CS225a Autumn 2014 Homework #1 Solution

Tutorial #2

Q1: Study effect of integration time step

The basic (tao) integrator used in this example is not very accurate. It is a forward Euler integrator. As such, the integrator's accuracy relies on the timestep dt. Change dt to span a range from 1 microsecond to 10 milliseconds. Plot the energy change as a function of timestep and integration length.

Simulation time is chosen to be 1s, 2s, 4s, 8s, 16s, and 32s. The time step spans from 1us to 10ms. Figure 1 shows the percentage change in energy. As is shown obviously, the percentage change in energy generally increases with the time step and simulation length.

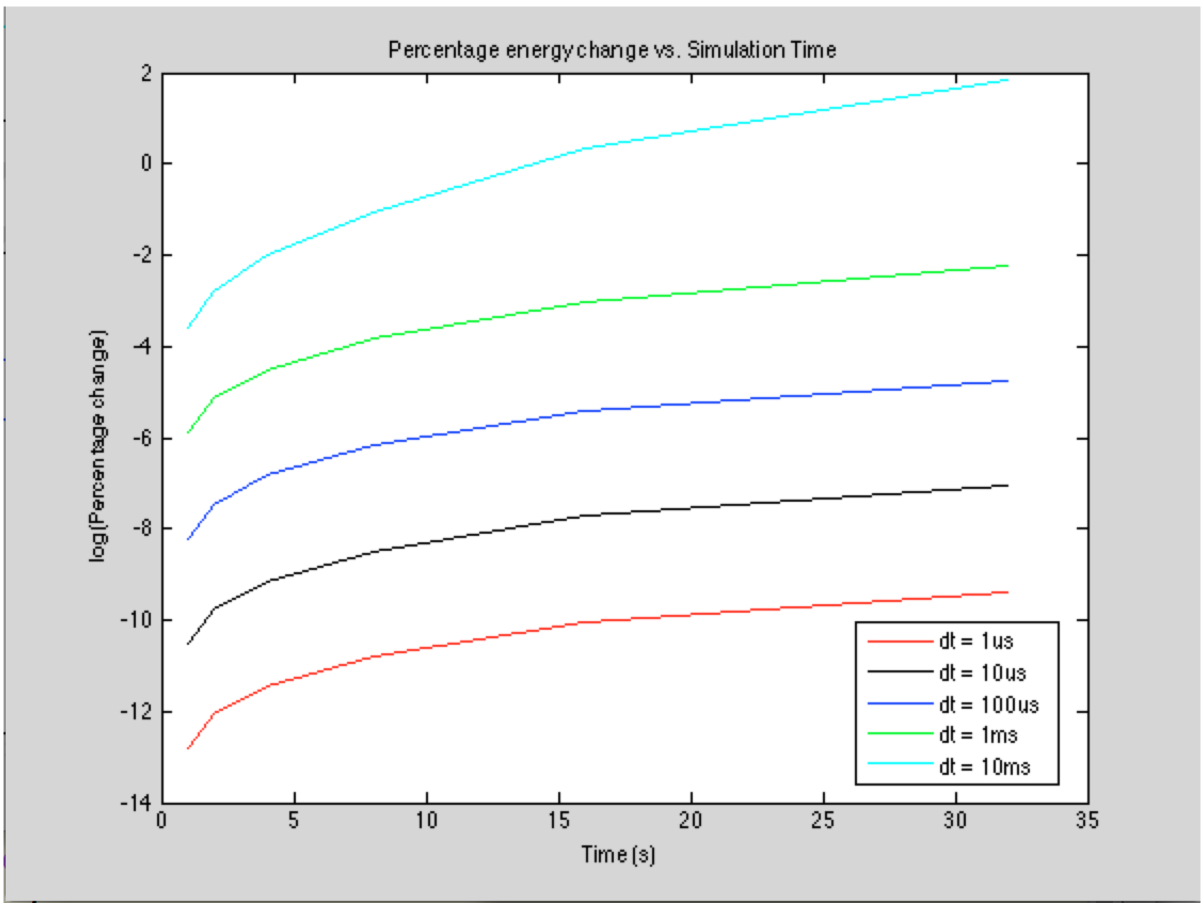

Figure 1: Energy change as a function of time step and integration length

Tutorial #2

Q2: Who do integration errors arise?

You should now be familiar with the tradeoff between integration parameters. What potential problems could arise if you test your controller with a physically inaccurate simulation environment?

HINT : Many existing physics simulators add damping to stabilize the integration process. How would this affect your control tests?

Testing your controller with a physically inaccurate simulation environment could possibly leave you with a wrong impression of how the robot would actually react to the same controller in real world. A stable controller that works in the simulation environment may become oscillatory or unstable for the real-world robot, and vice versa. For example, as the hint points out, many existing simulators add damping to stabilize the integration process. This is because the numerical integration process over a finite period of time tends to introduce errors and appears to inject energy into a conservative system as we find in the previous problem. On one hand, if we were to apply a fine-tuned controller in the simulation to a real-world robot, which has less damping than that implemented in the simulation, we could end up having a larger overshoot in the oscillatory behavior or even going into instability. On the other hand, if the real-world robot has a larger damping factor due to high resistance in joints, our fine-tuned controller would not have enough gain to generate adequate torque command, and therefore unable to meet control specs.

Tutorial #3

Q1: Integrating physics for complex robots

On completing the assignment in the previous tutorial, you probably found that the integrator starts to drift for some settings. The simple robot, however, doesn't properly test the physics integrator's abilities. Instead, re-run the analysis while adding more degrees of freedom to the robot specification. You must re-run the analysis for at least 5 different degree-of-freedom specifications. i.e., say robots like RRR, RRRR, RRRRPR, R6, R4P2R3.

Append more degrees of freedom by adding more rigid bodies to the xml file. Make sure to add them to the end of the chain. i.e., specify the parent link to be the leaf link in the current chain. Also make sure to give each link and each joint a unique name. You may make your life easy by writing some matlab/python/js code to print out the appropriate xml spec.

For each robot, find settings that limit the energy error to less than 1% across a 10second long simulation.

Here we analysis an RRRR robot for example. Other configurations can be done similarly. Figure 2 the graphic model moving freely only under the gravity.

Having re-run the analysis with time steps being 100us and 10ms, we get the following Figure 3. Apparently, 10ms is not a good choice as the percentage change in energy far exceeds 1% even for the shortest simulation time, 1s. 100us is acceptable as the percentage changes are well within the spec.

Next, we compare the simulation error of RRRR robot to that of the previous one in Tutorial 2 in order to see how the complexity of the robot configuration would affect the integrator. According to the Figure 4, increasing the complexity of the robot configuration by appending more DOFs has increased the error of integrator when a reasonable time step (100us) is selected. Also notice that a improperly selected time step could lead to over 100% change in the total energy of the system, which does not make much in the real world.

Extra Credit : Enabling collisions in the simulation environment dramatically increases the energy error. Unfortunately, this remains a challenging research problem. Discuss ways in which you could mitigate the effect of such physics errors on your (future) controller tests.

This is a pretty open-ended question. Ideas such that, prioritizing tasks that avoid collision, creating more accurate geometric model of collision, and compensating the error induced by collision are worth exploring. Other ideas that make sense are generally accepted.

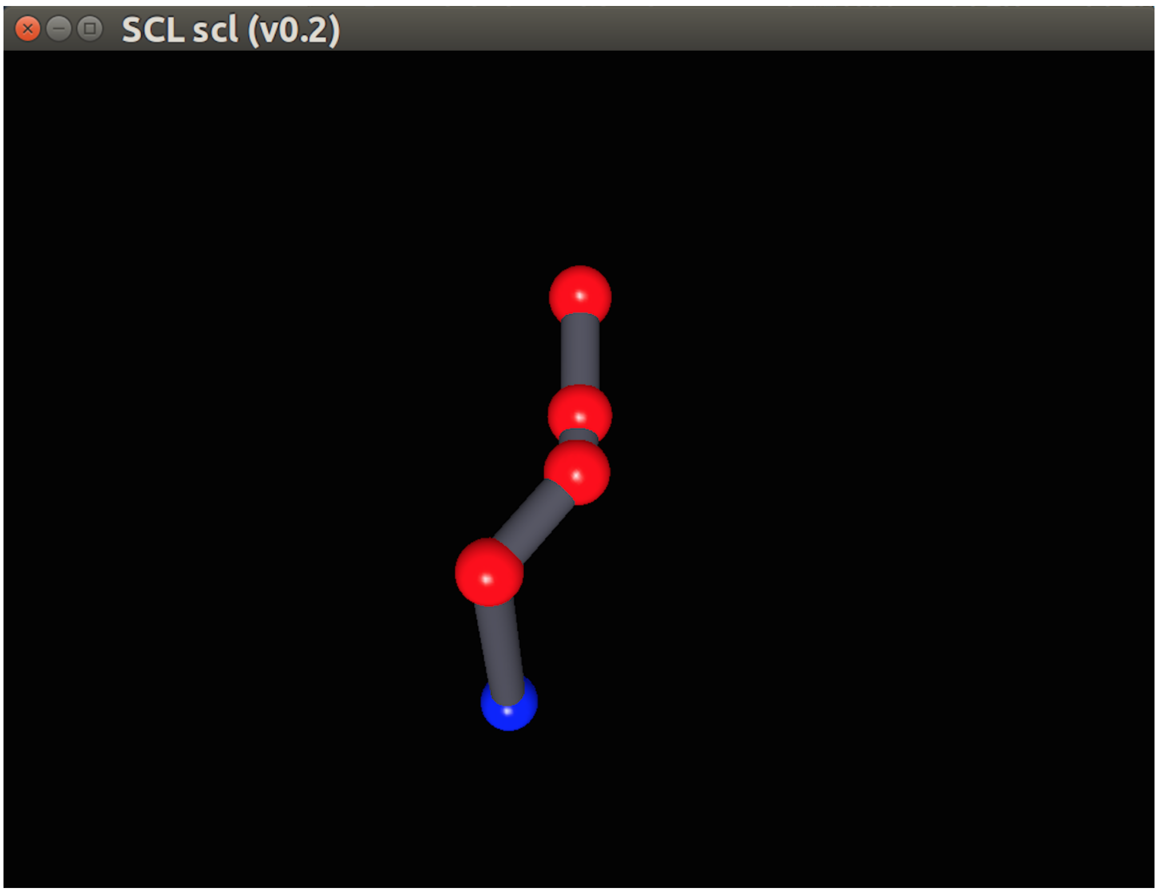

Figure 2: RRRR graphic model

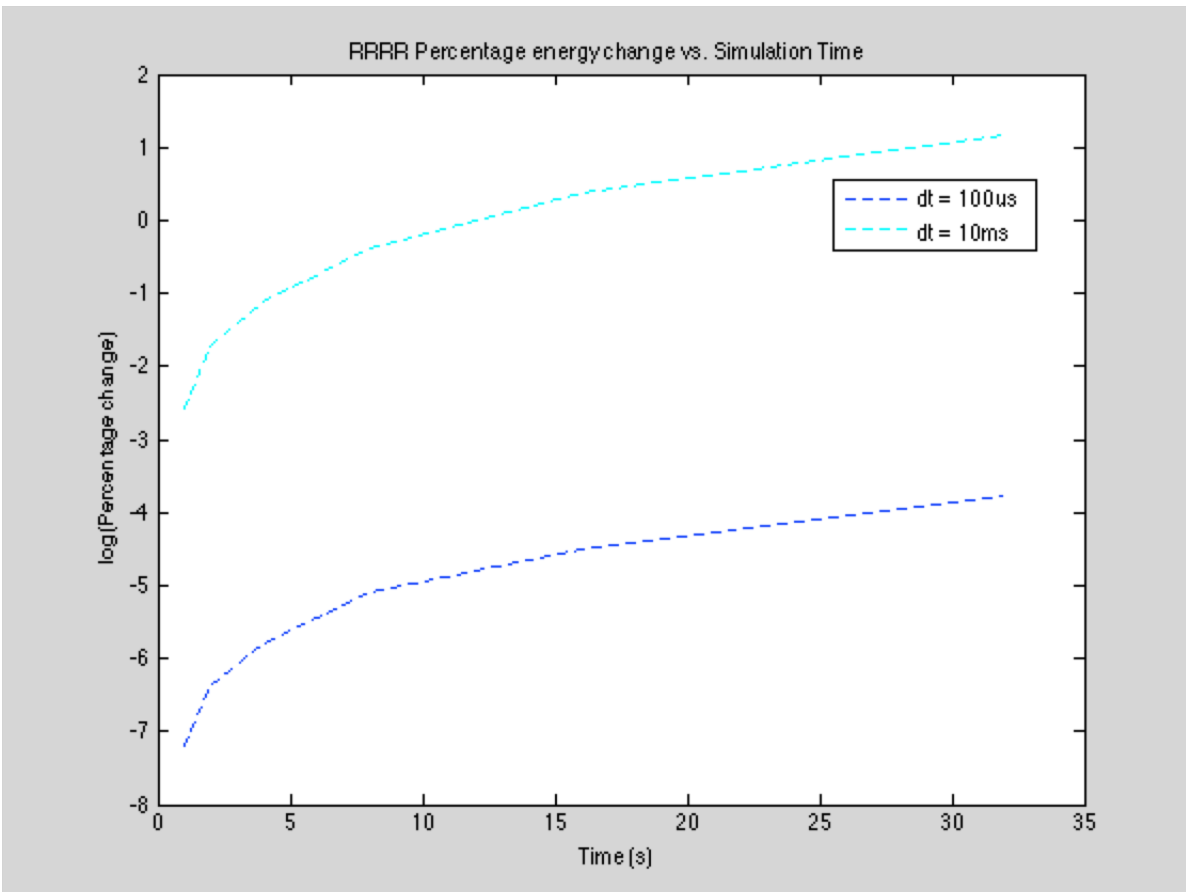

Figure 3: Energy change as a function integration length

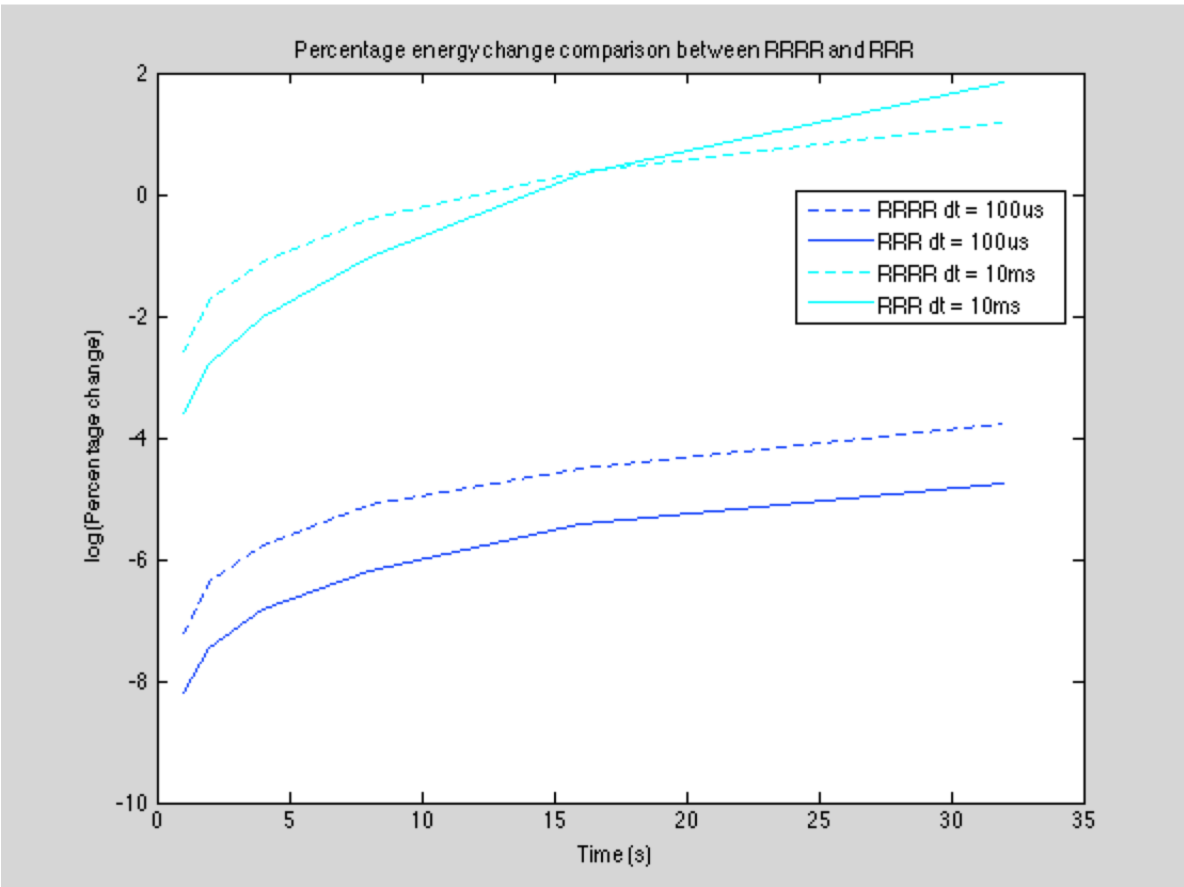

Figure 4: Comparison between RRR and RRRR