ME 449 Assignment 3 - Christopher Luey

Introduction

We simulate a UR5 robot arm to behave under gravity, with joint damping, and include a spring that the end effector is pulled by. This idea is implemented in the Puppet function which is based on the ForwardDynamicsTrajectory method in the Modern Robotics library.

The code is created in Google Colab. Simply "run all" and the CSV files are produced which can be downloaded and imported into CoppeliaSim. I also downloaded the repo and added it to a repo called code. It takes about 5 minutes to run everything.

https://colab.research.google.com/drive/1juZDbFiP9POTFIQ8zcIy0SYitP9RTKQX?usp=sharing

Part One

In part one, the robot is accelerated by gravity (9.81 m/s^2) in the -z direction. Damping and spring stiffness are 0.

a) t = 5, dt = 0.001 The simulation runs as expected for 5 seconds under this condition. The robot arm swings about like a pendulum.
b) t = 5, dt = 0.01. There is an accumulation of error which causes the robot to behave strangely throughout. The robot arm begins to swing faster and loops around in a way that would be physically infeasible if left swinging like in 1a. This strange behavior is due to the increase of total amount of energy due to integration error over time.

Explain how you would calculate the total energy of the robot at each timestep, if you wanted to plot the total energy to confirm that your simulation approximately preserves it.

The total energy is the sum of the kinetic and potential energies for all of the robot bodies.

Kinetic Energy: Sum of the angular kinetic energies and linear kinetic energies. For the angular kinetic energy use the square of the joint velocities multiplied by half the rotational inertia of the body. For the linear energy, use the joint velocity and the geometric lengths of arms to determine the x, y, z linear velocities. Use the square of these linear velocities multiplied by half the mass of each body to calculate the total linear kinetic energy of each body. Sum the linear and angular kinetic energies for all the bodies to calculate the total kinetic energy of the robot. I described the process used to calculate the kinetic energy component of the lagrangian. The masses and inertias can be computed from the G spatial inertia matrices. Transformations between bodies may also be used as well as M and B matrices.

Potential Energy: Use the joint angles, geometric lengths of arms, and masses of each body to calculate the height of each body and thus its potential energy due to gravity. The masses, lengths can be found in the G and M matrices. Sum all the potential energies of the bodies and add to total kinetic energy to calculate the total energy of the system.

Part Two

We add damping which affects the joint torques. The force caused by damping is proportional to the velocity and the damping coefficient. For both situations: t = 5, dt = 0.01, gravity is still active (-9.81 -z), and the spring (= 0) is not.

a) damping = 0.6. The robot loses energy over time as the oscillations converge.
b) damping = -0.001. The robot starts to swing higher over time as the oscillations grow. There may be an accumulation of error at the end.

Do you see any strange behavior in the simulation if you choose the damping constant to be a large positive value? Can you explain it? How would this behavior change if you chose shorter simulation timesteps?

When very large damping is used, the resistive damping force is more than that caused by gravity causing the joint angles to increase significantly and spiral out of control. The same occurs for negative damping that is large.

There is a feedback loop due to large jumps in joint angles causing large joint velocities, and thus causing large torques because torque is a function of joint velocity and damping values. A larger torque leads to a larger joint velocity.

A shorter timestep would help delay the onset of the chaotic behavior however, the accumulation of error overtime would lead to the same long term behavior.

Part Three

We simulate a spring connected at (1,1,1) in the {s} frame to the end effector without gravity. t = 10, dt = 0.01.

a) k = 4, damping = 0. The spring oscillates as expected around the spring point.
b) k=4, damping = 2. The spring oscillates but arrives nearly to rest by the end of the video.

Considering the system's total energy, does the motion of the robot make sense? What do you expect to happen to the total energy over time?

The first video makes sense because the robot appears to oscillate about the spring position. The total energy for a system without damping theoretically stays the same. However, numerically, an accumulation of error would occur causing the total energy to increase over time due to the robot overshooting its joint angles, then the force from the spring on the end effector increases, leading to this feedback loop, increasing the total energy. The second video makes sense because the damping causes the robot to oscillate but arrive at rest.

Describe the strange behavior you see if you choose the spring constant to be large; if you don't see any strange behavior, explain why.

When the spring constant is large, the spring pulls with more force to its restorative position causing more joint displacement and high joint accelerations. This increases the rate of oscillations. The behavior isn't strange because all movements appear as expected. There is strange behavior if the simulation is long because of the accumulation of error due to inaccuracies in the EulerStep method.

Extra Note: There may be some lag in the videos because my computer cannot handle screen recording and simulation simultaneously that well so the FPS noted at the top may drop.