AN INDUSTRIAL-GRADE CYBER-PHYSICAL PLATFORM FOR INTRODUCING MACHINE LEARNING CONCEPTS

by

Dylan Imbus

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

School of Engineering Technology

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Grant Richards, Chair

School of Engineering Technology

Dr. Huachao Mao

School of Engineering Technology

Dr. Gaurav Nanda

School of Engineering Technology

Approved by:

Dr. Duane D. Dunlap

School of Engineering Technology

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

K-values A set of coefficients that weigh the PID factors and are adjusted to influence controller response.

Rise Time The time a controlled system's output takes to ascend from ten percent to 90 percent of the desired setpoint.

Settling Time The time from the beginning of a control cycle until stabilization.

Stabilization The point which a controlled output is maintained between two percent of the setpoint for a defined period.

Percent Overshoot The maximum output a controlled system reaches divided by the systems setpoint.

LIST OF ABBREVIATIONS

EP – Evolutionary Programming

GP – Gaussian Process

IAE - Integral Absolute Error

IoT – Internet of Things

IR4 – The Forth Industrial Revolution

ISE – Integral Square Error

IT – Informational Technology

ITSE – Integral Time Squared Error

ML – Machine Learning

MSE – Mean Squared Error

OT – Operational Technology

PID – Proportional Integral Derivative

PLC – Programmable Logic Controller

PSO - Particle Swarm Optimization

RMSE – Root Mean Squared Error

SVM – Support Vector Machines

STEM – Science, Technology, Engineering, & Math

VFD – Variable Frequency Drive

ZNST – Ziegler Nicholas Strategy

ABSTRACT

Industry 4.0 holds many promises for manufacturers; however, a shortage of qualified employees has prevented a swift adoption of the revolution's new technologies. Engineer and Economist Klaus Schwab argues Education 4.0 is the key to addressing the employee shortage and preparing future generations for the shifting labor market. To support Education 4.0, classes must allow students to engage emerging technologies that help bridge Operational Technology (OT) and Informational Technology (IT). The thesis detailed an educational laboratory that demonstrates the application of data analytics (an IT tool) and optimize the performance of a cyber-physical system composed of industrial (OT) components. The lab experience focuses on a disc's controlled positioning (levitating) using a PLC-based PID controller and a VFD. The activity requires students to capture data of a moving discs, create a machine learning function representing the disc's movement, and use the machine learning function for classification and PID optimization problems. A comparative analysis of a PID cycle ensures a regressions model accurately represents the physical model using measurements including peak-overshoot, rise time, settling time, and the flight plots' Means of their Squared Error. Further, the study examines multiple ML models each built using various features to identify the systems relevant and redundant data.

CHAPTER 1. INTRODUCTION

The Fourth Industrial Revolution, termed Industrie 4.0 (IR4), holds great promise to manufacturers; however, challenges implementing the technologies have prevented a swift realization of its benefits. A 2018 study conducted by the World Economic Forum and their partner Mckinsey & Company predicts IR4 will create up to $3.7 trillion of new sales revenue through 2025 (Leurent, 2018, p. 4). The income will derive from Smart scheduling and optimized processing enabled by the nine pillars of IR4. Technologies like the Internet of Things (IoT) and Big Data analytics provide capabilities to optimize processes by limiting quality defects, solving production flow issues, and preventing excessive waste. The same technologies allow for Smart scheduling enabled by predictive maintenance and predictive ordering.

The benefits of IR4 also go beyond monetary value, as the revolution promises to bring higher quality jobs and stronger customer relations. Higher quality jobs come as advanced systems create a demand for programming and data science employees. Further, the technologies can eliminate or mitigate assignments that require repetitive motion, lifting heavy objects, hazardous material interactions, and work in dangerous environments. Customer relations will strengthen as companies gain the ability to meet more customer requests. Satisfied requests could include lower costs, on-time deliveries, and production demands like small batch sizes and higher ensured product quality.

Companies have yet to implement these new technologies and practices despite their promise of substantial benefit. The manufacturing industry often cites equipment costs, incompatibility, and a lack of employee expertise as barriers to adoption. A 2019 study conducted by Deloitte and MAPI included a survey of 600 US manufacturers and executive focus groups (Wellener, 2019, p. 3). The study confirmed that executives believe IR4 has potential, as "85 percent of respondents believe Smart Factory initiatives will be the main driver of manufacturing competitiveness in 5 years" (Wellener, 2019, p. 5). However, the study also shows the challenges of its implementation. The report finds only 49 percent of the respondents have not planned any initiatives, and only 3 percent have achieved full-scale smart factory adoption (Wellener, 2019, p. 3). One of the biggest obstacles of transitioning to a Smart Factory (a factory with full IR4 technology integration through all levels of the company) is the vertical integration that converges Informational Technology (IT) and Operational Technology (OT). A 2014 blind study conducted by Cisco surveyed 1230 IT and OT executives from several industries to understand the current industry environment (Noronha, 2014, p. 4). IoT is a fundamental pillar of IR4, serving as the spinal cord that collects and transfers data. The Cisco survey found that the most common need from vendors progressing toward IoT installation was strategic planning (Noronha, 2014, p. 16). A need for strategic planning shows companies seem to lack an understanding of how to establish technologies' use into an integrated multi-level model. Another hesitation that keeps companies from capitalizing on IR4 is the reward/risk ratio. Deloitte and MAPI's study quoted one executive regarding the challenges of the adoption of smart factories. Claiming, "in a production environment, making a mistake can bring a production line down or harm a human worker, so the stakes are very high" (Wellener, 2019, p. 16). The executive proves that companies do not feel qualified to shift to new technology out of fear of potentially putting an operational facility at risk.

Schools are needed to help build the workforce pipeline by preparing students to implement and utilize the new technologies to push IR4 into full adoption. Colleges have long been a source for employers to find employees with unique talents and fresh ideas. "*Education 4.0 Made Simple: Ideas For Teaching,*" written by Anealka Hussin, describes strategies for implementing Education 4.0 topics (2018). Schools can aid in developing a workforce capable of adopting IR4 by presenting IoT and Big Data Analytics applications to students through hand on learning experiences. Labs like optimizing a PID controller for a levitation system allows students to see how OT and IT converge as the experience brings the two worlds together on a small scale. A levitation system also provides an opportunity to introduce Big Data Analytics, like Machine Learning (ML), through classifications problems and applying ML-based optimization strategies.

1.1 Problem

Engineer and economist Klaus Schwab has been a strong voice in the discussion about the future of IR4. Since 2015, he has argued that the dramatic technology changes require support through education advancements capable of preparing students for work in the field (Schwab, 2016). Schwab acknowledges two conflicting scenarios possible to arise from IR4, each to benefit from the education system. One bolsters higher quality jobs and economic growth through dropping trade costs dropping due to increased business efficiency and productivity (Schwab, 2016). The scenario depends on the education system to supply adequate employees to fill positions and use their ingenuity to drive business decisions.

On the other hand, Schwab mentions a less favorable outcome which involves job losses due to automated machines, leading to greater inequality and social tensions (Schwab, 2016). But again, education systems can help bridge the gap between socioeconomic classes and provide a means to enter the technology workforce. With schools holding a vital role in the future of IR4, universities need to maintain a strong influence on the revolution by providing students with up-to-date laboratories and skill development.

The thesis addressed the need for introducing new IR4 technologies into an introductory industrial controls course. Incorporating emerging technologies into underclass coursework provides a platform for showing students how Operation Technology and Information Technology converge in an industry environment, showcasing how control systems can integrate data analytics, like ML. Further, students need a creative way to engage with the technology that may come with a traditional class. Multiple STEM studies have found theoretical based courses correlate to lower grades, academic interest, student confidence, and retention rates, all of which point to a lower quality student experience (Coleman, 2019; Wladis, 2015). One study found that implementing active learning increased "student performance on examinations and concept inventories" by 0.47 SDs (Freeman, 2014, p. 8410).

1.2 Significance of Problem

Klaus Schwab emphasizes STEM skills to prepare future generations for the IR4. In recognition of STEM skills' future demand, in 2012, the US President's Council of Advisors on Science and Technology called for a 33% increase in STEM bachelor's degrees (Freeman, 2014). A subsequent global study evaluated 25 countries based on 52 indicators to gauge the preparedness for the coming wave of automation (McCauley, 2018, p.10). The study put a particular emphasis on STEM skills and argues, "few countries have begun to address the impact of automation through educational policy" (McCauley, 2018, p.10). Alp Ustundag, author of *Industry 4.0: Managing the Digital Revolution* (2018), points out that the demand for technology skills has already created an employee shortage and an intensified competition for skilled workers. Ustundag furthers that the lack of well-established paths for learning the skills and technologies of IR4 is a primary contributor to the shortage.

Purdue's School of Engineering Technology has a long history of academic offerings supporting those entering the manufacturing industry. Multiple laboratories equipped with industrial grade devices commonly found in manufacturing facilities support the delivery of applied instruction. The Furnas Foundation Controls Lab is a vital element of the process as it serves multiple classes, including Introduction to Industrial Controls and Manufacturing Integration. The introductory controls course alone hosts over 260 students a year. It is a required course for several majors, such as Mechanical Engineering Technology, Electrically Engineering Technology, and Manufacturing Technology. In addition, students in degree programs outside the School of Engineering Technology regularly enroll in the course to gain operational experience with devices and systems.

1.3 Purpose

The work details the development of a learning platform and process capable of allowing students to explore the application of data analytics, a critical IR4 technology, on an industrial-grade system through the tuning of a PID loop to control the height of a disc hovering in the air. A tangent study also explores how the system and associated model can use ML for simulation or part identification. A regression function reflecting the disc's movement allows for students to solve the optimal PID flight path. Solving the PID values allows for the disc's stability and enables the disc to reach a target height in a minimum amount of time. Learning the effects of PID coefficients and how to optimize a PID is essential in control engineering. Industries also use classification problems for image processing and predictive maintenance. The platform allows students to apply simplified classification strategies to gain an understanding of classification programs.

1.4 Significance of Purpose

As the Furnas Foundation Laboratory serves a large student population in multiple courses, a teaching platform has an opportunity to influence many students. The platform would provide real-world applications that apply both data analytics and technical equipment skills. Again, the cross between data science and engineering skills is precisely what many industries desire but struggle to find. Lee et al. (2017, pp.2-3) attribute the shortage to the idea that these types of employees do not exist when he says, "few experts exist who have domain knowledge on manufacturing components and the physics of these components, as well as professional knowledge and skills for big data analytics." A lab experience, as proposed, can help develop these employees and deliver them to starving industries.

1.5 Assumptions

For the study, the data analysis will assume all measurements are consistent and accurate. The assumption assumes that the physical setup will remain unaltered, having: the laser measuring to the same position on the disc, an undisturbed airflow to the blower, and the tube remaining vertical. Performance characteristic measurements used the nearest data outside the area of focus. For instance, for rise time measurements which require knowing when the disc is 3.5 and 31.5 inches high, the time interval was based on the nearest point at which the disc was at or below 3.5 inches and the nearest point the disc was at or above 31.5. The study assumes rounding to the nearest value for any performance characteristic measurement does not affect the accuracy of the measurement, and therefore, is viable is comparisons purposes. Additionally, when calibrating the laser sensor, averaging the sensor's signals across various collections is expected to mitigate any noise captured in the laser sensor's analog signal.

1.6 Delimitations

The study created an introductory learning module universally suitable to multiple laboratory stations; however, all measurements collected in the thesis are from only one station. Inconsistencies amongst different workstations may exist, such as defects in the tube that affect the disc's movement. Discrepancies amongst the stations may cause the PID controller's optimal solutions to not necessarily translate to other lab stations. Nevertheless, the same optimization strategy presented here can calibrate any similar stations for predictable use.

1.7 Limitations

The experience is specific to equipment and software available to students in the Furnas Foundation Controls Laboratory during research activity. The OPT2011 laser sensor has a maximum range of three meters and would require an alternative sensor for a system that ranges beyond the detailed model. Additionally, the sensor can read measurements at 2ms intervals; however, programming challenges prevented capturing the height at rates beyond 60ms. While 60ms was suitable for the study, parallel work may require faster update rates. The blower was also constrained to lifting objects heavier than .13 grams due the blowers limited output force. And cylindrical pieces shorter than 1 ½ inches rocked sideways and either get stuck inside the tube or drag on the tube's wall, increasing friction and complicating the function that detailed the disc's movement. The original project design considering levitating a tennis ball. However, minute adjustments of the VFD frequency caused the ball to move too volatile and created unreliable ML models. The detailed lab platform fails to accommodate all hoovering objects and those with complicated movement functions.

CHAPTER 2. REVIEW OF LITERATURE

2.1 Methodology of Review

The research sought to develop and evaluate a lab experience capable of introducing students to data analytics in industrial control systems applications using industrial-grade equipment. The study used a machine learning regression model to help optimize a hoovering disc's flight path by solving the system's optimal PID values. A second ML model will classify predictions for two discs based on how the discs move at a known height and VFD frequency.

Preliminary research for the study included reviews of journal articles, industry white papers, Ph. D. dissertations, Master's theses, and company presentations. Journal databases and web search engines inform initial searches. The selected databases focus on data science, electronics, general engineering/technology, and Ph.D. publications. Databases utilized for the search included Google Scholar, Scopus, and IEEE Xplore. While Google Scholar and Scopus provide a wide range of resources, IEEE Xplore was most valuable for its expansive selection of electrical engineering and computer science papers, all relevant to the paper's system integration and data analysis. Keywords used for the search included: Machine Learning, Applied Engineering, Cyberphysical Systems, Teaching/Learning Model, PID optimization, and Hovering/Levitation Device. Placing the keywords in quotations (quotations to generate results that explicitly include both words of a phrase. For instance, "Teaching Model") and tying them together in diverse combinations with the Boolean operator "and" helped narrow the search for relevant articles. Publications are limited to those published after 2016.

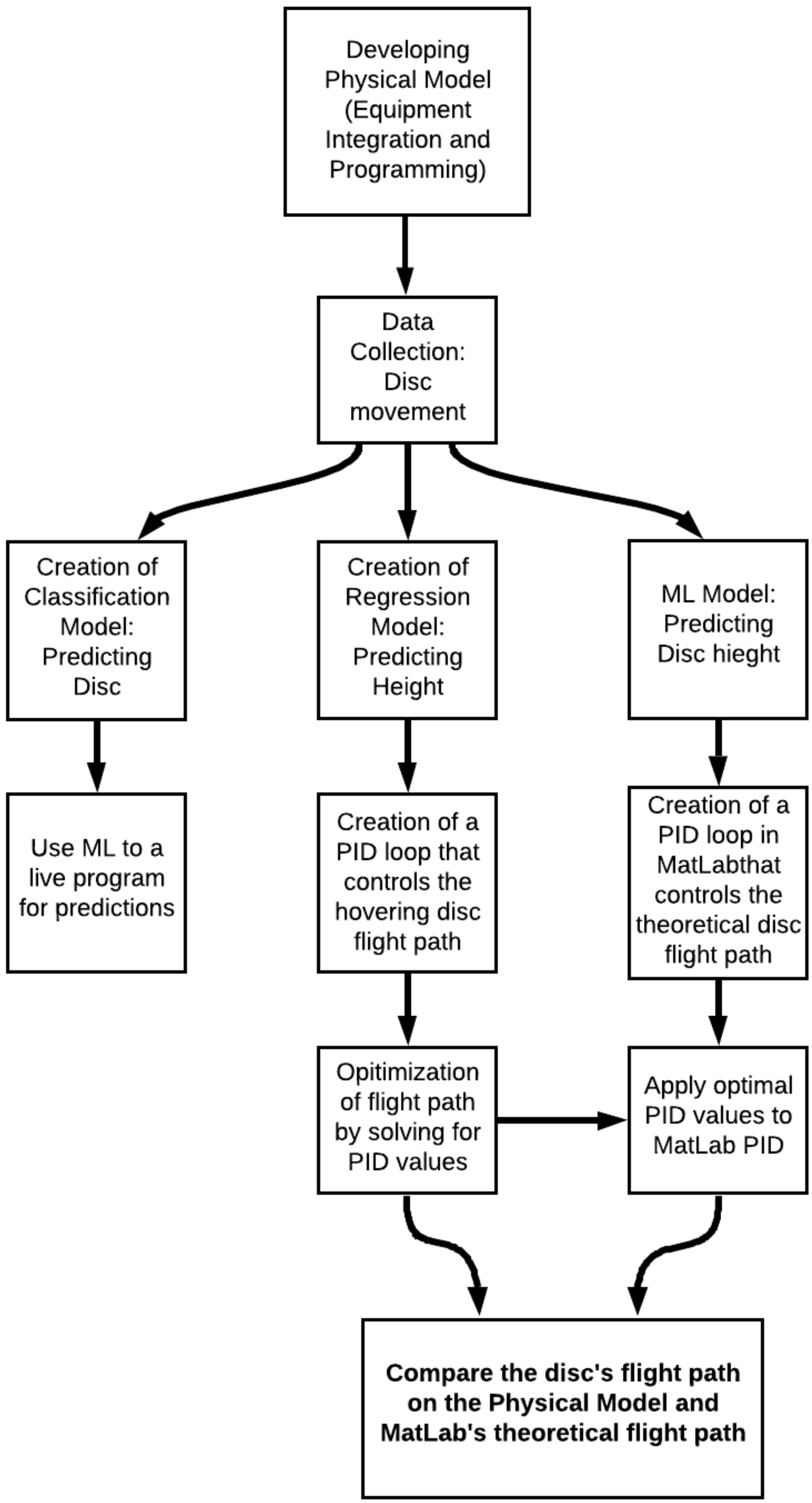


Figure 2.1 A concept map of the study

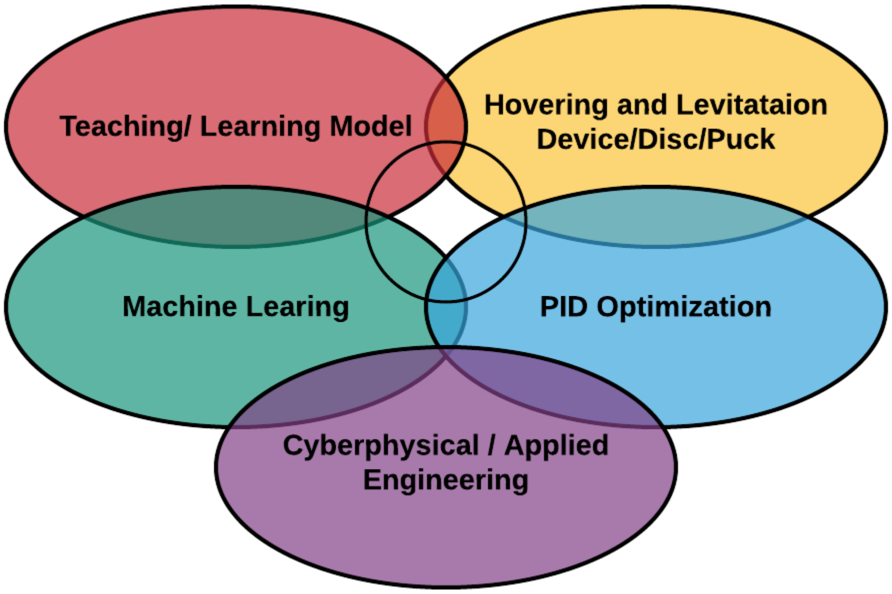


Figure 2.1 Keywords/Phrases use for the Literature Review

2.2 Review of Literature on the Problem/Purpose

There were three focus areas for creating the teaching module: PID control and optimization, ML regression models, and ML classification models. Multiple instructional experiences on controlling a levitation device have already been proposed and studied at an academic level. While these studies were unique in the components, software, and control strategies, they each utilized a closed feedback loop to control a blower to levitate an object, just as done in this study. The following sections will provide background regarding the three areas of focus for this study and how each of these areas will connect to a central theme of using a levitation system to incorporate ML principles.

2.2.1 PID Control and Optimization

Multiple parallel studies have optimized their control loops by solving PID values that optimize a flight path (Tootchi, 2019; Solihin, 2011; Paz-Ramos, 2004). PID optimization is a shared goal of this study. The teaching module will use ML algorithms to illustrate how data analytics can inform and aid in optimizing industrial control systems. In general, control loops are negative feedback, closed-loop systems that repetitively use measured values to calculate an output's deviation from a set value. The difference, or error, is used to generate a response to stabilize a parameter (for instance: height, position, trajectory, temperature, flow rate). A PID controller generates the output by using a sum of three separate response loops of reacting to changes in error (setpoint less actual position). These independent loops focus on proportional, integral, and derivative responses to a dynamically changing error value. Hence, the name PID. Based on the desired output response, weights are placed on the Proportional (P), Integral (I), and Derivative (D) calculations and represented as KP, KI, and KD. Tuning the controller can generate an ideal performance by solving the optimal *k* values. There are various optimization strategies capable of solving the *k* values to varying levels of success. These optimization methods include Ziegler Nichols (Ziegler, 1942), Cohen Coon (Cohen, 1953), and automatic tools like MATLAB's *pidtune* function that uses a unique underwritten algorithm. Each strategy uses a mathematic formula that interpolates the object's projected height based on a given height/velocity and applied response to solve the optimal *k* variables.

The expression representing the disk's height in the levitation chamber serves to optimize the coefficients of a PID controller tuned to stabilize the vertical position at a set point. The formula is generated from calculated theoretical forces or through experimental data of the object's movement. For a thesis at the University of Milan, Justin Kuzhandairaj (2018) also created an instructional experience based around a levitation device. Kuzhandairaj's system consisted of a ping pong ball and a slotted tube to contain the ball. Because his apparatus used a slotted tube, he used a Gaussian model to calculate airspeed due to the blower's airflow dispersing along the tube's length. The airspeed was a critical variable required to understand the ball's air pressure and how it would move. Jesus Chacon and colleagues at the University of Murcia and Ewelina Cholodowicz and Przemyslaw Orlowski at the West Pomeranian University of Technology each created similar levitation systems. They used solid tubes to channel ping pong balls, allowing them to regularly treat the air velocity (Chacon, 2017; Cholodowicz, 2017). Each of these studies, including Kuzhandairaj's, developed their ping pong ball movement's theoretical function by balancing a force equation and applying Bernoulli's principle. Kuzhandairaj (2018) would go on to use MATLAB's tuning toolbox to optimize his controller. Cholodowicz (2017) also valued MATLAB's virtual optimization tools, providing a potential learning phase for students through the software's manual tuning functions. Manually tuning would allow students to implement and test various PID coefficients and visually examine differences in the coefficient's effects (Cholodowicz, 2017). Cholodowicz (2017) went on to optimize her controller by applying a genetic algorithm, defining a cost function equal to the sum of the ball's height error, a population size of 50, and constraining the variables between -10 and 10.

PID controllers have applications beyond levitation devices with more practical industry applications. Adam Nyberg studied the use of PID's to control Robotic movement. Nyberg (2017) sought to investigate the potential of using an evolutionary programming (EP) method to solve optimal PID variables by examining their effects on the robot's angular and linear velocities. Nyberg (2017, p.3) defined a route for the robot and the system's average error to evolve and solve the K parameters. Nyberg designed his study as a parallel study to two other studies. In those studies, authors optimized their controllers using similar soft computing strategies, one applying a genetic algorithm (GA) and the other using Particle Swarm Optimization (PSO). Nyberg (2017) used inspiration from a third study in which the different optimization optimized the voltage draw of a theoretical DC motor. Nagaraj et al., from Kamaraj College of Engineering and Technology in India, conducted the original study. Like the studies revolving around the levitation devices, Nagaraj et al.'s (2008, p. 279) study used a theoretical mathematical equation to calculate the DC motor's transfer function. Their goal was to improve the motor's steady-state characteristics and performance indices by optimizing the motor output using the motor's transfer function. Nagaraj et al. (2008, p. 279) applied the three soft computing strategies EP, GA, and PSO, along with the two most traditional strategies Ziegler-Nichlos and Continuous Cycle, to generate five distinct PID loop cycles. Using an integral motor's PID, a defined setpoint served as a standard to compare the different strategies (Nagaraj, 2008, p. 279). The soft computing method proved to have a beneficial influence on the quality of the motor's PID loop by decreasing the rise time by up to 94 percent, the peak overshoot by up to 89%, and the settling time by up to 91 percent in contrast to the traditional methods (Nagaraj, 2008, p. 280).

2.2.2 Teaching Models Involving Machine Learning

As presented, ML has been used to optimize a PID controller. However, the study will instead use ML to develop a function to calculate an appropriate VFD frequency setting based on the PID loop output. ML has the potential for an unlimited number of applications through an array of programming methods. Some of the most common ways include Regression, Classification, Clustering, Reinforced Learning, and Neural Networks. Regression and Classification models, both Supervised Learning algorithms, are built using inputs data that corresponds known output values.

Bram van der Vlist et al., from the Eindhoven University of Technology, produced a case study of a teaching curriculum designed to teach students the programming and mathematic principles behind some ML methods. The curriculum revolved around two projects, each that used an NXT programmable LEGO set to provide a tangible platform to experience an applied ML program (van der Vlist, 2008, p. 213). They also designed their strategy to help communicate the ML complexities to students who are not necessarily Mathematics and Computer Science majors; therefore, they are not necessarily comfortable with programming (van der Vlist, 2008, p. 213). The goal of their projects was much like the intentions behind this paper's proposed model.

Van der Vlist's study had a shared value for a creating a tangible learning experience. Each study also hopes to accommodate students from various disciplines with a range of technology, programming, and mathematic skills. One of van der Vlist's projects consisted of creating a vehicle controlled by a Reinforced Learning algorithm (Q-Learning). In the case study, one group of students created a vehicle that used a mechanical arm with two joints that could be automatically adjusted to propel itself. It used a distance sensor to gauge its movement. The Reinforced Learning algorithm provided a positive reward if the vehicle moved forward and a negative penalty if it moved backward. With the initial program, the crawler makes seemingly random angle adjustments and forward/backward movements. However, with time, the crawler naturally refines its program as it develops an understanding of how to open and close its joints to move the vehicle forward (Van der Vlist, 2008, p. 210-211). A second project introduced students to Neural Networks. Students utilized the NXT's microphone sensor to sense a spoken word and display it on the NXT's screen. For their case study, the terms "Biertje" and "Champagne" were program into the system and linked to samples from the sensor. The NXT would distinguish between the two words with a neural network generated based on the sensor's amplitude signal (Van der Vlist, 2008, p. 213).

2.2.3 Machine Learning Regression Models

Regression models identify relationships between a set of inputs and outputs. The models produce mathematical functions helpful for making predictions based on given information. MATLAB's Regression Learner app can generate the optimal function to fit a collection of input values to scalar values. MATLAB uses multiple algorithms and evaluate the model's accuracy using withheld test data to measure RMSE of the functions predictions versus the systems actual output.

Liu et al. (2020, p.5) explored multivariable linear regression-based wavelet neural networks. Multivariable linear regression models seek to relate a dependent variable to two or more independent variables. Equation 1 provides a general formula for any number of *k* variables at any *i* interval (Brown). The equation weighs each independent variable *x* by multiply the variable with distinct coefficients *β*. The dependent variable *y* equates to the sum of all weighted coefficients within a standard deviation *e*.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + e_i, \quad i = 1,2,...,n. \tag{1}$$

Simple linear regression models fit the data to a straight line following the same multivariable equation, only incorporating a single *k* variable. Conversely, multivariable linear regression models fit data to a plane for a complex structure of more than two variables (Brown).

Liu et al.'s (2020) research revolved around a theoretical math-based simulated PID controller. The work utilized ML to predict the control system's output, enabling the calculation of the controller's optimal parameters (Liu, 2020). Liu et al. used the K values, inputs, and outputs of three preceding time intervals to predict the following output. Given the predicted output, Liu et al. could solve for optimal K values at the future time increment to stabilize the system.

While traditionally used for classification purposes, MATLAB's Regression Learner also offers support vector machines to solve regression functions. For regression models, SVM's define a line of best fit using margins to fit the data. For a two-variable system, SVM margins equally space support vectors to each side of a best-fit line. The margin widths and support vectors are calculated to optimally constrain data clusters and omit outliers. Kernels, valuable for multiple Machine Learning types, transpose data onto a higher dimension, allowing further separation between points and, thus, higher precision (Dufrenois, 2009). Planes and other complex structures can also project margins around multivariable functions. Many SVM programs can eliminate outliers and rerun the algorithms to further decrease sensitivity to false data (Zhu, 2001).

Like SVM's, tree diagram models also encompass multiple algorithms and are valuable for either regression or classification problems. Trees split data into classes using binary yes/no questions. Algorithms identify splits that divide the data into two groups with the "minimal impurity of the child nodes" (Loh, 2011, p.14). A tree can split until each node contains a single point or until the node impurity no longer decreases below of predefined threshold (Loh, 2011, p.14). The tree's output Y variables represent numeric values for a regression model that allows a regression function to fit the tree's nodes (Loh, 2011, p.14).

Gaussian Process (GP) models develop regression functions by analyzing all functions that fit a data set. For a collection of data, the GP identifies all possible functions that fit known values and creates a normal distribution of each function's output across a range of inputs. The prediction function is solved using the mean and covariance of all hypothetical outputs (Williams

1996). Williams (2006) describes how hyperparameters are adjusted, allowing the Gaussian function to miss the dataset and account for noise due to the data collection process.

2.2.4 Machine Learning Classification Models

Multiple regression learning algorithms also apply to classification problems. Classification models use a collection of inputs to predict an output among a discrete number of classes, contrary to predictions on a continuous scale. MATLAB's Classification learner supports classification models built using algorithms of tree diagrams, discriminant, Naive Bayes, nearest neighbor, and SVM's. Classification models are evaluated using test data and comparing the rate of misclassification divided by accurate predictions.

Again, with classification problems, tree diagrams use a series of nodes to divide data based on data features. Contrary to regression trees, classification models conclude each tree with a leaf that represents a single class (Murthy, 1998, p.349). Thus, deriving a prediction requires only working through the series of nodes. Trees use a "greedy top-down fashion" construction, first identifying an optimal initial split (the root node) and incrementally using nodes to discriminate the data (Murthy, 1998, p.349). MATLAB also offers bagged trees that support collections of trees to create a single output function. Kwon and Carter (1990) explored using bagged trees for a single data set to reduce performance variance.

The Nearest Neighbor strategy "assumes observations which are close together will have the same classifications" (Cover, 1967, p.21). Thus, examining a k-number of surrounding observations allows for the prediction of unclassified values. Multiple studies, including Sun and Huang (2010), explore adaptive k-value algorithms to identify the optimal k-value that increases model accuracy.

For classification problems, support vector machines seek to identify a function that divides the data by the different classes. Unlike regression models, the accuracy of an SVM for classification models increases as the function's side margins increase. Soft margins allow for misclassified data, preventing the overfitting of a function. Because SMV's use vectors to divide classes, the strategy can only apply to binary problems or a set of multiple binary problems.

Discriminant models are used exclusively for classification problems. The strategy assumes data classes fit normal distributions around distinct mean values (Xiaozhou, 2000). Algorithms calculate the mean and variance for distribution to determine the probability of a particular class at any input set. Areas where the probability of two different classes are equal define boundaries around classes (MATLAB, 2018).

2.3 Review of Literature pertaining to the Methodology

Nagaraj et al.'s work provided a comprehensive analysis of a DC motor running on a PID controller. As a result, they were able to compare the motor's output Nagaraj (2008) and his team used four of the five most common performance indices to evaluate the diverse PID strategies (EP, GA, PSO, Ziegler-Nichlos, and Continuous Cycle). The performance indices included: Integral Square Error (ISE), Integral Absolute Error (IAE), Integral time Squared Error (ITSE), and Mean of the Squared Error (MSE) (Nagaraj, 2008). Another standard performance index used is the Integral Time Absolute Error (ITAE). The subject's error is measured in contrast to its target height across a time interval to calculate these indices. Figure 2.3 provides a plot of Nagaraj et a.'s different optimization strategies and their PID cycles.

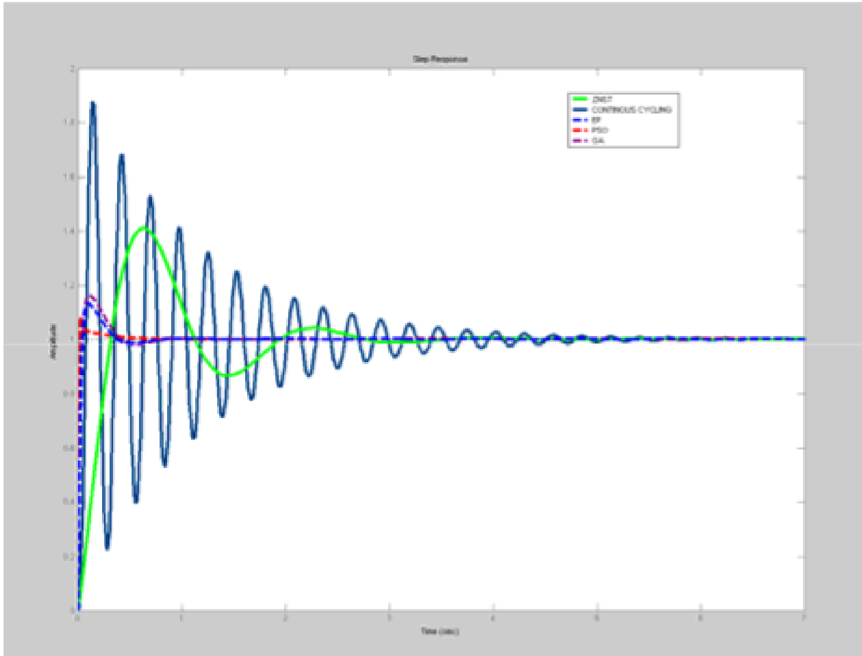


Figure 2.3 Motor output for each optimization strategy (Nagaraj, 2008, p. 280)

Table 2.3 provides the calculations of the performance indices for each PID cycle. Like Nagaraj et al., this study will use the MSE. The MSE will optimize the Physical system's PID which will serve as an index to compare the theoretical/virtual optimized models with the actual physical model.

Table 2.3 Comparisons of performance of optimization strategies (Nagaraj, pg. 280, 2008).

Performance Indices	**Z-N (step response)**	**Kappa-Tau**	**Continuous cycle**	**EP**	**GA**	**PSO**
ITAE	3.3805	3.3113	7.82	.0721	.3781	.0229
IAE	.5176	.5188	.56	.4891	.7712	.9161
ISE	2.3467	2.2503	3.2	1.0277	1.0435	1.0016
MSE	.0117	.0112	.016	.0051	.0052	.0050

For further analysis, Nagaraj (2008) also measured each cycle's steady-state characteristics, including their rise time, settling time, and peak overshoot. These steady-state characteristics will also serve as standards for comparing the theoretical/virtual models with the physical model. Table 2.3.2 displays Nagaraj's measurements.

Table 2.4 Comparisons of steady-state characteristics (Nagaraj, 2008, p. 280)

Characteristics	**ZNST**	**Continuous cycling**	**PSO**	**EP**	**GA**
Peak over shoot (%)	41.4	87.6	8.81	13	12.9
Settling time (sec)	2.56	4.31	.205	.324	1.15
Rise time (sec)	.242	.0474	.014	.0317	.0385

CHAPTER 3. RESEARCH METHODOLOGY

Chapter three contains five sections. The first section provides a detailed description of the proposed physical teaching model. The second describes creating a Machine Learning regression algorithm that captured the function of the physical system. The regression algorithm was necessary to optimize the systems' operation through a theoretical model. The third section covers the optimization strategy. The fourth section details the data collection process for building the classification model. Finally, the fifth section reviews the evaluation process that comparing the two models to ensure the virtual model is an accurate representation of the physical system and can provide solutions that translate to the physical system.

3.1 The Physical Platform

The regression model test used a 0.1-gram hovering disc was 3-D printed with PLA. And the classification used a second disc designed to reduce mass. Appendix A and B provide the dimensions for Disc One and Two, respectively. Figure 3.1 provides a picture of the two discs. Disc One, used for both the regression and classification testing, is on the right side of the figure. The disc levitates inside a clear six-foot-tall tube. The outside diameter of the tube is 3 inches, and the inside diameter is 2.75 inches. A 3-phase 240-volt blower fastened to the bottom of the tube propels the disc upward. An Allen Bradley PowerFlex-523 Variable Frequency Drive (VFD) controls the blower's speed by adjusting the motor's frequency with a precision of .01 Hz. The VFD works compatibly with an Allen Bradly Compact Logic 5380 PLC. The Compact Logix 5380 is programmed using Studio 5000 and serves as a PID controller for the project, calculating and controlling the VFD's frequency. The physical system also relies on a laser sensor to measure the disc's height. The sensor is an Opti2000 and measures the entire length of the tube, capable of a range between fifty millimeters and 3050 mm. A 3-D printed fixture holds the sensor at the top of the tube. The laser has a maximum sampling rate of 50 Hz, meaning the sensor can collect a reading once every 20 ms. Figure 3.2 displays a block diagram of the physical components in the system. Appendix C provides a wiring diagram for the system.

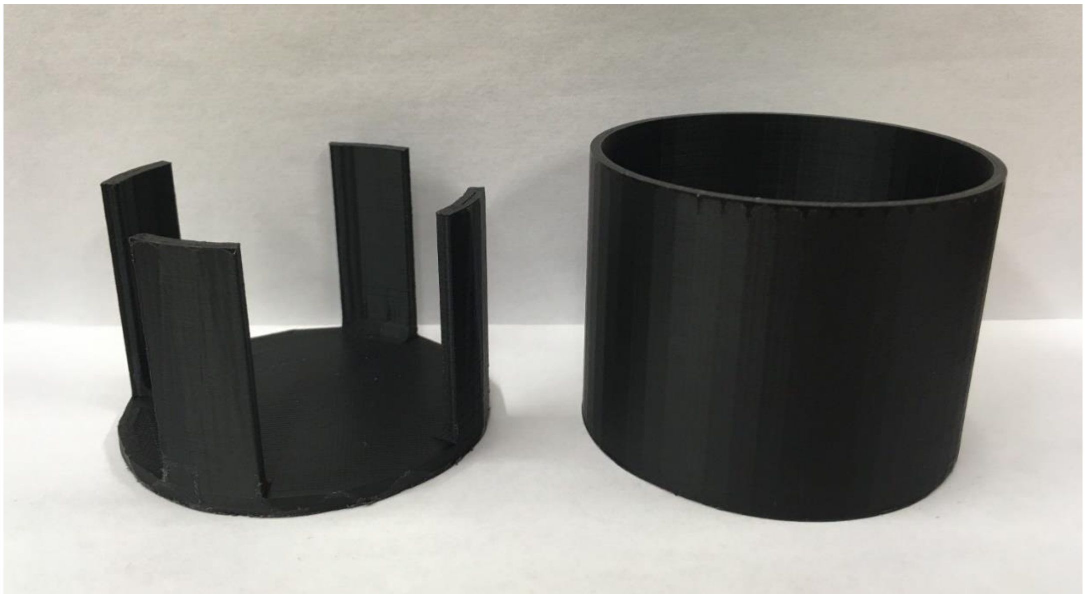


Figure 3.1 Discs used for classification and regression model testing

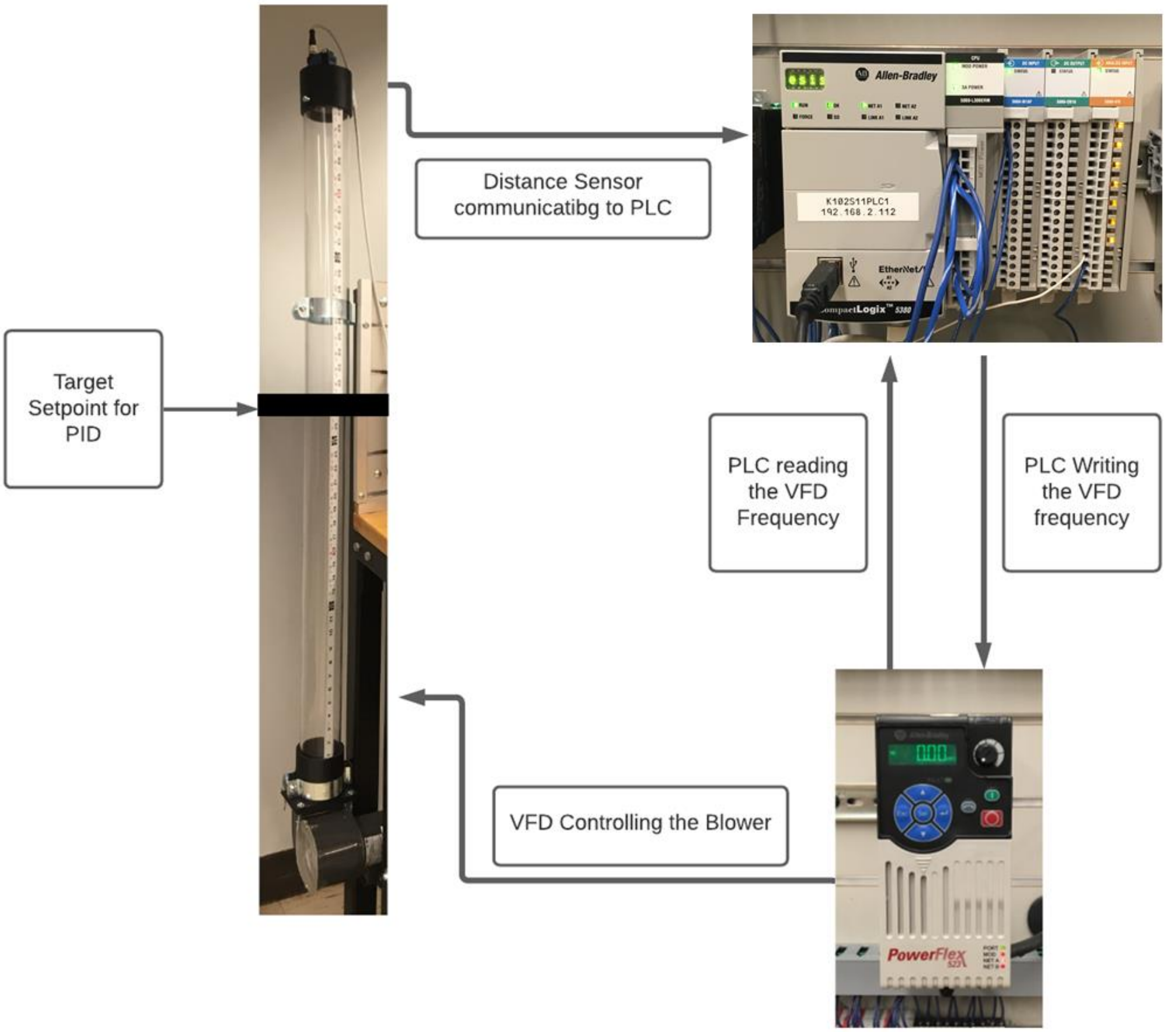


Figure 3.2 A block diagram of the physical system

3.2 Machine Learning Regression Model

A regression model was essential for the program to simulate the disc's movement allowing for PID optimization and controlling the program's disc movement. MATLAB's Regression Learner generated an algorithm capable of solving quantitative solutions based on inputs. The ML algorithm processed three input variables (initial disc height, initial disc speed, and VFD frequency) and calculated two output variables (resulting disc height and resulting disc speed). Predicting the disc's resulting height and velocity is critical to simulating the disc's flight path over time.

The ML model required a dataset of sample data to generate a relationship between the input and output variables. The dataset will come from samples of the disc's movement while running the physical system programmed with a PID controller and predefined PID values. A trace in Studio 5000 collected the output and input samples at .06 second intervals. The samples captured the disc's movement using an array of programmed PID values, each set defined for one minute. Each sample collection for the various PID values began with the disc at the height of zero. Using multiple sets of predefined PID values helped supply a variety of data for the basis of the ML model. Table 3.2 displays the PID values for the sample collection process. Additionally, the sampled of the disc moving at fixed frequencies. Fixed VFD frequencies for data sampling of the disc's movement included: 0, 10, 20, 25, 30, 32.5, 34,35,36,39, 40, 42, 46, 48, 50, 52, 54, 56, and 60 hz. Tests showed that frequencies at 36 hz and below caused the disc to fall. So, for fixed frequencies at and below 36 hz, a program set the motor frequency at 45 hz, holding the disc at the top of the tube until the frequency shifted to the predefined frequency. Frequencies at and above 39 caused the disc to rise, so sampling began while the disc was at the bottom of the tube and the VFD was set to zero hertz.

Table 3.1 PID values for data collection

Sample Collection	**P**	**I**	**D**
1	10	10	10
2	10	10	25
3	10	25	25
4	10	25	10
5	25	25	25
6	25	25	0
7	25	25	10
8	25	10	25
9	15	10	25
10	5	10	25
11	5	10	10
12	5	5	10

3.3 PID Optimization

MATLAB's automatic optimization function (*fminsearch*) optimized the disc's flight path for K_P, K_I, and K_D. The optimization process aimed to reduce the disc's settling time, rise time, and overshoot, based on a target height of 35 inches. For the study, hoovering between 34.25 inches and 35.75 inches for at least 3 seconds defined stabilization at 35 inches. The optimization program started the simulated disc at zero inches and constrained the disc's flight path to three minutes. The disc's flight path MSE versus the set point served as the cost function for MATLAB's *fminsearch* tool. A low and ideal MSE reflects a fast rise time, a slight overshoot, and a quick settling time. Using diverse K values, the PID controller automatically generated an array of MSE solutions, allowing a quantifiable comparison of the tuning parameters. Equation 2 provides a general formula for the controller.

$$U = K_P e(t) + K_I \int e(t) dt + K_P \frac{d}{dt} e(t) \tag{2}$$

The formula solves an appropriate response frequency to enable the disc to reach the target height. The PID controller will naturally constrain the ML response frequency to the limits of the motor and drive to ensure safe operating conditions. A continuous looping of the PID stabilizes the disc at the target of 35 inches. The ML predictors satisfy the formula's request for a continual update of the disc's height and speed, required in a continuous feedback loop.

3.4 Machine Learning Classification Model

Classification models have gain attention for predictive maintenance and part identification applications. The detailed lab platform can introduce students to classification problems by covering the transparent tube and challenging the students to identify a disc based on how the disc moves. Classifying between two discs corresponds to predictive maintenance industry challenges. For a concealed system that utilizes multiple components, classification models can help identify a particular part that has failed based on the system's operation or a consequential noise. Models can also identify types of problems. For example, relating to the levitation of a disc, rust on a metal disc may cause added friction that slows the disc down. Likewise, particle accumulation on the disc may add weight and destabilize the disc, affecting the movement of the disc. Certain problems, like rust, may be critical in contrast to other system failures; and, therefore, vital to distinguish. Students can experience the same classification challenges by creating a model capable of deciphering between two discs. In the detailed model, the lighter disc may represent a normal state. The heavier disc may correspond to an abnormal state like a disc with corrosion build-up, as the heavier disc has more surface area on the wall, thus is susceptible to more fiction and requires more lift.

Samples for Disc Two followed identical procedures for Disc One. MATLAB's classification learner applied multiple algorithms to the data to identify the ideal algorithm. Unlike the regression function, the classification functions are put directly to a test using testing data. Like for the regression model, testing data comes from sample data withheld during the building process.

3.5 Model Analysis

In theory, the two models, the physical and ML, create identical disc flight paths. However, because the ML model is a mathematical representation of the physical model and based on limited test samples, the ML model may not accurately represent the physical system. The study used comparative analysis to compare the disc flight paths on the physical and virtual systems using the same PID values to test if the ML is viable for deriving solutions. The physical system ran the optimal PID values to generate a plot of the disc movement from a starting height of 0 inches to stabilization at 35 inches. A Studio 5000 recorded the disc's height every 60 ms using the laser sensor. The theoretical model in MATLAB operates using the same starting height and set point. Comparing the two plots involves evaluating differences between peak-overshoot, rise time, settling time, and the MSE.

3.6 Lab Experience

The detailed platform was developed for integration into an entry-level industrial controls course. The laboratory can accommodate students from diverse majors and is completable with no prior experience outside of the course. The lab experience contains two parts: one to explore using a regression model to optimize the systems PID controller. The second is to explore using a classification model to identify a disc based on flight movement. One potential avenue for integrating the platform into curriculum involves students following the methodologies laid out by the paper to create ML models, optimize a PID controller, and identify discs based on disc movement. The study did not focus on the platform's efficacy to deliver material to students. Instead, the study was designed to explore the platform and test the effectiveness of using Machine Learning to solve associated problems.

CHAPTER 4. RESULTS

The Machine Learning models explored for the work provided a simplified and simulated means to solve problems that exist on a physical counterpart. Gauging the accuracy of the machine learning model against a physical system ensures the theoretical model is viable for deriving accurate solutions. Identifying relevant and redundant variables was key to establishing precise ML models. Comparing an array of multiple ML models, each built upon different features, provided clarity on the importance of each variable.

The blower's frequency is a critical variable as it controls the amount of air flowing through the tube and is responsible for the upward push on the disc. The disc's height is significant because of potential defects, like narrow spots along the length of the tube. The disc's height is also crucial for predicting the disc's future position. The disc's velocity and acceleration provide an idea of the disc's future height. Additionally, prior samples of each relevant variables are valuable for understanding the disc's resulting height. Table 4.1 shows that previous samples of the blower's frequency influence the lift of the disc due to a delayed windup of the blower. The table displays the disc's movement data at a fixed blower frequency of 60 hz. The data shows that from the initial time the VFD first read 60 hz, the blower took 480 milliseconds to start lifting the disc.

Table 4.1 Delay of blower impact

Time (s)	**VFD Frequency (hertz)**	**Height (in)**
0	0	-.2323
.06	60	-.1997
.12	60	-.1865
.18	60	-.2075
.24	60	-.1528
.3	60	-.1528
.36	60	-.1977
.42	60	-.2610
.48	60	-.2219
.54	60	-.1705
.6	60	.4366
.66	60	.4366
.72	60	2.2704
.78	60	4.8588
.84	60	8.2688

Similarly, past samples of the disc's height are also valuable. The iterative height measurements allow for velocity, acceleration, and jerk measurements. The 60 ms data collection rate corresponded MATLAB's height predicting increment. Thus, the sampling period became obsolete, allowing MATLAB's Regression Learner to determine the relationship between the two height measurements. Therefore, the required velocity calculations and prediction for the proposed ML model became redundant. Table 4.2 is an example set of data for the proposed model, capable of predicting height and velocity using previous velocity measurements. Table 4.3 is an example set of data for a height predicting model which excludes velocity data. For comparison purposes, the paper analyzes ML models, including and excluding velocity measures.

Table 4.2 Data collection example w/ velocity measurements

VFD_2	Height_2	VFD_1	Height_1	Velocity_1	VFD	Resulting_Height	Resulting_Velocity
4407	-.3073	4411	-.2195	1.4638	4467	.6300	14.1575
4411	-.2195	4467	.6300	14.1575	4227	1.1759	9.0987
4467	.6300	4227	1.1759	9.0987	4304	1.5759	6.6666
4227	1.1759	4304	1.5759	6.6666	4407	2.3464	12.8415

Table 4.3 Data collection example w/o velocity measurements

VFD_2	Height_2	VFD_1	Height_1	VFD	Resulting_Height
4407	-.3073	4411	-.2195	4467	.6300
4411	-.2195	4467	.6300	4227	1.1759
4467	.6300	4227	1.1759	4304	1.5759
4227	1.1759	4304	1.5759	4407	2.3464

4.1 Regression Model Analysis

Gaussian Process Regression and Stepwise Linear regression for Height Predictor consistently generated functions more accurately than other algorithms. The linear models stood out for their fast generation. Figure 4.1 provides a visual representation of the predicted height responses versus actual responses for the machine learning model that used 15 samples of the immediately prior data measurements. Table 4.4 includes accuracy measurements for each of the different models. All the models built excluding the velocity calculations generated models with a RMSE between .28 and .41. The models built using the velocity data required two models, one to predict height and one to predict velocity. Height predictors using velocity data had a RMSE of approximately 0, because the data existed to calculate the next height. However, the velocity predictors had a RMSE of 7, diminishing the overall accuracy of the height predictor which relied on the velocity predictor for future predictions.

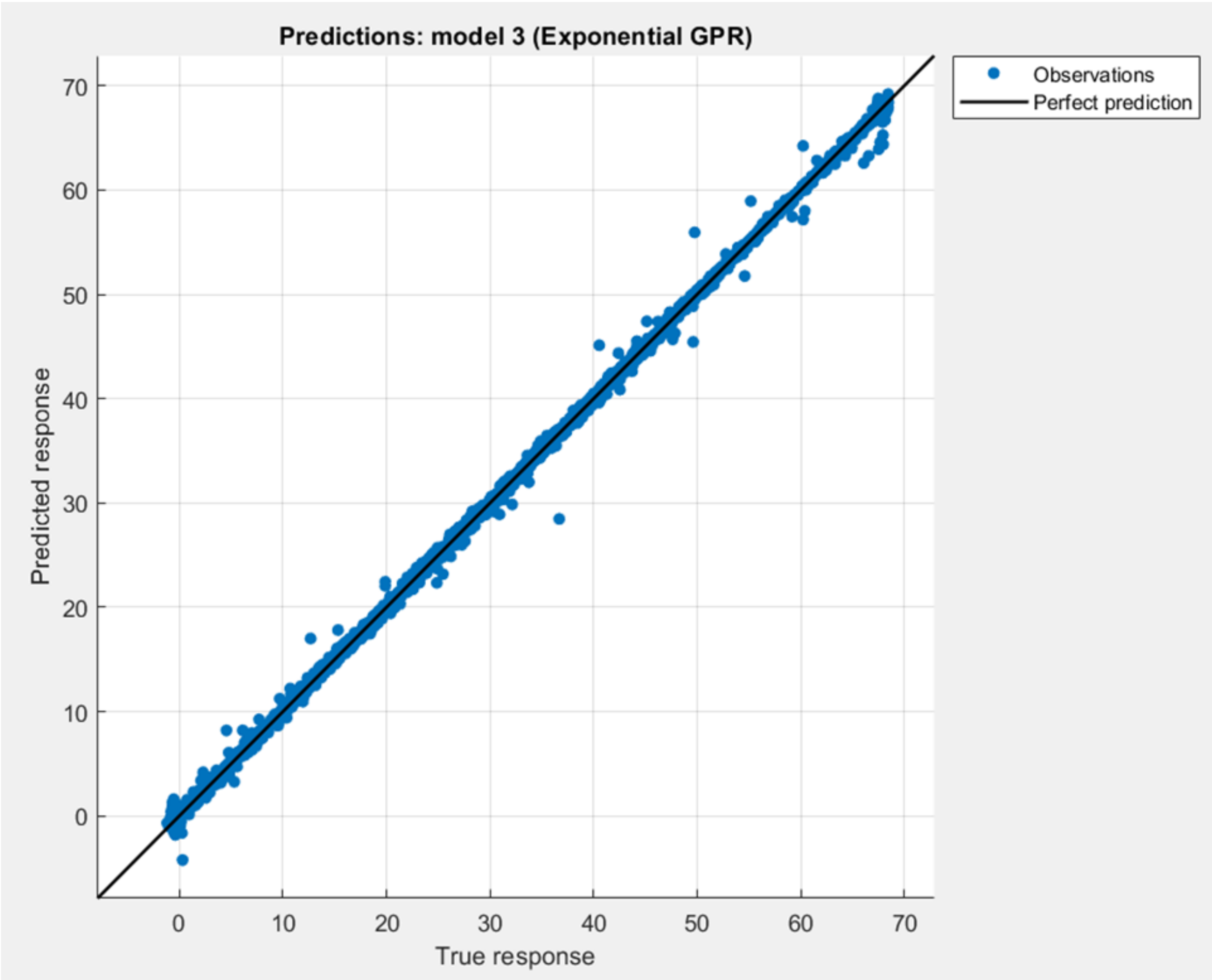


Figure 4.1 Height predictor model accuracy

Table 4.4 Comparison of ML models to physical model

Model	Accuracy (RMSE)	MSE	Rise_time (s)	Percent_Overshoot (%)	Settling_Time (s)
Physical System	-	74.49	1.32	14.97	31.68
ML_2it w/Velocity	~0/~7	78.4	1.86	19.34	32.76
ML_2it	.404	81.33	1.92	33.09	36.66
ML_3it	.302	78.45	1.92	30.28	32.16
ML_4it	.315	77.09	1.92	22.17	33.36
ML_5it	.312	78.56	1.92	23.9	30.9
ML_15it	.28	70.57	2.22	18.08	10.62

Note: #it signifies the number of immediately prior sample measurements of Height and VFD Frequency used to build the ML models

Appendix D displays an excerpt from the MATLAB-based PID controller. The code predicts the height to make PID calculations. To provide a simplified visualization of the relationship of the *k*-values, Figure 4.2 illustrates a contour plot when the derivative gain is set to zero. The chart details the *k*-values against the MSE calculations of the simulated disc flight path for a range of Ki and Kp values. Note the minimization of the MSE calculation as the K values simultaneously increase and the maximum MSE results when Ki is negligent. The pattern remains consistent for the 3D scatter plot detailed by Figure 4.3, which incorporates values for all three variables, simultaneously representing the proportional gain (X), integral gain (Y), and derivative gain (Z). Figure 4.4 also provides a scatter plot of the three variables, except having increased maximum constraints, showing the integral coefficient's diminishing effect.

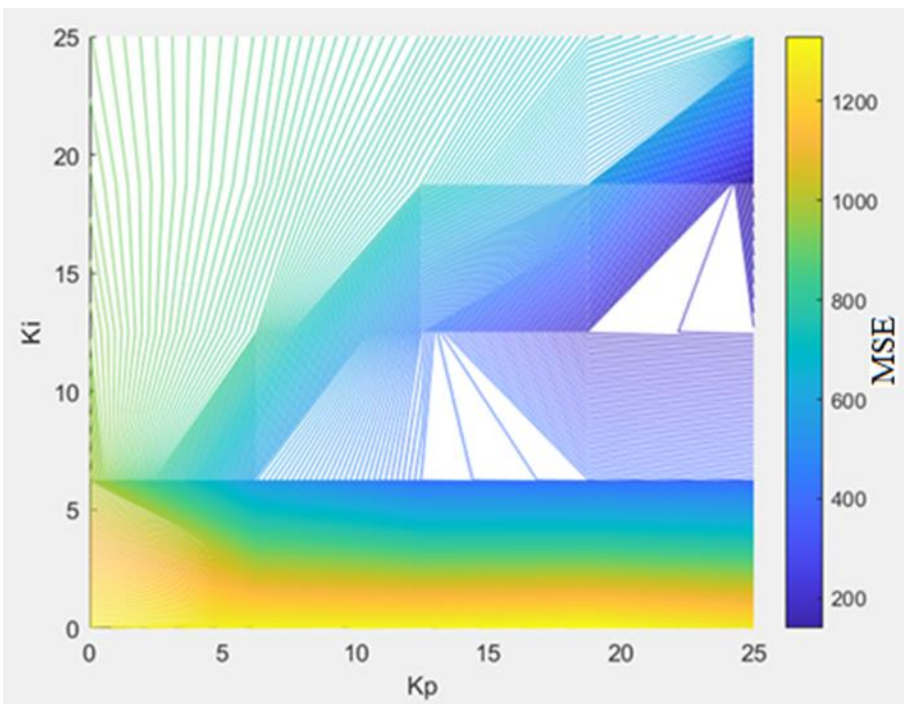


Figure 4.2 Contour of Kp vs. Ki

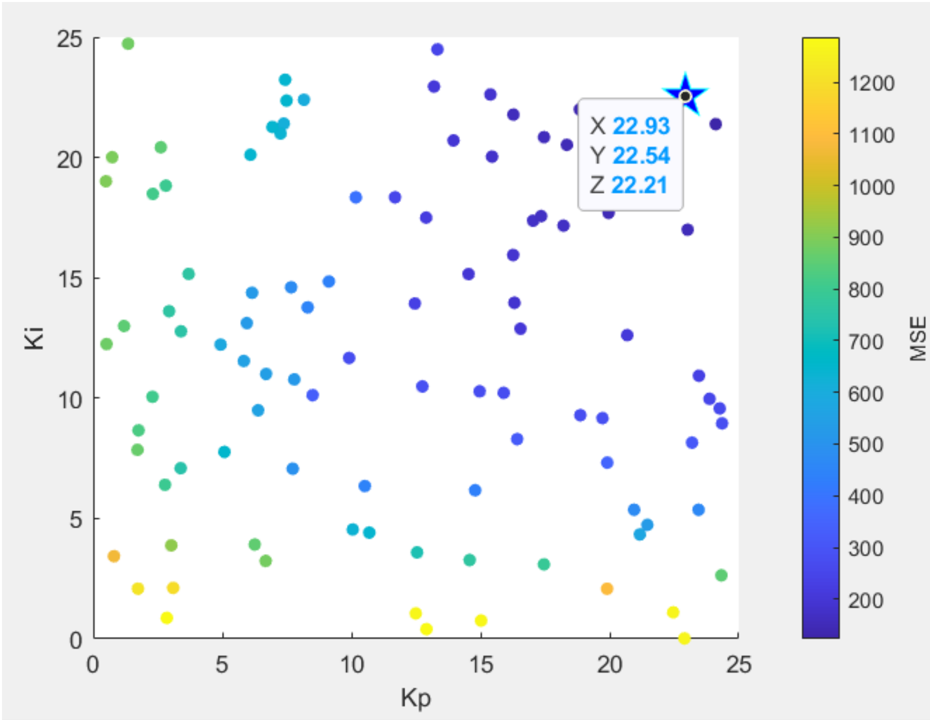


Figure 4.3 Optimized solution based on MSE

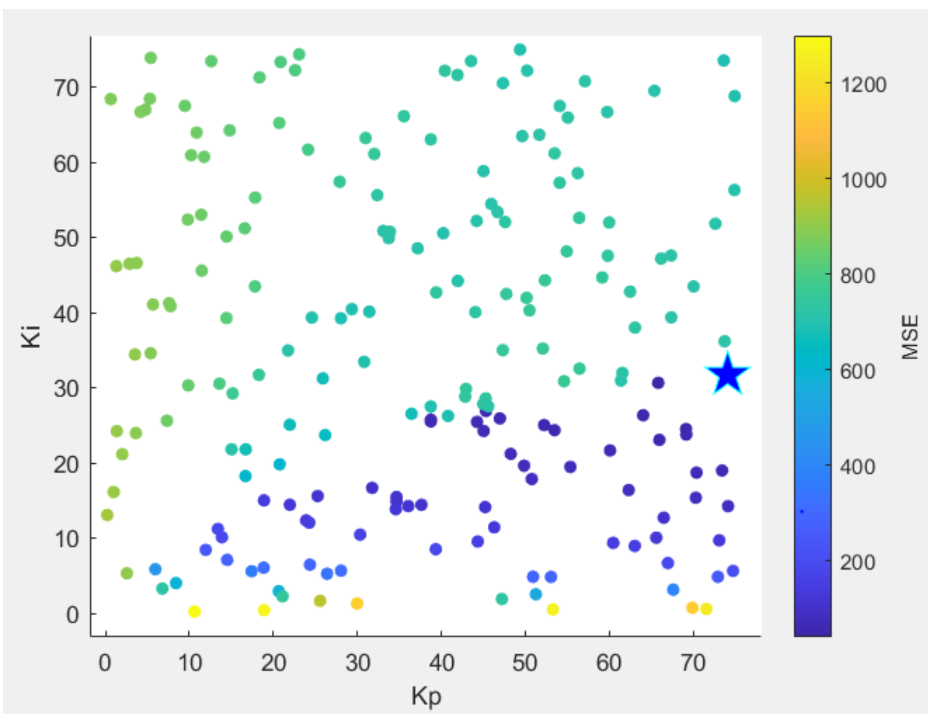


Figure 4.4 Optimized solution based on MSE (2)

The Monte Carlo sampling depicted in Figure 4.3 providing a starting point for the Nelder-Mead algorithm of the fminsearch. The Monte Carlo found a minimal solution of (22.93, 22.54, 22.21). After one hundred iterations of the Nelder-Mead algorithm, MATLAB found an optimal solution (26.707, 24.2246, 20.8477). Figure 4.5 depicts an actual and simulated flight path for a disc specific to the optimal solutions from the fminsearch. The figure illustrates a machine learning model's output using the complete data set including velocity data excerpted by Table 4.2. The simulated model shows the disc rising and falling corresponding to the physical system, each rising to a height less than two percent different and less than one second apart for the first two peaks. However, the physical system shows sensitivity to noise noticeable by the increasing oscillations after one hundred seconds. The noise captured by the physical system signals unaccounted features. Such as airflows blockage preventing consistent air pressure or the disc catching on the side of the tube.

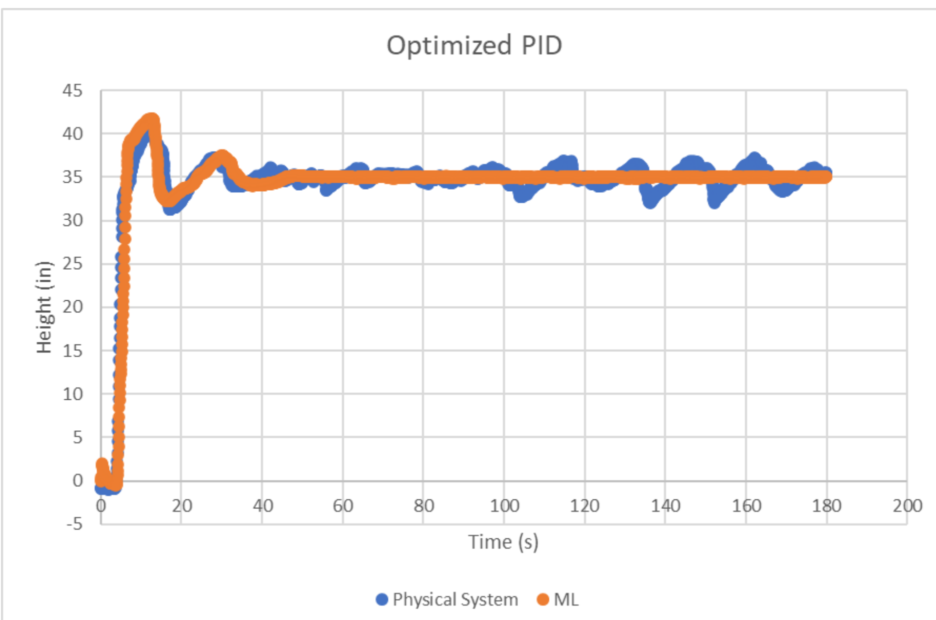


Figure 4.5 The PID comparisons

Figure 4.6 provides a compilation of the six ML models against the physical system to show the effect of using various data inputs. Table 4.4 (p. 32) displays the measurements for comparing the fight paths. The model based on fifteen samples of prior data generated the highest model accuracy for the exponential Gaussian process regression algorithm, consistent to Table 4.1's (p. 30) evidence that the system has a delayed effect. However, an overlay of the flight paths shows the model built using 15 prior samples of data has timing inaccuracies of the ML models peaks and valleys. The 15 prior sampled ML model created a smoother plot, lacking the rapid fluctuations that exist throughout the physical system's data. Again, the fluctuations are likely caused by the disc catching the side tube, air density inconsistency, and sensor noise. The performance characteristics listed by Table 4.4 also illustrate the inaccuracy of the model built using the fifteen prior samples. For comparison purposes, each performance characteristic proves uninformative when viewed individually but valuable when analyzed as a collective. Individual characteristics can create an illusion for a correlation between the physical model and the simulated, despite other elements of the flight paths showing discrepancies. The 15 sample based model had a rise time a full standard deviation (.24 seconds) beyond the other times. Likewise, the same model had a settling time one third the settling time of the physical system and other models.

Each model inaccurately predicts an immediate rise and fall of the disc. The fixed frequency data cause inaccurate steep spikes at the beginning of each flight path. The attenuation of the false spike's height as the number of prior samples used for a model increases is evidence of the fixed data creating the spikes. Each addition of a prior measurement to generate the ML models required one less sample of the resulting disc height. Subsequently, the ML models using more prior data samples have smaller datasets, showing the disc rising to a lower height for specific frequencies and falling from a lower elevation at certain frequencies. The ML that used velocity data showed a negligent false spike and overshoot closer to the physical system than the four other models.

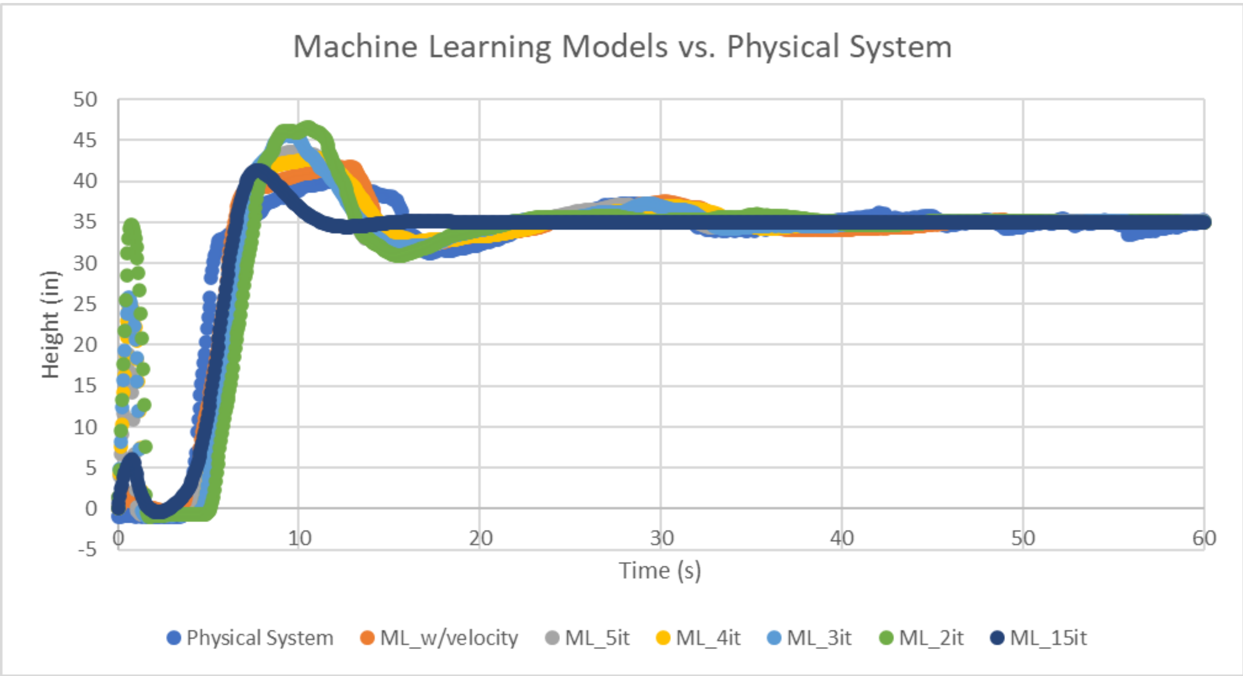


Figure 4.6 ML models vs. physical system

A compilation of the models programmed to fixed frequency, depicted by Figure 4.7, provides a simplified visual for model comparisons. At 48 hz, all models rise from zero inches to 60 inches no more than .06 seconds different from the physical system rise time. However, the models fail to match the delayed rise of the physical system's displayed response. The addition of prior measurement samples helps capture the delayed response. The ML model that uses velocity predictor, which performed closest to the physical system while using the PID controller, fails to recognize the constraints at the tube of the tube which limits the disc to a height of 68 inches.

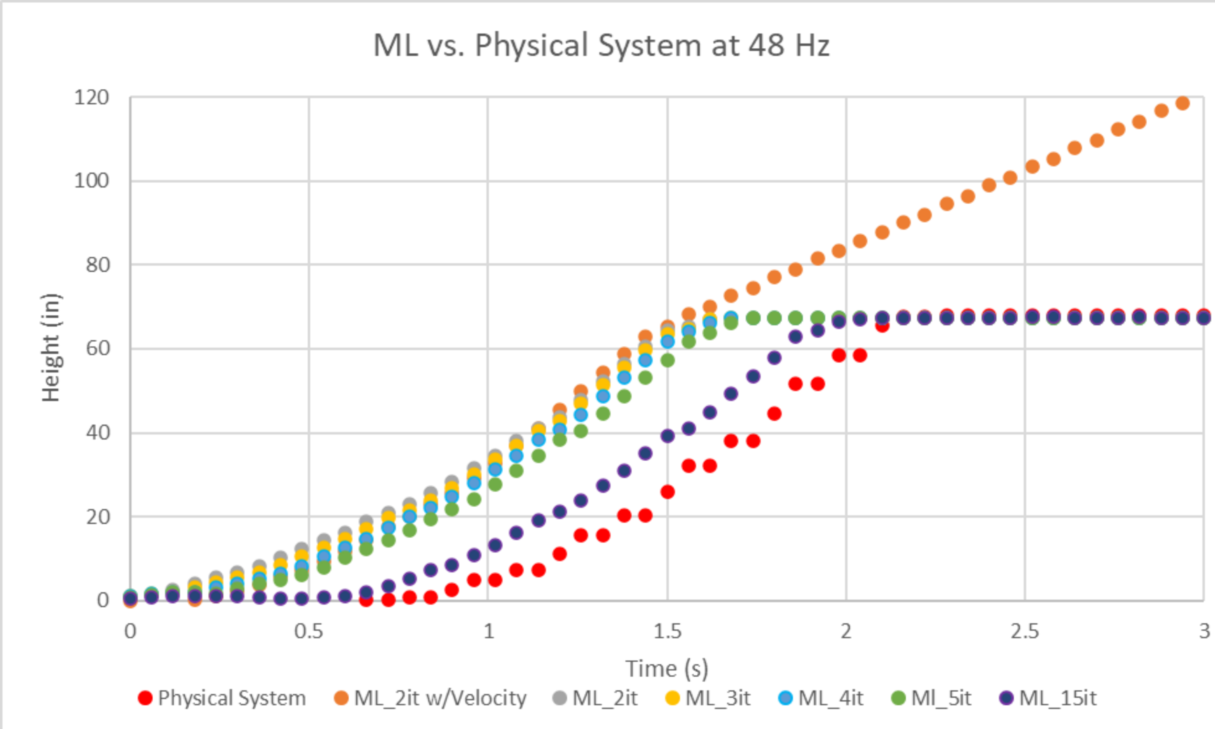


Figure 4.7 ML models vs. physical systems at 48 Hz

4.2 Classification Model Analysis

The value of classification models are direct, as they allow users to categorize an unknown object based on a set of features. The data height measurement data reflect the velocity and acceleration at various VFD frequencies. Since Disc Two weighs less than Disc 1, the two discs move at different speeds despite identical conditions, allowing the program to decipher the disc. The Classification Learner's K-Nearest Neighbor (KNN) consistently outperformed the other algorithms. Figure 4.8 displays a confusion matrix of test data results run through the algorithm contrasted to actual responses. Table 4.5 provides a comparison chart listing the accuracy of multiple models each built including and excluding velocity data and using a different number of prior samples. The models struggle to distinguish between the disc when the model is stable at the top and bottom of the tube, where the disc does not have velocity and is not accelerating. The randomly generated test samples challenge the comparison of the model accuracies. One test may capture values easily discernable by the model, while another evaluates samples where the separate disc's clusters converge. The RMSE calculations of the models differ 1.2 percent, representing 124 samples. However, the accuracy percentage differences may slightly change depending on the sample used for testing. A controlled experiment would use the same testing sample for each test.

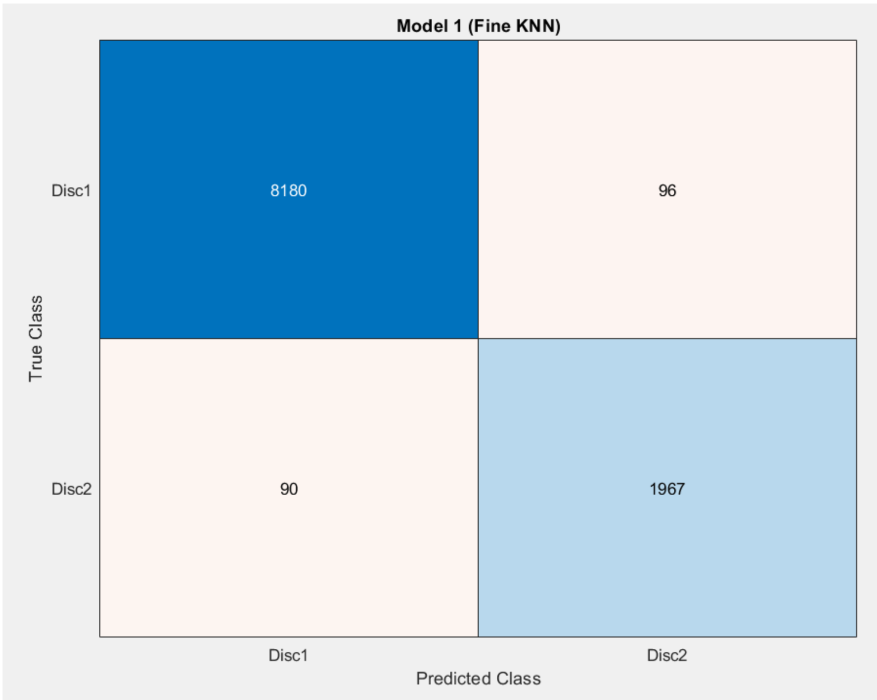


Figure 4.8 Classification confusion matrix

Table 4.5 Classification model comparisons

Model	Accuracy (%)	Accuracy w/velocity data (%)
Fine NKK 5it	98.2	98.3
Fine NKK 4it	98.2	98.4
Fine NKK 3it	98	97.9
Fine NKK 2it	97.6	97.2

Unfortunately, testing the classification model on a live program requires an external computer to execute a python file. MATLAB embeds the regression function, hiding the coefficients. A live program requires exporting the software's compact functions, which is impossible. Azure also provides a platform to build Machine Learning models and can export the model to a python code. The PLC can capture the disc data movement and communicate to a connected Raspberry Pi executing the ML python code. Time constraints prevented the connecting the Raspberry Pi and PLC.

CHAPTER 5. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The thesis explored a cyber-physical system as a potential tool to apply Machine Learning models to solve Industrial Control problems. Further steps center on crafting a curriculum that integrates the platform and tests the tools effectiveness of conveying experiences to students. The process proved viable for creating ML models useful for solving problems.

5.1 Conclusion

For both the regression and classification tasks, the ideal model balances model accuracy while limiting data processing. When examining the flight paths, the model built using four samples of prior data captures the tube's height limit, begins to capture the VFD's delayed effect, and matches the PID flight closers to the other models. The model is also the point of diminishing return for the classification task. Meaning adding further samples of prior data fails to increase the model's accuracy. The fact that the model is optimal for both tasks confirms that four samples of prior data are the most insightful information for describing the disc's movement.

The initial walkthrough of the experience provided a challenge that would push students to consider possible relevant variables and identify redundant ones, an important part of learning to build ML models. The experience also does well at illustrating PID controller principles. Contrast to using the ZN optimization strategy, which identifies a single solution, the Monte Carlo and Nelder-Mead methods show an array of *k*-value combinations exist. MATLAB plots provide a visual to understand how different *k*-value combinations compare and show the individual effect of each PID coefficient. MATLAB also provides granular control to obtain application specific flight paths. For instance, additional lines of codes could penalize an object for going over the target height, allowing students to find an optimal solution that prevents overshoot.

5.2 Recommendations

Condensing the material is critical to delivering the lab experience, as students will only have one or two lab periods to complete the task. Providing a data set of the disc movements would help reduce sampling time. Processing and filtering the data is tedious, so providing a data set would also save students time and prevent the students from making mistakes and affecting the model's accuracy. Students could add a small amount of data to the provided data set, allowing them to experience and understand the data processing and filtering, yet bypassing the full-time intensive task. A data set would also offer students a pattern to follow, guiding the incorporation of prior samples of data to ease the challenge. Students would only need to create their own PLC program to control the VFD and capture the height of the disc at 60 millisecond intervals.

Giving the students the PID optimization MATLAB code would further assist students to complete the lab. MATLAB based courses are not a prerequisite for the course; therefore, students may lack knowledge of navigating and programming the software. Various challenges can push the students to fully grasp the MATLAB program, such as adjusting the time frame, changing the target height, programming a dynamic target height, or preventing overshoot, all possible requirements for real-world applications.

The classification problem, which requires an external computer for live predictions, is ideal for manufacturing integration courses. Manufacturing Integration courses introduce students to device communication protocols and PYTHON programming, preparing students for the challenge.

LIST OF REFERENCES

Brown, S. (2009). Multiple linear regression analysis: A matrix approach with MatLab. *Alabama Journal of Mathematics.* Retrieved on 3/11/2021, from http://ajmonline.org/2009/brown.pdf

Chacon, J., Saenz, J., Torre. L., Diaz, J. & Esquembre, F. (2017). Design of a low-cost air levitation system for teaching control engineering. *Sensors*, *17*(10). Retrieved on 10/21/2020, from https://doi.org/10.3390/s17102321

Chołodowicz, E., & Orlowski, P. (2017). Low-cost air levitation laboratory stand using MATLAB/simulink and arduino. *Pomiary Automatyka Robotyka*, 21, 33-39. Retrieved on 10/21/2020, from DOI: 10.14313/PAR_226/33

Cohen, G. H., Coon, G. A. (1953, July). Theoretical consideration of retarded control. *ASME, 75,* 827-834. Retrieved on 3/14/2021, from https://folk.ntnu.no/skoge/puublications_others/Cohen%20and%20Coon%20(1953)%20-%20Theoretical%20Consideration%20of%20Retarded%20Control.pdf

Coleman, S., Smith, C. (2019). Evaluating the benefits of virtual learning for bioscience students. *Higher Education Pedagogies*, *4*(1), 287-299. Retrieved on 1/28/2021, from https://doi.org/10.1080/23752696.2019.1599689.

Cover, T. M. (1967, January). Nearest neighbor pattern classification. *IEEE Transactions and Information Theory 13*(1), 21-27. Retrieved on 6/5/2021, from DOI: 10.1109/TIT.1967.1053964

Dietz, M., Meissner B., & Schmidt-Vollus, R. (2020). Teaching digitalization and systems modeling for virtual commissioning using virtual labs. *IEEE 29th International Symposium on Industrial Electronics*, 440-445. Retrieved on 10/31/2020, from DOI: 10.1109/ISIE45063.2020.9152258

Dufrenios, F., Colliez, J., Hamad, D. (2009, November). Bounded influence support vector regression for robust single-model estimation. *IEEE Transactions on Neural Networks and Learning Systems, 20*(11), *1689-1706.* Retrieved on 3/12/2021, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5256161

Freeman, S., Eddy, S., McDonough, M., Smith M., Okoroafor, N., Jordt, H. & Wenderoth, M. (2014). Active learning increases student performance in science, engineering, and mathematics. *National Academy of Sciences, 111*(23), 8410-8415. Retrieved on 10/17/2020, from https://doi.org/10.1073/pnas.1319030111

Hussin, A. A. (2018). Education 4.0 made simple: Ideas for teaching. *International Journal of Education and Literacy Studies*, *6*(3), 92-98. Retrieved on 11/5/2020, from https://files.eric.ed.gov/fulltext/EJ1190812.pdf

Kotsiants, S. B. (2007). Supervise machine learning: A review of classification techniques. In J.B. Editor, R.D. Editor. N.G Editor, J.K. Editor, J.L. Editor, R.L. Editor, R.M. Editor, M.M. Editor & M.Z. Editor (Eds.), *Emerging Artificial Intelligence Applications in Computer Science* (pp.3-24). IOS Press. Retrieved on 6/5/2021, from https://books.google.com/books?hl=en&lr=&id=vLiTXDHr_sYC&oi=fnd&pg=PA3&dq=machine+learning+classification&ots=CZmtBB4Gmm&sig=STu2XANojhGmHwlKk__SjAS4SHQ#v=onepage&q&f=false

Kuzhandairaj, J. (2018) Development,control and testing of an air levitation system for educational purpose. *Politechico Milan*. Retrieved on 10/21/2020, from http://hdl.handle.net/10589/140083

Kwok, S.H., Carter, C. 1990. Multiple decision trees. *Machine Intelligence and Pattern Recognition 9*, 327-335. Retrieved on 6/5/2021, from https://doi.org/10.1016/B978-0-444-88650-7.50030-5

Lee, J., Yoon, J., & Kim, B. (2017). A big data analytics platform for smart factories in small and medium-sized manufacturing enterprises: An empirical case study of a die casting factory. *International Journal of Precision Engineering and Manufacturing*, *18*(10), 1353-1361. Retrieved on 30/7/2020, from https://link.springer.com/content/pdf/10.1007/s12541-017-0161-x.pdf

Leurent, H., & Boer, E. (2018). The next economic growth engine: scaling fourth industrial revolution technologies in production. *World Economic Forum*, 1-33. Retrieved on 30/7/2020, from http://www3.weforum.org/docs/WEF_Technology_and_Innovation_The_Next_Economic_Growth_Engine.pdf

Liu, J., Zhao, M., Li, G., Chen, J. (2020, September). Multiple linear regression prediction and wavelet neural network based intelligent online tuning control method. *IEEE*, 1-13. Retrieved on 3/10/2021, from DOI: 10.1109/ACCESS.2020.3020980

Loh, W. (2011, January). Classification and regression tress. *WIREs Data Mining and Knowledge Discovery, 11(2)*, 14-23. Retrieved on 3/13/2021, from https://doi.org/10.1002/widm.8

MatLab. (2018, October). Machine learning: What is discriminant analysis? YouTube. Retrieved on 6/6/2021, from https://www.youtube.com/watch?v=eBm8Uo9yhwI

McCauley, D. (2018, June 10). The automation readyness Index: Who is ready for the coming wave of automation? *The Economist Intelligence Unit.* Retrieved on 10/10/20, from https://www.automationreadiness.eiu.com/static/download/PDF.pdf

Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery 2*, 345–389. Retrieved on 6/5/2021, from https://doi.org/10.1023/A:1009744630224

Nagaraj. B., Subha. S., & Rampriya, B. (2008). Tuning algorithms for PID controller using soft computing techniques. *International Journal of Computer Science and Network Security*, 6(4), 278-281. Retrieved on 3/13/2021, from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.620.8198&rep=rep1&type=pdf

Nevaranta, N., Gräsbeck, K., Haapala, J., Lindh, T., Peltoniemi, P., Niemelä, H., Pyrhönen, O. (2018, August, 26-30). *Virtual learning for control engineering* education [Paper Presentation]. 2018 IEEE 18th Power Electronics and Motion Control Conference, Budapest, Hungary. Retrieved on 11/2/2020, from doi: 10.1109/EPEPEMC.2018.8521905

Noronha, A., Moriarty, R., O'Connell, K., & Villa, N. (2014). Attaining IoT value: How to move from connecting things to capturing insights. *Cisco*, 1-20. Retrieved on 30/7/2020, from https://theinternetofthings.report/Resources/Whitepapers/a8944733-f312-4c3b-968a-1e30db9291da_iot-data-analytics-white-paper.PDF

Nyberg, A. (2017). Optimizing PID parameters with machine learning. *Cornell University.* Retrieved on 10/27/20, from https://arxiv.org/pdf/1709.09227.pdf

Paz-Ramos, M. A., Torres-Jimenez, A., Quintero-Marmol-Marquez, E., & Estrada-Esquive, H. (2004). PID controller tuning for stable and unstable processes applying GA. *Springer,* 1-10. Retrieved on 3/14/2021, from https://link.springer.com/content/pdf/10.1007%2F978-3-540-24855-2_1.pdf

Schwab, K. (2016). The industrial revolution: what it means, how to respond. *The World Economic Forum.* Retrieved on 10/10/20, from https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond

Solihin, M. I., Tack, L. F., & Kean, M. L. (2011, January, 14-15). *Tuning of PID controller using particle swarm optimization (PSO)* [Paper Presentation]. International Journal on Advanced Science, Engineering and Information Technology. Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011, Selangor, Malaysia. Retrieved on 3/14/2021, from https://core.ac.uk/reader/296919280An

Sun, S., Huang, R. (2010, August, 10-12). *An adaptive k-nearest neighbor algorithm* [Paper Presentation]. 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, China. Retrieved on 6/5/2021, from DOI: 10.1109/FSKD.2010.5569740

Tootchi, A., Amirkhani, S., Chaibakhsh, A. (2019, November, 20-21*). Modeling and control of an air levitation ball and pipe laboratory setup* [Paper Presentation]. 2019 7th International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran. Retrieved on 3/14/2021, from DOI: 10.1109/ICRoM48714.2019.9071827

Truong, N. Lee, K.W. & Nguyen, T (2011, October, 26-29). *Development of virtual control engineering experiments* [Paper Presentation]. *20*11 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Korea. Retrieved on 1/28/2021, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6106278

Ustundag, A., Cevikcan, E. (2018). Industry 4.0: Managing the digital revolution. *Springer*. Retrieved on 9/10/2020, from https://doi.org/10.1007/978-3-319-57870-5.

Van der Vlist, B., Van de Westelaken, R., Bartneck, C., Hu, J., Ahn, R., Barakova, E., Delbressine, F., & Feijs, L. (2008). Teaching machine learning to design students. *Technologies for E-Learning and Digital Entertainment* (pp. 206-217). Springer. Retrieved on 27/10/2020, from https://www.drhu.eu/publications/2008-LNCS-TeachingMachineLearningToDesignStudents.pdf

Wellener, P., & Shepley, S., & Dollar, B., Laaper, S., Manolian, H., Beckoff, D. (2019). 2019 Deloitte and MAPI smart factory study: Capturing value through the digital journey. *Deloitte Insights*, 1-28. Retrieved on 30/7/2020, from https://www2.deloitte.com/content/dam/insights/us/articles/6276_2019-Deloitte-and-MAPI-Smart-Factory-Study/DI_2019-Deloitte-and-MAPI-Smart-Factory-Study.pdf

Williams, C. K. I. and Rasmussen, C. E. (1996). Gaussian processes for regressio*n.* IN: D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems.* MIT. Retrieved on 3/14/2021, from https://publications.aston.ac.uk/id/eprint/651/1/getPDF.pdf

Williams, C. K. I., Rasmussen, C. E. (2006). Gaussian process for machine learning. *Massachusetts Institute of Technology*. Retrieved on 3/13/2021, from http://www.gaussianprocess.org/gpml/chapters/RW.pdf

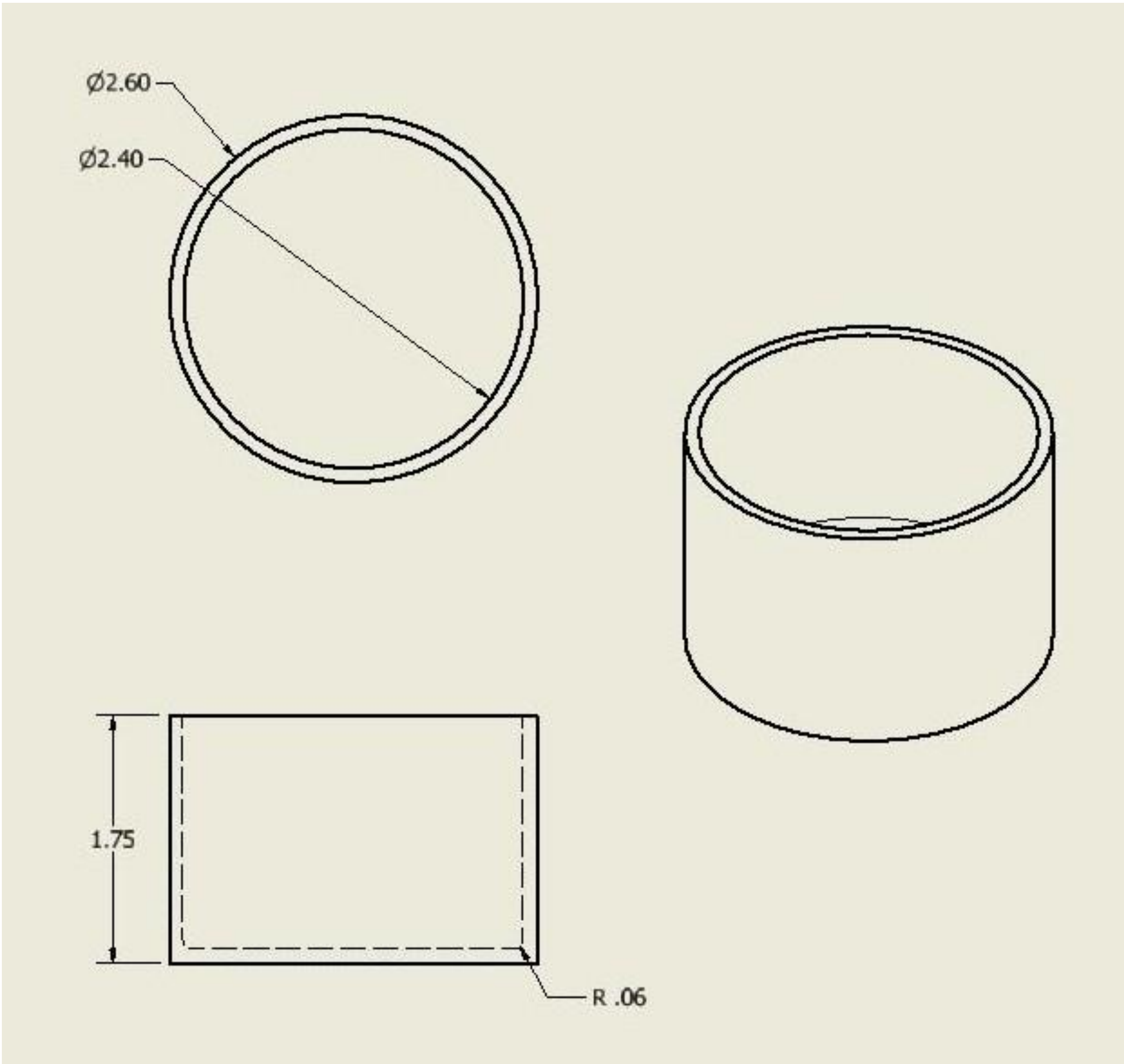
Wladis, C., Conway, K., Hachey, A. (2015). The online STEM classroom--Who succeeds? An exploration of the impact of ethnicity, gender, and non-traditional student characteristics in the community college context. *Community College Review*, 43. 142-164. Retrieved on 1/28/2021, from https://doi.org/10.1177/0091552115571729.

Xiaozhou, Yang. (2000, May). Linear discriminant analysis, explained. *Towards Data Science*. Retrieved on 6/6/2021, from https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b

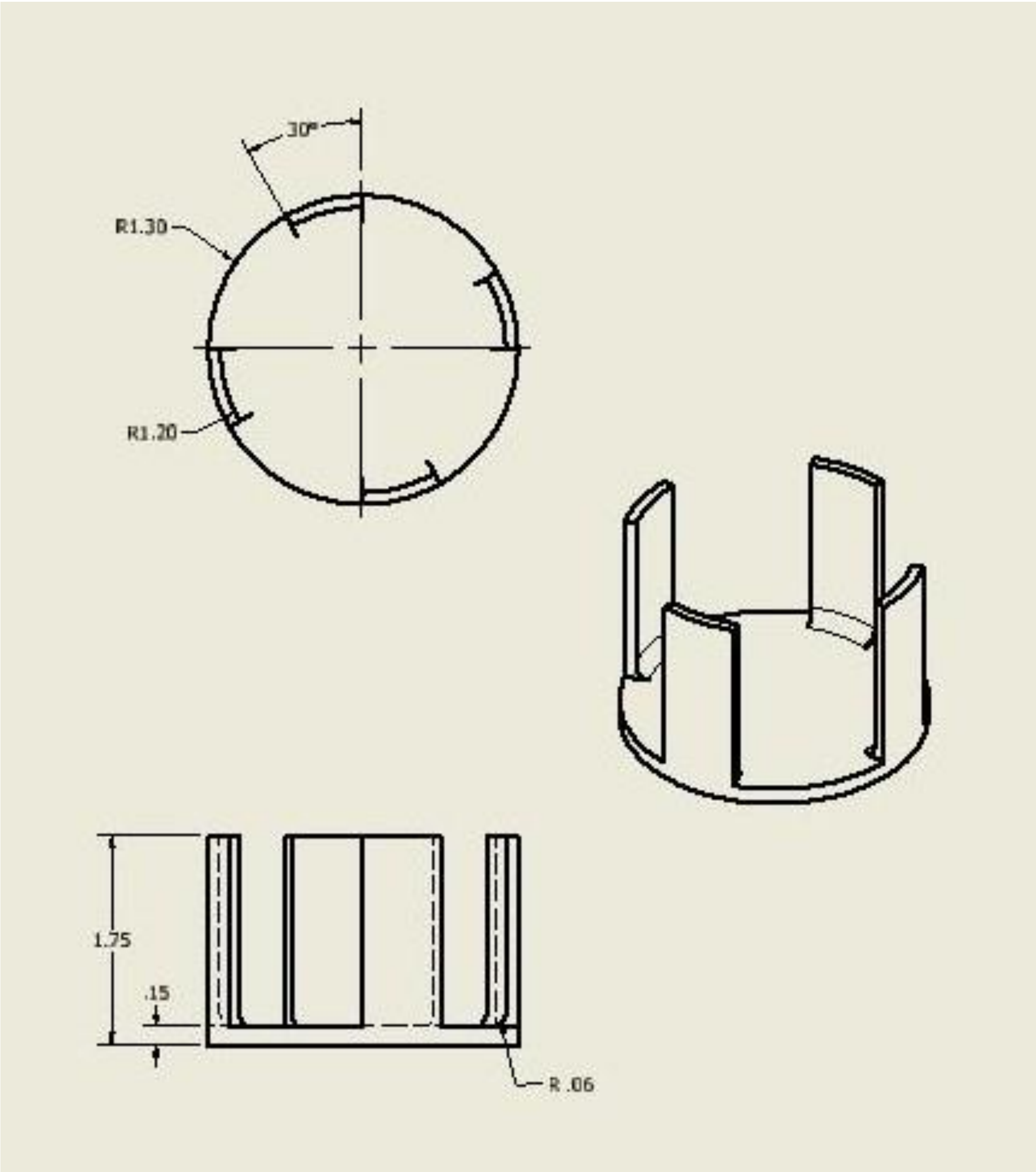
Ziegler, J. G., Nicholas, N. B. (1943). Optimum settings for automatic controllers. *ASME.* Retrieved on 3/14/2021, from http://www.davidr.no/iia1117/papers/Ziegler_Nichols_%201942.pdf

Zhu, W., Wang, S., Lin, R., Levinson, S. (2001, December, 8-14). *Tracking of an object with SVM regression* [Paper Presentation]. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, Hi, USA. Retrieved on 3/12/2021, from DOI: **10.1109/CVPR.2001.990966**

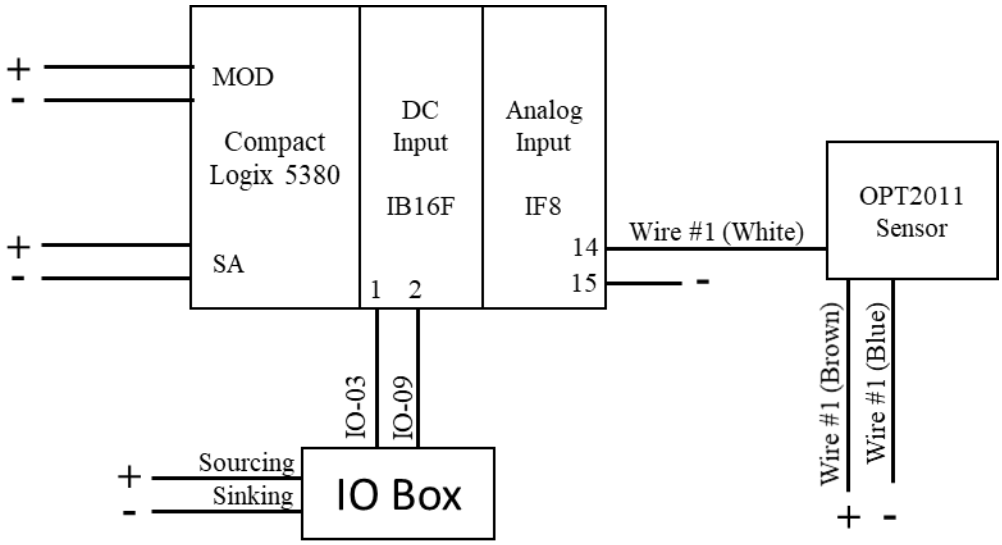
APPENDIX A DISC ONE DEMENSIONS



APPENDIX B DISC TWO DEMENSIONS



APPENDIX C WIRING SCHEMATIC



APPENDIX D MATLAB PID

```matlab
%Predictions
HeightPrediction = HeightPredictor_5it.predictFcn(S);
Height = HeightPrediction;


%PID Calculating VFD Frequency
error = 35 - Height;
integrals = integrals + (error * .06);
derivative = (error - pre_error) / dt;
CV = (Kp*error) + (Ki*integrals) + (Kd *derivative);
if CV > 6000
   CV = 6000;
elseif CV < 0
           CV = 0;
else CV = CV;
end
```