University of Waterloo

Department of Systems Design Engineering

BME 450: Sports Engineering

Lab 1: Golf Ball Trajectory Measurements and Simulation

Jordan Geurten (20559798)

October 12, 2018

**1 Introduction**

The purpose of the lab investigations was to learn analytical aerodynamic models to predict the 3D trajectory of golf shots based on initial launch conditions gathered on the Golf TeRF golf simulator. Using MATLAB’s ODE45 solver, the entire flight path of the golf ball can be solved and plotted.

**2 Data Collection**

All data were collected using the University of Waterloo Golf Teaching and Research Facility’s golf simulator where the initial launch conditions (speed, launch angle, azimuth, rotational velocities) and the apex and 3D final position of the ball is output.

**3 Modelling**

***Part I***

Using the provided aerodynamic coefficients and the initial launch conditions, the 3D trajectories for the 10 golf shots were simulated and plotted. The simulated carry distances, apex heights and their associated percent differences are presented below. It can be observed that the simulated carries are more accurate than the peak heights of the trajectories.

Table 1 - Experimental and Simulated Carry and Apex Values

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Shot ID** | **Experimental Carry (yds)** | **Simulated Carry (yds)** | **% Difference** | **Experimental Apex (yds)** | **Simulated Apex (yds)** | **% Difference** |
| 1 | 252 | 250.89 | 0.11 | 52.00 | 55.69 | 1.71 |
| 2 | 286 | 267.31 | 1.69 | 61.00 | 69.35 | 3.20 |
| 3 | 277 | 281.87 | 0.44 | 51.00 | 57.36 | 2.93 |
| 4 | 274 | 275.95 | 0.18 | 63.00 | 68.93 | 2.25 |
| 5 | 283 | 274.26 | 0.78 | 57.00 | 67.08 | 4.06 |
| 6 | 268 | 261.70 | 0.59 | 48.00 | 55.03 | 3.41 |
| 7 | 184 | 184.21 | 0.03 | 30.00 | 31.42 | 1.15 |
| 8 | 183 | 187.62 | 0.62 | 18.00 | 25.10 | 8.23 |
| 9 | 279 | 248.50 | 2.89 | 46.00 | 52.85 | 3.46 |
| 10 | 264 | 270.39 | 0.60 | 53.00 | 58.69 | 2.55 |

The 3 dimensional trajectories are presented below in Figure 1, the XY view of the trajectory in Figure 2 and the XZ view is presented in Figure 3.

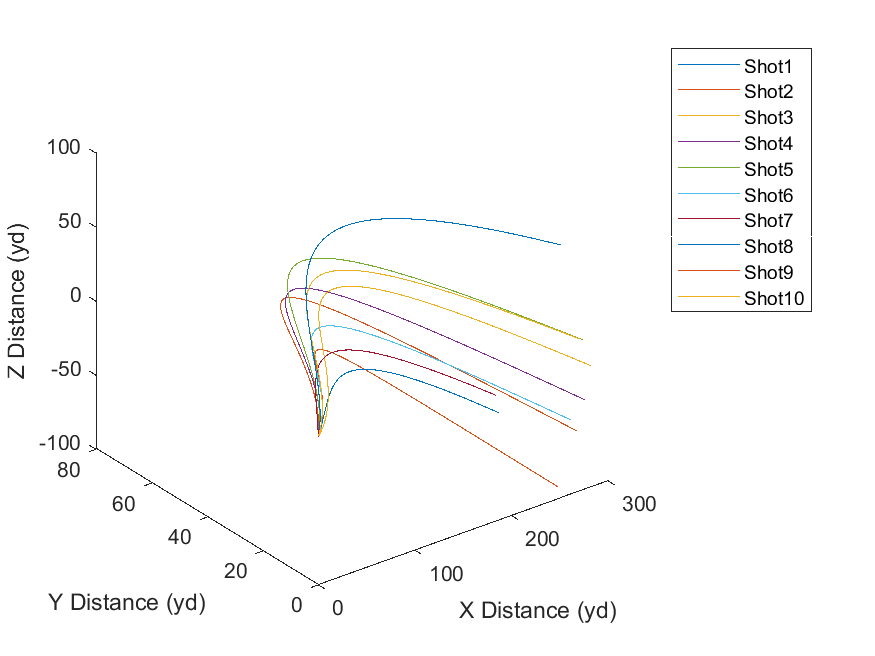


Figure 1 - All Shots 3D Trajectory

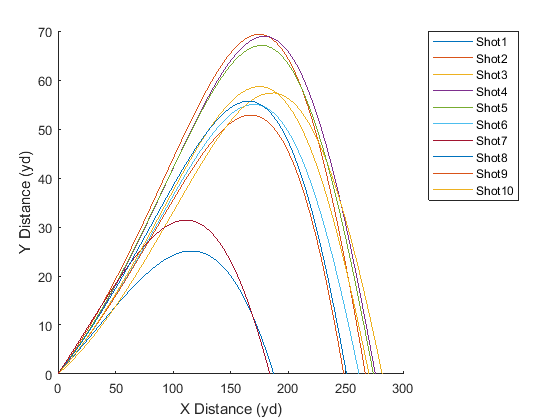


Figure 2 - All Shots 3D Trajectory XY View

The side view of the golf ball path is presented above. It can be observed that the peak height occurs at about 70-75% of the landing X distance. The overall geometry of the XY view of the 3D golf ball trajectory appears to be realistic. Prior to the peak, the ball’s high rotational backspin, magnus effect and large initial launch speed dominates the trajectory causing the peak height to occur after one half of the carry distance,

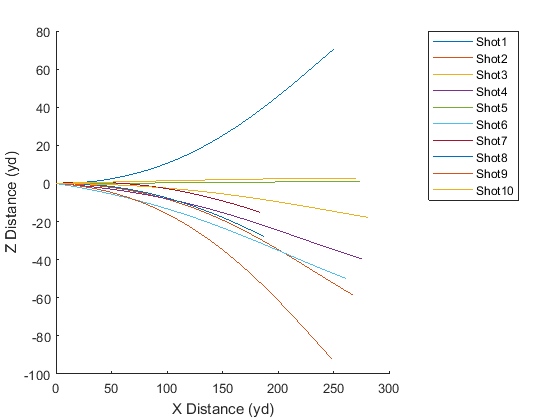


Figure 3 - All Shots 3D Trajectory XZ View

***Part II***

Using the initial parameters from shot 5, air density was varied for fixed initial speed, launch angle, azimuth, back and side spins. The carry, apex and offline distances are presented below as a function of air density.

Table - Golf Shot Trajectory Parameters as a Function of Air Density

|  |  |  |  |
| --- | --- | --- | --- |
| **Air Density** | **Carry (yds)** | **Apex (yds)** | **Offline (yds)** |
| 0.0012 | 377.4749 | 55.3865 | 1.2517 |
| 0.0015 | 352.4065 | 59.8102 | 1.2656 |
| 0.0017 | 327.7395 | 62.9838 | 1.2573 |
| 0.002 | 304.7721 | 65.1851 | 1.2365 |
| 0.0022 | 283.9044 | 66.5862 | 1.2092 |
| 0.0025 | 265.1305 | 67.442 | 1.1788 |
| 0.0028 | 248.2571 | 67.8576 | 1.1472 |
| 0.003 | 233.1019 | 67.9454 | 1.1155 |
| 0.0033 | 219.4505 | 67.8042 | 1.0845 |
| 0.0036 | 207.1134 | 67.4548 | 1.0544 |

The 3D trajectories are presented in the following 3 figures. The 3D view is presented in Figure 4, the XY view in Figure 5 and the XZ view in Figure 6.

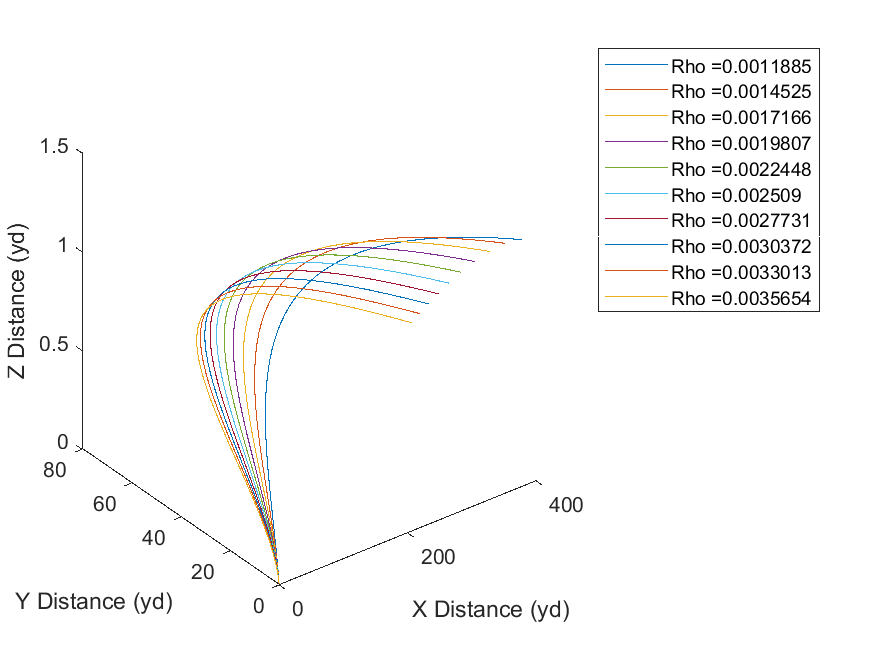
**

Figure 4 - 3D Trajectory as a Function of Air Density

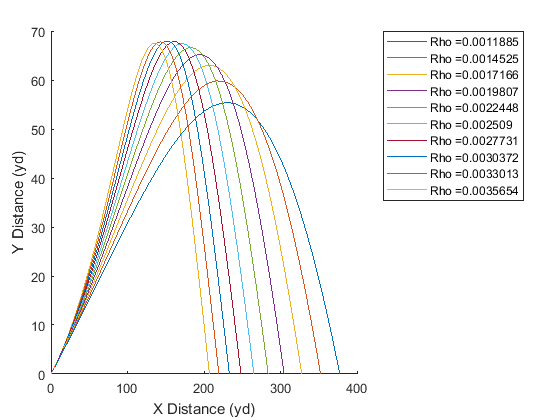
**

Figure 5 - 3D Trajectory as a Function of Air Density XY View

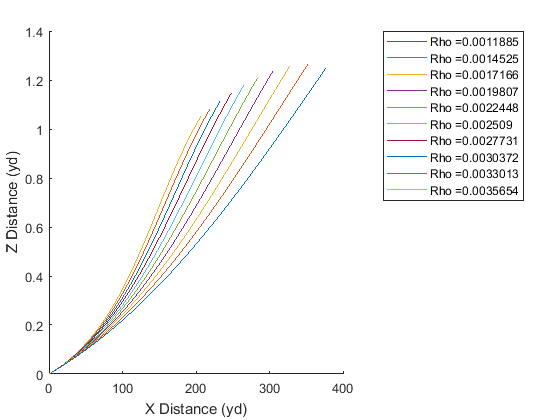
**

Figure 6 - 3D Trajectory as a Function of Air Density XZ View

**Based on your results, explain how playing at different altitudes might affect golf and other sports (e.g. 1968 Summer Olympics in Mexico City).**

It can be observed from the simulations presented in Figure 5 that as air density decreases, the shot carry increases. Furthermore, it is also noteworthy that a greater air density also leads to decrease in apex shot height. The drag force, or the retarding force of air molecules on the golf ball is given by the following formula,

where represents the drag coefficient, is the cross-sectional area of the golf ball, is the fluid density (air) and is the golf ball velocity. It can be observed that the drag force is directly correlated to the air density. Therefore, increasing the air density increases the drag force by the same factor. Since the forces acting in the X-direction are the magnus force (acting in the positive direction) and the drag force (acting in the negative direction, against the x-component of the ball velocity), a greater drag force results in a decrease in the x-component of the golf ball acceleration, resulting in a decreased carry. Consequently, a increase in altitude, and therefore a decrease in air density would allow players to drive the ball further with the same input force.

***Part III***

Using the MATLAB *fmincon* function, a set of aerodynamic coefficients presented below were determined which minimized the difference in the simulated and experimental peak heights and carry distances. The following cost function was written and fed into the *fmincon* function:

where W1, W2 and W3 represent weights 1, 2 and 3, respectively, and subscripts exp and sim represent experimental and simulated, respectively. Weight 1 was set to 0.5, weight 2 was set to 0.15 and weight 3 to 0.35 to weight the difference between the experimental and simulated position values greater than the difference in the experimental and simulated apex differences. By setting W1 and W3 greater than W2, an optimal solution favored an accurate position simulation over an accurate apex simulation.

Table 3 - Optimal Aerodynamic Coefficients

|  |  |
| --- | --- |
| Parameter | Value |
| CD | 0.1980 |
| CM | 0.1186 |
| CL | 0.2500 |

For each of the 10 shots, the simulated trajectories, using the optimal aerodynamic coefficients presented in Table 2, and the experimental trajectories are presented below in the next 10 figures.

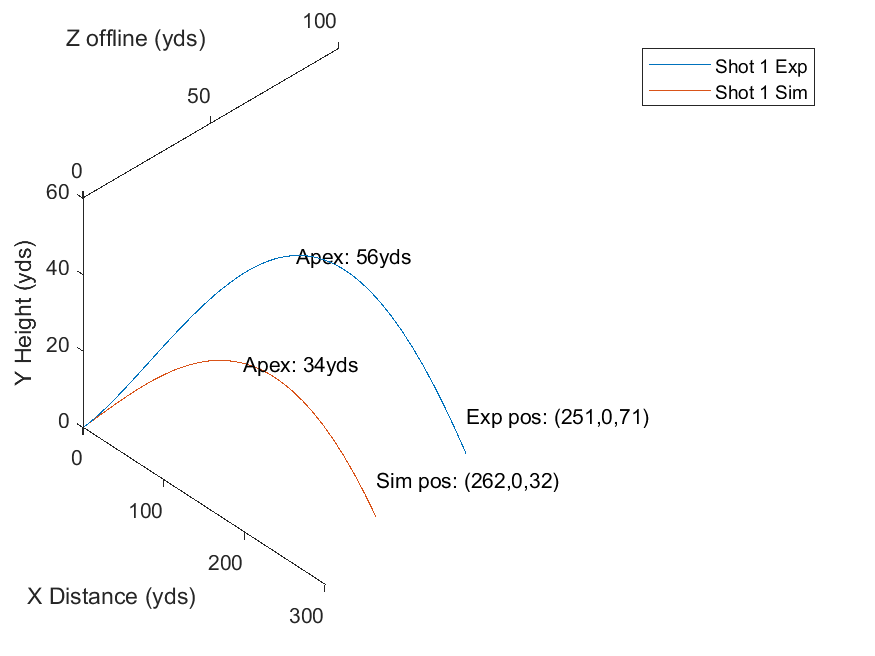


Figure 7 - Shot 1 Experimental versus Simulated Trajectory

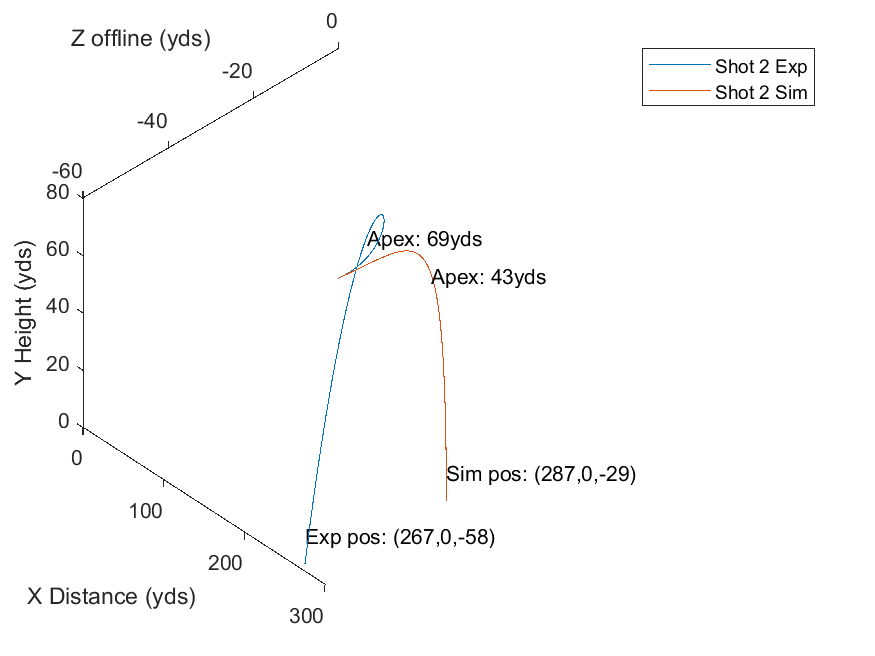


Figure 8 - Shot 2 Experimental versus Simulated Trajectory

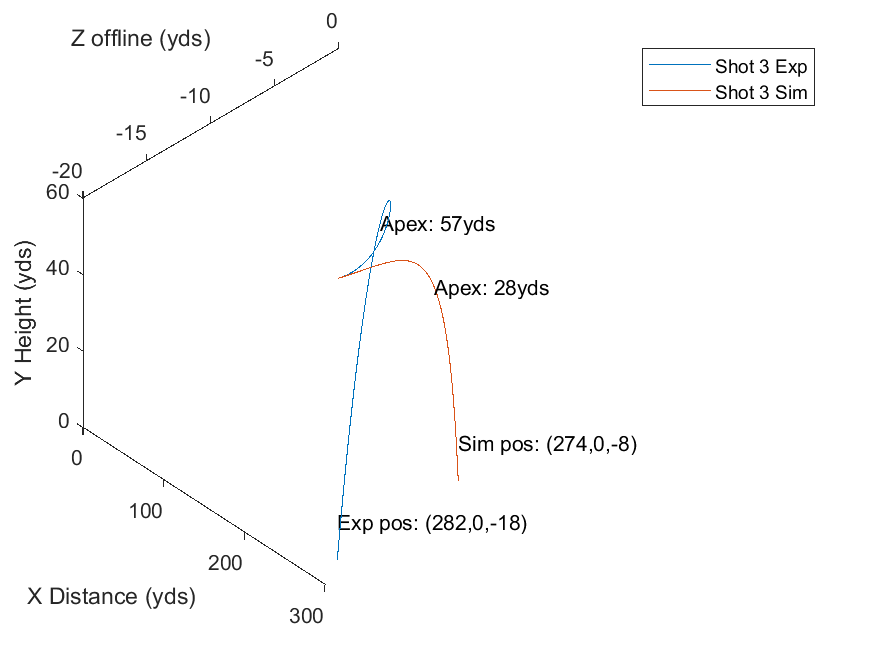


Figure 9 - Shot 3 Experimental versus Simulated Trajectory

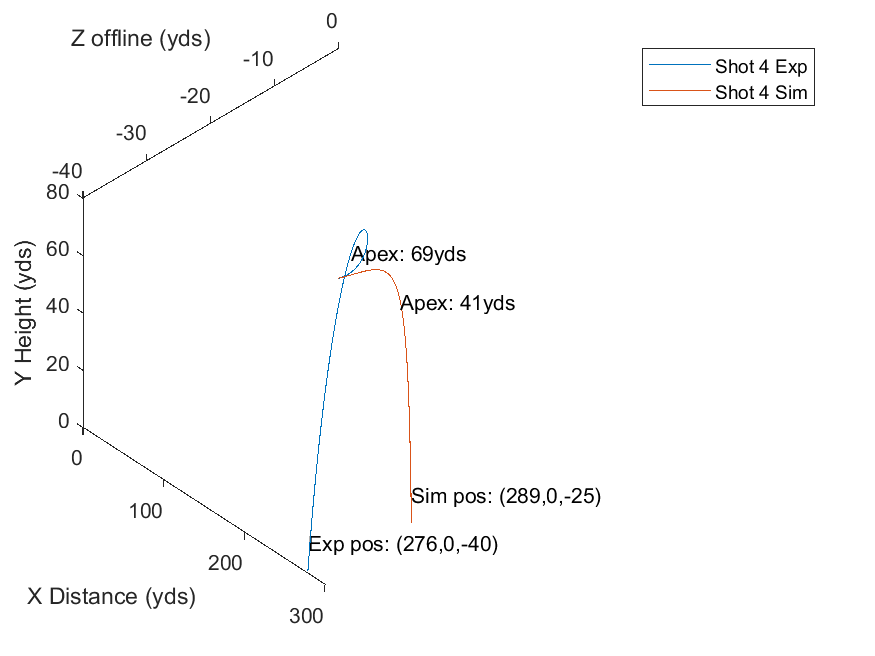


Figure 10 - Shot 4 Experimental versus Simulated Trajectory

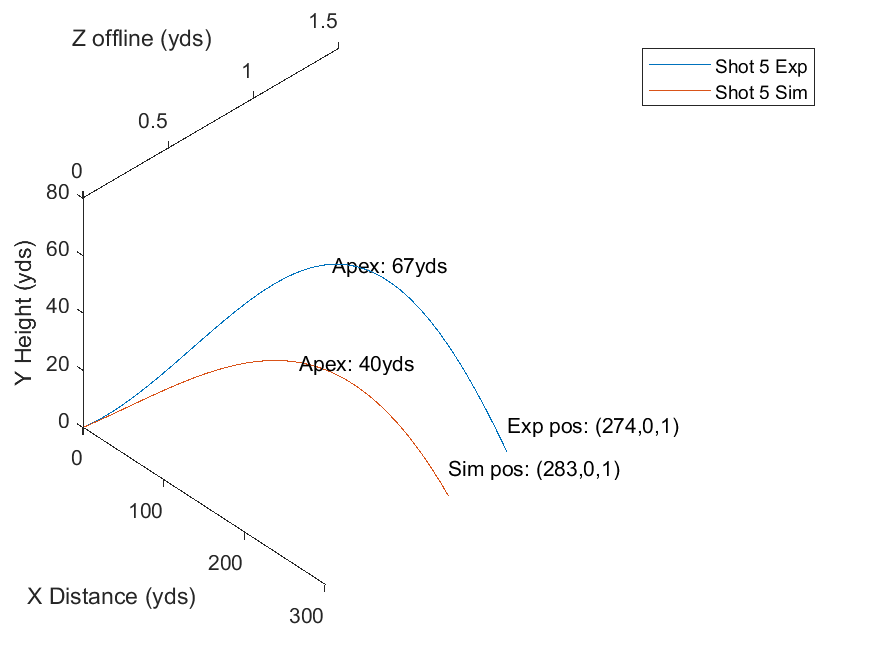


Figure 11 - Shot 5 Experimental versus Simulated Trajectory

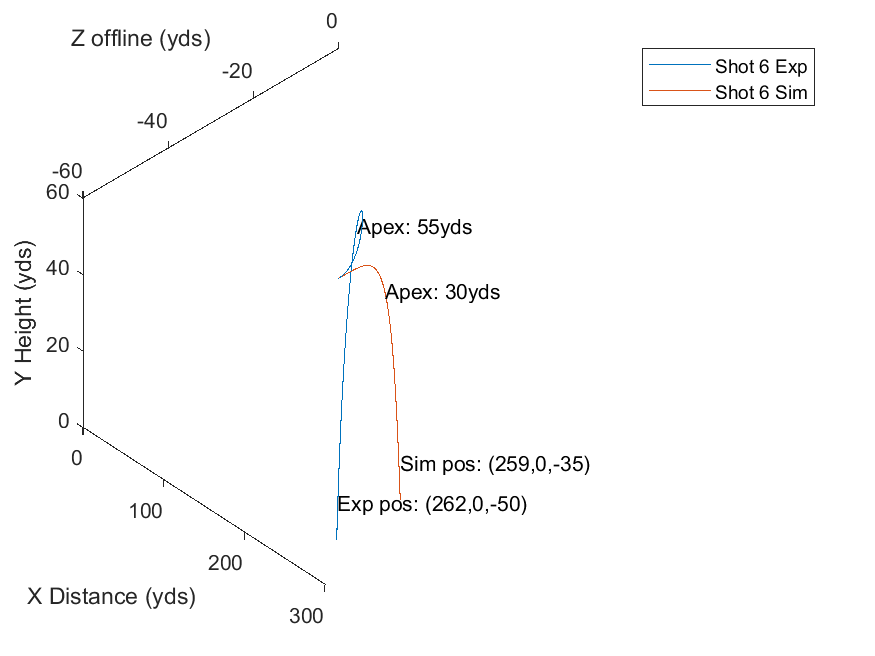


Figure 12 - Shot 6 Experimental versus Simulated Trajectory

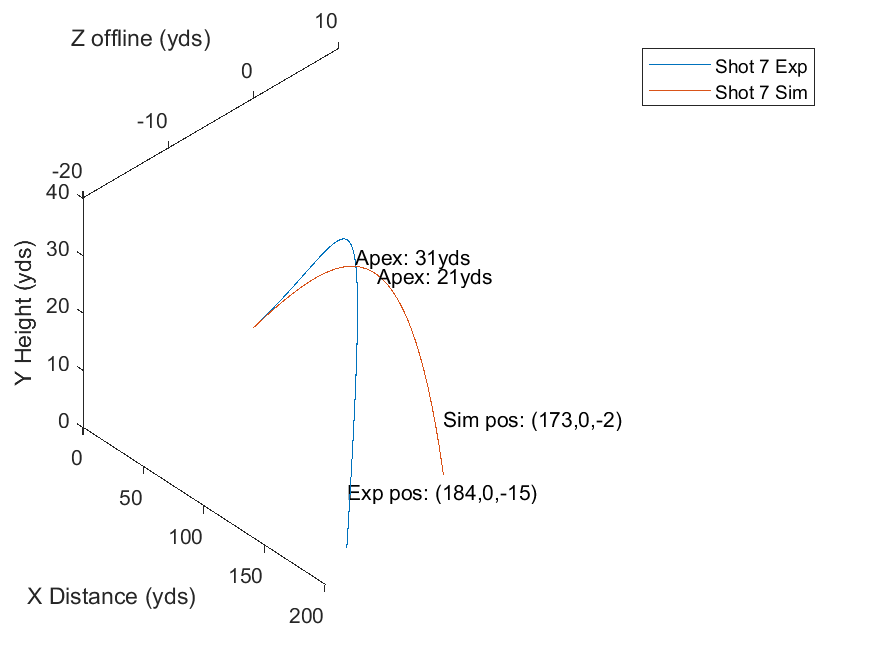


Figure 13 - Shot 7 Experimental versus Simulated Trajectory

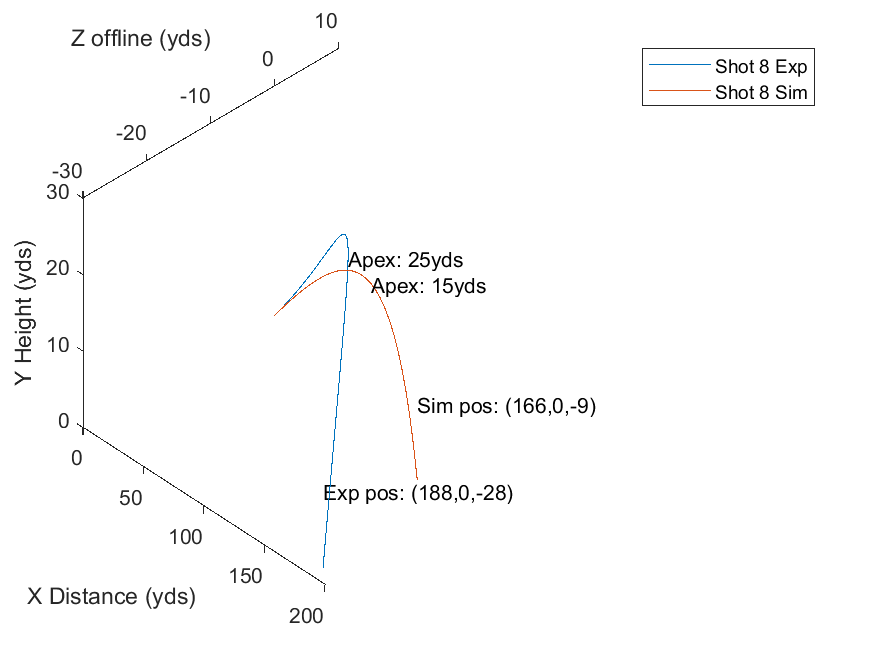


Figure 14 - Shot 8 Experimental versus Simulated Trajectory

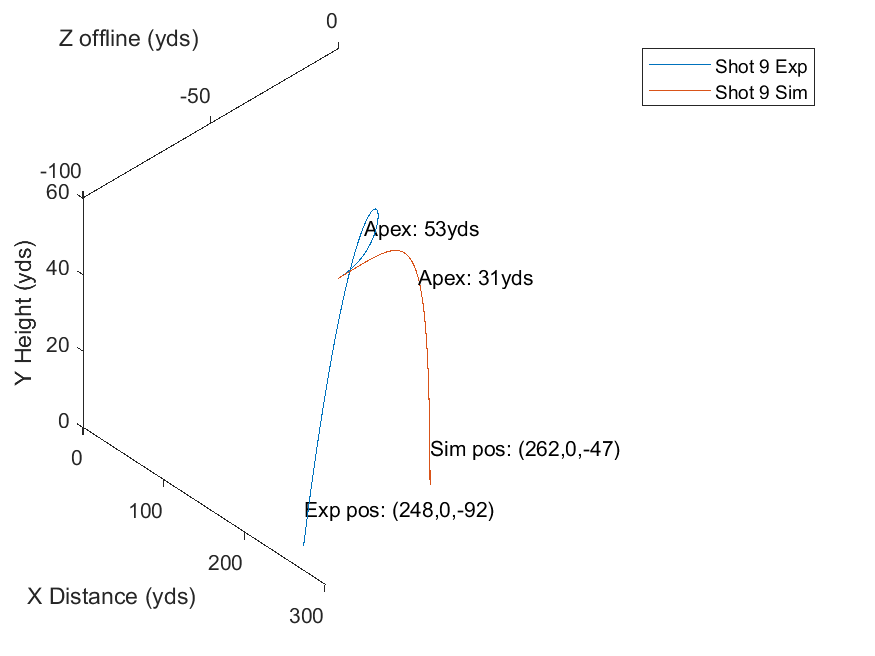


Figure 15 - Shot 9 Experimental versus Simulated Trajectory

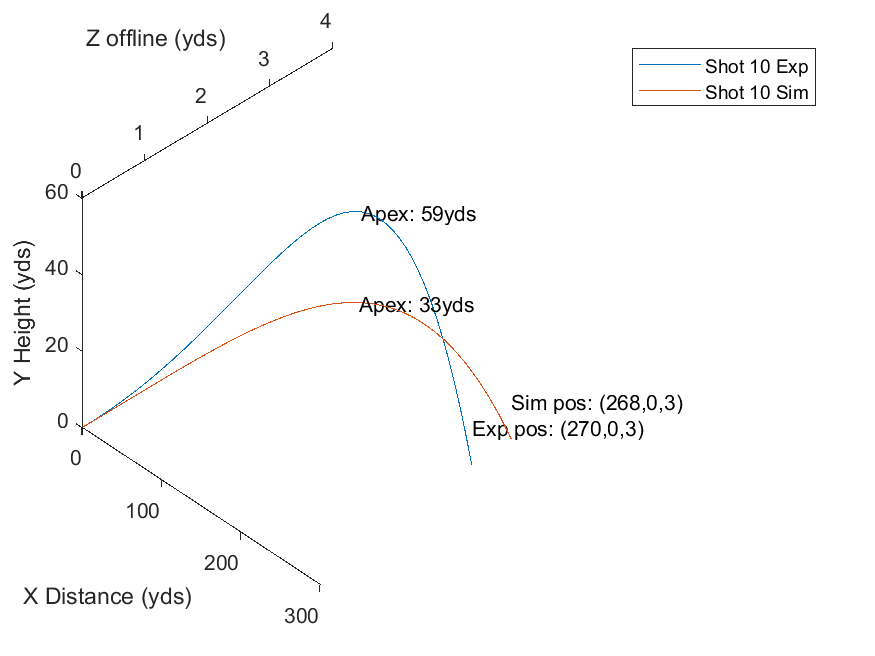


Figure 16 - Shot 10 Experimental versus Simulated Trajectory

The majority of the simulated shots (carry distances and offline distances) align well with the experimental data. The apexes for all shots vary significantly between the experimental and simulated

***Part IV***

The mean ball speed was calculated from the 10 shots to be 157 miles per hour, and fed into the *optimizeLaunch* function. The side spin and azimuth were set to zero, the launch angle was constrained to 0 – 45 degrees, and the back spin was constrained to 1000 – 4000 rotations per minute. Using the *fmincon* MATLAB function, and the optimal coefficients determined in *Part III,* optimal backspin speed and launch angle were determined to maximize the shot’s carry distance.

With multiple initial guesses for backspin of 1000, 2500, 3000 and 4000 rotations per minutes and a launch angle of 10, 22 and 40 degrees, a maximum distance of 276.4764 yards was always achieved. A unique launch angle was obtained for a variety of initial parameters, however, the optimal backspin varied. The optimal launch angle and most achievable backspin for an initial backspin guess of 2500 rpm are presented below.

Table - Optimal Initial Launch Conditions Maximizing Carry

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Backspin | 2500.0 rpm |
| Launch angle | 26.6 degrees |

The carry distance is plotted below as a function of initial launch angle in Figure 17. It can be observed that the optimal angle, 26.6 degrees (rounded to 27 degrees in the figure), does in fact maximize carry distance with a carry of 276.5 yards. It is interesting to note that a relatively acute angle a accomplishes maximum carry distance, unlike projectile motion with a smooth projectile where a launch angle of 45 degrees maximizes carry distance. Due to the dimpled surface on a golf ball, drag is reduced and lift is increased. The dimpled surface causes disturbance through the air leading to smaller air flow separation around the ball surface [1].

Although the rotational backspin speed is achievable, as the drive backspin speeds on the PGA tour range from 1500 – 6500 rpm, the optimal launch angle of 26.6 degrees is not feasible off of the tee using a commercial driver. The steepest launch angle on the PGA tour was recorded at 18.12 degrees by Brian Stuard [2]. The launch angle is determined by the loft of the golf club head. Most drivers have a loft of approximately 9-11 degrees. Since lift is proportional with the amount of backspin and launch speed of the ball [1], a more obtuse would put a larger backspin on the ball, causing the ball to “flare”, or to rise high and sacrifice carry distance [3]. The optimal angle of a golf shot is closer to 16 degrees [3]. The reason for the ambiguity between the derived optimal launch angle and the reported optimal angle is the limitation of the model. The backspin rotational speed does not influence the carry distance of this model, which is physically incorrect.

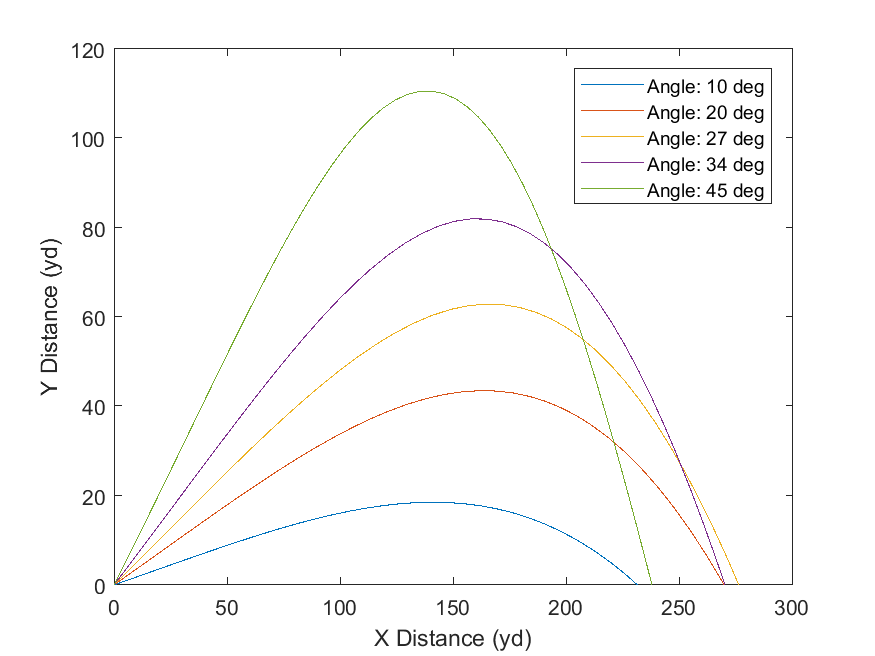


Figure - Golf Shot Trajectory versus Launch Angle

**References**

[1] T. Naruo and T. Mizota, “The Influence of Golf Ball Dimples on Aerodynamic Characteristics,” *Procedia Eng.*, vol. 72, pp. 780–785, 2014.

[2] “Statistics off the Tee: Launch Angle,” *PGA Tour*, 07-Oct-2018.

[3] Maximum projectile range with drag and lift, with particular application to golf. Herman Erlichson, American Journal of Physics 51:4 (1983), pp. 357-362.

**Appendix**

**A.1 Code**

**lab1.m – Lab Script**

%Lab 1 - Jordan Geurten for BME 450

%Read these in from .csv, and iterate through:

global SPEED\_COL ANGLE\_COL AZIM\_COL BACKSPIN\_COL SIDESPIN\_COL CARRY\_COL APEX\_COL

load('lab1data.mat');

SPEED\_COL = 2;

ANGLE\_COL = 3;

AZIM\_COL = 4;

BACKSPIN\_COL = 5;

SIDESPIN\_COL =6;

CARRY\_COL = 7;

APEX\_COL = 8;

rho\_in = 0.0023769;

Cd = 0.28;

Cl = 0.25;

Cm = 0.1;

aero\_coeffs = [Cd,Cl,Cm];

%From optimization section (0.25,0.25, 0.05)

aero\_coeffs = [0.25, 0.25, 0.05];

carry = zeros(10, 1);

apexes = zeros(10,1);

figure

for shotID = 1:size(lab1data, 1)

%Get X vector and final

v0 = table2array(lab1data(shotID, SPEED\_COL));

angle = table2array(lab1data(shotID, ANGLE\_COL));

azim = table2array(lab1data(shotID, AZIM\_COL));

backspin = table2array(lab1data(shotID, BACKSPIN\_COL));

side\_spin = table2array(lab1data(shotID, SIDESPIN\_COL));

%Add to the figure

[x, final,t] = simBallTrajectory([v0, azim, backspin, angle, side\_spin, rho\_in], aero\_coeffs);

dispName = strcat('Shot ', int2str(shotID));

plot3(x(1:final,4)/3,x(1:final,5)/3,x(1:final,6)/3, 'DisplayName', dispName);

carry = x(final,4)/3;

apexY = max(x(1:final,5)/3);

legend('-DynamicLegend');

hold on

end

xlabel('X Distance (yd)');

ylabel('Y Distance (yd)');

zlabel('Z Distance (yd)');

legend('show')

saveas(gcf, 'allShots.png');

%% Part 2 - Plot shot distance, X, as a function of air density

%Take one shot - take shot 5

%Get params for shot\_id 5

SHOT\_ID = 5;

rho\_default = 0.0023769;

v0 = table2array(lab1data(SHOT\_ID, SPEED\_COL));

angle = table2array(lab1data(SHOT\_ID, ANGLE\_COL));

azim = table2array(lab1data(SHOT\_ID, AZIM\_COL));

backspin = table2array(lab1data(SHOT\_ID, BACKSPIN\_COL));

side\_spin = table2array(lab1data(SHOT\_ID, SIDESPIN\_COL));

rho\_in = 0.5\*rho\_default:rho\_default/9\*(1.5-0.5):1.5\*rho\_default; %10 rho's to test

carry = zeros(10, 1);

apexes = zeros(10,1);

offline = zeros(10, 1);

count = 1;

figure

for rho = rho\_in

[x, final] = simBallTrajectory([v0, azim, backspin, angle, side\_spin, rho], aero\_coeffs);

dispName = strcat('Rho = ', num2str(rho));

carry(count) = x(final,4)/3;

offline(count) = x(final, 6)/3;

apexes(count) = max(x(1:final,5)/3);

count = count + 1;

plot3(x(1:final,4)/3,x(1:final,5)/3,x(1:final,6)/3, 'DisplayName', dispName);

legend('-DynamicLegend');

hold on

end

xlabel('X Distance (yd)');

ylabel('Y Distance (yd)');

zlabel('Z Distance (yd)');

legend('show')

saveas(gcf, 'shotFive\_RhoVariable.png');

%% Part 3 - Find Cl,Cd,Cm such that the difference between theoretical and simulated trajectories

guessCoeffs =[0.2,0.2,0.1];

lBounds = [0,0,0];

uBounds = [.5,.5,.5];

[optimalCoeffs, cost] = fmincon(@optimizeAero, guessCoeffs,[],[],[],[],lBounds, uBounds)

save('optimalCoeffs.mat', 'optimalCoeffs');

%optimalCoeffs1 = [.1713, .0847,.10]

%optimalCoeffs2 = [0.1241 0.0432 0.6342]

%% Part 3 - 2: Plot exp and sim trajectories

load('optimalCoeffs.mat')

for shotID = 1:size(lab1data, 1)

%Get experimental data

figure

v0 = table2array(lab1data(shotID, SPEED\_COL));

angle = table2array(lab1data(shotID, ANGLE\_COL));

azim = table2array(lab1data(shotID, AZIM\_COL));

backspin = table2array(lab1data(shotID, BACKSPIN\_COL));

side\_spin = table2array(lab1data(shotID, SIDESPIN\_COL));

%Experimental Trajectory

[x, final,t] = simBallTrajectory([v0, azim, backspin, angle, side\_spin, rho\_in], aero\_coeffs);

dispName = ['Shot ', int2str(shotID), ' Exp']; %strcat('Shot ', int2str(shotID), 'sim');

plot3(x(1:final,4)/3,x(1:final,5)/3,x(1:final,6)/3, 'DisplayName', dispName);

carry = x(final,4)/3;

[apexY, apexI] = max(x(1:final,5)/3);

txt = ['Apex: ', int2str(apexY), 'yds'];

text(x(apexI,4)/3,apexY, x(apexI, 6)/3,txt)

txt = ['Exp pos: (', int2str(carry), ',0,', int2str(x(final,6)/3), ')'];

text(carry,10,x(final,6)/3,txt);

hold on

%Simulated Trajectory

[x, final,t] = simBallTrajectory([v0, azim, backspin, angle, side\_spin, rho\_in], optimalCoeffs);

dispName = ['Shot ', int2str(shotID), ' Sim'];

plot3(x(1:final,4)/3,x(1:final,5)/3,x(1:final,6)/3, 'DisplayName', dispName);

carry = x(final,4)/3;

[apexY, apexI] = max(x(1:final,5)/3);

txt = ['Apex: ', int2str(apexY), 'yds'];

text(x(apexI,4)/3,apexY, x(apexI, 6)/3,txt)

txt = ['Sim pos: (', int2str(carry), ',0,', int2str(x(final,6)/3), ')'];

text(carry,10,x(final,6)/3,txt);

hold on

legend('show');

xlabel('X Distance (yds)');

ylabel('Y Height (yds)');

zlabel('Z offline (yds)');

view(3)

camup([0 1 0])

saveas(gcf, ['part3\_sim\_exp\_', int2str(shotID),'.png']);

end

%% Part 4 - Optimization between backspin and launch angle

lBounds = [1000, 0];

uBounds = [4000, 45];

initConds = [4000, 22]; %options = optimoptions('fmincon','Display','iter','Algorithm','sqp');

options = optimoptions('fmincon','Display','iter', 'OptimalityTolerance', 1e-12);

nonlcon = @unitdisk;

[optimalLaunchConds, distance] = fmincon(@optimizeLaunch, initConds,[],[],[],[],lBounds, uBounds, [], options)

save('optimalLaunch.mat', 'optimalLaunchConds');

angles = [10, optimalLaunchConds(2) - 7, optimalLaunchConds(2),optimalLaunchConds(2)+7,45 ];

%plot results:

avg\_velocity = 157;

%From optimization section

load('optimalCoeffs.mat');

aero\_coeffs = optimalCoeffs;

%Assume straight shot

side\_spin = 0;

azim = 0;

figure

for ang = angles

[x, final,t] = simBallTrajectory([avg\_velocity, azim, optimalLaunchConds(1), ang, side\_spin], aero\_coeffs);

dispName = ['Angle: ', int2str(ang), ' deg']; %, 'Angle: ', num2str(optimalLaunchConds(2)), ' deg', ];

plot(x(1:final,4)/3,x(1:final,5)/3, 'DisplayName', dispName);

legend('-DynamicLegend');

hold on

end

xlabel('X Distance (yd)');

ylabel('Y Distance (yd)');

zlabel('Z Distance (yd)');

legend('show')

saveas(gcf, 'optimizeLaunch.png')

**optimizeAero.m – aerodynamic coefficient optimizer (part III)**

**opimizeLaunch.m – launch conditions optimize (part IV)**

**simTrajectory.m – function to calculate ball trajectory**

**golf\_eqns.m – ODE45 integration function**