Lab 3: Motion Control

The Project: Motion Control

In order to assist NASA's future efforts for space exploration, Company X is seeking a thrust control algorithm to help a spacecraft land safely on an asteroid's surface. A NASA probe containing the spacecraft(s) will perform a fly-by maneuver close to the asteroid. At closest approach, the spacecraft will be ejected out the rear of the probe via a rail gun with a precise velocity to cancel all forward motion. At the moment the spacecraft leaves the probe, it will be at rest exactly 80 km above the asteroid's surface and will begin freely falling toward the asteroid.

The precise makeup of the asteroid is unknown, and hence so is its gravitational constant g. The first task in the mission is to determine g by ejecting a spacecraft into free fall and observing its motion. This sacrificial spacecraft will "crash land" on the asteroid's surface but provide valuable information about the asteroid's gravitational field for future missions. The second task is to release another spacecraft with a precisely-controlled constant thrust (applied force) opposing the gravitational acceleration of the asteroid. The goal in this step is to carefully choose a thrust value that results in the spacecraft touching down on the asteroid's surface with exactly zero velocity. Note that the spaceship will be in free fall (no thrust) for 32 km (in order to build up re-entry velocity.)

Prior to the real mission, your group has been tasked with running a simulated version of the mission and devising a method to 1) measure g for the asteroid; and 2) choose the appropriate thrust to ensure that the spacecraft reduces its velocity to be within a set range at a given point above the asteroid surface (manual thrusters can then be used by the astronaut.) Your colleagues in the Company X computing division have written a simulation of the mission which uses a CNC machine and attached model spacecraft to emulate the real mission. This simulation is a python code called ***lab.gravity_simulator***. To run the code with zero thrust applied to the spacecraft, you will need to type: ***python3 –m lab.gravity_simulator***

To run the simulation with an applied thrust of, for example, 100 N, type: ***python3 –m lab.gravity_simulator –-thrust 100***

The motion of this simulated spacecraft can be recorded and analyzed using the vision tracking software you are already familiar with (but with one new feature - see below).

In the simulation, a random value of g is chosen (and is different at every table). You are able to run the simulation either in "free fall" mode, or with a constant thrust applied. The simulation has been written such that the distance scale is 10^-5 times that of the real mission. Thus the simulated spacecraft will travel a total distance of **80 cm** from the launch point to the asteroid's surface. Any thrust you apply will be active from x = 32cm to x = 80 cm. The units of the simulated value of g should be taken as cm/s^2. The mass of your spacecraft will be provided by the simulation software at your table.

Your friends in the Company X computing division have also added a new feature to the vision control software: the ability to *automatically* track, calculate and record position, velocity, and acceleration in *real physical units*. The software will output a mp4 video and a csv text file with position, velocity and acceleration in units of cm, cm/s and cm/s2, respectively. Your instructor will explain the specifics of this new feature to you, including ways that you can incorporate it into your own custom software, and any pitfalls to be aware of.

Your team has now been hired to develop and implement a thrust control system for the spacecraft.

Lab 3: Motion Control

Equipment

- Camera
- CNC
- Model spacecraft
- Tracking stickers
- Meter Stick or Calipers
- Air hockey pucks

The Requirements:

Your friends in the Company X computing division have also added a new feature to the vision control software: the ability to automatically track, calculate and record position, velocity, and acceleration in real physical units. The software will output a mp4 video and a csv text file with position, velocity and acceleration in units of cm, cm/s and cm/s2, respectively. Your instructor will explain the specifics of this new feature to you, including ways that you can incorporate it into your own custom software, and any pitfalls to be aware of.

Next, to demonstrate that you can safely land a spacecraft on the asteroid's surface, your team is asked to use your vision tracking software and the simulation program to do the following:

1. Determine the value of **g** for the asteroid.
2. Determine an appropriate thrust value such that the simulated spacecraft has a velocity between 10 cm/s and 0 cm/s at the point 80 cm away from its initial launch point. This value should be determined by calculations based on the physics of the situation, not on trial and error.

Presentation of the solution:

Your team's two-to-three page report should contain the following information:

- A description of the basic physics principles used in your project, including equations used to calculate the appropriate thrust in your experiments.
- A determination of the value of g for your asteroid. Show example plots of kinematical quantities (position, velocity, acceleration) vs. time and discuss the method that you used to determine g.
- Plots of the position, velocity, and acceleration in the x- and y-directions as functions of time for each of your landings exhibited in the previous point.
- Discuss any sources of error or variance that you had to overcome during the course of the project.

In addition to your report, you should also save a set of example videos of landing approaches with various thrusts. This should include at least one successful landing (where v = 0 at the asteroid surface), and one unsuccessful landing. You should be prepared to show these to your instructor, if requested.