Pursuit Curves

Katy Steiner, Jonah Franchi

5/10/2011

Abstract

The purpose of this paper is to demonstrate and explain how pursuit curves are derived and modeled, give examples of different types of pursuit curves, and finally show a more complex pursuit curve in 3-D space.

1 Introduction & Assumptions

A pursuit curve is, as its name implies, a curve showing the path an object takes as it pursues another object. The velocity vector of the pursuer is always going directly towards the prey, which excluding the idea of bending space time, is a straight line. In order to model pursuit curves a few assumptions must be made. First, all prey are following a set path at a constant speed. Second, there is nothing in the way of either prey or predator and they have unlimited energy.

2 A Basic Pursuit Curve

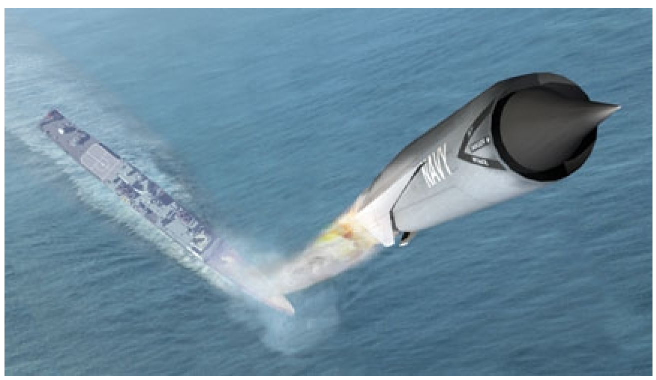
(All of the proceeding in the section was found and derived by "http://online.redwoods.cc.ca.us/instruct/darnold/deproj/sp08/mseverdia/pursuit.pdf")

For our first example of a pursuit curve, consider our predator being a missile and our prey, a ship.

The ship starts at $(x_0, 0)$ at $t \geq 0$ and is traveling at V_s along the vertical line $x = x_0$. The missile starts at $(0, 0)$ at time $t = 0$ and travels at a constant speed of V_m along a curved path so that its velocity vector is pointing directly at the ship.

If the missile at time $t \geq 0$ is the point (x, y) and the position of the ship is $(x_0, V_s\, t)$ then the slope of the tangent to the pursuit curve at (x, y) is

$$\frac{dy}{dx} = \frac{y - V_s\, t}{x - x_0}$$



Solve for t to get

$$t = \frac{y}{V_s} - \frac{x - x_0}{V_s}\frac{dy}{dx} \tag{1}$$

The missile has always traveled a distance of $V_m t$ along the pursuit curve so using the arc length formula we get

$$V_m t = \int_0^x \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dz$$

Then we solve this for t and get

$$t = \frac{1}{V_m}\int_0^x \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dz \tag{2}$$

Combining equation (1) and (2) gives

$$\frac{1}{V_m}\int_0^x \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dz = \frac{y}{V_s} - \frac{x - x_0}{V_s}\frac{dy}{dx}$$

Let $p(x) = dy/dx$

$$\frac{1}{V_m}\int_0^x \sqrt{1+[p(z)]^2}dz = \frac{y}{V_s} - \frac{x-x_0}{V_s}p(x)$$

Differentiate both sides

$$\frac{1}{V_m}\sqrt{1+[p(x)]^2} = \frac{1}{V_s}\frac{dy}{dx} - \frac{x-x_0}{V_s}\frac{dp}{dx} - \frac{1}{V_s}p(x)$$

Replace dy/dx with $p(x)$ and reduce

$$(x-x_0)\frac{dp}{dx} = -n\sqrt{1+[p(x)]^2} \tag{3}$$

Where $n = V_s/V_m$, which is our relative speed.
Next we solve this first by separating the variables, letting p=p(x)

$$\frac{dp}{\sqrt{1+p^2}} = \frac{-ndx}{x-x_0}$$

Integrate both sides with a table of integrals

$$\ln(p+\sqrt{1+p^2}) + C = -n\ln(x_0 - x) \tag{4}$$

Since $p = dy/dx$, and the missiles velocity is always pointed directly toward the ship, at time $t = 0$ the missiles velocity is $dy/dx = 0$. Solving for C, we find that $C = -n\ln(x_0)$.

Insert this result into equation (4) to get

$$\ln(p+\sqrt{1+p^2}) + \ln(x_0-x)^n - \ln(x_0)^n = 0$$

Using the properties of logarithms we can say

$$0 = \ln(p+\sqrt{1+p^2}) + \ln\left(\frac{x_0-x}{x_0}\right)^n$$

$$0 = \ln\left[(p+\sqrt{1+p^2})\left(1-\frac{x}{x_0}\right)^n\right]$$

Since $ln(1) = 0$

$$(p+\sqrt{1+p^2})\left(1-\frac{x}{x_0}\right)^n = 1$$

Now we try to solve for p

$$p+\sqrt{1+p^2} = \left(1-\frac{x}{x_0}\right)^{-n}$$

Let $q = (1-x/x_0)^n$

$$p+\sqrt{1+p^2}=q$$

$$1+p^2=(q-p)^2$$

$$1+p^2=q^2-2qp+p^2$$

$$2qp=q^2-1$$

$$p=\frac{1}{2}\left(q-\frac{1}{q}\right)$$

Substitute back in for p and q

$$\frac{dy}{dx}=\frac{1}{2}\left[\left(1-\frac{x}{x_0}\right)^{-n}-\left(1-\frac{x}{x_0}\right)^{n}\right]$$

Now integrate with respect to x

$$y(x)+C=\frac{1}{2}\int\left(1-\frac{x}{x_0}\right)^{-n}dx-\frac{1}{2}\int\left(1-\frac{x}{x_0}\right)^{n}dx$$

Let $u=1-x/x_0$, $du=(-1/x_0)dx$

$$y(x)+C=\frac{1}{2}\int -x_0u^{-n}du-\frac{1}{2}\int -x_0u^{n}du$$

$$y(x)+C=-\frac{1}{2}x_0\left(\frac{u^{-n+1}}{-n+1}\right)+\frac{1}{2}x_0\left(\frac{u^{n+1}}{n+1}\right)$$

Simplify

$$y(x)+C=\frac{1}{2}x_0\left[\frac{u^{n+1}}{n+1}-\frac{u^{-n+1}}{-n+1}\right]$$

$$y(x)+C=\frac{1}{2}x_0u\left[\frac{u^{n}}{n+1}-\frac{u^{-n}}{-n+1}\right]$$

Substitute back in for u

$$y(x)+C=\frac{1}{2}x_0\left(1-\frac{x}{x_0}\right)\left[\frac{\left(1-\frac{x}{x_0}\right)^{n}}{n+1}-\frac{\left(1-\frac{x}{x_0}\right)^{-n}}{-n+1}\right]$$

$$y(x)+C=\frac{1}{2}\left(x_0-x\right)\left[\frac{\left(1-\frac{x}{x_0}\right)^{n}}{n+1}-\frac{\left(1-\frac{x}{x_0}\right)^{-n}}{-n+1}\right]$$

Since the missile starts at the origin $y(0) = 0$ we can solve for C

$$C = \frac{1}{2}x_0 \left[\frac{1}{1+n} - \frac{1}{1-n}\right]$$

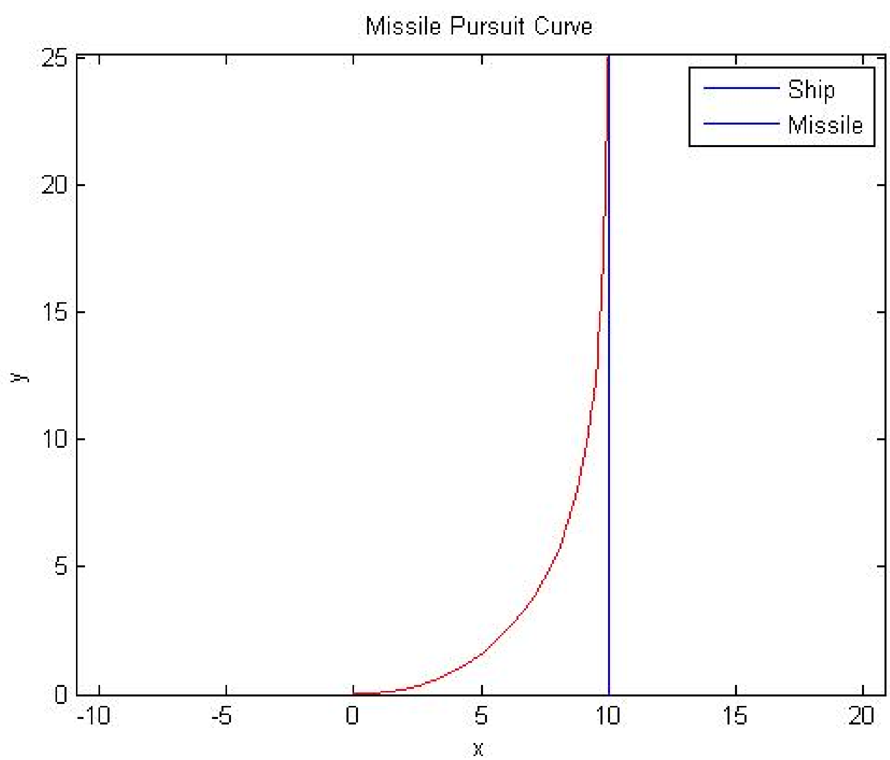
$$C = \frac{1}{2}x_0 \left[\frac{-2n}{1-n^2}\right]$$

$$C = \frac{1}{2}x_0 \left[\frac{-2n}{1-n^2}\right]$$

We now have our solution to (3)

$$y(x) = \frac{1}{2}\left(x_0 - x\right)\left[\frac{(1-x/x_0)^n}{1+n} - \frac{(1-x/x_0)^{-n}}{1-n}\right] + \frac{n}{1-n^2}x_0$$

where $n = V_s/V_m$
Solving and graphing in Matlab produces the following graph:



The above method just shown uses hand calculations in the most basic case to solve for the path of the predator. This method proves to be a bit lengthy for our purposes, so we will use a different method in the next sections. This method will allow for us to enter our equations into our numerical solver, MATLAB, and produce graphs of the pursuit curves.

3 A Circular Pursuit Curve

(All of the proceeding in the section was found and derived by
"http://online.redwoods.cc.ca.us/instruct/darnold/deproj/sp08/mseverdia/presentation.pdf")

Pursuit curves can take on many forms. Apart from a linear or parabolic form, pursuit curves can also be circular. For our example, consider our predator to be a bully and our prey to be a nerd. The nerd runs in a circular path in a unit circle and is chased by a bully from the inside. Our bully will be represented by $B(x(t), y(t))$, and our nerd by $N(p(t), q(t))$. The path of the nerd is represented by a vector equation:

$$N(t) = p(t)\mathbf{i} + q(t)\mathbf{j}$$

The pursuit curve of the bully is also represented by a vector equation:

$$B(t) = x(t)\mathbf{i} + y(t)\mathbf{j}$$

The path of the nerd is described by p and q as functions of time. For the pursuit curve of the bully, x and y are unknown because they depend on the path of the nerd. We find x and y by developing a system of differential equations using two unit vectors.

A unit vector is a vector whose magnitude is 1. It's formed by dividing any one vector, **n** by its magnitude.

$$\mathbf{u} = \frac{\mathbf{n}}{|\mathbf{n}|}$$

The first unit vector used is the unit tangent vector, **T**, or the unit vector tangent to **B**. $\mathbf{V_B}$ will represent the velocity of the bully and $\mathbf{V_N}$ is the velocity of the nerd. We know that $|\mathbf{V_B}| = k|\mathbf{V_N}|$ because the velocities are proportional (the bully is running at k relative to the speed of the nerd). Using this knowledge we find **T**.

Note that k is the same as n in our previous section, but as this was taken from a different source we left the values as the originals were.

$$\mathbf{T} = \frac{\mathbf{V_B}}{|\mathbf{V_B}|} = \frac{\mathbf{V_B}}{k|\mathbf{V_N}|}$$

The second unit vector is the unit difference vector, **D**, in the direction of $\mathbf{B} - \mathbf{N}$:

$$\mathbf{D} = \frac{\mathbf{N} - \mathbf{B}}{|\mathbf{N} - \mathbf{B}|}$$

The unit vectors **T** and **D** are equal because the bully is always heading toward the nerd.

$$\mathbf{T} = \mathbf{D}$$

Using $\mathbf{T}$ and $\mathbf{D}$, we can solve for $\mathbf{V_B}$:

$$\frac{\mathbf{V_B}}{k|\mathbf{V_N}|} = \frac{\mathbf{N}-\mathbf{B}}{|\mathbf{N}-\mathbf{B}|}$$

$$\mathbf{V_B} = k|\mathbf{V_N}|\frac{\mathbf{N}-\mathbf{B}}{|\mathbf{N}-\mathbf{B}|}$$

Converting $\mathbf{V_B}$ into vector notation will allow us to equate coefficients and develop a system of differential equations.

$$\mathbf{V_B} = k|\mathbf{V_N}|\frac{\mathbf{N}-\mathbf{B}}{|\mathbf{N}-\mathbf{B}|}$$

$$= k\sqrt{\left(\frac{dp}{dt}\right)^2 + \left(\frac{dq}{dt}\right)^2}\,\frac{(p-x)\hat{\imath} + (q-y)\hat{\jmath}}{\sqrt{(p-x)^2 + (q-y)^2}}$$

Equating coefficients gives:

$$\frac{dx}{dt} = k\sqrt{\left(\frac{dp}{dt}\right)^2 + \left(\frac{dq}{dt}\right)^2}\,\frac{(p-x)}{\sqrt{(p-x)^2 + (q-y)^2}}$$

$$\frac{dy}{dt} = k\sqrt{\left(\frac{dp}{dt}\right)^2 + \left(\frac{dq}{dt}\right)^2}\,\frac{(q-y)}{\sqrt{(p-x)^2 + (q-y)^2}}$$

Now we can use these equations for any k, p, and q.

So for our example, we will let the bully chase the nerd at 75% of the nerd's speed, so $k = .75$. Since the nerd is running in a unit circle, our equations are:

$$p(t) = \cos t$$

$$q(t) = \sin t$$

$$\frac{dp}{dt} = -\sin t$$

$$\frac{dq}{dt} = \cos t$$

When we substitute k, p, and q into our general differential equations, we get the following system of differential equations

$$\frac{dx}{dt} = (.75)\sqrt{(-\sin t)^2 + (\cos t)^2}\,\frac{(\cos t - x)}{\sqrt{(\cos t - x)^2 + (\sin t - y)^2}}$$

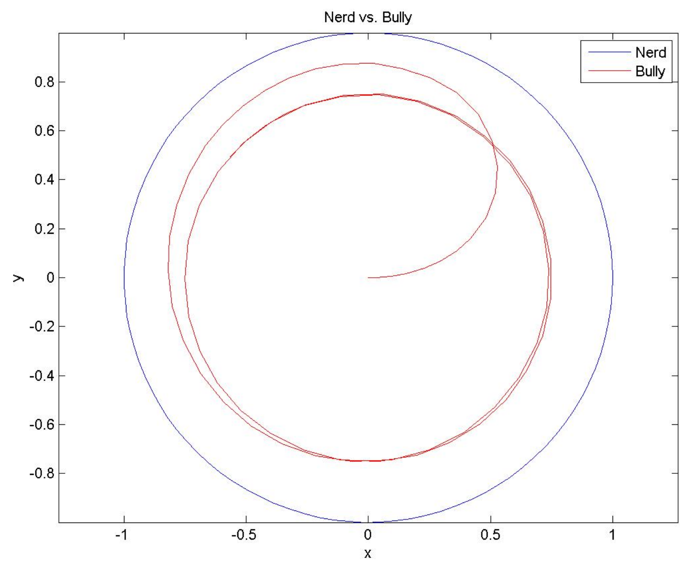
$$\frac{dy}{dt} = (.75)\sqrt{(-\sin t)^2 + (\cos t)^2}\,\frac{(\sin t - y)}{\sqrt{(\cos t - x)^2 + (\sin t - y)^2}}$$

which can be simplified.

$$\frac{dx}{dt} = (.75)\frac{(\cos t - x)}{\sqrt{(\cos t - x)^2 + (\sin t - y)^2}}$$

$$\frac{dy}{dt} = (.75)\frac{(\sin t - y)}{\sqrt{(\cos t - x)^2 + (\sin t - y)^2}}$$

Solving and graphing in Matlab produces the following graph:



This shows that as the bully chases the nerd following a standard pursuit path, as long as his speed isn't at least equal to the speed of the nerd he won't even be on the same path as him.

4 Other Pursuit Curves

Our next set of equations show other variations of paths that these curves can take.

4.1 Spider Curve

Given the path of the prey to be:

$$p(t) = t \sin t$$

$$q(t) = 8 \cos 6t$$

$$\frac{dp}{dt} = t \cos t + sint$$

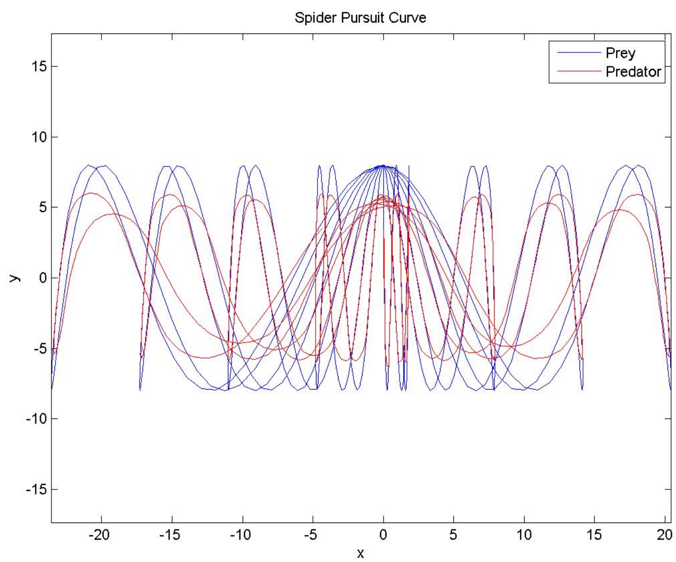
$$\frac{dq}{dt} = -48 \sin 6t$$

We will let $k = .7$. Our differential equations are as follows:

$$\frac{dx}{dt} = (.7)\sqrt{(t\cos t + sint)^2 + (-48\sin 6t)^2}\frac{(t\sin t - x)}{\sqrt{(t\sin t - x)^2 + (8\cos 6t - y)^2}}$$

$$\frac{dy}{dt} = (.7)\sqrt{(t\cos t + sint)^2 + (-48\sin 6t)^2}\frac{(8\cos 6t - y)}{\sqrt{(t\sin t - x)^2 + (8\cos 6t - y)^2}}$$

Solving and graphing in Matlab produces the following graph:



Because the speed of the predator doesn't match the speed of the prey the afterimage of the path is slightly below the prey at most times.

4.2 Lion and Gazelle

Our next example will be a lion chasing a gazelle with the following equations:

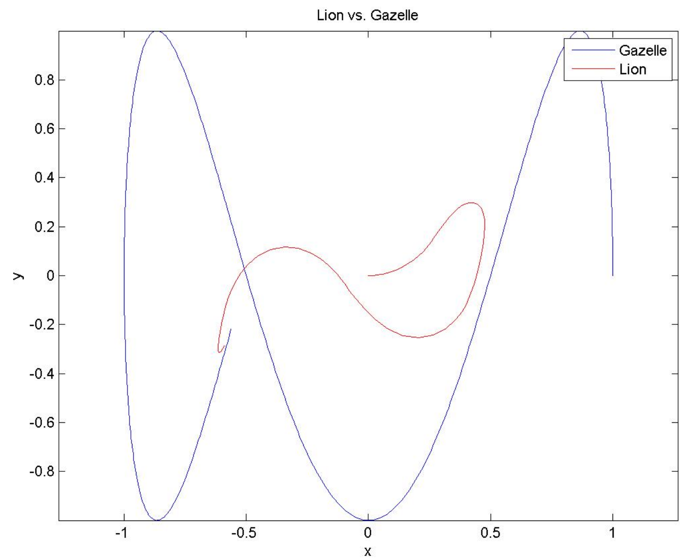
$$p(t) = \cos t$$

$$q(t) = \sin 3t$$

$$\frac{dp}{dt} = -\sin t$$

$$\frac{dq}{dt} = 3\cos 3t$$

Solving and graphing in Matlab produces the following graph:



5 A 3D Pursuit Curve

Now that we have delved a bit into pursuit curves in 2 dimensional space, we will now go into how pursuit curves work in 3 dimensional space. A third variable is needed for pursuit curves in 3 dimensions so we will have three differential equations. Thus the differential equations we will use look like the following:

$$\frac{dx}{dt} = k\sqrt{\left(\frac{dp}{dt}\right)^2 + \left(\frac{dq}{dt}\right)^2 + \left(\frac{dr}{dt}\right)^2}\,\frac{(p-x)}{\sqrt{(p-x)^2+(q-y)^2+(r-z)^2}}$$

$$\frac{dy}{dt} = k\sqrt{\left(\frac{dp}{dt}\right)^2 + \left(\frac{dq}{dt}\right)^2 + \left(\frac{dr}{dt}\right)^2}\,\frac{(q-y)}{\sqrt{(p-x)^2+(q-y)^2+(r-z)^2}}$$

$$\frac{dz}{dt} = k\sqrt{\left(\frac{dp}{dt}\right)^2 + \left(\frac{dq}{dt}\right)^2 + \left(\frac{dr}{dt}\right)^2}\,\frac{(r-z)}{\sqrt{(p-x)^2+(q-y)^2+(r-z)^2}}$$

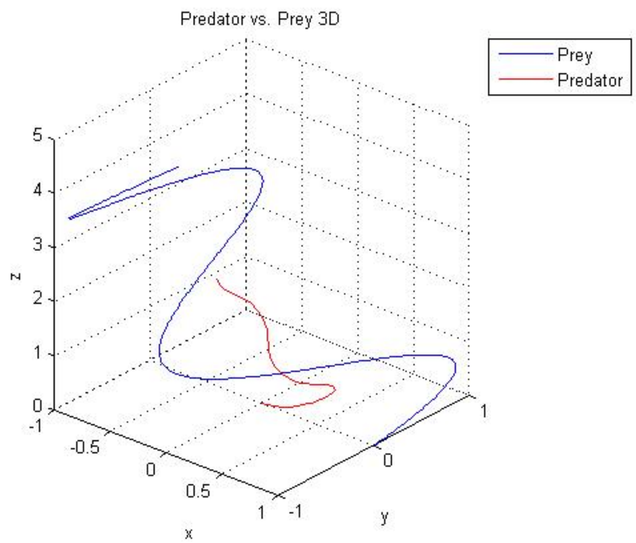
Since three equations are needed to graph a pursuit curve in 3D, we will let the third equation equal t in order to produce a graph of our curve verses time for our lion verses gazelle example. In this example, since it is our first in 3-D, we are simply projecting our graph into the t plane and not really making a new curve.

Our equations are as follows:

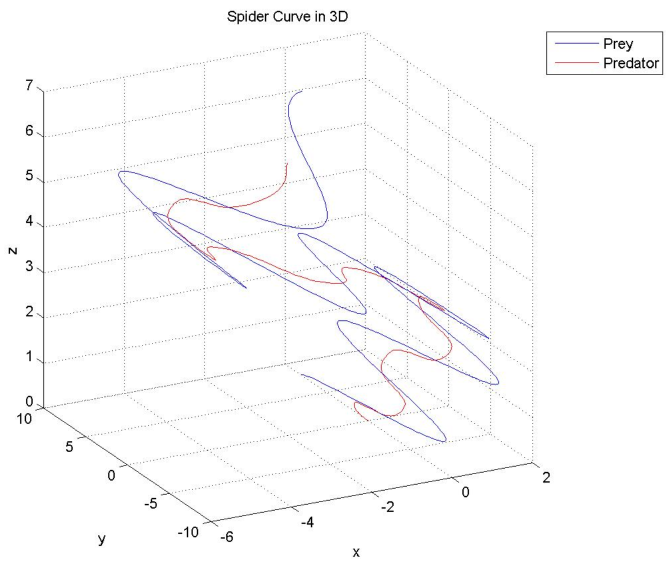
$$p(t) = \cos t$$

$$q(t) = \sin 3t$$
$$r(t) = t$$
$$\frac{dp}{dt} = -\sin t$$
$$\frac{dq}{dt} = 3\cos 3t$$
$$\frac{dr}{dt} = 1$$

Solving and graphing in Matlab produces the following graph:



Solving and graphing our "Spider" equation using the same r equation produces the following graph:



6 A Pursuit Curve on a 3D Surface

Graphing pursuit curves in 3D opens up the possibility of graphing a pursuit curve on a 3D surface. In order to do this the surface must be defined as a function of x and y. That way the pursuit curve will be directly graphed on the surface. In this example, the lion verses gazelle model will be used. The lion is pursuing the gazelle at 30% of the gazelle's speed and the surface equation is $Z = 20 \sin x \cos y$.

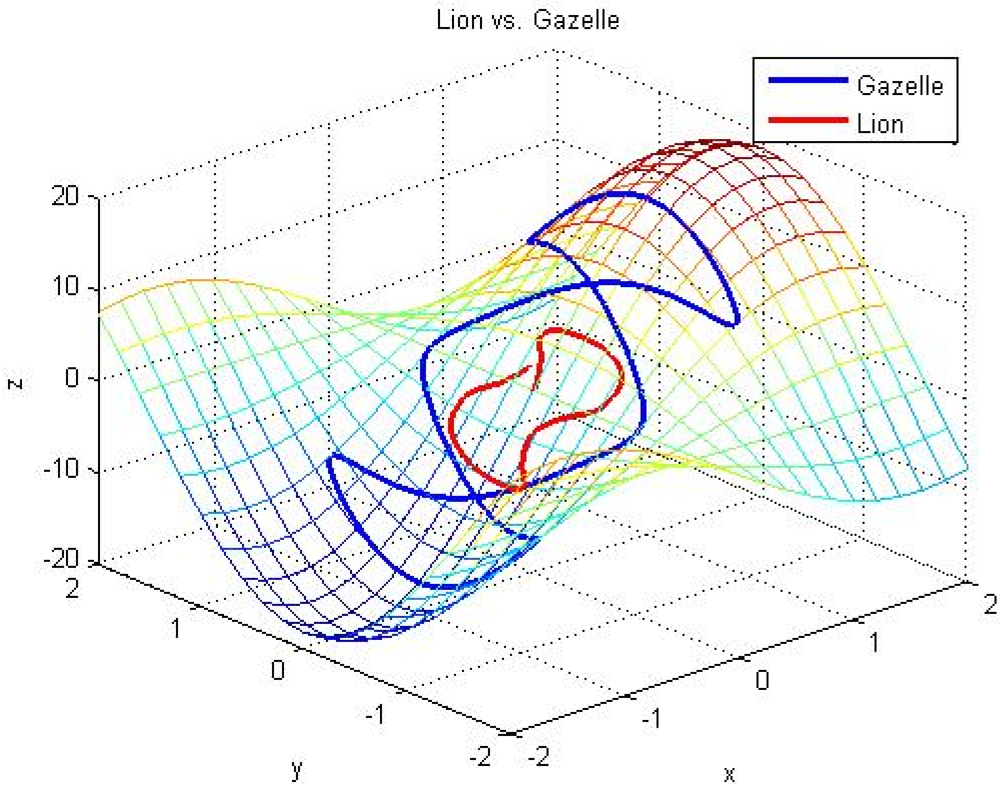
Our equations are as follows:

$$p(t) = \cos t$$

$$q(t) = \sin 3t$$

$$r(t) = 20 \sin x \cos y$$

Solving and graphing in Matlab produces the following graph:



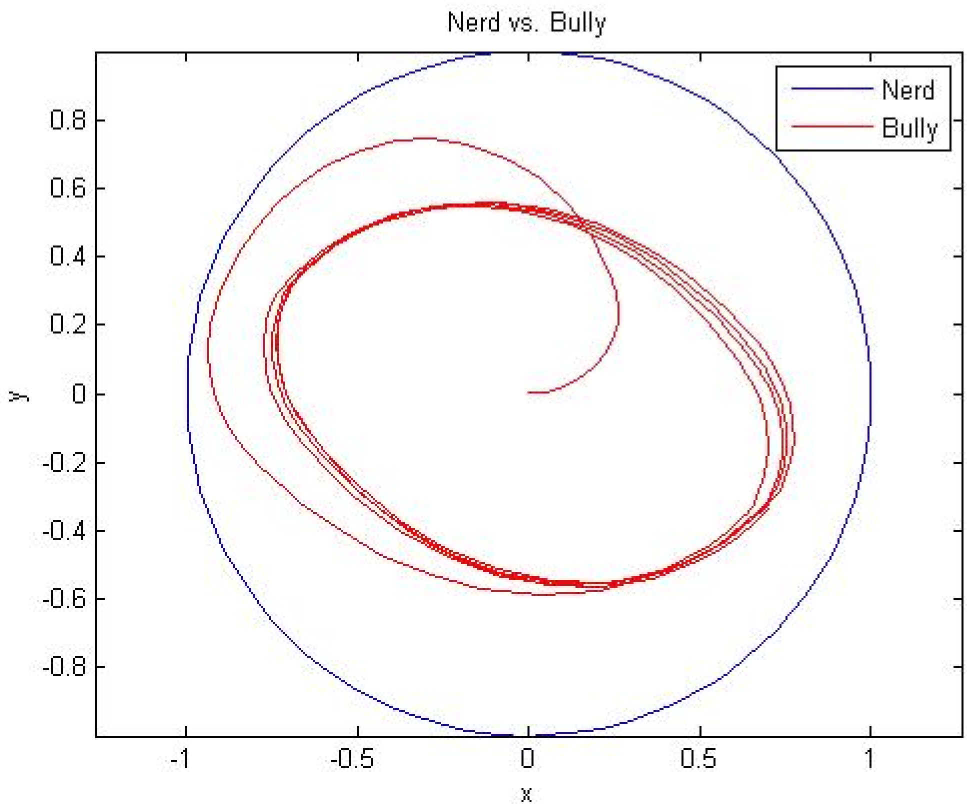
7 Varying Speeds

Another way to change up the pursuit curve equations is to vary the speed at which the predator chases the prey.

7.1 Bully and Nerd

The first example will be variation of a 2-Dimensional circular pursuit curve. Because our coordinate plane is divided into four quadrants, the velocity will vary depending on which quadrant the predator is in. For quadrants I, II, III, and IV, the predator will be chasing the prey at 30%, 100%, 30%, 100% respectively. This is accomplished in Matlab by defining k to equal .3, 1, .3, 1 for the specified quadrants.

Solving and graphing in Matlab produces the following graph:

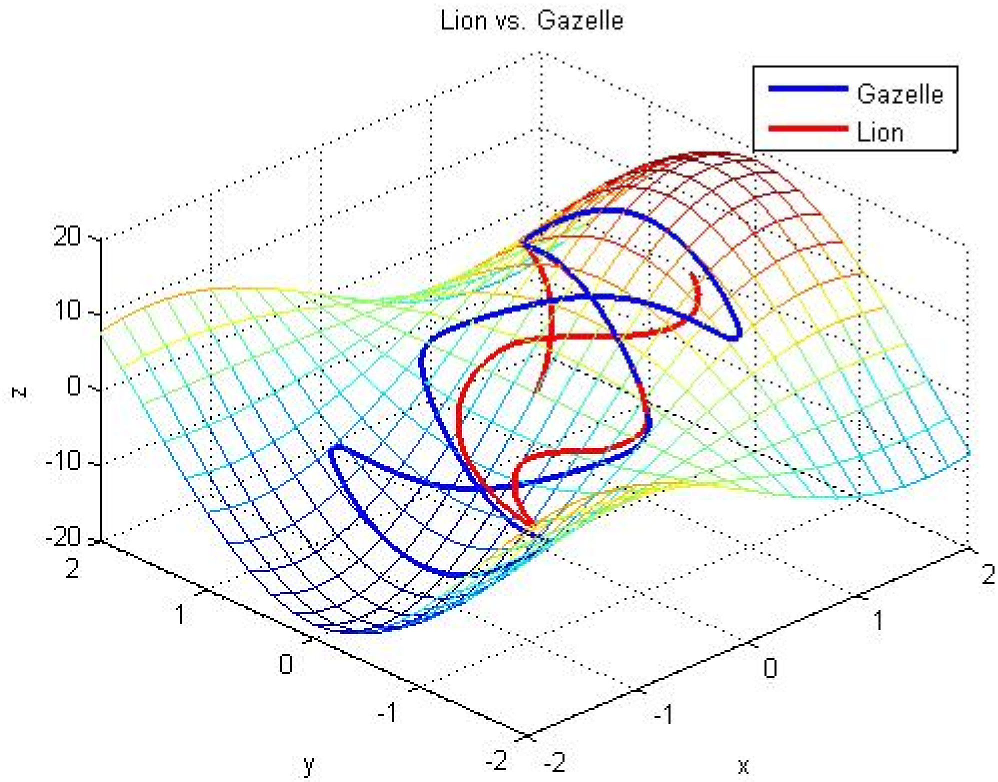


This gives us a path that is a smashed circle. In the end, as with the basic circular pursuit curve, the bully ends up following the same path infinitely.

7.2 Lion and Gazelle

Going on to vary the velocity for a 3-Dimensional curve follows the same principle. The velocity variable, k, will be defined for certain values.

Solving and graphing in Matlab produces the following graph:



Although the graph shows that the path of the two cross we can't actually say that the lion catches the gazelle. Just because their paths cross doesn't mean that they were there at the same time. In order to determine this we would need to be able to plot two comet plots on the same graph at the same time. Unfortunately our knowledge of Matlab is insufficient to allow us to do that.

8 Conclusion

In real life pursuit curves are used to model the path that a predator follows prey. This doesn't limit the applications to simply animals, but can be used in any instance where one object is following another, such as our idea of a missile following a ship. In a more realistic scenario the prey would not just follow a set path, but would change its course depending on where the predator was. There would also be dead zones where neither of the objects could travels, such as a tree being in the way, which would change the path. This doesn't degrade their usefulness though seeing as the can be and are used to model guided missile.

9 M-Files

These are the general codes we used from which we derived all the others.

General 2-D codes-

1)

```
function Yprime=general(t,y,flag,k)
Yprime = zeros(2,1);
P=t*sin(t);
Q=8*cos(6*t);
dP=t*cos(t)+sin(t);
dQ=-48*sin(6*t);
Yprime(1)=k*(sqrt((dP)^2+(dQ)^2)*(P-y(1)))/(sqrt((P-y(1))^2+(Q-y(2))^2));
Yprime(2)=k*(sqrt((dP)^2+(dQ)^2)*(Q-y(2)))/(sqrt((P-y(1))^2+(Q-y(2))^2));
```

2)

```
t=linspace(0,8*pi,500);
x=t.*sin(t);
y=8*cos(6*t);
[t,Y]=ode45('general',[0,8*pi],[0;0],[],0.75);
h=plot(x,y,'b',Y(:,1),Y(:,2),'r');
legend(h, 'Prey','Predator');
title('Spider Pursuit Curve')
xlabel('x')
ylabel('y')
axis equal;
shg
```

General 3-D Codes-

1)

Adding these 3 lines adds a 3rd parameter to our original pursuit equations.

```
R=t;
dR=1;
Yprime(3)=k*(sqrt((dP)^2+(dQ)^2+(dR)^2)*(R-y(3)))/(sqrt((P-y(1))^2+(Q-y(2))^2+(R-y(3))^2));
```

2)

This code graphs the new 3rd parameter.

```
t=linspace(0,2*pi,500);
x=cos(t);
y=sin(3*t);
z=t;
[t,Y]=ode45('general3d',[0,2*pi],[0;0;0],[],0.3);
h=plot3(x,y,z,'b',Y(:,1),Y(:,2),Y(:,3),'r');
legend(h, 'Prey','Predator');
title('Predator vs. Prey 3D')
```

```
xlabel('x')
ylabel('y')
zlabel('z')
grid on
```

General 3-D Surface Codes
By adding these 4 lines to our previous code we can add a surface to our 3-D graphs.
Z changes depending on our z value in the previous codes.

```
hold on
[a,b] = meshgrid(-2:.2:2, -2:.2:2);
Z = exp(cos(a)+sin(b));
mesh(a,b,Z)
hidden off
```

General Varying Speed-

To get this we simply changed our k from a constant value and made it like this...

```
k=.3*(P<0)+.9*(P>0);
```

References

http://online.redwoods.cc.ca.us/instruct/darnold/deproj/Sp98/PeterG/index.htm
http://online.redwoods.cc.ca.us/instruct/darnold/deproj/sp08/mseverdia/presentation.pdf
http://online.redwoods.cc.ca.us/instruct/darnold/deproj/sp08/mseverdia/pursuit.pdf