

Understanding the Effects of Spin In Table Tennis Through Empirical Modelling

0707837

Abstract

Spin is an important aspect of table tennis which has a profound effect on both the trajectory of a table tennis ball and on how the ball bounces off a bat. This paper uses Empirical Modelling to demonstrate how topspin and backspin affect the trajectory of a table tennis ball and how the different types of spin affect the way the ball bounces off a bat.

1 Introduction

This paper shows how EM techniques have been used to create a model of a table tennis ball's trajectory under different types of spin and how different types of spin affect the way the ball bounces off a bat. Previous projects which are related to this model include Bridge (1991) which is a cruise control simulation containing dynamic observables and (MacDonald, 1998) which uses physical laws within its model. Modelling the effects of spin will include both dynamic observables such as the ball's position, velocity and spin as well as classical mechanics used to describe the motion of the ball under the effect of spin.

2 Research

2.1 Table Tennis

An introduction to topspin and backspin is given below for those unfamiliar with them.

Topspin occurs when the top half of the ball is rotating in the direction of the ball's trajectory. Watching a topspin ball from the side on view moving from left to right the ball would be rotating clockwise. Topspin causes the ball to dip more than a ball with no spin once the ball has reached the apex of the ball's trajectory. Once the ball bounces on the table the topspin accelerates the ball causing it to jump forward and upon contact with the opponent's bat the ball will jump upwards. The effects of topspin allow the ball to be hit much harder than a ball with no spin or backspin and still make contact with the other side of the table. Furthermore the resulting jumping up of the ball after making contact with the opponents bat results in the opponent having to compensate for the topspin by altering the angle of their bat.

Backspin occurs when the bottom half of the ball

is rotating in the direction of the ball's trajectory. It can be considered the opposite of topspin. Backspin causes the ball to initially lift or float however upon contact with the opponents bat the ball will bounce downwards towards the table. The initial lift on the ball limits the speed at which the ball can be hit for it to still bounce on the other side of the table. The downward bounce caused by backspin forces an opponent to compensate by altering their bat angle.

There are a variety of different types of rubbers which can be used on a table tennis bat. These rubbers have different properties which affect how a table tennis ball with a particular spin bounces off the bat as well as the spin on the ball after the bounce. Inverted rubbers are the most common type of rubbers used in table tennis and will be used for the bat in the model.

2.2 Simulating Ball Physics

Creating a simulation of how spin affects a table tennis ball required knowing what forces are acting on a spinning ball while it is moving through the air. Whilst the trajectory of a ball only under the influence of gravity was rather simple to calculate. Adding topspin and backspin into the equation made things more complicated. It quickly became apparent that creating an extremely realistic simulation would require a lot of work and therefore a simpler approach should be taken. The equations for the motion of the ball including collisions was obtained from Denn (2002) which includes a lot of information on physics for computer games.

2.3 EM Tools

The model will be creating using the Empirical Modelling tool tkeden which supports multiple definitive

notations (Beynon, 2010). The definitive notations used in the model include Eden (Beynon, 2006b), DoNaLD (Beynon, 2006a), Scout (Beynon, 2006c) and Angel (Harfield, 2006). Eden is used to calculate where to position the ball during the simulation and to update the simulation repeatedly using eden-clocks. DoNaLD is used to draw the objects in the simulation, Scout is used to capture mouse clicks and keyboard presses whilst Angel is used to create a GUI menu.

3 Creating The Model

The model displays a side on view of a table tennis ball floating above the left side of a table and bat on the other side of the table. In a separate window is a list of sliders for selecting the initial velocity of the table tennis ball in the x and y directions as well as the level of topspin or backspin on the ball. Then by clicking on the "Run" button the simulation starts and the ball moves across the screen reflecting the initial velocity and spin determined by the user. If the ball reaches the bat its bounce is also simulated to show how the spin on the ball changes the direction of the ball when it bounces off the bat. If the ball does not reach the bat then it can be moved sideways using the "a" and "d" keys so that the ball reaches the bat before it hits the table.

The amount of spin on the ball can be set to beyond the levels capable of being produced by table tennis players which leads to interesting results. For example setting the backspin too high causes the ball to lift upwards very quickly due to the massive amount of lift generated by the backspin. Conversely an excessive amount of topspin will cause the ball to drop very quickly.

The settings window was created using the Angel notation Harfield (2006) which allows for easy generation of GUIs. This window is not required to run the model but can be used to quickly and easily adjust some of the model parameters.

3.1 Ball Physics

The main forces acting on the ball throughout its trajectory are gravity and the Magnus effect. The force of gravity constantly exerts a downward force throughout the balls trajectory whilst the Magnus effect is generated by the spin on the ball and causes lift. This lift can either be exerted upwards or downwards, depending on what spin is on the ball.

The Magnus effect is estimated using the following formula where rho is the density of air, radius is the

radius of the ball and omega is the spin in radians per second.

$$C = \pi \times \rho \times r^2 \times \omega \quad (1)$$

The velocity of the ball at time t is estimated in the horizontal and vertical directions using the following two formulas where v1X and v1Y are the start velocities of the ball in the x and y directions at t=0, m is the mass of the ball and g is the acceleration due to gravity.

$$v2X = \frac{1}{1 - \left(\frac{t}{m}\right)^2} \times C^2 \quad (2)$$

$$v1X + C \times v1Y \times \frac{t}{m} - C \times g \times \frac{t^2}{m} \quad (3)$$

$$v2Y = v1Y + \frac{t}{m} \times C \times v2X - g \times t \quad (4)$$

The position of the ball in terms of x and y coordinates can then be calculated at time t using the following two formulas where s1X and s1Y are the start coordinates of the ball at t=0.

$$s2X = s1X + v1X \times t + \frac{1}{2} \times \left(\frac{C}{m} \times v2Y\right) \times t^2 \quad (5)$$

$$s2Y = s1Y + v1Y \times t + \frac{1}{2} \times \frac{(C \times v2X) - m \times g}{m} \times t^2 \quad (6)$$

The trajectory of the ball after a bounce was estimated using the following formula. This produced a reasonable estimate of the actual bounce of the ball when it had topspin, backspin and no spin. This bounce formula is used when the ball bounces on the table and also on the bat.

Calculating the velocity of the ball after a bounce required calculation of the linear collision impulse. This was calculated for the vertical and horizontal directions using the two formulas below. fCr is the coefficient of restitution which ranges from 0 to 1 with 0 being no bounce and 1 being an elastic collision where no energy is lost in the collision. dx and dy are the normalised collision normals in the horizontal and vertical directions. vrnX and vrnY are the relative normal velocities in the horizontal and vertical directions respectively.

$$jX = \frac{-(1 + fCr) \times vrnX}{dx^2 \times \frac{1}{m}} \quad (7)$$

$$jY = \frac{-(1 + fCr) \times vrnY}{dy^2 \times \frac{1}{m}} \quad (8)$$

Using the linear collision impulse values the new velocity of the ball can then be calculated using these final two equations.

$$vX = \frac{jX \times dx}{m} \quad (9)$$

$$vY = \frac{jY \times dy}{m} \quad (10)$$

4 Analysis and Conclusions

4.1 Utility of Model

The model shows how the trajectory of a table tennis ball is affected by the the ball's initial velocity and amount of topspin and backspin on the ball. Furthermore the effect of topspin and backspin on the ball when bouncing off a table tennis bat is demonstrated.

The model although aimed at simulating a table tennis ball can be used to simulate various other circular shaped objects by changing the mass and radius of the ball. Both of these parameters will affect how much spin will alter the ball's trajectory. Even further gravity and the density of the air the ball is travelling through can be altered so a simulation of how a table tennis ball travels on different planets could be done.

4.2 Limitations

Whilst the model does simulate the trajectory of a table tennis ball with topspin or backspin reasonably well it does not show the user the actual technique required to achieve topspin or backspin. A more interactive approach could be applied by using a Wii mote as an input device which would allow the user to perform a table tennis stroke which the model could then interpret and show the resulting ball trajectory. This could be built upon the existing model as the initial velocity and spin of the table tennis ball can be obtained from the Wii mote instead of using the sliders or manually entering values.

The model does not show the effects of side spin and combinations of topspin and backspin with side spin, both of which are also important in table tennis. Whilst topspin and backspin are predominantly used all types of spin should be simulated for completeness. This would require the addition of a birds eye view of the table as side spin could only be shown from the side as the ball getting larger or smaller depending on whether the ball is moving towards or

away from the screen. An even better example although much harder to implement using Eden and / or Cadence would be to do a 3D simulation.

Lastly the rubber on the bat in the model is inverted rubber. Having the option of changing the rubbers to short or long pimples or anti spin would improve the model as the effects of these rubbers can be demonstrated.

4.2.1 EM Tools

Constructing the model provided a lot of hands on experience with the various definitive notations and using tkeden. Whilst these tools are powerful they are not as easy to use as they could be. The different syntaxes used in each notation along with inconsistent naming of observables when accessing them from Eden complicate the creation of a model for those not well versed in each notation. 0523756 (2009) shares these opinions and proposes using one consistent notation to solve these problems.

Another issue which came up during the construction of the model was a lack of any kind of IDE in which the model script could be viewed. There were times when the model would not work as expected when modifying it and commonly this was down to an observable being spelt wrong. As variables do not have to be declared in Eden this does not usually result in an error and instead manifests as an unexpected side effect in the model. Having a simple IDE which could highlight occurrences of any undefined variables being used in the model would be very useful and speed up debugging.

4.3 Conclusion

The model manages to demonstrate the effects of topspin and backspin in table tennis both on the ball's trajectory across the table and upon contact with a table tennis bat. Furthermore through experimentation with the model parameters, simulations of more than just a table tennis ball can be created. This model also sets a framework upon which others wanting to create a physical simulation could use as a starting point.

Acknowledgements

To Meurig Beynon for organising the CS405 Introduction to Empirical Modelling module.

References

0523756. What empirical modelling tools should look like, 2009. URL <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/05/tools.pdf>.

Meurig Beynon. Donald - a definitive notation for line drawings, 2006a. URL <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/software/eden/langref/>.

Meurig Beynon. Eden language reference guide, 2006b. URL <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/notations/donald/>.

Meurig Beynon. Scout - a definitive notation for screen layout, 2006c. URL <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/notations/scout/>.

Meurig Beynon. Introduction: Notes on definitive notations, 2010. URL <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/notations/intro/>.

Ian Bridge. Cruise control, 1991. URL <http://empublic.dcs.warwick.ac.uk/projects/cruisecontrolBridge1991/>.

Robert Denn, editor. *Physics for Game Developers*. O'Reilly, 2002.

Antony Harfield. Gel: A definition-based graphical environment language for eden, 2006. URL <http://empublic.dcs.warwick.ac.uk/projects/gelHarfield2006/doc/gelguide.html>.

Andy MacDonald. Room viewer 3d, 1998. URL <http://empublic.dcs.warwick.ac.uk/projects/room3dMacDonald1998/>.