# Implementation of Umpire's Call for LBW Further Makes Cricket a Batsman's Game

**Author: Sushant S. Mahajan, PhD (Astronomy)**

Solar Physics Postdoctoral Fellow,

Institute for Astronomy, University of Hawaii, USA

non-official email: sushant.mahajan.itbhu@gmail.com

**Table of Contents**



**Disclaimer:**

I didn't get paid for this. I used my personal time on the new year's weekend to compute and write this up. So, no complaints if it is not meticulously detailed. This is not my job, but someone has to point these discrepancies and biases out.

**What this study is not meant for:**

This short study is not meant to appeal to or ridicule any decisions made by Umpires, TV Umpires or ICC officials in any circumstances on or off the field. This is a strictly statistical study and should be looked at accordingly.

**GOAL:**

This is a short statistical study of the inherent bias in the implementation of Umpire's call in DRS for LBW. I compute the probability distribution function(PDF) of the impact of a ball predicted by hawk eye ball tracking on the stumps and infer from it the probability of the ball hitting or missing the stumps in each of the scenarios: hitting, umpire's call and missing. My findings indicate that Umpire's call in the case of LBW review is not actually being used in a 50-50 type situation(as the public and many commentators generally believe) and that the current implementation of Umpire's call is excessively biased towards the batsmen. A potential fix discussed in section 5 is to change the null hypothesis from a batsman being "not out" to the Umpire's call.

**How to read this document:**

1. If you are not interested in the mathematical model, read section 0 then go straight to section 3
2. If you want the details, go through all sections
3. If you want the full monty, you can download this document as a live script (.mlx) and run it in MATLAB or convert it into a language you want

## 0. Initialization

```
clear all;
```

### 0.1 Dimensions of Wickets:

```
swidth = 0.2286;  % 9 inches in width
sheight = 0.71;   % 28 inches high
m2inch = 39.3701; % converts meters to inches
```

### 0.2 Dimension of Ball:

```
ballradius = 0.03645;  % based on 22.9 cm circumference for Men's cricket ball
ballwidth = 2*ballradius;
```

### 0.3 Assumptions

1. Let us assume a Gaussian (Normal) probability distribution for the predicted location of the **ball center** hitting the stumps
2. Let us further assume that half of the ball width is $3\sigma$ error so that there is only a 0.3% chance that the ball would deviate by a distance larger than its half width when it reaches near the stumps

Gaussian Distribution:

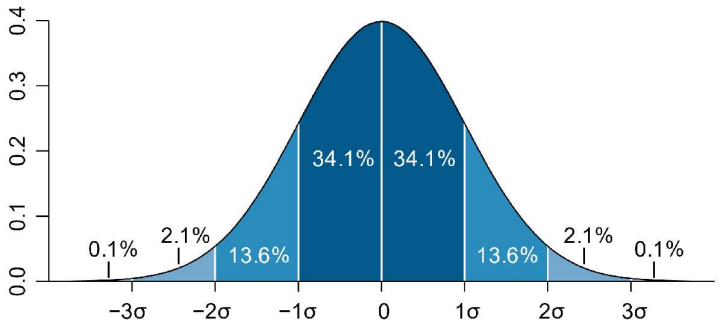



Image courtesy: M.W. Toews (uploaded to Wikipedia.org)

We will consider a two dimensional Gaussian distribution for the probability of the **ball center hitting stumps at a specific location**. In a Gaussian distribution, the probability of the ball being within the vicinity of the expected(predicted) location reduces as we look farther from the center of the predicted location. There is a 68.2% chance that the ball will be within $\pm\sigma$ of the expected location, whereas there is over a 95% chance that it will be within $\pm 2\sigma$ of the expected location. In the case of hawk-eye prediction of ball impact with stumps, the developers of hawk eye ball tracking claim an accuracy of 1 mm, where as we are going to be conservative and set $3\sigma =$ ball radius. We choose to do this because ICC also uses the ball radius as their error estimate to decide whether the Umpire's decision should be witheld. With this, there is only a 0.3% chance that the ball may deviate by more than the ball radius from the predicted location. Now, let us briefly go over the methodology that will be used to calculate the ball hitting/missing probability before we run tests with different scenarios.

## 1. Methodology

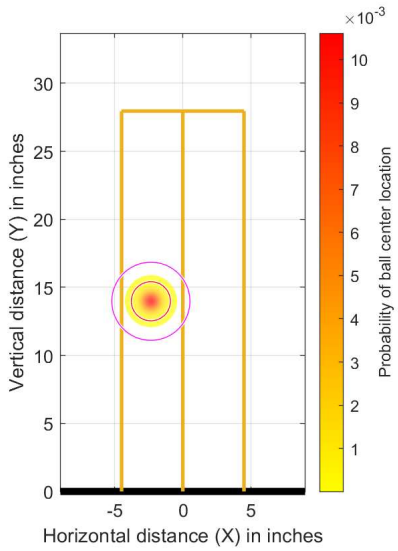
Let us first define the space we will be working in.

Horizontal span(x) of two times the width of wickets. Vertical span(y) of stump height + 2 x ballwidth. We will calculate the probability of ball hitting the stumps with a resolution of 0.5 mm. This should be enough for our task given that 0.5 mm is much smaller than the cricket ball.

```
X = -swidth:0.0005:swidth;           % resolution set to 1/2 mm
Y = 0:0.0005:sheight + 2*ballwidth;  % resolution set to 1/2 mm

posx = -swidth/2+1.5*ballradius;          % x coordinate of predicted ball location when hitting stumps
posy = sheight/2;                    % y coordinate of predicted ball location when hitting stumps

% Plot the wickets:
figure
color = get(gca,'colororder');
defcolor = color;
plot([-1 1]*swidth*m2inch,[0 0],'k','linewidth',4);
hold on
plot([-0.5 -0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2); % Outer edge of off stump
plot([0.5 0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);   % Outer edge of leg stump
plot([-0.5 0.5]*swidth*m2inch,[sheight sheight]*m2inch,'Color',color(3,:),'linewidth',2); % straight line on top. No time to include bails
plot([0 0],[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);  % Just a place holder for the middle of middle stump for aesthetics
xlabel('Horizontal distance (X) in inches')
ylabel('Vertical distance (Y) in inches')
axis('equal'); grid('on');
xlim([min(X) max(X)]*m2inch);
ylim([min(Y) max(Y)]*m2inch);
```

Construct a 2D Gaussian Probability Density Function for the predicted ball center location:

```
[x,y] = meshgrid(X,Y);
sigma = ballradius/3;    % standard deviation of Gaussian probability distribution function (PDF)

% Calculate the Gaussian PDF:
g = exp(-((x-posx).^2+(y-posy).^2)/(2*sigma^2));
% Normalize the Gaussian PDF so that total sum of probabilities is 1
g = g./nansum(g(:));
% Disregard the extremely less likely locations beyond 4sigma
g((x-posx).^2 + (y-posy).^2 > 16*sigma^2) = NaN;

% Overplot the PDF of ball on the stumps
h = pcolor(x*m2inch,y*m2inch,g*m2inch); set(h,'EdgeAlpha',0,'FaceAlpha',0.7); c = colorbar; colormap(flipud(autumn));
c.Label.String= 'Probability of ball center location';

% Mark the area in which ball center should be with 99.7% confidence
viscircles([posx posy]*m2inch,ballradius*m2inch,'Color','r','linewidth',0.5);
% Mark the area of the actual ball (not just the center) corresponding to
% above center locations
viscircles([posx posy]*m2inch,ballwidth*m2inch,'Color','m','linewidth',0.5);
```




```
fprintf('Probability of hitting wickets: %6.2f %%',100*nansum(g(x>-swidth/2-ballradius)))
```

```
Probability of hitting wickets:  99.97 %
```

```
fprintf(['Figure: The off side and leg side yellow lines represent the farthest edges of the stumps.\n' ...
    ' The middle yellow line goes through the middle of middle stump. The inner circle represents \n' ...
    'the size of the ball and the area within which 99.97%% of all possible ball centers would be \n' ...
    'found for a particular scenario. The colors inside the inner circle represent the probability \n' ...
    'of the ball center being there.  The outer circle shows the area of influence which would be \n' ...
    'covered if you place balls at all possible ball center locations in the inner circle.'])
```

```
Figure: The off side and leg side yellow lines represent the farthest edges of the stumps.
 The middle yellow line goes through the middle of middle stump. The inner circle represents
the size of the ball and the area within which 99.97% of all possible ball centers would be
found for a particular scenario. The colors inside the inner circle represent the probability
of the ball center being there.  The outer circle shows the area of influence which would be
covered if you place balls at all possible ball center locations in the inner circle.
```

Calculating the probability of ball hitting (off stump):

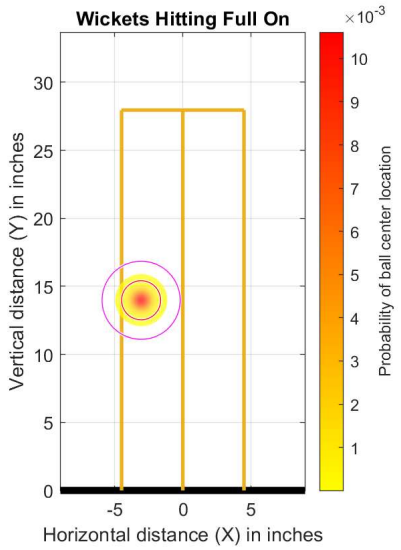
Sum of the probability at all ball centers in which case some part of the ball would lie inside or on the outer edge of the stump. In the case shown in above figure, wherever the ball center lies in the inner circle, the ball will touch the off-stump. So, there is a 99.7% probability that the ball will hit the off stump.

## 2. Testing Scenarios

### 2.1. Wickets hitting full on

```
posx = -swidth/2 + ballradius;        % x coordinate of predicted ball location when hitting stumps
posy = sheight/2;        % y coordinate of predicted ball location when hitting stumps

% Plot the wickets:
figure
color = get(gca,'colororder');
plot([-1 1]*swidth*m2inch,[0 0],'k','linewidth',4);
hold on
plot([-0.5 -0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2); % Outer edge of off stump
plot([0.5 0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);   % Outer edge of leg stump
plot([-0.5 0.5]*swidth*m2inch,[sheight sheight]*m2inch,'Color',color(3,:),'linewidth',2); % straight line on top. No time to include bails
plot([0 0],[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);  % Just a place holder for the middle of middle stump for aesthetics
xlabel('Horizontal distance (X) in inches')
ylabel('Vertical distance (Y) in inches')
axis('equal'); grid('on');
xlim([min(X) max(X)]*m2inch);
ylim([min(Y) max(Y)]*m2inch);
[x,y] = meshgrid(X,Y);
sigma = ballradius/3;    % standard deviation of Gaussian probability distribution function (PDF)

% Calculate the Gaussian PDF:
g = exp(-((x-posx).^2+(y-posy).^2)/(2*sigma^2));
% Normalize the Gaussian PDF so that total sum of probabilities is 1
g = g./nansum(g(:));
% Disregard the extremely less likely locations beyond 4sigma
g((x-posx).^2 + (y-posy).^2 > 16*sigma^2) = NaN;

% Overplot the PDF of ball on the stumps
h = pcolor(x*m2inch,y*m2inch,g*m2inch); set(h,'EdgeAlpha',0,'FaceAlpha',0.7); c = colorbar; colormap(flipud(autumn));
c.Label.String= 'Probability of ball center location';
```

```
% Mark the area in which ball center should be with 99.7% confidence
viscircles([posx posy]*m2inch,ballradius*m2inch,'Color','r','linewidth',0.5);
% Mark the area of the actual ball (not just the center) corresponding to
% above center locations
viscircles([posx posy]*m2inch,ballwidth*m2inch,'Color','m','linewidth',0.5);
title('Wickets Hitting Full On')
```

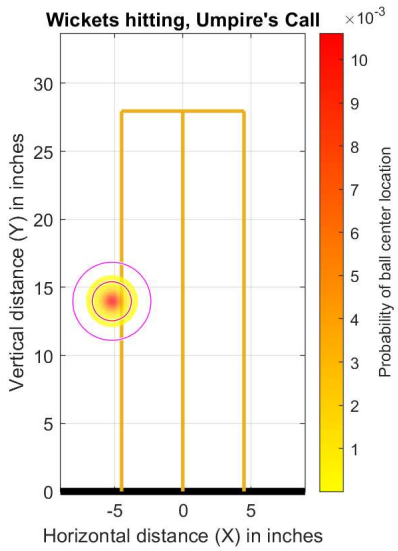



```
fprintf('Probability of hitting wickets: %6.2f %%',100*nansum(g(x>-swidth/2-ballradius)))
```

```
Probability of hitting wickets:  99.97 %
```

### 2.2. Wickets hitting, Umpire's Call

```
posx = -swidth/2 - ballradius/2;        % x coordinate of predicted ball location when hitting stumps
posy = sheight/2;        % y coordinate of predicted ball location when hitting stumps

% Plot the wickets:
figure
color = get(gca,'colororder');
plot([-1 1]*swidth*m2inch,[0 0],'k','linewidth',4);
hold on
plot([-0.5 -0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2); % Outer edge of off stump
plot([0.5 0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);   % Outer edge of leg stump
plot([-0.5 0.5]*swidth*m2inch,[sheight sheight]*m2inch,'Color',color(3,:),'linewidth',2); % straight line on top. No time to include bails
plot([0 0],[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);  % Just a place holder for the middle of middle stump for aesthetics
xlabel('Horizontal distance (X) in inches')
ylabel('Vertical distance (Y) in inches')
axis('equal'); grid('on');
xlim([min(X) max(X)]*m2inch);
ylim([min(Y) max(Y)]*m2inch);
[x,y] = meshgrid(X,Y);
sigma = ballradius/3;    % standard deviation of Gaussian probability distribution function (PDF)

% Calculate the Gaussian PDF:
g = exp(-((x-posx).^2+(y-posy).^2)/(2*sigma^2));
% Normalize the Gaussian PDF so that total sum of probabilities is 1
g = g./nansum(g(:));
% Disregard the extremely less likely locations beyond 4sigma
g((x-posx).^2 + (y-posy).^2 > 16*sigma^2) = NaN;

% Overplot the PDF of ball on the stumps
h = pcolor(x*m2inch,y*m2inch,g*m2inch); set(h,'EdgeAlpha',0,'FaceAlpha',0.7); c = colorbar; colormap(flipud(autumn));
c.Label.String= 'Probability of ball center location';

% Mark the area in which ball center should be with 99.7% confidence
viscircles([posx posy]*m2inch,ballradius*m2inch,'Color','r','linewidth',0.5);
% Mark the area of the actual ball (not just the center) corresponding to
% above center locations
viscircles([posx posy]*m2inch,ballwidth*m2inch,'Color','m','linewidth',0.5);

title('Wickets hitting, Umpire''s Call')
```

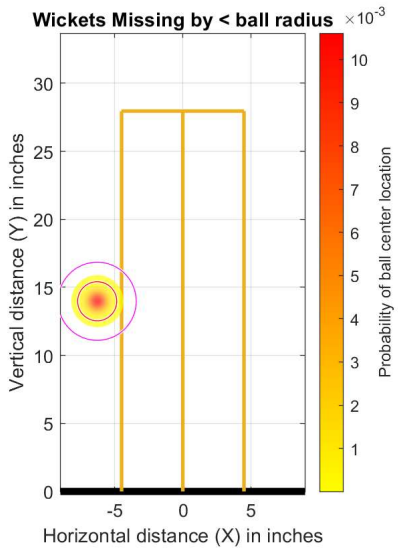



```
fprintf('Probability of hitting wickets: %6.2f %%',100*nansum(g(x>-swidth/2-ballradius)))
```

```
Probability of hitting wickets:  93.41 %
```

### 2.3. Wickets Missing by < ball radius

```
posx = -swidth/2 - (5/4)*ballradius;        % x coordinate of predicted ball location when hitting stumps
posy = sheight/2;        % y coordinate of predicted ball location when hitting stumps

% Plot the wickets:
figure
color = get(gca,'colororder');
plot([-1 1]*swidth*m2inch,[0 0],'k','linewidth',4);
hold on
plot([-0.5 -0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2); % Outer edge of off stump
plot([0.5 0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);   % Outer edge of leg stump
plot([-0.5 0.5]*swidth*m2inch,[sheight sheight]*m2inch,'Color',color(3,:),'linewidth',2); % straight line on top. No time to include bails
plot([0 0],[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);  % Just a place holder for the middle of middle stump for aesthetics
xlabel('Horizontal distance (X) in inches')
ylabel('Vertical distance (Y) in inches')
axis('equal'); grid('on');
xlim([min(X) max(X)]*m2inch);
ylim([min(Y) max(Y)]*m2inch);
[x,y] = meshgrid(X,Y);
sigma = ballradius/3;    % standard deviation of Gaussian probability distribution function (PDF)

% Calculate the Gaussian PDF:
g = exp(-((x-posx).^2+(y-posy).^2)/(2*sigma^2));
% Normalize the Gaussian PDF so that total sum of probabilities is 1
g = g./nansum(g(:));
% Disregard the extremely less likely locations beyond 4sigma
g((x-posx).^2 + (y-posy).^2 > 16*sigma^2) = NaN;

% Overplot the PDF of ball on the stumps
h = pcolor(x*m2inch,y*m2inch,g*m2inch); set(h,'EdgeAlpha',0,'FaceAlpha',0.7); c = colorbar; colormap(flipud(autumn));
c.Label.String= 'Probability of ball center location';

% Mark the area in which ball center should be with 99.7% confidence
viscircles([posx posy]*m2inch,ballradius*m2inch,'Color','r','linewidth',0.5);
% Mark the area of the actual ball (not just the center) corresponding to
% above center locations
viscircles([posx posy]*m2inch,ballwidth*m2inch,'Color','m','linewidth',0.5);

title('Wickets Missing by < ball radius')
```

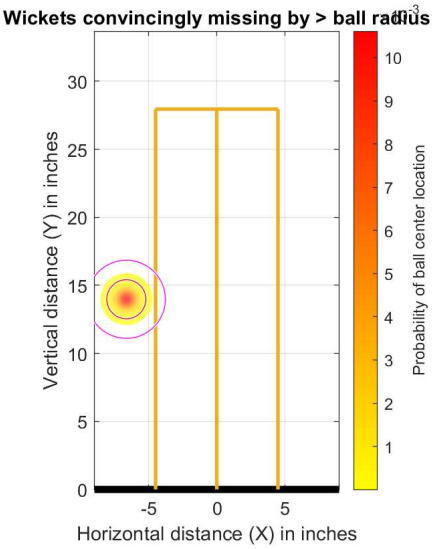



```
fprintf('Probability of hitting wickets: %6.2f %%',100*nansum(g(x>-swidth/2-ballradius)))
```

```
Probability of hitting wickets:  22.89 %
```

### 2.4. Wickets convincingly missing by > ball radius

```
posx = -swidth/2 - 1.5*ballradius ;        % x coordinate of predicted ball location when hitting stumps
posy = sheight/2;        % y coordinate of predicted ball location when hitting stumps

% Plot the wickets:
figure
color = get(gca,'colororder');
plot([-1 1]*swidth*m2inch,[0 0],'k','linewidth',4);
hold on
plot([-0.5 -0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2); % Outer edge of off stump
plot([0.5 0.5]*swidth*m2inch,[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);   % Outer edge of leg stump
plot([-0.5 0.5]*swidth*m2inch,[sheight sheight]*m2inch,'Color',color(3,:),'linewidth',2); % straight line on top. No time to include bails
plot([0 0],[0 sheight]*m2inch,'Color',color(3,:),'linewidth',2);  % Just a place holder for the middle of middle stump for aesthetics
xlabel('Horizontal distance (X) in inches')
ylabel('Vertical distance (Y) in inches')
axis('equal'); grid('on');
xlim([min(X) max(X)]*m2inch);
ylim([min(Y) max(Y)]*m2inch);
[x,y] = meshgrid(X,Y);
sigma = ballradius/3;    % standard deviation of Gaussian probability distribution function (PDF)

% Calculate the Gaussian PDF:
g = exp(-((x-posx).^2+(y-posy).^2)/(2*sigma^2));
% Normalize the Gaussian PDF so that total sum of probabilities is 1
g = g./nansum(g(:));
% Disregard the extremely less likely locations beyond 4sigma
g((x-posx).^2 + (y-posy).^2 > 16*sigma^2) = NaN;

% Overplot the PDF of ball on the stumps
h = pcolor(x*m2inch,y*m2inch,g*m2inch); set(h,'EdgeAlpha',0,'FaceAlpha',0.7); c = colorbar; colormap(flipud(autumn));
c.Label.String= 'Probability of ball center location';

% Mark the area in which ball center should be with 99.7% confidence
viscircles([posx posy]*m2inch,ballradius*m2inch,'Color','r','linewidth',0.5);
% Mark the area of the actual ball (not just the center) corresponding to
% above center locations
viscircles([posx posy]*m2inch,ballwidth*m2inch,'Color','m','linewidth',0.5);

title('Wickets convincingly missing by > ball radius')
```




```
fprintf('Probability of hitting wickets: %6.2f %%',100*nansum(g(x>-swidth/2-ballradius)))
```

```
Probability of hitting wickets:   6.77 %
```

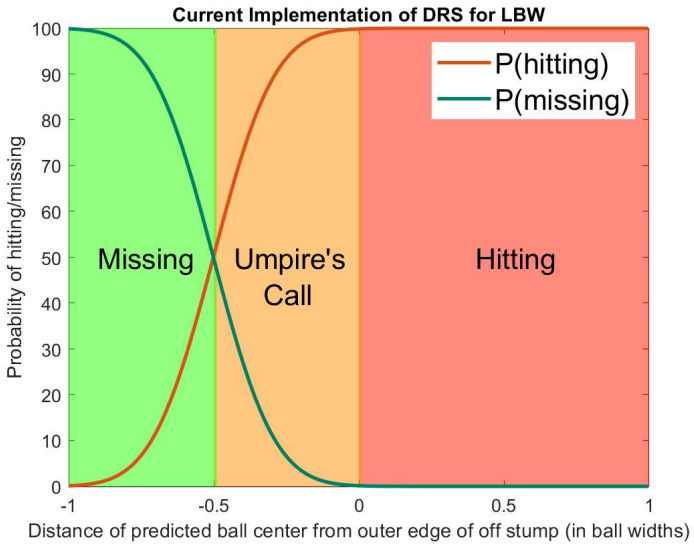
## 3. Current Implementation of DRS for LBW

Probability of hitting wickets as a function of distance from outer edge of off stump to predicted ball center

```
d = -ballwidth: 0.0005 :ballwidth;

for i=1:length(d)
    locx(i) = -swidth/2 + d(i);
    locy(i) = sheight/2;

    posx = locx(i);
    posy = locy(i);

    % Calculate the Gaussian PDF:
    g = exp(-((x-posx).^2+(y-posy).^2)/(2*sigma^2));
    % Normalize the Gaussian PDF so that total sum of probabilities is 1
    g = g./nansum(g(:));
    % Disregard the extremely less likely locations beyond 4sigma
    g((x-posx).^2 + (y-posy).^2 > 16*sigma^2) = NaN;

    prob(i) = 100*nansum(g(x>-swidth/2-ballradius));
end
```

```
figure

% Shade areas: hitting, umpire's call, missing
xx = -1:0.01:1;
yy = 0:0.02:100;
[xx,yy]=meshgrid(xx,yy);

hitting = ones(size(xx));
hitting(xx<0) = NaN;
umpire = ones(size(xx));
umpire(xx<-0.5) = NaN;
umpire(xx>0) = NaN;
missing = ones(size(xx));
missing(xx>-0.5) = NaN;

ax1 = axes;
% ax1.XTick = [];
% ax1.YTick = [];
h1=pcolor(ax1,xx,yy,hitting); set(h1,'EdgeAlpha',0,'FaceAlpha',0.5); color = hsv; colormap(ax1,color(5,:));
% ax1.Visible = 'off';

ax2 = axes;
ax2.XTick = [];
ax2.YTick = [];
h2=pcolor(ax2,xx,yy,umpire); set(h2,'EdgeAlpha',0,'FaceAlpha',0.5); color = hsv; colormap(ax2,color(25,:));
ax2.Visible = 'off';

ax3 = axes;
ax3.XTick = [];
ax3.YTick = [];
```

```
h3=pcolor(ax3,xx,yy,missing); set(h3,'EdgeAlpha',0,'FaceAlpha',0.5); color = hsv; colormap(ax3,color(80,:));
ax3.Visible = 'off';

ax4 = axes;
color = summer;
plot(ax4,d/ballwidth,prob,'Color',defcolor(2,:),'linewidth',2);
hold(ax4,'on')
plot(ax4,d/ballwidth,100-prob,'Color',color(1,:),'linewidth',2);
ax4.Visible = 'off';
ax4.XTick = [-2:0.5:2];

legend(ax4,{'P(hitting)','P(missing)'},'FontSize',16,'EdgeColor','w')
xlabel(ax1,'Distance of predicted ball center from outer edge of off stump (in ball widths)');
ylabel(ax1,'Probability of hitting/missing')

text(ax4,.4,50,'Hitting','FontSize',16)
text(ax4,-.43,50,'Umpire''s','FontSize',16)
text(ax4,-.33,42,'Call','FontSize',16)
text(ax4,-0.9,50,'Missing','FontSize',16)
title(ax1,'Current Implementation of DRS for LBW')
```




## 4. Discussion of Bias

This excercise busts one of the most common myths about Umpire's call. Umpire's Call is not being used for 50-50 calls. It is actually being used when the DRS system is over 50% certain that the ball is hitting the stumps. Whereas, even if the DRS system is 51% certain that the ball is missing the stumps, the Umpire's call of out will be overturned. This is completely biased towards batsmen. In the current system, there will be an overwhelming number of cases where Umpire's call is relied upon even when DRS is more than 90% certain that the ball is hitting the stumps (even with a large estimate of uncertainty used = half of ball width). If we want to keep this kind of an implementation, then the DRS system should be more than 90% certain that the ball is going to miss the stumps in order to overturn the Umpire's decision. As it stands, this is currently not the case. Even a 51% possibility of the ball missing the stumps can overturn the Umpire's call if it is out. Therein lies the bias against the bowling team.

In the permutations and combinations of possible scenarios, both the DRS and Umpire's Call have to go hand in hand to get a batsman out (see table below). So it is clearly advantage batsmen.

| Umpires Call | Probability (hitting) | Result |
|---|---|---|
| Out | 50-100% | Out |
| Out | 0 - 50% | Not Out |
| Not out | 50-100% | Not Out |
| Not out | 0-50% | Not Out |

## 5. Potential Fix

**In the current implementation of DRS for LBW, the null hypothesis is that the batsman is not out.** Therefore, even if DRS says there is over a 90% chance of the ball hitting the stumps (conservative estimate, the chance may be much higher if you believe the accuracy quoted by the ball tracking software) but the Umpire says not out, it stays not out. Whereas a 51% chance of the ball missing the stumps (when the predicted ball path is just grazing the stumps) is enough to overturn a decision of "out".

**The null hypothesis could be changed to the Umpire's call.** This would truly mean that the Umpire's call will come into play in situations close to 50-50. Moreover, the criteria need to be the same whether the ball is hitting or missing the stumps in order to void the null hypothesis and overturn the Umpire's call. This is not something new, the Umpire's call is already the null hypothesis when a catch(grassed or not) is reviewed and a soft signal is given. The same could be done for LBW instead of biasing it away from the bowling team.

The exact number or thresholds can be worked out by the ICC officials and people who actually are making money from this technology (again, I am using my personal time and resources to do this), but a potential solution will look something like this:

```
figure

% Shade areas: hitting, umpire's call, missing
xx = -1:0.01:1;
yy = 0:0.02:100;
[xx,yy]=meshgrid(xx,yy);

hitting = ones(size(xx));
hitting(xx<-0.35) = NaN;
umpire = ones(size(xx));
umpire(xx<-0.65) = NaN;
umpire(xx>-0.35) = NaN;
missing = ones(size(xx));
missing(xx>-0.65) = NaN;

ax1 = axes;
% ax1.XTick = [];
% ax1.YTick = [];
h1=pcolor(ax1,xx,yy,hitting); set(h1,'EdgeAlpha',0,'FaceAlpha',0.5); color = hsv; colormap(ax1,color(5,:));
% ax1.Visible = 'off';

ax2 = axes;
ax2.XTick = [];
ax2.YTick = [];
h2=pcolor(ax2,xx,yy,umpire); set(h2,'EdgeAlpha',0,'FaceAlpha',0.5); color = hsv; colormap(ax2,color(25,:));
ax2.Visible = 'off';

ax3 = axes;
ax3.XTick = [];
ax3.YTick = [];
h3=pcolor(ax3,xx,yy,missing); set(h3,'EdgeAlpha',0,'FaceAlpha',0.5); color = hsv; colormap(ax3,color(80,:));
ax3.Visible = 'off';

ax4 = axes;
color = summer;
plot(ax4,d/ballwidth,prob,'Color',defcolor(2,:),'linewidth',2);
hold(ax4,'on')
plot(ax4,d/ballwidth,100-prob,'Color',color(1,:),'linewidth',2);
ax4.Visible = 'off';
ax4.XTick = [-2:0.5:2];

legend(ax4,{'P(hitting)','P(missing)'},'FontSize',16,'EdgeColor','w')
xlabel(ax1,'Distance of predicted ball center from outer edge of off stump (in ball widths)');
ylabel(ax1,'Probability of hitting/missing')

text(ax4,.2,50,'Hitting','FontSize',14)
text(ax4,-.66,90,'Umpire''s','FontSize',14)
text(ax4,-.575,82,'Call','FontSize',14)
text(ax4,-.975,50,'Missing','FontSize',14)
title(ax1,'Potential Fix in DRS for LBW')
```