Bowling a cricket ball: an EM simulation

0705678

Abstract

This paper presents and discusses an attempt to model the physical mechanics behind the act of bowling a cricket ball using Empirical Modelling techniques and its use as a tool for developing bowling tactics and technique. The scenario will be described and discussed along with an explanation of the parameters and dependencies involved in calculating the trajectories of the ball. Bowling a cricket ball is actually a complex physics calculation - much of the skill of bowling well comes from understanding and mastering the rules that govern the movement of the ball. Being able to calculate the path of a ball is also important as an umpire, because many decisions are dependent on whether the ball is going to hit the stumps. The Hawkeye system currently used by professionals is used for this purpose. There are many factors that the movement of the ball is dependent on. Dependencies can include the effect of the height of the bowler, pace of bowling and texture of the pitch on the height of the bounce of the ball. A corresponding EM model that illustrates these dependencies, developed in the Eden tool, is introduced with a description of its implementation and use. Limitations of the current modelling approach are discussed - limitations such as complex environmental factors that may be beyond the scope of this paper to incorporate. Finally, the suitability and effectiveness of the Empirical Modelling approach to this problem will be assessed with conclusions.

1 Introduction

One of the great challenges that faces any fast bowler in professional cricket is the ability to consistently bowl balls that end up in the right place. As Australian fast bowler Mitchell Johnson could tell you, this is absolutely vital. Much of wicket-taking in Test match cricket relies not on one killer delivery, but rather on patiently building up pressure as the bowler pins the batsman down with accurate delivery after accurate delivery that ultimately frustrates them into playing a rash shot in a bid to score runs.

However, as an amateur cricketer myself, I am aware that it can be very difficult to know what your next delivery is going to do. The problem I am addressing in this paper is actually particularly pertinent to the amateur cricketer. Since an amateur can rarely achieve anything close to the pace of delivery achieved by Test players, it can be difficult to know how you should bowl. For example, a delivery pitched at a certain length, that would be an intimidating bouncer when bowled by a Test bowler, would be a limping long hop that begged to be dispatched for four when bowled by me.

This paper addresses how Empirical Modelling principles can be brought to bear to create a model that would allow an aspiring bowler to discover the effects of different factors on the flight of a cricket ball from release to reaching the batsman.

2 Previous Work

Newtonian physics is an area of EM that has attracted attention in the past. Previous work [1] found that Eden was not a sufficiently rich tool for creating an environment that would allow non-technical users to create their own mechanics scenarios, but for specific mechanics problems (such as the one I am addressing) it was found to be useful. However, this paper did not address the issue of animation, which is of central value to a cricket simulation. An earlier paper[2] more directly addresses the question of simulating the movement of a ball, albeit in the more general context of a free space in which a ball is bouncing. Whilst my scenario has a more immediate application, a precedent has been set showing the capabilities of Eden for animating a ball governed by Newtonian physics. The model perhaps closest to my purpose is a teaching tool for linear kinematics [3] which uses Eden to model the path of an object as it travels through the air from a fixed platform to the floor.

My final application will be a combination of the objectives of these last two. It simulates a flight path similar to that in the latter model, yet seeks to maintain the free interactivity of the *Newtonian Mechanics* paper, and also contains that models capacity for a bouncing ball (in a cricket scenario, what happens after the ball bounces is, if anything, more important than what happens before!

3 Bowling a cricket ball

The flight of a cricket ball is governed by Newtonian physics, but to make this model worthwhile, it is important to simulate a variety of dependencies, since it is the various forces and environmental conditions acting on a cricket ball that make bowling on such a difficult art to master. Some aspects of bowling, such as when a cricket ball will swing in the air, are not fully understood, and so there are limits that must be placed on the scope of the model. However, there are still multiple dependencies that can be covered.

3.1 Dependencies

The behaviour of a cricket ball is entirely dependent on the laws of physics and the values that fill the variables in the physical equations. EM is all about being able to experimentally discover the way in which the behaviour of the ball changes as those forces change.

3.1.1 Speed of release

It is very useful to know how fast you are bowling, but the speed of the ball is not constant over the period of a delivery, so this must be calculated. As well as being computationally more challenging to specify the average delivery speed, it is more in the spirit of EM to specify the speed of release and observe the effect on the speed of the delivery. The philosophy of EM would be undermined by allowing the user to explicitly control the behaviour by specifying the average speed – it is much better to allow the user to learn the effect of release speed on the average delivery speed experimentally. Pragmatically it also makes sense, since discovering the effects of different release speeds is more useful for the user seeking to improve their bowling, since they can discover in a given situation 'how hard they need to throw it'.

3.1.2 Bowler's height

The height of a bowler can have a significant impact on the shape of the flight of a delivery. A tall bowler releases the ball from a higher start position and this changes the angle that the ball hits the ground at – which can enable them to gain more bounce from a delivery bowled at a longer length. As with the release speed of the ball, rather than allowing the user to explicitly specify the height of release (i.e. how high their hand is in the air), it is better to allow them to enter their height (a value they are far more likely to know) and discover experimentally the effect their height has. The release height would be calculated by extrapolating from the height of the bowler according to standard human proportions.

3.1.3 Nature of the pitch

One of the factors that is the focus of much discussion amongst pundits before and during the course of a cricket match is that state of the pitch. The nature of the surface has a significant effect on the amount the ball bounces off the surface and the amount of pace it loses when it hits the ground. For example, a dry, hard pitch would cause more bounce and would preserve more pace than a soft pitch. This governs a value called the Coefficient of Restitution (COR) – this value describes what proportion of the vertical speed is retained in the opposite direction when the ball bounces [4].

3.2.1 Targeted pitching point

How a ball behaves is heavily dependent on where the bowler aims it – a ball that bounces after a shorter distance is at a steeper angle, and will thus bounce higher, conversely a ball that is pitched longer will not bounce as high by the time it reaches the batsman. As for release speed, it is in the spirit of experimentation to aim for a pitching distance and allow the dependencies in the model to actually determine where the ball will bounce. This will be a combination of the angle the ball is release at (due to the aimed length), gravity and the speed of release.

3.1.5 Gravity

Whilst a very obvious factor, gravity does have a significant impact on the flight of the ball, as without gravity, the ball's flight would not decay and a delivery would have a very different shape. This said, gravity is more or less constant all over the world, yet I have included it as a changeable value the model so that a user could toy with the possibilities (such as what it would be like bowling on the moon!). It can also serve an instructive purpose as it can help the user understand how the various forces affect the movement of the ball by seeing how it would behave differently if gravity were to have no effect.

3.2 Observables

It would be fruitless to create a model with a world full of dependencies if there were no output. Interaction is only meaningful if there is some form of feedback. In the context of a cricket ball, there are outcomes that a bowler wants to be able to understand so that they can learn how changing their bowling affects the end result. Thus there are certain things we want to observe in order to understand the behaviour of the ball as a result of the dependencies just described.

3.2.1 Average speed of the delivery

As described in section 3.1.1, we want to be able to know how fast you are bowling, this is one of the most important things to know. This is determined by the average speed of the ball over the length of the pitch. Having specified the release speed of the ball (the factor that the bowler would actually be able to control), we can then observe the average delivery speed.

3.2.2 Final height of delivery

Possibly the joint most important thing to know about a delivery is at what height the ball will reach a batsman. The speed at which the ball reaches the batsman governs how much time they have to play a shot, but the height at which the ball arrives determines what shot they will play. A ball that does not reach the batsman at the desired height may change from a dangerous potential wicket-taking delivery into one that sits up and begs to be hit to the boundary. Thus it is very important to be able to observe the effect of different factors on the final height. In actual fact, this would be a valuable observable for a learning batsman as well, as it would give them more of a sense of what a given ball would be likely to do – which would provide knowledge they could draw on for a real match situation.

3.2.3 Real time sense of the shape of the delivery

A very useful thing to understand is what a delivery will look like in reality. It is one thing to watch a Test bowler on the television, but it is very difficult to get a sense of how fast they are bowling (in actual fact, when seen from the side rather than the front perspective of the television, you realise it is very fast indeed!). This is a useful observable for both bowlers and batsman as it gives a general view of the complete behaviour of the ball and shows you what a delivery would be like in reality. Observing a real time animation of the delivery allows you to see all the dependencies working together.

3.2.4 Actual pitching length of delivery

Due to the effects of gravity, the flight of a ball will decay, and so the point at which the bowler actually aims will not be the point at which it actually hits the ground. Thus the bowler can benefit from learning how to modify the initial speed of their delivery to achieve the desired length.

4 The model

To construct this model requires the use of appropriate EM tools and suitable equations capable of calculating the physics required to implement the dependencies described above.

I chose to use Eden as the tool for writing this model, largely down to my familiarity with the language and the availability of tools to write it on my own machine, but also from a modelling perspective because it supports the spirit of EM by allowing the modeller to interact with the parameters of the system.

4.1 Fixed values

Whilst it is in the spirit of EM to be able to interact with the model freely, there are some parameters that should remain fixed. In cricket there are specifications of the size of various objects, and it would not enrich the modelling experience to be able to tamper with these.

For the drawing of the interface, the lengths have all been scaled to pixel values, but are all proportional to each other, thus producing an accurate scale model of reality. The pitch is officially 22 yards along (20.12m) [5], the height of the stumps is required to be 28in (71.1cm) with the bails not projecting more than ½ inches (1.27cm) above the tops [6]. For the sake of my model I have chosen to use 72cm as the value. There is some license allowed for the size of the cricket ball, however, the MCC specifies that it should not exceed 9in (22.9cm) in circumference [7], this works out at a diameter of approx. 7cm. I have used a value close to this for the model, although the exact size of the ball does not overly affect the accuracy of the model, since it does not affect the flight of the ball and measurements are taken from the centre of the ball (using the traditional physics simplification of treating it as a point mass). There are no specific rules on the 'bounciness' of the cricket ball, unlike in sports such as Tennis. However, a guideline value is that when bounced on a sheet of steel, the COR is judged to be 0.58 [4]. Whilst this is a good initial value, it only applies to a new cricket ball off a hard surface – the user must be allowed to experiment with this value to see the effects of an older ball and softer surface.

4.2 Calculating observables

All of the observables and dependencies have values that are determined by calculation. Each of the specified dependency variables are controlled by the user interface, and it is the values from this that are assigned to the variables in the model. Trigonometry and the equations of motion then convert these values into the model's behaviour. Animation is provided by moving the circle representing the ball – the equations of motion are recalculated every 100th of a second to determine how far the ball should have travelled in that time.

It is an established rule in physics that the horizontal and vertical movement of an object are independent. Thus they can be calculated independently.

4.2.1 Horizontal motion

To calculate the horizontal motion, you need the horizontal component of the velocity of the ball. This is done by using trigonometry:

$$InitialHorizontalVelocity\ (u) = \cos(\theta) * SpeedofRelease$$

θ in both this section and the following indicates the angle of incidence of the ball with the ground:

$$\theta = \tan^{-1}\left(\frac{ballreleaseheight}{targetedlengthofdelivery}\right)$$

The distance the ball moves in 10ms is then calculated using the equations of motion:

$$FinalVelocity\ (v) = u + (acceleration * 0.01)$$

N.B. Horizontal acceleration is actually 0ms^{-2} but is included for completeness.

$$HorizontalDisplacement = \frac{(u+v) * 0.01}{2}$$

4.2.2 Vertical motion

Likewise the vertical motion is calculated by finding the vertical component of the velocity and using the equations:

$$InitialVerticalVelocity\ (u) = \sin(\theta) * SpeedofRelease$$

It should be noted that for the purposes of calculation, this is multiplied by -1 as the ball is being aimed down – therefore its height is reducing.

The distance of movement is likewise calculated using the equations of motion:

$$FinalVelocity(v) = u + (acceleration * 0.01)$$

N.B. In this case the vertical acceleration is governed by the strength of gravity, which for interest's sake is changeable by the user, but starts at the correct value for the Earth at -9.8ms^{-2}.

$$VerticalDisplacement = \frac{(u+v) * 0.01}{2}$$

4.2.3 Converting scales

It should be noted that whilst the values in the interface are controlled by the user according to their most familiar units (e.g. bowling speed is in mph), however for the equations of motion and for displaying on the interface, they need to be converted into the correct units. The equation for converting mph into ms^{-1} is:

$$speedinms = \frac{speedinmph * 1.609344 * 1000}{3600}$$

Rearrange for the reverse process.

4.3 GUI

The GUI is made using a combination of DoNaLD and Scout. Scout controls the layout of the interface, positioning the DoNaLD windows – one containing a line drawing of a cricket pitch setup from the side to allow the animation to be viewed, and the others being used for buttons to start and reset the animation and to adjust the values of the dependency variables.

Each button click is handled by a dedicated procedure that carries out the relevant operation.

5 Development process

The model development process was performed iteratively. A static, non-interactive environment with the physical features of the cricket pitch was created in Scout & DoNaLD. The main difficulty to overcome in this regard was correctly creating and positioning the window and devising a good scale of pixels-to-metres. The second challenge after this was to create real time animation. This was achieved after wrestling with the edenclocks functionality. At this stage, the ball was animated but was not behaving according to any of the laws of physics as previously described. Therefore the next stage in the process was to implement the physical laws. It was designed so that the values in the equations would be updated every 1/100th of a second. By the end of this stage the model was behaving in the way it was supposed to, but lacked any ability to interact with it in a user-friendly manner. The present (and final) model contains buttons that control when everything is started.

6 Analysis

6.1 Suitability of EM

For an application like this, the development of a tool for learning about the behaviour of a physical object, an approach that allows the user to have a constructivist stance is most appropriate. The idea is not for the user to learn by reading a set of rules that describe the behaviour of the cricket ball, but rather to be able to create that set of rules themselves – learning by doing, not simply by ingesting. In this sense, the EM approach, implemented in Eden suits the situation perfectly.

The value of Eden is not quite as clear cut in a scenario like this where there are defined limits to the time period of the computation, i.e. the ball starts and then stops, and the parameters are unlikely to be changed in the middle of the process. In this sense a conventional programming language would serve just as well. However, Eden is excellently suited for creating a model that genuinely fits with the ethos of EM. Rather than specifying behaviours as you would in a conventional language, it encourages the programmer to create a set of underlying principles and dependencies from which behaviour can be observed. In a sense, the program has no more of an idea than the user what the outcome will be until it happens. It is a journey of discovery, and the Eden family of languages are very helpful in encouraging this practice.

6.2 Success of model

Without carrying out rigorous numerical analysis of the results produced by the model by comparing them to the results yielded by live cricketers performing to the same parameters, the model appears to be quite a fair representation of reality.

The model is certainly perfectly functional – all the relevant parameters can be changed by the user and the interface outputs the desired observables. The animation of the cricket ball operates in the expected way (although there is a slight eccentricity whereby at certain finishing heights, the ball disappears from the edge of the screen when it reaches the wicket, however this does not invalidate the functionality of the model).

This said, an intuitive look at the model shows a surprising level of decay in the flight of the ball under gravity. The actual pitching length of the ball is often shorter than targeted, and the ball appears to bounce lower than I would have expected. However, by comparing some values it actually shows my own expectations are in error. For example, I was surprised that balls that I considered to be pitched short did not seem to be bouncing to full bouncer height (chest height or over), but some brief research on the internet shows that a length that is general considered short in cricket is between 7 and 11 yards [8]. Putting 9m into the model as a targeted length (approx 10 yards) yields a finishing height of 1.4m, which is just shy of 5ft. This actually turns out to be a very reasonable result. The benefits of an EM approach are illustrated very effectively here, because my intuitive notion of how the ball should behave actually turns out to be false. This is possibly caused by the lack of an accurate sense of distance you get when watching cricket on television, and translating this to a side view of the model resulted in my inaccurate expectations. Thus there is a clear benefit to being able to experiment with the values of the system to discover the results, because it enables me to discover what is actually true rather than simply relying on my own (unreliable) intuition.

6.3 Improvements and extensions

As with any attempt to model a complex environment such as the physical world, it is inevitable that the construal is not an exact mirror image of reality. The design of the model makes many of the classic simplifications involved in physics problems (especially of an A level complexity). There is no allowance for air resistance, there is no friction effect on the ball as it hits the ground, no allowance for other eccentricities unique to cricket.

There are many complex forces that are at work on a cricket ball, and not all of these are actually understood. However, there are some aspects that can be developed further. For example, the model requires the user to choose the value they want for the COR, whereas in actual fact it would be more in the spirit of EM for them to choose the age of the ball and the hardness of the surface and discover the effect this has on the COR. Furthermore, the roughness of a pitch should have some effect on the horizontal motion of the ball as friction should slow the ball down.

Another complexity that is not accounted for is the seam position of the ball. Cricket balls have a large seem that runs around the equator and holds the two halves of the leather exterior together. This seam is very important when bowling. For example, spin bowlers rely on the seam to grip the pitch and thus allow the rotations they put on the ball to convert into a change of direction of the movement of the ball. Seam bowlers (who bowl fast and straight) keep the seam straight because when the ball lands the seam has an elastic effect that provides some extra bounce that can surprise a batsman.

The most significant simplification, and the most obvious field of extension is that the model is 2-dimensional. The assumption is that the ball travels from directly over one set of stumps to the other stumps in a completely straight line. However, not only is this not the case because the bowler stands to one side of the stumps to bowl, but also different styles of bowling do not even travel in straight lines. For example, spin bowling, as just mentioned, is entirely based on the idea of getting a ball to change direction in a manner that challenges the batsman. Thus to simulate a side view that simply assume the ball is travelling in a straight line will not accurately capture the movement of the ball towards the stumps, as the speed in that direction will decrease as the ball turns away from the straight line. To a lesser extent swing bowling has the same problem. This is a fast bowling form that relies on shining one side of the ball so that the air travels over that side more quickly so that at high enough speeds the ball swerves visibly in the air before it bounces. This is not captured by a side view that assumes that the ball moves in a straight line.

This leads to the realisation that the simple equations used to govern the motion of the ball from the side view are not actually completely adequate because they take no account of the side-to-side motion of the ball. Furthermore, it leads to an obvious area of extension work. This would be to simulate a top-down view that would allow the user to specify parameters that govern the side-to-side motion of the ball. This could include the number of revolutions per minute on a spin ball, or the disparity of shininess on a ball that is being swung. However, this would be a very large step up in terms of complexity and actually raises the challenge of whether it would even be possible. This is because the physics of what governs swing bowling are actually not fully known.

Physics can account for spin bowling, and this would not be that difficult to make allowances for, although it would be difficult to quantify the effects of a more pronounced seam or a rough, high-friction pitch. However, swing bowling remains something of a mystery. It is acknowledged that shining one side of the ball more affects the amount of swing, but this is not a sufficient explanation. Another factor that seems to have an effect is whether the weather is cloudy or not. Cloudy weather seems to aid swing, but not always. Thus in actual fact this would be a very difficult model to implement because the rules governing the behaviour are not yet fully understood. It would be possible to implement a simplification in the same way that the existing model is a simplification, but it would not be possible to accurately take account of the simple fact that some days the conditions are perfect for swing bowling and the ball simply does not bend.

7 Conclusion

In conclusion, this exercise has demonstrated the value of EM techniques as a means of learning about the physical world in general, but particularly in the context of a cricket ball. The behaviour of a cricket ball is extremely complex and not fully understood, but it is still possible to construct a model that is a fair reflection of reality and allows a user to experiment with several variables, discover the effects they have and even challenge and correct previously incorrect intuitions they may have held to. This is an effective way to learn and would provide them with valuable knowledge that they can apply to their own bowling as they develop a greater understanding of the effects of their techniques and tactics before having to put them into action.

Suggested Weighting

Model:Paper 40:60

Acknowledgements

Some people are just so wonderful.

Notes on running the model

The model was developed in tkeden version 1.67, and can be easily run by loading the Scout file into the Eden interpreter and running it as Scout code.

References

[1] Anon. (2008). An educational system for modelling mechanics problems. *The Fourth Warwick Empirical Modelling Bulletin.*

[2] Anon. (2006). An Educational Model for Newtonian Mechanics. *The Second Warwick Empirical Modelling Bulletin.*

[3] Anon. (2008). Using Empirical Modelling to Demonstrate Linear Kinematics. *The Fourth Warwick Empirical Modelling Bulletin.*

[4] Cross, R. (2005). *The Physics of Cricket.* [WWW] University of Sydney. Available from: http://www.physics.usyd.edu.au/~cross/cricket.html [Accessed 31/01/2011].

[5] Marylebone Cricket Club (2010). *Law 7 (The pitch).* [WWW] Available from: http://www.lords.org/laws-and-spirit/laws-of-cricket/laws/law-7-the-pitch,33,AR.html [Accessed 31/01/2011].

[6] Marylebone Cricket Club (2010). *Law 8 (The wickets).* [WWW] Available from: http://www.lords.org/laws-and-spirit/laws-of-cricket/laws/law-8-the-wickets,34,AR.html [Accessed 31/01/2011].

[7] Marylebone Cricket Club (2010). *Law 5 (The ball).* [WWW] Available from: http://www.lords.org/laws-and-spirit/laws-of-cricket/laws/law-5-the-ball,31,AR.html [Accessed 31/01/2011].

[8] Hinchliffe, D (2009). *Classic bowling dismissals: Seam.* [WWW] Available from: http://www.pitchvision.com/classic-bowling-dismissals-seam [Accessed 01/02/2011].