545 NW 26th St.
Unit 700
Miami, FL 33127

Software Assessment

The following is a problem representative of what one might encounter as a Software Engineer at Neocis.

Our system includes a serial robotic manipulator with a surgical instrument as its end-effector. The end-effector is able to **instantaneously** sense forces, to which the control software reacts by updating the desired position of the end-effector. This desired position is then executed by the hardware, to the best of its ability, using an unchangeable algorithm, resulting in an actual position that can be read from the robot's encoders. This process repeats every **Δt=1 ms**. For this problem, assume the control software is attempting to make the lightweight surgical instrument appear as a **m=10 kg** neutrally-buoyant object immersed at all times in a viscous fluid with a damping coefficient **b=30 N*s/m**, in 1D space:

$$\frac{d^2 z_{des}}{dt^2} = \frac{1}{m}\left(F_{sensor} - b\frac{dz_{des}}{dt}\right)$$

In discrete time with Euler Integration, this becomes:

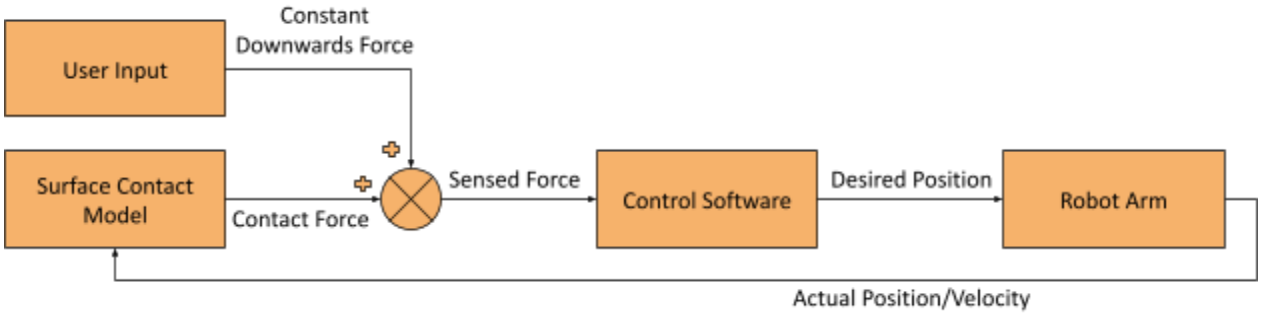
$$acc_{des}[n] = \frac{1}{m}\left(F_{sensor}[n] - b \cdot vel_{des}[n-1]\right)$$

$$vel_{des}[n] = vel_{des}[n-1] + acc_{des}[n] \cdot \Delta t$$

$$z_{des}[n] = z_{des}[n-1] + vel_{des}[n] \cdot \Delta t$$

Unfortunately, this implementation for updating the desired position of the end-effector gives rise to severe bouncing if the surgical instrument is pressed down into a springy surface by the user, as you can see from the plots at the end of this document and the attached real-life data and video.

In solving the problem, you may also refer to the following system diagram:



545 NW 26th St.
Unit 700
Miami, FL 33127

Tasks

1. If the surgical instrument were in reality the heavy object floating in a viscous fluid that we are trying to simulate with the robotic arm, pressing it into a springy surface would only cause it to bounce slightly and then settle. Write code to simulate the behavior of the real object being pressed into the surface, producing similar plots of force on the object and object position.

   Tip: For this task and the next, you will need to come up with a contact model for the surface, such that the sensed force approximates the real-life data seen below. It is not necessary to over-refine your model beyond one that qualitatively produces the bouncing seen in the real data. Quantitative correctness for your models is appreciated, but not required.

2. Write code to simulate the behavior of the robotic system from the diagram, producing plots of sensed force, desired position, and actual position of the surgical instrument. Assume the robot arm hardware only introduces a delay (i.e. actual position is a delayed version of desired position).
3. Explain in detail why the robotic system creates a behavior differing from that of a real object, and the mechanism behind the sustained bouncing. A mathematically quantified explanation will point you to the optimal solution for the next task.
4. Modify only the control software such that the robotic system is able to more accurately mimic the behavior of the real object, removing the sustained bouncing. You may use robot encoder information (actual position and/or actual velocity) as an input in addition to the sensed forces. Re-create the plots of sensed force, desired position, and actual position.

   Tips:

   - A good solution will result in plots similar to those you produced for task 1, showcasing that the robotic system more accurately mimics the dissipative behavior of the real object when pushed against the surface.
   - A good solution will remove bouncing, but not significantly affect the user experience in free space (such as making the instrument feel more sluggish than the real object would feel). As a sanity check, you could include in your code plots of velocities when accelerating in free space under a constant user force. Does the instrument still reach the same terminal velocity as the real object?
   - There exist solutions that do not explicitly try to detect surface contact or bouncing and treat them as a special case — these are preferred for robustness reasons.

545 NW 26th St.
Unit 700
Miami, FL 33127

Deliverables

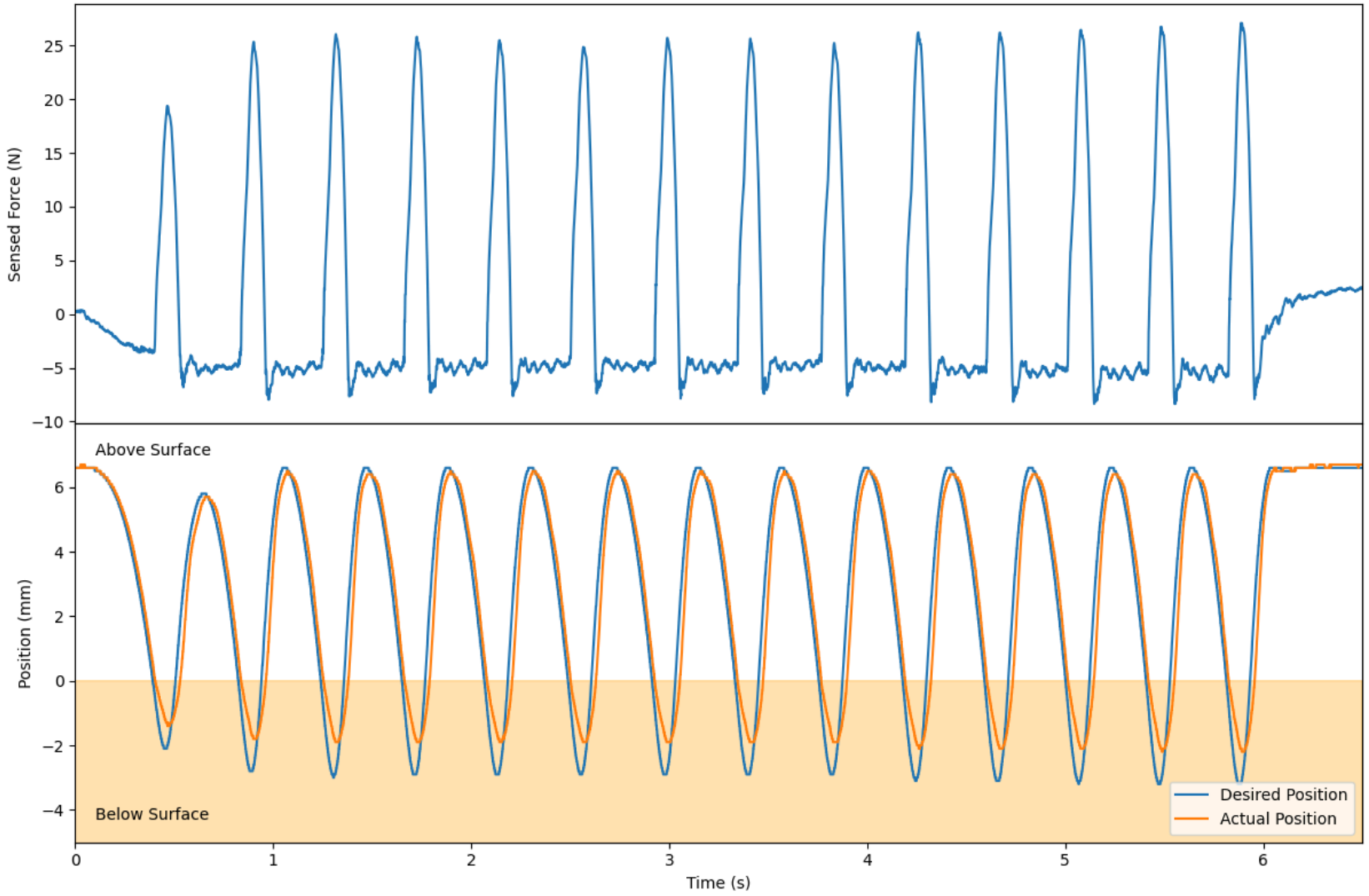
- Three separate, well-documented, sources and executables for tasks 1, 2, and 4 that generate plots of the position and force on the object (task 1), and the desired + actual position and sensed force for the surgical instrument (tasks 2 and 4). The difference between 2 and 4 should be only your bouncing fix, such that we can quickly assess the solution. Code quality and documentation is part of the evaluation, and an object-oriented solution is preferred.
- An explanation document for task 3. Figures and equations are welcome.

Requirements for submission

- Turn your result in by the end of the second day. Earlier if possible.
- Please use any standard development language (e.g. C/C++, Java, Python, JavaScript or equivalent). LabView, MATLAB or equivalent are not considered to be standard development languages.
- Attach all documentation and source code in an email to **jtieman@neocis.com**.
- Include working Windows executables, Python scripts, executable jar files or equivalent.

545 NW 26th St.
Unit 700
Miami, FL 33127

Real-life plots showing the bouncing behavior described in the document



545 NW 26th St.
Unit 700
Miami, FL 33127

Zoom starting at t=3.06 s