Laser Calibration Software Challenge with Control Theory

Context

Here at IonQ, we make money by shooting lasers at ions suspended in an electric field. The more accurate the laser is, the more money we make (more or less). Because of various mechanical/optical drifts over time, we have to recalibrate the alignment of our lasers fairly often.

One such programmatically controlled routine for doing this involves rotating a mirror with a software controlled mirror mount over a certain range (“a scan” is the local jargon). One high level algorithm that accomplishes the procedure is:

```
ion_responses = {}

for pos in range(start, stop, step):
  move_mirror_to_position(pos)
  ion_response = measure_ion_response()
  ion_responses[pos] = ion_response

return pick_best_position(ion_responses)
```

Methods

`move_mirror_to_position`

This function would send a device command over a serial interface, and would rotate the mirror mount to the new angle, changing the path of the beam slightly and ultimately moving it across the ion chain.

`measure_ion_response`

Measuring the ion’s response is a fairly involved process, but essentially involves
1. shoot a laser pulse at the ion that’s expected to make the ion “bright”
2. shoot another laser pulse at the ion from a different, already calibrated laser that will make the ion give off a photon if in the “bright” state
3. repeat about 100 times
4. the percentage of the time the ion gives off a photon is roughly how “good” the ion response is

Requirements

Assume that the mirror is initially aimed correctly. However, because of unimportant physics reasons, the optimal mirror position drifts. Write an algorithm that continues to aim the mirror optimally using control theory principles.

1. mirror mount positions may range from 0 to 1 (or any arbitrary continuous range)
2. ion responses may range from 0 to 100 photons per measurement round (or any arbitrary, small, integer range)

Explain how control theory concepts, such as feedback control, can be applied in this context to efficiently and continuously improve the laser calibration process. The underlying physics for the system drift are not well understood, so you can assume that an accurate theoretical model of the system can not be developed.

You may consider discussing the following points:

- How can feedback control be used to continuously adapt the mirror mount position based on real-time ion response measurements?
- What control algorithms or strategies could be employed to achieve optimal performance?
- What sensors or feedback mechanisms would be needed to implement this control approach?

Feel free to mention any specific control theory concepts or libraries you would utilize to address this problem effectively.

In general, we highly value well-commented, well-tested code, able to properly handle a range of possible start/stop values and step sizes.

Please include with your code the following infrastructure to support the reviewer:

1. A README file describing your solution and how to execute items 2 and 3 below
2. A setup script to set up the development environment and dependencies if applicable
3. A “runme” script to feed your algorithm some example data and produce outputs

Extra Credit

1. Include a unit test file. Include a mechanism to run the tests either in the existing “runme” script or described elsewhere in the README.