Experiment 22 - Monte Carlo Simulation

ELEC273

November 10, 2021

Abstract

Monte Carlo method has been widely used in simulating and dealing with statistic problems. This report describes the process of using Monte Carlo method to analyze the penalty kick problem in different situations, with or without a goalkeeper. There are six models, which are generated by MATLAB, to show the results of the problems. These six models are all based on uniform distribution or normal distribution. Some other techniques like tables, scatter plots will be used to analyze the results. Finally, there will be a conclusion to summarize the experiment and point out some suggestions.

Contents

1 Introduction

1.1 Background

The Monte Carlo method can be thought of as a collection of computational techniques for solving mathematical problems, which make fundamental use of random samples. It has been used to build some complex systems which can not be solved by using traditional techniques[1]. The penalty kick process will be discussed in this report by using the Monte Carlo method.

1.2 Objectives

- Explore the Monte Carlo method and use it to solve real-life problems.
- The ability to use MATLAB.
- Summarize some conclusions from this experiment

1.3 Experimental Precdure

The experiment can be solved by using MATLAB code following several steps, which are summarized based on the tasks.

- Plot a graph that shows the area of the shooting range and the area of the goal.
- A random coordinate sample of size N should be drawn based on the normal distribution or uniform distribution, which is used to display the shots for each turn.
- Plot all the points on the graph. The points inside the goal will be painted with a colour, while other points will be painted in a different colour.
- Count the points inside the point and calculate the probability.
- Repeat these steps R times. Then, calculate the average probability.

2 Results

2.1 Apparatus

MATLAB will be used to simulate results. There are totally six model to slove all the problems. All the models can be seen in the Appendices: Program Listings. The command and results will be displayed in the following sections.

2.2 Results for Part I: No goalkeeper Tests

2.2.1 Task 1

The probability of entering the goal can be considered as the area of the rectangle divided by the area of the circle.

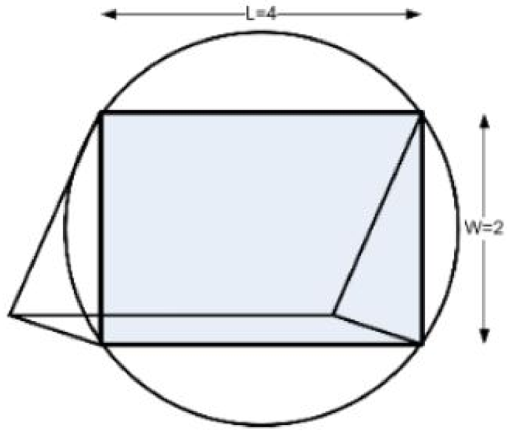

Figure 1: The diagram of the goal arrangement [2]

As Figure 1 shows, the radius of the circle is half of the diagonal (D) of rectangles. The length of the two sides has been given, which are 4 and 2. Therefore, we can calculate the radius (r).

$$r = \frac{1}{2} \times D = \frac{1}{2}\sqrt{W^2 + L^2} = \frac{1}{2}\sqrt{4^2 + 2^2} = \sqrt{5} \tag{1}$$

Then, the area of the circle (A_{circle}) and rectangle ($A_{rectangle}$) can be calculated:

$$A_{circle} = \pi \times r^2 = \pi \times (\sqrt{5})^2 = 5\pi \tag{2}$$

$$A_{rectangle} = W \times L = 4 \times 2 = 8 \tag{3}$$

Finally, we can obtain the probability (P) of entering the goal

$$P = \frac{A_{rectangle}}{A_{circle}} = \frac{8}{5\pi} \approx 0.5092 \tag{4}$$

The probability is about 0.5092.

2.2.2 Task 2

The code is shown in the Program Listings: Part I code: Model I.

The function is represented as Partone(N,R). This is an typical example of the result:

```
>> Partone(200,6);

poss =

    0.5050
```

2.2.3 Task 3

The code is also Model I. The order is shown below

```
>> Partone(1000,1);

poss =

    0.5085
```

The graph is shown below

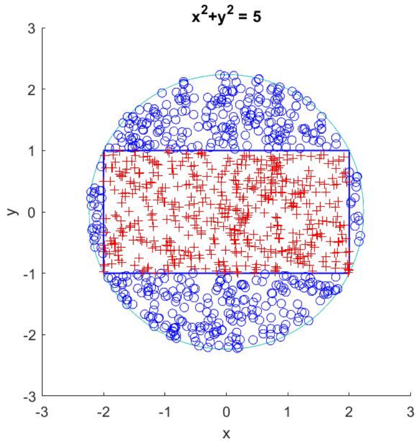

Figure 2: scatter plot

2.2.4 Task 4

In this task, change the value of N and collect different values of possibility. Table 1 shows all the statistics. The graph of the probability against N is shown in Figure 3. From the graph, when the value of N increases, the slope of the line will decrease. Moreover, the magnitude of probability becomes closer to the theoretical value.

Table 1: The results with different values of N when R = 5

N	P (Probability)
100	0.4851
1000	0.5005
10000	0.5076
100000	0.5088
Theoretical value	0.5092

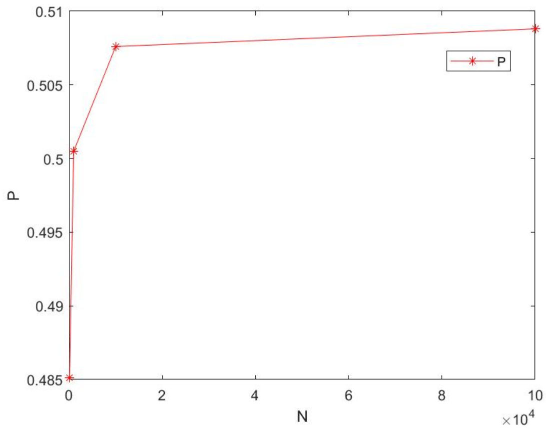

Figure 3: Task 4

2.2.5 Task 5

When N = 1000, the probability for different values of R is shown in Table 2.

Table 2: The results with different values of R when N = 1000

R	P (Probability)
5	0.5115
10	0.5127
15	0.5096
20	0.5089
Theoretical value	0.5092

The graph of probability against R can be observed in Figure 4. From Figure 4, with the increasing of R, the probability becomes closer to the theoretical value.

2.2.6 Task 6

From the two tasks, some conclusions can be listed.

- Results

  From Figure 3 and Table 1. When N increases from 100 to 100000, the result becomes more closer to the theoretical result.

  For Task 5, R increses from 5 to 20, the result will also become closer to the theoretical result(0.5092).

- Explaination
  - The probability will become closer to the theory when the value of N or R grows. That is understandable. When lots of experiments carry on and take the average value, the random error will decrease. Therefore, the probability will become closer to the theory. When the simualtion times is infinite, the experimwntal results can be considered as equal to the real result.

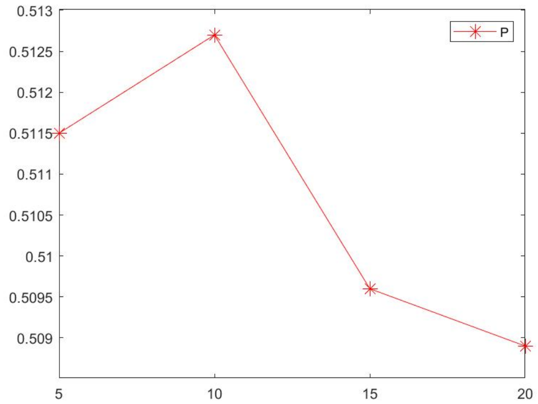

Figure 4: Task 5

- When the value of R increases, the precision can be improved while the accuracy will not. For N, on the contrary, the accuracy can be improved but precision will not improve.

2.2.7 Task 7

The code for this section can be found in the Appendices: Program Listings: Model II.

- **Repeat Task 2**

  There is an typical example shows how the code works on the matlab command window

```
>> N = 100;
>> R = 5;
>> Task7

 poss =

        0.7380
```

- **Repeat Task 3**

  Input the following orders on the command window. The result is shown in Figure 5.

```
>> N = 1000;
>> R = 1;
>> Task7

 poss =

        0.7490
```

- **Repeat Task 4**

  The probability for different values of R is shown below:

  The graph of probability against N is shown in Figure 6:

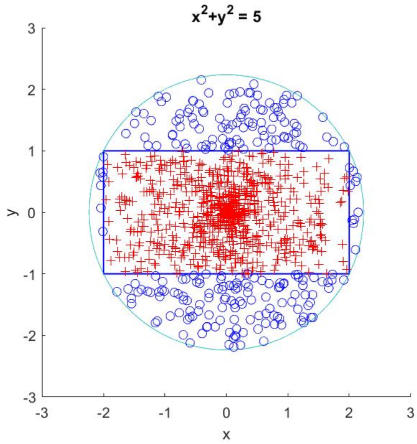

Figure 5: The scatter plot

Table 3: The results with different values of N when R = 5

N	P (Probability)
100	0.7300
1000	0.7424
10000	0.7353
100000	0.7321

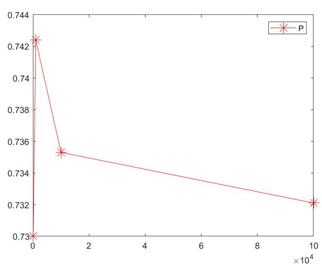

Figure 6: Repeat Task 4

- **Repeat Task 5**

  Table 4 shows the different probability with the different values of R, when N = 1000. The graph of probability against R can be seen in Figure 7.

Table 4: The results with different values of R when N = 5

R	P (Probability)
5	0.7320
10	0.7267
15	0.7368
20	0.7358

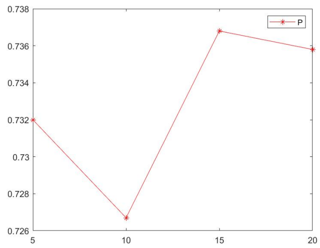

Figure 7: Repeat Task 4

- **Repeat Task 6**

  - **Results discussion**

    From Table 3 and Figure 6, it can be seen that the value of probability becomes closer to a certain value when N is increasing. The value is about 0.730.

    A similar conclusion can be obtained from Table 4 and Figure 7. All the values are fluctuating in a small region.

  - **Comparison**

    Compared with the uniform random number generator, the probability increases about 0.2 by using normal random number generator.

    That is because the normal random number generator accumulates values in the center of the circle. That means more balls shoot into the rectangle. Thus, the probability increases.

2.3 Results for Part II: With Goalkeeper Tests

2.3.1 Task 8

The code for this task can be seen in the Appendices: Program Listings: Model III.

- **Theory**

  As Figure 8 shows, there are five different possibilities of a goal keeper action. Each action has a probability of 0.2.

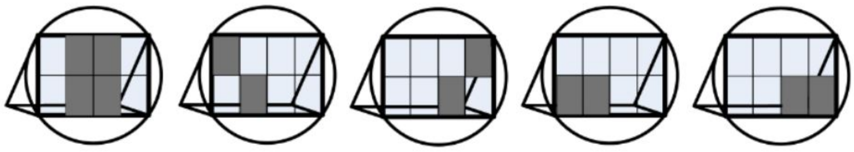

Figure 8: Possibilities of a goalkeeper action to a penalty shoot-out [2]

The calculation of probability can be divided into two conditions. The first condition is shown on the left side of Figure 8, in which 4 rectangles are covered.The other four options can be considered as another condition. Following the same idea in part I, the probability is equal to the area of the rectangle divided by the circle. Therefore, we have:

$$P = P_1 + P_2 = 0.2 \times 4 \times (\frac{\frac{3}{4}A_{rectangle}}{A_{circle}}) + 0.2 \times (\frac{\frac{1}{2}A_{rectangle}}{A_{circle}}) \tag{5}$$

The area of the rectangle is 8 while the area of the circle is 5π. Therefore, the theoretical value is about 0.3565.

- **Examples**

  A typical example is shown below:

```
>> N = 100;
>> R = 1;
>> Task8

 poss =

       0.3762
```

When N = 1000 and R = 1, the command and scatter plot are shown below:

```
>> N = 1000;
>> R = 1;
>> Task8

 poss =

       0.3516
```

- **Discussion**

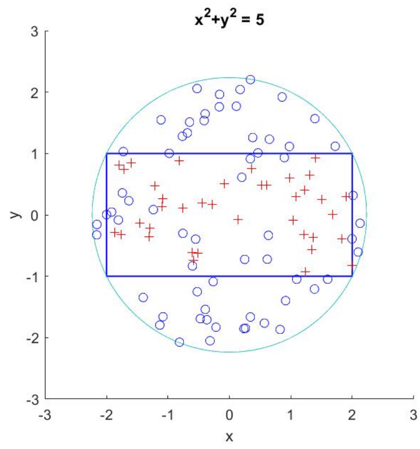

Figure 9: The scatter plot for N = 100

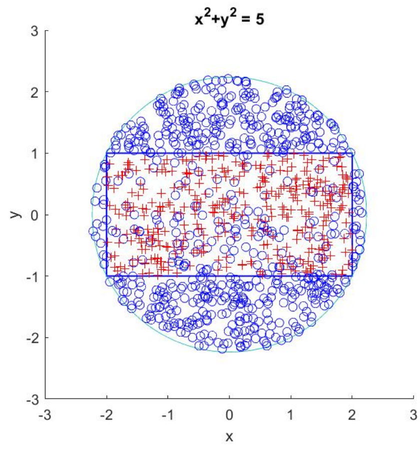

Figure 10: The scatter plot for N = 1000

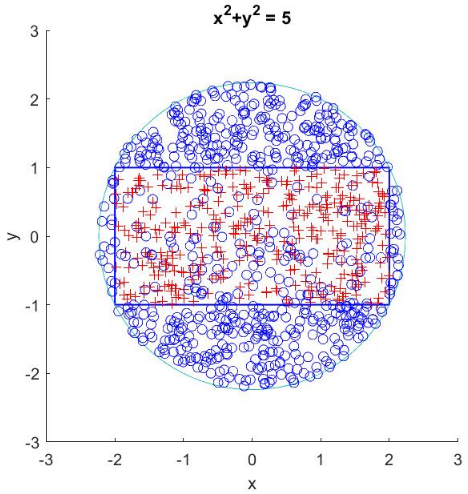

Figure 11: The scatter plot with goalkeeper when N = 1000 and R = 1

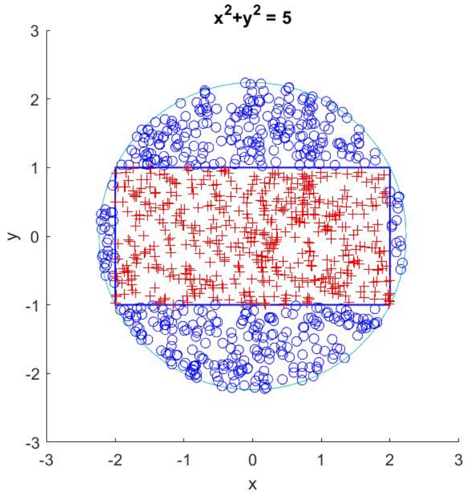

Figure 12: The scatter plot with no goalkeeper when N = 1000 and R = 1

From the theory and experiments, it can be clearly seen that the probability of scoring becomes smaller than the condition with no goalkeeper($0.5085 > 0.3516$). That is more closer to the real life situation.

2.3.2 Task 9

The code for this task is in the Appendices:Program Listings: Model IV.

In order to distinguish the scatter plot between different tasks. In task 9, the green points means getting score while red means no score.

- **Examples**

  Two examples are shown below. The scatter plots can be seen in Figure 13 and Figure 14.

```
>> N = 200;
>> R = 2;
>> Task9

 poss =

       0.4600
```

```
>> N = 1000;
>> R = 1;
>> Task9

 poss =

       0.4780
```

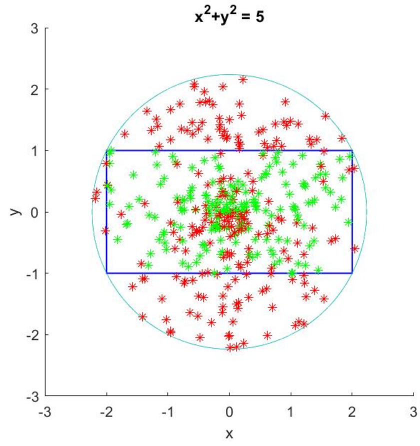

Figure 13: The scatter plot for N = 200 and R = 2

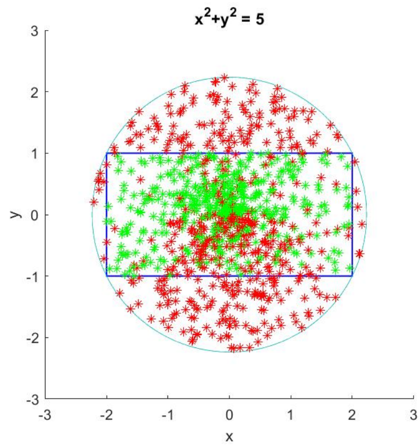

Figure 14: The scatter plot for N = 1000 and R = 1

- **Repeat Task 4 in Task 9**

  The results are shown in Table 5. The relationship between N and P is shown in Figure

Table 5: The results with different values of N when R = 5

N	P (Probability)
100	0.4640
1000	0.4676
10000	0.4596
100000	0.4637

15.

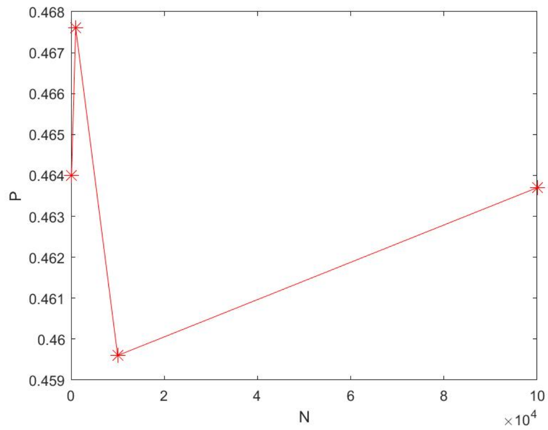

Figure 15: The relationship between N and P

- **Repeat Task 5 in Task 9**

  The results is shown in Table 6.

Table 6: The results with different values of R when N = 1000

R	P (Probability)
5	0.4616
10	0.4608
15	0.4698
20	0.4628

The relationship between R and P is shown in Figure 16.

From the two repeat tasks, it can be find that the probability is floating in a certain range.

- **Comparsion**

  Comparsion between Task 8 and Task 9

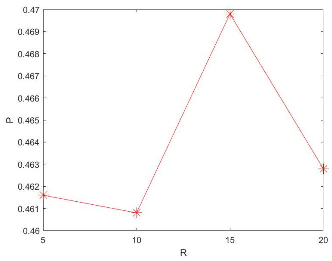

Figure 16: The relationship between N and P

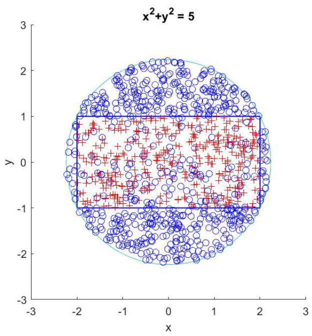

Figure 17: The scatter plot for uniform random distribution when N = 1000 and R = 1

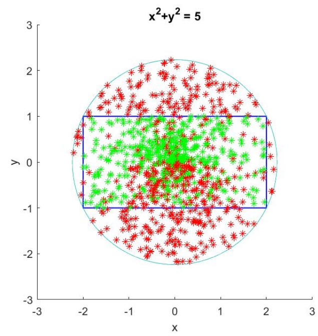

Figure 18: The scatter plot for Gussian random distribution when N = 1000 and R = 1

The probability for Figure 17 is 0.3516, while the probability for Figure 18 is 0.4780. The probability rises about 0.1264. That is also because the Gaussian random distribution accumulates the points in the centre. Therefore, the player can get more score.

Comparsion between Task 7 and Task 9

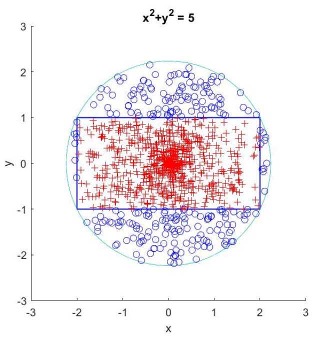

Figure 19: The scatter plot for Gussian random distribution when N = 1000, R = 1 and no goalkeeper

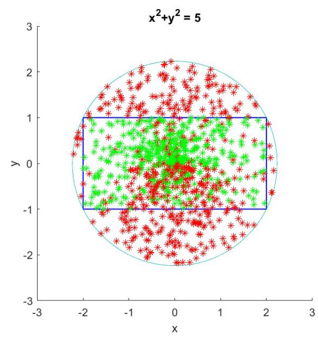

Figure 20: The scatter plot for Gussian random distribution with goalkeeper when N = 1000 and R = 1

The probability for Figure 19 is about 0.7490, while the probability for Figure 20 is 0.4780. The difference is about 0.271. It is easy to conclude that the goalkeeper keeps some shots away from the rectangle. Then, the probability decreases.

2.3.3 Task 10

The code for this task is shown in Appendices: Program Listings: Model V and Model VI. Model V is for uniform random generator and model VI is for Gaussian random number generator.

- **Experiment Results**

  uniform random number generator

  For N = 100 and N = 1000, the command and scatter plot can be seen below:

```
>> N = 100;
>> R = 1;
>> Task10

poss =

      0.3465
```

```
>> N = 1000;
>> R = 1;
>> Task10

poss =

      0.3806
```

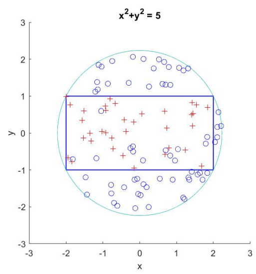

Figure 21: Scatter plot when N = 100 and R = 1

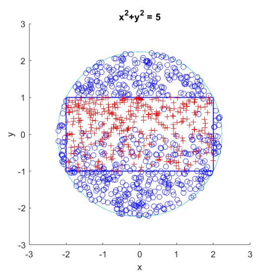

Figure 22: Scatter plot when N = 1000 and R = 1

Gaussian random number generator

For N = 100 and N = 1000, the command and scatter plot are shown below:

```
N = 100;
>> R = 1;
>> Taskten

 poss =

        0.5600
```

```
>> N = 1000;
>> R = 1;
>> Taskten

 poss =

        0.5550
```

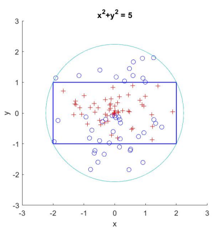

Figure 23: Scatter plot when N = 100 and R = 1

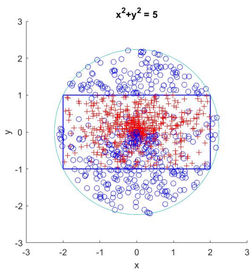

Figure 24: Scatter plot when N = 1000 and R = 1

- **Comparsion**

  From the two scatter plots, it is easy to find that the goalkeeper tends to jump to the two lower corners. Therefore, the probability that four rectangles are covered in Figure 8 will decrease. That results in the player can get scores more easily. What is more, the probability for Gaussian random generator is larger than uniform random generator (0.5600>0.3465;0.5550>0.3806) That also proves the conclusions about Gaussian random generator in the previous sections.

3 Review questions

1. Q1: In terms of what you've done in this experiment, comment on the advantages and disadvantages (or drawbacks) of the Monte Carlo experiment.

- Advantages
  - Flexibility: The simulation method can be used for stochastic process and deterministic process. It provides statistical sampling for numerical experiments using softwares.
  - Straightforward: The Monte Carlo simulation is quite simple as long as the converge can be assured by the theory. Monte Carlo simulation provides better physical visibility of a complex system rather than a set of complex equations. That makes non-mathematicians use it more easily.
  - Practicality: Monte Carlo method has been widely accepted and used in different fields, such as telecommunications, economy, chemistry.
- disadvantages
  - Limitation: Monte Carlo simulation was designed for studying the properties of the equilibrium system. It is not comprehensively acceptable for disequilibrium system, for example, the transient state. What is more, the use of Monte Carlo simulation requires a large number of sampling. It is not suitable for some experiments.
  - Accuracy: The results of Monte Carlo are only an approximation of the real value. During the simulation, it will produce random numbers. Neither uniform ramdom number or normal random number are not real uniform/normal random numbers.

2. Q2: Discuss the ways in which the above model could be made more accurate and realistic.

- Firstly,reducing the area of the circle is shown in Figure 2. The new area is shown in Figure 25. That is because the player can not kick the ball under the ground. Therefore, we can reduce some areas.

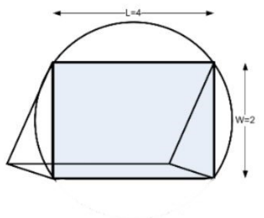

Figure 25: The diagram of a new arrangement

- Secondly, the ball itself has a volume. In this experiment, the ball was treated as a point. If the volume of the ball is considered, the solution might be closer to the real situation.
- Thirdly, the actions of the goalkeeper in figure 8 are too simplified. It is recommended that the rectangles should be divided smaller and more actions of the goalkeeper should be considered.
- Fourthly, the circle in figure 2 is too small to describe the shooting range for a player. A more realistic total area should be used to replace the current area. For example, change the magnitude of the radius according to the performance of the player.

3. Q3: With reference to Task-7 and Task-9, discuss the effect of changing the standard deviation of the Guassian distribution on both the accuracy and precision of the penaltyshots.

- According to the definition of normal distribution, the smaller standard deviation means more numbers accumulate in the centre (as Figure 26).

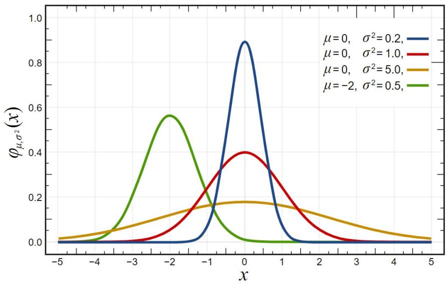

Figure 26: The diagram of normal distribution

In theory, the deviation influences the precision. However, in this experiment, the accuracy is also influenced by standard deviation. The accuracy increases with the decrease of the standard deviation.

4. Q4: If a large number of balls are kicked on the goal (i.e. ifNis sufficiently large), the valueof can be estimated using (some function of) the ratio of the number of scores to thetotal number of the shots. Hence, find the relation that estimates the value of π. Verifythis using your results for both uniform and Gaussian distributions.

   - The probability of scoring is $\frac{8}{5\pi}$. When N is sufficiently large, the simulation results is quite close to this value. If N = 100000, the simulation result for uniform random number generator is 0.5088. That is quite close to the theory. Then, we have the equation:
   $$\frac{8}{5\pi} = 0.5088 \tag{6}$$
   The magnitude of π is about 3.145. The result is close to the value of π.
   - For normal random number generator, when N = 100000, the probability is approximately 0.7321. Then we can calculate the value of π:
   $$\frac{8}{5\pi} = 0.7321 \tag{7}$$
   The magnitude of π is about 2.185. That is not close to the real value of π. That is because lots of points are located at the center. Therefore, it is not suitable for an ideal situation, which points can locate at any point in the circle.

   In conclusion, uniform random number is better than normal random number in estimating the value of π.

5. Q5 From your observation and results of Part II, what is the best strategy that should beadopted by the penalty taker? What is the best strategy that should be adopted by thegoalkeeper?

   - It is suggested that the penalty taker shoots the ball to the left upper corner and right upper corner of the goal since 90% of the time goalkeeper tends to protect the two lower corners.

- The goalkeeper is recommended to protect the according to the strategies in Figure 8 since more small rectangle areas can be covered. Moreover, if the shots are distributed normally, balls will lie in the middle of the rectangle. That is the right area protected by the goalkeeper.

4 Conclusions

4.1 Objective achievement

In conclusion, all the tasks are solved successfully. All the tasks are sloved by using Monte Carlo method. All the different conditions have been considered using six MATLAB code models. What is more, some useful conclusions have been summarized during the experiments.

4.2 Limitations

There are some limitations to this experiment

1. The experiment results are based on normal random generator and uniform random generator. These two generators are not 100 percent suitable for simulating the real-life condition when N is not quite large. What is more, the area of shooting range, the action of the goalkeeper and other factors also influence the experiment results.

2. The simulation time for a large number of shots is too long, such as R = 5 and N = 100000. It takes more than 8 minutes to calculate the value.

4.3 Future development

In the following experiment, the model can be settled more closer to the real situation, for instance, considering the volume of the ball and improving the mathematical method to simulate the shooting process.

5 References

[1] A.M. Johansen, *Monte Carlo Method.*International Encyclopedia of Education (Third Edition),2010.

[2] University of Liverpool, ”Monte Carlo Simulation script”,Ver 3.4, Sept, 2019.

Appendices

A Program Listings

A.1 Part I code

- **Model I**

```
function [N,R] = Partone(N,R)
p = 0;%define initial possibility for each turn
new = [];
rectangle('Position',[-2,-1,4,2],'Curvature',[0,0],
'Edgecolor','b','LineWidth',1);
% plot the rectangle and the circle
hold on
ezplot('x^2+y^2 = 5');
axis equal;
xlim([-3 3])
ylim([-3 3])
for turns = 1:R %turns to shoot the ball
    b = 0;
for a = 1:N % how many shoots for each turn

    r = sqrt(5*unifrnd(0,1));
    aphla = 2*pi*unifrnd(0,1);
    x = r*cos(aphla);
    y = r*sin(aphla); %generate a random point in the circle

    if(abs(x)<2 && abs(y)<1)
       b = b+1;
         shoot = plot(x,y,'+','color','r');
        % if the player shoots inside the rectangle, the color is red
        hold on
    else
         miss  = plot(x,y,'o','color','blue');
        % if the player shoots outside the rectangle, the color is blue

        hold on
    end

end
p0 = b/a; % calculate possibility for each turn
new = [new p0];% show the possibility for each turn
p = p+p0; %calculate the total possibility
end
poss = p/R %calculate the average possibility
```

- **Model II**

```
P = 0; %define the initial possibility
rectangle('Position',[-2,-1,4,2],'Curvature',[0,0],'Edgecolor',
'b','LineWidth',1);% plot the rectangle and the circle
hold on
ezplot('x^2+y^2 = 5');%plot the circle
axis equal;
for turns = 1:R %turns to shoot the ball
    b = 0;% Balls which shoot inside the goal
    T = 0;% The number of shoots
    c = 0;% Balls which shoot outside the circle
while(T-c<N) %Ensure N shots in circle
    r = normrnd(0,sqrt(5));
    aphla = 2*pi*rand();
    x = r*cos(aphla);
    y = r*sin(aphla); %generate a random point in the circle
    xlim([-3 3]);
    ylim([-3 3]);
    if(abs(x)<2 && abs(y)<1)
       b = b+1;
        shoot = plot(x,y,'+','color','r');% if the player shoots inside ...
            the rectangle, the color is green
        hold on
    else
        if (x^2+y^2>5) % Balls outside the circle
       c = c+1;
    else
        miss  = plot(x,y,'o','color','blue');% if the player shoots ...
            outside the rectangle, the color is red

        hold on
        end
       end
 T = T+1;
end
P0 = b/N;%possibility for each turn
P = P+P0;% The sum of all possibility
end
poss = P/R
%calculate the average possibility
```

A.2 Part II code

NB: The symbol "≤" and "≥" should be "<=" and ">=" in the MATLAB editor.

- **Model III**

```
p = 0;%define initial possibility for each turn
rectangle('Position',[-2,-1,4,2],'Curvature',[0,0],'Edgecolor',

'b','LineWidth',1);% plot the rectangle and the circle
hold on
ezplot('x^2+y^2 = 5');
axis equal;
xlim([-3 3]);
ylim([-3 3]);
for turns = 1:R %turns to shoot the ball
    b = 0;
for a = 1:N % how many shoots for each turn
r = sqrt(5*unifrnd(0,1));
aphla = 2*pi*unifrnd(0,1);
x = r*cos(aphla);
y = r*sin(aphla); %generate a random point in the circle
goalkeeper=round(unifrnd(0,1)*5+0.5);
switch(goalkeeper)%Consider the different actions for goal keeper
    case{1}
        if((x≥-2&&x≤-1)&&(y≥-1&&y≤1)||(x≤2&&x≥1)&&(y≥-1&&y≤1))
          b = b+1;
          shoot = plot(x,y,'+','color','r');% if the player shoots inside ...
              the rectangle, the color is red
          hold on
        else
          miss  = plot(x,y,'o','color','blue');% if the player shoots ...
              outside the rectangle, the color is blue
          hold on
        end
    case{2}
        if(((x≤-1&&x≥-2)&&(y≥-1 && y≤0))||((x≥-1 && x≤0)&&(y≥0 && y≤1))
        ||((x≥0 && x≤2)&&(y≥-1 && y≤1)))
        b = b+1;
        shoot = plot(x,y,'+','color','r');
        hold on
        else
        miss  = plot(x,y,'o','color','blue');
        hold on
        end
    case{3}
        if(((x≤1 && x≥2)&&(y≥-1 && y≤0))||((x≥0 && x≤1)&&(y≥0 && y≤1))
        ||((x≥-2 && x≤0)&&(y≥-1 && y≤1)))
        b = b+1;
        shoot = plot(x,y,'+','color','r');
        hold on
        else
       miss  = plot(x,y,'o','color','blue');
        hold on
        end
    case{4}
        if(((x≥-2 && x≤0)&&(y≥0 && y≤1))||((x≥0 && x≤2)&&(y≥-1 && y≤1)))
        b = b+1;
```

```
            shoot = plot(x,y,'+','color','r');
            hold on
            else
            miss  = plot(x,y,'o','color','blue');
            hold on
            end
        case{5}
            if(((x≥-2 && x≤0)&&(y≥-1 && y≤1))||((x≥0 && x≤2)&&(y≥0 && y≤1)))
             b = b+1;
           shoot = plot(x,y,'+','color','r');
            hold on
            else
            miss  = plot(x,y,'o','color','blue');
            hold on
            end
end

end
p0 = b/a; % calculate possibility for each turn
p = p+p0;
end
poss = p/R %calculate the average possibility
```

- **Model IV**

```
p = 0;%define initial possibility for each turn
rectangle('Position',[-2,-1,4,2],'Curvature',[0,0],'Edgecolor',
'b','LineWidth',1);% plot the rectangle and the circle
hold on
ezplot('x^2+y^2 = 5');
axis equal;
xlim([-3 3]);
ylim([-3 3]);
for turns = 1:R %turns to shoot the ball
    b = 0;% Balls which shoot inside the goal
    T = 0;% The number of shoots
    c = 0;% Balls which shoot outside the circle
while(T-c<N) %Ensure N shots in circle
    r = normrnd(0,sqrt(5));
    aphla = 2*pi*rand();
    x = r*cos(aphla);
    y = r*sin(aphla); %generate a random point in the circle
    xlim([-3 3]);
    ylim([-3 3]);
 if((x^2+y^2)>5)
     c = c+1;
 else
goalkeeper=round(unifrnd(0,1)*5+0.5);
switch(goalkeeper)
    case{1}
        if((x≥-2&&x≤-1)&&(y≥-1&&y≤1)||(x≤2&&x≥1)&&(y≥-1&&y≤1))
          b = b+1;
          shoot = plot(x,y,'*','color','g');% if the player shoots inside ...
              the rectangle, the color is green
          hold on
        else
          miss  = plot(x,y,'*','color','r');% if the player shoots ...
              outside the rectangle, the color is red
          hold on
```

```
            end
        case{2}
            if(((x≤-1&&x≥-2)&&(y≥-1&&y≤0))||((x≥-1&&x≤0)&&(y≥0&&y≤1))
            ||((x≥0&&x≤2)&&(y≥-1&&y≤1)))
            b = b+1;
            shoot = plot(x,y,'*','color','g');
            hold on
            else
            miss  = plot(x,y,'*','color','r');
            hold on
            end
        case{3}
            if(((x≤1&&x≥2)&&(y≥-1&&y≤0))||((x≥0&&x≤1)&&(y≥0&&y≤1))
            ||((x≥-2&&x≤0)&&(y≥-1&&y≤1)))
            b = b+1;
            shoot = plot(x,y,'*','color','g');
            hold on
            else
            miss  = plot(x,y,'*','color','r');
            hold on
            end
        case{4}
            if(((x≥-2&&x≤0)&&(y≥0&&y≤1))||((x≥0&&x≤2)&&(y≥-1&&y≤1)))
            b = b+1;
            shoot = plot(x,y,'*','color','g');
            hold on
            else
            miss  = plot(x,y,'*','color','r');
            hold on
            end
        case{5}
            if(((x≥-2&&x≤0)&&(y≥-1&&y≤1))||((x≥0&&x≤2)&&(y≥0&&y≤1)))
             b = b+1;
            shoot = plot(x,y,'*','color','g');
            hold on
            else
            miss  = plot(x,y,'*','color','r');
            hold on
            end
end

end
T = T+1;
end
p0 = b/N; % calculate possibility for each turn
p = p+p0;
end
poss = p/R %calculate the average possibility
```

- **Model V**

```
p = 0;%define initial possibility for each turn
rectangle('Position',[-2,-1,4,2],'Curvature',[0,0],'Edgecolor',
'b','LineWidth',1);% plot the rectangle and the circle
hold on
ezplot('x^2+y^2 = 5');
axis equal;
xlim([-3 3]);
ylim([-3 3]);
```

```
for turns = 1:R %turns to shoot the ball
    b = 0;
for a = 1:N % how many shoots for each turn
    a = a+1;
r = sqrt(5*unifrnd(0,1));
aphla = 2*pi*unifrnd(0,1);
x = r*cos(aphla);
y = r*sin(aphla); %generate a random point in the circle
jump = rand();
if jump≤0.9 % 90% of the time goalkeepers tend to jump to the  lower  two ...
     corners
    goalkeeper = round(rand()+4);
else
goalkeeper=round(rand()*3+0.5);
end
switch(goalkeeper)%Consider the different actions for goal keeper
    case{1}
        if((x≥-2&&x≤-1)&&(y≥-1&&y≤1)||(x≤2&&x≥1)&&(y≥-1&&y≤1))
          b = b+1;
          shoot = plot(x,y,'+','color','r');% if the player shoots inside ...
              the rectangle, the color is red
          hold on
        else
          miss  = plot(x,y,'o','color','blue');% if the player shoots ...
              outside the rectangle, the color is blue
          hold on
        end
    case{2}
        if(((x≤-1&&x≥-2)&&(y≥-1&&y≤0))||((x≥-1&&x≤0)&&(y≥0&&y≤1))
        ||((x≥0&&x≤2)&&(y≥-1&&y≤1)))
        b = b+1;
        shoot = plot(x,y,'+','color','r');
        hold on
        else
        miss  = plot(x,y,'o','color','blue');
        hold on
        end
    case{3}
        if(((x≤1&&x≥2)&&(y≥-1&&y≤0))||((x≥0&&x≤1)&&(y≥0&&y≤1))
        ||((x≥-2&&x≤0)&&(y≥-1&&y≤1)))
        b = b+1;
        shoot = plot(x,y,'+','color','r');
        hold on
        else
       miss  = plot(x,y,'o','color','blue');
        hold on
        end
    case{4}
        if(((x≥-2&&x≤0)&&(y≥0&&y≤1))||((x≥0&&x≤2)&&(y≥-1&&y≤1)))
        b = b+1;
        shoot = plot(x,y,'+','color','r');
        hold on
        else
        miss  = plot(x,y,'o','color','blue');
        hold on
        end
    case{5}
        if(((x≥-2&&x≤0)&&(y≥-1&&y≤1))||((x≥0&&x≤2)&&(y≥0&&y≤1)))
         b = b+1;
       shoot = plot(x,y,'+','color','r');
```

```
            hold on
            else
            miss  = plot(x,y,'o','color','blue');
            hold on
            end
    end

    end
    p0 = b/a; % calculate possibility for each turn
    p = p+p0;
    end
    poss = p/R %calculate the average possibility
```

- **Model VI**

```
p = 0;%define initial possibility for each turn
rectangle('Position',[-2,-1,4,2],'Curvature',[0,0],'Edgecolor',
'b','LineWidth',1);% plot the rectangle and the circle
hold on
ezplot('x^2+y^2 = 5');
axis equal;
xlim([-3 3]);
ylim([-3 3]);
for R = 1:R %turns to shoot the ball
    b = 0;% Balls which shoot inside the goal
    T = 0;%The number of shoots
    c = 0;% Balls which shoot outside the circle
while (T-c<N)%Ensure N shots in circle
r=normrnd(0,sqrt(5)); %Normal random generators of the
aphla=2*pi*rand(); %radius and the angle
x=r*cos(aphla); %Convert the polar coordinates to
y=r*sin(aphla); %the rectangular coordinates
if ((x^2+y^2)>5) %Count the points outside circle
c=c+1;
else
action=rand(); %possibility of action from 0 to 1
if action≤0.9 %90% protect the lower corner
goalkeeper=round(rand()+4);
else
goalkeeper=round(rand()*3+0.5);
end
switch(goalkeeper)%Consider the different actions for goal keeper
    case{1}
        if((x≥-2&&x≤-1)&&(y≥-1&&y≤1)||(x≤2&&x≥1)&&(y≥-1&&y≤1))
          b = b+1;
          shoot = plot(x,y,'+','color','r');% if the player shoots inside ...
              the rectangle, the color is red
          hold on
        else
          miss  = plot(x,y,'o','color','blue');% if the player shoots ...
              outside the rectangle, the color is blue
          hold on
        end
    case{2}
        if(((x≤-1&&x≥-2)&&(y≥-1&&y≤0))||((x≥-1&&x≤0)&&(y≥0&&y≤1))
        ||((x≥0&&x≤2)&&(y≥-1&&y≤1)))
        b = b+1;
        shoot = plot(x,y,'+','color','r');
        hold on
```

```
            else
            miss   = plot(x,y,'o','color','blue');
            hold on
            end
        case{3}
            if(((x≤1&&x≥2)&&(y≥-1&&y≤0))||((x≥0&&x≤1)&&(y≥0&&y≤1))
            ||((x≥-2&&x≤0)&&(y≥-1&&y≤1)))
            b = b+1;
            shoot = plot(x,y,'+','color','r');
            hold on
            else
           miss   = plot(x,y,'o','color','blue');
            hold on
            end
        case{4}
            if(((x≥-2&&x≤0)&&(y≥0&&y≤1))||((x≥0&&x≤2)&&(y≥-1&&y≤1)))
            b = b+1;
            shoot = plot(x,y,'+','color','r');
            hold on
            else
            miss   = plot(x,y,'o','color','blue');
            hold on
            end
        case{5}
            if(((x≥-2&&x≤0)&&(y≥-1&&y≤1))||((x≥0&&x≤2)&&(y≥0&&y≤1)))
             b = b+1;
           shoot = plot(x,y,'+','color','r');
            hold on
            else
            miss   = plot(x,y,'o','color','blue');
            hold on
            end
end
end
T = T+1;
end
p0 = b/N; % calculate possibility for each turn
p = p+p0;
end
poss = p/R %calculate the average possibility
```