GNC Engineer Test

Task 1 (of 2):

To examine how you can succinctly and robustly solve a numerical problem using C++, the following task is presented.

Overview:

Write a C++ object-orientated program that will complete the following steps:

1) Prompt the user to enter an integer from 0 to 99999
2) Let A be the number entered, let B be the number with the digits reversed, calculate C = A to the power of B, i.e.

$$C = A^B$$

3) Print the resulting solution C

Constraints:

- You may *not* use any non-standard libraries (i.e. Boost, GMP, etc, are *not* allowed).
- You may submit any number of files.
- If you use AI tools for assistance, please make that clear in your submission and include the prompts you used.

Example:

```
Enter a number (0-99999): 123

Result: 123^321 is
723670338063716731491098941411637786288117926575716589060105583903958703637984017440952806
8615550773640492165707028496172182896059297790954263709889769722310262262856678765409132
7825453991595140205701412961364188732408936197890553699715836951569999800431957769217006
7433210262575179327641646623194879149625333027413682072111894946153265527906677204112854
7416263676516890721192413497337430449601963537666585855994173570392483646775691724799546958
3487467791524582153744522107597865277798136080074161485280424274076931083994487111719562
2497025403628557129111322659662357543553535167033390430015061185207603595777378694720186
1794212059087317071080507869637173890637572178572 3
```

Tips:

Computational efficiency of your software is *not* the primary goal of this exercise. Clear, well documented and maintainable code is preferred.

The submitted code will be peer reviewed and should be representative of your coding style. Your submission should also include comments on how you reached your solution.

Most importantly, at Rocket Lab you will be writing C++ to run on the flight computer of a rocket; the reviewers will be looking for how you consider that in your submission.

Task 2 (of 2):

As a GNC Engineer, you will also be tasked with solving mathematical problems using tools and languages such as MATLAB and Python. An example of such a problem is modelling the dynamics of spacecraft as they are deployed from the Electron launch vehicle's Kick Stage. To examine how you can solve problems involving dynamic systems and data analysis, the following task is presented.

Overview:

One of the first steps in characterizing the deployment dynamics of a spacecraft (satellite) is a model of the dispenser itself. For cubesats, this will be a protective housing with a spring assembly at the rear, and a motorized or tensioned door at the front. When the dispenser receives an electrical signal to dispense the spacecraft, the door opens, and the spring(s) push the spacecraft out with an "ejection velocity".

As a launch vehicle provider, Rocket Lab is interested in both the transient behavior of the deployment, to model the force and moments applied to both the Kick Stage and the spacecraft, but also the total "delta-V" (change in velocity) between the Kick Stage and the spacecraft, in order to predict the separation distance between the Kick Stage and spacecraft over the next several orbits.

A simplified model that can be used for estimating both of these properties is the mass-spring-damper model:

$$m\ddot{x} = -b\dot{x} - kx$$

where *m* is the spacecraft mass in kg, *b* is the damping coefficient in Ns/m, *k* is the spring constant in N/m and *x* is the displacement of the spring in m. The dynamics notes from Prof. Longoria, attached to this test, provide a good introduction to this model.

An example dispenser is the CSD 3U dispenser, which contains one or two springs, and can deploy 3U sized spacecraft. As the parameters of each spring are fixed, the ejection velocity of the dispenser varies with the mass of the spacecraft being deployed, as shown below:

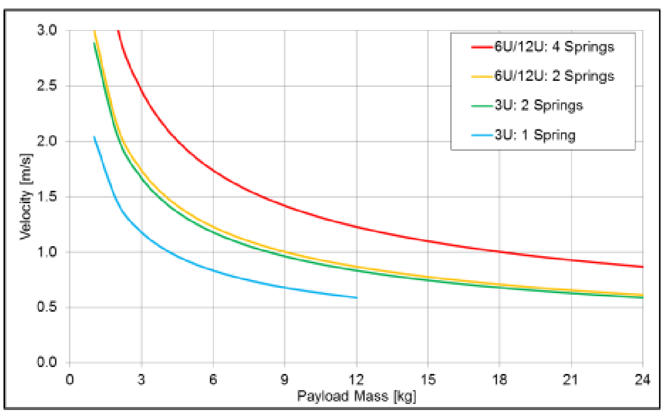

Figure 15-1: Estimated payload ejection velocity

Your task is to take the CSD *3U 2-spring* data from the CSD datasheet (attached, green line in the above figure) and derive the equivalent spring parameters *k* and *b* such that the mass-spring-damper model can accurately predict the ejection velocity of the dispenser for the range of spacecraft (payload) masses from *3kg to 15kg*. You may use MATLAB or Python to complete this task.

Your submission must include both the parameters, as well as the software you have written to estimate these parameters. As per the C++ task above, your code will be peer reviewed, but additionally your problem-solving methodology will also be reviewed. Please annotate your software with comments and state your final solution in either a text file, png or pdf document.

Assumptions:

The following assumptions can be made to simplify the problem:

- The spacecraft is assumed to have left the dispenser (i.e. the spring is no longer acting on it) once the spring force is equal to 0N.
- The spring stroke (distance from compressed to free length) within the dispenser is 0.15312m.

Tips:

- To obtain the data from the plot above you can use a tool such as: https://automeris.io/WebPlotDigitizer/
- The time taken in seconds for the mass-spring-damper system to go from its initial state (spring compressed) to when the resulting force = 0N (spring extended) is given by the following analytical expression:

$$\text{Duration} = -\frac{m\log_e\left(\dfrac{2\sqrt{-km}}{bi+\sqrt{-b^2+4km}}\right)2i}{\sqrt{4km-b^2}}$$

  where *i* is the complex number.

- All Python modules/MATLAB Toolboxes are permitted to be used.
- You may submit any number of files.