align*}
\dot{x}_1 &= J(x_1)x_2, \\
\dot{x}_2 &= \Phi^T(x_2)p + \Theta u.
\end{align*}
\] (5.26-5.27)
5.4 Model Reference Adaptive Control with Backstepping

We start by defining a stable reference model system

$$\dot{x}_1^m = A_m x_1^m + B_m r,$$

(5.28)

and defining the trajectory error as $e_1 \triangleq x_1 - x_1^m$. Then the trajectory error dynamics are

$$\dot{e}_1 = Jx_2 - A_m x_1^m - B_m r.$$

(5.29)

We also choose to design an ideal value $\alpha$ for $x_2$, the virtual input. We choose $\alpha$ such that $x_1$ behaves like the reference model,

$$\alpha \triangleq J^{-1}(A_m x_1 + B_m r).$$

(5.30)

This in turn means that our second error should be defined $e_2 = x_2 - \alpha$. Substituting $x_2 = e_2 + \alpha$ we get:

$$\dot{e}_1 = J(e_2 + \alpha) - A_m x_1^m - B_m r,$$

(5.31)

$$= Je_2 + JJ^{-1}(A_m x_1 + B_m r) - A_m x_1^m - B_m r,$$

(5.32)

$$= Je_2 + A_m e_1.$$

(5.33)

We then define the first Lyapunov candidate to determine the stability of our first subsystem

$$V_1 = \frac{1}{2} e_1^T P_1 e_1,$$

(5.34)

where $P_1$ is the solution to $A_m^T P_1 + P_1^T A_m = -Q_1$ where $P_1$ and $Q_1$ are positive definite matrices. Taking the derivative of the Lyapunov candidate we get

$$\dot{V}_1 = \frac{1}{2} e_1^T (P_1 + P_1^T)(A_m e_1 + J e_2),$$

(5.35)

$$= -\frac{1}{2} e_1^T Q_1 e_1 + e_1^T P_1 J e_2.$$

(5.36)

We can see that a proof for error convergence cannot be finished until we prove the convergence of $e_2$. The error dynamics are:

$$\dot{e}_2 = \dot{x}_2 - \dot{\alpha},$$

(5.37)

$$= \phi \gamma + \theta u - \dot{\alpha},$$

(5.38)
where the derivative of \( \alpha \) is
\[
\dot{\alpha} = \omega \frac{\partial}{\partial \psi} (J^{-1})J \alpha + A_m x_2.
\] (5.39)

The Lyapunov candidate for the dynamics is
\[
V_2 = V_1 + \frac{1}{2} e_2^T P_2 e_2,
\] (5.40)
and the derivative of the candidate is
\[
\dot{V}_2 = -\frac{1}{2} e_1^T Q_1 e_1 + e_2^T (J^T P_1^T e_1 + P_2 \phi \gamma + \theta u - \dot{\alpha}).
\] (5.41)

We isolate the terms multiplying \( e_2 \)
\[
\dot{V}_2 = -\frac{1}{2} e_1^T Q_1 e_1 + e_2^T (J^T P_1^T e_1 + P_2 \phi \gamma + \theta u - \dot{\alpha}),
\] (5.42)
in order to find our input
\[
u = \theta^{-1} \left( -\phi \gamma + \dot{\alpha} + C_m e_2 - P_2^{-1} J^T P_1^T e_1 \right).
\] (5.43)

Simplifying this expression yields
\[
u = \theta^{-1} \left( C_m e_2 - P_2^{-1} J^T P_1^T e_1 + \dot{\alpha} \right) + \Phi^T \hat{p},
\] (5.44)
where \( P_2 \) is the solution to \( C_m^T P_2 + P_2^T C_m = -Q_2 \) and \( P_2 \) and \( Q_2 \) are positive definite matrices. Now we replace the parameters in our control law with their estimates
\[
u = \hat{\theta}^{-1} \left( C_m e_2 - P_2^{-1} J^T P_1^T e_1 + \dot{\alpha} \right) + \Phi^T \hat{p}.
\] (5.45)

Then we plug our input back into our second error dynamics
\[
\dot{e}_2 = \phi \gamma + \theta (\hat{\theta}^{-1} \left( -\phi \gamma + \dot{\alpha} + C_m e_2 - P_2^{-1} J^T P_1^T e_1 \right) - \dot{\alpha},
\] (5.46)
and simplify to
\[
\dot{e}_2 = -\theta \Phi^T p + \theta \left[ \hat{\theta}^{-1} \left( C_m e_2 - P_2^{-1} J^T P_1^T e_1 + \dot{\alpha} \right) + \Phi^T \hat{p} \right] - \dot{\alpha}
\] (5.47)
\[
= C_m e_2 - P_2^{-1} J^T P_1^T e_1 + \theta \Psi^T \hat{p},
\] (5.48)

where \( \Psi^T \) is
\[
\Psi^T = \begin{bmatrix}
\omega^2 & -u & -v_1 & 0 & 0 & 0 \\
0 & 0 & 0 & -u\omega & -\omega & -v_2
\end{bmatrix},
\] (5.49)
and the $v$ terms are

$$
\begin{bmatrix}
v_1 \\
v_2
\end{bmatrix} = v = C_m e_2 - P_2^{-1} J^T P_1^T e_1 + \dot{\alpha}.
$$

(5.50)

We define our final Lyapunov candidate as a function of the previous Lyapunov candidate along with our parameter errors

$$
V_3 = V_2 + \frac{1}{2} \tilde{p}^T \Gamma^{-1} \tilde{p},
$$

(5.51)

and take the derivative as

$$
\dot{V}_3 = -\frac{1}{2} e_1^T Q_1 e_1 + e_1^T P_1 J e_2 + e_2^T P_2 \dot{e}_2 + \tilde{p}^T \Gamma^{-1} \dot{\tilde{p}},
$$

(5.52)

$$
= -\frac{1}{2} e_1^T Q_1 e_1 - \frac{1}{2} e_2^T Q_2 e_2 + \tilde{p}^T (\Gamma^{-1} \dot{\tilde{p}} + \Psi \dot{\theta} P_2^T e_2).
$$

(5.53)

In the last line since $\theta$ is a constant and $\Gamma^{-1}$ are gains we design, we can just have $\Gamma^{-1}$ 'absorb' $\theta$. Solving for the adaptation law we get

$$
\dot{\tilde{p}} = -\Gamma \Psi P_2^T e_2.
$$

(5.54)

We now show sets of results for two different reference signals. The first reference signal is a circular trajectory given by:

$$
r_x = 5 + 3 \sin(.1 t + \pi/2),
$$

(5.55)

and

$$
r_y = 5 + 3 \sin(.1 t).
$$

(5.56)

The second trajectory is another circular trajectory, but has three small loops inside. The second reference signal is given by:

$$
r_x = 3 \sin(.1 t + \pi/2) - 2 \sin(.4 t + \pi/2),
$$

(5.57)

and

$$
r_y = 3 \sin(.1 t) - 2 \sin(.4 t).
$$

(5.58)

The results for the circle trajectory are shown in Fig. 5.1-5.6 and the results for the loop trajectory are shown in Fig. 5.7-5.11.
Figure 5.1. BSMRAC circle path response.

Figure 5.2. BSMRAC control input for circle path.
Figure 5.3. BSMRAC trajectory errors for circle path.

Figure 5.4. BSMRAC velocity errors for circle path.
Figure 5.5. BSMRAC parameter convergence for circle path.

Figure 5.6. BSMRAC $p_4$ adaptation for circle path.
Figure 5.7. BSMRAC looped path response.

Figure 5.8. BSMRAC control input for looped path.
Figure 5.9. BSMRAC trajectory errors for looped path.

Figure 5.10. BSMRAC velocity errors for looped path.
Figure 5.11. BSMRAC parameter convergence for looped path.

Figure 5.12. BSMRAC $p_4$ parameter adaptation for looped path.
5.5 Model Reference Adaptive Control with Feedback Linearization

We first restate the equations of motion for the ground vehicle:

\[
\dot{x} = u \cos(\psi) - a \omega \sin(\psi), \tag{5.59}
\]
\[
\dot{y} = u \sin(\psi) + a \omega \cos(\psi), \tag{5.60}
\]
\[
\dot{\psi} = \omega, \tag{5.61}
\]
\[
\dot{u} = b_1 \omega^2 - b_2 u + b_3 u_{ref}, \tag{5.62}
\]
\[
\dot{\omega} = -b_4 u \omega - b_5 \omega + b_6 \omega_{ref}. \tag{5.63}
\]

We choose our outputs to be \(x\) and \(y\) so we can have autonomous navigation, and define the vector of outputs as

\[
h \triangleq \begin{bmatrix} x \\ y \end{bmatrix}^T. \tag{5.64}
\]

We then take derivatives until the dynamics containing our inputs show up, resulting in:

\[
\dot{h} = \begin{bmatrix} \cos(\psi) & -a \sin(\psi) \\ \sin(\psi) & a \cos(\psi) \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}, \tag{5.65}
\]
and

\[
\ddot{h} = \begin{bmatrix} \cos(\psi) & -a \sin(\psi) \\ \sin(\psi) & a \cos(\psi) \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} -u \omega \sin(\psi) - a \omega^2 \cos(\psi) \\ u \omega \cos(\psi) - a \omega^2 \sin(\psi) \end{bmatrix}. \tag{5.66}
\]

First we define new terms:

\[
J \triangleq \begin{bmatrix} \cos(\psi) & -a \sin(\psi) \\ \sin(\psi) & a \cos(\psi) \end{bmatrix}, \tag{5.67}
\]
\[
\Lambda \triangleq \begin{bmatrix} -u \omega \sin(\psi) - a \omega^2 \cos(\psi) \\ u \omega \cos(\psi) - a \omega^2 \sin(\psi) \end{bmatrix}, \tag{5.68}
\]
and

\[
\Xi \triangleq \begin{bmatrix} \omega^2 & -u & 0 & 0 \\ 0 & 0 & -u \omega & -\omega \end{bmatrix}. \tag{5.69}
\]
Then we replace our output dynamics with our newly defined terms and find our control law:

\[
\ddot{h} = J(\Xi\sigma + \theta u) + \Lambda, \quad (5.70)
\]

\[
\theta u + \Xi^T\sigma = J^{-1}(\ddot{h} - \Lambda), \quad (5.71)
\]

and

\[
u = \theta^{-1}(J^{-1}(\ddot{h} - \Lambda)) - \theta^{-1}\Xi^T\sigma. \quad (5.72)
\]

To further simplify, we define another new term

\[
\Phi^T = \begin{bmatrix}
-\omega^2 & u & a\omega^2 & 0 & 0 & 0 \\
-\omega & u\omega | \omega & \omega & -u\omega & u
\end{bmatrix},
\]

so that our control law is now

\[
u = \theta^{-1}J^{-1}\ddot{h} + \Phi^T\tilde{p}. \quad (5.74)
\]

Replacing our parameters with their estimates our control law becomes

\[
u = \hat{\theta}^{-1}J^{-1}\ddot{h} + \Phi^T\hat{p}, \quad (5.75)
\]

where \(\nu \triangleq \ddot{h}_d - A_m e\). Then we use the equivalent control to solve for our error dynamics and get an expression in terms of the parameter errors:

\[
\theta^{-1}J^{-1}\ddot{h} + \Phi^T\tilde{p} = \hat{\theta}^{-1}J^{-1}\nu + \Phi^T\tilde{p}, \quad (5.76)
\]

\[
\theta^{-1}J^{-1}\ddot{h} + \Phi^T\tilde{p} = (\theta^{-1} - \hat{\theta}^{-1}J^{-1}\nu + \Phi^T(p - \tilde{p}), \quad (5.77)
\]

\[
\theta^{-1}J^{-1}(\ddot{h} - \ddot{h}_d) = -\theta^{-1}J^{-1}A_m e - \hat{\theta}^{-1}J^{-1}\nu - \Phi^T\tilde{p}, \quad (5.78)
\]

and finally

\[
\dot{e} = -A_m e + \theta(\hat{\theta}^{-1}\nu + J\Phi^T\tilde{p}). \quad (5.79)
\]

We then define \(\Psi^T\) as the following matrix

\[
\Psi^T = \begin{bmatrix}
\omega^2 \cos(\psi) & -u \cos(\psi) & -\nu_x - a\omega^2 \cos(\psi) & au\omega \sin(\psi) & a\omega \sin(\psi) & -u\omega \sin(\psi) \\
\omega^2 \sin(\psi) & -u \sin(\psi) & -a\omega^2 \sin(\psi) & -au\omega \cos(\psi) & -a\omega \cos(\psi) & -\nu_y - u\omega \cos(\psi)
\end{bmatrix},
\]

so that we may simplify our error dynamics become

\[
\dot{e} = -A_m e + \theta \Psi^T\tilde{p}. \quad (5.80)
\]
We now construct a Lyapunov candidate
\[ V = \frac{1}{2} e^T P e + \frac{1}{2} \tilde{p}^T \Gamma^{-1} \tilde{p}, \] (5.82)
where \( P \) satisfies \( A^T P + P^T A = -Q \). We then take the derivative to perform a Lyapunov analysis as follows:
\[ \dot{V} = e^T (P + P^T)(-A_m e + \theta \Psi^T \tilde{p}) + \tilde{p}^T \Gamma^{-1} \dot{\tilde{p}}, \] (5.83)
\[ \dot{V} = -e^T Q e + e^T \theta \Psi^T \tilde{p} + \tilde{p}^T \Gamma^{-1} \dot{\tilde{p}}, \] (5.84)
and
\[ \dot{V} = -e^T Q e + \tilde{p}^T (\Gamma^{-1} \dot{\tilde{p}} + \Psi \theta^T e). \] (5.85)
Solving for our adaptation law
\[ \dot{\tilde{p}} = -\Gamma \Psi P e, \] (5.86)
we again get a gradient descent relation as a function of error.

Now we present results for the circle and looped trajectories in Fig. 5.13-5.16 and Fig. 5.17-5.20 respectively.

Figure 5.13. FLMRAC circle path response.
Figure 5.14. FLMRAC control inputs for circle path.

Figure 5.15. FLMRAC errors for circle path.
Figure 5.16. FLMRAC parameter convergence for circle path.

Figure 5.17. FLMRAC looped path response.
Figure 5.18. FLMRAC control inputs for looped path.

Figure 5.19. FLMRAC errors for looped path.
Figure 5.20. FLMRAC parameter convergence for looped path.

5.6 Adaptive Sliding Mode Control with Backstepping

We start out in the exact same way as before by obtaining our error dynamics for the trajectory,

\[ \dot{e}_1 = Je_2 + A_m e_1. \]  \hspace{1cm} (5.87)

In this case, we determine that the sliding surface can only be \( e_2 \), so our error dynamics become

\[ \dot{e}_1 = Js + A_m e_1. \]  \hspace{1cm} (5.88)

Thus the sliding surface dynamics are the same as \( \dot{e}_2 \),

\[ \dot{s} = \phi \gamma + \theta u - \dot{\alpha}. \]  \hspace{1cm} (5.89)

We construct our first Lyapunov candidate

\[ V_1 = \frac{1}{2} e_1^T P_1 e_1, \]  \hspace{1cm} (5.90)

and take the derivative

\[ \dot{V}_1 = \frac{1}{2} e_1^T (P_1 + P_1^T)(Js + A_m e_1). \]  \hspace{1cm} (5.91)
Our Lyapunov candidate is now also a function of $s$, so we have to look at the dynamics of $s$,

$$-ksat(s/\phi) = \phi\gamma + \theta u - \dot{\alpha}. \quad (5.92)$$

We construct our second Lyapunov candidate

$$V_2 = V_1 + \frac{1}{2} s^T s, \quad (5.93)$$

and take the derivative

$$\dot{V}_2 = -e_1^T Q_1 e_1 + e_1^T P J s + s^T (\phi\gamma + \theta u - \dot{\alpha}). \quad (5.94)$$

From this we can solve for our sliding mode control law and plug in our parameter estimates

$$u = \hat{\theta}^{-1}(-\dot{\phi} + \dot{\alpha} - ksat(s/\phi) - J^T P_1 e_1), \quad (5.95)$$

and simplify to

$$u = -\Phi^T \hat{\rho} + \hat{\theta}^{-1}(\dot{\alpha} - KT - J^T P_1 e_1). \quad (5.96)$$

Plugging back into the equation for $\dot{s}$ we get

$$\dot{s} = \phi\gamma + \theta \left[-\Phi^T \hat{\rho} + \hat{\theta}^{-1}(\dot{\alpha} - KT - J^T P_1 e_1)\right] - \dot{\alpha}. \quad (5.97)$$

In order to change $\phi\gamma$ into $\Phi^T \rho$, we multiply by $\theta\hat{\theta}^{-1}$. After simplification we get:

$$\dot{s} = -KT - J^T P_1 e_1 + \theta \left[\Phi^T \hat{\rho} - \hat{\theta}^{-1}(\dot{\alpha} - KT - J^T P_1 e_1)\right]. \quad (5.98)$$

and then

$$\dot{s} = -KT - J^T P_1 e_1 + \theta \Psi^T \hat{\rho}, \quad (5.99)$$

which we can easily use in the final Lyapunov analysis. We then construct our final Lyapunov candidate

$$V_3 = V_2 + \frac{1}{2} \hat{\rho}^T \Gamma^{-1} \hat{\rho}, \quad (5.100)$$

take the derivative

$$\dot{V}_3 = -e_1^T Q_1 e_1 + e_1^T P J s + s^T (-KT - J^T P_1 e_1 + \theta \Psi^T \hat{\rho}) + \hat{\rho}^T \Gamma^{-1} \hat{\rho}, \quad (5.101)$$

and simplify to

$$\dot{V}_3 = -e_1^T Q_1 e_1 - s^T KT + s^T \theta \Psi^T \hat{\rho} + \hat{\rho}^T \Gamma^{-1} \hat{\rho}. \quad (5.102)$$
Then we determine our adaptation law

\[ \dot{\hat{p}} = -\Gamma \Psi s, \]  

(5.103)

which is still a gradient descent relation, but dependent on our sliding surface which also happens to be the error of the dynamics. Now we present results for the circle and looped trajectories in Fig. 5.21-5.26 and Fig. 5.27-5.32 respectively.

Figure 5.21. BSASMC circle path response.
Figure 5.22. BSASMC control inputs for circle path.

Figure 5.23. BSASMC trajectory errors for circle path.
Figure 5.24. BSASMC velocity errors for circle path.

Figure 5.25. BSASMC parameter convergence for circle path.
Figure 5.26. BSASMC $p_4$ adaptation for circle path.

Figure 5.27. BSASMC looped path response.
Figure 5.28. BSASMC control inputs for looped path.

Figure 5.29. BSASMC trajectory errors for looped path.
Figure 5.30. BSASMC velocity errors for looped path.

Figure 5.31. BSASMC parameter convergence for looped path.
5.7 Adaptive Sliding Mode Control with Feedback Linearization

We start by defining our sliding surfaces. In this case, we want our sliding surfaces to be functions of the position errors $e_x$ and $e_y$:

\[
s_x \triangleq \left( \frac{d}{dt} + \lambda_x \right) \int_0^t e_x \, d\tau, \quad (5.104)
\]

and

\[
s_y \triangleq \left( \frac{d}{dt} + \lambda_y \right) \int_0^t e_y \, d\tau, \quad (5.105)
\]

where $r = 2$, $e_x = x - x_d$, and $e_y = y - y_d$. We then take the derivative and expand the expression to:

\[
\dot{s}_x = \left( \frac{d^2}{dt^2} + 2\lambda_x \frac{d}{dt} + \lambda_x^2 \right) e_x, \quad (5.106)
\]

and

\[
\dot{s}_y = \left( \frac{d^2}{dt^2} + 2\lambda_y \frac{d}{dt} + \lambda_y^2 \right) e_y. \quad (5.107)
\]
Simplifying we get stable error dynamic equations:

\[
\dot{s}_x = \ddot{e}_x + 2\lambda_x \dot{e}_x + \lambda_x^2 e_x, \quad (5.108)
\]

and

\[
\dot{s}_y = \ddot{e}_y + 2\lambda_y \dot{e}_y + \lambda_y^2 e_y. \quad (5.109)
\]

We define two new terms, \( \Lambda \) and \( \nu \):

\[
\Lambda \triangleq \begin{bmatrix}
-u \omega \sin(\psi) - a \omega^2 \cos(\psi) + 2\lambda_x \dot{e}_x + \lambda_x^2 e_x \\
u \omega \cos(\psi) + a \omega^2 \sin(\psi) + 2\lambda_y \dot{e}_y + \lambda_y^2 e_y
\end{bmatrix},
\]

\[
\nu = \begin{bmatrix}
\ddot{x}_d - 2\lambda_x \dot{e}_x - \lambda_x^2 e_x \\
\ddot{y}_d - 2\lambda_y \dot{e}_y - \lambda_y^2 e_y
\end{bmatrix},
\]

and then substitute our error dynamics into the equation for \( \dot{s} \)

\[
\dot{s} = -\nu + J\Phi \gamma + J\theta u + \Lambda. \quad (5.112)
\]

We rearrange for our control law, replacing \( \dot{s} \) with the discrete controller \(-KT\) where \( T \) is a diagonal matrix of signum or saturation functions (saturation in our case) to get

\[
u = (J\theta)^{-1}(-JT + \nu - J\Phi \gamma - \Lambda).
\]

To further simplify, we define

\[
\Xi p = (J\Theta)^{-1}(-J\Phi \gamma - \Lambda),
\]

and substitute this into our control law

\[
u = (J\theta)^{-1}(\nu - \dot{s}) + \Xi p,
\]

which finally results in

\[
u = (J\theta)^{-1}(\nu + KT) + \Xi \dot{p}.
\]
We set our two control laws (ideal & estimated) equal to each other in order to find the final sliding surface dynamics in terms of the parameter errors

\[(J\theta)^{-1}(\nu - \dot{s}) + \Xi p = (J\dot{\theta})^{-1}(\nu + KT) + \Xi \dot{p},\]  

(5.117)

which simplifies to

\[\dot{s} = -KT + J\theta(\Xi \dot{p} + (J\theta)^{-1}(\nu + KT)).\]  

(5.118)

We combine several terms into the matrix

\[
\Psi^T = \begin{bmatrix}
\omega^2 \cos(\psi) & -u \cos(\psi) & -\nu_x - a\omega^2 \cos(\psi) & au \omega \sin(\psi) & a\omega \sin(\psi) & -u\omega \sin(\psi) \\
\omega^2 \sin(\psi) & -u \sin(\psi) & -a\omega^2 \sin(\psi) & -au \omega \cos(\psi) & -a\omega \cos(\psi) & -\nu_y - u\omega \cos(\psi)
\end{bmatrix},
\]

(5.119)

in order to simplify the sliding surface dynamics to

\[\dot{s} = -KT + \theta \Psi^T \dot{p}.\]  

(5.120)

Once again, we construct a Lyapunov candidate, take the derivative, and solve for our adaptation law in:

\[V = \frac{1}{2} s^T s + \frac{1}{2} \dot{p}^T \Gamma^{-1} \dot{p},\]  

(5.121)

\[\dot{V} = s^T (-KT + \theta \Psi^T \dot{p}) + \dot{p}^T \Gamma^{-1} \dot{p},\]  

(5.122)

\[\dot{V} = -s^T KT + \dot{p}^T (\Gamma^{-1} \dot{p} + \Psi \theta^T s),\]  

(5.123)

and

\[\dot{\dot{p}} = -\Gamma \Psi s.\]  

(5.124)

Now we present results for the circle and looped trajectories in Fig. 5.33-5.36 and Fig. 5.37-5.40 respectively.
Figure 5.33. FLASMC circle path response.

Figure 5.34. FLASMC control inputs for circle path.
Figure 5.35. FLASMC errors for circle path.

Figure 5.36. FLASMC parameter convergence for circle path.
Figure 5.37. FLASMC looped path response.

Figure 5.38. FLASMC control inputs for looped path.
Figure 5.39. FLASMC errors for looped path.

Figure 5.40. FLASMC parameter convergence for looped path.
5.8 Extremum Seeking Model Reference Adaptive Control

Defining the control laws for ESMRAC after we’ve already looked at MRAC is quite easy. All we must do is replace the original parameter estimate relations from MRAC, with those of the extremum seeking algorithm. That is,

\[ u = \dot{\theta}^{-1}(-\phi \dot{\gamma} + \dot{\alpha} + C_m e_2 - P_2^{-1} J^T P_1^T e_1), \]

(5.125)

and

\[ u = \dot{\theta}^{-1} J^{-1} \nu + \Phi^T \ddot{p}, \]

(5.126)

where \( \ddot{p} = \dot{p} + c \sin(\omega t) \). These lead to error dynamics of the form

\[ \dot{e} = -A_m e + \theta^{-1} \Psi^T (CS - \ddot{p}), \]

(5.127)

where \( C \) is a diagonal matrix collecting perturbation gains, and \( S \) is a vector of the sinusoid perturbations. Please note that the inclusion of \( CS \) in the error dynamics means that our error will only oscillate about a neighborhood of zero. Our parameter estimates and adaptations are then defined by:

\[ \dot{\hat{p}} = C_i JS, \]

(5.128)

and

\[ \dot{\ddot{p}} = GJS + \tau(p, \ddot{p}), \]

(5.129)

where \( G \) is a collection of gains, and \( \tau(p, \ddot{p}) \) is a collection of extra terms that show up depending on the compensator \( C_i \) we use (refer to Chapter 4).

One of the major downsides to using extremum seeking for parameter adaptation is that it introduces many more design parameters that we must choose to get a stable solution. Since extremum seeking is an unconstrained optimization algorithm parameter estimates always start at zero, which may be too far removed from their real values. One of the problems with adaptive controllers is that even though they drive the parameter errors to zero, if the initial estimates are too far removed, the system may not be stable since it takes time for those errors to converge. We face this same problem with extremum seeking since we cannot control the system’s response to large parameter errors. The results for the circle trajectory are shown for both backstepping and feedback linearization.
We see that in the case of backstepping, we were able to find gains that eventually achieve error convergence, but still has unacceptable results. The adaptation takes too long, and the error build-up is so large the robot clearly goes off course. Unfortunately increasing the gains leads to more instability, as does increasing the frequency and magnitude of the disturbances which would give faster convergence. For the feedback linearization case, it is clear that we weren’t even able to find extremum seeking parameters that achieve error convergence. The performance in this case is also unacceptable. We believe that formulating a constrained optimization version of extremum seeking might allow us to provide the control designer with bounds on the controller parameters to get decent results, but this is beyond the scope of the current work. We present the backstepping and feedback linearization results for the circle trajectories in Fig. 5.41-5.45 and Fig. 5.46-5.49 respectively.

![Figure 5.41. BSESMRAC circle path response.](image)
Figure 5.42. BSESMRAC control inputs for circle path.

Figure 5.43. BSESMRAC trajectory errors for circle path.
Figure 5.44. BSESMRAC velocity errors for circle path.

Figure 5.45. BSESMRAC parameter convergence for circle path.
Figure 5.46. FLESMRAC circle path response.

Figure 5.47. FLESMRAC control inputs for circle path.
Figure 5.48. FLESMRAC errors for circle path.

Figure 5.49. FLESMRAC parameter convergence for circle path.
6. NONLINEAR DISTURBANCE OBSERVERS

We cannot talk about real systems without considering disturbances. The purpose of this chapter is to introduce the nonlinear disturbance observer for disturbance estimation in nonlinear systems.

6.1 Single-Input-Single-Output Disturbance Observer Design

We start by again showing the general nonlinear system:

\[
\dot{x} = f(x) + g(x)u + p(x)d, \quad (6.1)
\]
\[
y = h(x). \quad (6.2)
\]

We assume that the disturbance \(d\) is a linear exogenous system given below [36]

\[
\dot{\xi} = A\xi, \quad (6.3)
\]
\[
d = C\xi. \quad (6.4)
\]

A large group of disturbances can be modeled by this very system, so the observer design is actually quite robust. We start by trying to design an estimation for \(\xi\)

\[
\dot{\hat{\xi}} = A\hat{\xi} + l(x)(p(x)d - p(x)d), \quad (6.5)
\]

Substituting for \(p(x)d\), we get an observer design

\[
\dot{\hat{\xi}} = A\hat{\xi} + l(x)(\dot{x} - f(x) - g(x)u - p(x)d), \quad (6.6)
\]
\[
\hat{d} = C\hat{\xi}. \quad (6.7)
\]
The problem with this design is that we usually don’t know $\dot{x}$, so we need to find some internal state with a filtered error. Introducing our internal state

$$z = \xi - q(x),$$

we arrive at [36],

$$\dot{z} = (A - l(x)p(x)C)z + Aq(x) - \frac{\partial q(x)}{\partial x} \dot{x} + l(x)(\dot{x} - f(x) - p(x)q(x)C).$$

(6.9)

We define the disturbance error below, and analyze our disturbance error dynamics:

$$e = \xi - \hat{\xi},$$

(6.10)

$$\dot{e} = A\xi - \left( \dot{z} + \frac{\partial q}{\partial x} \dot{x} \right),$$

(6.11)

and

$$\dot{e} = A\xi - (A - l(x)p(x)C)z + Aq(x) - \frac{\partial q(x)}{\partial x} \dot{x} + l(x)(\dot{x} - f(x) - p(x)q(x)C)$$

$$+ \frac{\partial q(x)}{\partial x} (f(x) + g(x)u + p(x)d).$$

(6.12)

(6.13)

We need to remove occurrences of $\dot{x}$, so we choose $l(x) = \frac{\partial q(x)}{\partial x}$, and our error dynamics become

$$\dot{e} = A(\xi - z - p(x)) - l(x)p(x)C (\xi - z - p(x)),$$

(6.14)

which simplifies to

$$\dot{e} = (A - l(x)p(x)C) e.$$

(6.15)

At this point, we still don’t know what $l(x)$ is, but we can choose it to be any function, as long as it is always positive for any $x$. In the feedback linearization process, $L_p L_f^i h = 0$ for $i$ less than $\rho - 1$ where $\rho$ is the relative degree between the disturbance and the output. We also know that $L_p L_f^\rho h$ cannot be zero at any time, so we may choose it to be positive, and use it as our nonlinear function $q(x)$ [36], that is

$$q(x) = KL_f^{\rho-1} h(x).$$

(6.16)
Our final error dynamics are then
\[ \dot{e} = \left( A - KL_p L_f^{\rho-1} h(x) C \right) e, \] (6.17)

but now the question is, how do we choose our gain matrix \( K \)? This can be done by performing a Lyapunov analysis on the system, and requiring that the transfer function is strictly positive real. This is performed in:

\[ V = e^T P e, \] (6.18)
\[ \dot{V} = e^T (P + P^T) e, \] (6.19)
\[ \dot{V} = 2e^T P \left( A - KL_p L_f^{\rho-1} h(x) C \right) e, \] (6.20)

and

\[ \dot{V} = e^T \left( A^T P + PA \right) e - 2e^T PKL_p L_f^{\rho-1} h(x) Ce. \] (6.21)

We know the first term is stable from the Lyapunov equation, and simplifies to \(-e^T Q e\) giving:

\[ \dot{V} = -e^T Q e - 2e^T C T L_p L_f^{\rho-1} h(x) Ce. \] (6.22)

Both terms are negative definite, and we thus prove stability of the system, and can use the SPR condition \((PK = C^T)\) to find our gain matrix \( K \). The final observer system is then given as:

\[ \dot{z} = (A - l(x)p(x)C)z + Aq(x) - l(x)(f(x) + p(x)q(x)C), \] (6.23)
\[ \dot{\xi} = z + q(x), \] (6.24)
\[ \dot{d} = C\xi, \] (6.25)

### 6.2 First Order Nonlinear Example

Consider the first order nonlinear system and its disturbance below:

\[ \dot{x} = -ax^2 + bu + cx^3 w, \] (6.26)
\[ y = x, \] (6.27)
\[ w = d \sin(2\pi ft). \] (6.28)
We start by formulating our disturbance as a linear system. A general sinusoidal signal is given by the ODE

\[ \ddot{\xi} + 2\zeta\omega\dot{\xi} + \omega^2\xi = 0, \quad (6.29) \]

where \( \zeta = 0 \) and \( \omega = 2\pi f \). We choose \( \xi \) and \( \dot{\xi} \) as our state variables to form our disturbance generation system:

\[
\begin{bmatrix}
\dot{\xi} \\
\ddot{\xi}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
-\omega^2 & 0
\end{bmatrix}
\begin{bmatrix}
\xi \\
\dot{\xi}
\end{bmatrix},
\quad (6.30)
\]

and

\[
w = \begin{bmatrix}
d & 0
\end{bmatrix}
\begin{bmatrix}
\xi \\
\dot{\xi}
\end{bmatrix},
\quad (6.31)
\]

which is of the form required for our disturbance observer:

\[ \dot{\xi} = A\xi, \quad (6.32) \]
\[ w = C\xi. \quad (6.33) \]

Now constructing our observer for \( \xi \) we get

\[ \dot{\hat{\xi}} = A\hat{\xi} + l(x)\left[ cx^3w - cx^3\dot{w}\right]. \quad (6.34) \]

We obviously don’t know \( w \), so we replace the whole term by the remaining system dynamics

\[ \dot{\hat{\xi}} = A\hat{\xi} + l(x)\left[ \dot{x} + ax^2 - bu - cx^3\dot{w}\right]. \quad (6.35) \]

We cannot measure \( \dot{x} \) and we have no way (currently) of removing this term, so we must introduce a new filtered variable just as before. Following the previous section, we introduce this new state variable

\[ z \triangleq \hat{\xi} - q(x). \quad (6.36) \]

Then rearranging for \( \hat{\xi} \) and substituting into our observer equation we get

\[ \dot{z} + \frac{d}{dt}q(x) = (A - cx^3l(x)C)\left( z + q(x) \right) + l(x)\left( \dot{x} + ax^2 - bu \right). \quad (6.37) \]
In order to remove $\dot{x}$ we choose $\frac{d}{dt}q(x) = l(x)\dot{x}$. A simple choice to make this happen is $q(x) = KL_f^{-1}h(x)$ ($q(x) = Kx$) and then $l(x) = \frac{\partial q}{\partial x}$ ($l(x) = K$). The final observer form is:

$$
\dot{z} = (A + cx^3 KC) (z + Kx) - K (ax^2 - bu),
$$

(6.38)

$$
\hat{\xi} = z + Kx,
$$

(6.39)

$$
\hat{w} = C\hat{\xi}.
$$

(6.40)

We can see from the figures below that the disturbance observer performs very well and tracks the disturbance to an extremely small neighborhood of zero.

![Figure 6.1. Response with disturbance.](image-url)
Figure 6.2. Observer tracking error.

Figure 6.3. Observer estimate of disturbance.
6.3 Multi-Input-Multi-Output Disturbance Observer Design (LMI)

Now let us consider the following system with a slightly modified disturbance term [23]
\[ \dot{x} = f(x) + g(x)u + Ed, \quad (6.41) \]
and the same exogenous disturbance system
\[ \dot{\xi} = A\xi, \quad (6.42) \]
\[ d = C\xi. \quad (6.43) \]

We construct our observer in the same way as before (only the dimensions are different) with:
\[ \dot{\hat{\xi}} = A\hat{\xi} + L(Ed - E\hat{d}), \quad (6.44) \]
and
\[ \hat{d} = C\hat{\xi}. \quad (6.45) \]

We replace the \( Ed \) term with the rest of our system dynamics
\[ \dot{\hat{\xi}} = A\hat{\xi} + L(\dot{x} - f(x) - g(x)u - EC\hat{\xi}), \quad (6.46) \]
and replace \( \hat{\xi} \) with \( z + Lx \) to get
\[ \dot{z} + L\dot{x} = A(z + Lx) + L(\dot{x} - f(x) - g(x)u - EC(z + Lx)). \quad (6.47) \]

Finally, we arrive at our observer system:
\[ \dot{z} = (A - LEC)(z + Lx) - L(f(x) + g(x)u), \quad (6.48) \]
\[ \dot{\hat{\xi}} = z + Lx, \quad (6.49) \]
\[ \hat{d} = C\hat{\xi}. \quad (6.50) \]

This gives us similar error dynamics to the SISO case
\[ \dot{e} = (A - LEC)e. \quad (6.51) \]

However, this time when we define our Lyapunov candidate and take the derivative we run into a problem. Considering the Lyapunov candidate
\[ V = \frac{1}{2} e^T M e, \]  
(6.52)

when we take the derivative we get

\[ \dot{V} = e^T (A^T M + M A - C^T E^T W^T - W E C) e. \]  
(6.53)

Since we cannot always guarantee the number of disturbances, inputs, or outputs, our gain matrix \( L \) may not be square. Our \( L \) in general will be a positive symmetric definite matrix \( M \) multiplied by some rectangular matrix \( W \). To prove stability and error convergence of the system, we must solve the feasibility problem of a linear matrix inequality \([23],[37],[38]\). The linear matrix inequality is

\[ A^T M + M A - C^T E^T W^T - W E C < 0, \]  
(6.54)

where the observer gain \( L = M^{-1} W \) if there exist \( M \) and \( W \) that satisfy the LMI. If a solution exists, then it is clear that our error will converge to a small neighborhood of zero.

Suppose we have the following disturbance system coefficients:

\[
A = \begin{bmatrix}
0 & 5 & 0 & 0 \\
-5 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & -1 & 0
\end{bmatrix}, \tag{6.55}
\]

and

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}. \tag{6.56}
\]

The eigenvalues of the neutrally stable disturbance system are:

\[ e_1 = 0 + 5i, \]  
(6.57)
\[ e_2 = 0 - 5i, \]  
(6.58)
\[ e_3 = 0 + 1i, \]  
(6.59)

and

\[ e_4 = 0 - 1i. \]  
(6.60)
A feasibility solution to the LMI problem for this disturbance system are the matrices $M$ and $W$:

$$
M = \begin{bmatrix}
29.3480 & -2.8773 & 0 & 0 \\
-2.8773 & 29.3480 & 0 & 0 \\
0 & 0 & 29.3480 & -9.7827 \\
0 & 0 & -9.7827 & 29.3480
\end{bmatrix},
\tag{6.61}
$$

$$
W = \begin{bmatrix}
29.0603 & 0 \\
0 & 0 \\
0 & 24.4567 \\
0 & 0
\end{bmatrix}.
\tag{6.62}
$$

The eigenvalues then become:

$$
e_1 = -0.4999 + 5.0240i,
\tag{6.63}
$$

$$
e_2 = -0.4999 - 5.0240i,
\tag{6.64}
$$

$$
e_3 = -0.4687 + 1.0454i,
\tag{6.65}
$$

and

$$
e_4 = -0.4687 - 1.0454i,
\tag{6.66}
$$

which are in the left hand plane, giving stability.

### 6.4 Ground Vehicle Disturbance Observer

We consider our ground vehicle system again,

$$
\dot{x} = f(x) + g(x)u + Ed,
\tag{6.67}
$$

with the full system expressed as

$$
\dot{x} = \begin{bmatrix}
 u \cos(\psi) - a \omega \sin(\psi) \\
 u \sin(\psi) + a \omega \cos(\psi) \\
 \omega \\
 b_1 \omega^2 - b_2 u \\
 -b_4 u \omega - b_5 \omega
\end{bmatrix} + \begin{bmatrix}
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 b_3 & 0 \\
 0 & b_6
\end{bmatrix} \begin{bmatrix}
 u_{ref} \\
 \omega_{ref}
\end{bmatrix} + \begin{bmatrix}
 0 & 0 \\
 0 & 0 \\
 0 & 0 \\
 1 & 0 \\
 0 & 1
\end{bmatrix} \begin{bmatrix}
 \delta u \\
 \delta \omega
\end{bmatrix}.
\tag{6.68}
$$
Considering the same disturbance system form, we also design our disturbance observer in the same fashion as before with \( \hat{\xi} \) and \( \hat{d} \) as

\[
\dot{\hat{\xi}} = A\hat{\xi} + L(\dot{x} - f(x) - g(x)u - E\hat{d}),
\]

(6.69)

and

\[
\dot{\hat{d}} = C\hat{\xi}.
\]

(6.70)

After plugging in the internal state to remove the \( \dot{x} \) term we get

\[
\dot{z} = (A - LEC)(z + q(x)) + (l(x) - \frac{\partial q}{\partial x})\dot{x} - l(x)(f(x) + g(x)u).
\]

(6.71)

We choose \( q(x) \) to be

\[
q(x) = KL_f^{p-1}h(x) = K\begin{bmatrix} u \\ \omega \end{bmatrix}^T,
\]

(6.72)

which means

\[
L = K.
\]

(6.73)

Now we need to check that our error dynamics are stable, where

\[
e = \xi - \hat{\xi}.
\]

(6.74)

The error dynamics are

\[
\dot{e} = A\xi - A\hat{\xi} - L(ED\xi - EC\hat{\xi}),
\]

(6.75)

\[
= (A - LEC)e,
\]

(6.76)

which is exponentially stable provided we have a suitable observer gain \( L \). In order to find this gain we must perform a Lyapunov analysis, which will give the same LMI as the previous section,

\[
A^T M + MA - C^T E^T W^T - WEC < 0.
\]

(6.77)
We consider the following disturbance coefficients:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 \\
0 & 0 & -0.1 & 0
\end{bmatrix},
\]

and

\[
C = \begin{bmatrix}
10 & 0 & 0 & 0 \\
0 & 0 & 10 & 0
\end{bmatrix}.
\]

After solving the feasibility problem, we get the following LMI solution:

\[
M = \begin{bmatrix}
1.3410 & -0.4470 & 0 & 0 \\
-0.4470 & 1.3410 & 0 & 0 \\
0 & 0 & 1.3610 & -0.5103 \\
0 & 0 & -0.5103 & 1.3610
\end{bmatrix},
\]

and

\[
W = \begin{bmatrix}
0.1061 & 0 \\
0 & 0 \\
0 & 0.0665 \\
0 & 0
\end{bmatrix},
\]

where the observer gain \( L = M^{-1}W \) is

\[
L = \begin{bmatrix}
0.0890 & 0 \\
0.0297 & 0 \\
0 & 0.0568 \\
0 & 0.0213
\end{bmatrix}.
\]

The results of the observer with a backstepping based model reference controller results are shown in Fig. 6.4-6.8. We can clearly see that the observer tracks both disturbances to an extremely small neighborhood of zero and has good performance.
Figure 6.4. BSMRC response with disturbances.

Figure 6.5. Observer error for disturbance in $u$. 
Figure 6.6. Observer error for disturbance in $\omega$.

Figure 6.7. Observer estimate of disturbance in $u$. 
Figure 6.8. Observer estimate of disturbance in $\omega$. 
7. SUMMARY

To sum up, adaptive nonlinear control is an extremely useful tool for solving difficult problems. The motivation and background for autonomy was introduced, along with a simple design example for an autonomous ground vehicle. We explored some of the most interesting theoretical aspects of nonlinear systems and control, why they are important, and how they are used for real world systems. We also provided a tutorial on some of the most common and effective control design methods along with the most common and effective trajectory linearization techniques with several examples. Combinations of the aforementioned methods were used to design adaptive nonlinear controllers for a ground vehicle, and their performances were shown in simulation results. Finally, we provided SISO and MIMO techniques for estimating disturbances in nonlinear systems and tested these in simulation as well.
LIST OF REFERENCES


[8] National Aeronautics and Space Administration. Martian methane reveals the red planet is not a dead planet.


