

# Finding the range of the robot.

The robot will throw a ball which is attached to the end of an arm of length 0.25 m rotating around the origin. We want to experiment with the rotation speed of the robot to figure out what the maximum range and height of the throw are.

Our basic setup is that while on the arm, the ball follows the path

```
In[1]= xArm[t_] := (1 / 4) * Cos[k * t];  
yArm[t_] := (1 / 4) * Sin[k * t];
```

and once in the air, the ball follows the path (constant speed motion in x, constant acceleration motion in y):

```
In[3]= g = -9.8;
```

```
In[4]= xAir[t_] := a * t + b;  
yAir[t_] := (g / 2) * t^2 + c * t + d;
```

At the time of release ( $t_0$ ), the position and velocity of these two curves are supposed to agree. We can use this information to solve for a, b, c, and d.

```
In[6]= absol = Solve[{xArm[t0] == xAir[t0], D[xArm[t], t] == D[xAir[t], t] /. {t -> t0}}, {a, b}]
```

```
Out[6]= {{a -> -1/4 k Sin[k t0], b -> 1/4 (Cos[k t0] + k t0 Sin[k t0])}}
```

```
In[7]= cdsol = Solve[{yArm[t0] == yAir[t0], D[yArm[t], t] == D[yAir[t], t] /. {t -> t0}}, {c, d}]
```

```
Out[7]= {{c -> 9.8 t0 + 0.25 k Cos[k t0], d -> -4.9 t0^2 - 0.25 k t0 Cos[k t0] + 0.25 Sin[k t0]}}
```

We now use this information to write the path in the air very concretely in terms of  $k$  and  $t_0$  as  $x_{\text{Flight}}$  and  $y_{\text{Flight}}$ . These functions will describe the path of the ball as a function of  $t_0$  and  $k$ .

```
In[8]= xFlight[t_] := xAir[t] /. absol[[1]];  
yFlight[t_] := yAir[t] /. cdsol[[1]];
```

We can see that these formulae look pretty unpleasant:

```
In[10]= xFlight[t]  
yFlight[t]
```

```
Out[10]= -1/4 k t Sin[k t0] + 1/4 (Cos[k t0] + k t0 Sin[k t0])
```

```
Out[11]= -4.9 t^2 - 4.9 t0^2 - 0.25 k t0 Cos[k t0] + t (9.8 t0 + 0.25 k Cos[k t0]) + 0.25 Sin[k t0]
```

But they actually look a lot better when you write them in terms of  $(t - t_0)$ :

```
In[15]= Simplify[xFlight[t]]
```

```
Out[15]= 1/4 (Cos[k t0] + k (-t + t0) Sin[k t0])
```

The computer isn't smart enough to do this on its own for y, but we can write it out ourselves:

```
In[22]= yFlightNew[t_] := (g / 2) * (t - t0)^2 + (1 / 4) * k * (t - t0) * Cos[k t0] + (1 / 4) * Sin[k t0];
```

```
In[32]= yFlightNew[t]
```

```
Out[32]= -4.9 (t - t0)^2 + 1/4 k (t - t0) Cos[k t0] + 1/4 Sin[k t0]
```

We can check that this formula is the same as the old one by subtracting them and simplifying:

```
In[24]:= Simplify[yFlightNew[t] - yFlight[t]]
```

```
Out[24]= 0.
```

Now these equations are still fairly easy to work with using the computer! We want to now find the maximum height reached by the ball in terms of  $k$  and  $t_0$ . This is a classic max-min problem: we want to find the maximum of  $yFlight$  on the interval  $[t_0, \infty)$ . To do that, we need to differentiate  $yFlight$  and find the point where the derivative is zero.

```
In[25]:= yflightcriticalpt = t /. Solve[D[yFlight[t], t] == 0, t][[1]]
```

```
Out[25]= -0.102041 (0. - 9.8 t0 - 0.25 k Cos[k t0])
```

Now we have to be a little careful! This is the time at which the projectile reaches the maximum height IF this time is  $> t_0$ . We can write down this function in *Mathematica* using an “If” statement:

```
In[26]:= yMax[T0_, K_] := (If[yflightcriticalpt > T0, yFlight[yflightcriticalpt], yFlight[T0]] /. {t0 -> T0, k -> K}) /. {t0 -> T0, k -> K};
```

```
In[27]:= yMax[3, 10]
```

```
Out[27]= -0.239421
```

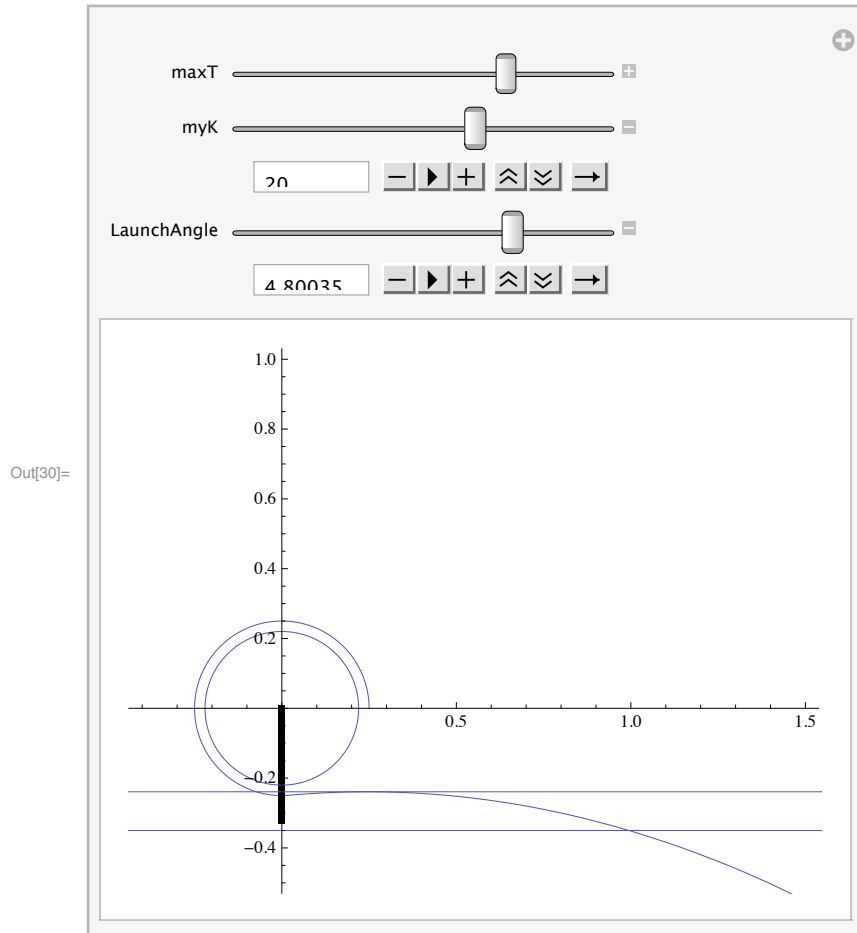
It is now time for some graphics, to make sure that all our formulas look like they are making sense:

```
In[28]:= ballPath[t_, T0_, K_] := If[t > T0, {xFlight[t], yFlight[t]}, {xArm[t], yArm[t]}] /. {t0 -> T0, k -> K};
```

```
In[29]:= Launcher[r_, theta_] := {Graphics[Rectangle[{-0.01, -1.5 r}, {0.01, 0.01}]], ParametricPlot[{r * Cos[u], r * Sin[u]}, {u, 0, 2 * Pi}]};
```

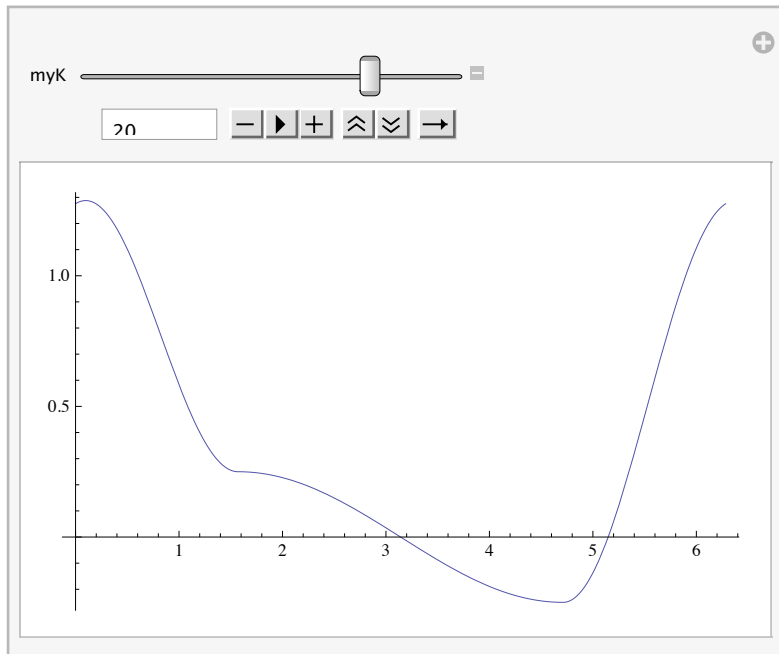
We can now see the ball and the launcher and the maximum height on a single plot:

```
In[30]:= Manipulate[
  Show[ParametricPlot[ballPath[t, LaunchAngle / myK, myK], {t, 0, maxT}], Launcher[0.22, t0],
  ParametricPlot[{t, -0.35}, {t, -2, 2}], Plot[yMax[LaunchAngle / myK, myK], {t, -2, 2}],
  PlotRange -> {{-0.4, 1.5}, {-0.5, 1}}, AxesOrigin -> {0, 0},
  {maxT, 0.01, 4}, {myK, 1, 30}, {LaunchAngle, 0, 2 * Pi}]
```



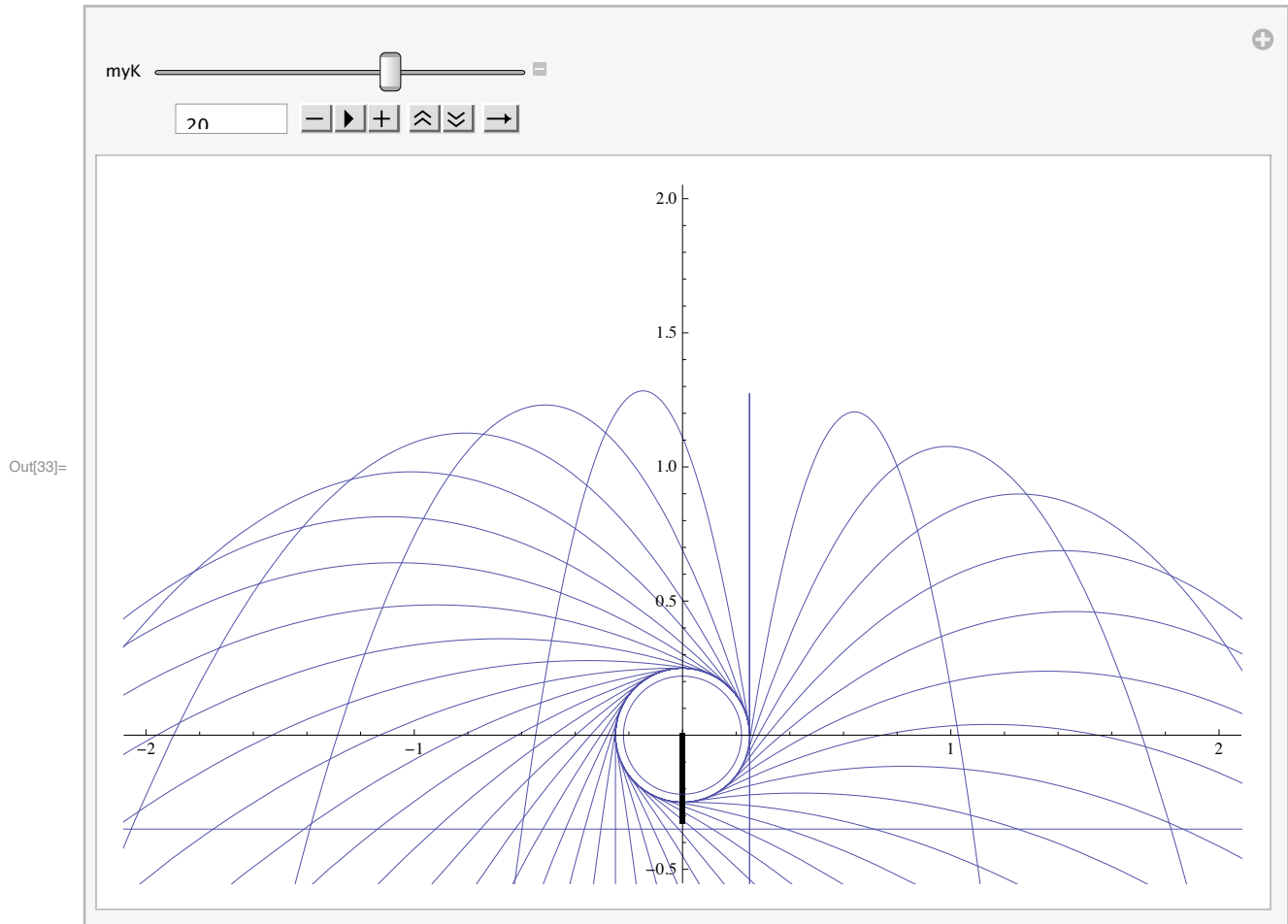
Let's get more abstract. Suppose we study the maximum value as a function of  $K$  alone. We'll plot the maximum height for each  $t_0$  value between 0 and  $2\pi/K$ .

```
Manipulate[Plot[yMax[LaunchAngle / myK, myK], {LaunchAngle, 0, 2 * Pi}], {myK, 1, 25}]
```



This is a really interesting function! We can see that the maximum occurs close to zero across the entire range. Now what is the region of the plane that we can target? The simplest way to answer that question is to plot all of the trajectories for a given K.

```
In[33]:= Manipulate[Show[Table[ParametricPlot[ballPath[t, LaunchAngle / myK, myK], {t, 0, 10}],
  {LaunchAngle, 0, 2 * Pi, 2 * Pi / 40}],
  Launcher[0.22, T0], ParametricPlot[{t, -0.35}, {t, -3, 3}],
  PlotRange -> {{-2, 2}, {-0.5, 2}}, AxesOrigin -> {0, 0}, ImageSize -> 600], {myK, 1, 30}]
```



Now we get to the group work problems for this week:

1. Find the  $t_0$  and  $t$  values which make the ball land on the table ( $y = -0.35$ ) at the location  $x = 1$  when  $k = 20$ . This means that you have to solve the system of equations:

$$\begin{aligned} x_{\text{Flight}}[t] &= 1 \\ y_{\text{Flight}}[t] &= -0.35 \end{aligned}$$

or

$$\frac{1}{4} (\cos[k t_0] + k (-t + t_0) \sin[k t_0]) = -1$$

$$-4.9 (t - t_0)^2 + \frac{1}{4} k (t - t_0) \cos[k t_0] + \frac{1}{4} \sin[k t_0] = -0.35$$

for  $t_0$  and  $t$ , given that  $k = 20$ .

2. Now suppose that our robot can launch the ball at this time  $t_0$  (accurate to within  $1/1000$  of a second). Estimate the resulting error in the  $x$  position of the ball when it hits the table. There are a few steps to this problem.

a. Write the  $x$  position of the ball when it hits the table in terms of  $t_0$  alone. You'll need to use the  $y_{\text{Flight}}$  equation

$$-4.9 (t - t_0)^2 + \frac{1}{4} k (t - t_0) \cos[k t_0] + \frac{1}{4} \sin[k t_0] = -0.35$$

to figure out **when** (at what value of  $t$ ) the ball hits the table (in terms of  $t_0$ ) and then plug that time into the  $x_{\text{Flight}}$  equation

$$x_{\text{Flight}}[t] = \frac{1}{4} (\cos[k t_0] + k (-t + t_0) \sin[k t_0])$$

in order to figure out **where** the ball hits the table. Your answer should be a function

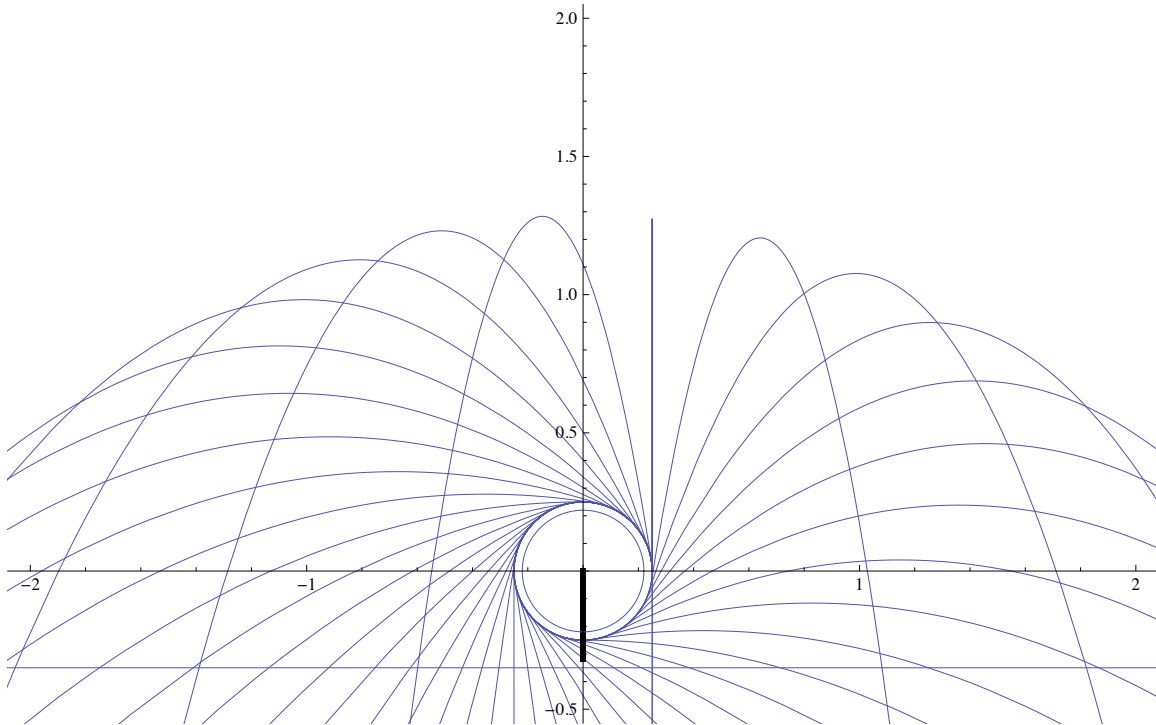
$$x_{\text{Strike}}[t_0]$$

b. Now do an error estimate for this new function  $x_{\text{Strike}}$  of  $t_0$ , using  $t_0$  as the variable, the value of  $t_0$  you got in part 1 for the point (that is, the  $x_0$  value in the examples we did in class) and  $1/1000$  for the error ( $\Delta t_0$ ). As we did before, you'll have to differentiate  $x_{\text{Strike}}[t_0]$  and evaluate at the given value of  $t_0$ .

c. What size basket can we hit with this accuracy? Suppose the error in the release time were  $1/100$  of a second because we bought cheaper parts. How would that affect our accuracy? Suppose the error was reduced to  $1/10,000$  of a second because we bought better parts? How would that affect our accuracy?

Extra Credit Problems.

I. I am fascinated by the question of the “envelope” curve in the picture



as a function of  $k$ . Is it really a parabola? If so, what's the formula for the parabola in terms of  $k$ ?

a. For a given  $x$  value, the height of this function  $f(x)$  is the largest  $y$  of any trajectory of the ball (that is, it is the largest  $y$  value for that  $x$  as we vary  $t_0$  and keep  $k$  fixed). Looking at the  $x$ Flight and  $y$ Flight functions again, we see that the question here is to solve

$$x\text{Flight}[t] = x \text{ or } \frac{1}{4} (\cos[k t_0] + k (-t + t_0) \sin[k t_0]) = x$$

for  $t$  (in terms of  $t_0$ ,  $k$ , and  $x$ ) and then plug that into  $y$ Flight[ $t$ ] or  $-4.9 (t - t_0)^2 + \frac{1}{4} k (t - t_0) \cos[k t_0] + \frac{1}{4} \sin[k t_0]$  to get a function  $y$ Flight[ $x$ ] in terms of the variables  $t_0$  and  $k$ .

Now our function is defined by

$$f[x] = \text{Max}_{t_0} y\text{Flight}[x]$$

We can get this by taking the derivative of  $y$ Flight[ $x$ ] **with respect to  $t_0$**  and finding the maximum value.

b. Once you get the  $f[x]$ , we have to ask whether it is quadratic in  $x$ . I have no idea whether this is true, but it might take some fairly serious algebra in order to prove it.

II. Suppose that in the classwork problem, there was error in the rotation speed  $k$  as well as in the time of release  $t_0$ . Supposing we release the ball at exactly the correct time  $t_0$  (calculated in part 1) by the rotation speed  $k$  varies by 1% from the assumed value of  $k = 20$ . Estimate the resulting error in the  $x$  position when the ball strikes the table.