PSBC Project 2: Cannonball Motion

Student ID: 9641876

April 19, 2018

1 Cannonball motion with air resistance

The problem posed here, is given a cannonball of mass m fired at initial velocity $450ms^{-1}$ and subject to an air resistance force of $F = K|\mathbf{v}|^2$ (where $K = 0.00002kgm^{-1}$), calculate the maximum horizontal distance it can travel and the angel at which it must be fired to achieve this. The very first step in questions such as this, is deriving ODEs to represent the movement of the object, in this case the cannonball.

Since both the vertical and horizontal directions must be considered, the position, velocity and acceleration will all be vectors. The position vector is

$$\mathbf{x} = (x, y).$$

And since $\mathbf{v} = \dot{\mathbf{x}}$ and $\mathbf{a} = \ddot{\mathbf{x}}$ the velocity and acceleration vectors are

$$\mathbf{v} = (\dot{x}, \dot{y}) \qquad \mathbf{a} = (\ddot{x}, \ddot{y}).$$

The magnitude of the air resistance force is $K|\mathbf{v}|^2$ as previously stated, and it moves in the opposite direction to v, so the direction vector is $-\frac{\mathbf{v}}{|\mathbf{v}|}$. Therefore the air resistance vector is

$$\mathbf{F} = -K|\mathbf{v}|^2 \frac{\mathbf{v}}{|\mathbf{v}|} = -K|\mathbf{v}|\mathbf{v}.$$

And finally the gravity vector is

$$\mathbf{g} = (0, -g)$$

Where $g = 9.8ms^{-2}$

So now, resolving horizontally and vertically and using Newton's second law we derive the ODEs for horizontal and vertical movement respectively:

$$m\ddot{x} = -K\dot{x}\sqrt{\dot{x} + \dot{y}}.$$
$$m\ddot{y} = -K\dot{y}\sqrt{\dot{x} + \dot{y}} - mg.$$

Now that the motion of the cannonball has been described in ODEs, the next step is to write a function in MATLAB to calculate how far horizontally the cannonball will travel at a given launch angle. This involves first solving the ODEs simultaneously with the initial conditions determined by the launch angle. and then finding the time at which the y-position of the cannonball is 0. In order to do this the function `fzero` is used. Evaluating the x-position of the cannonball at this found time will give an answer to how far it has travelled horizontally.

Below is the code needed to define such a function, `xdistance(k)`, where `k` is the launch angle.

```
function f = xdistance(k)

m = 6;
K = 0.00002;
g = 9.8;
z0= [0, 0, 450*cos(k), 450*sin(k)];

% z = [x y u v]

projectile_ode = @(t, z) [z(3); z(4); -(K/m)*z(3)*sqrt((z(3)
    ^2)+(z(4)^2)); -(K/m)*z(4)*sqrt((z(3)^2)+(z(4)^2)) - g];

sol = ode45(projectile_ode, [0 100], z0);

time = fzero(@(r)deval(sol, r, 2), [sol.x(2), sol.x(end)]);

f = deval(sol, time, 1);

end
```

The next task is to maximise the x-distance and find the launch angle at which the cannonball reaches this distance. Using `fminbnd` we can find the launch angle `ang` which will minimise the negative value of `xdistance`, thereby maximising `xdistance` itself. The maximum distance can then be found by making `ang` the value of `k` in `xdistance(k)`. This takes just two lines of code which are as follows.

```
ang = fminbnd(@(k)-xdistance(k), 0.0001, pi/2); % the optimal
    launch angle
dist = xdistance(ang); % the greatest horizontal distance the
    cannonball can travel
```

Which gives the results

```
ang =

   0.778422249698371


dist =

     1.961776750944602e+04
```

So to conclude, the cannonball can reach a maximum horizontal distance of $19617.77m$ when it is launched at an angle of 0.7784 radians above the horizontal.

2 Cannonball motion with moving inteceptors

In this problem, the cannonball is being fired with the aim of hitting a target at a horizontal distance of $15000m$ from the origin. However there are also interceptors at a horizontal distance of $12000m$ from the origin, each $1000m$ long and $1000m$ away from each other, moving upwards with a speed of $100ms^{-1}$. A new one is released every 20 seconds. The task is to find the launch angles at which the cannonball will hit the target and the launch times at which it can do so without hitting an interceptor.

Finding the launch angles simply requires considering the previously written function `xdistance(k)`, subtracting 15000 from it and finding the values of `k` at which this is equal to zero. Using `fzero` makes this very straightforward.

```
launchangle1 = fzero(@(k)xdistance(k)-15000, [0.0001, ang])
launchangle2 = fzero(@(k)xdistance(k)-15000, [ang, pi/2])
```

And the output is

```
launchangle1 =

   0.425365584660216


launchangle2 =

   1.133637131774581
```

In order to find the launch times necessary to avoid the interceptors, we must find the time the cannonball reaches $12000m$ horizontally, and also the height (ie. y-distance) it is at when it reaches $12000m$. This involves solving the ODEs at each of the two launch angles, using `fzero` and `deval` to find the time at which $x - 12000 = 0$, equivalent to $x = 12000$, and then using `deval` again to find the value of y at this time.

Substituting each of the launch angles for `k` in the following code achieves this.

```
z0 = [0, 0, 450*cos(k), 450*sin(k)];
projectile_ode = @(t, z) [z(3); z(4); -(K/m)*z(3)*sqrt((z(3)
    ^2)+(z(4)^2)); -(K/m)*z(4)*sqrt((z(3)^2)+(z(4)^2)) - g];

sol = ode45(projectile_ode, [0 100], z0);

intercept_time = fzero(@(r)deval(sol, r, 1)-12000, [sol.x(1),
     sol.x(end)]);

height = deval(sol, intercept_time, 2);
```

Finally, the easiest way to find the appropriate times is to find a time interval within which any launched cannonball will reach the target without being intercepted, and add multiples of 20 to this to find all such time intervals, since the interceptors must move $2000m$ to be in the same formation again, and their speed is $100ms^{-1}$, hence the time taken is 20 seconds.

To find the first time interval, notice that the cannonball will be able to pass unintercepted when its height at $12000m$ from the origin is less than the height of the bottom of one interceptor, and greater than the height of the top of the next interceptor. So we use $speed = \frac{distance}{time}$ to calculate a time interval for the cannonball passing through, and subtract the intercept time to find a time interval for the launch of the cannonball.

```
time_interval = [(height/100)-10, (height/100)] -
    intercept_time
```

In both cases this will initially give negative times, which of course cannot be launch time intervals, but simply adding multiples of 20 as previously justified will give the positive time intervals required. Using this methon finally gives these launch time intervals for each launch angle.

Launch angle (rad)	Launch time interval (sec)
0.4254	[0, 1.2805] and [11.2805, 21.2805] + 20n
1.1336	[0, 8.1711] and [18.1711, 28.1711] + 20n