

Calibration and Troubleshooting

Lev Gruber

5/31/2024

Contents

1 Introduction

You're starting to do work with Professor Lynn's Quantum Optics Group at Harvey Mudd College; congrats! It's a super cool lab doing fascinating research, and I am sure you'll have an amazing time. The only thing in your way on the road to success are random bugs, apparatus errors, and worst of all, when you have to fully recalibrate everything from scratch. This guide, much of it adapted or copied from Alec Roberson's work *How to Calibrate Everything* in the GitHub's Summer 2023 repo, acts as a one stop shop for all of these issues. Please reference Alec's write up for more information on *why* the calibration procedure works, as this guide will keep its focus on what we do rather than why. Alec's guide also contains literature on Jones' matrices, laser drift, and other useful topics that I won't copy verbatim for this document. Therefore, I fully recommend you read through *How to Calibrate Everything*, especially if you're struggling through the calibration from scratch.

2 Calibration from Scratch

Some important notes before you begin calibration include:

- Remember to always wear goggles when the laser is unblocked, to never turn on the lights unless the detectors are off, and to never look down the path of the laser. It is ok to remove your goggles when sitting behind the compute due to the screen and elevation.
- The black motors with the tick marks are Thorlabs motors and the red motors with no tick marks are Elliptec motors.
- The Thorlabs motors are controlled by the software 'APT User', and the Elliptec motors 'Ello'.
- If power is lost, Thorlabs motors will assume their new 'zero' is wherever they were when they lost power. Calibration in the config.json is based off of the Thorlabs motors at their visual zero, so visual zero is a good guess at calibration (to be close to previous values for a new calibration). Oppositely, we have no clue what the Elliptec motor thinks zero is post-power loss—our best guess is `np.random.rand()`.
- All calibration files by sweeping a wave plate, or moving it while seeing how counts changes. Ensure that the sweep parameters are set to be 10 above and below the previously calibrated value in the config for a given wave plate.

2.1 The Quartz Plate (QP)

First, we need to calibrate the quartz plate. Open 'Ello' on the computer, select COM5 in the upper left, and hit connect. There should be three tabs pop up, click the one with address B. You can jog the motor to ensure you've selected the QP.

Note: **Wear goggles and do not look down the path of the laser.**

Turn off the lights, unblock the laser (wear goggles!), and use a business card or small piece of paper to identify the light bouncing off of the QP back towards the mirrors and laser. Begin with the QP visually perpendicular with the incoming light, and jog it slightly CW or CCW until a portion of the incoming beam retroreflects directly below the incoming beam on the laser's front. Note

this will be below and likely not directly over the laser due to table slant. Furthermore, note that each element in the apparatus' state creation section retroreflects, so use a second business card to ensure you're looking at the correct reflection.

Once retroreflecting correctly, update the home offset within Ello by adding the current offset to the current position; if the result is negative, add 360.

2.2 Alice's Wave Plates (and the UV Half Wave Plate)

We begin measurement-side calibration with Alice's side. Open VS Code and go to the calibration folder and within that, AQWP.py (Alice Quarter Wave Plate). For this we also need to remove Bob's BBO and Alice's half wave plate. After removing, run that, the minima of the curve being where the calibrated AQWP zero should be. Add this to the current value for AQWP in the config.json. It should look something like INSERT FIG.

Next is the UV half wave plate (UVHWP), so navigate to that folder and run UVHWP.py—the parabola minimum is where you should update the config to (plus the original config value).

Moving to Alice's half wave plate (AHWP), put it back into the set up, then run AHWP.py. Once again, the minimum value should be added to the config.json value and config.json updated.

Finally, run check.py within checka. This sweeps all plates around their calibrated zeros (does a minisweep) to ensure all are centered around zero. We aim for a < 0.1 degree error on each plate. You may have to run this a few times, updating the wave plates each time, to converge with low error.

2.3 Bob's Wave Plates

The first step is to reinsert Bob's BBO—be careful not to jog the crystal out of place. Then remove Bob's creation half wave plate (BCHWP) and Bob's half wave plate (BHWP). BCHWP is screwed down, with tape markings where it should be re-inserted later. It can be unscrewed and gently moved out of the photon's path. BHWP is on a magnetic mount, and can be gently removed from the mount at the base and placed out of the photon's path.

Begin with Bob's Quarter Wave Plate (BQWP)—run the sweep and update the config accordingly. An example is shown in INSERT FIG.

Next is BCHWP, which should be screwed into the table following the tape guidelines. Sweep and update the config, the plot looking as in INSERT FIG.

Then, reinsert BHWP onto the magnetic mount gently, and run AHWP. You should get a plot like INSERT FIG; update the config accordingly.

Run check.py within checkb, which has the same functionality as checkpoint A but for Bob's side. This functions as a final check that our measurement side is calibrated correctly. Again, it may be an iterative process to converge on a calibration with 0.1 deg uncertainty. If you are not able to achieve this uncertainty after 10 or more iterations, then it is okay to have the uncertainty < 0.25 deg—you should still likely talk to Professor Lynn to confirm.

As the time of this writing, Bob's creation quarter wave plate is not included in the calibration nor usage of the experimental apparatus.

2.4 Phi Plus and the PCC

Finally, we want to calibrate the bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}|HH\rangle + \frac{1}{\sqrt{2}}|VV\rangle$, as it is extremely useful for various experiments we may want to run. This will also allow us to calibrate the pre-compensation crystal (PCC).

Calibration of phi plus and the PCC is a three step process:

1. Run `ratio_tuning.py`, which balances the HH and VV counts to yield the even superposition wanted by phi plus. Input the value it provides into the config under the `phi_plus` section.
2. Run `phase_finding.py`, which finds where the phase is zero. Then, you can run `qp_plotting.py` to generate a fitted plot of this data; you can zoom into the zero on the plot and insert that degree value into the config.
3. Run `pcc_sweep.py`, which calculates the purity of the state. Purity is a measure of how well we've created the state phi plus and is defined by

$$p = \frac{\langle DD \rangle + \langle AA \rangle - \langle AD \rangle - \langle DA \rangle}{\langle DD \rangle + \langle AA \rangle + \langle AD \rangle + \langle DA \rangle}$$

. PCC sweep will calculate the purity at various values—you can then enter the csv where these values are saved and choose the PCC degree value corresponding to the highest purity—this can be put in `config.json` under `phi_plus` as well. We aim for a purity > 0.95 . Note that purity is dependent on lab temperature (aim for $< 20^\circ\text{C}$) and how long the laser has been on for (wait at least 10 minutes from when the laser is turned on).

You are now ready to run experiments!

3 Random Issues with the Apparatus

3.1 The Quartz Plate Refusing to Move

Sometimes, the QP will return the error `RuntimeError: Sent instruction "b'Bma0001FB24'" to ElliptecMotor-C_QP expecting response length 11 but got response b'BGS0A' (length=5)`. As of writing, we have solved this issue by allowing the QP to be attached to its own bus (COM12), such that whenever it loses connection/bugs out we can reset it in this order: remove USB, remove power supply cable, remove ribbon cable, replace ribbon cable, replace power supply cable, replace USB. We then recalibrate as per section 2.

3.2 COM5/7 Refusing to Open

Make sure you've run `m.shutdown()` if the Manager was initialized. Then, run the Task Manager program on the PC and kill VS Code. Open VS Code again and try your program. If it still does not work, you can enter Ello, connect to the COM in question, disconnect, close Ello, and try running your file again.

3.3 Serial Number Errors

Sometimes the Thorlabs motors connected to the measurement wave plates will lose connection with the computer—this is recognized by an error in the Python terminal similar to "Internal serial

number is incorrect”, and identified by closing VS Code, opening APT Users, and seeing some or all of the motors missing from the home page. Assuming that the motors had been homed before this error and are at their visual zero, this can be fixed by: Closing APT User, unplugging the motor block at their base, opening APT User with motors unplugged, closing it again, plugging back in the motor bases and waiting until all four have a green LED, opening APT User, and verifying there that all four motors now show there.

If the motor is not set to zero and not visible in APT User, you can try manually jogging it on its base to see if it is recognized. If not, I am sorry you have to unplug, replug as described above then recalibrate as in section 2.

3.4 Fitting Errors

As each fitting error is different, we describe here general troubleshooting steps:

1. During a QP sweep \rightarrow often the bounds are set too wide, such that the QP itself blocks the light at certain points. Print out the sweep right before fitting and you should see this occurring at the boundary. Fix by narrowing bounds.
2. Anything else? Best advice I can give is to *print()* statement the bajeezus out of the code. Good luck!

4 Calibration not from Scratch

Here I will describe various *extremely* useful terminal/Python commands for the Manager object that will be helpful for troubleshooting and calibrating the setup. They are:

1. `'from lab_framework import Manager'` \rightarrow imports the Manager framework into your file or terminal.
2. `'m = Manager(config = './config.json)'` \rightarrow sets the Manager object as m. The CCU will automatically start displaying data. You may need to adjust the number of dots pending what directory you're in.
3. `'m.OBJECT_goto(value)'` \rightarrow , where OBJECT is named as in your config.json and value is 0 to 360. An example is `'m.C_UV_HWP_goto(22.5)'`.
4. `'m.take_data(num - points, time - per - point)'` \rightarrow takes data into the Manager object, example is `'m.take_data(5, 3)'`.

5 Extraneous Advice

5.1 Github (Please Push)