Physics 312: Homework #6

Nicholas Nelson

Figure 1: What happens when a Ford Focus and a Honda Ridgeline meet at 75 mph, courtesy National Highway Transportation Safety Board, 2007

Thanks to modern electronics, accelerometers are now easy to use, cheap to manufacture, very accurate, and quite reliable. This allows us to do things like use crash test dummies to measure the forces applied to various body parts in a car accident. For example, they are used to measure the accelerations (and hence the net forces) on a person's head when a 2006 Honda Ridgeline hits a 2002 Ford Focus head on at a closing speed 75 miles per hour. The accelerometer data for the first 0.35 s of the crash shown above can be found in the ASCII file `v05686txyz.txt` It contain the x, y, and z components of the acceleration of a crash test dummy's head during a crash text. The zeroth column of the file is the time in seconds and the first, second, and third are the x, y, and z components of acceleration in g's (1 g = 9.8 m/s^2). The time of initial contact between the two vehicles is defined at $t = 0$ in these data.

In principle, once you know an objects acceleration history you can also find its velocity and position histories via integration. In practice, however, this method suffers from accumulated error, and so is only suitable for short time integrations. The error on the acceleration measurements can be found by looking at the readings before $t = 0$ s, when the dummy's head should have been moving at constant velocity. If the velocity $v(t)$ is related to the acceleration $a(t)$ by

$$v(t_k) = \int_{t_0}^{t_k} a(t)\, dt \approx \sum_{i=0}^{k} a(t_i)\Delta t \tag{1}$$

then the error propagation for that integral is

$$\sigma_v(t_k) = \sigma_a \Delta t \, (k+1) \, , \tag{2}$$

assuming that Δt is known exactly.

The error for the position at time t_k, $\sigma_x(t_k)$, is then

$$\sigma_x(t_k) = \Delta t \sum_{i=0}^{k} \sigma_v(t_i) = \sigma_a \Delta t^2 \sum_{i=0}^{k} (k+1) \ . \tag{3}$$

You may find the functions `scipy.integrate.cumulative_trapezoid` and `numpy.cumsum` helpful.

In addition, numerical integration of discrete data also introduces errors. This error can be lessened if more data points are added, however for real data the number of points is often fixed (as it is here). When using the trapezoid rule you can estimate the integration error using the technique outlined in Section 5. Remember that you can decrease the number of points in your integral by ignoring every other data point.

Suppose that the position of the dummy's head was known at the time of the first acceleration measurement in the file and we call that point $(0, 0, 0)$ m, and further that we know that the dummy's head had a velocity at that time of $(16.7, 0, 0,)$ m/s, because that's how fast the car it was in was going. Write a program that reads in the accelerometer data for the crash test dummy's head and then calculates and plots the acceleration, velocity, and position for that crash test dummy's head and the uncertainty on each value from both the accelerometer and the numerical integration. Report your final velocity and position in the form of

$$\vec{x}_f = [x_f, y_f, z_f] \pm \sigma_x \pm \sigma_I \ , \tag{4}$$

where σ_x is given above, and σ_I is the error from the numerical integration. Create as many plots of whatever type as needed to effectively convey the information.