Bucknell University
Bucknell Digital Commons

Master's Theses | Student Theses

Spring 2019

Development of Physics Based Machine Learning Algorithms

Rob Jennings
rjj008@bucknell.edu

Follow this and additional works at: https://digitalcommons.bucknell.edu/masters_theses

Part of the Other Mechanical Engineering Commons

Recommended Citation
Jennings, Rob, "Development of Physics Based Machine Learning Algorithms" (2019). *Master's Theses*. 221.
https://digitalcommons.bucknell.edu/masters_theses/221

This Masters Thesis is brought to you for free and open access by the Student Theses at Bucknell Digital Commons. It has been accepted for inclusion in Master's Theses by an authorized administrator of Bucknell Digital Commons. For more information, please contact dcadmin@bucknell.edu.

DEVELOPMENT OF PHYSICS BASED MACHINE LEARNING ALGORITHMS

by

Robert Jennings

A Thesis

Presented to the Faculty of
Bucknell University
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in the Department of Mechanical Engineering

Approved:

Indranil Brahma
Thesis Advisor

Constance Ziemian
Chair, Department of Mechanical Engineering

Andrew Sloboda
Engineering Thesis Committee

Joshua Stough
Engineering Thesis Committee

May 6, 2019

Acknowledgments

I would first like to thank my parents for their love and support in whatever I do. They have been a constant positive influence in my life. Thank you to my family for all their support throughout my life.

I'd like to extend my thanks to my advisor, Dr. Indranil Brahma, who has been a great mentor for me during this degree. I'm also grateful for Dr. Andrew Sloboda and Dr. Joshua Stough for agreeing to be on my committee and their help along the way.

I'd like to thank Hugh Weber, Aaron Clarke, Tim Baker, Dan Johnson, and Wade Hutchison for their help in implementing much of the apparatuses. I'd like to thank Dr. Keith Buffinton, who graciously allowed me use of his load cell.

Contents

Nomenclature **xi**

Abstract **xiii**

1 Introduction **1**

1.1 Background Information 1

1.2 Goals 6

2 Literature Review **8**

2.1 Baseball Pitch System 8

2.2 Physics Based Machine Learning 11

2.3 Motivation 12

3 System Setup **14**

3.1 Overview 14

3.2 Ball Rotation Angle Table 15

3.3 Baseball Spin Assembly 16

3.4 Apparatus Control 16

3.5 Computational Setup 18

3.6 Physics Based Machine Learning Setup 20

4 Testing Methodology **25**

4.1 Experimental Testing Plan 25

4.2 Experimental Post Processing 28

4.2.1 Filtering and Calibration 28

4.2.2 Air Velocity 33

4.2.3 Lift and Drag 33

4.2.4 Force Coefficients 34

5 Results And Discussion **35**

5.1 Experimental Results 35

5.2 Ball Trajectory 50

5.3 Computation Results 53

5.4 Complex System Exploration 77

5.5 Physics Based Machine Learning Algorithm 77

5.5.1 Extrapolation 82

5.5.2 Interpolation 86

5.6 Uncertainty Analysis 88

5.7 Future Work 90

Conclusion **92**

References **94**

Appendix A **96**

List of Tables

3.1 Feature Spaces 23

4.1 Input Variable Ranges 27

4.2 Butterworth Attentution 32

5.1 Extrapolation NN R^2 Values - Side force 84

5.2 Extrapolation NN R^2 Values - Drag force 85

5.3 Extrapolation NN R^2 Values - Lift force 86

5.4 Interpolation NN R^2 Values 87

5.5 Data Order 96

List of Figures

1.1 Smooth Sphere Drag Coefficient 6

2.1 Ball Point of Contact [5] 9

2.2 Baseball Pressure Plot [1] 10

2.3 Ball Trajectory [1] 10

3.1 Full Experimental Assembly Setup 15

3.2 Ball Spin Assembly 16

3.3 Motor Driver Board 17

3.4 Encoder Shield 18

3.5 Baseball Experimental and Computational Orientation Axes 19

3.6 Computational Mesh 20

3.7 Neural Network Layer Setup 21

3.8 Physics Based Machine Learning Algorithm Setup 24

4.1 Setup Variable Visualization 26

4.2 Raw Data FFT 29

4.3 Raw Data Example 31

4.4 Frequency Response 32

5.1 Air Velocity Effect on Force 37

5.2 Air Velocity Effect on Baseball Coefficients 40

5.3 Smooth Sphere Drag Coefficient vs. Reynolds numbers 40

5.4 Reynolds Number effect on Baseball Coefficients 42

5.5 Sting Angle Effect on Forces 44

5.6 Spin Rate Effect on Side Force 46

5.7 Spin Rate Effect on Baseball Coefficients 48

5.8 Seam Orientation Effect on Side Force 50

5.9 Ball Trajectory 52

5.10 Computational Velocity Effect on Forces 56

5.11 Computational Velocity Effect on Baseball Coefficients 59

5.12 Computational Sting Angle Effect on Baseball Force 62

5.13 Computational Seam Orientations Effect on Force 64

5.14 Computational Spin Rate Effect on Side Force 65

5.15 Baseball Streamline 66

5.16 3D pressure plot 67

5.17 Sting Angle Effect on Baseball Pressure Cut Plot 69

5.18 Sting Angle Effect on Baseball Velocity Cut Plot 70

5.19 Air Velocity Effect on Baseball Pressure Cut Plot 71

5.20 Velocity Cut Plot 72

5.21 Direct Comparison: Experimental and Computational Results 74

5.22 Direct Comparison: Experimental and Computational Results with Error Bar 76

5.23 Additional Data Locations 79

5.24 R^2 Asymptote Example 80

5.25 Physics Based Machine Learning Algorithm Setup 81

5.26 Force Sensor Error 88

5.27 Repeatability Trial 90

5.28 Baseball Computational Volume Setup 97

5.29 Inclinometer 98

5.30 Spin Rate Effect on Drag and Lift force 99

5.31 Baseball Rotation Shaft 100

5.32 Motor with Mounted Encoder 100

5.33 Experimental and Computational Effect of Spin Rate on Forces . . . 103

5.34 Top Adapter Plate Drawing 104

5.35 Apparatus Bracket Drawing 105

5.36 Bottom Table Drawing 106

5.37 Top Adapter Table Drawing 107

5.38 Gear To Shaft Coupler Drawing 108

5.39 Sting Outer Shaft 109

5.40 Apparatus Slider 110

5.41 Top Adapter Plate Drawing 111

Nomenclature

C_s : Side Coefficient

C_d : Drag Coefficient

C_l : Lift Coefficient

θ_y : Seam Orientation angle: Y axis

θ_z : Seam Orientation angle: Z axis

θ_b : Baseball Sting Angle

ω_b : Baseball Spin rate

v : Air Velocity

ρ : Air Density

P : Air Pressure

PWM : Pulse Width Modulation

SS : Stead State

NN : Neural Network

RHR : Right Hand Rule

B_s : Systematic Bias

B_d : Random Bias

B_t : Total Bias

PBMLA : Physics Based Machine Learning Algorithm

SNR : Signal to Noise Ratio

Abstract

In this study, a baseball pitch was examined to try to understand its behavior, and make a predictive model of it. A baseball pitch was tested experimentally with a wind tunnel and modeled computationally with COMSOL CFD software. Five input variables (spin rate, sting angle, seam orientation: Y axis, seam orientation: Z axis, and air velocity) were controlled, with force in three axes recorded as outputs. The experimental and computational results were examined and seen to be interdependent for all input variables. Experimental and computational data were both insufficient for predicting system behavior. Experimental data collection would have required an unreasonable amount of time, while computational data collection provided adequate qualitative results, but lacked quantitative accuracy.

Both of these attempts to understand the system behavior fell short, indicating that the baseball was a complex system. This lack of system understanding from either experimental or computational results necessitated a different approach to predicting system behavior. This led to the application of a physics based machine learning algorithm, which aimed to combine experimental and computational data. This combination of data improved the predictive ability of the system, showing that a physics based machine learning algorithm can be used to better understand baseball pitch behavior. This result points to the possibility that a physics based machine learning algorithm can help facilitate the understanding of a complex system.

Chapter 1

Introduction

1.1 Background Information

Understanding how a system works is beneficial for knowing how it will react in different situations and optimizing it for a specific purpose. For instance, knowing how an engine utilizes fuel for power would be beneficial when making an engine with more power or more fuel efficiency. Some systems have intuitive behavior, with clearly correlated cause and effect relationships between inputs and outputs. For instance, a car's driving range has a clear correlation with the amount of fuel and mpg. However, many systems are more complicated than this, with changes in input variables having an unclear correlation with output. Many of these systems would be considered complex, which is a system that is too complicated to be meaningfully modeled by the known physics of the model, or the physics are known, but the computational power required is unreasonably high. Predicting these complex systems can be very hard and inaccurate. Various techniques can be used in an attempt to predict the behavior of complex systems, but each has significant drawbacks.

One technique is to physically build and experimentally test a system to create a data repository that can be referenced to predict future behavior. This technique is only applicable within the bounds of the tested input variables, so to get a full understanding of the system, each input variable must be varied individually. This poses a problem for many complex systems since often there are many potential variables, which makes the amount of trials go up exponentially as parameters increase.

Then, each variable step change must be sufficiently small since this physical model data can only be used to interpolate future behavior. These factors can quickly make testing a system for all parameter changes require millions or billions of trials, depending on number of variable inputs and data density. This can make physically modeling the system very time consuming and cost ineffective. This problem is known as the curse of dimensionality. Another problem is the ability to properly measure quantities, with it sometimes being impossible or very cost and time prohibitive. For instance, if emissions for an engine were to be mapped for a range of operating situations, parameters such as piston temperature, air to fuel ratio, engine ignition, air inlet pressure etc would be useful to know. Utilizing sensors for all these parameters would be very hard to accomplish without compromising the engine behavior itself.

This has given rise to computer models that attempt to simplify the problem into a model based on given equations. This model negates the necessity of building a physical model and measuring the actual parameters, and replaces that with a computer generated model of the system based on governing equations. For example, in fluid dynamics a common equation used to approximate fluid flow is the Navier-Stokes equation. This equation can be used to calculate fluid flow over a wide range of conditions. However, it is too computationally expensive to solve and fully describe system behavior. Theoretically, if infinite computing time were used, a truly accurate result could be obtained from a computational fluid dynamic (CFD) model. However because this is not possible, approximations are used to be able to estimate an answer, with a k-epsilon method often used to describe turbulent flow. This results in potentially inaccurate models, making predicting a complex system with CFD models flawed. Pouya[1] uses CFD analysis to predict the trajectory and end position of specific baseball pitches and when compared to real pitches, is seen to be an inaccurate and off by about 1 foot compared to a real pitch (for a typical MLB pitch length of 18.4 meters). It was seen to have the same direction of movement, indicating that CFD is only useful for general trends in turbulent models, and not able to truly replicate the physics.

The next option could be to try to take real world data for some range in variables that can be readily changed, and then train a machine learning algorithm to attempt to get a more complete picture of the ball's movement pattern based on input parameters. This would work well, if the model was only asked to interpolate results based on similar trials done. If the model was asked to extrapolate potential movement of pitches far away from existing data, the model would break down because the machine learning algorithm can only interpolate results successfully, and produces inaccurate results when extrapolating. This would limit the amount of useful predictions gained over simply experimentally testing situations.

To show specifically how each technique can independently fall short for predicting a result, a baseball pitch's movement can be examined. To know the movement, the forces acting on the baseball as it moves through the air and spins would need to be predicted. The baseball's aerodynamics rely heavily on its raised stitches, and a generalized model for the ball movement would be very expensive and difficult to produce. Experimentally modeling the baseball would mean taking data for the full breadth of ball angles, axis of rotation, spin rates, and ball speeds. This would be very difficult, expensive, and time consuming. A CFD model would also fail to completely model the system, since CFD modeling cannot handle turbulence accurately and solving the fluid flow equation for the baseball numerically would be too computationally expensive. A machine learning algorithm with the experimental data would not meaningfully extend behavior prediction because it would be forced to extrapolate for any ball movement not experimentally investigated.

This means that for a complex system, there are significant drawbacks to each individual technique at predicting complex system behavior. Taking real world data can be to complicated and expensive for many complex systems. CFD models attempt to approximate the physics, which then results in inaccurate results. Machine learning necessitates accurate data for an accurate result, and cannot cope with predicting results based on significantly varying an input variable.

Improving the ability to predict the behavior of a complex system accurately and for a wide range of variables (that might not be tested in real life) is important. Since the three aforementioned methods alone cannot succeed, we propose combining them. Combining these methods could allow for a more accurate model that can predict results for a wider range of input variables. The proposed work aims to utilize multiple models to better predict behavior of a complex system when using any one model is not feasible due to computation time, cost, or accuracy. A baseball pitch is specifically selected as the complex system explored, as it is able to be modeled physically and computationally. It is a good example of a system that can be physically modeled and computationally modeled, but with potential problems with both techniques.

To improve the predictive ability, a physics based machine learning algorithm can be used to combine these techniques to make a better predictive model. The intention is to give the machine learning algorithm data that is relevant to the underlying physics in the system, which allows the algorithm to base results off of known physical changes. By using the experimental and CFD data, these physics based input variables can be determined and used in the machine learning algorithm.

Originally, a machine learning algorithm might take the real world data and

attempt to make a meaningful model from it, but the model breaks down as the input variables diverge from existing data, because machine learning algorithms can interpolate well, but extrapolate poorly. By combining data from the experimental tests and the physics associated with the CFD simulation models, an improvement to this algorithm can be explored.

To understand how the combination of models will work, some terms must be defined. The feature space of the machine learning algorithm is the input variables for the algorithm to do some sort of computation to get an output of the system that is hopefully accurate. For example, if a car range prediction algorithm was developed, a potential feature space would be amount of gas in the tank and car's mpg and the output would be the car's range. This feature space is relatively simple and well correlated to the output, and the algorithm to predict car range would be very accurate. However, a complex system's input variables may not have a clear correlation with the output, and there might be too many input variables to intuitively know the output. This means that the feature space for these complex systems might have a high dimension and not correlate well to the output, and thus produce poor predictive results. The feature space might also be sparsely populated, meaning that each variable is not independently varied for a full range of values and without a sufficiently small step change. This leads to the algorithm not having enough data to make accurate predictions due to the algorithm being forced to predict results far away from known data points.

Combining data from the physical and CFD model aims to improve this feature space to allow the machine learning algorithm to make better predictions. To improve upon the loosely correlating input data, the feature space can be changed to instead have physics based input variables that correlate better to the output. Tinoco [2] combined physical variables to successfully predict the depth-averaged velocity for turbulent flow along an open channel with vegetation in it. Similarly, in a baseball pitch, the CFD data (boundary layer thickness, pressure etc) for each trial would be examined to look at correlations between any one, or combination of this extra data and the ball forces. These new variables would be the new feature space and allow the algorithm to have a better correlation between inputs and outputs derived from the physics of the system.

The new feature space has another benefit for the model. The new feature space variable is a function of original feature space variables, which makes multiple original feature space variables represented by one new feature space variable. This means a change in multiple original feature space variables becomes one change in one new feature space variable. This is important because it addresses the data density

problem. For example, say there were three variables in the original feature space which were combined to 1 variable in the new feature space. If each original variable had ten potential values in a range of ten, then the algorithm would have data every 1 unit change. After the variables were combined to a new variable that has a range of ten, there is now 103 data points for the new feature space. This new feature space is much more dense and the machine learning algorithm can handle more change in the original feature space because it can translate it to a new feature space variable with a much higher data density. This allows the algorithm to go from having to extrapolate based on many sparsely populated original feature space variables to interpolating based on only a few densely populated new feature space variables.

To explain this better, the Reynolds number relationship to drag coefficient for a smooth sphere will be used as an example as to how these two techniques can be used to improve the feature space. Figure 1.1 shows a clear relationship between Reynolds number and C_d. If we had a smooth sphere and wanted to predict C_d values based on some data we can measure, we could reasonably measure reference length of the ball, temperature, velocity, and viscosity of the fluid flowing for many instances of these input variables. With this data, we would not see any one variable having a strong correlation with the C_d value. A predictive model would have a tough time predicting a C_d value for every instance of every variable. However, if the feature space were changed to Reynold's number, a relationship can easily be seen between the feature space and the C_d. Since the Reynold's number is derived from a combination of the old feature space variables, a change in multiple original feature space variables translates into just one change in Reynolds number. This allows for a larger change in the original feature space to be predicted. The new feature space is also more densely populated since it has the data from all the original feature space data points. A machine learning algorithm with the new feature space would have a better predictive ability because of the higher data density, and be able to predict a larger change in any one original feature space variable.

This same technique will be used to explore how combining variables may result in a better predictive model for the baseball pitch. The original feature space of the baseball pitch is the seam orientation: Y axis, seam orientation: Z axis, spin rate, sting angle, and air velocity. This amounts to five input variables with each one having many potential values. It would be very time consuming to gather sufficient data, and the correlation between the input and output (the forces on the ball) would likely be poor. This means that the machine learning algorithm would be able to predict forces for only a small change from any given trial. If the input data could somehow be combined with the known physics to make a meaningful input parameter that correlates well to the forces, then the algorithm could much more accurately predict

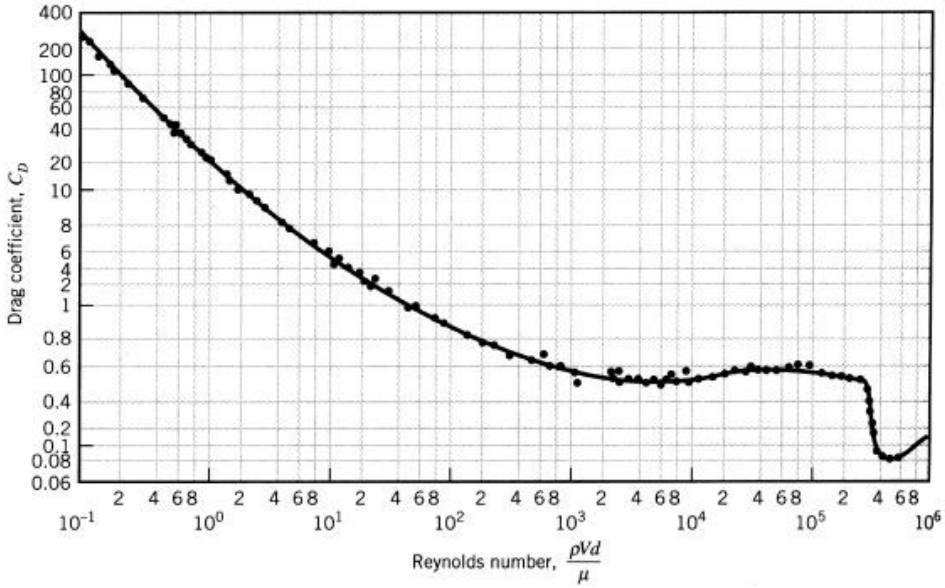


Figure 1.1: Reynolds Number Vs. Drag Coefficient of a smooth sphere.

outcomes. This is where a CFD model, such as COMSOL, becomes useful. By running a simulation of the ball in the same situation as an experiment, more data can be obtained about a trial. This would include results like pressure distributions, air velocity at certain points, boundary layer thickness, and other potentially useful data that were not experimentally obtained. Using these new data points for all the trials, a combination of results can be explored to try and find a new input variable that has a strong correlation to the forces acting on the ball. Since this new variable would be a combination of original variables, it would have a higher data density than the original variables. This new variable (or variables) would be used in the new feature space and allow the algorithm to more accurately predict the outcomes, as well as predict a larger range of baseball behavior.

1.2 Goals

The first goal was to collect data on a spinning baseball at various angles. An apparatus was designed and fabricated to position a ball at a specified angle, relative to the airflow, and rotate it. It also allowed the baseball seam orientation to be controlled. This change in baseball seam orientation changed the turbulence profile,

and the forces acting on the ball. Another apparatus was constructed to record the forces acting on the ball along three axes in order to describe the behavior of the baseball.

A computer simulation of a baseball pitch was also produced. This aimed to replicate the experimental setup. This required a baseball to be obtained and a mesh applied to it to allow for computational fluid dynamics (CFD) analysis to be done. A parametric study was then set up to allow relevant input variables like spin rate and seam orientation to be controlled.

The next goal was to get the relevant pitch data both experimentally and computationally. Varying the baseball spin rate, sting angle, seam orientations, and air speed would get a data set of baseball forces and behavior for the varying baseball pitch trials. The experimental apparatus was used to experimentally obtain this data, while CFD modeling would simulate the baseball at the same test points, as well as additional points. This repository of turbulent flow data extended past works that mostly deal with a baseball at one axis of rotation and spin angle, or no spin at all [5] [1]. Then, the baseball pitch data was looked at to see if a baseball pitch is in fact a complex system based on its behavior and the agreement of computational and experimental data.

Then, modeling techniques were combined to explore how incorporating the known physics of a system can improve prediction ability. Specifically, the change in feature space to incorporate physics based input variables and addition of computational data points was utilized to explore improving a physics based machine learning algorithm's predictive ability of a baseball pitch.

Chapter 2

Literature Review

Previous work related to the subjects of baseball behavior and physics based machine learning were used to inform this thesis. Previous work on baseball behavior has included papers on experimental and computational attempts to obtain data for a baseball. Previos work on physics based machine learning demonstrates how others have attempted to use physics to change a machine learning algorithm and predict some behavior. This work aims to progress in both of these areas. This study aims to expand the baseball data obtained both experimentally and computationally. It also aims to approach a physics based machine learning algorithm in a new way, and predict a complex system with these physics based parameters.

2.1 Baseball Pitch System

Previous work that focused on a baseball pitch attempted to predict some set of behavior, through either experimental or computational means. Higuchi [5] explores the effect of backspin on the hitting accuracy of a batter. He goes on to talk about how backspin can impact how well a pitch is tracked, and ultimately hit by a batter. Figure 2.1 shows the effect that backspin has on the point of impact of a pitch. This shows a correlation between the spin rate and the placement of the ball at point of contact. This study looked at real world data from pitchers, which replicates the actual behavior of a baseball closely. This is useful for comparing this study's data to real world data. The study does not specify important parameters about the baseball,

such as seam orientation. It also does not attempt to make any generalizations about why the baseball might behave different ways, making direct comparison to experimental data difficult.

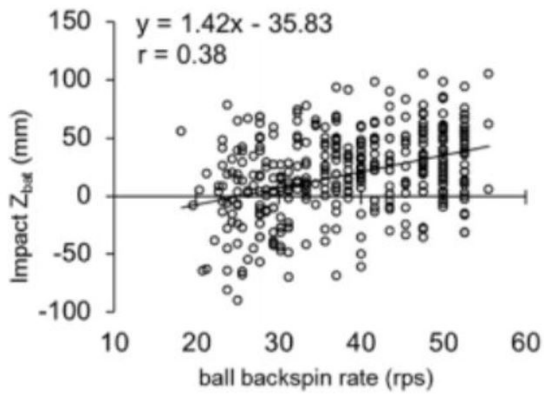


Figure 2.1: Scatterplot with the linear regression equation coefficient for impact Z_{bat} at the moment of ball-bat impact against the ball backspin rate of the projected ball [5].

Jalilian [1] examines the aerodynamics of common sports balls, including baseballs through the use of CFD analysis. Through this exploration, various physical results are computed, and visual plots are examined. The C_d and C_l are determined, as well as the trajectory of a typical baseball. A pressure slice surrounding these balls is shown in Figure 2.2. The pressure cut plot shown for the baseball shows a very similar pressure pattern to those attained in this thesis. A high pressure zone at the front of the ball was followed by a low pressure zone on the sides, and a non-symmetrical trailing pressure zone behind the baseball. A baseball pitch trajectory is also visualized in Figure 2.3. This study was useful in comparing computational visualizations and ball trajectories. The study lacked any experimental data and only had a couple data points of baseball forces, making predicting general ball behavior difficult.

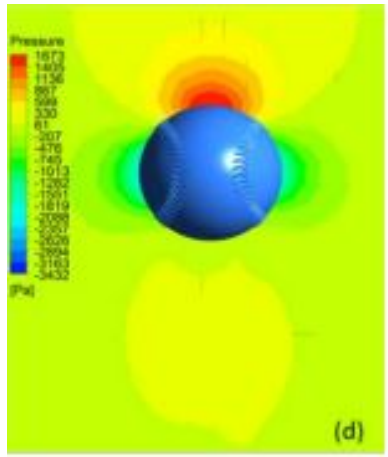


Figure 2.2: Pressure cut plot for a baseball [1].

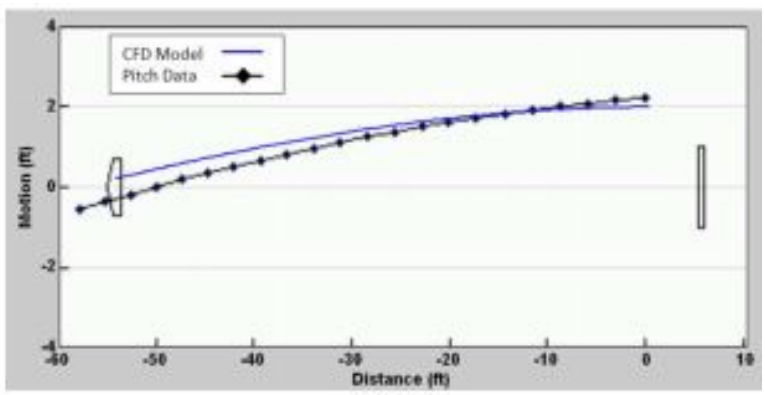


Figure 2.3: Baseball trajectory for CFD and actual pitch data [1].

These studies provide a useful guideline for what has been previously explored with baseball behavior. They do however lack much of the detail that this study aims to explore. Previous works lack the full breadth of baseball input variables, not asserting a seam orientation or sting angle. They also often do not have more than a handful of data points, making behavioral trends hard to discern. This study aims to have better control over the input variables of the baseball, and attain a much larger data repository.

2.2 Physics Based Machine Learning

Swischuk et al. [9] explores how physics variables can recreate a pressure and strain field. They incorporate variables such as Mach number and ply information to inform different kinds of machine learning algorithms, including a feed forward neural network. They then examine lower dimension feature spaces to determine if the machine learning algorithms can still accurately predict a pressure and strain field. The accuracy and computational load for each algorithm is examined to see how well and quickly a computational result can be replicated. This study uses only computational data to inform its machine learning algorithms, and does not attempt to examine the benefits of combining experimental and computational result.

Dazhong et al. [10] attempts to better understand chemical mechanical planarization, which is a common technique to create planar surfaces in the semiconductor industry. This process is very complicated, due to associated chemical and mechanical phenomena. Tree-based ensemble learning is used in an attempt to predict mass removal rate, which is important for a uniform surface finish. Various decision tree-based ensemble learning algorithms used experimental data collected from a chemical mechanical planarization tool. They conclude that tree based learning algorithms could be used to accurately predict the removal rate. This attempts to use experimental data to predict complex system behavior, but lacks any computational data that could be useful in augmenting the machine learning algorithm.

Atkinson [11] explores neural network based dynamic engine emission modeling. Dynamic engine behavior is very complicated, and is based on a wide variety of engine parameters, as well as the time in history. This is very cost and time prohibitive by traditional means, and neural networks were used to attempt to understand and predict the nonlinear behavior of an engine. Neural networks were seen to be relatively accurate at predicting emissions for a given engine, and could be used as a calibration tool. This study lacked computational models of engines to expand the feature space, and applications of the neural network, but did show a good example of how physics based parameters can be useful in a physics based machine learning algorithm to predict system behavior.

Brahma [16] uses artificial feature spaces to better predict diesel engine calibration information. He uses feature selection to explore what feature spaces allow for the most accurate interpolation and extrapolation of engine calibration. This was used in an effort to make what was originally extrapolation in the original feature space, interpolation in the artificial feature spaces. There was a marked increase in prediction

ability with this approach. This study uses the addition of artificial feature spaces to inform a physics based machine learning algorithm, but lacks an experimental boolean variable to further help with the combination of data. An experimental boolean variable was a one or zero used to designate if the some of a data point's features were derived from experimental or computational methods. It also looks at an engine model, not a baseball pitch system. Exploring a baseball pitch will offer another insight into the use of artificial feature spaces to augment a physics based machine learning algorithm to further show its use for complex system understanding.

These studies provide a useful framework for understanding what has been done in machine learning of complex systems. These studies usually do not combine experimental and computational data. They also lack the use of an experimental boolean variable to help with machine learning accuracy. This study aims to extend these studies by combining experimental and computational data with the help of an experimental boolean variable into a physics based machine learning algorithm.

2.3 Motivation

Understanding how complex systems work allows for control of these systems. Typically, complex systems fail to be predicted based on solely experimental data, and computer models struggle to quantitatively replicate behavior. This makes it hard for many complex systems to be better understood and predicted. If a model combining experimental and computer results could be made, a better understanding of the complex system may be possible. This study aims to combine the accuracy of experimental data with the full breadth of computational data to make a physics based machine learning algorithm to predict a complex system.

This study will extend and improve previous work in various areas. It aims to control a more complete set of baseball input variables, such as sting angle and seam orientation. This means the study will be able to model a more complete set of baseball pitch behavior. It also aims to increase the amount of baseball force data, both experimentally and computationally (CFD generated data). This data will then be expanded to include new CFD generated feature space variables to increase the data associated with each point. These extensions mean that a greater understanding of the baseball system could be seen through the greater control of the system, CFD generated data, and CFD generated features.

This data will inform a new approach to using a physics based machine learning algorithm. It will combine experimental and computational data to create a physics based machine learning algorithm, which will improve the understanding of the complex baseball system. It does this by including the CFD generated data and features to the training data, as well as the addition of an experimental boolean variable to help differentiate experimental and CFD generated data. The accuracy of these machine learning algorithms could then be applied to show that other complex systems could be better understood with the help of physics based machine learning.

Chapter 3

System Setup

3.1 Overview

In order to test a baseball pitch in a wind tunnel and control the necessary input variable, different apparatuses had to be designed and fabricated. These systems were designed and fabricated to physically hold the baseball in the wind tunnel and adjust the input variables. These physical systems were then controlled by various hardware and software to allow for system control, experimental data collection. Experimental systems were designed to control the seam orientations, sting angle, spin rate, and wind tunnel air velocity. This required multiple assemblies to be made, each having a different purpose. The first assembly made was the Ball Rotation Angle Table (BRAT), which was designed to securely hold the rest of the assembly, while varying the ball's angle of rotation. The next assembly was the Baseball Spinning Assembly (BSA), which held the baseball securely and spun it at a given rate. The last part of the test apparatus was the electronic control, which spun the baseball motor, controlled the encoder, set the air speed, and communicated with the lab computer to record data.

A computer model was also made to analyze the same baseball system via CFD analysis. The goal was to set up a model that accurately replicated the experimental testing conditions. COMSOL was chosen to accomplish this task, and a baseball model was made and run in this software.

3.2 Ball Rotation Angle Table

The BRAT was designed to adjust the ball rotation angle, and hold the rest of the assembly securely. A plate was secured to a non moving base with two hinges. The angle rotation table could then be rotated about these hinges, and thus adjust the baseball spinning assembly's angle as well. The BRAT was designed to have the minimal amount of sting length needed to place the baseball in the wind tunnel. It accomplished this by having the proper height needed to mount the BSA as close to the wind tunnel as possible, and to have the angle of rotation in the middle of the load cell. This allows the top of the BSA at zero degrees baseball angle of rotation to be the highest point. The assembly can be seen in Figure 3.1. Drawings for this table can be found in Figures A5.35, A5.36, A5.37, and A5.40.

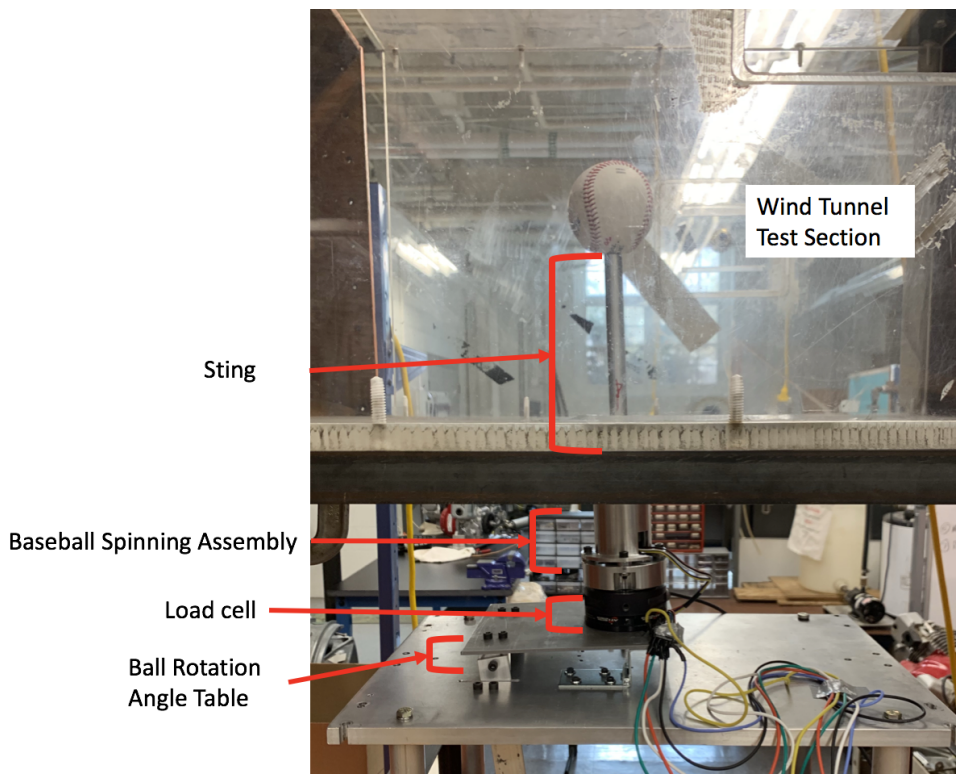


Figure 3.1: Full assembly setup for a typical experimental trial.

3.3 Baseball Spin Assembly

The BSA was made to securely spin the baseball at a specified rate. It accomplished this by coupling a motor to the baseball rotation shaft(Figure A5.31). The baseball is then held securely by a sting, which has a roller bearing at the bottom and top. This makes the potential vibration of the spinning baseball more controlled. This assembly can be seen in Figures 3.2 and 3.1. Drawings for this assembly can be found in Figures A5.41, A5.38, A5.39, and A5.34.

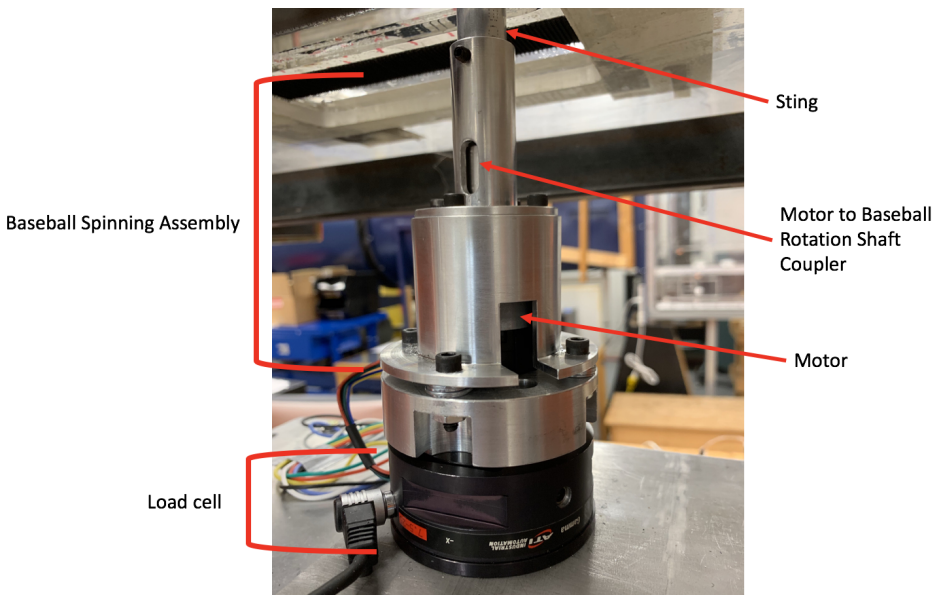


Figure 3.2: Ball Spin Assembly and load cell used to spin baseball and record forces.

3.4 Apparatus Control

All of these systems had to be controlled and data gathered. This meant a central control point had to be established to simultaneously control systems and record information. This was primarily done with a Matlab script that communicated with Arduino micro-controllers and NI-DAQ modules to accomplish an experimental trial.

This Matlab script first had to control various parts of the system, and can be seen in the Github repository [8]. It was first able to control the ball spin rate by

communicating with an Arduino via serial communication, which then controlled a motor driver board (Figure 3.3), which then drove the motor (Figure A5.32). This Arduino code can be seen in the Github repository [8]. The Matlab script also controlled the wind tunnel's wind speed by controlling an output voltage of a NI-DAQ 9263 module, which controlled the air speed of the wind tunnel.

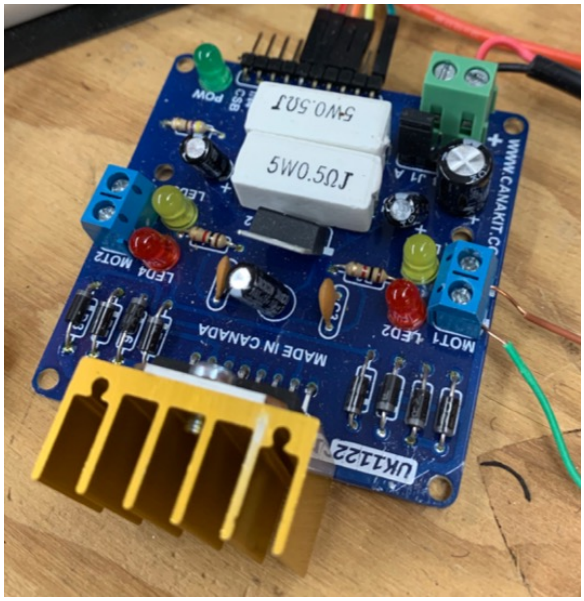


Figure 3.3: Motor driver board used to spin the baseball. Pulse width modulation used to regulate spin rate.

The Matlab script then collected data for a given trial. It communicated with an Arduino via serial communication to query the ball spin rate. This Arduino had a mounted encoder shield, which would then determine the motor speed based on the motor mounted encoder. This Arduino code can be seen in the Github repository [8]. An encoder shield was used in addition to the Arduino itself to offset some of the computing required to count pulses, which at high speeds can alias the data due to the Arduino's relatively slow clock speed.

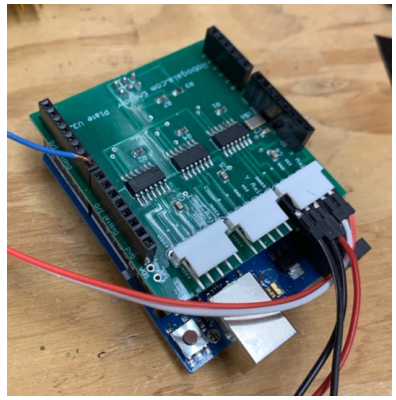


Figure 3.4: Encoder shield to measure ball spin rate.

Matlab then had two separate NI-DAQ sessions open for data acquisition (both utilizing a NI-DAQ 9215 module). The first was recording wind tunnel temperature and pressure (this pressure translated into air speed described in section 4.2.2) at 10 Hz. This slower acquisition rate was sufficient due to the steadiness of these readings at steady state. The second session was recording the load sensor's three axes of force at 2500 Hz. This high acquisition rate was necessary due to the inherent noise associated with the ball spinning, and this was the highest speed the standalone load sensor box was capable of outputting. The script also took user input for the seam orientation and ball angle of rotation, which are set up manually.

3.5 Computational Setup

The baseball system was replicated in a computational model. This aimed to mirror all conditions present in the experimental model, with COMSOL being the program used to accomplish this. A program from COMSOL was used as the basis for the computational setup [6].

Figure 3.5 shows the geometric setup of the model, where the baseball on a sting can be seen inside a volume that mimics a wind tunnel. Baseball geometry was first imported into the simulation [12], and rotating wall conditions were set to mimic the spin rate of the baseball pitch. A sting was then added to mirror the experimental setup. Relevant boundary conditions were set to replicate an inlet and outlet similar

to a wind tunnel.

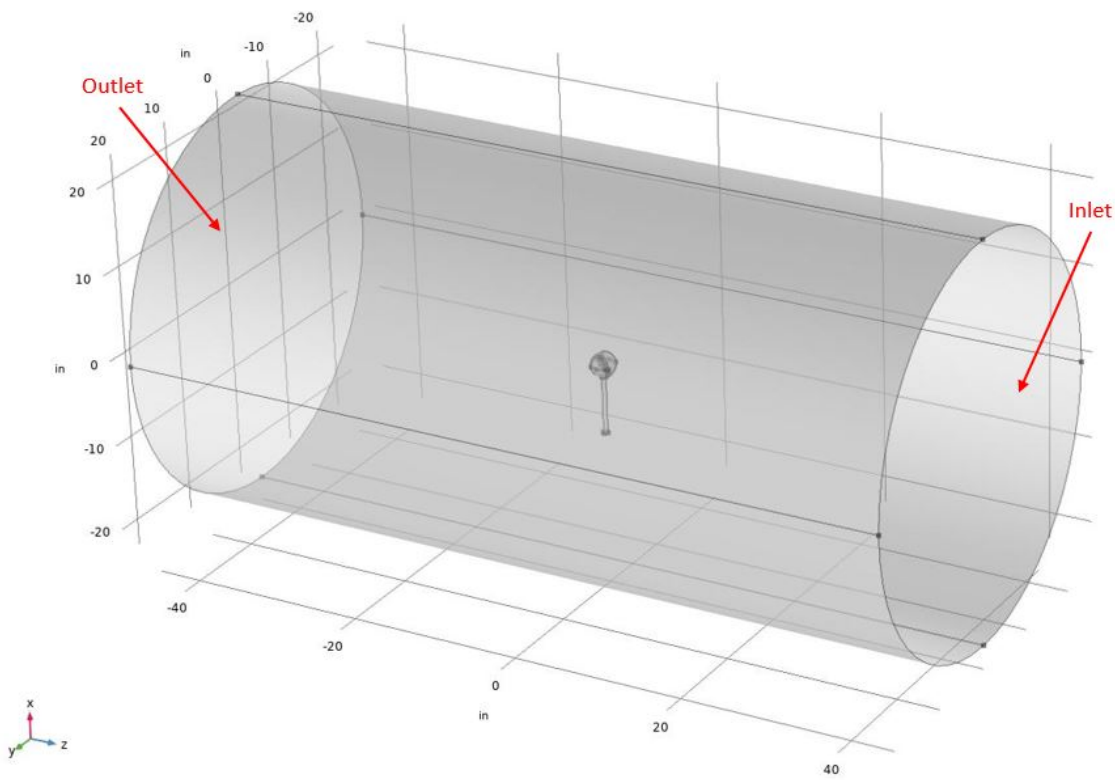
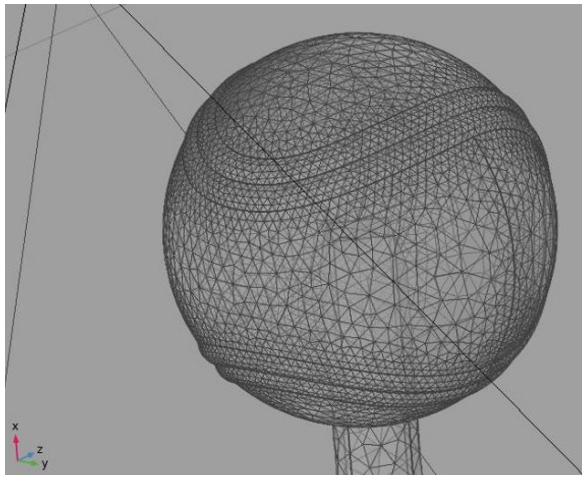
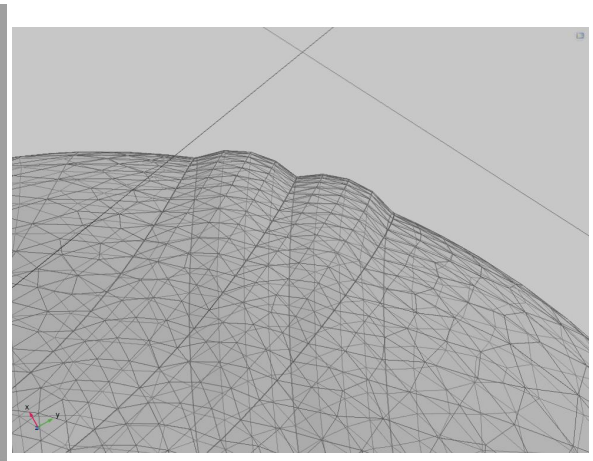


Figure 3.5: Baseball Experimental and Computational Orientation Axes

A mesh then needed to be generated for the study and can be seen in Figure 3.6. The default physics based mesh would make the study fail when attempted to be solved, and was seen to not adequately replicate the geometry of the baseball. A custom mesh for the baseball surface was then made to more closely replicate the geometry of the baseball and allow the program to solve properly. Figure 3.6b shows how the mesh replicates the raises along the baseball seams. The individual laces were not able to be made with the available computing power. The entire baseball can be seen to be closely replicated with a mesh in Figure 3.6a.



(a) Full Baseball



(b) Stitch close-up

Figure 3.6: Computational mesh.

A parametric study was necessary in order to change the input variables controlled in the experimental setup. These six input variables (seam orientation: y axis (θ_y), seam orientation: z axis (θ_z), sting angle (θ_b), spin rate (ω_b), and air velocity (v), experimental boolean designation) were put into the parametric study, and were varied accordingly. This allowed the experimental data sets to be replicated in this simulation, as well as extending the experimental data by putting in a larger range of input variables into the computer simulation.

3.6 Physics Based Machine Learning Setup

Neural networks (NN) are a type or machine learning algorithm that are useful for understanding non-linear systems. They work to estimate the output of the system by using weights and biases within the neural network. Each input has an associated weight, which is trained to produce the most accurate prediction of the output. Figure 3.7 shows the sequential hidden layers of neurons that eventually produce the output layer. These hidden layers serve to allow the neural network to predict more complicated behavior, such as behavior seen in a complex system.

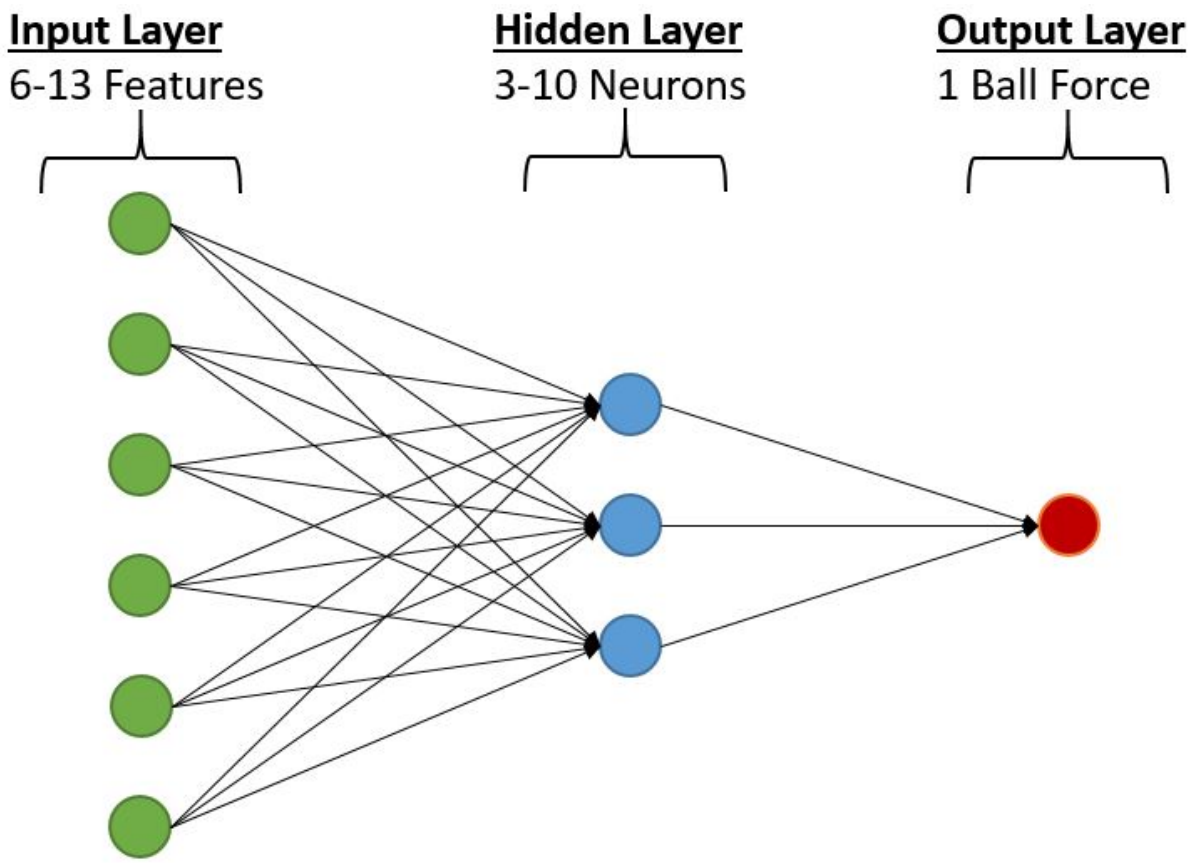


Figure 3.7: Neural Network layer setup [10]. This study's neural network has 6 - 13 inputs, 1 hidden layer with 3 - 10 neurons, and 1 output. Each ball force was trained individually.

These neurons use an "activation function" [13], which allows the network to learn. In this study, a sigmoid function is used as the activation function, and can be seen in equation 3.1. The non-linearity of this equation allows the neural network to incorporate non-linearities seen in the training data. This means that if the baseball pitch behavior is seen to be non-linear, it can be more closely predicted with this activation function.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

To train a neural network and test its predictive ability, the available data must

be split up into testing and training data. The training data is the data that will make up the input layer for the NN, and is what is used to actually train the NN. The testing data is what is used to determine how well the trained NN can predict outputs. It is important for this training and testing data to be kept separate, so that the neural network is not training itself with its testing data.

In this study, a machine learning algorithm is used to attempt to combine experimental and computational data to create a more accurate predictive algorithm of a complex system. In this study, a neural network was used to combine both wind tunnel and CFD in a physics based machine learning algorithm (PBMLA). This attempted to improve the accuracy compared to just experimental or just computational data to predict the baseball system. The NN by default uses the experimental data inputs taken from the experimental trials, and attempts to predict the ball forces based on just that information.

There were two ways to combine experimental and computational data in this NN. Figure 3.8 shows the various setups of the NN. The default NN setup was seen in the top left quadrant in Figure 3.8, which shows just experimental data informing the NN. Adding in computational data points to the training data increased the amount of training data points, and can be seen in the top right box in Figure 3.8. This means that the number of training data points has increased because both the experimental and computational data points are in the training data set.

The experimental boolean designation helps to differentiate whether a data point was derived from experimental or computational trials. If the original five input variables (seam orientation: Y axis (θ_y), seam orientation: Z axis (θ_z), sting angle (θ_b), spin rate (ω_b), and air velocity (v)) were from an experimental trial, the experimental boolean would be 1, conversely a computational trial would be a 0. This was used in an attempt to differentiate kinds of data points since the two methods might predict quantitatively different results (but qualitatively similar).

Another option was to enlarge the feature space of the experimental data. To first understand this, the different features spaces used must be explained. A feature space is the different variables used in the input of a NN. Table 3.1 shows the different feature spaces used in this study. Feature space 1 corresponds to variables able to be recorded in the experimental setup. Feature space 2-4 are combinations of the experimental input variables and input variables only able to be attained via the computer simulations run for the baseball system. Feature space 5 has only CFD generated features, and lacks any experimental features.

Increasing the feature space size with computational features means that each experimental data point has more associated information. This option corresponds to the bottom left box in Figure 3.8. In this case, each experimental data point now has additional computational features, such as computational side force.

Feature Space 1	**Feature Space 2**	**Feature Space 3**	**Feature Space 4**	**Feature Space 5**
Seam Orientation-Y axis (θ_y)	Seam Orientation-Y axis (θ_y)	Seam Orientation-Y axis (θ_y)	Seam Orientation: Y axis (θ_y)	
Seam Orientation: Z axis (θ_z)	Seam Orientation: Z axis (θ_z)	Seam Orientation: Z axis (θ_z)	Seam Orientation: Z axis (θ_z)	
Air Speed (v)	Air Speed (v)	Air Speed (v)	Air Speed (v)	
Sting Angle (θ_b)	Sting Angle (θ_b)	Sting Angle (θ_b)	Sting Angle (θ_b)	
Spin Rate (ω_b)	Spin Rate (ω_b)	Spin Rate (ω_b)	Spin Rate (ω_b)	
Experimental Boolean Designation	Experimental Boolean Designation	Experimental Boolean Designation	Experimental Boolean Designation	Experimental Boolean Designation
	Velocity Right	Velocity Right		Velocity Right
	Velocity Left	Velocity Left		Velocity Left
	Pressure Right	Pressure Right		Pressure Right
	Pressure Left	Pressure Left		Pressure Left
		Computational Side Force (F_s)	Computational Side Force (F_s)	Computational Side Force (F_s)
		Computational Drag Force (F_d)	Computational Drag Force (F_d)	Computational Drag Force (F_d)
		Computational Lift Force (F_l)	Computational Lift Force (F_l)	Computational Lift Force (F_l)

Table 3.1: Feature spaces for used for machine learning algorithm.

Both of these techniques can also be combined, to make the NN have both more input data points, as well as having more associated data with each data point. This option corresponds to the bottom right box in Figure 3.8. This option give the NN the most information to work with, and may be seen to be the best option for predicting this complex system.

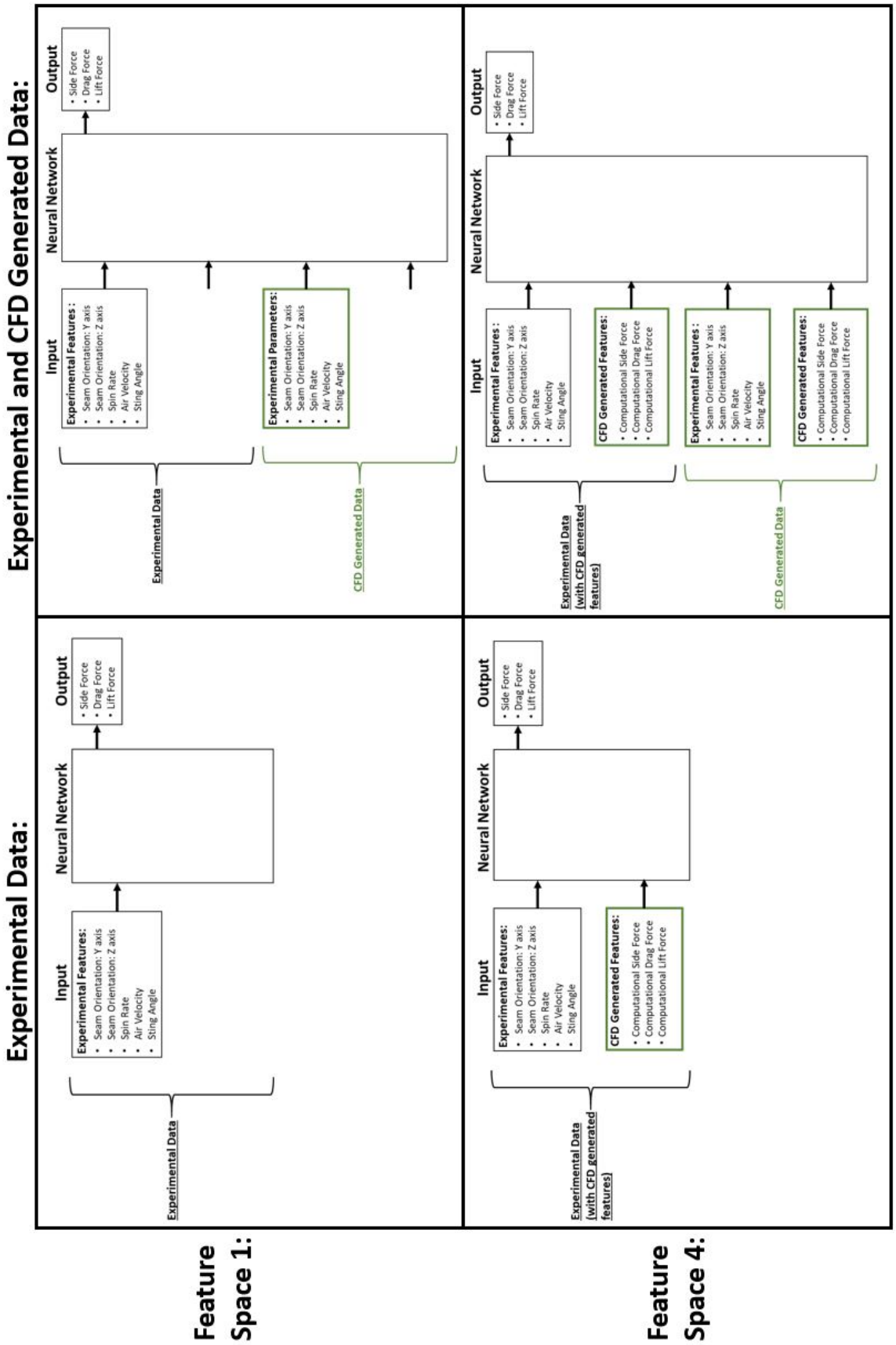


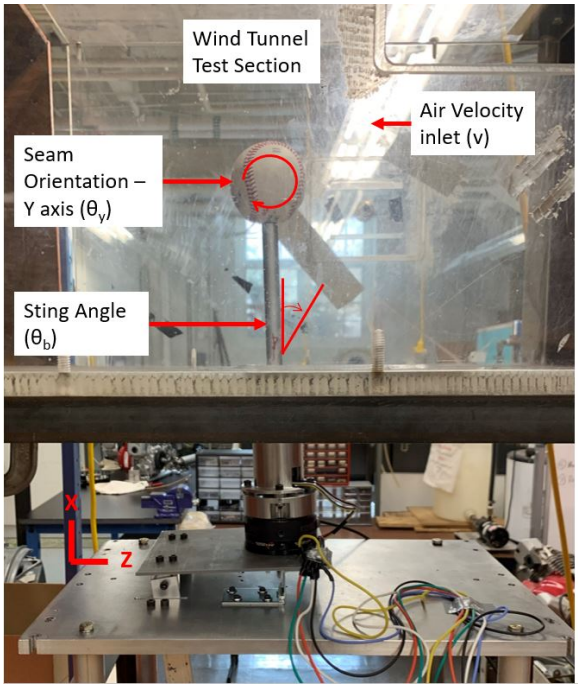
Figure 3.8: Physics based machine learning algorithm potential setups. Feature space variables can be found in table 3.1.

Chapter 4

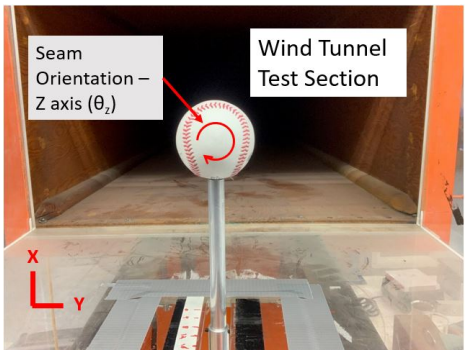
Testing Methodology

4.1 Experimental Testing Plan

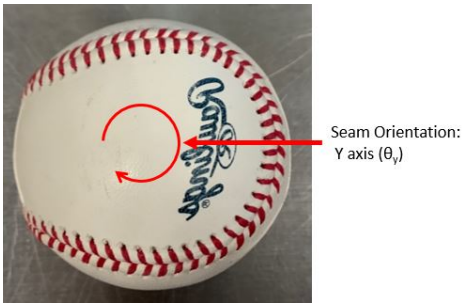
The goal of the trials was to vary all of the five potential variables (seam orientation: Y axis (θ_y), seam orientation: Z axis (θ_z), sting angle (θ_b), spin rate (ω_b), and air velocity (v)) to understand a wide variety of baseball behavior. These variables are shown in Figure 4.1.



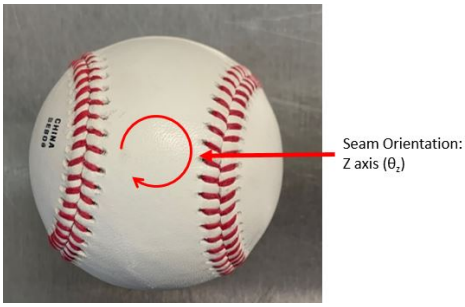
(a) Side View



(b) Front View



(c) Seam Orientation: Y axis



(d) Seam Orientation: Z axis

Figure 4.1: Setup variable visualization. The baseball spin about the sting. Seam orientation changed by using multiple baseballs with sting mounted to a different part of the baseball.

For each trial, a baseball was first mounted on a 3/16" shaft to set both its seam orientation axes; multiple baseballs were used and mounted in varying orientations to accomplish this. Then, the baseball was mounted on the testing apparatus, and coupled to the motor (Figure A5.38). This leaves the other 3 input variables to be changed in any given trial. For a typical trial, the sting angle, spin rate, and air velocity were all changed to record a wide range of baseball behavior.

Variable Name	**Low Range**	**High Range**	**Increment**
Air velocity (v)	50 [mph]	100 [mph]	10 [mph]
Ball spin rate (ω_b)	3700 [rpm]	6500 [rpm]	500 [rpm]
Seam orientation: Y axis (θ_y)	0 [deg]	90 [deg]	N/A
Seam orientation: Z axis (θ_z)	0 [deg]	30 [deg]	N/A
Sting angle (θ_b)	0 [deg]	24 [deg]	12 [deg]

Table 4.1: Input variables and ranges controlled in experimental data.

Table 4.1 shows the variables controlled in the experimental trials, and their respective ranges. These variables were all varied in ranges that aim to replicate the system as fully as possible, while also being safe and operating within the physical limitations of the equipment. This meant the baseball sting angle can be varied from 0 - 24 degrees. This limitation was due to a variety of factors including the slot size able to be used in the wind tunnel and the angle that the BRAT is able to be set at. The ball spin rate range is 3500 - 6500 RPM, and is spun in the positive direction based based on the right hand rule. The lower range was set due to a natural frequency being observed at lower spin rates (which shook the table and ball), as well as the motor's inability to hold a consistent spin rate at lower speeds. The higher spin rate is due to increased vibration at high spin rate, which could overload the force sensor. The air velocity could be set between 50 - 100 mph. This range is based on the capabilities of the wind tunnel motor itself. The symmetry of a baseball meant that the seam orientations of the ball could be limited to a smaller range.

For a trial to begin, the Matlab script that controls the input variables and records data was started. This established two serial connections to Arduino micro-controllers, one to the baseball motor control to control spin rate and the other to an encoder to measure spin rate. It then established three NI-DAQ sessions: One to record three axes of force, one for wind tunnel temperature and pressure, and one to control the wind tunnel air speed. The wind tunnel was then set to a selected air velocity and allowed to attain steady state by waiting for roughly ten minutes, which was a sufficient time for wind tunnel temperature and air velocity to become steady. The baseball was then set at a specified sting angle by rotating the BRAT. The angle is known by referencing a inclinometer and pre-marked angles on the wind tunnel lower slot. The ball was then spun up to a specified speed (typically 3500-6500 RPM) and swept through all the wanted spin rates. This means that for a typical trial the baseball will first have its wind speed set, then sting angle, then spin rate. This order was chosen due to the increasing difficulty and time associated with changing the ball spin rate, sting angle, and wind speed respectively.

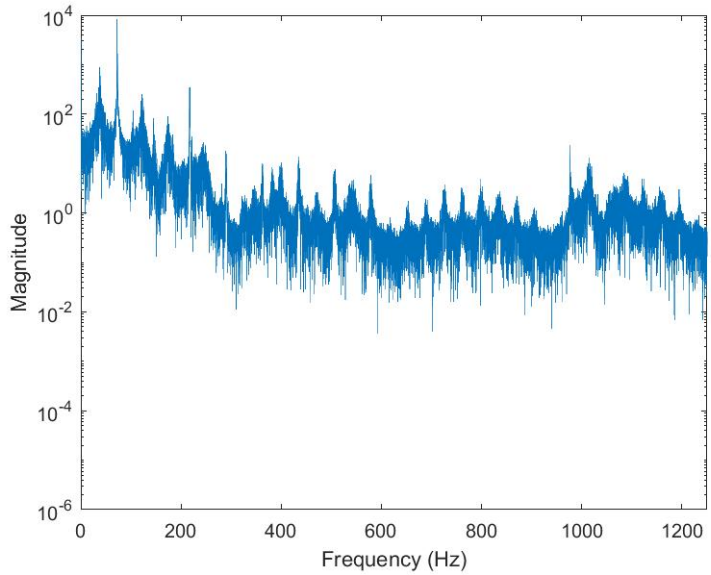
When the baseball variables were set, and data was ready to be taken, there was a set procedure to gather data. First, the air pressure and temperature were recorded at 10 Hz for 5 seconds. The pressure and temperature were not seen to change once steady state (SS) was achieved, so a lower frequency was adequate. Then, the ball spin rate was taken. The three axes of force data was then taken at 2500 Hz for 20 seconds. This high frequency and period of data collection was necessary due to the high amount of high frequency noise associated with the force data. The ball spin rate was then taken again to ensure constant spin rate, which was in fact seen.

4.2 Experimental Post Processing

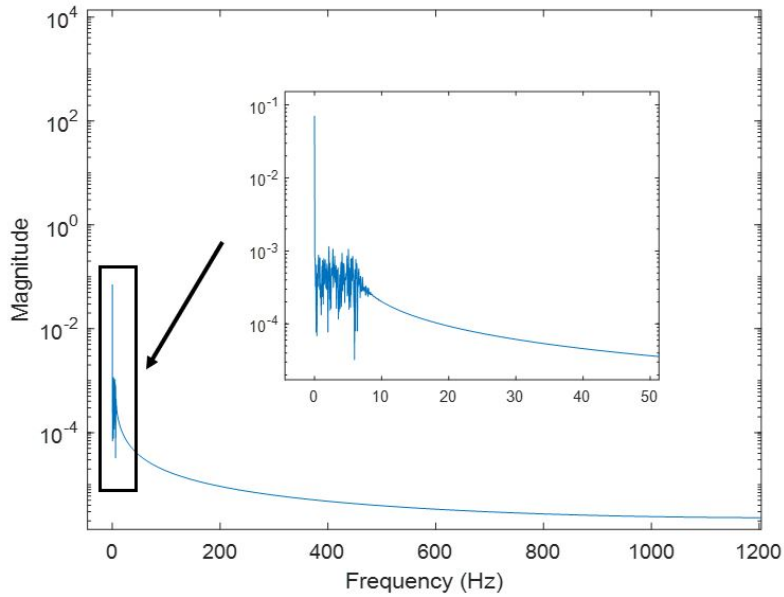
After obtaining data, it was post-processed into a usable form. The wind tunnel pressure and temperature were averaged. The three forces were filtered to get a single force value for each axis of force (section 4.2.1). The relevant data gathered for each trial included seam orientations, sting angle, ball spin rate, air velocity, air temperature, side force, drag force, lift force, ball coefficients, and metadata. This data organization can be see on Table A5.5.

4.2.1 Filtering and Calibration

Filtering was first done on force data. This was necessary due to the high amount of noise introduced at the frequency of the baseball spin rate, making signal to noise ratio (SNR) of roughly 0.017. Figure 4.2a shows the FFT of a specific trial's unfiltered data. The largest spike occurred near the spin rate (roughly 95 Hz for this trial), and could be seen to be much larger than the 0 Hz forcing. This meant that the ball vibration had a significant impact on the overall forcing recorded. To alleviate this and make the 0 Hz forcing the predominant signal, a Butterworth filter was applied with a cutoff frequency of 12.5 Hz. This allowed for a large amount of high frequency attenuation, while allowing for steady forcing to remain unaffected.



(a) Unfiltered FFT

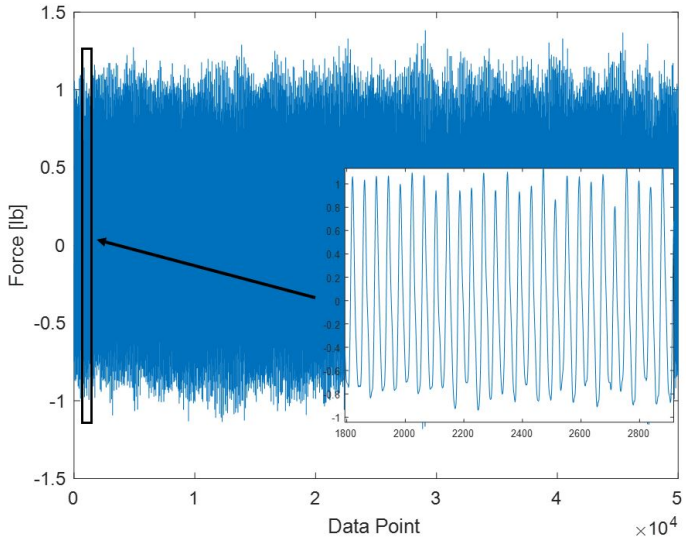


(b) Filtered FFT

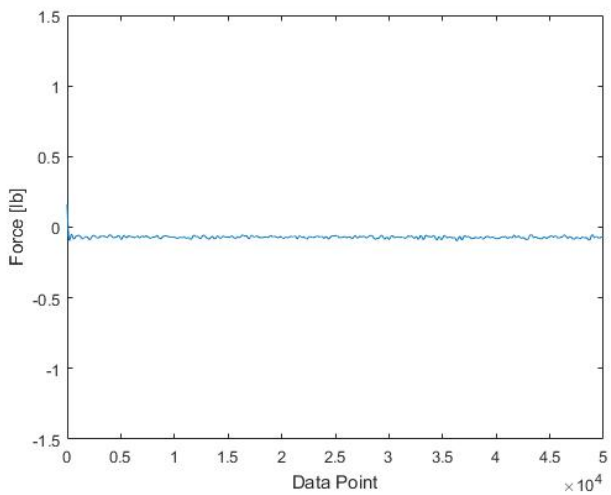
Figure 4.2: Example side force raw data FFT before and after low pass filter.

Figure 4.2 shows the unfiltered and filtered FFT for a typical data set, with an increase in 0 Hz forcing and attenuation of higher frequency forcing post filtering.

The associated unfiltered and filtered data in Figure 4.3 showed the effect filtering had on the forcing itself. Cyclical noise at roughly the ball spin rate can be seen in Figure 4.3, and was very large compared to 0 Hz forcing. Visually, the unfiltered data looked much noisier than the unfiltered data. This translated into a signal to noise ratio of the unfiltered and filtered data of 0.017 and 53 respectively. This showed a clear attenuation of higher frequency noises, and an increase in the steady state forcing of the ball after filtering. The steady state forcing was also equal to the mean of the data as whole. To get a singular forcing value, the 0 Hz forcing was obtained for all axes of force.



(a) Unfiltered Data



(b) Filtered Data

Figure 4.3: Example side force raw data and data after low pass filter applied.

Examining the validity of this filtering, the 0 Hz frequency force was examined to see if it was affected. The ratio of unfiltered to filtered 0 Hz forcing was 1:1, indicating that the filter did not change the underlying 0 Hz force measured. The Butterworth filter was also examined to see its frequency response in Figure 4.4. This showed that the filter would not change the 0 Hz forcing, but significantly attenuates the higher frequency forcing. Table 4.2 show the attenuation of the ball spin forcing for the

range of ball spin rates.

Ball Spin Rate [rpm]	**Butterworth Attenuation**
3500	1.4 E-6
4000	6.3 E-7
5000	1.8 E-7
6000	5.6 E-8

Table 4.2: Butterworth attenuation of baseball forcing at selected spin rates.

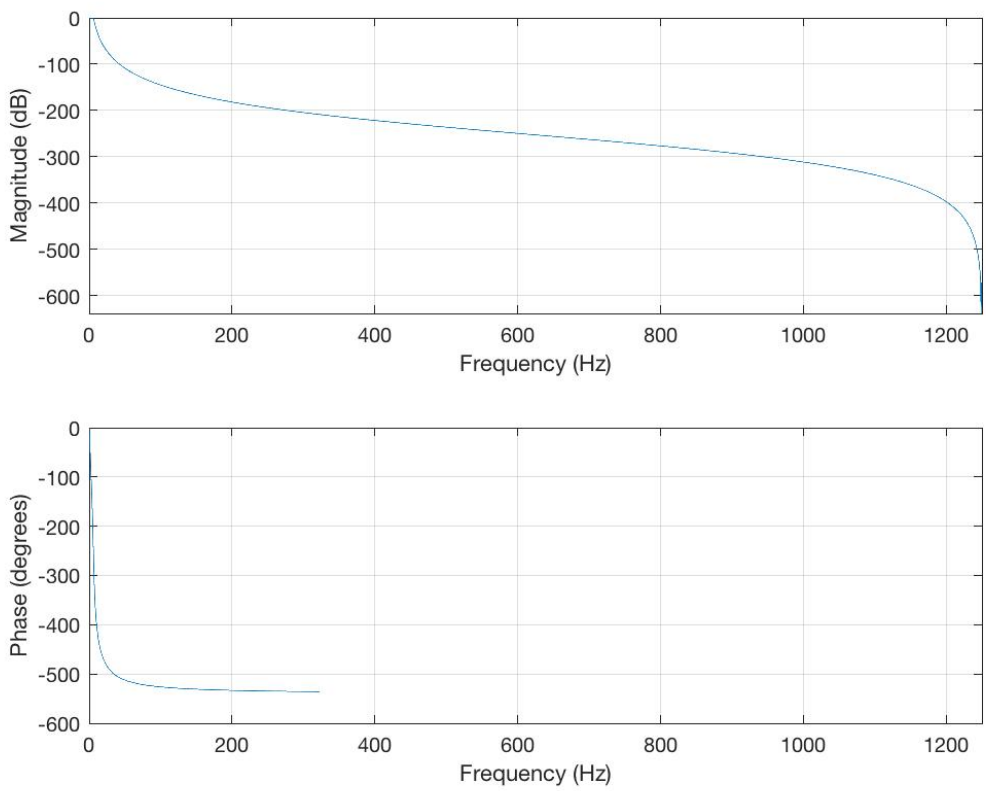


Figure 4.4: Frequency response for Butterworth filter used on force data. Note 20 dB is equal to an order of magnitude of attenuation.

The data was then calibrated to take out the gravitational effects of the apparatus, and the aerodynamics of the sting itself. First, the weight of the apparatus was taken

out of each force data point, followed by the aerodynamic forcing of just the sting. This left primarily the aerodynamics of the baseball as the contributing factor to the force in all axes.

4.2.2 Air Velocity

The air velocity was calculated by Bernoulli's Principle. This principle relates a fluid's velocity and pressure and can be seen in equation 4.1

$$(\frac{1}{2})\rho v_1^2 + \rho g h_1 + P_1 = (\frac{1}{2})\rho v_2^2 + \rho g h_2 + P_2 \tag{4.1}$$

where ρ is density, v is velocity, g is gravity, h is height, and P is pressure. First, the height term can be disregarded, since there is no height change in the wind tunnel for our measurement. Then the relationship between two special points in the wind tunnel, the testing cross section and the convergence zone can be examined. The testing cross section is assumed to be at atmospheric pressure, and the convergence zone is assumed to have zero velocity. The equation can then be used to get the air velocity of the test section if the pressure of the convergence zone is known. Since that pressure is measured, the air velocity for the baseball can then be calculated. It should be noted that a pitot tube mounted in the testing cross section was used to get a correction factor for the pressure reading attained under normal trial data acquisition, and is also incorporated into the air velocity calculation [15]. This correction factor was necessary due to the contraction ratio of the wind tunnel.

4.2.3 Lift and Drag

To calculate the lift and drag on the baseball, the raw forces on the load cell had to manipulated since the load cell rotated with the sting angle. The drag (F_d) and lift (F_l) were calculated in equation 4.2 and 4.3

$$F_d = -z\cos(\theta_b) - x\sin(\theta_b) \tag{4.2}$$

$$F_d = -z\sin(\theta_b) + x\sin(\theta_b) \tag{4.3}$$

where z is z-direction load cell force, x is x-direction load cell force, and θ_b is the ball angle of rotation.

4.2.4 Force Coefficients

Force coefficients are then calculated to look at a dimensionless representation of the baseball forces. The force coefficients calculated include the side coefficient(C_s), drag coefficient(C_d), and lift coefficient(C_l). All these force coefficients were calculated in similar manners, with the only difference being which force was used. These were calculated in equations 4.4, 4.5, and 4.6

$$C_s = \frac{2F_s}{\rho v^2 A} \tag{4.4}$$

$$C_d = \frac{2F_d}{\rho v^2 A} \tag{4.5}$$

$$C_l = \frac{2F_l}{\rho v^2 A} \tag{4.6}$$

where F_s is side force, F_d is drag force, F_l is lift force, and A is corresponding area of object.

Chapter 5

Results And Discussion

This chapter presents the results obtained through the experimental and computational avenues explained in previous chapters. Discussion and analysis of the data is also provided.

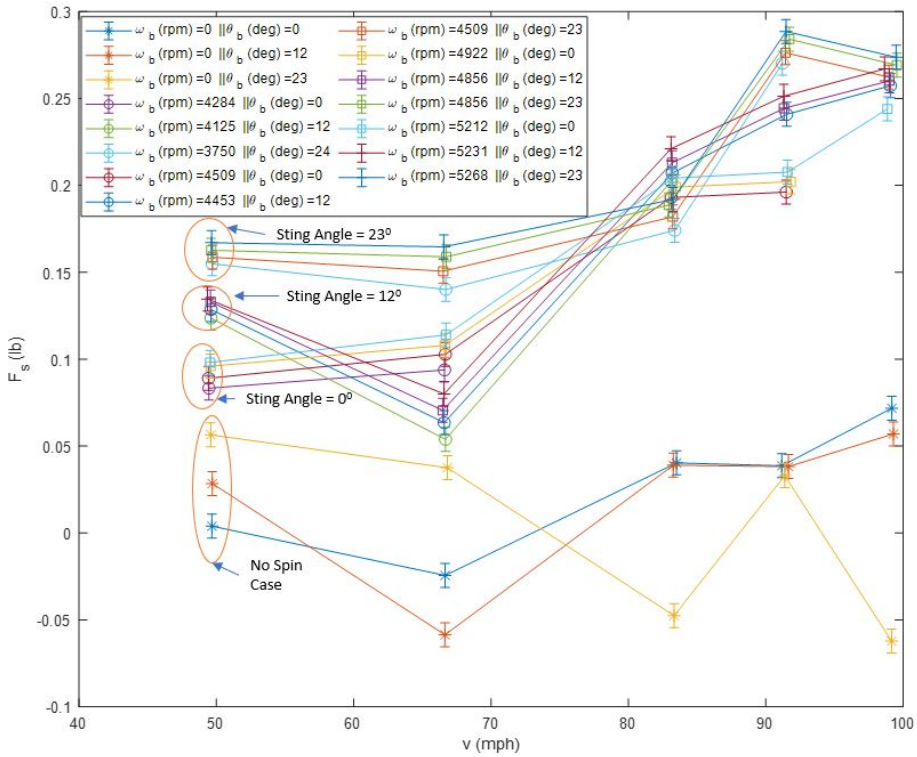
5.1 Experimental Results

Experimental data was gathered using the baseball apparatus outlined in previous sections. Force data was collected along all three axes (side force (F_s), drag force (F_d), and lift force (F_l)) for a wide variety of input variables: seam orientation: Y axis (θ_y), seam orientation: Z axis (θ_z), sting angle (θ_b), spin rate (ω_b), and air velocity (v). Since there are five dimensions of input variables, and three of baseball forces, it was necessary to perform a sensitivity analysis. This was accomplished by holding all input variables constant except one, and examining just that one input variable's effect on a baseball force.

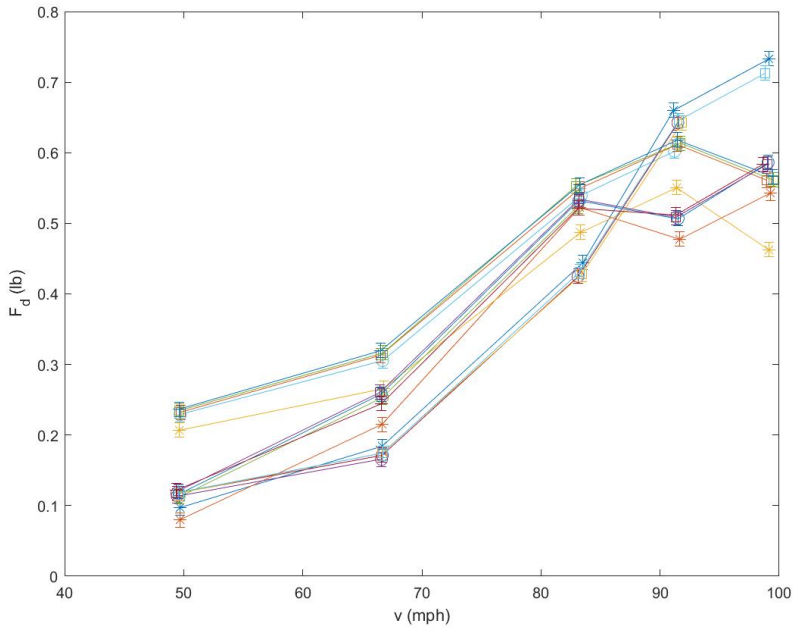
Examining the air velocity's effect on baseball forces produced clear trends and was seen to be one of the most important determining factors for ball behavior. Figure 5.1 shows the effect air velocity had on ball forces when other variables are held constant. As expected, all forces appeared to increase as the air velocity is increased. The increase from 50 to 100 mph showed the most significant change to ball forces out of any other input variable, with side force increasing 50% - 150% , drag force

200% - 650%, and lift force 100% - 300%.

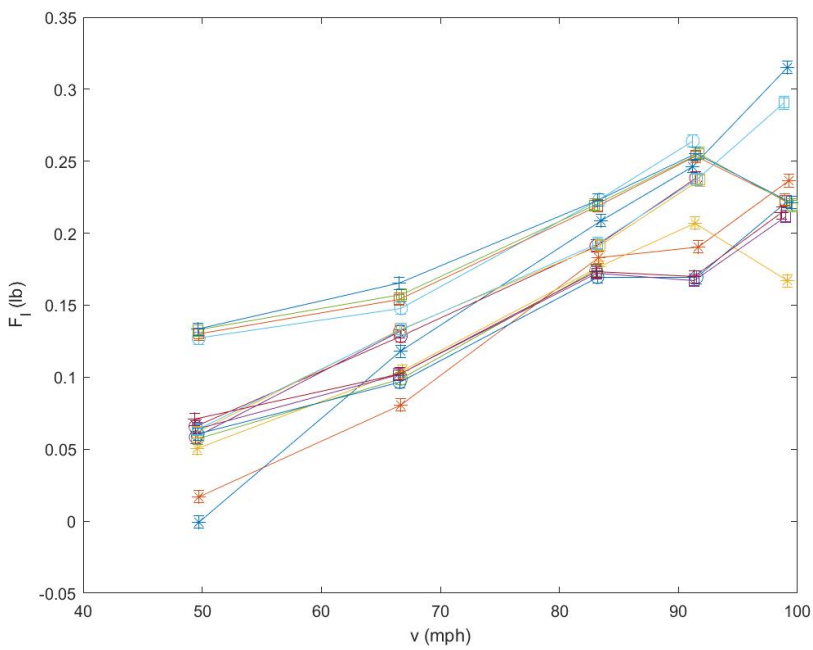
The different curves corresponded to different spin rates. There appeared to be four groupings of trend lines, corresponding to the three different sting angles and a zero spin case. These groups are highlighted by circles in Figure 5.1a. The different sting angles produced different forces for the same air velocity. The no spin case produced a unique trend. This points to the sting angle, spin rate, and air velocity having interdependent effect on all forces.



(a) Side Force (F_s)



(b) Drag Force (F_d)



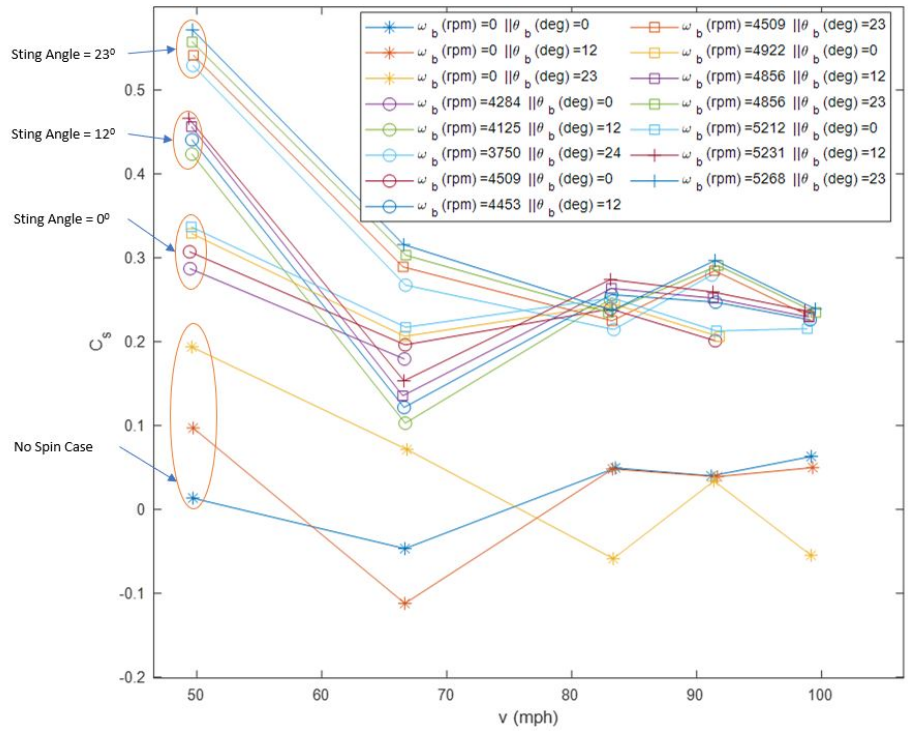
(c) Lift Force (F_l)

Figure 5.1: Air velocity, v, effect on baseball forces. Variables held constant are $\theta_y = 90$ and $\theta_z = 0$. Legends are consistent for all sub-figures. Error bars are for random bias.

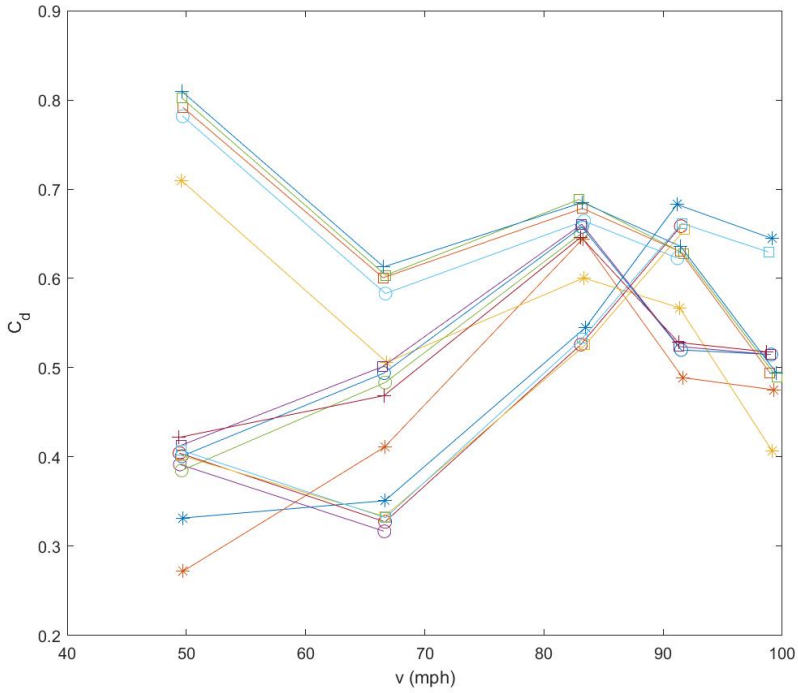
Examining the validity of the data in figure 5.1, the random bias can be examined. The random bias would impact how valid the data trends are. The sytematic bias would shift the entire force data up or down. For this reason, only the random bias is examined here to look at the validity of the data trends. More information on this can be seen in section 5.6. The side force bias was seen to be relativly small comared to the chamge in side force, this same magnitude of bias can also be seen for the drag and lift force. This indicated that the data trends could be replicated, and can be considered valid.

Figure 5.2 was identical to Figure 5.1, except it shows the ball coefficients instead of forces. Substantial variation with increasing air velocity was observed. These variations were significant because the random uncertainties are small compared to the force values recorded. These trends are highly non-linear and non-monotonic, indicative of a complex behavior.

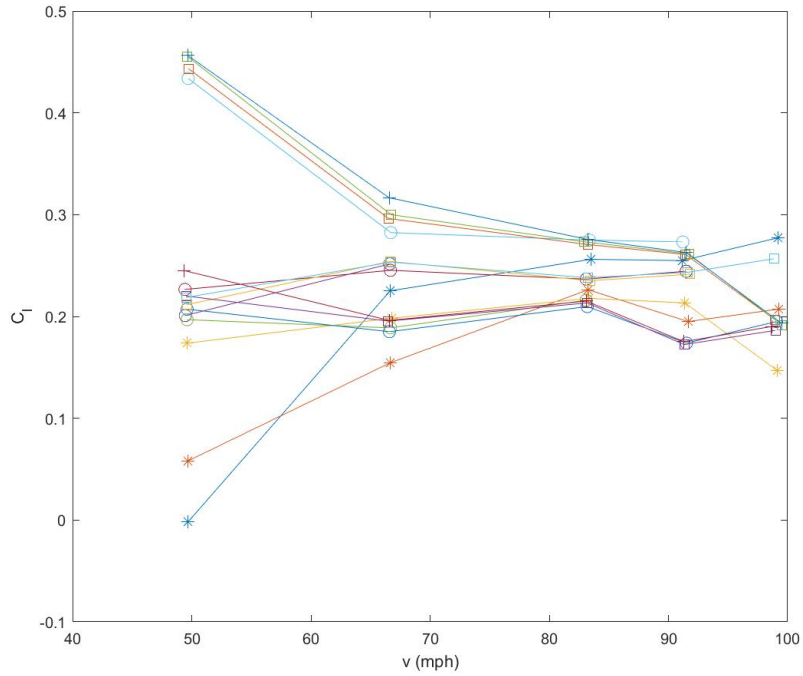
Looking at Figure 5.2, the effect that air velocity has on the ball coefficients can be examined. The side coefficients appeared to decrease, then level out. The drag coefficient appeared to go up and down, and seemed to have a nonlinear correlation to air velocity. The lift coefficient appears to be similar in some respects to the side coefficient. For a sting angle of 23 degrees, the lift coefficient decrease then levels out. For sting angles of 0 and 12 degrees, it appears to hold a steady lift coefficient value (Figure 5.2c).



(a) Side Coefficient (C_s)



(b) Drag Coefficient (C_d)



(c) Lift Coefficient (C_l)

Figure 5.2: Air velocity, v, effect on baseball coefficients. Variables held constant are θ_y = 90 and θ_z = 0. Legends are consistent for all sub-figures.

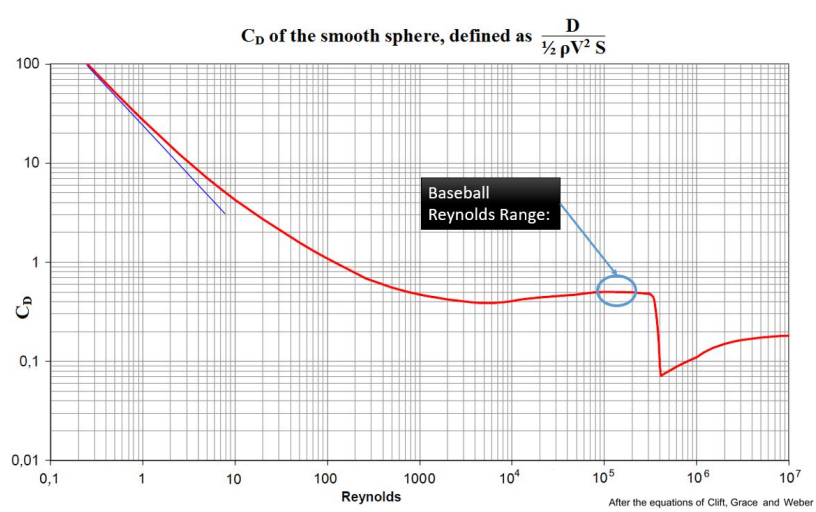
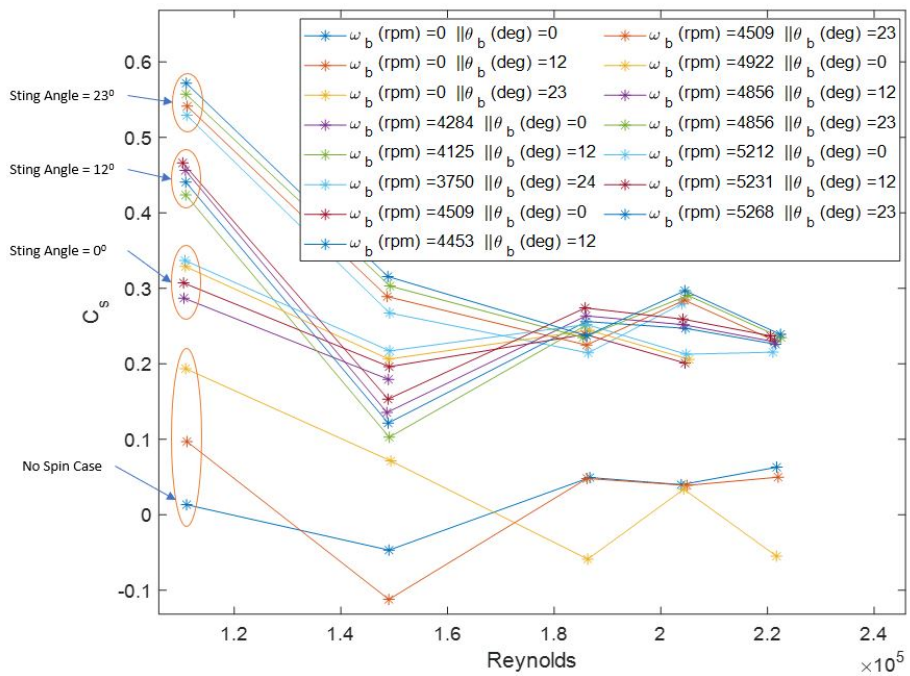
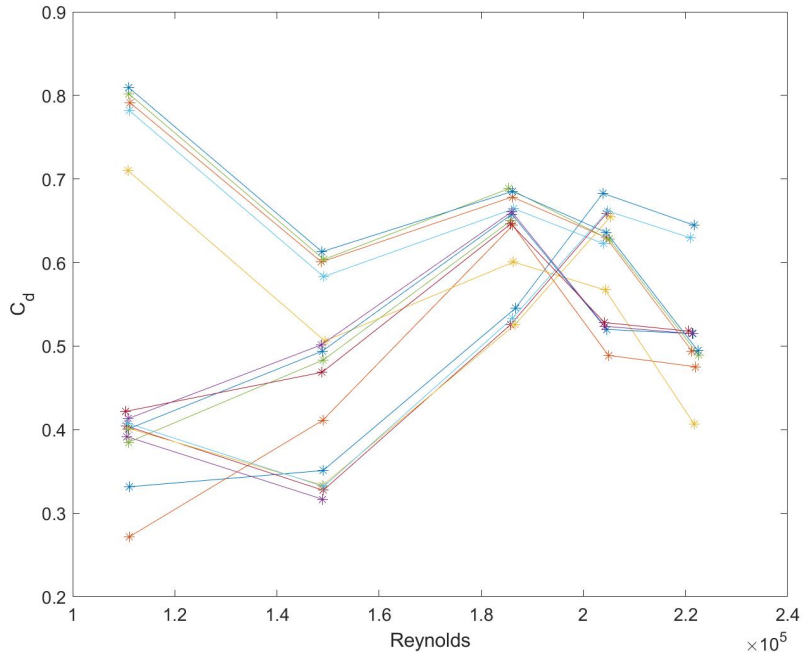


Figure 5.3: Smooth sphere Drag Coefficient for given Reynolds numbers [7].

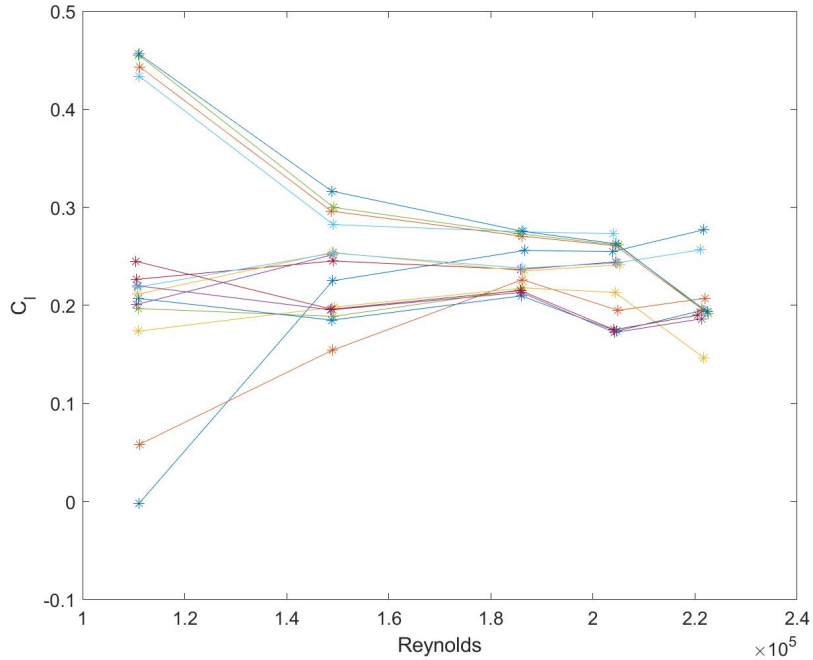
Extending the observations that velocity had on ball behavior, ball coefficients can be compared to Reynolds number. Figure 5.3 shows a **smooth** sphere's drag coefficient for a wide range of Reynolds number, with the range of Reynolds attained in experimental data highlighted in the figure, which was roughly 110,000 - 220,000. Figure 5.4 shows the baseball coefficients for the range of Reynolds numbers for experimental data. This showed that the side and lift coefficients converge at higher Reynolds numbers, and that drag coefficient appeared to increase as Reynolds number increased. The coefficients also appeared to have different behavior based on the sting angle and zero spin case. The drag coefficient increase is seen to be contrary to the trend seen for a smooth sphere in Figure 5.3, pointing to the baseball geometry and spin fundamentally changing the drag behavior of a baseball as compared to a typical smooth sphere.



(a) Side Coefficient (C_s)



(b) Drag Coefficient (C_d)

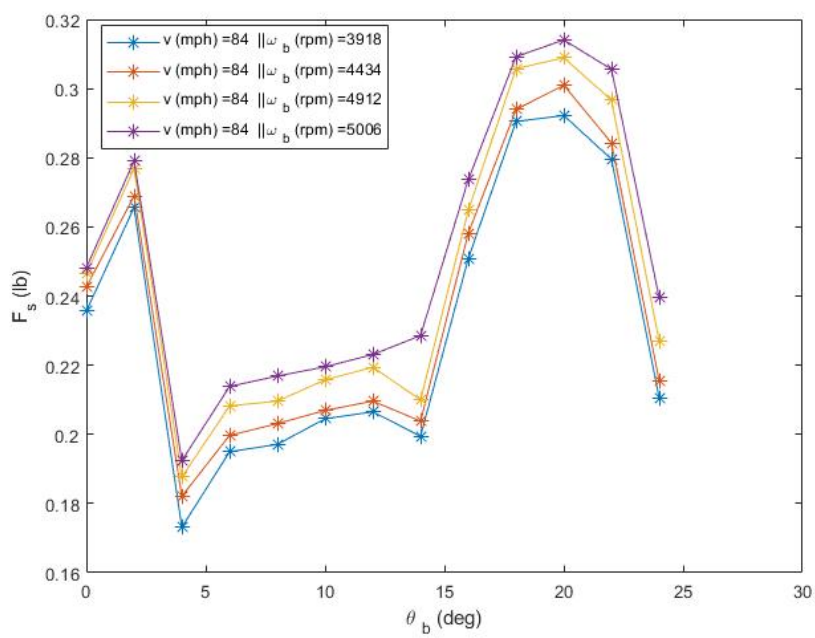


(c) Lift Coefficient (C_l)

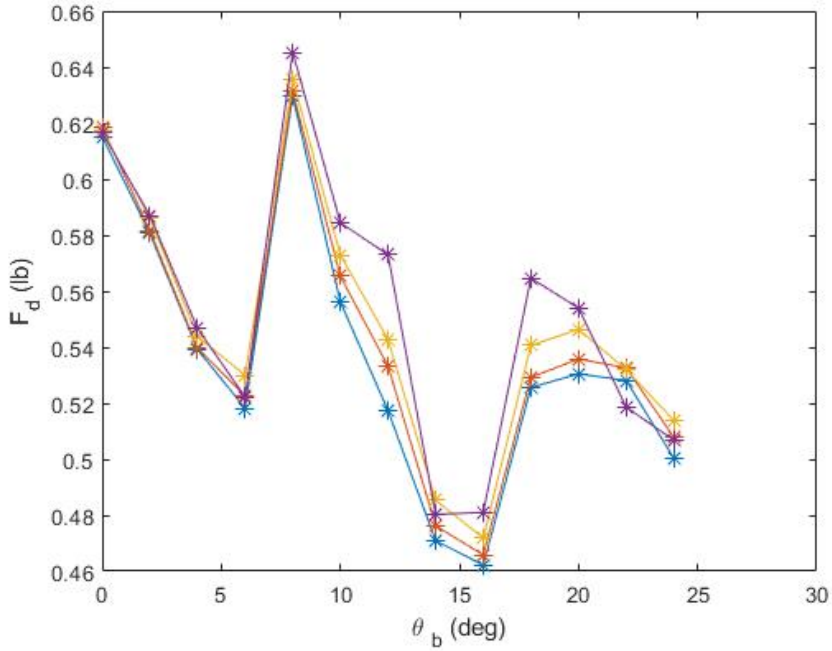
Figure 5.4: Reynolds number effect on baseball coefficients. Variables held constant are $\theta_y = 90$ and $\theta_z = 0$. Legends are consistent for all sub-figures.

The sting angle, θ_b, was seen to be another important variable influencing ball behavior. Figure 5.5 shows a sensitivity study done to examine the effect sting angle had on ball forces for various spin rates. Side force seemed to have a non-linear relationship with sting angle and can be seen in Figure 5.5a. Additionally, drag force and lift force also had a non-linear relationship to sting angle, and can be seen in figures 5.5b and 5.5c respectively.

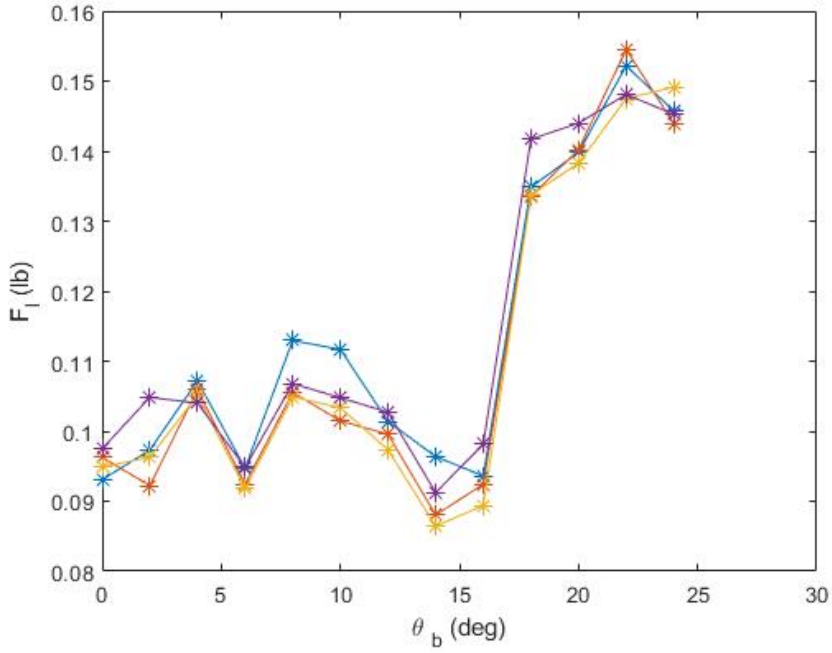
Due to the uncertainty associated with this data, the lift force could be considered statistically inconsequential if all sources of error were taken into account. However, the main part of this uncertainty was with the systematic error, which would shift the entire curve of data up or down. The systematic error was 0.1125 , 0.1125, and 0.375 [lb] for side force, drag force, and lift force respectively. The random error is more important to consider for the data trend itself, and is much smaller than systematic error. This random error was 0.0069, 0.0102, and 0.0043 [lb] for side force, drag force, and lift force respectively. This meant that the absolute force could be called into question, due to the high amount of systematic uncertainty, but the trend is more significant since the random uncertainty is small compared to the magnitude of force.



(a) Side Force



(b) Drag Force



(c) Lift Force

Figure 5.5: Effect of sting angle, θ_b, on baseball forces. Variables held constant $\theta_y = 90$ [deg], $\theta_z = 0$ [deg], v = 84 [mph]. Legends are consistent for all sub-figures. Ball coefficients follow same trend as respective ball forces.

The effect that ball spin rate, ω_b, had on forces can be seen to be a less important factor for ball behavior. Figure 5.6 shows the effect ball spin rate has on side force when other conditions are held constant. This data set is separate from other data, since the sting angle was changed with high resolution, and other input variables changed with less resolution. Side force increases as spin rate increases, with a ball spin rate change from 4000 to 6500 rpm translating to a side force increase of 10% - 20%. This is a significant amount of change, but small when compared to the effect a large change in velocity or sting angle had on force. The drag and lift do not appear to be sensitive to ball spin rate, and do not change as spin increases (Figure A5.30).

Examining a zero spin case in Figure 5.6, it can be seen that zero spin rate results in almost no side force. The small range of side force was due to not being able to hold the baseball seam exactly at a symmetrical position, resulting in this small amount of side force, this side force would be analogous to the side force seen in a baseball pitch with negligible spin, such as as screwball. This figure shows that the introduction of spin is a defining characteristic of side force. Looking closer at the amount of side force at a given spin rate, it appears that if the trend line of force is extended to a zero spin rate case, the side force would still be non-zero. Since the actual amount of side force is seen to be essentially zero at zero spin rate, this points to a break in the observed trend for lower spin rates. No experimental data is known for these lower spin rates, but a possibility to explain this contradiction is that there is a certain spin rate at which a jump in side force is developed or that a different relationship between spin rate and side force is developed at lower spin rates.

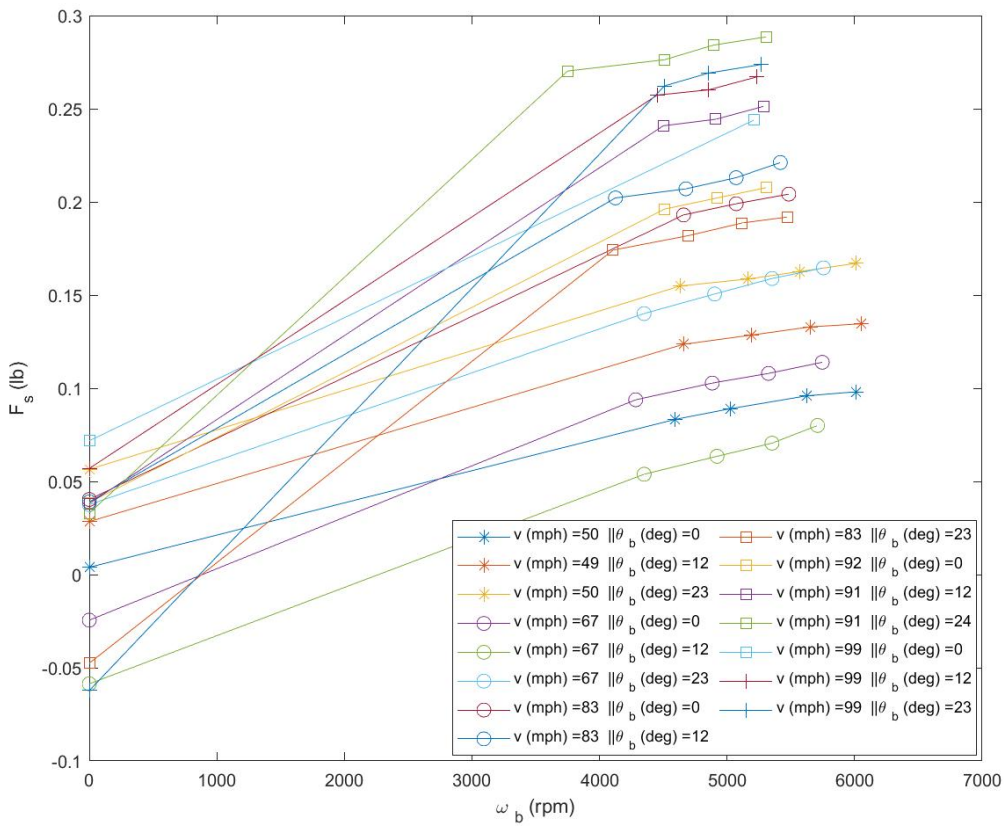
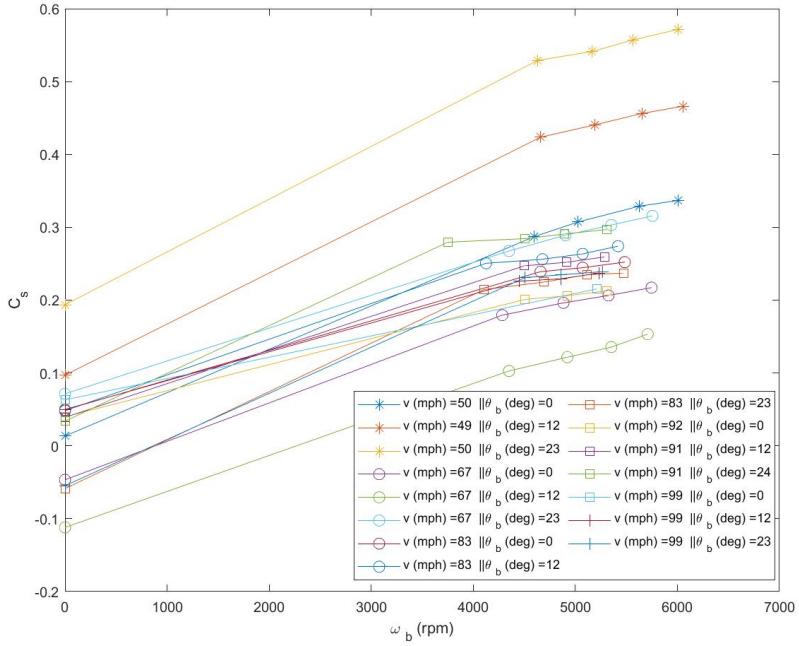
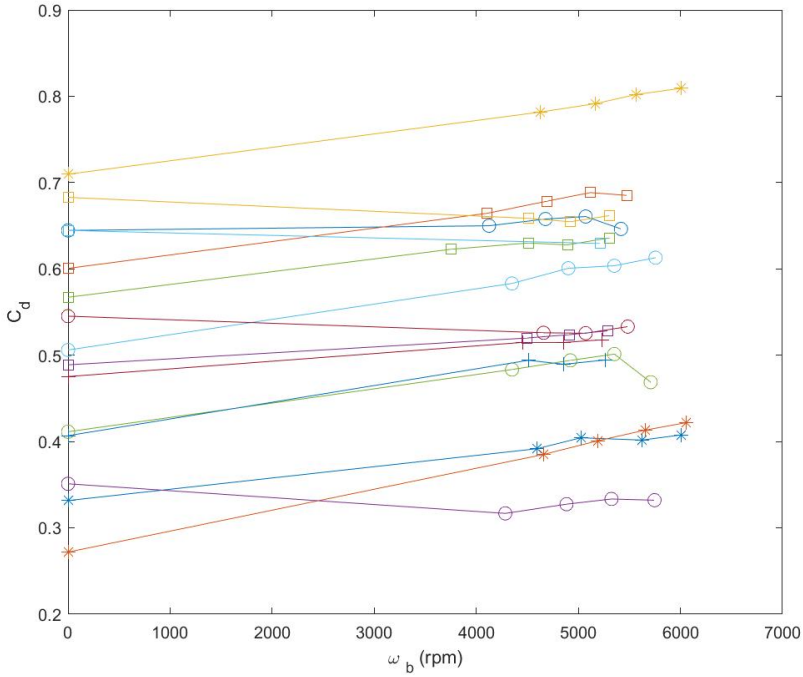


Figure 5.6: Spin rate effect on baseball side force. Variables held constant are $\theta_y = 90$ [deg] and $\theta_z = 0$ [deg].

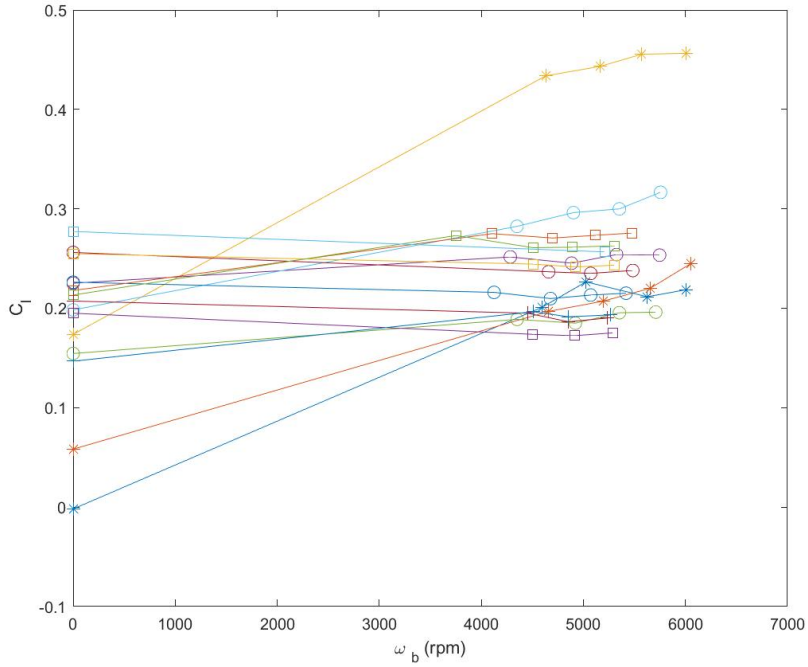
Figure 5.7 shows the effect spin rate has on ball coefficients. The behavior of spin rate on ball coefficients mirror the forces observed on the ball. This makes intuitive sense since the coefficients ares calculated without spin rate. Spin rate only appears to effect the side coefficient, with an increase in spin rate increasing the side coefficient. Drag and lift coefficient are not sensitive to spin rate.


0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
C_s
0
1000
2000
3000
4000
5000
6000
7000
ω_b (rpm)
v (mph) =50 $\|\theta_b$ (deg) =0
v (mph) =49 $\|\theta_b$ (deg) =12
v (mph) =50 $\|\theta_b$ (deg) =23
v (mph) =67 $\|\theta_b$ (deg) =0
v (mph) =67 $\|\theta_b$ (deg) =12
v (mph) =67 $\|\theta_b$ (deg) =23
v (mph) =83 $\|\theta_b$ (deg) =0
v (mph) =83 $\|\theta_b$ (deg) =12
v (mph) =83 $\|\theta_b$ (deg) =23
v (mph) =92 $\|\theta_b$ (deg) =0
v (mph) =91 $\|\theta_b$ (deg) =12
v (mph) =91 $\|\theta_b$ (deg) =24
v (mph) =99 $\|\theta_b$ (deg) =0
v (mph) =99 $\|\theta_b$ (deg) =12
v (mph) =99 $\|\theta_b$ (deg) =23

(a) C_s


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
C_d
0
1000
2000
3000
4000
5000
6000
7000
ω_b (rpm)

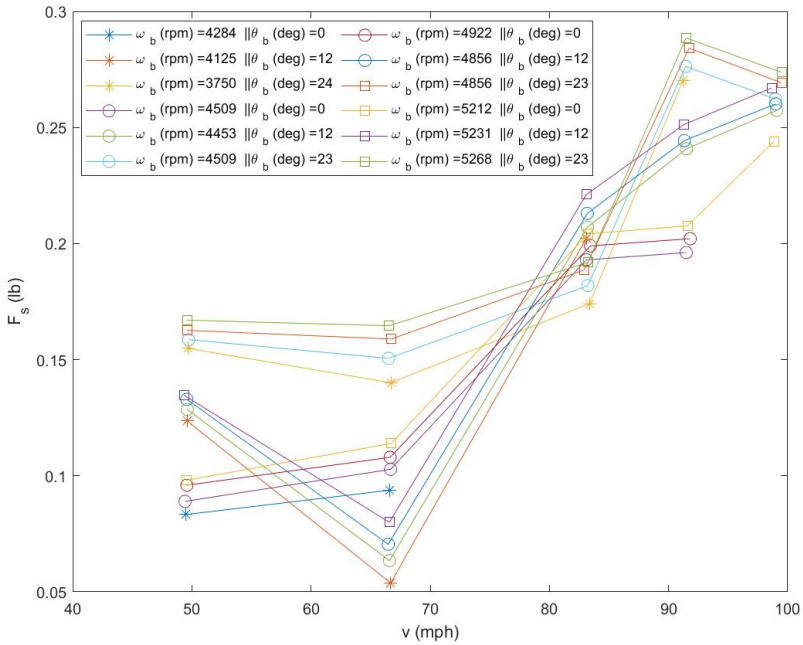
(b) C_d



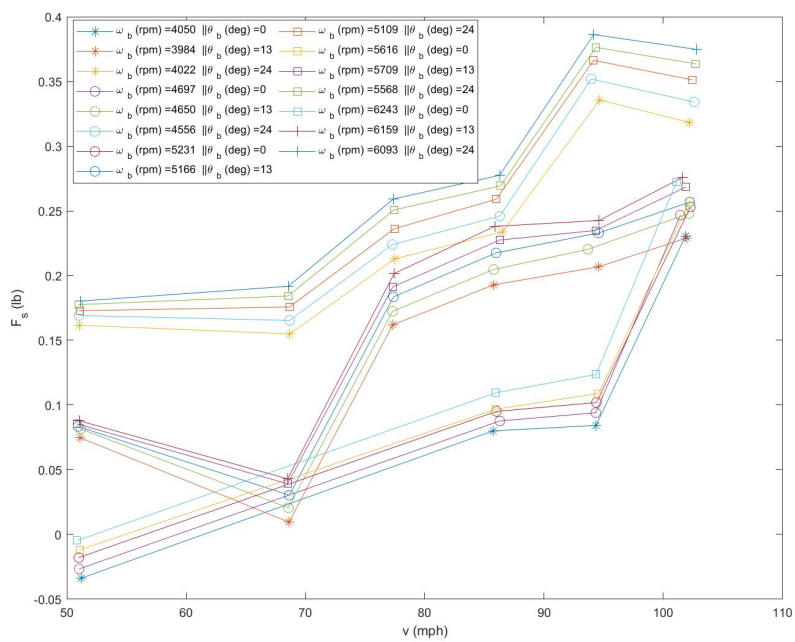
(c) C_l

Figure 5.7: Spin rate effect on ball coefficients. Variables held constant at $\theta_y = 90$ [deg] and $\theta_z = 0$ [deg]. Legends are consistent for all sub-figures.

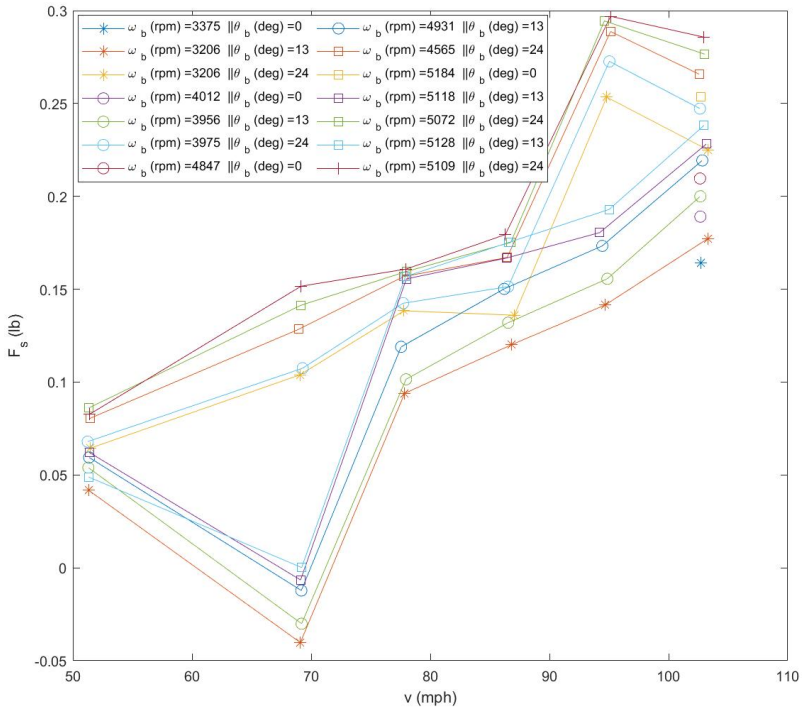
Exploring the differences that seam orientation (θ_y and θ_z) had on ball behavior was more difficult experimentally due to the difficulty changing the seam orientation with enough resolution, however 3 different seam orientations were examined. Figure 5.8 shows the effect that air velocity has on side force for 3 seam orientations and points to seam orientation having an effect on side force. This is an important revelation, showing that seam location on the baseball having an important role in determining the aerodynamics of the system as a whole. While it is apparent that seam orientation has an effect on ball behavior, observing an overall trend for seam orientation is impossible due to only 3 data points within these input variables being observed. Another important observation to see in this figure is that in all seam orientations, the sting angle can be seen to group the trendlines for the ball side forces, as seen previously.



(a) $\theta_y = 90$, $\theta_z = 0$



(b) $\theta_y = 15$, $\theta_z = 2$



(c) $\theta_y = 10$, $\theta_z = 30$

Figure 5.8: Comparison of seam orientations to overall side force.

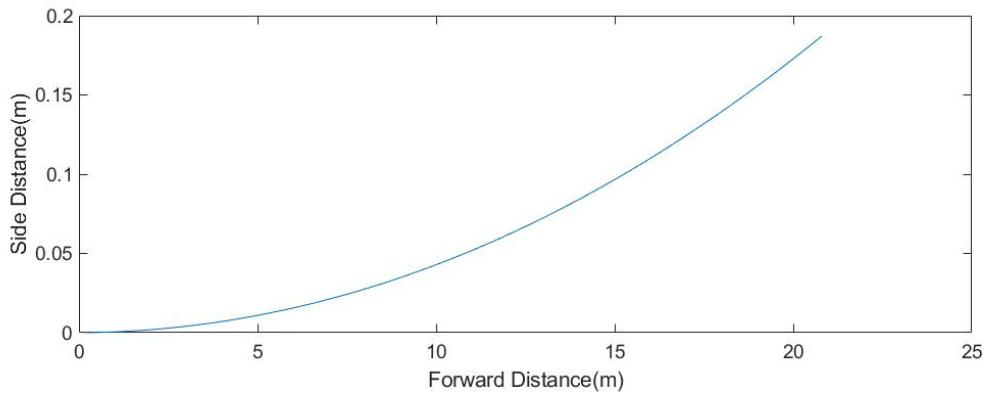
5.2 Ball Trajectory

It is possible to visualize the ball path from the forces gathered. By using Newton's Second Law of motion's equation

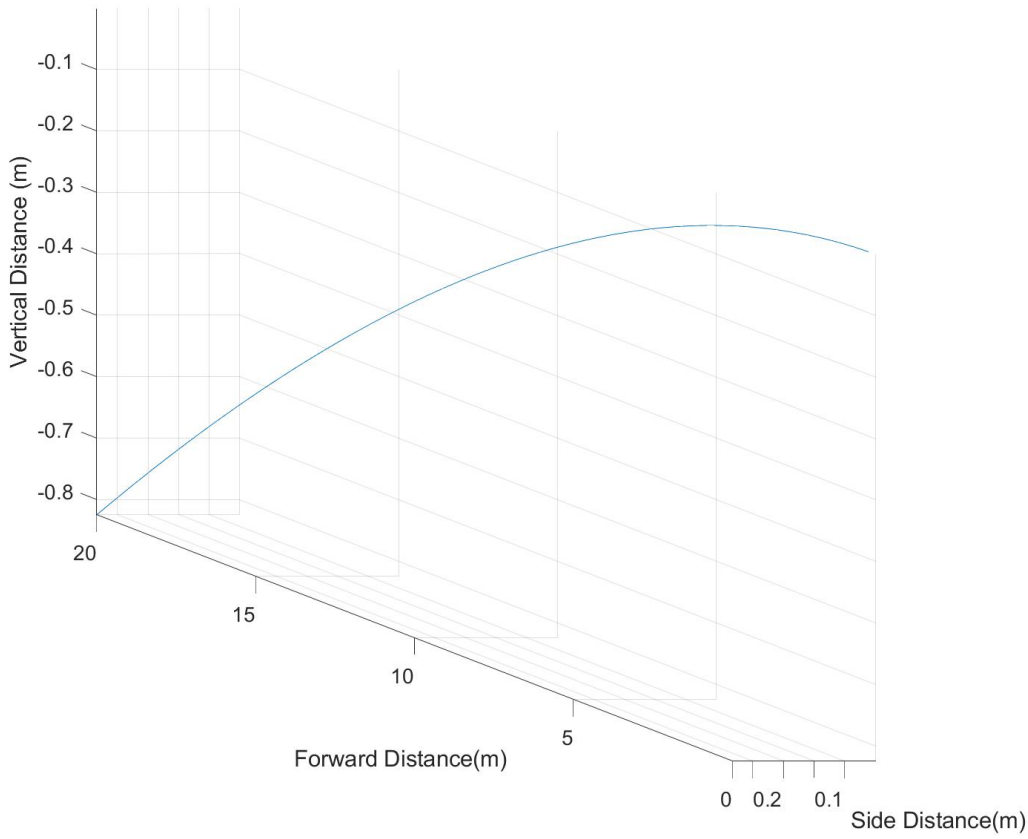
$$F = ma \tag{5.1}$$

where F is force, m is mass, and a is acceleration, the acceleration of the baseball in all 3 directions can be determined. This equation can then be integrated with respect to time to the velocity and again to position of the baseball. Since the ball forces are measured in three axes, the 3D trajectory of the ball can also be attained for any

given trial. Figure 5.9 shows the 3D and top view of the ball's trajectory for a typical pitch distance (18.4 meters). This shows a horizontal travel of roughly 0.17 meters, and a drop of 0.8 meters for a typical pitch length. Other trials (with ball spin) show the horizontal travel ranged from 0.04 - 0.45 meters. These horizontal travel distances are both within ranges of [5] and [1], indicating that the results attained are reasonable.



(a) Top view



(b) 3D view.

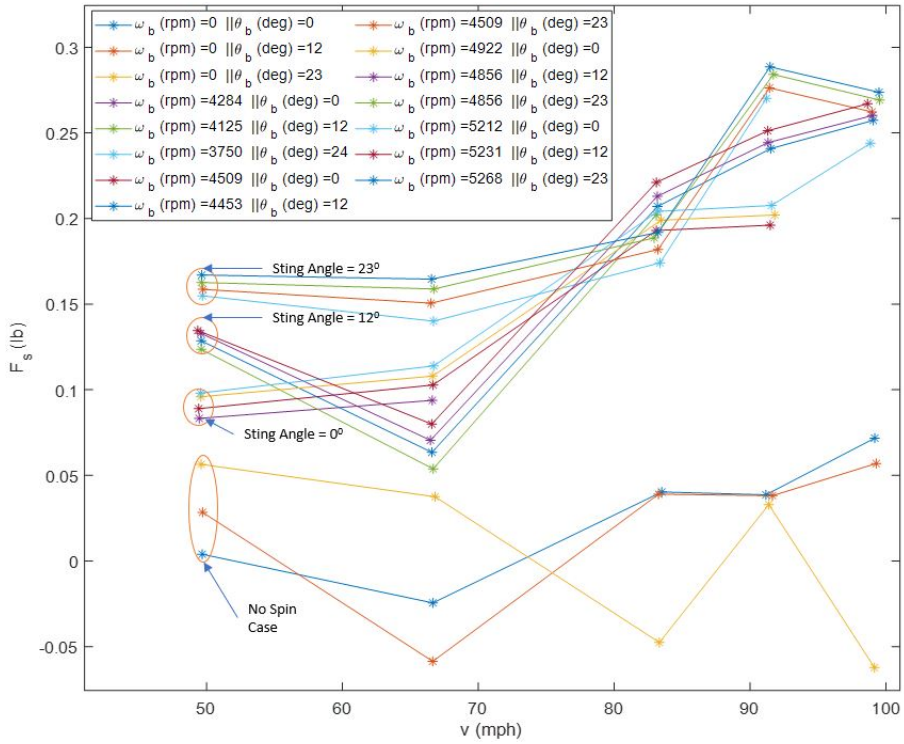
Figure 5.9: Typical ball trajectory based on experimental data for a MLB pitch length.

5.3 Computation Results

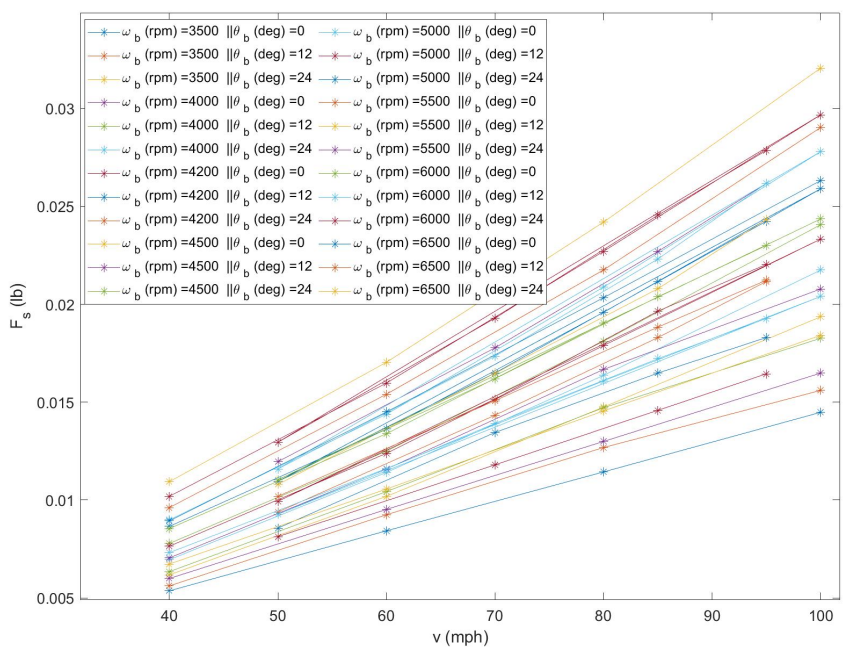
Computational simulations were used to replicate the experimental setup as closely as possible. The data gathered in this way was split up into two primary categories. Some data was primarily run to replicate a particular experimental trial, while other data was gathered to vary one or two input variables with high resolution to examine their specific impact on results. Both of these approaches were important because a comparison to experimental data was useful to explore, as well as varying different input variables not possible through experimental avenues (such as seam orientation with high resolution). Due to the curse of dimensionality in the input variables and relatively small amount of computing power available, a full breadth of all input variables was not possible.

Examining the computational data, some common trends appear when it is compared to the experimental data. The drag and side forces were predicted well qualitatively, but not quantitatively. Lift was predicted with reasonable accuracy.

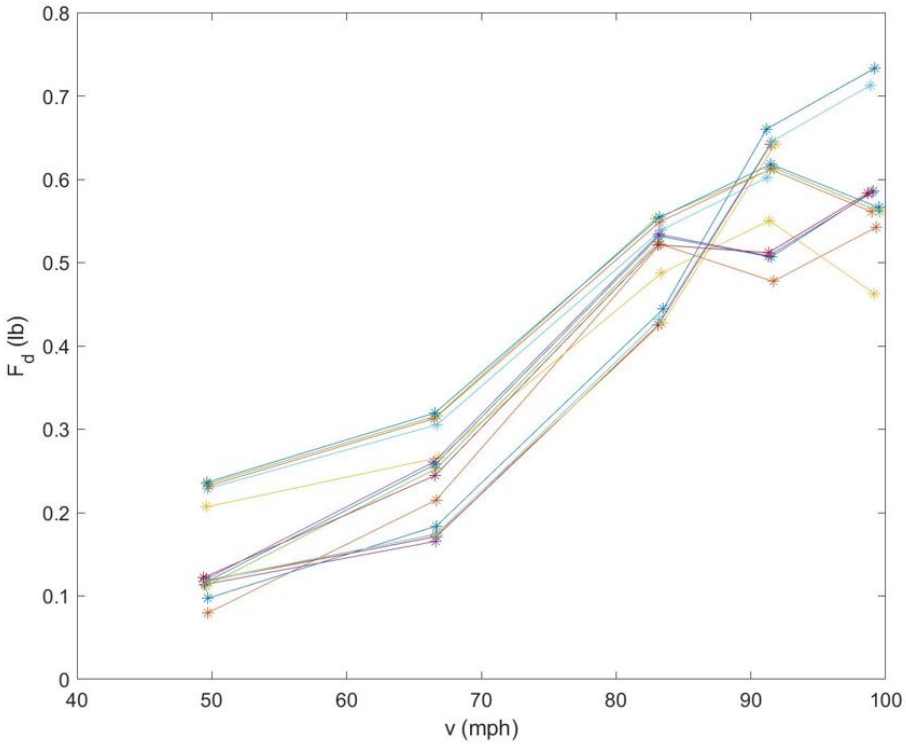
When examining the effect of air velocity on baseball forces, the computational F_s appears to hold the same trend as the experimental data, however it also has a magnitude of roughly 10 times smaller value(Figure 5.10a, 5.10b). This same effect can be seen in comparing F_d, where the trend is consistent, but the computational data is roughly half the magnitude of the experimental data (Figure 5.10c, 5.10d). The computational F_l data can also be seen to closely mirror its experimental counterpart (Figure 5.10e, 5.10f). This same behavior can be seen in figure A5.33 when examining the effect of ω_b on ball forces.



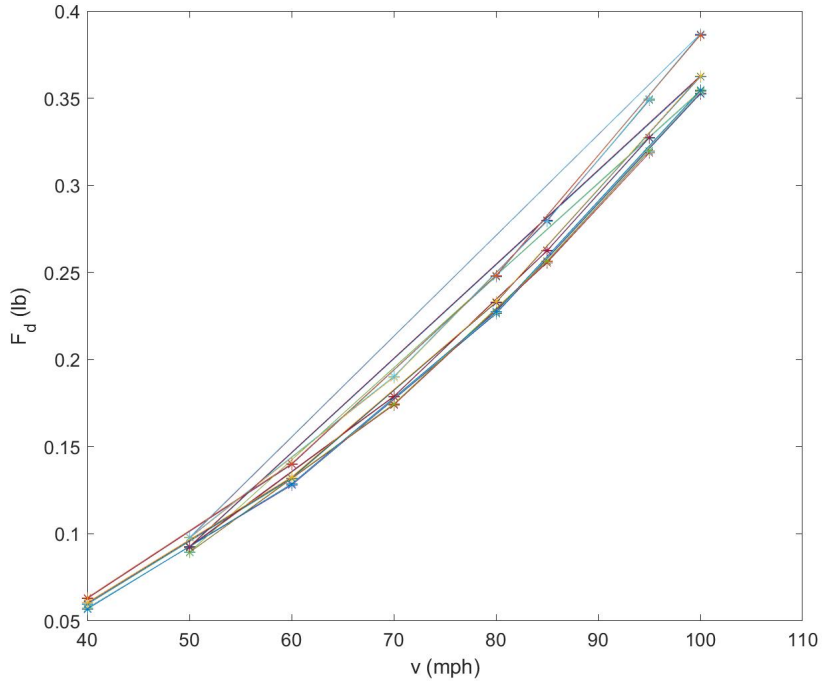
(a) Side Force - Experimental



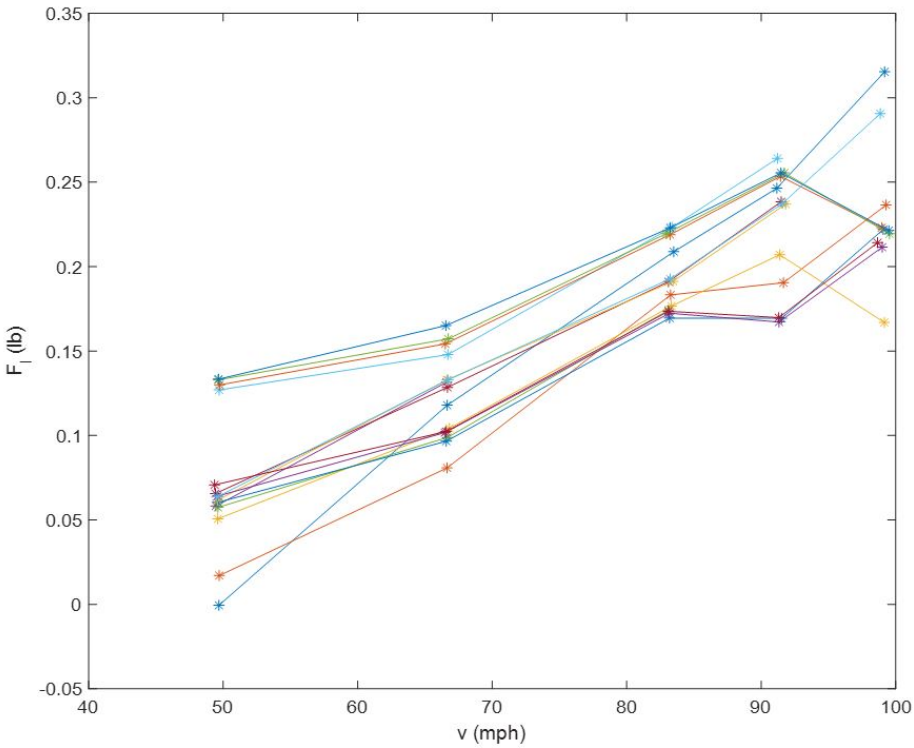
(b) Side Force - Computational



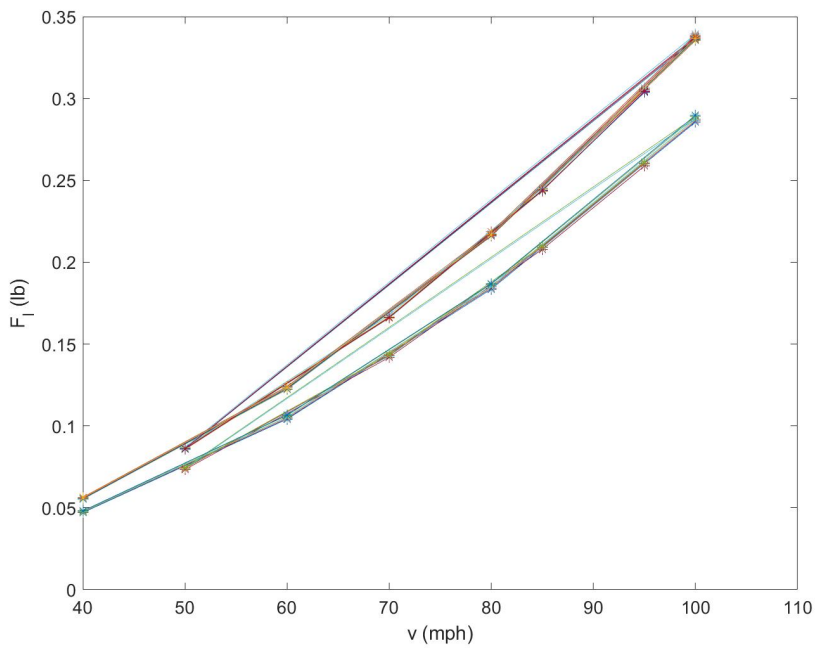
(c) Drag Force - Experimental



(d) Drag Force - Computational



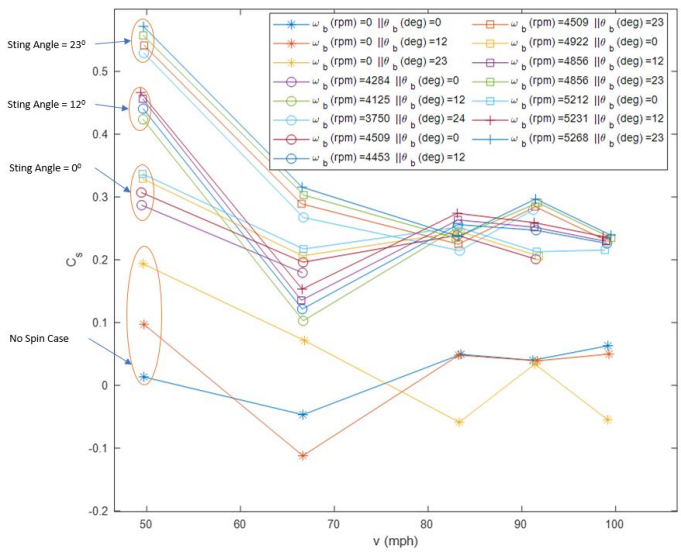
(e) Lift Force - Experimental



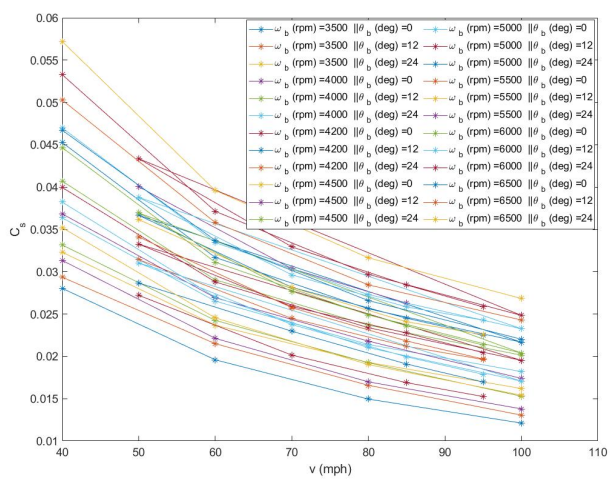
(f) Lift Force - Computational

Figure 5.10: Comparing experimental and computational effect of air velocity, v, on baseball forces. Variables held constant are $\theta_y = 90$ [deg] and $\theta_z = 0$ [deg]. Legends are the same for experimental and computational data respectively.

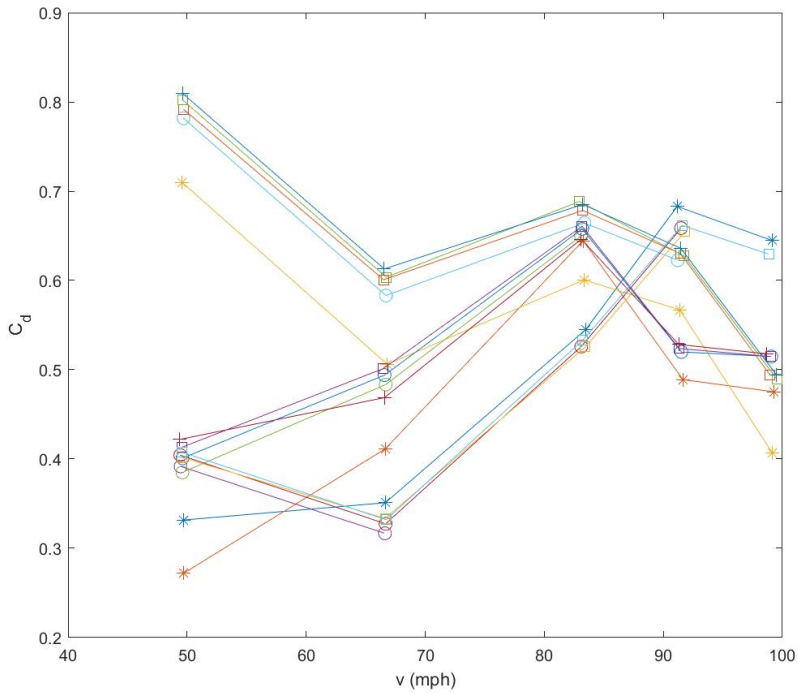
Similar comparisons of velocity's effect on experimental and computational ball coefficients can be seen in Figure 5.11. There appeared to be some qualitative agreement between side coefficients, but the magnitude was off by an order of magnitude (similar to the side force comparison). Drag coefficients were seen to be quantitatively dissimilar, just as the drag forces were as well. The lift coefficients seen to exhibit some similar trends, and were quantitatively close.



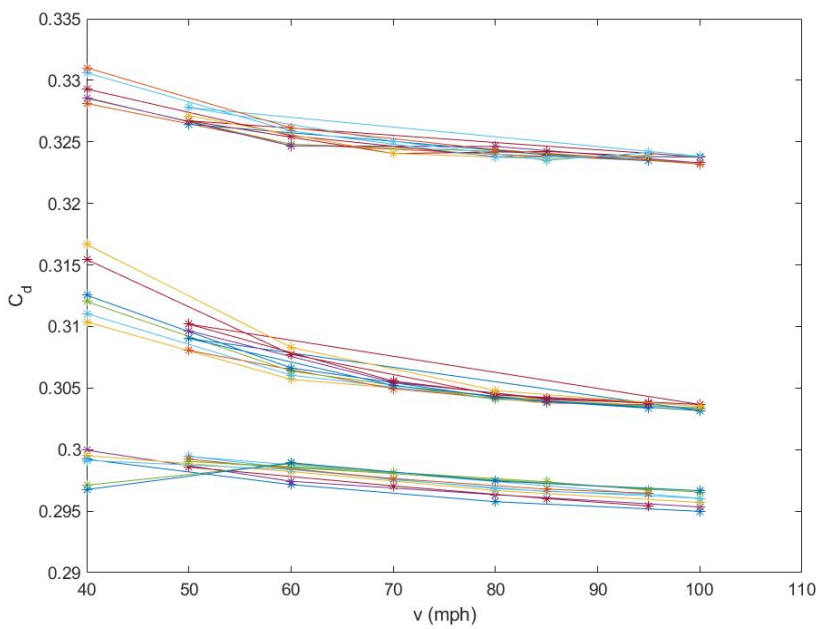
(a) Side Coefficient - Experimental



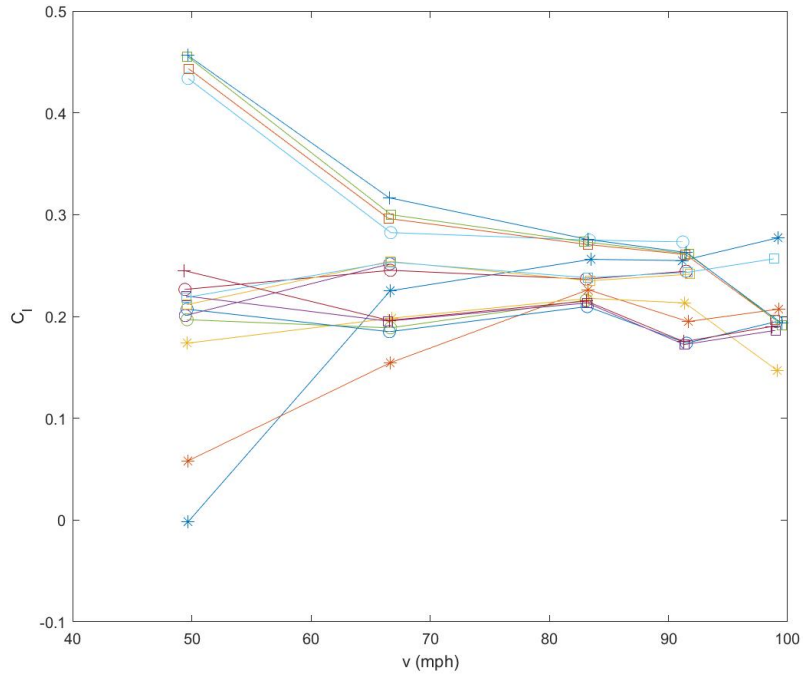
(b) Side Coefficient - Computational



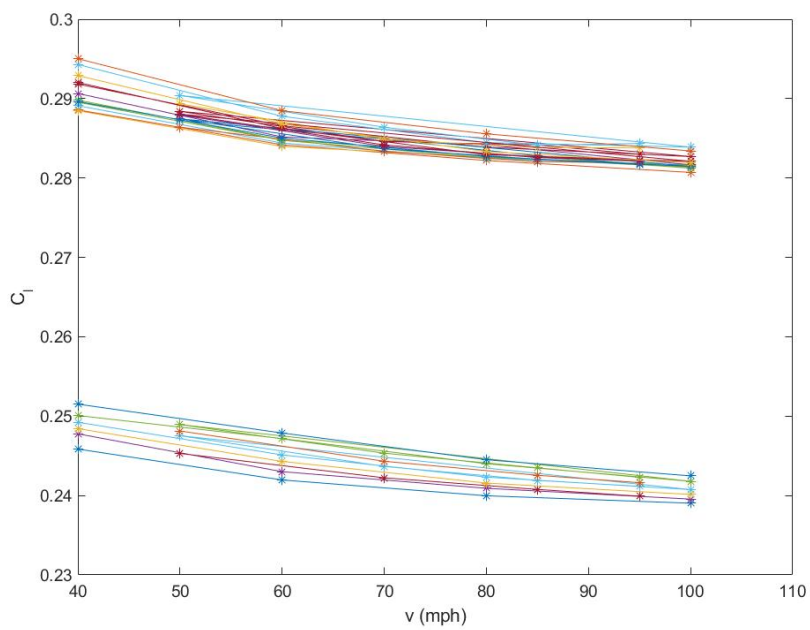
(c) Drag Coefficient - Experimental



(d) Drag Coefficient - Computational



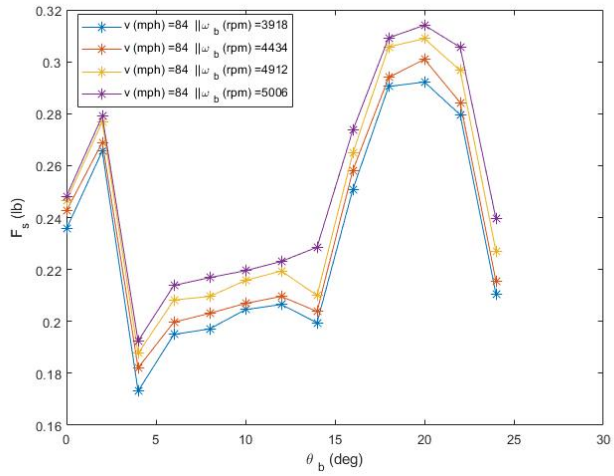
(e) Lift Coefficient - Experimental



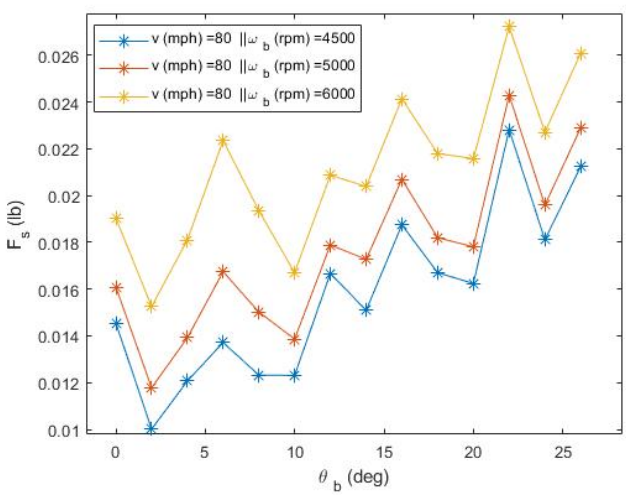
(f) Lift Coefficient - Computational

Figure 5.11: Comparing experimental and computational effect of air velocity, v, on baseball coefficients. $\theta_y = 90$ [deg] and $\theta_z = 0$ [deg]. Legends are the same for experimental and computational data respectively.

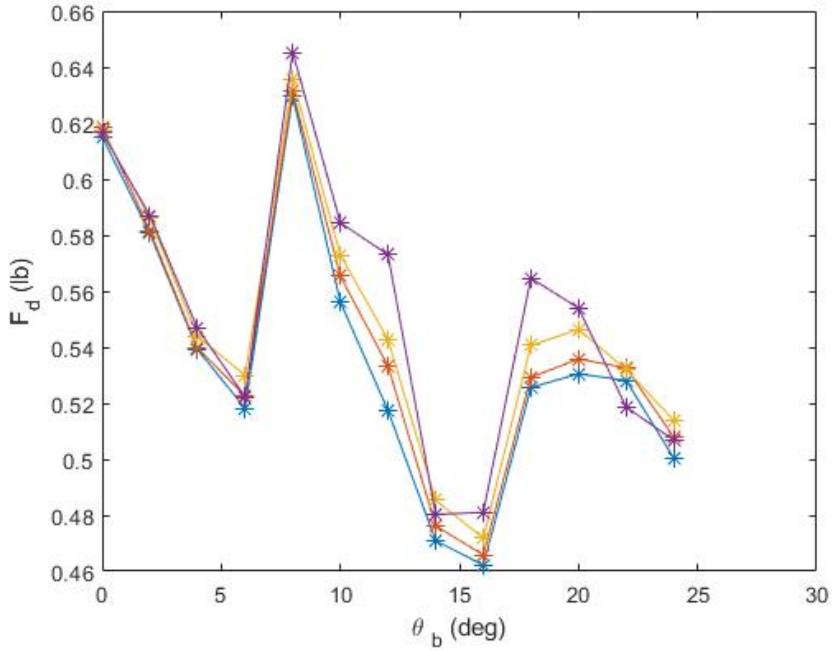
Looking at Figure 5.12 showed the effect that sting angle (θ_b) had on the ball forces. All forces appeared to exhibit non-linear behavior. Side force was off by an order of magnitude, and was hard to compare qualitatively. Drag force was closer in magnitude, but was also difficult to compare. The lift force appeared to be very close both quantitatively and qualitatively.



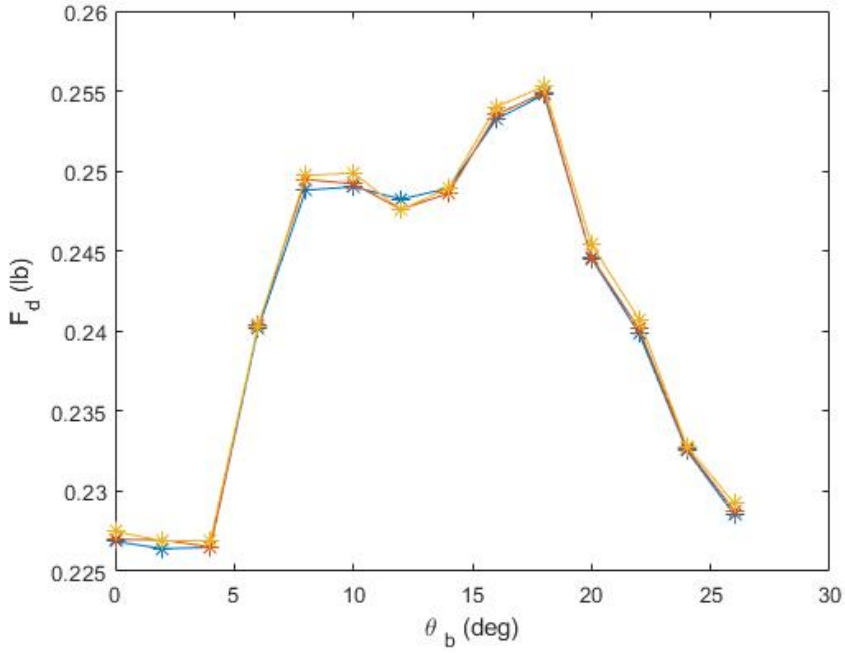
(a) Side Force - Experimental



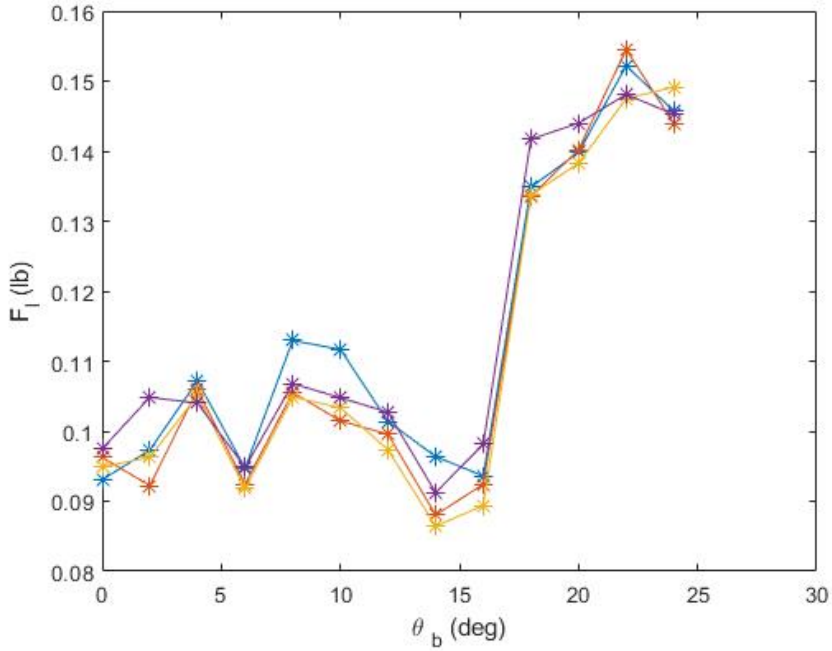
(b) Side Force - Computational



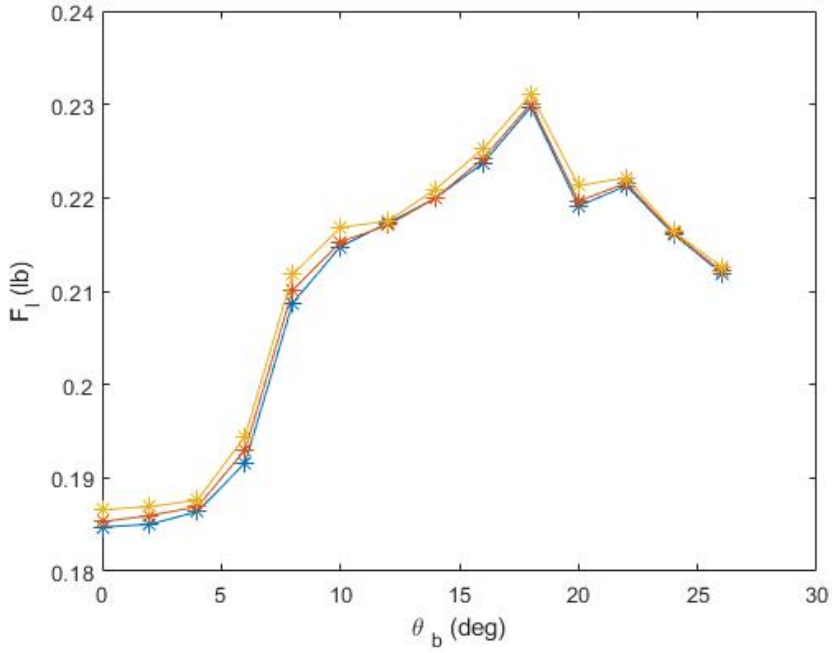
(c) Drag Force - Experimental



(d) Drag Force - Computational



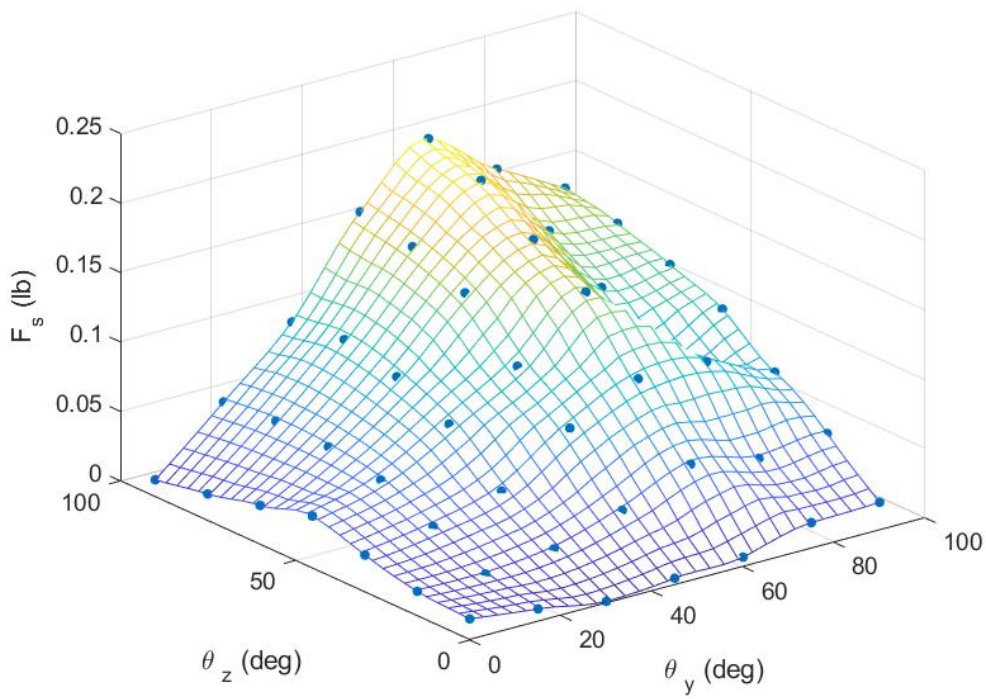
(e) Lift Force - Experimental



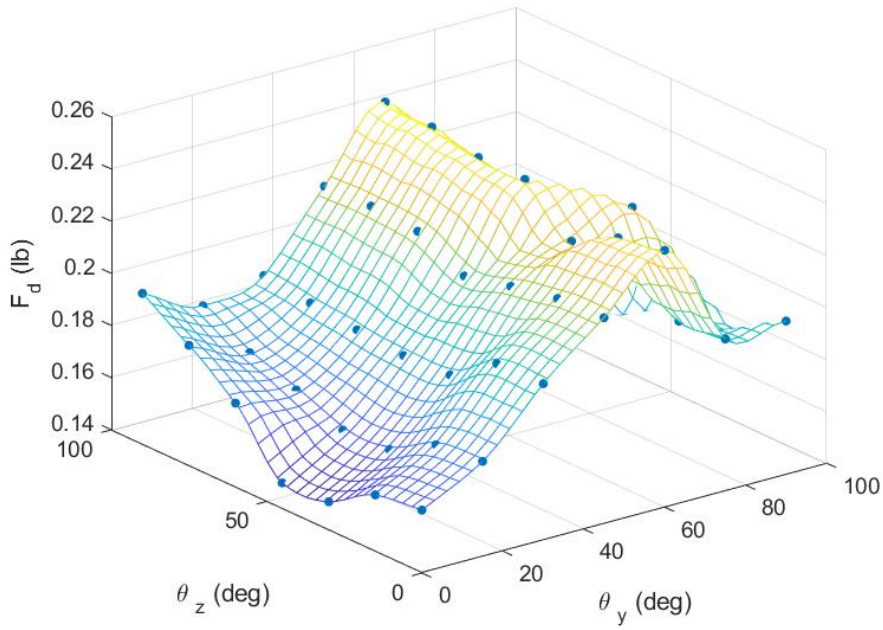
(f) Lift Force - Computational

Figure 5.12: Computational θ_b effect on side forces. $\theta_y = 90$ [deg], $\theta_z = 0$ [deg], v = 80-84 [mph]. Legends are the same for experimental and computational data respectively.

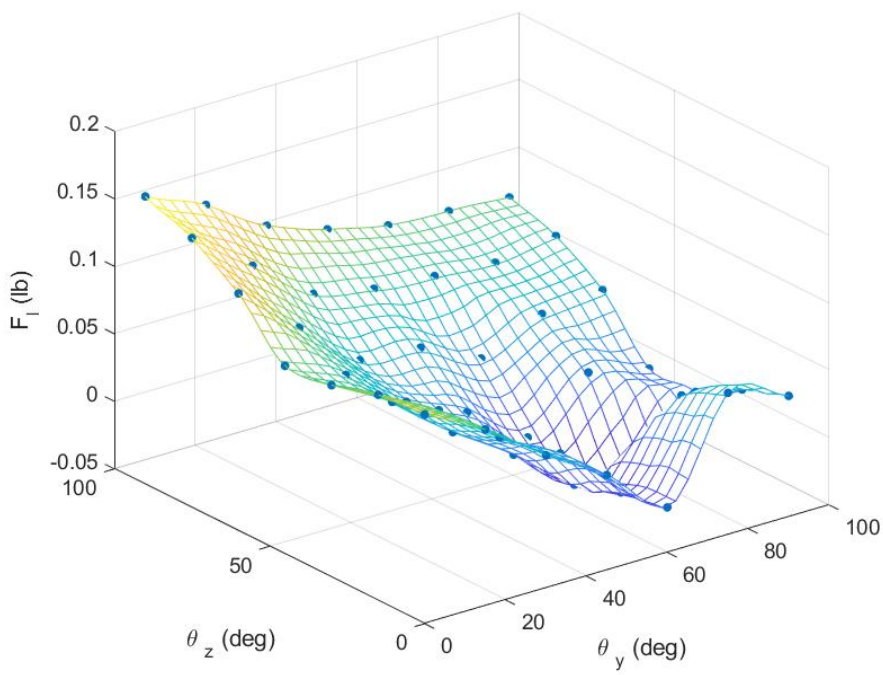
Computational data that does not directly aim to replicate an experimental data set, and instead aims to look at other variations of input variables not explored in experimentation can show some previously unexplored system behavior. Specifically, the seam orientation, θ_y and θ_z, and the sting angle can be further explored in the computational domain. By varying these variables, an example of its impact on results can be seen. Figure 5.13 examines the effect of the seam orientation to its forces. This appeared to be a highly non-linear relationship between either seam orientation and all ball forces.



(a) Side Force - Computational



(b) Drag Force - Computational



(c) Lift Force - Computational

Figure 5.13: Computational seam orientation effect on side forces. Variables held constant are v = 80 [mph], ω_b = 5000 [rpm], θ_b = 0 [deg]. Points are results from CFD, surfaces are best fit cubic surfaces of points.

Looking at a wider range of spin rates could also be explored computationally, and can be seen in Figure 5.14. The spin rate had a linear relationship with side force for a spin rate between 4000 and 15,000. Under 4000 rpm, there appeared to be a slight change in behavior. This may be indicative of a change in ball behavior below 4000 rpm. This change in behavior at low spin rates might explain a phenomena seen in the experiential data. Figure 5.6 showed the experimental data spin rate's effect on side force, and showed that if the spin cases trend-line for the ball was extended to a zero spin case, that a side force would still exist. However, it was seen in this same figure that the zero spin case did not have a significant side force. This disagreement might be explained by the computational data's differing behavior regions for spin rate's effect on side force at low spin rates.

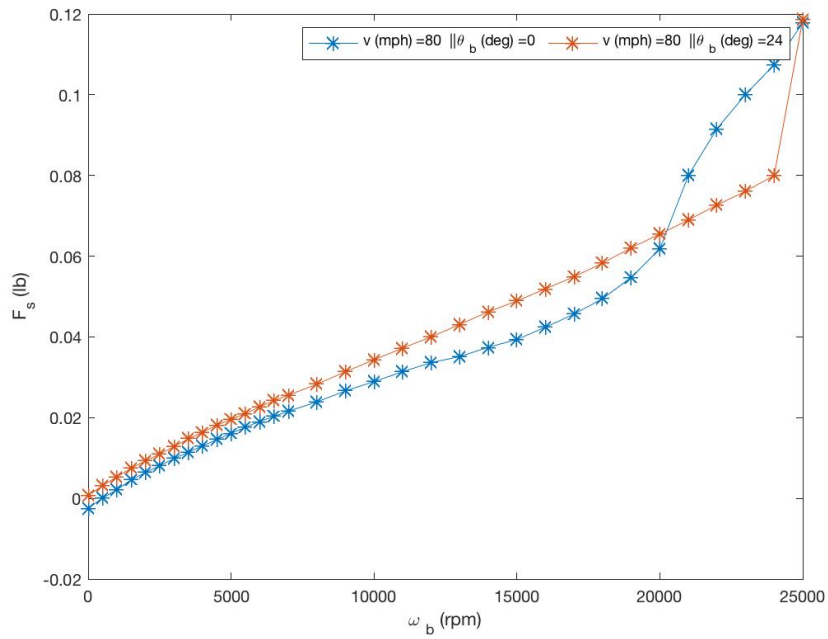


Figure 5.14: Computational ball spin rate effect on side force. Variables held constant at v = 80 [mph], $\theta_y = 0$ [deg], $\theta_z = 90$ [deg].

Visually examining the computational results can allow for further understanding system behavior and help discern why the baseball might exhibit certain behavior at different conditions. Looking at the streamlines of the baseball can help to see how air travels around the ball, and show the magnus effect in action (Figure 5.15).The magnus effect is when a spinning ball travels through the air and exhibits a side

movement. The spin makes the air boundary layer separate at different points on the ball. This non-symmetrical separation point means the air has a non-symmetrical change in momentum. This unequal change in momentum exerts a net side force on the ball, and thus a side movement. In this system, air first flowed onto and around the ball's front face, with the boundary layer breaking toward the rear of the ball. This break point was not symmetrical, with the right side breaking sooner than the left. This means that the air had a net momentum change to the right, which was counteracted by the baseball's net side force going to the left.

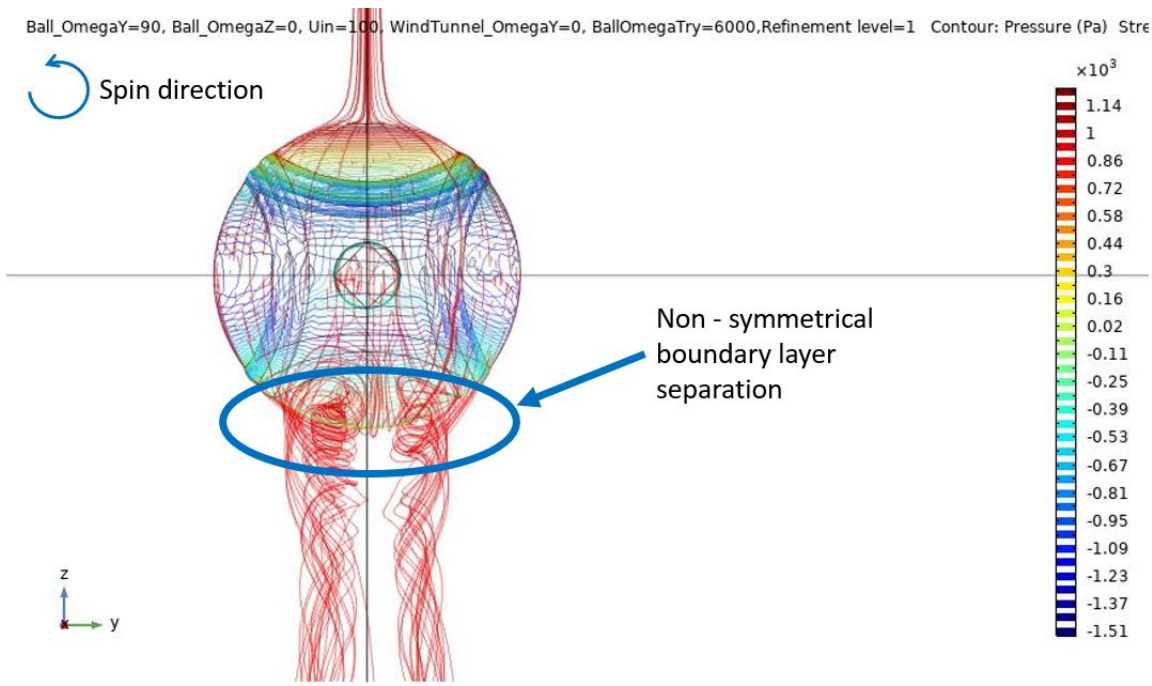
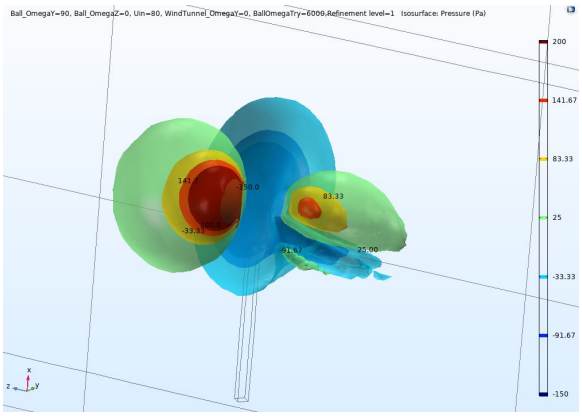


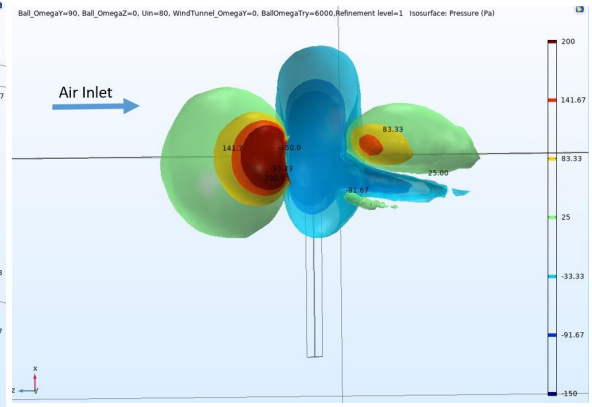
Figure 5.15: Computational ball streamline and surface pressure shown. Variables held constant at v = 100 [mph], ω_b = 6000 [rpm], θ_b = 0 [deg], θ_y = 90 [deg], θ_z = 0 [deg].

Looking at a 3D representation of the pressure around the ball can show a better picture of what is happening in the area surrounding the baseball. It provides insight into what may be happening to a baseball as it travels through the air during a typical baseball pitch. Figure 5.16 shows how the air pressure changes as it moves around the ball. There is a high pressure zone in the front of the ball, followed by a low pressure zone that looks like a donut appearing halfay down the baseball. This zone is non-symmetrcal due to the ball spin, and would have a small impact on a net side force on the ball. There are two low pressure zones behind the ball, resulting from

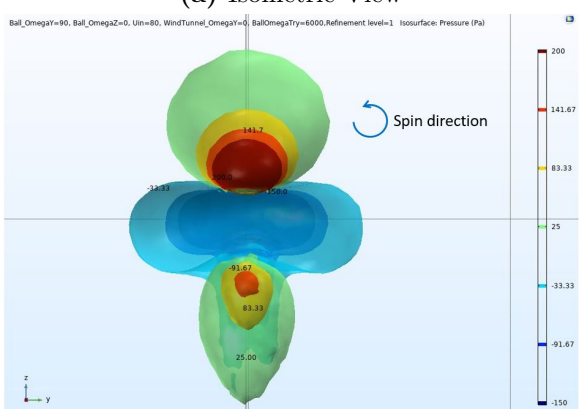
the boundary layer separation at different locations. This too, would result in a net side force to the ball. These non-symmetrical phenomena seen here were the magnus effect at work.



(a) Isometric View



(b) Side View

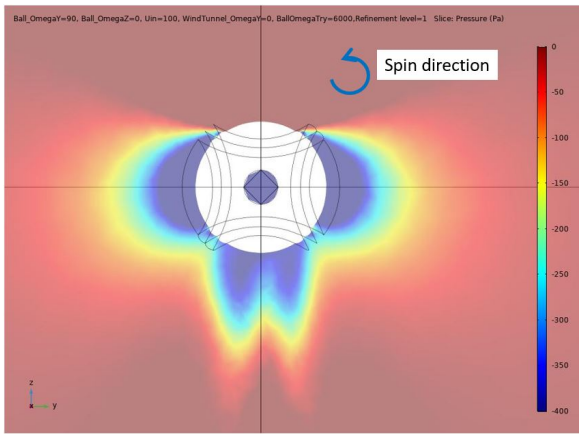


(c) Top View

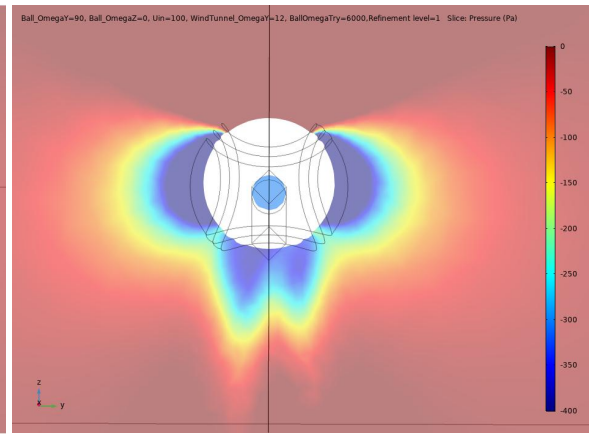
Figure 5.16: Typical computational 3D pressure plot. Pressure ranges limited to show volume around baseball better. Variables held constant at v = 80 [mph], θ_y = 90 [deg], θ_z = 0 [deg], θ_b = 0 [deg], ω_b = 6000 [rpm].

Visualizations can show behavior changes based on differing input variables. For example, Figure 5.17 shows the pressure around the ball at different sting angle, while other variables are held constant. This shows, in general, that a higher θ_b results in smaller low pressure zones on the back and side of the ball. These low pressure zones on the back end of the ball can be seen to be non-symmetrical, as can the low pressure zones on the sides of the ball. This shows the non-symmetrical effect that the spin rate of the ball has on pressure around the ball, and thus its effect on side force as

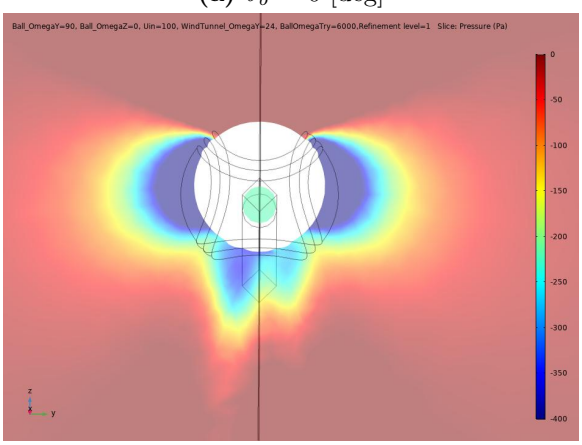
well. It is important to note the change in shape of the low pressure zones, but also the actual pressure of these zones, as they are different for the sub-figures. Figure 5.17 adjusts the pressure range to better examine the low pressure zones, and thus removes the higher pressure zones from Figure 5.17. These different low pressure zones may be indicative of the changing side force, but since the effect sting angle had on side force was seen to be non-linear, it is difficult to point to any one visualization to explian the complicated behavior. Likewise, figure 5.18 shows the same trials as figure 5.17, but shows a the velocity profile for a slice through the side of the baseball. This might be useful for examining a different perspective of what is happening around the baseball at varying sting angles. The slight change in boundary layer separation on the bottom side of the baseball might explain some of the change in lift force for the baseball. Since the lift is again seen to have a non-linear relationship with sting angle, these visualizations are hard to use to make any definitive determinations about what actually causes the difference in ball behavior.



(a) $\theta_b = 0$ [deg]

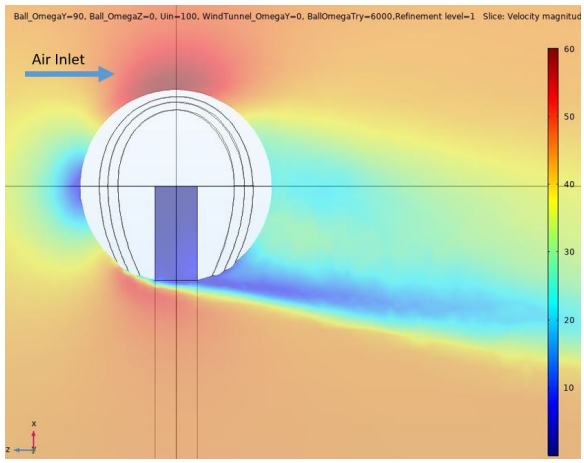


(b) $\theta_b = 12$ [deg]

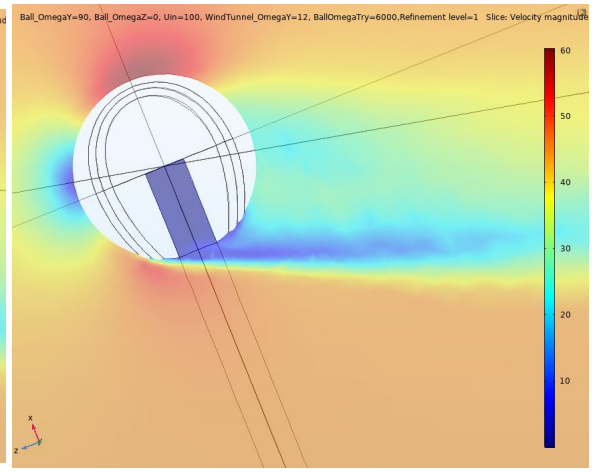


(c) $\theta_b = 24$ [deg]

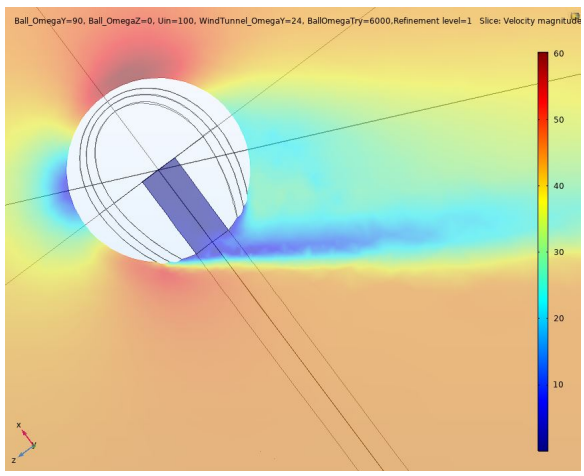
Figure 5.17: Computational sting angle effect on pressure slice of the baseball. Pressure colors were adjusted to examine low pressure zones only. Variables held constant at v = 100 [mph], $\omega_b = 6000$ [rpm], $\theta_y = 90$ [deg], $\theta_z = 0$ [deg].



(a) $\theta_b = 0$ [deg]



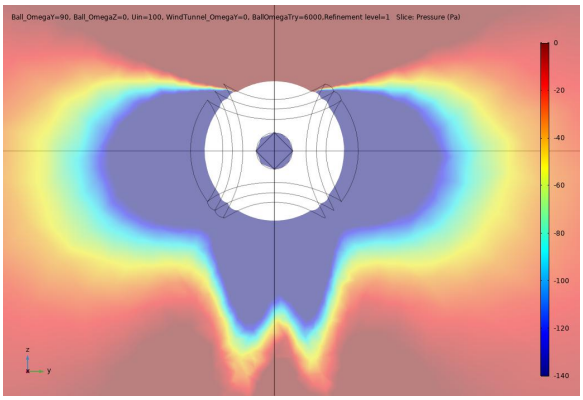
(b) $\theta_b = 12$ [deg]



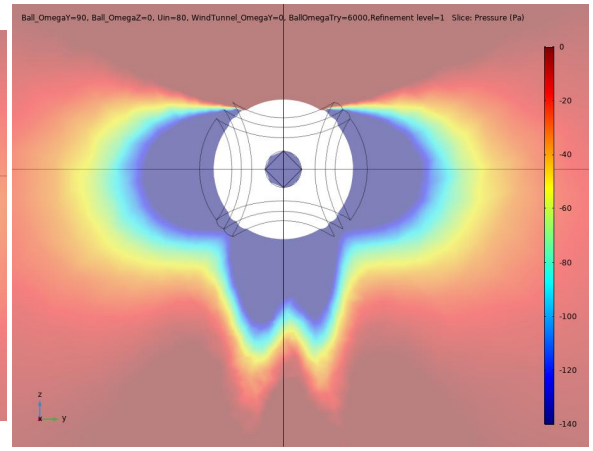
(c) $\theta_b = 24$ [deg]

Figure 5.18: Computational sting angle effect on velocity side slice of the baseball. Pressure colors were adjusted to examine low pressure zones only. Variables held constant at v = 100 [mph], $\omega_b = 6000$ [rpm], $\theta_y = 90$ [deg], $\theta_z = 0$ [deg].

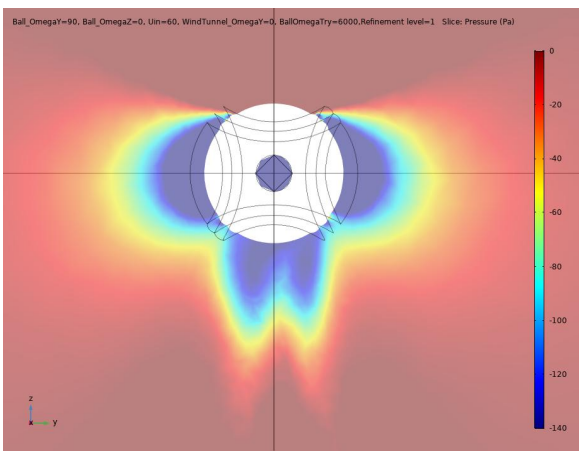
Examining the same kind of cut plot when looking at different air velocities, Figure 5.19 shows the changing pressure zones around the ball. The lower air velocities had a much smaller sphere of low pressure both behind and to the side of the ball. The low pressure zone in Figure 5.19a is substantially larger than the low pressure zone in Figure 5.19d. Both figures do exhibit the same non-symmetric trailing tail of low pressure that can be seen to be the magnus effect in action. The larger amount of low pressure zones would account for the increase in drag force for the higher air velocities.



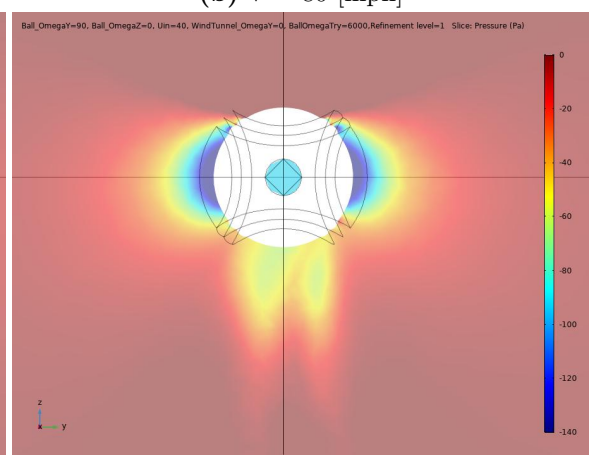
(a) v = 100 [mph]



(b) v = 80 [mph]



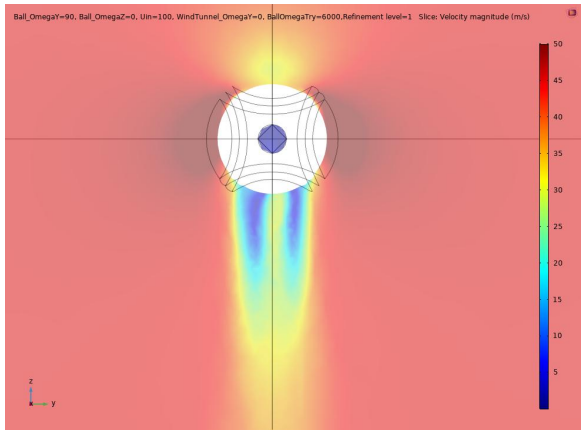
(c) v = 60[mph]]



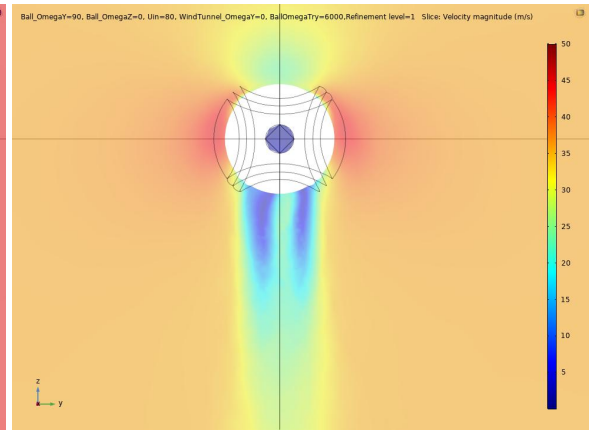
(d) v = 40[mph]]

Figure 5.19: Computational velocity effect on pressure slice of the baseball. Pressure colors were adjusted to examine low pressure zones only. Variables held constant at $\omega_b = 6000$ [rpm], $\theta_b = 0$ [deg] $\theta_y = 90$ [deg], $\theta_z = 0$ [deg].

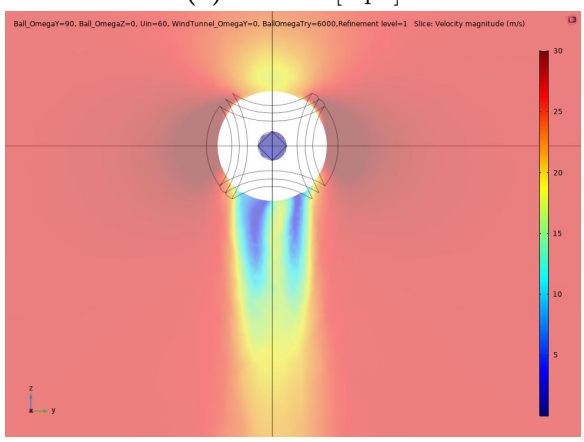
Examining a cut plot of velocity around the baseball can further understanding of how air velocity can change the baseball's behavior. Figure 5.20 shows the velocity cut plots for different air velocities and highlight some features that may effect the ball behavior. The non-symmetrical trailing low velocity section shows how this may result in a net side force on the ball, mirroring the equivalent pressure cut plot. The varying air velocities also show the varying sizes and speeds of the air exiting the rear of the ball, with a higher velocity imparting a smaller low velocity zone directly behind the ball.



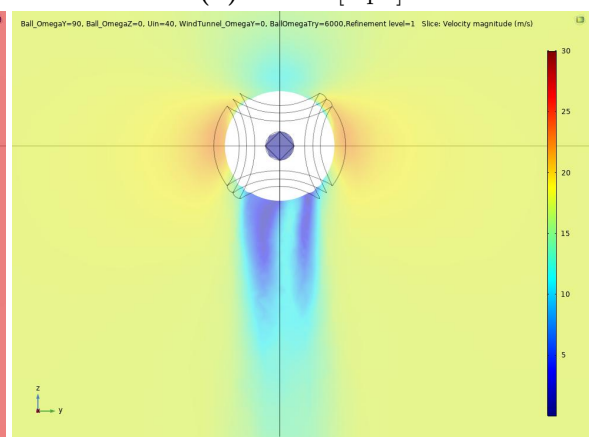
(a) v = 100 [mph]



(b) v = 80 [mph]



(c) v = 60[mph]]

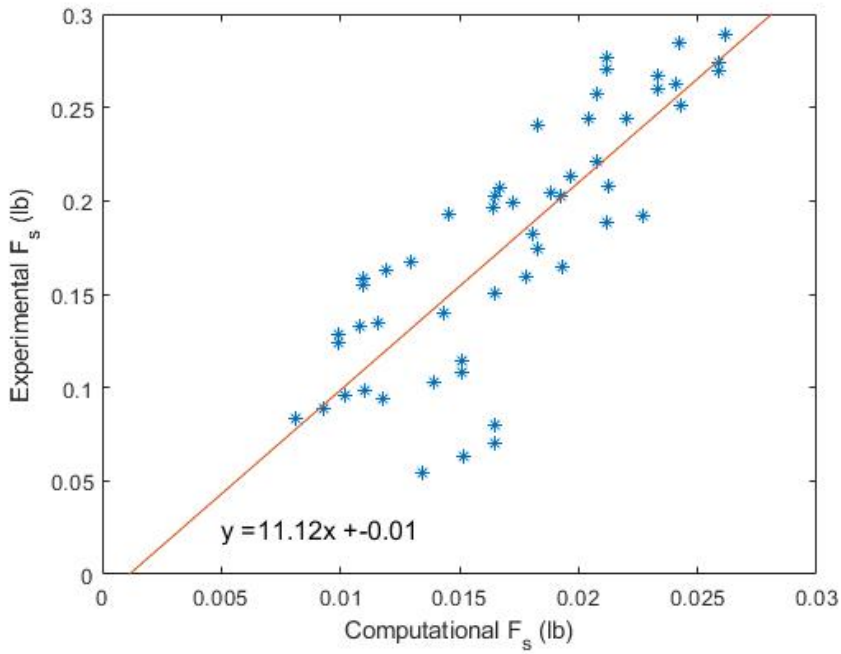


(d) v = 40[mph]]

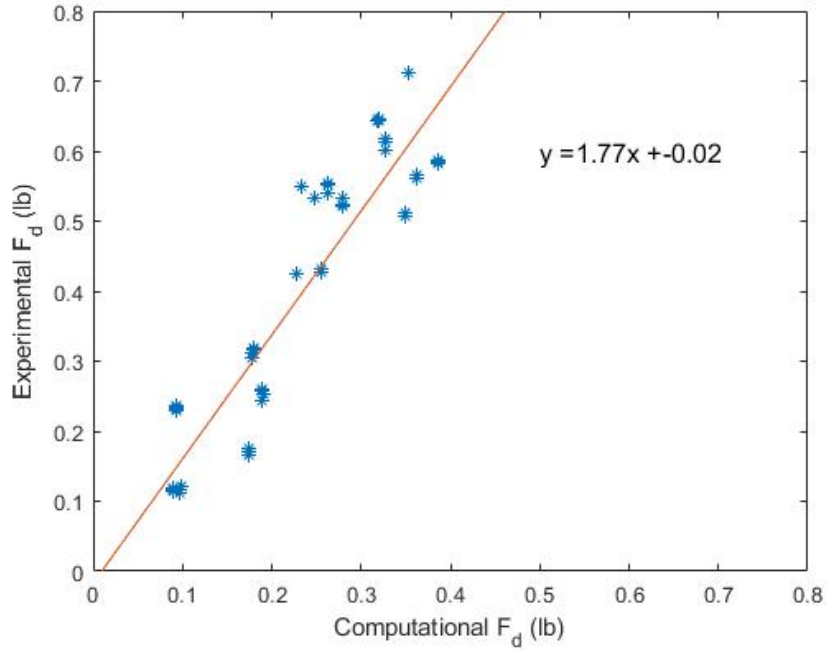
Figure 5.20: Computational velocity effect on velocity slice of baseball. Velocity range were adjusted to better examine baseball features, with ranges differing for two highest and two lowest air velocities. Variables held constant at ω_b = 6000 [rpm], $\theta_b = 0$ [deg] $\theta_y = 90$ [deg], $\theta_z = 0$ [deg].

Examining the agreement between experimental and computational results can be looked at directly with figure 5.21. If the results matched exactly, then the data would be plotted along a straight line with a slope of one. Examining the agreement between side force, the experimental side force was typically 11 times that of the equivalent computational side force. This indicated a large discrepancy in the magnitude between them. They did exhibit a positive correlation between them, meaning that as the experimental side force increased, so did the computational. The drag and lift force were closer in magnitude, and also exhibited a positive correlation between experimental and computational forces. This indicated that the

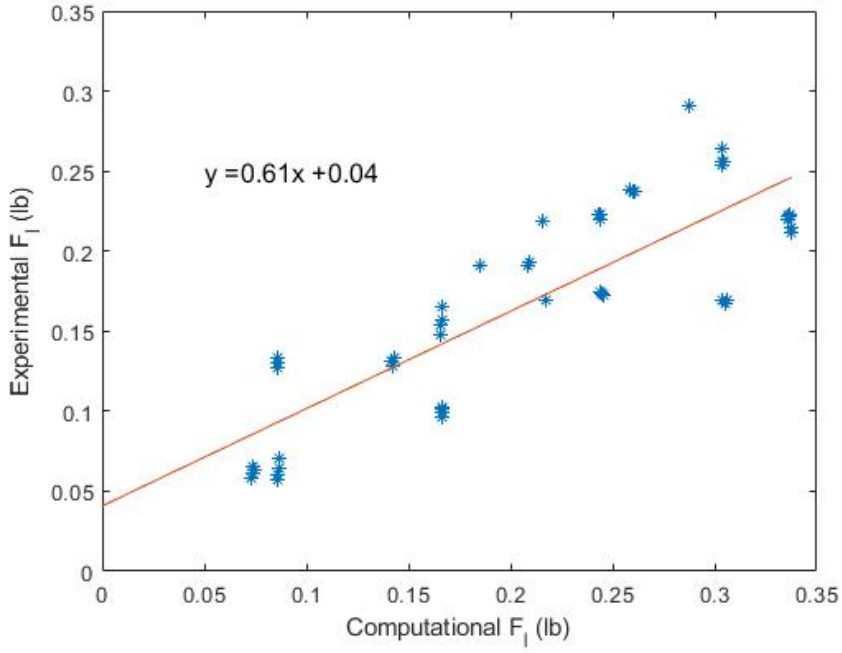
computational results could more accurately predict the drag and lift force, and was not very accurate at predicting the magnitude of side force.



(a) Side Force



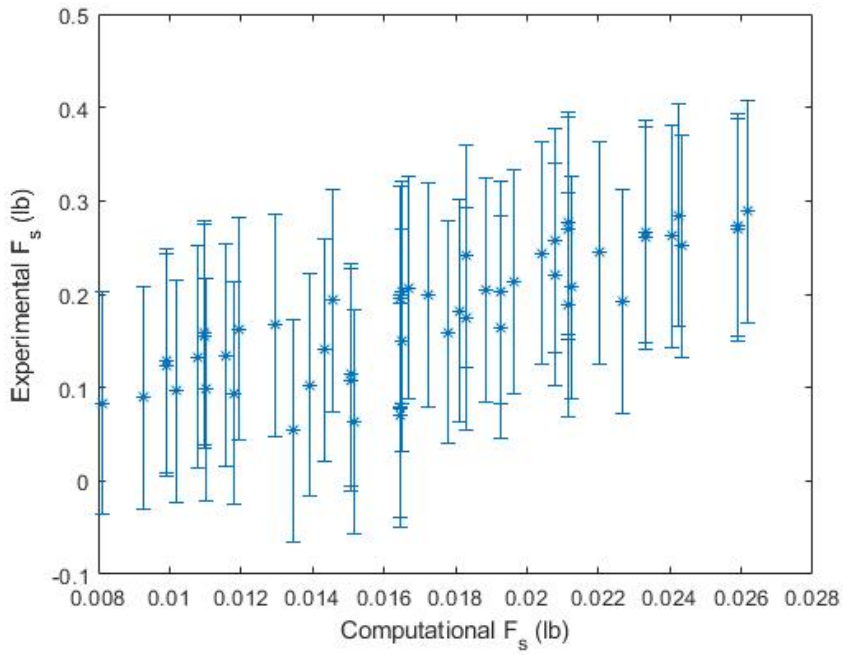
(b) Drag Force



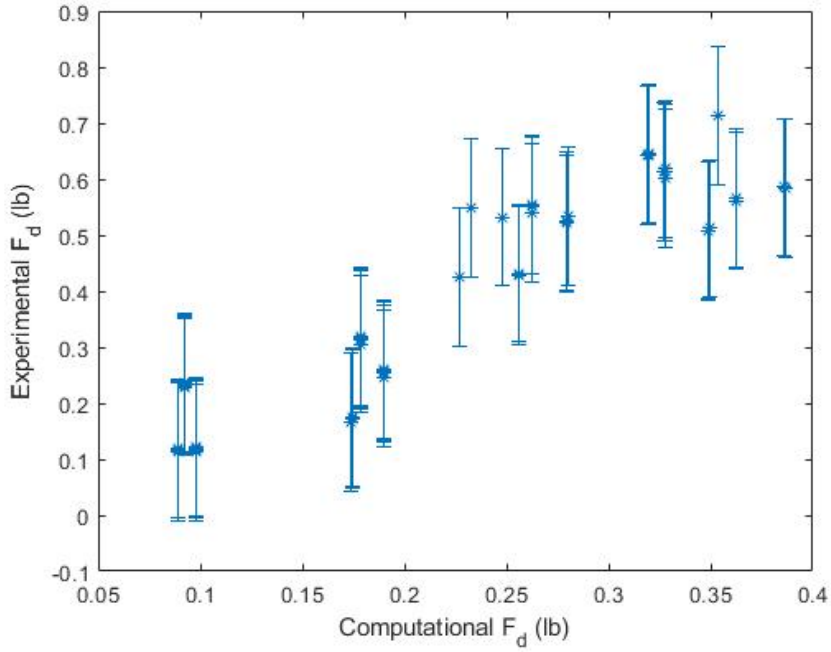
(c) Lift Force

Figure 5.21: Direct comparison of experimental and computational results. Axes are equal for dragf and lift, and not for side force.

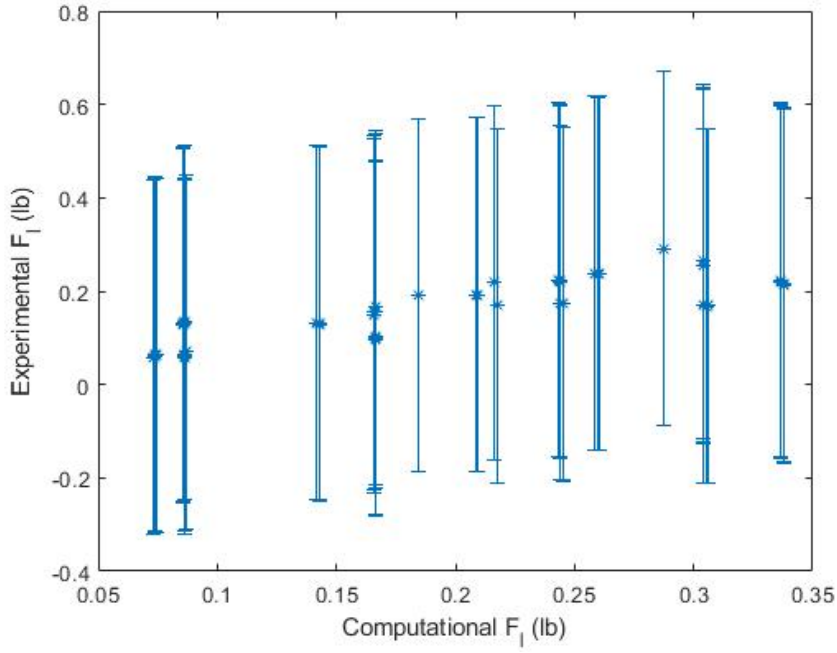
Error bars denoting the full boas including random and systematic bias are shown in figure 5.22. The systematic bias would shift all the experimental forces of the baseball up or down, and would affect how they are compared to computational results. This is different than the random bias, which could change the inherent behavior seen in the baseball forces. The systematic bias is significantly larger than the random bias, and can be seen to have a large impact on the validity of the true data values. This signals that a more accurate load cell would be needed if more valid conclusions between experimental and computational quantitative behavior are to made. For more information on system bias see section 5.6.



(a) Side Force



(b) Drag Force



(c) Lift Force

Figure 5.22: Direct comparison of experimental and computational results with error bars.

5.4 Complex System Exploration

After examining the the ability of the experimental and computational methods to describe the system, the system can be evaluated to see if it can be considered a complex system. A complex system would not be able to be adequately explained by either experimental or computational methods and lack an analytical solution. Looking at experimental results, it can be seen that there are many different variables that go into the behavior of a pitch. Looking at a limited number of input dimensions can allow for some insight, however examining all of the input variables it appears the system behavior is non-linear and analytical solutions are not possible. There is simply too much interdependent change for all of the input variables for solely experimental data to adequately describe the system. Not all variables can be completely modeled, and the sheer number of data points needed would be unfeasible to get.

Computational data also inadequately describes the system. More data can be gathered for the system, with the impact of all input variables able to be more fully explored. Given a sufficient amount of computational power, a larger data cache could be obtained. However, it was seen that the accuracy of the data was severely compromised, with force data able to be off by a factor of ten. This would make the computational data unable to describe the system due to its quantitative shortcomings.

Since both methods are unable to produce a reasonable description of the system, the system can be considered complex. This means that other avenues need to be explored to obtain a model that can describe the system. One avenue to explore is a Physics Based Machine Learning Algorithm (PBMLA). This would attempt to bridge the gap between both previous methods of system behavior exploration.

5.5 Physics Based Machine Learning Algorithm

Combining experimental and computational results in a machine learning algorithm can result in a better predictive explanation of the system behavior. A neural network (NN) was used in this capacity to explore this combination of experimental and computational data to increase the amount of training data points and expand the feature space to include physics based parameters. The additional computational

data points are called "CFD generated data", and physics based parameters used to augment the feature space are called "CFD generated features". This combination of experimental and computational data in a NN will henceforth be referred to as a physics based machine learning algorithm (PBMLA).

Neural networks were set up with the input parameters for each feature space given in Table 3.1. Feature space 1 contains only the input data available from experimental testing, and feature space 2-4 contain combinations of experimental data features and CFD generated features. These CFD generated features include pressure and velocity at the two points given in Figure 5.23, and were added to feature spaces 2 and 3. The CFD generated features computational side force, drag force, and lift force, were added to feature spaces 3 and 4. CFD generated data could also be added to increase the number of training data points. A table of what changes in the NN for the addition of CFD generated data and CFD generated features can be seen in Figure 5.25. The NN was run repeatedly for a range of neurons, until the coefficient of determination (R^2) values asymptoted to a steady value. An example of the R^2 value asymptoting can be seen in Figure 5.24. The R^2 value was calculated with equation

$$R^2 = 1 - \frac{\sum(y - y_{hat})^2}{\sum(y - y_{bar})^2} \tag{5.2}$$

where y is the actual value, y_{bar} is the mean of the y values, and y_{hat} is the predicted value of the actual y value.

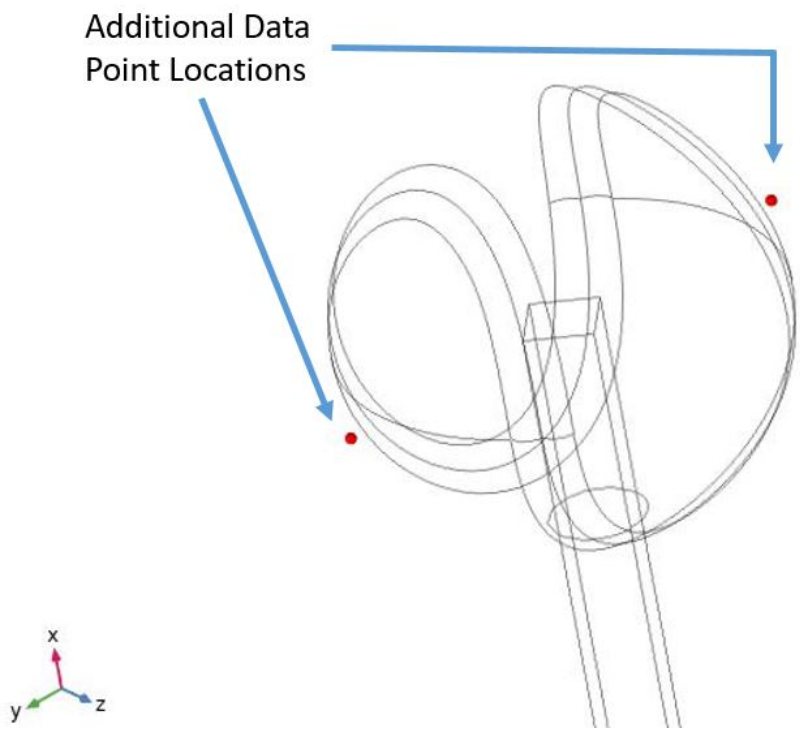


Figure 5.23: Additional data locations where pressure and velocity were gathered in CFD analysis. The poiints were placed on the direct left and right of the ball, 1.7 cm from the center of the ball.

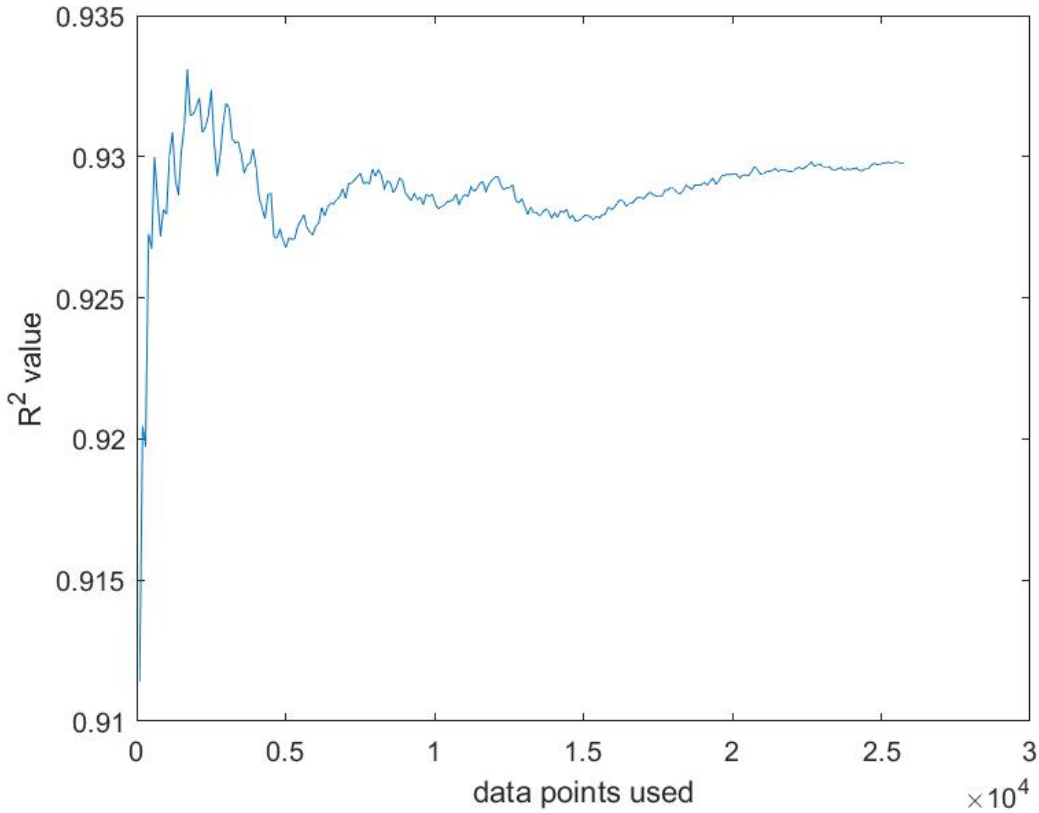


Figure 5.24: R^2 value as more neural network data was taken to compare real data to NN predictive data.

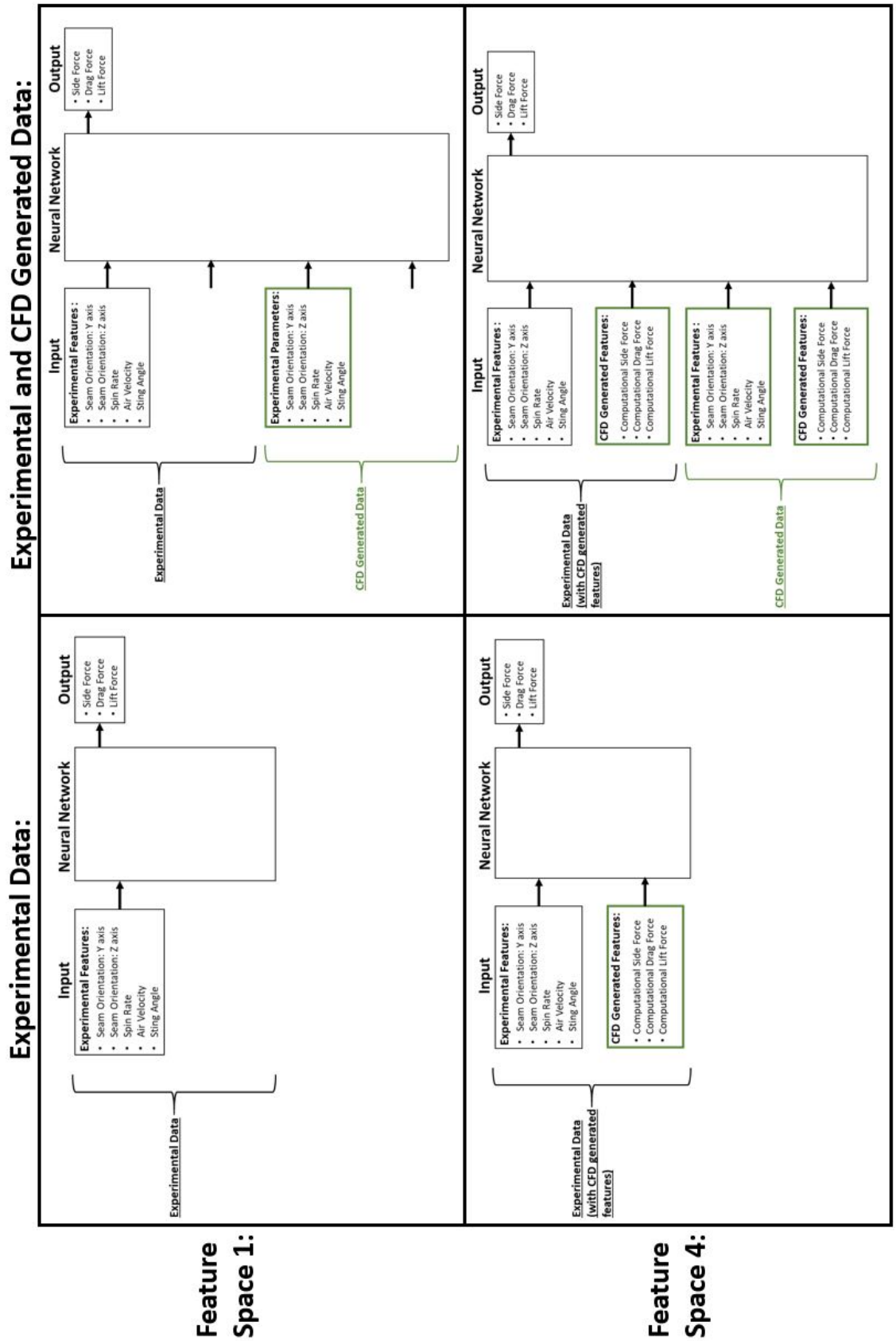


Figure 5.25: Physics based machine learning algorithm potential setups. Feature space variables can be found in table 3.1.

5.5.1 Extrapolation

Complex systems were generally seen to have large gaps in its experimental data (which was true for this study). This necessitated an extrapolation of results to better understand the behavior of the complex system. Neural networks were then used to see if computational data could be used to predict behavior for these gaps in the experimental data. To replicate this situation, the experimental data was manually split into training and testing data, with a specified range of experimental data being taken out of the training data and put into the testing data. This meant that the NN lacked a certain range of experimental training data, and would attempt to predict that missing gap of information. Using the techniques outlined in section 3.6, physics based information was used to increase the understanding of a complex system. These techniques include adding CFD generated data and CFD generated features to the experimental data.

Looking at the the addition of computational data, common themes appeared. Table 5.1b shows how well side force can be predicted from just experimental data when looking at feature space 1. Drag and lift accuracy can bee seen in Table 5.2b and 5.3b. These tables removed a certain range of training data, and then attempted to predict those forces. Looking at how well a NN can extrapolate behavior with just experimental training data, poor predictive ability typically resulted (denoted by a small or negative R^2 value). This pointed to insufficient breadth of experimental data being unsuccessful at describing a complex system well.

This inability for experimental data to fully describe a complex system led to CFD generated data being added to evaluate its impact on NN predictive ability. Table 5.1 shows the NN accuracy for predicting ball side force with and without additional CFD generated data. Equivalent drag and lift force NN can be seen in tables 5.2 and 5.3. When NN trials with CFD generated data were compared to NN trials with only experimental data, the predictive ability of the NN typically increased with CFD generated data. This pointed to CFD generated data being able to help fill the gaps in the experimental data to allow for a better understanding of the complex system.

The effect CFD generated features on NN predictive ability can also be examined in tables 5.1, 5.2, and 5.3 for each ball force by comparing different feature space R^2 values. Feature space 1 had only experimental data, while feature spaces 2-4 contained CFD generated features. Looking at the R^2 values between feature spaces 1 and 2-4, the addition of CFD generated features typically had a positive effect on NN accuracy. This means that CFD generated features could also help fill the

gap of experimental data and make a more robust predictive model. Looking closer at the different feature spaces with physics based parameters (2-4), feature space 3 had the highest increase in R^2 value. This feature space had the most features, indicating that the kind and amount of CFD generated features added to the feature space was important to consider. Since feature space 2 and 4 still typically had a better predictive ability than feature space 1, it lead to the observation that physics based parameters generally help the accuracy of a NN, and choosing which additional information goes into the feature space is an important decision.

Using both CFD generated data and CFD generated features in tandem could also be explored to see if the combination has an even greater increase in NN predictive ability. This can be seen in tables 5.1a ,5.2a, and 5.3a. Feature spaces with CFD generated data and CFD generated features had the highest respective R^2 values. This showed that the combination of techniques had a combined benefit to the understanding of a complex system. Since an increase in NN predictive abilities can be seen with either additional computational data or physics based parameters, this indicated that the addition of CFD generated data and CFD generated features can both help to increase the predictive ability of the model.

Experimental Variable Removed	**Range Excluded**	F_s-R^2: **(Feature Space 1)**	F_s-R^2: **(Feature Space 2)**	F_s-R^2: **(Feature Space 3)**	F_s-R^2: **(Feature Space 4)**	F_s-R^2: **(Feature Space 5)**
Sting Angle [deg]	18 - 30	-2.59	-2.49	-2.3	-2.19	-2.82
Sting Angle [deg]	0 - 8	-0.42	-0.42	-0.22	-0.2	0.37
Sting Angle [deg]	10 - 18	-0.58	-0.33	-0.13	-0.73	-0.08
Motor Speed [RPM]	5000 - 8000	0.67	0.67	0.69	0.67	-0.78
Motor Speed [RPM]	4500 - 8000	0.39	0.42	0.48	0.39	-1.07
Motor Speed [RPM]	2000 - 5000	0.48	0.51	0.51	0.49	-0.79
Motor Speed [RPM]	4000 - 8000	-0.06	0.1	0.16	0.02	-1.31
Air Velocity [mph]	90 - 120	-1.05	-0.92	-1.02	-1	-8.5
Air Velocity [mph]	80 - 120	-1.69	-1.93	-1.27	-1.76	-6.47
Air Velocity [mph]	30 - 80	-1.56	-1.49	-0.7	-0.8	-0.68

(a) Experimental and CFD generated data

Experimental Variable Removed	**Range Excluded**	F_s-R^2: **(Feature Space 1)**	F_s-R^2: **(Feature Space 2)**	F_s-R^2: **(Feature Space 3)**	F_s-R^2: **(Feature Space 4)**	F_s-R^2: **(Feature Space 5)**
Sting Angle [deg]	18 - 30	-3.87	-3.05	-1.56	-2.96	-0.17
Sting Angle [deg]	0 - 8	-0.44	-0.9	-0.5	-0.05	-0.09
Sting Angle [deg]	10 - 18	-0.87	-0.24	-0.24	-0.32	-0.05
Motor Speed [RPM]	5000 - 8000	0.53	0.58	0.61	0.59	0.54
Motor Speed [RPM]	4500 - 8000	-0.15	0.14	0.13	0.03	0.43
Motor Speed [RPM]	2000 - 5000	0.35	0.43	0.46	0.35	0.47
Motor Speed [RPM]	4000 - 8000	-0.63	-0.31	-0.48	-0.6	0.15
Air Velocity [mph]	90 - 120	-1.49	-0.68	-1	-0.93	-1.05
Air Velocity [mph]	80 - 120	-1.89	-1.48	-1.85	-1.82	-1.44
Air Velocity [mph]	30 - 80	-2.74	-2.83	-1.77	-2.49	-2.02

(b) Experimental Data Only

Table 5.1: R^2 values for neural network extrapolation of baseball side force.

Experimental Variable Removed	**Range Excluded**	F_d-R^2: **(Feature Space 1)**	F_d-R^2: **(Feature Space 2)**	F_d-R^2: **(Feature Space 3)**	F_d-R^2: **(Feature Space 4)**	F_d-R^2: **(Feature Space 5)**
Sting Angle [deg]	18 - 30	-0.57	-0.18	0.23	-0.24	0.49
Sting Angle [deg]	0 - 8	0.78	0.83	0.85	0.85	0.72
Sting Angle [deg]	10 - 18	0.37	0.6	0.64	0.43	0.62
Motor Speed [RPM]	5000 - 8000	0.95	0.94	0.95	0.94	0.30
Motor Speed [RPM]	4500 - 8000	0.91	0.9	0.91	0.91	0.26
Motor Speed [RPM]	2000 - 5000	0.92	0.92	0.92	0.91	0.32
Motor Speed [RPM]	4000 - 8000	0.64	0.73	0.75	0.67	0.15
Air Velocity [mph]	90 - 120	-6.6	-3.49	-2.52	-3.07	-17.10
Air Velocity [mph]	80 - 120	-5.41	-2.59	-2.73	-3.48	-7.2
Air Velocity [mph]	30 - 80	-2.22	-0.93	-1.19	-1.14	0.35

(a) Experimental and CFD generated data

Experimental Variable Removed	**Range Excluded**	F_d-R^2: **(Feature Space 1)**	F_d-R^2: **(Feature Space 2)**	F_d-R^2: **(Feature Space 3)**	F_d-R^2: **(Feature Space 4)**	F_d-R^2: **(Feature Space 5)**
Sting Angle [deg]	18 - 30	-2	-1.13	-0.72	-1.03	0.47
Sting Angle [deg]	0 - 8	0.78	0.84	0.85	0.85	0.74
Sting Angle [deg]	10 - 18	0.06	0.62	0.62	0.44	0.67
Motor Speed [RPM]	5000 - 8000	0.89	0.93	0.95	0.91	0.85
Motor Speed [RPM]	4500 - 8000	0.54	0.7	0.74	0.62	0.77
Motor Speed [RPM]	2000 - 5000	0.84	0.86	0.86	0.84	0.79
Motor Speed [RPM]	4000 - 8000	-0.38	-0.14	0.01	-0.24	0.62
Air Velocity [mph]	90 - 120	-4.19	-2.89	-2.56	-2.8	-4.41
Air Velocity [mph]	80 - 120	-4.68	-2.59	-3.43	-4.19	-4.91
Air Velocity [mph]	30 - 80	-4.93	-4.11	-4.33	-4.23	-4.97

(b) Experimental Data Only

Table 5.2: R^2 values for neural network extrapolation of baseball drag force.

Experimental Variable Removed	**Range Excluded**	F_l-R^2: **(Feature Space 1)**	F_l-R^2: **(Feature Space 2)**	F_l-R^2: **(Feature Space 3)**	F_l-R^2: **(Feature Space 4)**	F_l-R^2: **(Feature Space 5)**
Sting Angle [deg]	18 - 30	-4.65	-4.13	-3.29	-4.58	-1.15
Sting Angle [deg]	0 - 8	0.74	0.76	0.8	0.76	0.85
Sting Angle [deg]	10 - 18	-1.8	-1.12	-0.93	-1.03	-0.82
Motor Speed [RPM]	5000 - 8000	0.87	0.86	0.86	0.83	0.32
Motor Speed [RPM]	4500 - 8000	0.78	0.79	0.79	0.75	0.17
Motor Speed [RPM]	2000 - 5000	0.82	0.78	0.8	0.76	0.05
Motor Speed [RPM]	4000 - 8000	0.01	0.25	0.28	0.18	-0.05
Air Velocity [mph]	90 - 120	-3.53	-1.89	-2	-1.69	-4.49
Air Velocity [mph]	80 - 120	-7.86	-5.3	-5.7	-6.1	-3.53
Air Velocity [mph]	30 - 80	-5.43	-2.96	-2.14	-3.21	-0.25

(a) Experimental and CFD generated data

Experimental Variable Removed	**Range Excluded**	F_l-R^2: **(Feature Space 1)**	F_l-R^2: **(Feature Space 2)**	F_l-R^2: **(Feature Space 3)**	F_l-R^2: **(Feature Space 4)**	F_l-R^2: **(Feature Space 5)**
Sting Angle [deg]	18 - 30	-6.3	-5.72	-4.26	-4.03	-0.26
Sting Angle [deg]	0 - 8	0.69	0.75	0.76	0.76	0.8
Sting Angle [deg]	10 - 18	-1.95	-0.91	-0.74	-1.12	-0.14
Motor Speed [RPM]	5000 - 8000	0.71	0.79	0.81	0.78	0.75
Motor Speed [RPM]	4500 - 8000	0.16	0.47	0.49	0.31	0.68
Motor Speed [RPM]	2000 - 5000	0.62	0.63	0.66	0.57	0.62
Motor Speed [RPM]	4000 - 8000	-0.65	-0.28	-0.3	-0.52	-0.45
Air Velocity [mph]	90 - 120	-1.9	-1.31	-1.08	-1.3	-2.5
Air Velocity [mph]	80 - 120	-3.6	-2.6	-2.14	-2.48	-3.87
Air Velocity [mph]	30 - 80	-5.44	-3.82	-3.99	-4.38	-3.18

(b) Experimental Data Only

Table 5.3: R^2 values for neural network extrapolation of baseball lift force.

5.5.2 Interpolation

To examine what happens when a full breadth of experimental data is known and computational data was added, Table 5.4 shows the R^2 values for a NN using all experimental data and all feature spaces. This was accomplished by artificially limiting the testing data range of baseball input variables, and allowed a NN to predict values only within the limit of its training data. This means that these trials are useful to see if the addition of CFD generated data and CFD generated features could be useful to a NN that already has a full breadth of experimental data. In

reality, this would not be the case for most complex systems. As seen previously, the experimental data is incomplete in its breadth due to the curse of dimensionality and physical limitations with the experimental setup. 10-fold cross validation was used with the experimental data, with CFD generated data being all added to the training data set to ensure the NN was predicting only experimental data.

The results of Table 5.4 show that just the experimental data led to a NN being able to accuratly predict future baseball behavior. A small increase in NN predictive ability was seen some of the time with the addition of computational data to the training data set, indicating that an increase in data density can be slightly helpful. The addition of physics based parameters showed no significant change to the NN predictive ability. This lack of increase in NN predictive ability points to the possibility that an increase in information to the NN has a limit, and that if the NN already has a sufficient amount of experimental data to predict the system, additional CFD generated data and CFD generated features has limited effectiveness.

Feature Space	F_s - R^2	F_d - R^2	F_l - R^2
Feature Space 1	0.85	0.98	0.91
Feature Space 2	0.86	0.97	0.91
Feature Space 3	0.88	0.97	0.91
Feature Space 4	0.85	0.98	0.90
Feature Space 5	0.75	0.92	0.85

(a) Experimental and CFD generated data.

Feature Space	F_s - R^2	F_d - R^2	F_l - R^2
Feature Space 1	0.84	0.97	0.90
Feature Space 2	0.82	0.97	0.90
Feature Space 3	0.83	0.97	0.91
Feature Space 4	0.84	0.97	0.89
Feature Space 5	0.72	0.90	0.80

(b) Experimental data only.

Table 5.4: R^2 values for neural network interpolation of baseball forces. Ten fold cross validation used.

Examining feature space 5 in Table 3.1, a neural network with just CFD generated features can be examined. Looking at its ability to predict system behavior when a gap of experimental data is taken out of the training data, this feature space did a reasonable job as compared to other feature spaces. It generally had an R^2 value close to other feature spaces, and even sometimes outperformed other feature spaces. It also performs almost as well as other feature spaces when there is no gap in experimental

data, seen in Table 5.4.

Comparing the increase in NN accuracy for additional CFD generated data and CFD generated features indicated that the NN could benefit from this addition for extrapolation, but not interpolation. This pointed to the potential diminishing return of computational data when sufficient experimental data is present. This demonstrated that the addition of CFD generated data and CFD generated features can typically aide in extrapolation, but not interpolation of a complex system.

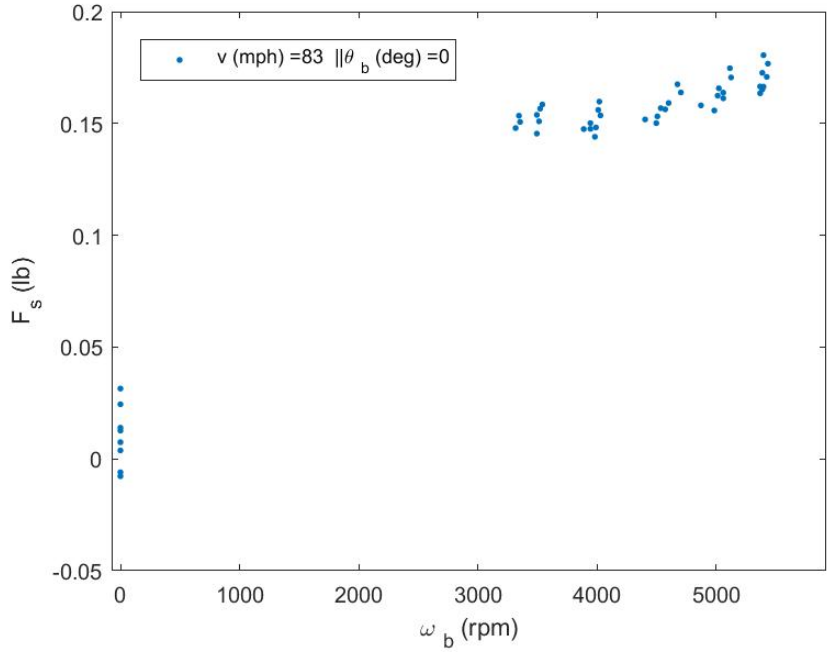
5.6 Uncertainty Analysis

To understand if the assertions about the data are relevant, it is necessary to see how large the uncertainty is with these numbers. This can show if the trends observed are significant, or simply fall into the inherent error in the system. Major sources of uncertainty include the load cell's inherent error (B_s) and the random error associated with data collection (B_d). The load cell's error is determined from the manufacturers data, and can be seen in Figure 5.26. The manufacturer did recommend to go with a more conservative error percentage of 1% - 1.5% of full scale due to the age of the load cell. This gives a conservative average sensor bias of 0.1125 , 0.1125, and 0.375 [lb] for F_s, F_d, and F_l respectively.

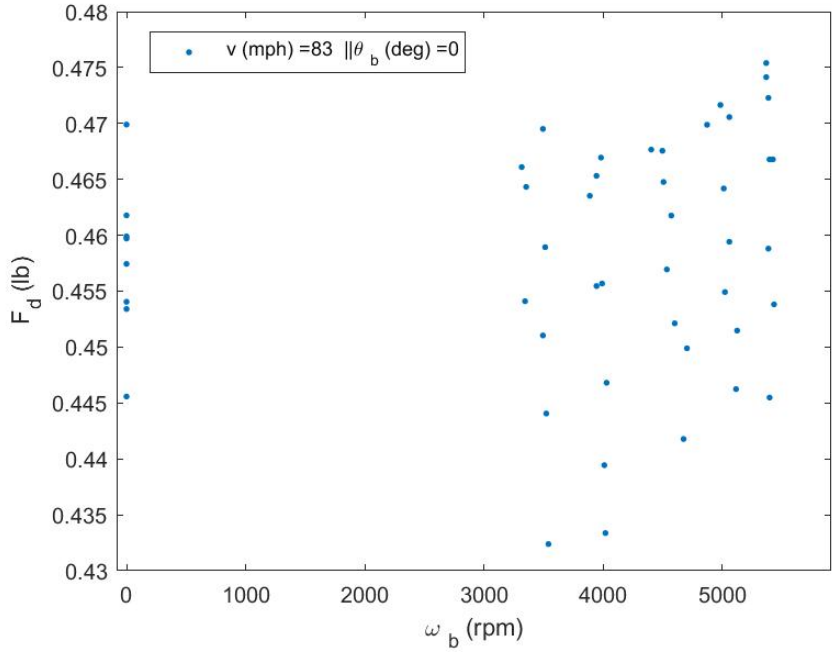
Gamma	**Fx**	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**
	0.75 %	1.00 %	0.75 %	1.00 %	1.25 %	1.50 %
	Rated Fxy		**Rated Fz**	**Rated Txy**		**Rated Tz**
	7.5		25	25		25
US-7.5-25	**Fx Window**	**Fy Window**	**Fz Window**	**Tx Window**	**Ty Window**	**Tz Window**
	0.056	0.075	0.187	0.250	0.313	0.375
	Force Units		**Torque Units**			
	lbf		lbf-in			

Figure 5.26: ATI 6 axis force sensor error.

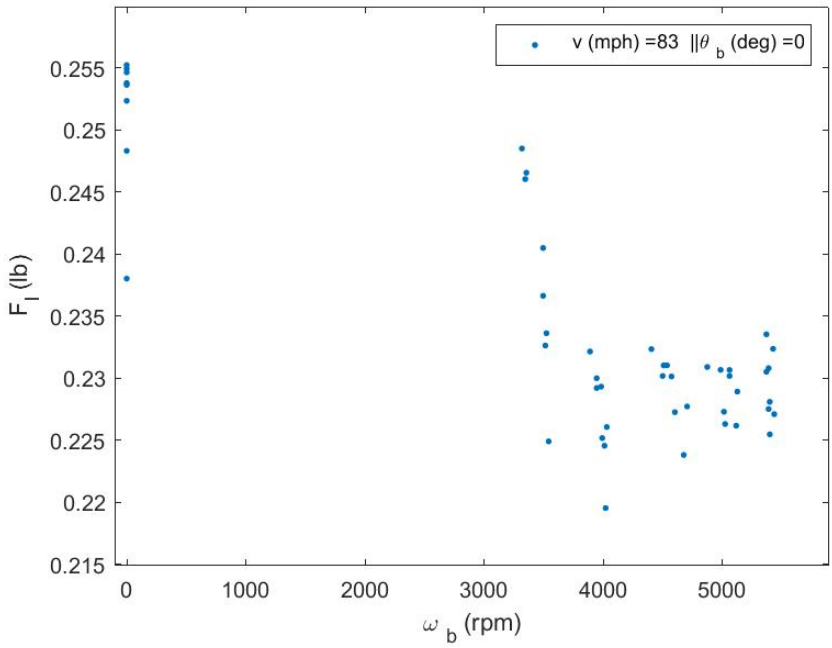
The random error based on repeateability data can be calculated by determining the standard deviation of repeated data. Baseball forces were taken multiple times with all input variables held constant, and ball spin rate varied. Figure 5.27 shows this data for each force axis, and standard deviations can be calculated from these results. This gives an average standard deviation of 0.0069, 0.0102, and 0.0043 [lb] for F_s, F_d, and F_l respectively.



(a) Side Force



(b) Drag Force



(c) Lift Force

Figure 5.27: Experimental repeatability trial. Variables held constant at v = 83 [mph], $\theta_b = 0$ [deg] $\theta_y = 90$ [deg], $\theta_z = 0$ [deg]. Note raw data used, no calibration applied.

From these two sources of error, equation 5.3 can be used to get the total amount of bias for a given force. This calculation gives the total bias to be 0.1194, 0.1227, 0.3793 [lb] for F_s, F_d, and F_l respectively. Most of the error was from the load cell bias, and a more accurate load cell would be the primary way to lessen this total bias.

$$B_t = B_s + B_d \tag{5.3}$$

5.7 Future Work

This study could be extended in several ways. One includes expanding the experimental and computational data set. This could involve taking a greater density of data, as

well as making an apparatus to allow for a greater range of input variables to be obtained in the wind tunnel. These would require fabrication of new apparatuses for the baseball, and a significant amount of computational power to get a marked increase in data density and data breadth.

Obtaining more physics results from the computer model could also be explored. This would include taking more measurements at a large amount of points around the baseball. This data could then be added to a physics based machine learning algorithm to increase system understanding.

Exploring further refining of the machine learning algorithm is also possible. More permutations of feature space variables could be explored to get an ideal feature space. This could help inform which physics based variables are most important to describing ball behavior. New feature spaces could also include new physics variables which are combinations of existing variables. This would be similar to making a Reynolds number, but out of baseball physics variables.

Conclusion

In this study, a baseball pitch was experimentally and computationally modeled to inform a physics based machine learning algorithm in an attempt to understand this complex system more accurately.

First, a baseball pitch was investigated experimentally. This was done with an apparatus mounted in a wind tunnel that allowed the baseball input variables (seam orientation: Y axis (θ_y), seam orientation: Z axis (θ_z), sting angle (θ_b), spin rate (ω_b), and air velocity (v)) to be controlled. Force data was gathered for a wide range of input variables, which was used to understand the behavior of the baseball. A limited range of input variables was explored due to physical limitations inherent to the experimental setup and the curse of dimensionality associated with combinations of input variables.

The baseball system was then modeled computationally in an attempt to replicate the experimental setup. Using the CFD software COMSOL. All experimental trials were replicated by the computer model, as well as some trials that were not performed experimentally.

The data was then examined to see if either the experimental or computational data was adequate to describe the baseball system. The experimental data pointed to interrelated behavior associated with the input variables. It also showed that only a small range of input variables were able to be experimentally investigated, leaving large gaps in experimental data. Both of these findings pointed to the experimental data being unable to describe the system adequately.

The computer model had different shortcomings when it came to describing the system. While a larger breadth of input variables examined, the forces were often not of the same magnitude as the equivalent experimental forces. The computational

results did typically produce the same trend of behavior for the baseball. This meant that even though a larger breadth of data could be gathered for the baseball system, the inaccuracies associated with the results meant that the computer results alone could not adequately describe the baseball system. The computational model did have the added benefit of being able to produce additional results such as pressure and velocity at a specified point. While these physics based results are not directly indicative of ball movement, they can be used later in a machine learning algorithm to bolster a feature space.

These shortcomings in both the experimental and computational data showed that the baseball system was a complex system, and was unable to be explained by any one approach. This led to the physics based machine learning algorithm, which attempted to combine both experimental and computational data to make a better predictive model of the baseball system. Adding together both experimental and computational data points, as well as the potential for physics based input variables from the computer model, allowed for a physics base machine learning algorithm to be made. The addition of computational results and accompanying physics based variables typically increased the predictive ability of the algorithm. The addition of computer data and additional physics based variables allowed for both and increase in general predictive ability and predictions in the potential gaps in experimental data.

This points to a physics based machine learning algorithm had the ability to combine data and physics based variables to make a better predictive model of a complex system.

References

[1] Pouya Jalilian, Patrick K. Kreun, Mohammad Hady M. Makhmalbaf, William W. Liou. *Computational Aerodynamics of Baseball, Soccer Ball and Volleyball.* American Journal of Sports Science Vol. 2, No. 5 pp. 115-121,(2014).

[2] Tinoco R , Goldstein E , Coco G. *A data-driven approach to develop physically sound predictors: Application to depth-averaged velocities on flows through submerged arrays of rigid cylinders.* Water Resources Research, 51(2), 1247-1263, (2015).

[3] Princeton University. *Fluid Mechanics.* The Trustees of Princeton University, www.princeton.edu/ eturkoz/fluids.html.

[4] Latchman Richie, et. al. *Conventional Swing of a Cricket Ball Using COMSOL Multiphysics.* Proceedings of the 2015 COMSOL Conference in Boston(2015).

[5] Takatoshi Higuchi, et. al. *The Effect of Fastball Backspin Rate on Baseball Hitting Accuracy.* Journal of Applied Biomechanics, 29, 279-284 (2013).

[6] COMSOL Staff, *The Magnus Effect.* `https://www.comsol.com/model/the-magnus-effect-18527`, 2015, accessed 2018.

[7] Wikipedia, *Drag coefficient of a sphere as a function of Reynolds number.* `https://www.upload.wikimedia.org`, 2014, accessed 2019.

[8] Rob Jennings, *Bucknell Thesis 2019.* Github Repository `https://github.com/rjj008/Bucknell_Thesis_2019`, 2019, accessed 2019.

[9] Swischuk, Renee, et. al. *Projection-based model reduction: Formulations for physics-based machine learning.* Computers and Fluids, 2018, accessed 2019.

[10] Dazhong, Wu, et. al. *Prediction of Material Removal Rate for Chemical Mechanical Planarization Using Decision Tree-Based Ensemble Learning.* Journal of Manufacturing Science and Engineering, 2018, accessed 2019.

[11] Atkinson, Chris, Mott, Greg. *Dynamic Model-Based Calibration Optimization: An Introduction and Application to Diesel Engines*. SAE Technical Paper Series, 2005, accessed 2019.

[12] Nyberg,Stephen. *Baseball CAD*. `https://grabcad.com/library/baseball--3`, GrabCad, 2012, accessed 2018.

[13] Stanford Staff. *Multi-Layer Neural Network*. `http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/`, Stanford, accessed 2019.

[14] Barlow, Jewel, Rae, William, Pope, Alan. *Low-speed Wind Tunnel Testing*. 1999, accessed 2018.

[15] Hernández, Miguel, et al. *Design Methodology for a Quick and Low-Cost Wind Tunnel*. Intech, 2013, accessed 2019.

[16] Brahma, Indranil. *Extending the Range of Data-Based Empirical Models Used for Diesel Engine Calibration by Using Physics to Transform Feature Space*. SAE Int. J. Engines, 2019, accessed 2019.

Appendix A

Colm 1	Colm 2	Colm 3	Colm 4
Side Force (lb)	Drag Force (lb)	Lift Force (lb)	Seam Orientation: Y axis (deg)

Colm 5	Colm 6	Colm 7	Colm 8
Seam Orientation: Z axis (deg)	Air Velocity (mph)	Wind Tunnel Angle (deg)	Ball Spin Rate (rpm)

Colm 9	Colm 10	Colm 11	Colm 12
Wind Tunnel Input (Hz)	Ball Motor PWN	Raw Force X axis (lb)	Raw Force Y axis (lb)

Colm 13	Colm 14	Colm 15	Colm 16
Raw Force Z axis (lb)	Temperature (C)	Side Coefficient	Drag Coefficient

Colm 17	Colm 18	Colm 19	
Lift Coefficient	Experimental Boolean	Reynolds Number	

Table 5.5: Data order for experimental and computational results.

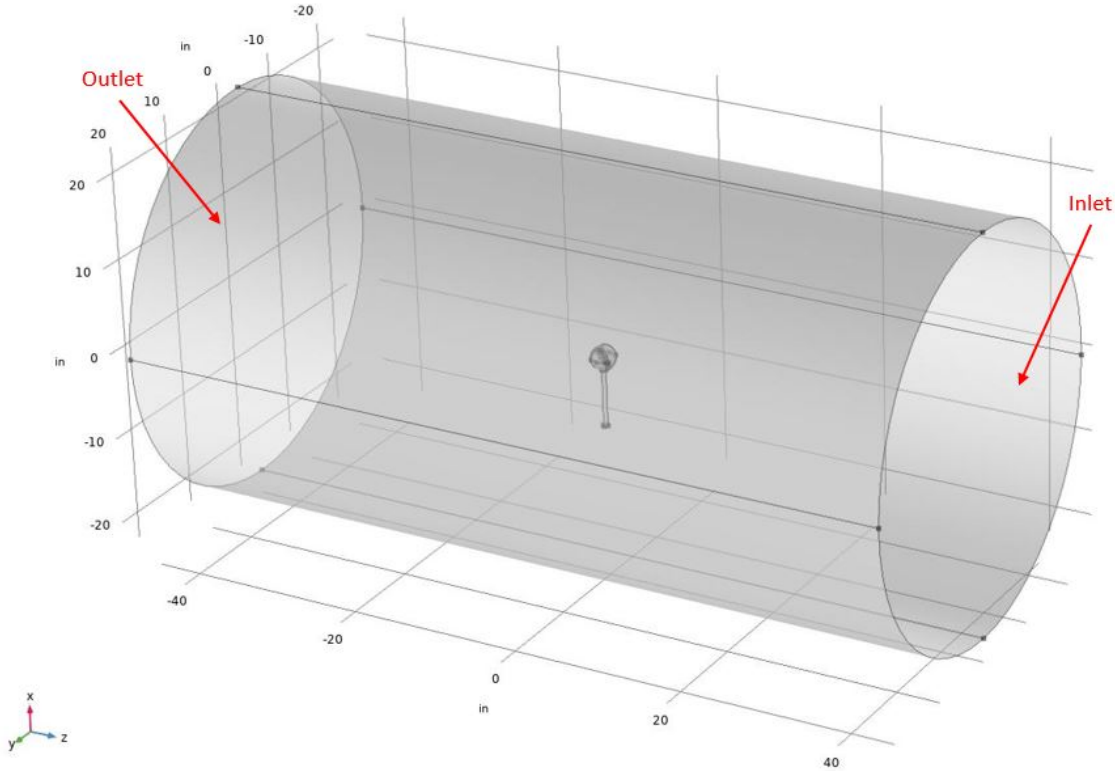
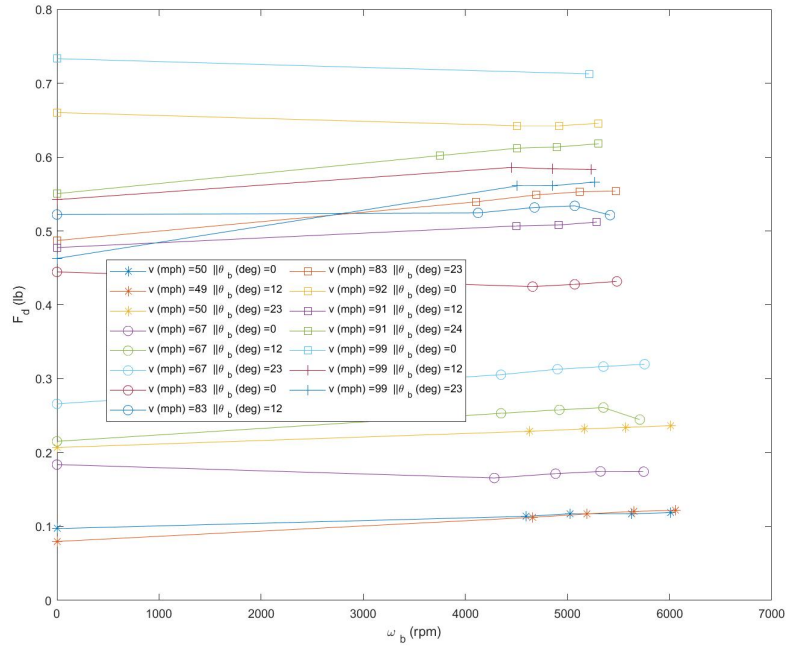
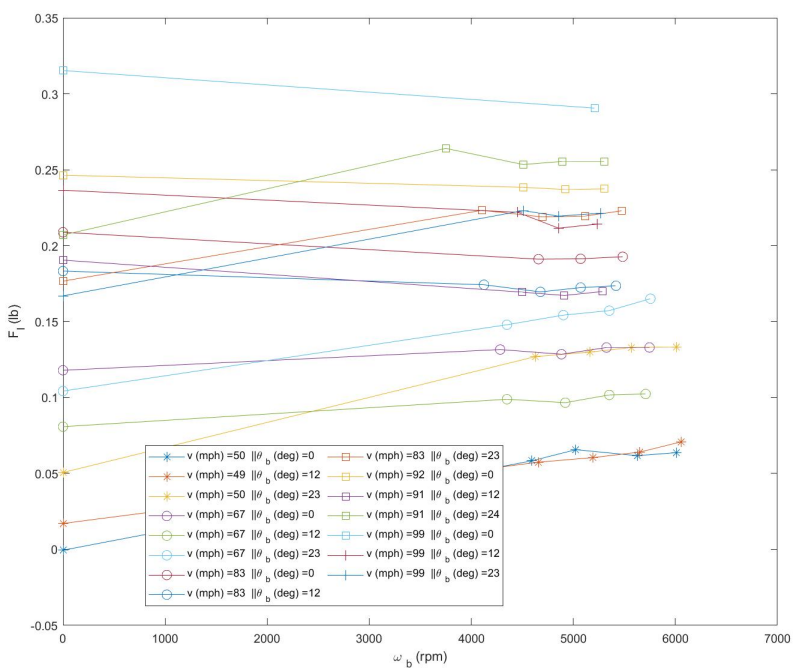


Figure 5.28: Baseball Computational Volume Setup. Orientation axes shown.

Figure 5.29: Inclinometer used to get sting angle.



(a) Drag Force



(b) Lift Force

Figure 5.30: Spin rate effect on drag and lift force. Variables held constant are $\theta_y = 90$ and $\theta_z = 0$.

Figure 5.31: Baseball Rotation Shaft used to spin baseball about a specific axis. This shaft was held inside the sting with two rotation bearings.

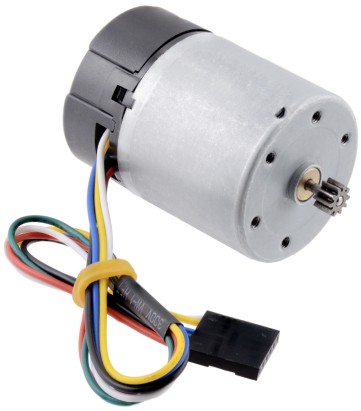
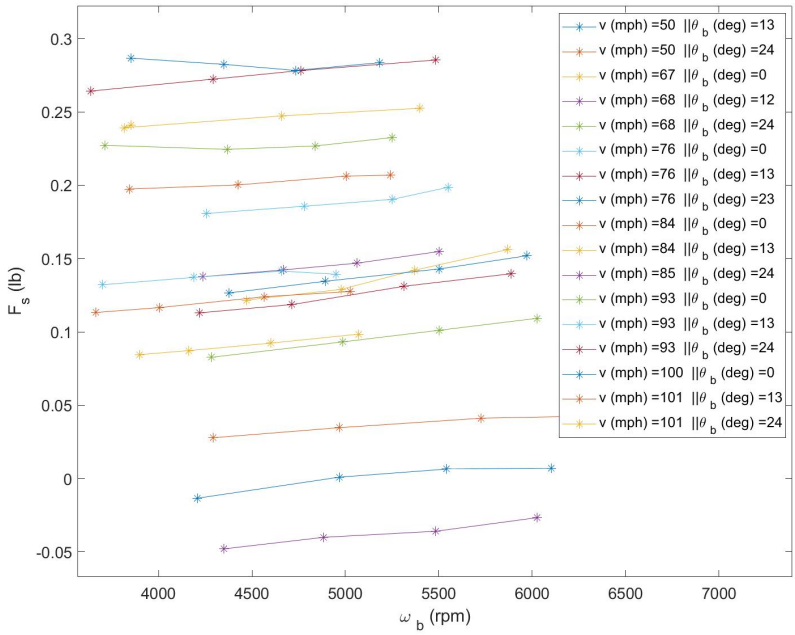
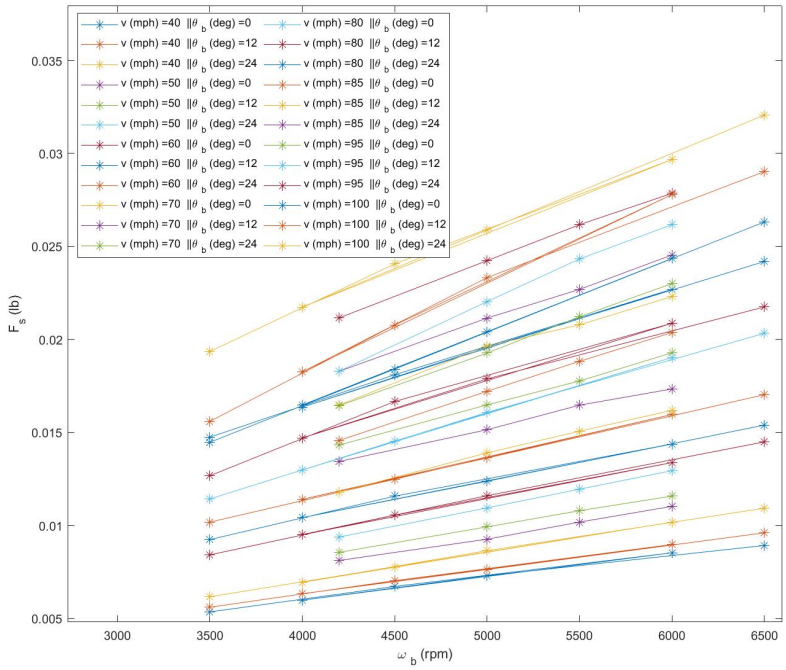


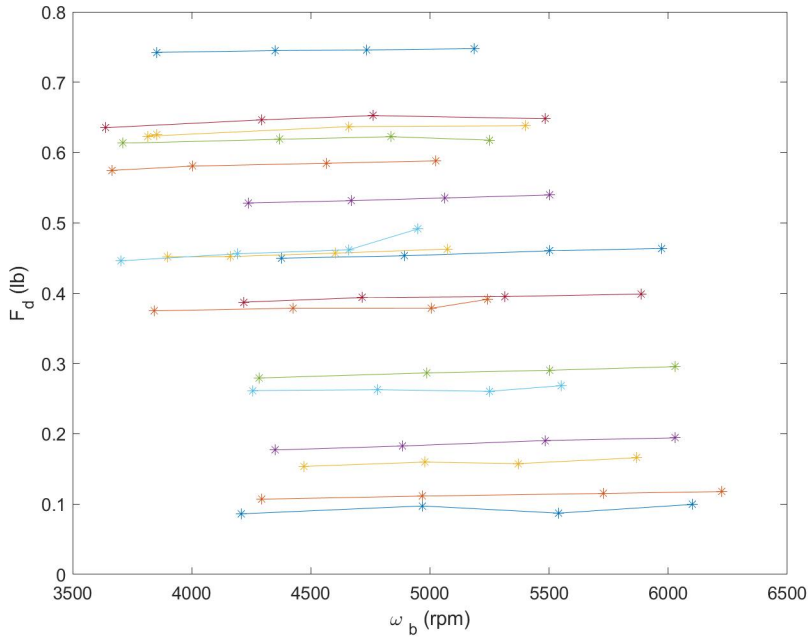
Figure 5.32: Motor with mounted encoder used to spin baseball and record baseball spin rate. Note that the output shaft gear was removed.



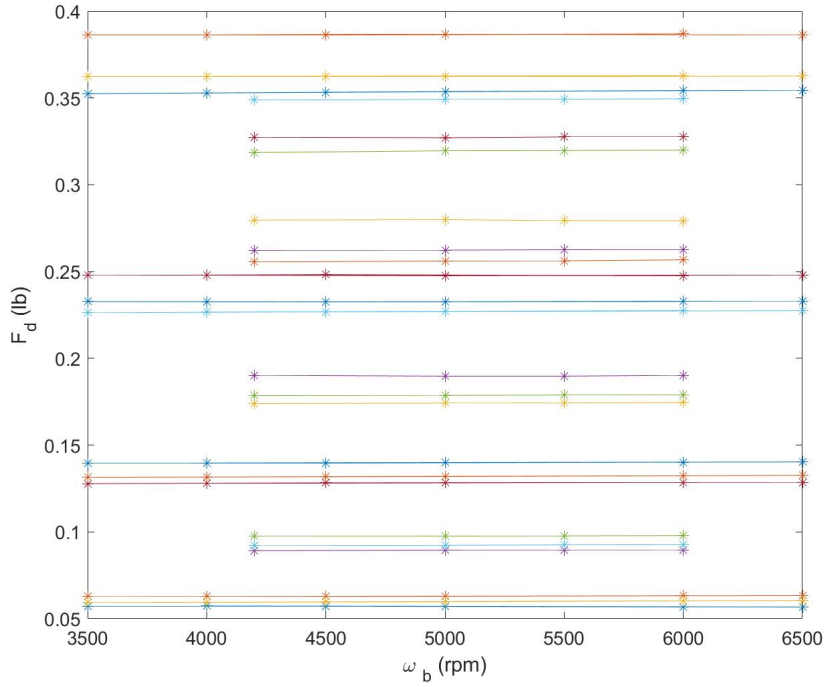
(a) Side Force - Experimental



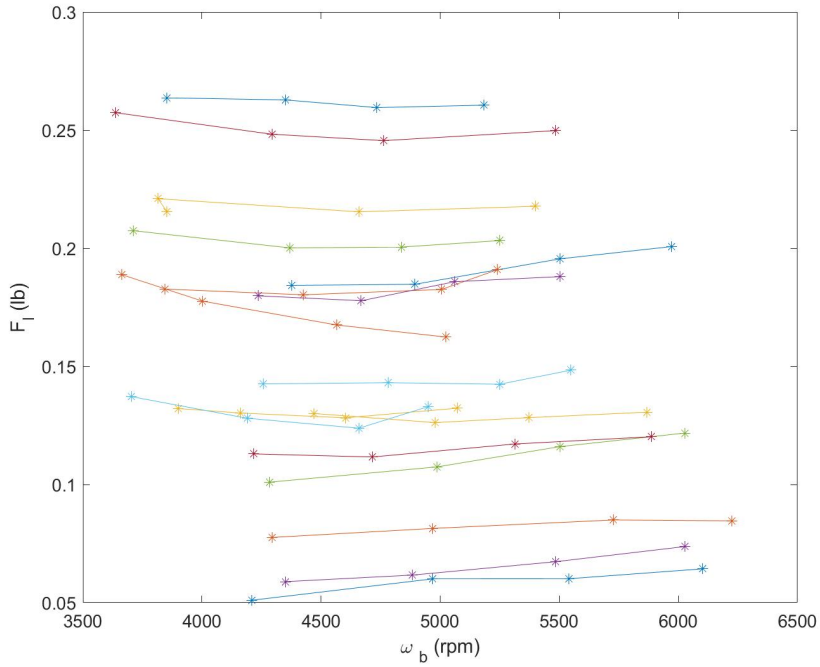
(b) Side Force - Computational



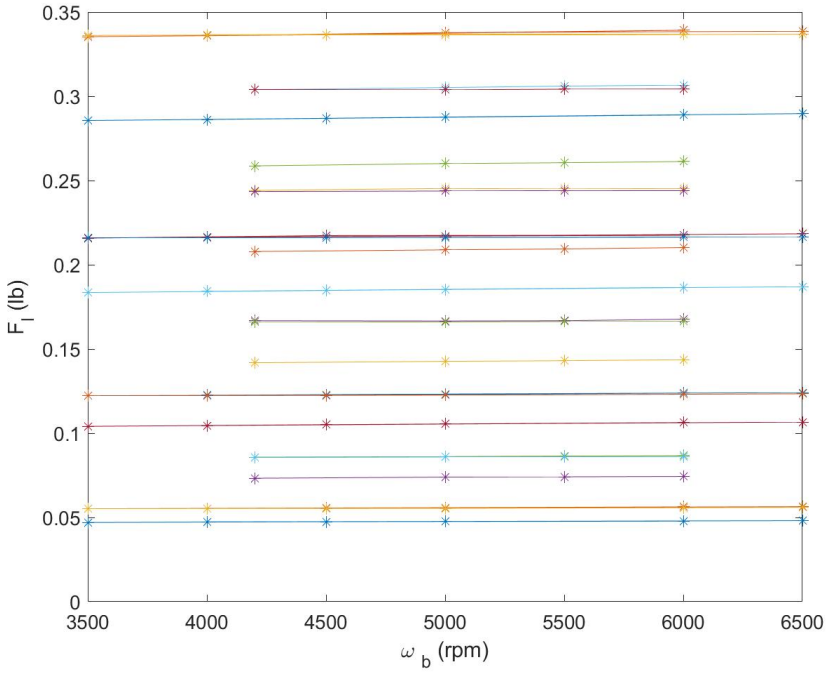
(c) Drag Force - Experimental



(d) Drag Force - Computational



(e) Lift Force - Experimental



(f) Lift Force - Computational

Figure 5.33: Comparing experimental and computational effect of ball spin rate, ω_b, on baseball forces. Variables held constant are $\theta_y = 90$ and $\theta_z = 0$. Legends are the same for experimental and computational data respectively.

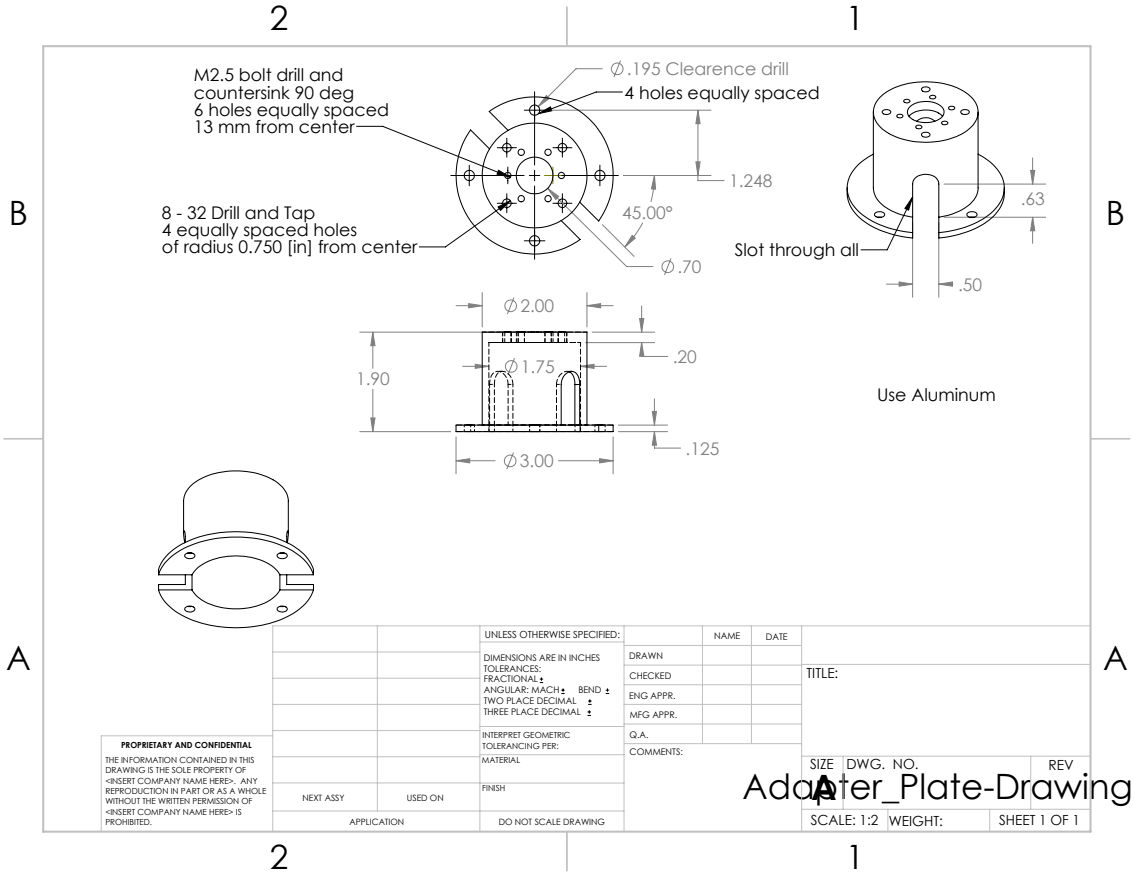


Figure 5.34: Top Adapter Plate drawing.

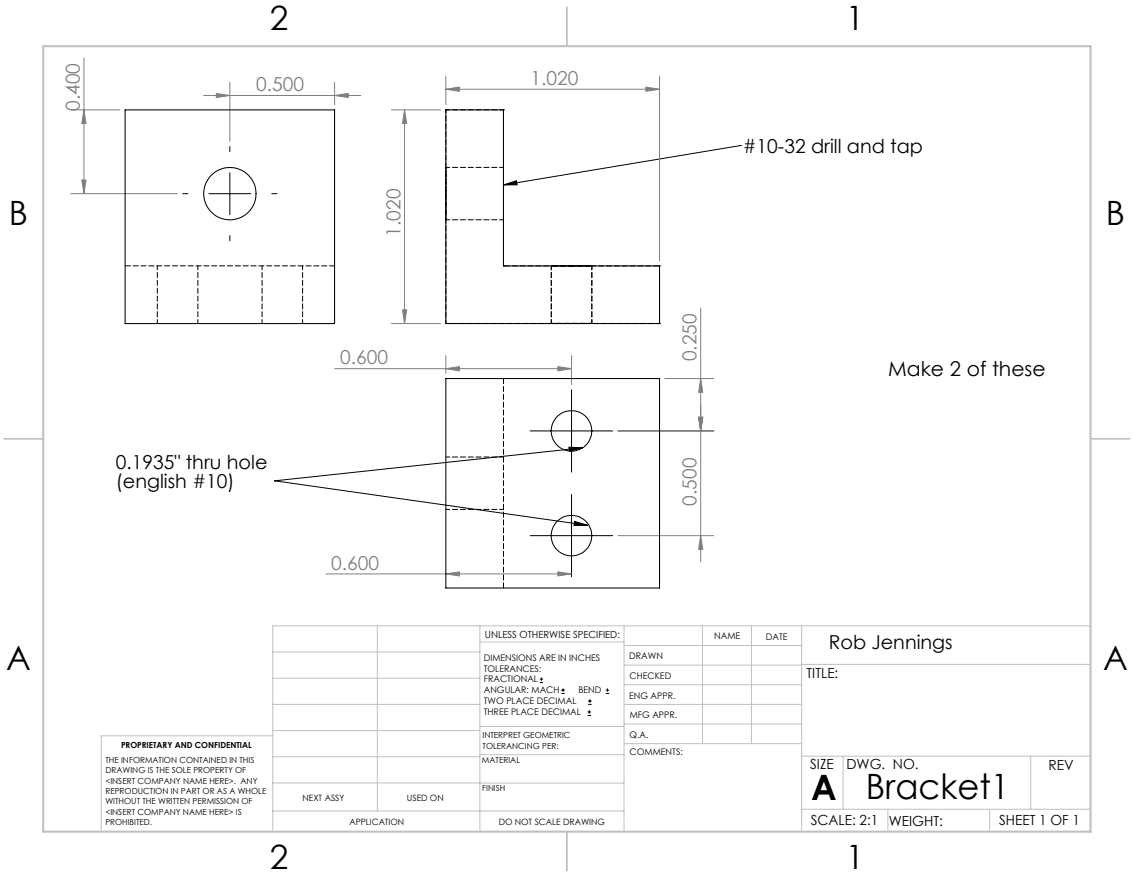


Figure 5.35: Apparatus Bracket drawing.

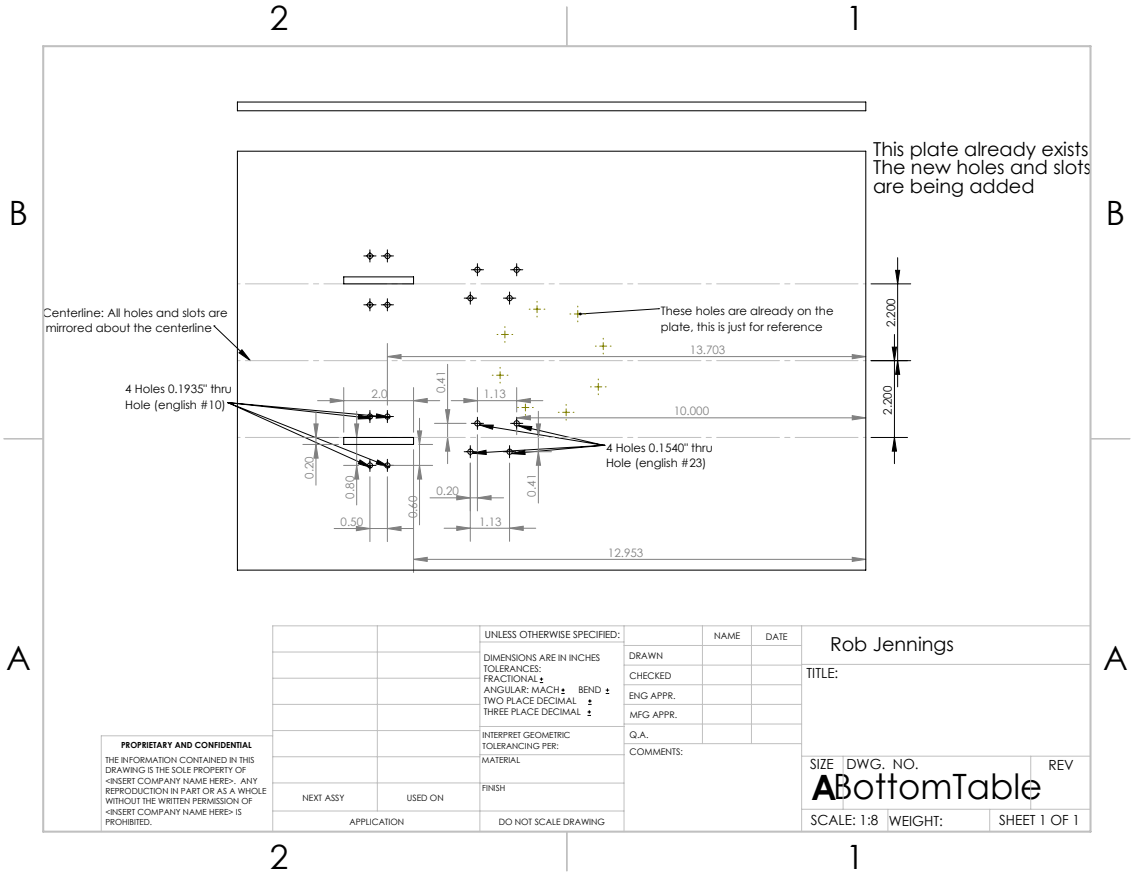


Figure 5.36: Bottom Table drawing.

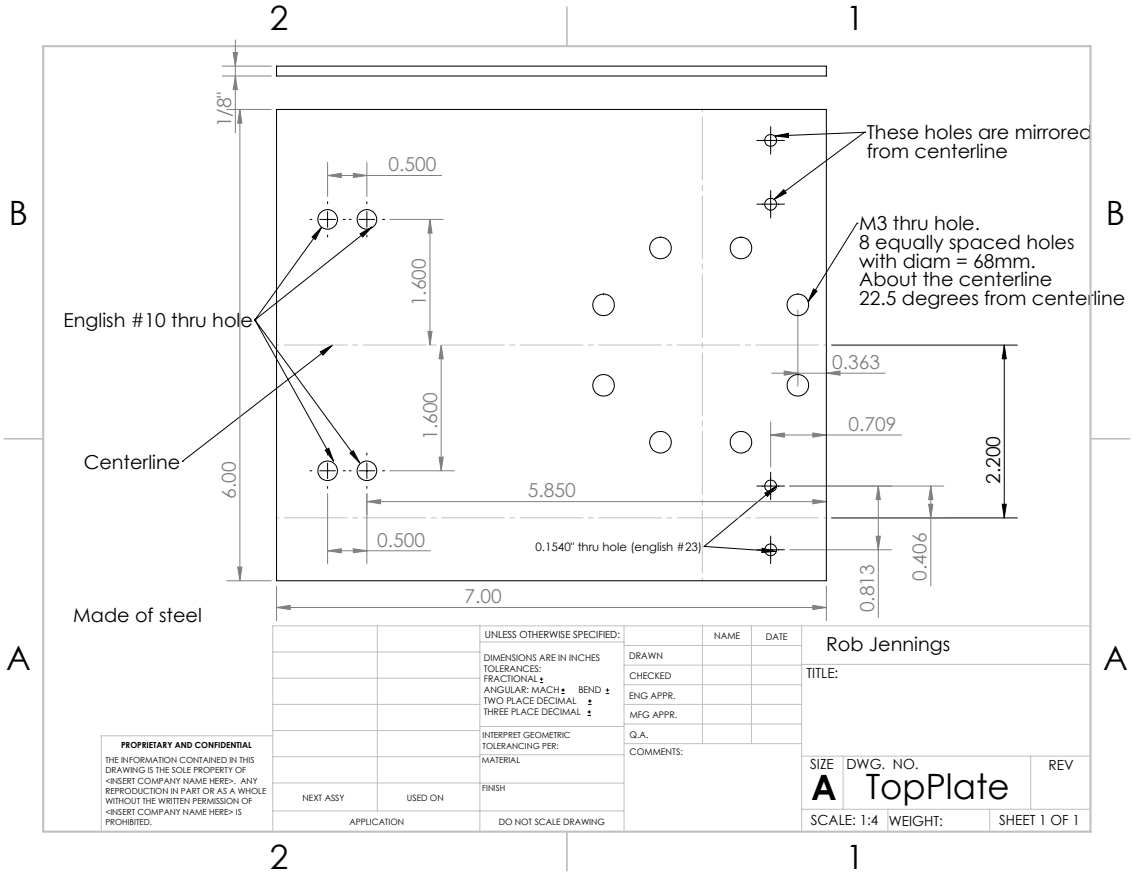


Figure 5.37: Apparatus Top Adapter Table drawing.

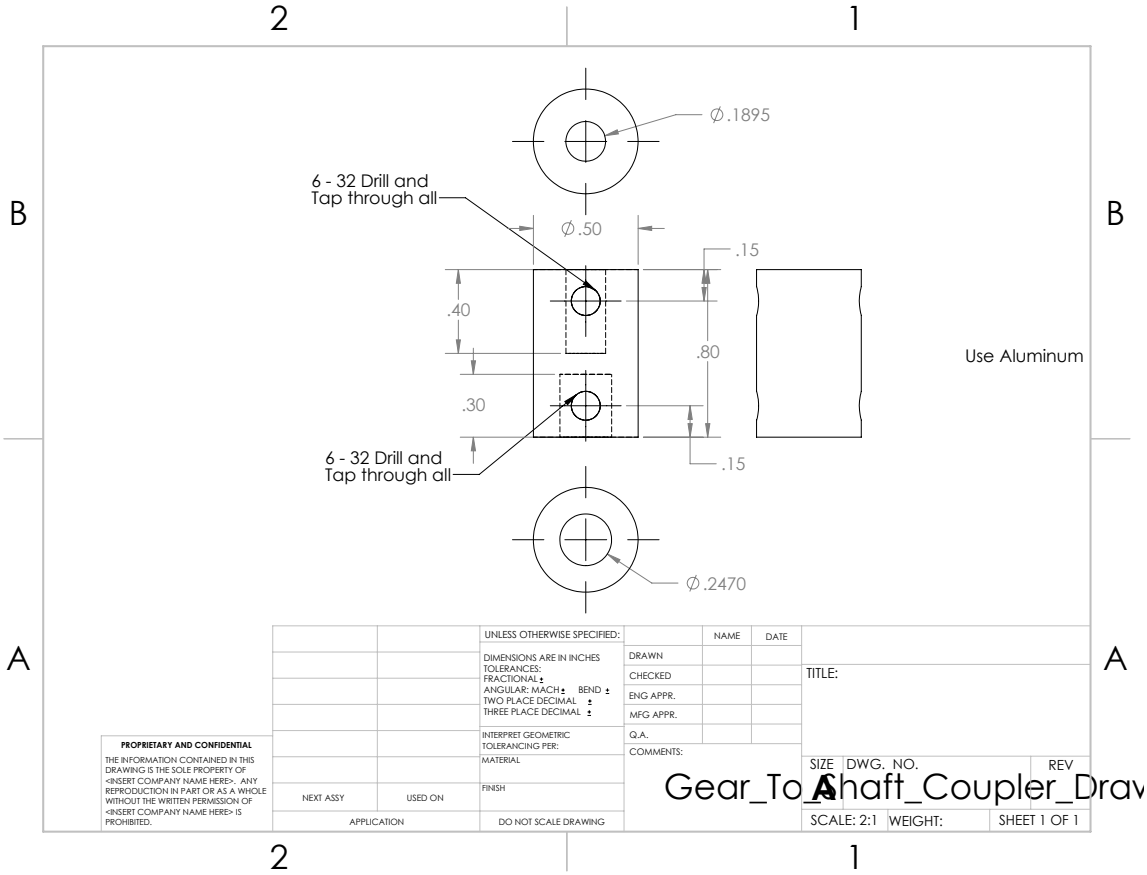


Figure 5.38: Gear To Shaft Coupler drawing.

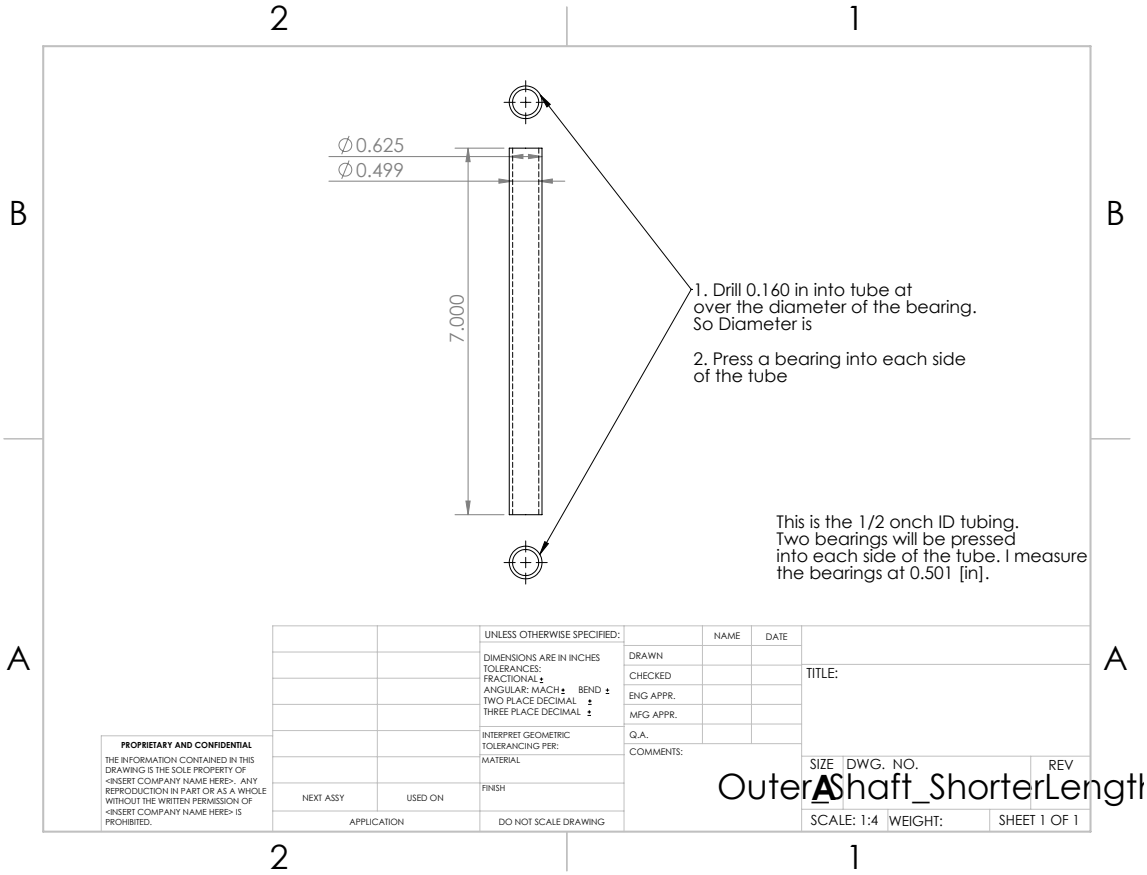


Figure 5.39: Sting Outer Shaft drawing.

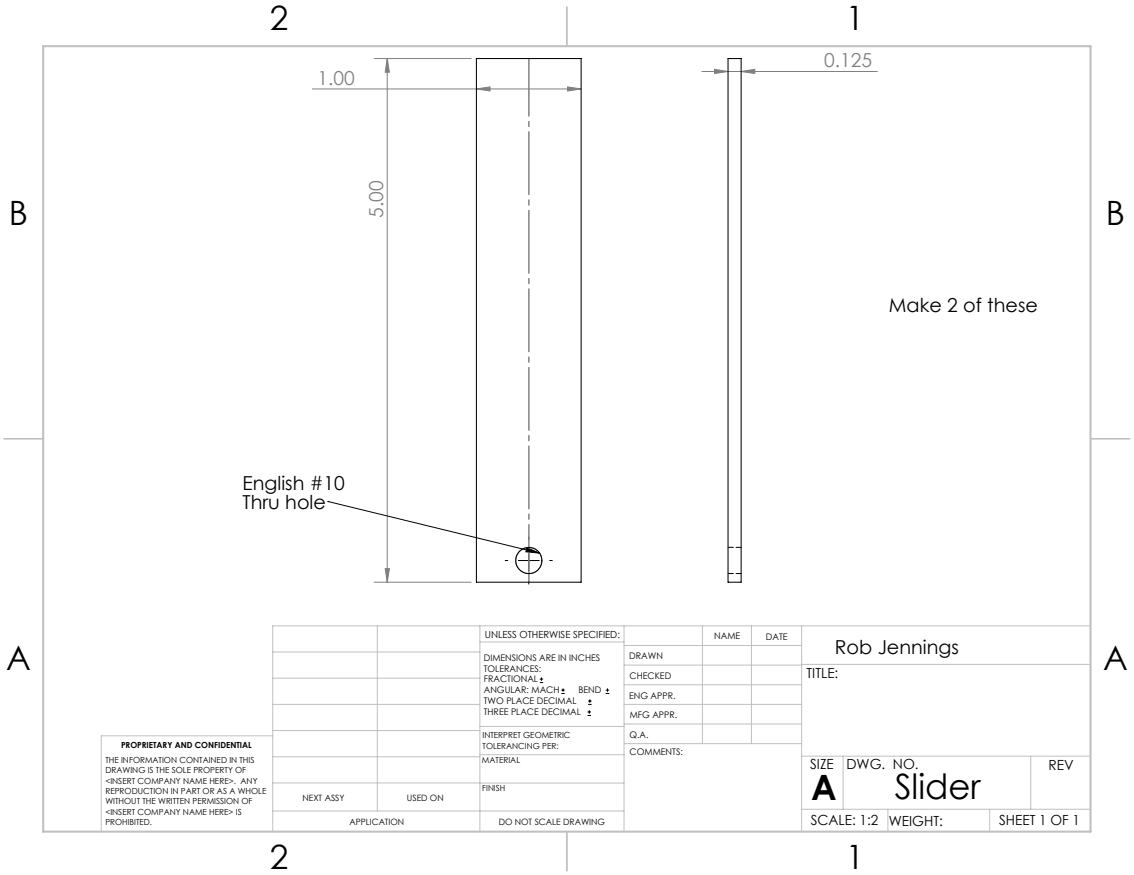


Figure 5.40: Apparatus slider drawing.

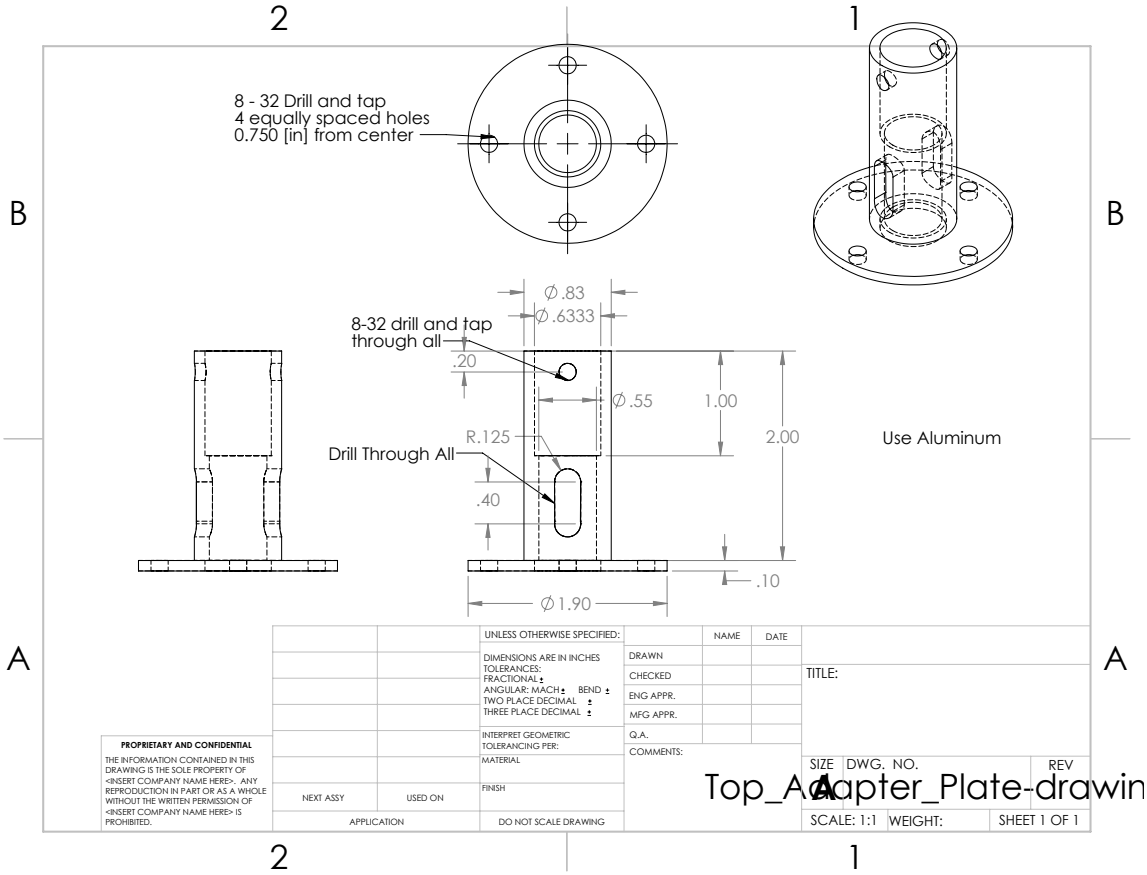


Figure 5.41: Apparatus Top Adapter Plate drawing.