Predicting the Flight of a Golf Ball: Comparing a Physics-Based Aerodynamic Model to a Neural Network

Spencer Ferguson[1], William McNally[1], and John McPhee[1]
[1] Systems Design Engineering, University of Waterloo, 200 University Ave W, Waterloo, ON, N2L 3G1, Canada

Accurately predicting the flight of a golf ball using its initial launch conditions has great utility within the golf industry, with applications ranging from launch monitors to dynamic golfer models [1]. Aerodynamic models that achieve this are generally physics-based and solve a set of ordinary differential equations (ODEs) for the flight of the ball with time-varying aerodynamic coefficients [2]. This work develops both a physics-based aerodynamic model and a neural network from the same dataset to determine if neural networks show promise for this application.

Shots by golfers of varying skill level were recorded using 2021 Titleist Pro-V1 golf balls. A GCQuad camera-based launch monitor (Foresight Sports, San Diego, USA) was used to record the initial launch conditions: ball speed, vertical and horizontal launch angle, backspin, and sidespin. A FlightScope X3 radar tracking system (FlightScope, Orlando, USA) recorded the ball's carry distance, offline landing position, and apex. All shots were recorded with less than 1.3 m/s of wind at ground level, and at 190 m above sea level. In total, N = 1040 shots were recorded: 521 driver shots and 519 non-driver shots distributed among fairway woods, irons, and wedges. The combined dataset has a median carry distance of 166 m (183 yds), and was split 80%/20% for training and testing for both models.

The proposed physics-based aerodynamic model (PBAM) bases its aerodynamic coefficients on the ball's spin ratio S, given by equation (1), with ball radius r, total angular speed Ω, and total linear speed v. The equations for coefficient of drag (2) and lift (3) were assumed to be second-order functions of S, and the moment coefficient (4) was assumed to vary linearly. Unknown constants are denoted by a through g. For the training dataset, the spin ratio at launch varies between 0.02 (a driver shot with high ball speed and low spin) and 0.75 (a wedge shot with low ball speed and high spin).

$$S = r\Omega / v \tag{1}$$
$$C_D = a + bS + cS^2 \tag{2}$$
$$C_L = d + eS + fS^2 \tag{3}$$
$$C_M = gS \tag{4}$$

Corresponding author email: psgfergu@uwaterloo.ca

Each set of launch conditions from the training dataset was simulated using the ball flight ODEs, with MATLAB 2021a's *fmincon* optimization algorithm determining the unknown constants *a* through *g*. The cost function minimized the sum of the squared differences between the simulated and experimental carry, apex, and offline distances for the entire training dataset. Constants *a* through *f* were bounded between +2 and -2, while *g* was bounded between 0.010 and 0.015 to reflect realistic spin decay [3]. The constants were found to be {0.1304, 0.9287, -0.8259, 0.0504, 1.2031, -1.1490, 0.01}, respectively.

A two-layer feedforward artificial neural network (NN) was used to map the five input launch conditions to the outputs of interest: carry, offline, and apex. Each layer used 256 nodes, batch normalization, and the ReLU activation function. The network was trained for 300 epochs using a batch size of 16, a mean squared error loss, and the Adam optimization algorithm. The initial learning rate was set to 0.001 and was decayed using cosine annealing. The inputs and outputs were normalized using min-max normalization. TensorFlow 2.4 was used for the implementation.

Table 1 shows the mean absolute error (MAE) and percentage error (MAPE) for both the PBAM and NN in predicting carry, offline, and apex for the test dataset.

Table 1: MAPE and MAE comparison between the PBAM and NN for the test dataset

Output	**PBAM: MAPE, MAE [m] (yds)**	**NN: MAPE, MAE [m] (yds)**
Carry	1.52%, 2.51 (2.74)	10.8%, 15.0 (16.4)
Offline	28.4%, 1.54 (1.68)	169%, 5.45 (5.96)
Apex	5.08%, 1.17 (1.28)	35.0%, 5.06 (5.53)

The PBAM performs much better than the NN in predicting all three ball flight outputs, as outlined in Table 1. A PBAM has the inherent advantage of containing the ball's equations of motion, meaning just a small number of training shots are required to develop a robust model. On the other hand, a NN has no inherent understanding of the flight of a golf ball, and would likely require many orders of magnitude more training shots to produce comparable results. Better results could be achieved with both the PBAM and NN by focusing on shots from just one club, but this would limit the applicability of the models significantly.

1. McNally W, McPhee J (2018) Dynamic Optimization of the Golf Swing Using a Six Degree-of-Freedom Biomechanical Model. *Proceedings* 2(6), 243.
2. Quintavalla SJ (2002) A Generally Applicable Model for the Aerodynamic Behavior of Golf Balls. In: Thain E (ed.) Science and Golf IV, Routledge, Abingdon, pp 341-348
3. Smits AJ, Smith DR (1994) A New Aerodynamic Model of a Golf Ball in Flight. In: Cochran AJ and Farrally MR (eds.) Science and Golf II, E & FN Spon, London, pp 340-347

Corresponding author email: psgfergu@uwaterloo.ca