Homework 2:

Introduction:

This pdf is meant to accompany the code and the csv files attached in the zip file to show my knowledge of the homework. I would like to point out that in my code I have done two things that make it work.

1. I have added another value to the spring_pos value so that I am able to multiply it by the transformation matrix. This value does not affect the end result since it is parsed out.
2. For my function, I have added another variable called "part" which acts as a state variable to run the same function in different ways depending on which part of the homework I am on. This should be commented under each if statement.
3. When using part 3, I feed a second variable gravity3 which is all 0.

Running the code, there is already a pre-populated call for the function to run part two which should output a positively damped system csv file called "some.csv".

Part 1:

Simulating a falling robot.

Part 1a:

The total energy of this robot should be constant because it is simply oscillating between kinetic and potential energy since there is no outside factor like springs, friction or damping to disturb the system. To model the total energy, we could consider this simply the kinetic energy and potential energy summed.

T = 5 seconds
Dt = 0.01 seconds

Part 1b:

When the time step is too coarse, the euler integration has to take steps that are too discontinuous and this lack of smooth evaluation allows the algorithm to add energy in the system since the potential energy and kinetic energy cannot change properly in tandem. This has been proven in ME 314 where we simulate the difference between Euler integration and RK4.

T = 5 seconds
Dt = 0.05 seconds

Part 2:

Damped system.

Part 2a:

In this part the damping constant is a positive integer where we see a nicely damped system. If the robot were to be damped with a huge value, then the robot falls very slowly and almost floats since it is hindered from changing its joint velocities.

Damping = 20 Nms/rad
T = 5 seconds
Dt = 0.01 seconds

In a coarser system, we see that the system gets a little bit choppy as the damping is based on the velocities of the joints, but because it takes large timesteps, the motion is not smooth.

T = 5 seconds
Dt = 0.05 seconds

Part 2b:

When the damping is set to negative, the system is considered to "lose negative energy" and as the velocity increases, the damping adds more. This means that the system runs away and goes crazy since it is constantly adding more and more energy to the system.

T = 5 seconds
Dt = 0.01 seconds

Part3:

Part 3a:

Here we see the robot's end-effector oscillates or "bounces" towards and away from the origin of the spring. With a nice spring stiffness of stiffness = 10 N/m we get the motion seen in the video. When we make the stiffness = 100 N/m the spring oscillates wildly and seems to become even more crazy. This is due to the fact that the additional energy of the spring is so high that it accumulates error from the Euler integration. Normally we would expect to see the total energy stay with the system because the energy shifts between kinetic and elastic potential energy (noting we have “turned off gravity” so there is no gravitational potential energy). This is functionally creating a gravitational field towards one point instead of the ground.

Note for this part I have kept damping at 0.
T = 10 seconds
Dt = 0.01 seconds

Part 3b:

For this part, I have added a positive damping constant. Now stiffness = 10 N/m and damping = 1 Nms/rad. In this video we see that the damping leeches the energy from the system until it stabilizes pointing towards the spring.

T = 10 seconds
Dt = 0.01 seconds