Computationally Deriving the Perfect Squash Boast

Jackson D. D. McAtee

Department of Physics, University of Richmond

PHYS 498: Senior Seminar

Dr. Christine Helms and Dr. Jerry Gilfoyle

March 24, 2021

Abstract:

The research we conducted was an extension of a theoretical derivation of the perfect squash boast. We used the computational Euler-Cromer method to solve our set of differential equations derived from Newtonian mechanics. Then, we used a Monte Carlo simulation and High-Performance Computing to repeat the process 500,000 times. Finally, we used the k-nearest neighbors algorithm to predict and outline the continuous spaces of initial angles and velocities that resulted in certain shot results.

We found that out of 500,000 tests of our model using randomized initial conditions, 160,680 resulted in a boast. Of those 160,680 boasts, 76 resulted in a perfect boast: a success rate of .05%, given that the shot was a boast. After applying the k-nearest neighbors algorithm to our testing data, we found that our model resulted in a classification rate of 95.7% for the six types of shot categories we wanted to identify. We conducted this experiment to better understand the game of squash and try to provide scientific evidence for the conditions of a winning boast.

Introduction:

Squash is a game of geometry and physics, and thus an excellent case study to model classical mechanics. The sport is played by two players in a court with four walls. The game is played by hitting a rubber ball with a racquet against the front wall in alternation with the opponent. The objective of which is to gain points by hitting a shot that the opponent cannot return to the front wall. The squash ball is hollow and has relatively thick walls. Thus, the nature of the bounce of a squash ball is largely subject to the internal pressure of the ball and its temperature, which affect its coefficient of restitution upon contact with the walls (Lewis & Arnold, 2010).

There are different types of balls that are used depending on the skill levels of the players. In this paper, we will consider the double yellow dot ball which is used by the most advanced players. This ball is the smallest of the options and has the lowest coefficient of restitution, meaning it will bounce the least. The low bounce property intensifies the aerobic activity required to play the game.

During gameplay, the optimal tactic is to tire the opponent out by making them move around to different corners of the court while maintaining an advantageous position in the middle, commonly referred to as the “T”. Players that hold position at the T can more or less control the dynamic of the rally by speeding up the pace with a volley or moving the opponent from the back of the court to the front of the court by utilizing various drop shots. The opponent not on the T will typically aim to hit a shot that requires the player on the T to move. Then, if possible, the opponent will try to take possession of the T. These successions will take place until a winning shot is hit or a player makes an error.

An important subset of the types of shots available to a player is a boast. The boast is a specific three-walled shot that a player can either hit as an offensive or defensive tactic, typically played from one of the back corners of the court. This can either be played by initially hitting the ball into the right or left side wall, followed by contact with the front wall, and ultimately contact with the opposing side wall or ground. In a time of distress, the player can use a boast to buy time and get back into position. This can be done by using an upwards trajectory and low velocity of the initial shot so that the ball travels slowly and completes a longer distance along the path of the boast. Conversely, if the boast is hit at a lower initial angle into the wall and at a higher speed, the ball will travel quickly along a shorter path. If hit correctly, the ball can end up in the front opposing corner of the court. Since the opponent will have to make a significant effort to move from their position to the front of the court, this can be leveraged as an offensive tactic.

The reason we decided to study the boast is because it is a complex and often misunderstood shot. Theoretically, if it is hit perfectly, it can result in a winner every time. For a boast to be completed it must be hit with an incident angle down from the z axis (the traditional θ in physics spherical coordinates), incident angle from the x axis in the x-y plane (the traditional ϕ in physics spherical coordinates), and initial velocity. This research defines the x direction as the width of the court, y direction as length, and z direction as the height, with the origin located at the back-left corner of the court. Along the ball's path, it will hit the first side wall, then the front wall, and end up somewhere on the opposing side wall (Figure 1, see appendix). In the instance of the perfect boast, referred to as a nick, the ball's contact with the second side wall will be at the line intersection of the x-y plane (the floor) and y-z plane (the left side wall) where x and z values are 0. When the ball hits this point of intersection, the ball essentially dies and is not able to be returned by the opposing player.

We sought to find a relationship between the incident angles and velocity to understand the likelihood of hitting a perfect boast from a certain location and the initial conditions required to do so. Although due to the limitations of the human reaction time, it is unlikely that a player will be able to process the exact initial conditions to hit the perfect boast in real time, the relationship we derived will serve as a fundamental insight to scholars of the sport. Additionally, this analysis is a precursor for the application of physics in squash, and will hopefully inspire a more scientific approach of the game.

Theories and Methods:

We used several computational methods to model the boast given an initial set of conditions, randomize these conditions and repeat the process 500,000 times, and predict which initial conditions are required to hit a certain type of boast. The first step we took was to derive a model of a projectile and track its motion along its shot path. We split the model into three different projectile paths: the path of the ball hit to the first side wall, the path from the first side wall to the front wall, and the path from the front wall to the opposing side wall. The model of these paths incorporates the effect of gravity and a drag force due to air resistance (Figure 2, see appendix). We accounted for energy loss due to collision with the walls using the coefficient of restitution for a ball at 40 degrees Celsius. We extrapolated this number to be 0.62 from research done in 2013 (*The Influence*, n.d.). The underlying assumptions for the rest of our model are that the ball is a point mass, it is not subject to rotational motion, and the ball is always 40 degrees Celsius. Using these assumptions, we derived three differential equations and used the Euler-Cromer method to track the position of the ball over time. We used Newtonian mechanics to construct our initial force equations[1] using $F = m \cdot a$ and rearranged them to get a set of three differential equations describing our motion:

$$\frac{d^2x}{dt^2} = -\frac{k}{m} \cdot |v_x| \cdot v_x$$

[1] g is the acceleration due to gravity on earth, m is the mass of the ball, k is the drag coefficient on the ball which can be calculated with the density of air, the drag coefficient of a sphere, and the cross-sectional area of the ball.

$$\frac{d^2y}{dt^2} = -\frac{k}{m} \cdot |v_y| \cdot v_y$$

$$\frac{d^2z}{dt^2} = -g - \frac{k}{m} \cdot |v_z| \cdot v_z$$

We decided to employ a computational rather than analytical method to solve this set of equations with the intention of adding effects of spin and rotational motion to the ball in future research, which cannot be solved analytically. Thus, our code is dynamic and readily available for these changes. The Euler-Cromer method uses a linear approximation to outline a set of two functions over an independent variable: in our case time. We used a step size of .001 seconds to solve these equations and outlined the path of the ball in three-dimensional space. We used Wolfram Mathematica to create a code that iterates through these equations and saves data points of the position of the ball over time. This outlined a parabolic path of the ball in three dimensions. Once the ball came into contact with the side wall, we stopped the iteration of the Euler-Cromer method. We then began a new block of code to solve for the second path of the ball with initial velocities in the x, y, and z components equal to the final velocities at the end of the first path multiplied by the coefficient of restitution of 0.62. Additionally, the component not in the plane of the wall that the ball hits changes the sign on its velocity. For example, when the ball traveling along path 1 hits the right wall (the y-z plane), the x component of velocity changes its sign. We repeated the identical process for the third path of the ball and aggregated our data points to show the total path of the ball in three dimensions over time (Figure 3, see appendix). Along the path of the ball, we created special conditions for which the ball did not complete the ideal three paths. For example, the ball may hit the second side wall before the front wall or hit the floor or ceiling before completing all three paths. We stopped the iteration of the code at these points and classified the result as a non-boast.

The code was dynamic in that we could easily change the initial starting point, velocity, angle into the side wall, and angle above the ground. We used spherical coordinates to identify this system, thus calling these angles ϕ and θ respectively. If we think about our model from a data science perspective, each time we ran our code our input is ϕ, θ, velocity and our output is a location in x, y, and z space after the ball completes the three paths, all assuming a fixed initial position of the shot. We used a Monte Carlo simulation and high-performance computing to repeat this process 500,000 times with randomized combination of ϕ, θ, and velocity, within reasonable ranges for a potential boast. Our dataset had columns of ϕ, θ, and velocity and final x, y, and z locations where each of the 500,000 shots represented an observation. We then defined a “Nick”, “Good Boast”, and “Bad Boast” and used our definitions along with Python and the pandas library to create a category column for type of boast. These categories were: the ball not completing a boast, the ball being hit in a way that it is “out” which is a loss of point for that player, a nick, a good boast that has an end location within a certain set of parameters, and a bad boast classified outside of those parameters.

Finally, we used the k-nearest neighbors algorithm to classify the outcome of the shot given our explanatory variables: initial velocity, ϕ, and θ. We used a training set and testing set of 350,000 and 150,000 observations respectively, with the data stratified to avoid sampling bias. We used the knn() function from the Scikit-learn library in Python to employ the k-nearest neighbors method for classification. To ensure that we maximized our classification rate, we used hyperparameter tuning to find the optimal value for k.

Results:

We successfully created a code that shows the path of a ball hit from an initial point on the court with any given ϕ, θ, and velocity (Figure 3, see appendix). Our three-dimensional model will only show a graph if the initial conditions result in a boast. As shown in the graph, the parabolic nature of each path is evident, but not obvious. This is because the typical squash boast takes a short period of time and the ball does not accelerate downwards long enough for this to be visually overt. We were able to better interpret the motion of the ball by isolating the positional components in x, y, and z over time (Figure 4, see appendix).

Figure 4 shows the graphs for different combinations of initial θ and velocity values. The sharp points in the y and x components correspond to the ball hitting a wall and changing directions in that component. Concurrently at these points, there will be a loss of energy and a decrease in the slope of all components with respect to time. The z component will demonstrate initial angle into the wall followed by a parabolic downwards motion on each path due to the force of gravity.

We can examine figure 4 which shows four graphs with initial conditions of a constant $\phi = 0.95$ and combinations of velocity at 13 or 8 and $\theta = 1.4$ or $\theta = 1.2$. In the trials with an initial velocity of 13 meters per second, we observe that the ball does not actually lose height and remains fairly constant after contact with the first wall. This is because the ball completes a boast fairly quickly (around 2 seconds) and does not have enough time for the gravitational acceleration to make a significant impact on the motion. When the initial angle θ is adjusted to be 1.2 radians, we observe a similar behavior where the ball remains at a higher height due to the sharper angle into the wall. At lower velocities, the ball actually falls in the z direction after its initial upwards motion. This is because the ball takes more time to complete the boast.

Each graph in figure 4 represents one shot given initial conditions. We stored the input conditions and output positions as a data point and repeated the process 3,000 times to create a three-dimensional graph with axes of initial conditions and point color corresponding to outcome of the shot (Figure 5, see appendix). This visual representation served as a qualitative indicator of the relationship between variables and an inspiration to leverage certain data science classification techniques. We reran the code 500,000 times with randomized initial conditions $45° \leq \theta \leq 100°,\ 20° \leq \varphi \leq 60°, 10\frac{m}{s} \leq v \leq 50\frac{m}{s}$, which are physically reasonable bounds for which a player with at least minimal skill would hit a boast. We found that over 60% of the shots traveled from initial side wall to opposing side wall without hitting the front: a shot that is considered invalid. 32% of shots completed some sort of boast being either a “Bad Boast”, “Good Boast”, or “Nick”. The remaining 8% were invalid shots. (Figure 6, see appendix). Looking strictly at shots that successfully completed a boast, the percentages were 86.52%, 13.43%, and .05% for “Bad Boast”, “Good Boast” and “Nick” respectively (Figure 7, see Appendix). Although our experiment required the person hitting the shot to have little to no skill since the model uses randomized conditions, the rarity of the nick is exemplified given its incredibly low probability. Moving forward, we could improve our model by using a normal distribution centered around optimal values of φ, θ, and velocity that lead to a good boast.

To help interpret our results from the 500,000 trials we employed the k-nearest neighbors algorithm. To ensure that our results were not subject to overfitting, we iterated through the different values of the hyperparameter k for both the training and the testing sets (Figure 8, see appendix). Our results showed that k = 7 maximized the classification rate in our testing set with an accuracy of 95.7%. This means that given a set of initial conditions, our model correctly predicts the outcome of the shot from seven possible categories with an accuracy of 95.7% on the unseen testing dataset. Notice that we did not choose the hyperparameter k = 1, although this does have a higher classification rate in the testing set than k = 7. We decided to use the higher k value to eliminate potential overfitting on unseen datasets than can arise from such a low k-value.

To create a visual representation of our k-nearest neighbors model, we ignored velocity and looked at the continuous decision boundaries for combinations of ϕ and θ. This is a loose interpretation of how our model actually worked because it did not account for changes in velocity, however, it outlined several important ideas (Figure 9, see appendix). The thin dark blue line at around $0.82 \leq \varphi \leq 0.83$ corresponds to the small window necessary to hit a "Good Boast". This translates to roughly $47°$, which is intuitive in what a squash player might expect. Additionally, we see that the type of shot varies considerably more with ϕ than with θ, indicating that the player should focus on the depth of the boast into the side wall as opposed to the height. Since this representation does not control for initial velocity in the analysis, the results are not necessarily reliable and we hypothesize it to be the reason that there is a thin continuous region denoting a "Side" shot between the "Good Boast" and "Bad Boast" by slightly varying φ. Additionally, we believe this is the cause for the noise on the line of $\theta = 1.6$ and that this result is not an accurate representation of reality.

Conclusion:

Our research used the Euler-Cromer computational technique to solve a set of differential equations that model the path of a squash ball along a boast. We derived a code that output a location on the court at the end of the boast, given input variables of initial location of the player and initial ϕ, θ, and velocity. To find statistical support for our results, we used a Monte Carlo simulation to repeat this code 500,000 times on a high-performance computer using randomized values of initial ϕ, θ, and velocity while keeping initial location of the player constant.

After collecting these 500,000 observations, we created a code to take the quantitative information and classify the results into different types of shots. This took the form of "Bad Boast", "Good Boast", "Nick", and other categorical variables indicating that the ball either did not boast or was hit out of the court as an invalid shot. This formed a dataset of three explanatory variables which were the initial conditions and a target categorical variable, or the type of shot. Using this dataset, we used a classification technique from the machine learning field to model the boast for continuous values of ϕ, θ, and velocity.

We computationally proved that it is highly unlikely to hit a nick, but were able to outline the initial conditions that were most likely to result in this type of shot. After applying a classification model to the dataset that we created, we were able to achieve a classification rate of 95.7%. When we extended this model to a two-dimensional visual representation, we found that hitting a good boast varied more with ϕ than with θ. Thus, our conclusion is that a player hitting a boast should put more effort and practice into precision in their ϕ value than their θ value. In squash terms, this means that the player should focus on the depth into the side wall as opposed to the height on the side wall to hit a good boast.

Our research and data are subject to the assumptions that we made in model: the ball was a point mass, the ball is not subject to rotational spin, and the ball is a constant at 40 degrees Celsius with a coefficient of restitution of 0.62. These are unrealistic assumptions and could have large repercussions for the results of our research. Additionally, in our graphical representation of the k-nearest neighbors algorithm, we did not include the effect of velocity which produced several unreliable results. Moving forward in the research process, we would like to address and mitigate these three assumptions. There is current research on rotational spin upon contact of a tennis ball. We will try to replicate the results of the paper and determine if it is reasonable, apply them to the motion of a squash ball. If this is the case, we will computationally implement this factor into our code, which would change our final dataset. Additionally, this would affect the coefficient of restitution in the x, y, and z direction. Another avenue of further research is to conduct experimental tests to determine the proper parameters for our results.

The final step that we would like to continue with is implementation of different machine learning classification techniques. We chose k-nearest neighbors for its simplicity, but we would like to use other techniques such as logistic regression, support vector machines, decision trees, and random forest. Using these different techniques will give us different insights into importance of parameters as well as confidence in other model results.

Appendix:

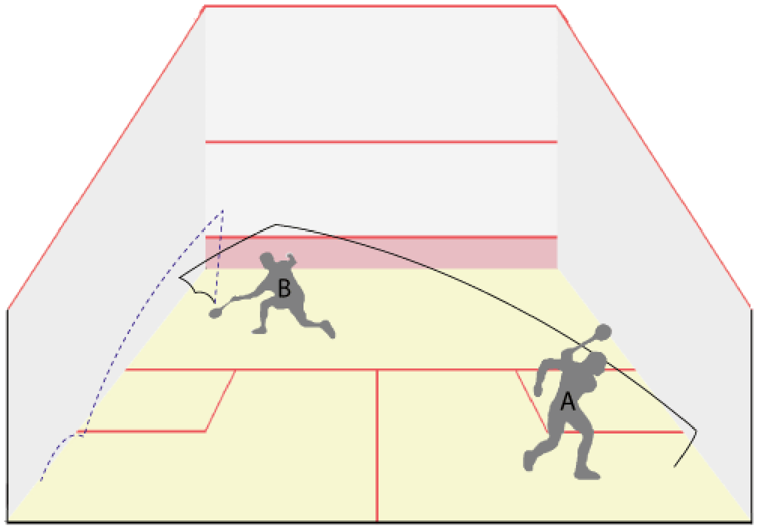

Figure 1: The picture above shows player A hitting the boast from the back right of the court as a defensive shot (*Boast - Drive*, n.d.). The perfect squash boast would end hitting the point where the left side wall meets the floor, making it extremely difficult for player B to return the shot.

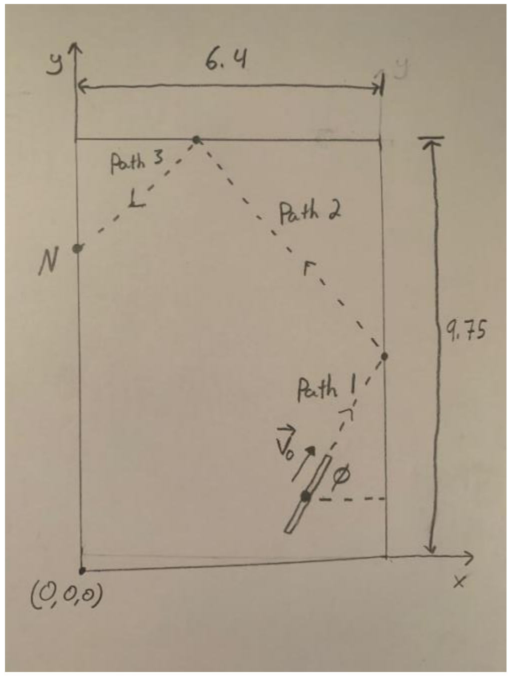

Figure 2: The figure above is a bird's eye view of the x-y plane of the squash court. It shows the three paths of the boast hit at an incident angle ϕ from the x-z plane.

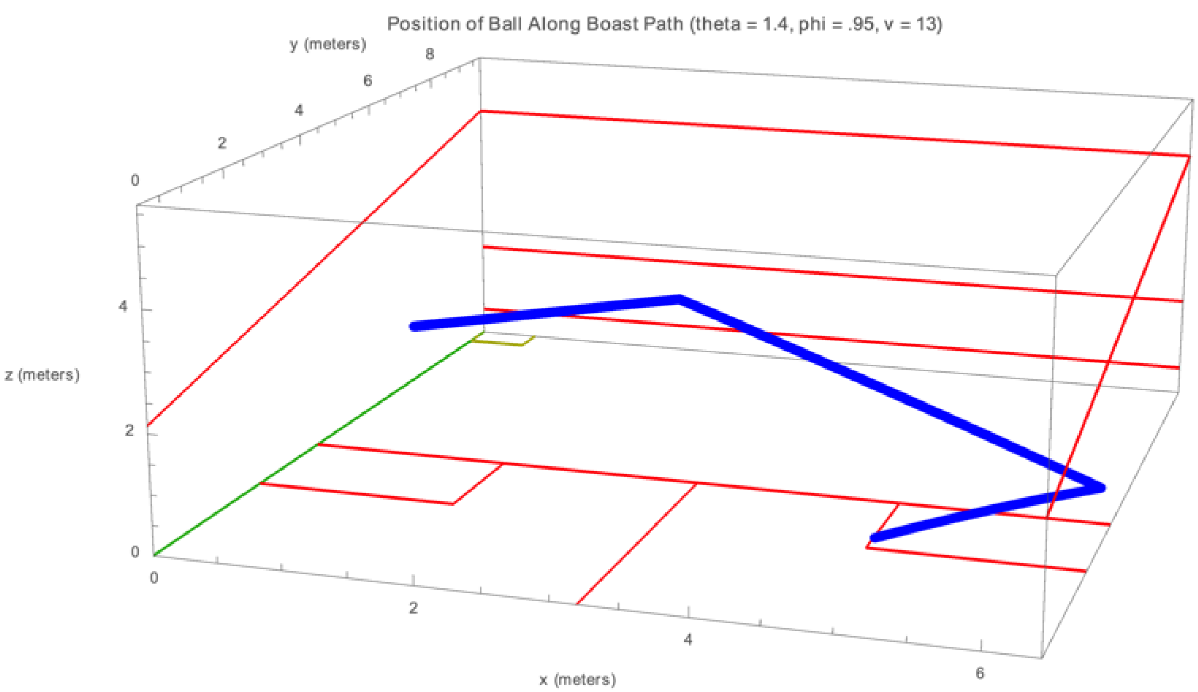

Figure 3: The figure above is a model of a boast hit from the point (5.01, 1.68, .67) with initial conditions $\theta = 1.6$ radians, $\phi = .95$ radians, and velocity of 13 meters per second. The green colored line is the line of the "Nick" and the yellow box is the area denoting a "Good Boast", which is a .5-meter * .5-meter area.

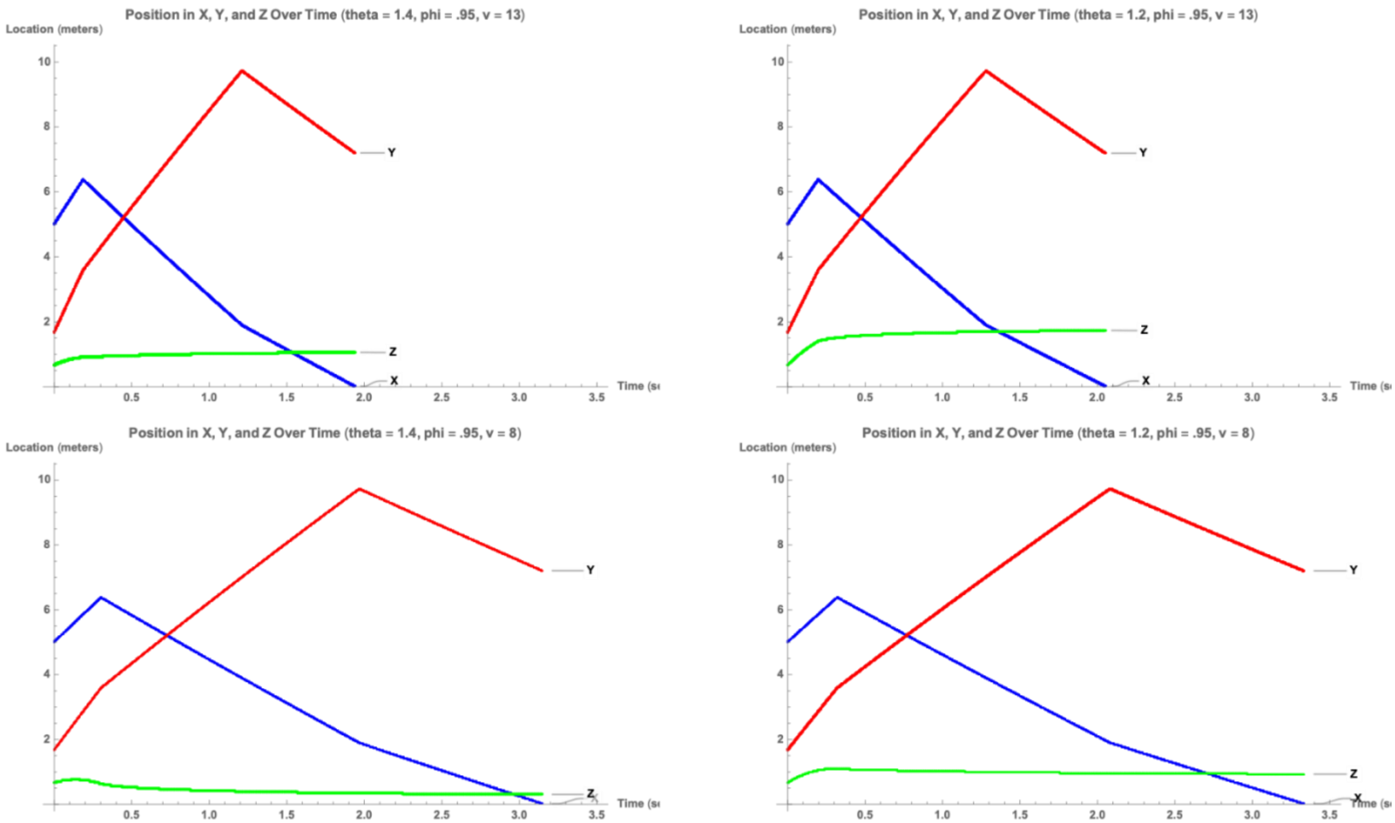

Figure 4: The figure above shows the position of x, y, and z over time for the initial conditions as noted above.

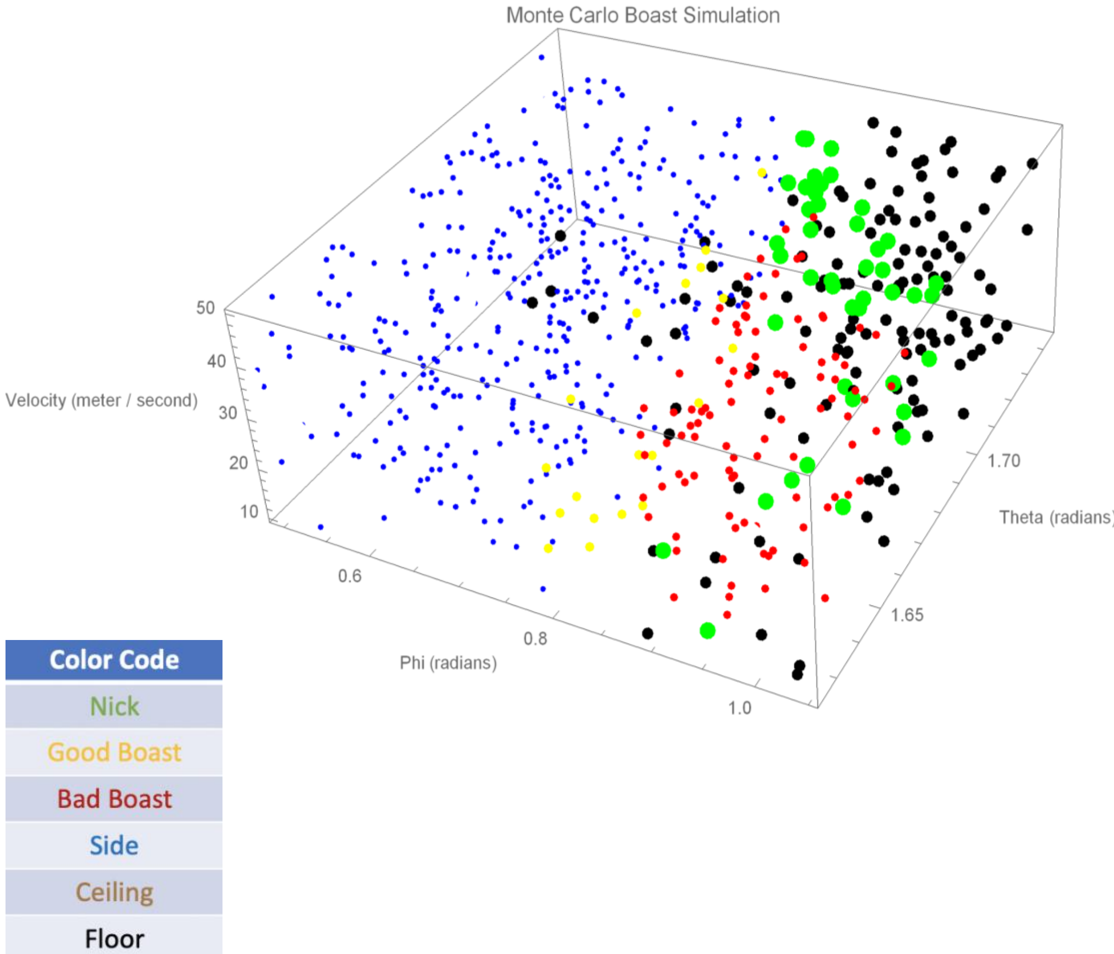

Figure 5: The figure above plots the initial conditions ϕ, θ and velocity of the 3,000 trials we attempted. The color code represents what the ball did along its path. Green corresponds to a ball hitting a nick, blue corresponds to a ball hitting the second side wall before the front wall, black corresponds to a ball hitting the floor before the front wall, yellow corresponds to the ball being a good boast, brown corresponds to the ceiling, and red corresponds to the ball being a bad boast.

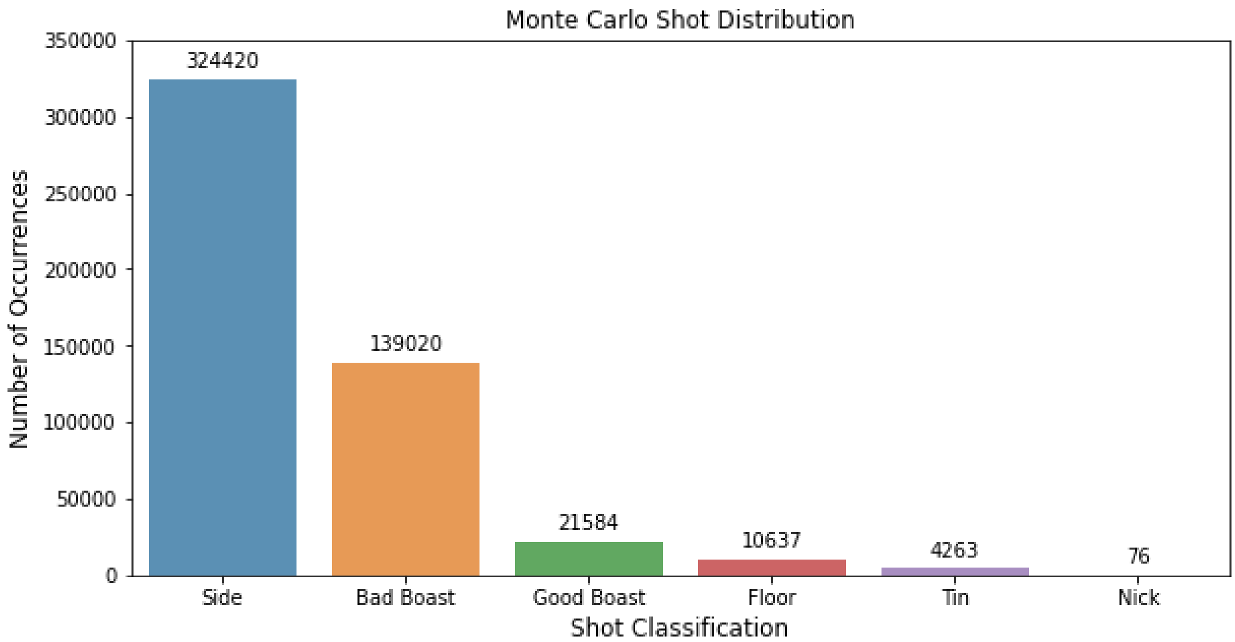

Figure 6: This figure shows the distribution of the 500,000 trials of shots given randomized initial conditions between the bounds $45° \leq \theta \leq 100°$, $20° \leq \varphi \leq 60°$, $10\frac{m}{s} \leq v \leq 50\frac{m}{s}$.

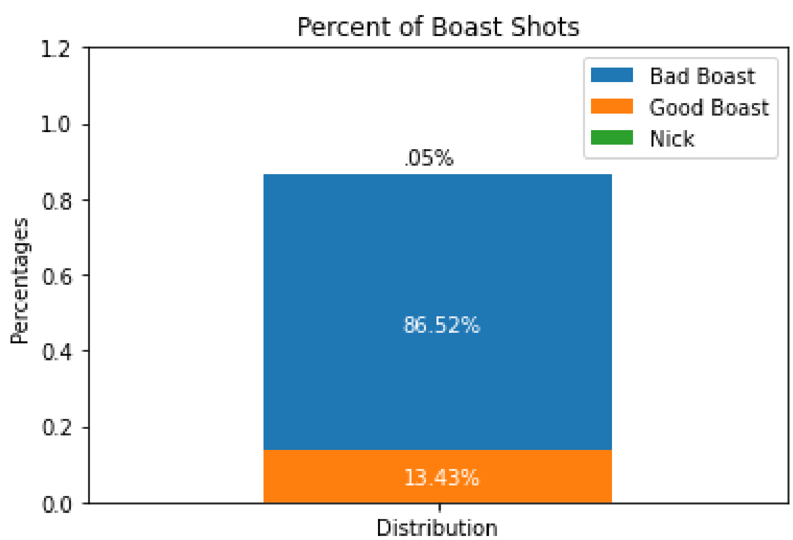

Figure 7: This figure shows the percentages of the 160,680 trials that completed a boast.

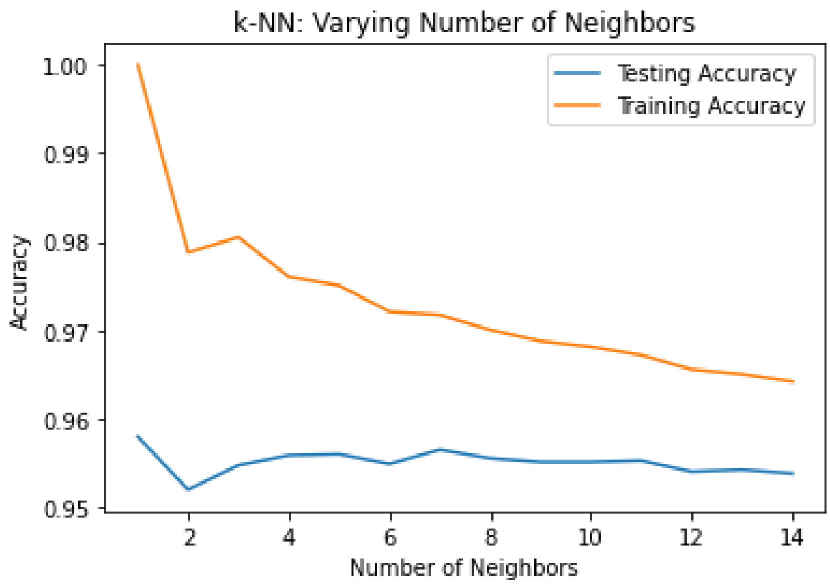

Figure 8: This figure shows the classification rates for the training set of 350,000 trials and testing set of 150,000 trials for different values of the hyperparameter k. We want to maximize the classification rate in the testing set without overfitting our model and thus chose the value k = 7.

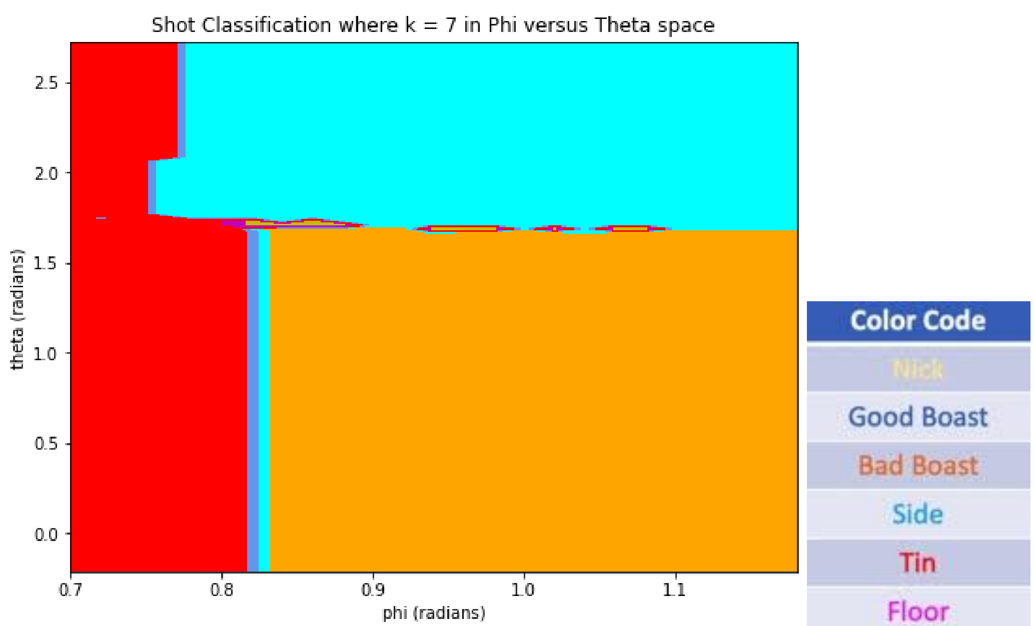

Figure 9: This figure shows the continuous decision boundaries for shot classification given initial conditions φ, θ, and velocity is held constant.

References

Boast - Drive [Photograph]. (n.d.). http://www.insidesquash.com/drills/boast-drive.html

The Influence of Temperature on Bouncing Balls. (n.d.). https://www.parliament.qld.gov.au/documents/committees/EIC/2013/QldAssessment/tp-20Mar2013-Assignment.pdf

Lewis, G. J., & Arnold, J. (2010). The dynamic behavior of squash balls. *American Journal of Physics*, *79*.