

Is it Possible to Escape a Velociraptor?

Ryan Mikulec

Siena College, Loudonville, NY

(Dated: December 15, 2015)

Due to the underdevelopment in today's Jurassic cloning technology, we are unable to obtain an actual velociraptor to find out if us, humans, are able to escape one in an encounter. The next best thing is to run a computer simulation to test this unusual situation. The simulation is broken down into 4 parts, with a few assumptions. The first part is graphing the positions of the two subject. The second is finding out where and when the raptor will catch up. The third is where an how far will the raptor and human have traveled when they are one meter apart. The final part involves finding the probability that a human will escape this dangerous encounter. For the entirety of the code we will be assuming that the raptor and the human both accelerate instantaneously. Upon running the code it was found that it would take around 2 seconds for the velociraptor to travel 6 meters to catch up to the human, assuming the human got a 30 meter head start. At around 1.94 seconds the raptor would have travel almost 35 meters, placing it 1 meter behind the human. When it is 1 meter behind, it was found, using a Monte Carlo simulation, that there would be a 63 percent chance of escaping.

I. GRAPHING THE POSITIONS

The purpose of this section was to display a position versus time graph containing both the velociraptor and the human. Below is the code that was used.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

n = 1000
times = x=np.linspace(0,10,n)
human_positions = 30+3*times
raptor_positions = 18*times

plt.xlabel("Time (seconds)")
plt.ylabel("Position (meters)")
plt.plot(times,human_positions,'b-',markersize=6,label='Human')
plt.plot(times,raptor_positions,'r-',markersize=6,label='Raptor')
plt.title("Human vs. Raptor")
plt.legend(loc='lower right')
```

Code Section 1: This is the code used to obtain the graphs.

In order the graph the position, the first part was to derive an equation for both the raptor and the human. By starting with the distance kinematics equation:

$$x_1 = x_0 + v_0t + \frac{1}{2}at^2$$

Because we are assuming they accelerating to their max velocities instantaneously acceleration is equal to 0, and by plugging in their velocities and starting positions, the two linear equations come out to be:

$$RaptorPosition = (18\frac{m}{s}) * t$$

$$HumanPosition = 30m + (3\frac{m}{s}) * t$$

In order to graph this, there has to be a corresponding set of time values to plug into these equations. That is what the function linspace is for. Linspace, imported from numpy, is used to create a list numbers with a given number of evenly spaced intervals between two numbers. In this case, a list called times was created, consisting of numbers between zero and ten, with n number of points. By using the times in the equations for positions, both human_positions and raptor_positions become lists that is then used in the plot function, imported on the top, to graph the values. The rest of the lines of code use functions part of plot to add a title, labels, and a legend. Below is the graph that was produced

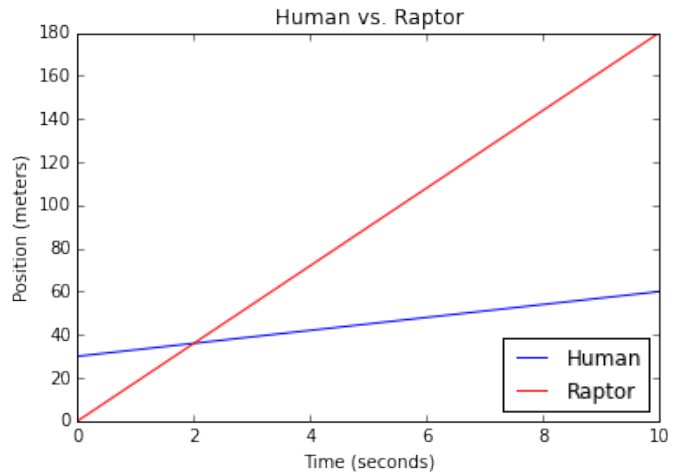


Figure 1: Graph of the Human and Raptor positions

II. FINDING WHERE THEY MEET

The purpose of the section is to use the results from part 1, to determine where and the the raptor and human will be at the same position. Below is the code that was used to do so.

```
for i in range(0, n):
    if(int(human_positions[i])!=int(raptor_positions[i])):
        time = x[i]
        position = human_positions[i]
print "At %i seconds, the raptor and human will both be at %i meters"%(time, position)
```

Code Section 2: Used to determine equal position and time

The main feature of this code is the for statement. This for statement runs n+1 amount of times, each time changing the integer i to the corresponding value of the trial its on. It is set to n, because n was used in part 1 to determine the number of values in the lists. By using this, the for statement is running to the number of values in the lists from part 1. Inside the for loop is an if statement that is checking to see if the value of human_positions at i is equal to the value of raptor_positions at i. If they are equal, it will continue to set a new variable called position to the value in human_position, and set another new variable called time to the value in the times list at i. In summary, this loop runs through all the values in the human positions list, and the corresponding values in the raptor positions list, checking to see if they are equal, then recording them if they are. The values of the positions were turned into integers because due to linspace, there may not have existed a value in which this statement is true. By forcing them to be integers, it guaranteed that there is. It was found that after 2 seconds, both the positions are 36 meters.

III. 1 METER AWAY

The purpose of this section is to find when, and how far the raptor and human are when they are one meter apart, then graphing these two points. Below is the code that was used:

```
for i in range(0, n):
    if(int(human_positions[i])-1==int(raptor_positions[i])):
        distance_close_raptor = raptor_positions[i]
        distance_close_human = human_positions[i]
        time_close = x[i]

plt.plot(times, human_positions, 'b-', markersize=6, label='Human')
plt.plot(times, raptor_positions, 'r-', markersize=6, label='Raptor')
plt.plot(time_close, distance_close_raptor, 'r*', markersize=12, label='Raptor: 1 meter')
plt.plot(time_close, distance_close_human, 'b*', markersize=12, label='Human: 1 meter')
plt.title("1 meter away")
plt.legend(loc='lower right')
plt.xlabel("Time")
plt.ylabel("Position")
plt.xlim(1.8, 2.3)
plt.ylim(30, 40)
```

Code Section 3: Used to find distance 1 meter away

A very similar approach to part 2 was used to accomplish this with one key difference. Instead of the if statement checking to see if they are equal, it is checking the see where the human position minus the raptor position is equal to one. This was found with the process below:

$$RaptorPosition + 1 = HumanPosition$$

$$HumanPosition - RaptorPosition = 1$$

These new values where then stored in the new variables, distance_close_raptor, distance_close_human, and time_close. The final part was to produce the graph. The imported plot function was used once again to create a graph very similar to that in part 1. Then the plot function was used again to plot the points that were obtained using the new close variables. The final step was adjusting the limits in order to show the relationship better. After around 1.94 seconds, the raptor travels around 34.8 meters, and the human travels around 5.8 meters from its initial 30, making them 1 meter apart. Below is the graph that was produced:

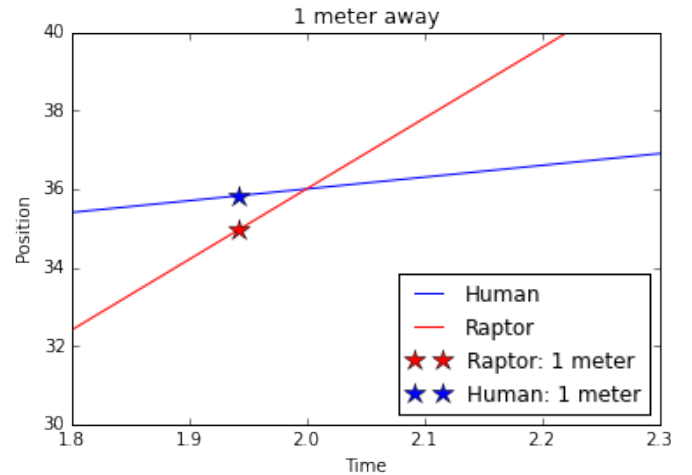


Figure 2: Graph of the Human and Raptor positions when 1 meter apart

IV. WHAT ARE THE CHANCES?

The purpose of this final section is to carry out a basic simulation to determine a probability given a few assumptions. The first time the raptor tries to bite, there is a 20% chance it will bite you. If it misses and tries a second time, there is only a 15% chance, and if it misses that time, there is only a 7% chance on the third try. The last assumption, based on common raptor behavior, is that if it misses that it will give up, and you will escape.

The code below uses a Monte Carlo process to find the probability: The purpose of a Monte Carlo simulation is

```

tries = 1000000
prob = 0
for i in range(0,tries):
    first = np.random.random()
    if(first>.2):
        second = np.random.random()
        if(second>.15):
            third = np.random.random()
            if(third>.07):
                prob+=1
prob = (float(prob)/tries) * 100
print "The probability of you escaping is %s percent" %(prob)

```

Code Section 4: Monte Carlo simulation

to create a large amount of random numbers in order to obtain a simulated probability. This is done using three nested if statements inside a for loop that is executed an amount of times equal to the variable tries. Before the loop is a variable called prob then is set to be equal to 0, this is the counter. The first statement inside the for loop generates a random float between 0 and 1. The first if statement sees if this float is greater than the probability of the raptor biting you on its first try, which is .20. If it is greater then it continues to generate another random number and runs it through the second if statement, which sees if its greater than the second probability, which is .15. If this is true then it generates on last random number, throws it through a third if statement that sees if its greater than the third probability, which is .07. If this last statement is true then it will add one to the variable prob. By repeating this a tries number of times you are left with a ratio between prob, the number of times the third if statement was true, and tries, the number of times the loop was carried out. This ratio is the probability, and by multiplying it by 100, a percentage is obtained. The chance of you escaping based on this simulation came out to be around 63 %.