BMME 201 Summer 2021
PA8: Bouncing Ball
Due: SUNDAY, 6/20, at 8 pm

Objective

On the video available online, extract the centroid to plot the position/velocity/acceleration and potential/kinetic/total energy of the system as it bounces. Use these results to calculate its coefficient of restitution, or how much energy is lost with each bounce.

Theory

Recall from mechanics that collisions can be elastic or inelastic. If a ball is dropped from some height h_0, it can bounce back to its original height if the collision is perfectly elastic. No energy is lost as heat or noise or anything else. If it hits the ground and stops moving immediately, the all the energy is lost, and the collision is perfectly inelastic. In most cases, the collision is somewhere in between these. Define a parameter called the coefficient of restitution, *e*. For the possible collisions,

$e = 0$	Perfectly inelastic collision
$0 < e < 1$	Typical collision
$e = 1$	Perfectly elastic collision

In a 1D case, *e* of the bouncing ball is given by the ratio of the pre- and post-collision velocities:

$$e = \frac{velocity\ of\ ball\ away\ from\ floor}{velocity\ of\ ball\ towards\ floor}$$

It is possible for *e* to be less than 0 or greater than 1. In the former case, this might be if the ball traveled through the ground (e.g., bowling ball dropping through a glass pane). For the latter case, the ball might have some sort of explosive in it and rebounded higher than before. However, in this problem, *e* should be between 0 and 1.

The velocity change, as mentioned earlier, is due to a loss of energy in the system. Neglecting friction, the only point of energy loss is during a bounce. As the ball bounces, though, its kinetic and potential energies change. These are given by the following pair of equations:

$$PE = mgh$$
$$KE = \frac{1}{2}mv^2$$

where *m* is the mass of the ball, *g* is gravity, *h* is the height above the ground, and *v* is the velocity. The total energy of the system is the sum of the potential and kinetic energy. This should remain constant, but for experimental reasons, it changes a bit.

The potential energy is easy to relate to the coefficient of restitution:

$$e = \sqrt{\frac{max\ PE\ after\ bounce}{max\ PE\ before\ bounce}} = \sqrt{\frac{max\ height\ after\ bounce}{max\ height\ before\ bounce}}$$

This is more accurate in this experiment, since the velocity is inaccurate just before and just after the bounce.

BMME 201 Summer 2021
PA8: Bouncing Ball
Due: SUNDAY, 6/20, at 8 pm

Part 1

Download the program *bouncingBall_LASTNAME.m* and video. This program is complex and has many sections in it. After the declarations, the next section loads the video from the local storage to MATLAB. Some frames are unnecessary (before dropping the ball or after it is on the floor); you can use the variables `frameStart` and `frameStop` to ignore the unnecessary frames. You can access video file properties (like the frame rate) by `vid.Property`. Type vid and see the list of properties.

In the first section, use a `for` loop to process each frame individually by cropping out just the ball's path, thresholding for red, and then finding the centroid. Figure out a safe amount of the image to crop so that EVERY frame's ball is visible. Estimate the RGB thresholding criteria with Tools > Data Cursor, and tweak as necessary. Finally, make a figure with two subplots: the cropped binary image on the left and the centroid on the right. See the outline code for details. The centroid plot should show a solid line showing where it was and an "X" at the current location. The axes should not resize, so fix them. Finally, use `pbaspect` so that the image and plot are the same size.

The next section calculates the velocity and acceleration from the measured position of the centroid. Use the `gradient` or `diff` functions to take derivatives.

Using the position, plot the starting height and subsequent bounce heights as a function of bounce number using a stem plot (see `stem`). The command `findpeaks` may be useful here. Then, find the coefficient of restitution, which should be roughly constant throughout the video. Exclude any bounces where you may have insufficient data to get reliable results.

Lastly, plot the potential, kinetic, and total energy on a single set of axes as a function of time. Use the formulae described above in the "Theory" section. Finally, something short!

After running, your code should produce four separate plots:

1) Two horizontal subplots: left one is a binary image; right is a plot of the centroid position throughout the video. For the centroid plot, make the current position appear with some symbol, but also show all previous locations with a solid line.
2) Three vertical subplots: position, velocity, and acceleration versus time.
3) Stem plot of the bounce height versus bounce number INCLUDING the initial starting position.
4) Plot of potential, kinetic, and total energy as a function of time.

Your code should print to the command window the average coefficient of restitution.

Part 2

Now make your own video of a ball bouncing. I will have some in lab or you can bring your own. Find a high contrast background (white wall, blue recycling bin, etc.). Record a video with your phone, and you will repeat the same exercise here as you did in Part 1. You will have to change the following:

1) Pixel-to-meter conversion factor. Measure the ball diameter BEFORE you leave for this.
2) Mass of the ball. I will have a scale in lab to find the mass if you want to use your own balls.
3) Video frame rate and number of frames.
4) RGB upper/lower thresholds.

Update the ball property variables and your pixel-to-meter conversion factor. Then, make the same plots as in Part 1, and submit both your code for Part 2 and your video.

Note: Sometimes videos from an iPhone loaded on a PC are rotated. Use `imrotate` to fix.