Computer Science Solutions
(http://computer-science-solutions.blogspot.com/)

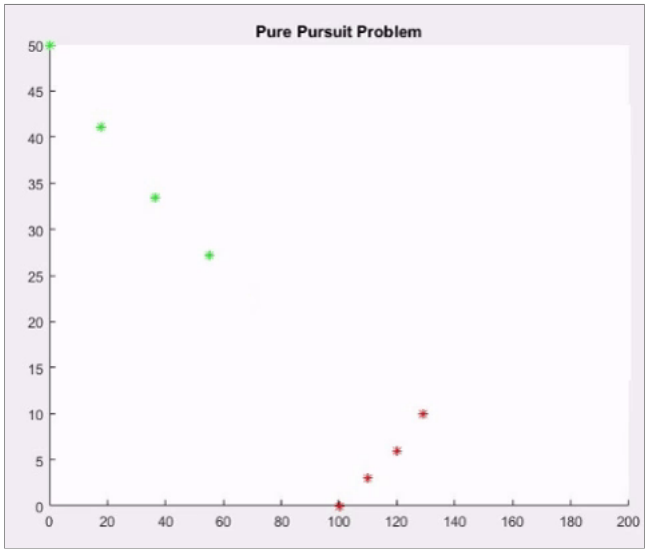
Pure Pursuit Problem Code - Matlab and C code of Pure pursuit problem

Maniruzzaman-Akash December 05, 2017 (2017-12-05T02:08:00-08:00)
Mathematics (http://computer-science-solutions.blogspot.com/search/label/Mathematics) ,
MatLab Code (http://computer-science-solutions.blogspot.com/search/label/MatLab%20Code) ,
Simulation and Modeling (http://computer-science-solutions.blogspot.com/search/label/Simulation%20and%20Modeling)
No comments (http://computer-science-solutions.blogspot.com/2017/12/pure-pursuit-problem-code-matlab-and-c.html#co

Pure Pursuit Problem Code - Matlab and C code of Pure pursuit problem

What is Pure pursuit:

Pure pursuit is a type of pursuit curve used in aerial combat in which an aircraft pursues another aircraft by pointing its nose directly towards it.



What is mainly Pure pursuit is:

Logic Behind the pure pursuit problem of simulation:

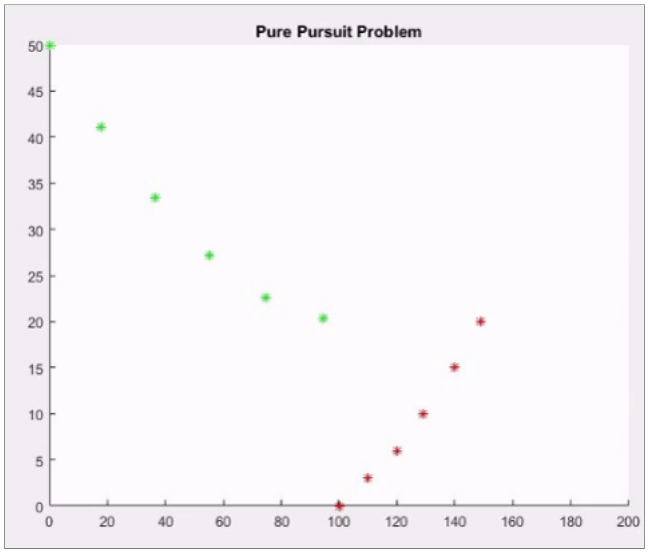
1. Bomber Aircraft and a Fighter Aircraft are flying in the a horizontal plane.
2. Fighter aircraft and bomber aircraft both are moving inside the rectangular range.
3. The fighters and bombers have a velocity given, suppose **s = 20** in our code.
4. When the distance of the Bomber and the Fighter is less than **12 units**, it is assumed that the Bomber is shot down or destroyed.
5. The distance between this the bombers and fighter follows the distance rule - dist[t] = √((yb[t]-yf[t])² + (xb[t]-yf[t])²).
6. In matlab the distance will be dist = sqrt(y^2+x^2).
7. Look the pure pursuit problem code now and hope it'll clear to you.

Pure Pursuit Problem MatLab Code

```
clc;
hold all;
xb=[100 110 120 129 140 149 158 168 179 188 198 209 219 226 234 240];
yb=[0 3 6 10 15 20 26 32 37 34 30 27 23 19 16 14];

xf = [];
yf = [];
xf(1)=0;
yf(1)=50;
s=20;
dist=0;

for i=1:15
   pause on;
    plot(xb(i),yb(i),'r*');
    title('Pure Pursuit Problem');
    pause(1);
    plot(xf(i),yf(i),'g*');
    y=yb(i)-yf(i);
    x=xb(i)-xf(i);

  dist=sqrt(y^2+x^2);

  if(dist<=12)
        fprintf('Bomber destroyed at  %d s',i);
        break;
  end

  xf(i+1)=xf(i)+s*((xb(i)-xf(i))/dist);
  yf(i+1)=yf(i)+s*((yb(i)-yf(i))/dist);
end
```

Pure Pursuit Problem Output - Matlab:

```
>> Bomber destroyed at  11 s
>>
```



Pure Pursuit problem in C language:

Code:

```
#include < stdio.h >
#include < math.h >
#include < stdlib.h >

void main()
  {
   float xf,yf, xb,yb,d,distance;
   int flag=0,
   vf=20,
   time=0;
   randomize();
   xf=rand()%1001;
   yf=rand()%1001;
   xb=rand()%1001;
   yb=rand()%1001;
   while(flag==0)
    {
 d= (yb-yf)*(yb-yf)+(xb-xf)*(xb-xf);
 distance=sqrt(d);
 printf("time=%d   xf=%5.2f  yf=%5.2f  xb=%5.2f  yb=%5.2f  distance=%5.2f\n\n",time,xf,yf,xb,yb
 if(distance >100)
   {
    printf("The bomber plain was shot down at %d second\n",time);
    flag=1;
   }
 else if(distance>900)
   {
     printf("The bomber plane escaped from sight at %d second\n", time);
     flag=1;
   }
 else
   {
     xf=xf+vf*(xb-xf)/distance;
     yf=yf+vf*(yb-yf)/distance;
     xb=rand()%1001;
     yb=rand()%1001;
     time=time+1;
   }
    }
    getch();
}
```

Output C code Pure Pursuit:

```
CASE 1:   BOMBER   IS SHOT DOWN BY FIGHTER

time=0    xf=688.00   yf=796.00   xb=366.00   yb=119.00   distance=749.68

time=1    xf=679.41   yf=777.94   xb=563.00   yb=771.00   distance=116.62

time=2    xf=659.45   yf=776.75   xb=419.00   yb=939.00   distance=290.07

time=3    xf=642.87   yf=787.94   xb=87.00   yb=931.00   distance=573.98

time=4    xf=623.50   yf=792.92   xb=960.00   yb=247.00   distance=641.30

time=5    xf=633.99   yf=775.90   xb=197.00   yb=203.00   distance=720.54
```

Links Where you can learn more on Pure pursuit problem:

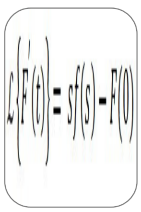
1. Wikipedia - Pure Pursuit (https://en.wikipedia.org/wiki/Pure_pursuit)
2. Mathworks - Pure Pursuit (https://www.mathworks.com/help/robotics/ug/pure-pursuit-controller.html)

Tags:

Pure Pursuit Problem Code - Matlab and C code of Pure pursuit problem, Pure pursuit problem, Pure pursuit problem code, pure pursuit solution in c language, pure pursuit simulation, pure pursuit logic, pure pursuit algorithm, pure pursuit learning, pure pursuit explanation,

Related Posts

Single Server Queuing System - MatL...
(http://computer-science-solutions.blogspot.com/2017/12/single-server-queuing-system-matlab-c.html)



Laplace Transform Derivatives Theor...
(http://computer-science-solutions.blogspot.com/2017/11/laplace-transform-derivatives-theorem.html)

Mathematics Book Differential Equat...
(http://computer-science-solutions.blogspot.com/2017/10/mathematics-book-differential-equation.html)

Newer Post (http://computer-science-solutions.blogspot.com/2017/12/gambling-game-code-implementation-in.html)

Older Post (http://computer-science-solutions.blogspot.com/2017/12/single-server-queuing-system-matlab-c.html)

No comments :

Post a Comment

(https://www.blogger.com/comment-iframe.g?blogID=8056803275198378549&postID=1533550053528956611&blogspotRpcToken=5198814)

Subscribe to: Post Comments (Atom) (http://computer-science-solutions.blogspot.com/feeds/1533550053528956611/comments/default)

Search

Search

Recent

About Author

(https://maniruzzaman-akash.blogspot.com/p/contact.html)

Maniruzzaman Akash, Student of Patuakhali Science Technology University (www.pstu.ac.bd) in Bangladesh at Computer Science & Engineering department.

He is a free minded web developer, designer. Expert with HTML, Bootstrap, PHP, Codeigniter-3, Laravel-5, Javascript, Jquery

Email: manirujjamanakash@gmail.com (mailto:manirujjamanakash@gmail.com)