The LATEXTemplate for MCM Version4.0

Edit by:ChinaTEX http://blog.sina.com.cn/wangzhaoli11

February 8, 2018

Abstract

We determine the sweet spot on a baseball bat. We capture the essential physics of the ballCbat impact by taking the ball to be a lossy spring and the bat to be an Euler-Bernoulli beam. To impart some intuition about the model, we begin by presenting a rigid-body model. Next, we use our full model to reconcile various correct and incorrect claims about the sweet spot found in the literature. Finally, we discuss the sweet spot and the performances of corked and aluminum bats, with a particular emphasis on hoop modes.

Key Words: keyword1;keyword2

Contents

1 Introduction

Although a hitter might expect a model of the batCbaseball collision to yield insight into how the bat breaks, how the bat imparts spin on the ball, how best to swing the bat, and so on, we model only the sweet spot.

There are at least two notions of where the sweet spot should bełan impact location on the bat that either

- minimizes the discomfort to the hands, or
- maximizes the outgoing velocity of the ball.

We focus exclusively on the second definition.

- the initial velocity and rotation of the ball,
- the initial velocity and rotation of the bat,
- the relative position and orientation of the bat and ball, and
- the force over time that the hitters hands applies on the handle.

We assume that the ball is not rotating and that its velocity at impact is perpendicular to the length of the bat. We assume that everything occurs in a single plane, and we will argue that the hands interaction is negligible. In the frame of reference of the center of mass of the bat, the initial conditions are completely specified by

- the angular velocity of the bat,
- the velocity of the ball, and
- the position of impact along the bat.

The location of the sweet spot depends not on just the bat alone but also on the pitch and on the swing. The simplest model for the physics involved has the sweet spot at the *center of percussion* [Brody 1986], the impact location that minimizes discomfort to the hand. The model assumes the ball to be a rigid body for which there are conjugate points: An impact at one will exactly balance the angular recoil and linear recoil at the other. By gripping at one and impacting at the other (the center of percussion), the hands experience minimal shock and the ball exits with high velocity. The center of percussion depends heavily on the moment of inertia and the location of the hands. We cannot accept this model because it both erroneously equates the two definitions of sweet spot and furthermore assumes incorrectly that the bat is a rigid body. Another model predicts the sweet spot to be between nodes of the two lowest natural frequencies of the bat [Nathan 2000]. Given a free bat allowed to oscillate, its oscillations can be decomposed into fundamental modes of various frequencies. Different geometries and materials have different natural frequencies of oscillation. The resulting wave shapes suggest how to excite those modes (e.g., plucking a string at the node of a vibrational mode will not excite that mode).

Theorem 1.1. *LaTeX*

Lemma 1.2. *TeX.*

Proof. The proof of theorem. □

1.1 Other Assumptions

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

-
-
-
-

Under the above and basic assumptions, we can set out to construct our model (show our approach in detail).

2 Analysis of the Problem

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hope-

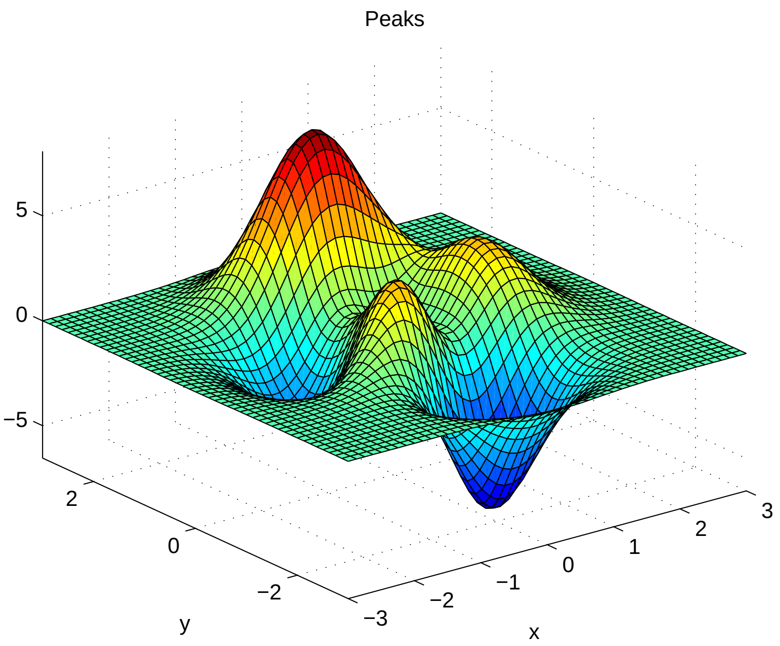

Figure 1: aa

ful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.(1)

$$a^2 \tag{1}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \frac{Opposite}{Hypotenuse} \cos^{-1}\theta \arcsin\theta$$

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

$$p_j = \begin{cases} 0, & \text{if } j \text{ is odd} \\ r!\,(-1)^{j/2}, & \text{if } j \text{ is even} \end{cases}$$

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

$$\arcsin\theta = \oiiint_{\varphi} \lim_{x\to\infty} \frac{n!}{r!\,(n-r)!} \tag{1}$$

3 Calculating and Simplifying the Model

"A is equivalent to B" Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

4 The Model Results

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

5 Validating the Model

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said. Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

6 Conclusions

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

7 A Summary

Although Mr. Gore has expressed concerns to some associates about the damage a brokered convention could cause, several associates said he was hopeful that one candidate would soon break through, sparing the party such an outcome. He told a close friend recently that his decision not to endorse feels like the right thing and that he remained optimistic the race is going to tip at some point, the friend said.

8 Evaluate of the Mode

9 Strengths and weaknesses

Like any model,the one present above has its strengths and weaknesses. Some of the major points are presented below.

9.1 Strengths

- **Applies widely**
  This system can be used for many types of airplanes, and it also solves the interference during the procedure of the boarding airplane,as described above we can get to the optimization boarding time.We also know that all the service is automate.

- **Improve the quality of the airport service**
  Balancing the cost of the cost and the benefit, it will bring in more convenient for airport and passengers.It also saves many human resources for the airline.

-

References

[D. E. KNUTH] D. E. KNUTH The TeXbook the American Mathematical Society and AddisonCWesley Publishing Company , 1984-1986.

[1] Lamport, Leslie, LaTeX: “ A Document Preparation System ”, Addison-Wesley Publishing Company, 1986.

Appendices

Appendix A First appendix

some text...

Here are simulation programmes we used in our model as follow.

Input matlab source:

```
function [t,seat,aisle]=OI6Sim(n,target,seated)

% OI6
%    n
%    target
%    seated
%   t simulation time
%   seat ,seat interference
%   aisle ,aisle interference


% Initial data
% setting passengers’ Value:
%                   walking          .....   1
%                   waiting          .....   2
%                   putting luggage  .....   3
%                   passing the seat .....   4
%                   sitting           .....  0


% on initial time, everyone is waiting,except the first one

seat=0;
aisle=0;

status=2*ones(1,n);
status(1)=1;

%
%           ł 0 .6 row
pos=-(0:0.6:(n-1)*0.6);



   pri=[-1,1:n-1];
   next=[2:n,-1];


RowSpeed=trirnd(0.6,0.95,1.3,1,n);

pab=rand(1,n);
for i=1:n
    if pab(i)<0.4
       aisleTime(i)=0;
    else
        aisleTime(i)=trirnd(3.2,7.1,38.7);
    end
end

% seat interference time
seatTime=trirnd(7.4,9.7,15.5);

t=0;
while sum(status) ~=0

    t=t + 0.1;
    for i=1:n
```

```
        switch status(i)
            case {0}

                 if  next(i)>0 &&abs(status(next(i))-2)<0.1
                        status(next(i))=1;
                     end

%                  disp('have sit down');

            case {1}

%                  disp('Walking');

                     if  next(i)>0 &&abs(status(next(i))-2)<0.1
                        status(next(i))=1;
                     end


                     pos(i)=pos(i)+RowSpeed(i)*0.1;

                     if abs(pos(i)-target(1,i))<0.2

                         status(i)=3;
                         if abs(aisleTime(i))<0.01
                             aisle=aisle+1;
                         end

                             if  next(i)>0 &&abs(status(next(i))-1)<0.1

                                 status(next(i))=2;
                         end

                      end


            case {2}

%                   disp('Blocking');
                  if  next(i)>0 &&abs(status(next(i))-1)<0.1

                          status(next(i))=2;
                     end

            case {3}          %put luggage     aisle interference

                   disp('aisle interference');

                  if abs(aisleTime(i))<0.01

                    if  n==12 && target(2,i)==1 && seated(target(1,i),2)==1
                          status(i) = 4;
                          seat=seat+1;
                      elseif  n==12 && target(2,i)==4 && seated(target(1,i),3)==1
                          status(i) = 4;
                          seat=seat+1;
                      else

                           status(i)=0;
                           seated(target(1,i),target(2,i)) = 1;
                      end

                  else
                      aisleTime(i)=aisleTime(i)-0.1;

                  end

            case {4}                     % seat interference

                if abs(aisleTime(i))<0.01   %
                    status(i)=0;
                    seated(target(1,i),target(2,i)) = 1;
                else

                    seatTime(i)=seatTime(i)-0.1;
                end


        end  %switch
    end %for

end %while
```

Appendix B Second appendix

some more text**Input C++ source:**

```
//============================================================================
// Name        : Sudoku.cpp
// Author      : wzlf11
// Version     : a.0
// Copyright   : Your copyright notice
// Description : Sudoku in C++.
//============================================================================

#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;
```

```
int table[9][9];

void shuffle(int arr[], int n){

    int tmp, rd;

    //
    for(int i = 0; i < n; i++){
        rd = rand() % 9;
        tmp = arr[rd];
        arr[rd] = arr[i];
        arr[i] = tmp;
    }
}

bool test(int x, int y, int v){

    int _x = x / 3 * 3;
    int _y = y / 3 * 3;

    //   3  * 3
    for(int i=_x; i<_x + 3; i++)
    {
        for(int j = _y; j < _y + 3; j++)
        {
            if(table[i][j] == v)
            {
                return false;
            }
        }
    }

    //
    for(int m= 0; m< 9; m++)
    {
        if(table[x][m] == v || table[m][y] == v)
            return false;
    }

    return true;
}

bool put(int line, int index){

    if(index > 8)
        return true;

    //                    0
    if(table[line][index] != 0){
        return put(line, index + 1);
    }
    int num[] = {1,2,3,4,5,6,7,8,9};
    //
    shuffle(num, 9);

    for(int i = 0; i < 9; i++){

        //
        if( test(line, index, num[i]) == true ){

            table[line][index] = num[i];

            //
            if( put(line, index + 1) == true ){
                return true;
            }
        }
    }

    table[line][index] = 0; //

    return false;
}

bool put_line(int line){

    if(line > 8)
        return true;

    int b[9];

    //
    for(int i = 0; i < 9; i++){
        b[i] = table[line][i];
    }

    if( put(line, 0) == true ){
        //
        if( put_line(line + 1) == true )
            return true;
    }

    //
    for(int n = 0; n < 9; n++){
        table[line][n] = b[n];
    }

    return false;
}

int main() {
    //
    for(int i = 0; i < 9; i++){
        table[0][i] = i + 1;
    }

    srand((unsigned int)time(NULL));
    //
    shuffle((int *)&table[0], 9);

    //         ł
    while(!put_line(1))
    {
        //         ł
        shuffle((int *)&table[0], 9);
    }

    //
    for(int x = 0; x < 9; x++){
        for(int y = 0; y < 9; y++){
            cout << table[x][y] << " ";
        }

        cout << endl;
    }

    return 0;
}
```