LAB 5: SINGLE PHOTON INTERFEROMETRY

by: Gordon Jones, Brad Moser, Brian Collett, Megan Smith, and Viva Horowitz

PHYSICS BIG IDEAS:

- Light travels like a wave, including traveling through a beam-splitter.
- We detect light using the photoelectric effect.
- Light acts like a particle – a packet of energy -- when it undergoes the photoelectric effect.
- The change from acting like a wave to acting like a particle is an example of "collapse"

WHAT TO EXPECT THIS WEEK:

- Aligning laser beams
- Fiber optics (The Nobel Prize in in Physics, 2009, was awarded partly for the invention of fiber optics!)
- Python automation
- Mind-blowing quantum mechanics

PRE-LAB QUESTIONS

1. A 1 mW laser beam (λ=632.8 nm) is attenuated by neutral density filters with optical depth OD = 8. The laser is then directed into the Michelson Interferometer we set up in the last lab.
   a. Given this attenuation, how many photons per second enter the detector?
   b. What is the average distance between photons after the light passes through the neutral density filters?
   c. Qualitatively, is it likely that there is a photon in each arm of the interferometer at the same time, if you only consider the particle model of light?

2. When you assembled the interferometer and tested it, you noted that if you blocked one arm or the other, then no interference pattern appeared on the screen. If both arms are unblocked, then an interference pattern is produced. But what happens, as in the scenario from question #1, if only one arm or the other has a photon at any given moment, even though neither arm is blocked? Is an interference pattern produced?

AN INTERFEROMETER OPERATED BY CODE

This week is Part II of the Michelson Interferometer investigation. We will use computer code to move a mirror mount and collect data from a photodetector, and show what happens when a single photon passes through the apparatus. While *qualitative* observations are very useful, we are going to get more *quantitative* by measuring the interference fringes with a photodetector as a function of the mirror position.

To explore this phenomenon, the length of one arm of the interferometer will be computer controlled. The mirror for one arm is mounted on a linear translation stage, like the one shown in Figure 1. Normally, linear states are positioned by a micrometer screw, but we will insert a piezoelectric actuator on the end of the micrometer. The piezo actuator is a material that expands when an electric voltage is placed across it. We will use a LabJack data acquisition board to produce the expanding voltage and to read the output voltage of a photodiode. The whole automation system is shown in Figure 2.

Figure 1: A linear translation stage (reprinted from Thorlabs.com, yours might look different), piezo actuator (reprinted from Thorlabs.com[i]), and LabJack data acquisition board (reprinted from LabJack.com[ii])

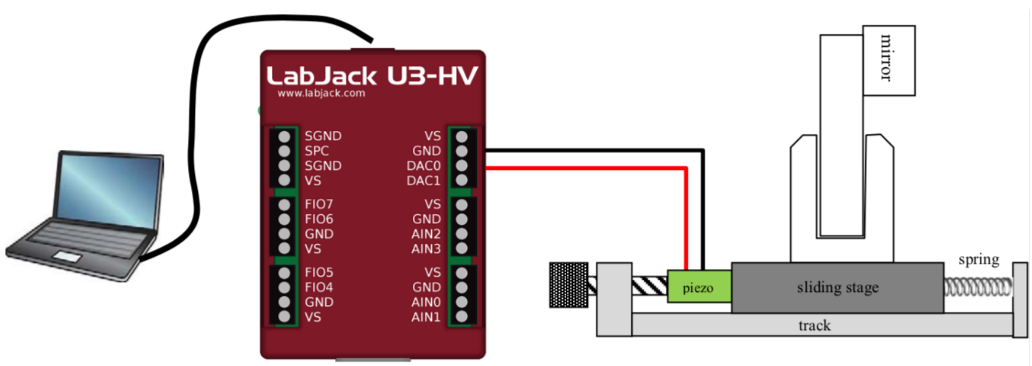

Figure 2: The computer controls the LabJack data acquisition board, which can produce a voltage to expand the piezo actuator. The mirror is mounted on stage that slides when the piezo actuator expands. (LabJack image reprinted from LabJack.com[iii])

Task Summary:

1. **Align the Michelson interferometer so that you see the distinct fringe pattern.**
2. **Use python to move the mirror on a translation stage and observe the fringes changing.**
3. **Align the beam into an optical fiber and collect the light.**
4. **Adjust LabJack connections**
5. **Use python to move the mirror and measure the amount of light. Repeat the measurement three times in a row and graph the results.**
6. **Adjust LabJack connections to prepare for single photon interferometry, add filters, and request the single photon detector.**
7. **Observe single-photon interference. Marvel at the bizarre single-photon physics.**
8. **Take data (2 passes) with one arm of the interferometer blocked.**

Apparatus Assembly

Make sure the mirrors and beam-splitter are tightly screwed down. This is precise laser alignment, so make sure nothing is loose.

The apparatus should still be working: check to make sure the interferometer still produces fringes. Align it with the bullseye.

For now, put the lens and screen in place to visually inspect the interference pattern and make sure the mirror on the new base is aligned. Later, you will replace the lens/screen pair with an optical fiber to a photodiode.

Software

You need a program to slowly move the mirror while recording the amount of light at each position. The LDrive and Single Photon files should already be installed on your lab computer if you log in as pguest.

Automation Hardware

A circuit board has been installed on the LabJack at pins FIO5, FIO4 (IO=input and output), GND (ground), VS (voltage supply). This controls the motorized translation stage. The translation stage connector plugs to the four-pin input. A 15V potential difference is required to drive the motor; this is connected to the two-pin input (the +15V wire should be attached to the bottom (outside) pin).

The photodetector input is connected to FIO7 and GND, but it's different for the single-photon code.

Computer Controlled Hardware (Automation)

It is time to connect the electronics to the optics. First, attach the piezoelectric actuator to the voltage output of the LabJack (red wire in DAC0, other wire in GND – use a slotted screwdriver from your supplies) and run the program. The mirror should now move due to the 0-5V voltage you are applying to the translation stage. By visual inspection on the screen, does the interference pattern behave as you expect?

Software: Writing python

Your task is to write a python program that moves the photodetector mounted on a translation stage. Your program should move the stage a certain number of steps, then stop to record the light signal at that position before moving on. As an initial step, set the voltage to adjust the mirror position.

Open the Python editor on the lab computer. As you have seen, your code needs to import a few modules:

```
import matplotlib.pyplot as plt
import numpy as np
```

```
import time
import u3
```

Check to see if that runs before you go on. If it has an error that says "no module named matplotlib" then your instructor can help you. Instructor: open Applications -> Utilities -> Terminal and pip install matplotlib. For u3: pip install LabJackPython

Additionally, you need to import LDrive, the module written by Brian Collett.

```
from LDrive import LDrive
s = LDrive()
```

Here, we've named it ***s*** (for scan). This gives you access to a few very helpful functions that communicate with the LabJack. You are welcome to read the LDrive.py file and make sense of these functions for yourself, but the summary is listed here.

1. We write s = LDrive() to initialize the LabJack communication. After that, LDrive commands all have the format s.*command*(*input*). The LDrive commands for talking to the LabJack are:

    1. s.read(chan): read 1 voltage from AIN (analog input) channel 'chan'. Eg. AIN0: 'read(0)'

    2. s.DAC0Write(volts): Set DAC0 output to 'volts', a number from 0 to 5.0.

    3. s.DAC1Write(volts): set DAC1 output to 'volts'.

    - The DAC Write command is enough so that you can now use python to move the mirror on a translation stage and observe the fringes changing.
    - Once you have observed the fringes changing, you will adjust the optics (see Optics on page 6) and continue writing python code (see below). *You can do either one of those first.*

2. Write a program to sequentially set each element in an array to numbers between 0 and 5 in steps of 0.1. Some commands for creating arrays are listed below. You can look up any of these online and/or ask your instructor about them.

    a. np.zeros, np.zeros_like – initializing/creating an array of zeros.

    b. np.arange, np.linspace – initializing/creating an array of numbers in order (or use a for loop)

    c. np.copy – copying an array (to create a new array).

3. For each value of the array, also set the output value on the LabJack to the same voltage. Use the DAC0 connector on the LabJack. DAC0 stands for "digital to analog converter channel 0." You can check the voltage by measuring the voltage between the DAC0 connector and the GROUND connector next to it with a voltmeter (in this case, an oscilloscope).

4. Once the output is set each time, read in the input voltage (a.k.a. the signal) on the LabJack analog input channel 0. Use AIN0 which stands for "analog input channel 0." If done correctly, your program should sequentially set the output voltage to a value from 0 – 5 on DAC0, and then it

will read an input voltage on AIN0, before going on to the next iteration. The LabJack is fairly slow, so put a pause (sleep) of 0.03 seconds in your script after setting the output voltage but before reading the input. Sleep info: https://www.geeksforgeeks.org/sleep-in-python/

5. Plot the input signal vs the output voltage. With the input unconnected, the signal voltage will drift to a random value. To test your script before you actually try to take data: Use a wire to directly connect the analog output (DAC0) to the analog input (AIN0). When you run your script, the signal voltage should now record the voltage applied on the output. It probably won't look like what you expect because the LabJack is very slow to respond. At the beginning of your loop, set the initial output to zero and then have a **short pause** before starting the loop. This should look better.

6. Finally, put your script into a loop to repeat the voltage ramp 3 times.

7. In general, visualizing your results involves making choices. I highly recommend that you plot the result at the end of each voltage ramp rather than waiting for all three to finish. The axes will be in "arbitrary units". Here are some valid options.

    1. You might graph the results with three plots on top of each other, each with their own noise. or

    2. You might calculate the average of the three scans and plot that. (But if you do, you should also choose one of the other options.) Since the axes of the graph are arbitrary units, it doesn't matter whether you calculate an average or simply add the new signal onto the old signal and plot the sum. Or

    3. If you want to be very efficient with memory, you might use one array for the measurements and add the results from each scan to the existing array. Then you could plot the three curves and see how the signal grows with each repetition of the scan.

    Just-for-fun aesthetics option:
    If you want to make your plot use different colors, look up [colors matplotlib] and add an option like color='palevioletred' to your command for making the plot. Now all your curves are the same color, which you might like or maybe don't like – it's a stylistic choice. Welcome to the world of aesthetics in scientific communication: now you are thinking about *data visualization.* You could make each curve a different color: if that's what you want, then make the name of the color a variable and change it for each of the three loops. A professional would also consider if each of the colors shows up distinctly for someone with various forms of color blindness.

Optics

Start with a bullseye fringe pattern and adjust the movable mirror so that the center is bright.

In order to measure the signal quantitatively, replace the screen with the mounted fiber optic cable. The post holder will need to go on the optical table; place it toward the edge of the table. You may still use the diverging lens to broaden the beam if that is helpful for you. It gives you the ability to move the beam around.

Optical fiber alignment:

1. This is no easy task! The goal is to put the fiber on the optical breadboard in just the right place without changing anything else about your optical apparatus.
2. Pro-tip for laser alignment: in a perfectly aligned setup, the laser *should* land on the fiber mount even if you don't put the diverging lens in, but if you do use the lens, it doesn't really have to. Since it doesn't have to, the lens gives you some wiggle room: you can adjust the position of the lens to help you get the light in the right place.
3. Hold the opposite end of the fiber over a piece of paper. When the laser light is entering the fiber properly, you should see red light pass through the fiber onto the paper. You can make small adjustments to the lens and/or fiber mount to get as much light through the fiber as possible. (Do NOT adjust any mirror mounts to maximize light through the fiber, since that will negatively impact your interference pattern.)
4. Once you see light coming through, mount the free end of the fiber into the DET10A photodiode with the adaptor[1]: you want the end of the fiber close to the silicon detector square but DEFINITELY not scraping it. You can ask your instructor for help.
5. Once you have light reaching the photodiode, electrically connect the photodiode to the oscilloscope[2] and use the "autoset" function to get a measurable signal. (Block one arm of the interferometer for this measurement, since the interference pattern makes it tricky to measure a steady signal.) With the oscilloscope, you can fine tune the lens position, and then the vertical and horizontal control on the fiber optic mount, to maximize the signal on the oscilloscope. (Did you remember to turn the detector on? If it doesn't work, or only works poorly, the photodiode battery might be dead.)

Now that the fiber is aligned, it's time to send the signal to the LabJack. Unplug the BNC cable from the oscilloscope and connect it to the adapter the red and black wire. The red wire goes into AIN0 and the black into GND. When you run the program, you should see the photodiode voltage change as the interference pattern moves over it.

Use your python script. Observe the graph of photodiode signal vs. translation stage actuator voltage. Explain how the interference pattern shifts when the length of one of the arms changes. What happens if you block one of the arms? Light must be in both arms for the interference pattern to appear. Save a copy of your graph for your records. Make sure the axes are labeled. You'll be submitting the graph with a caption. (Cmd-Shift-4 is useful for taking a screen shot on a Mac.)

Single Photon Interference?

It's time to see if you can measure an interference pattern with only *one* photon in the Michelson Interferometer at a time. Place neutral density filters that total OD = 7 or 8 at the entrance to your

[1] The adaptor is a Thorlabs SM1FC - FC/PC Fiber Adapter Plate. It's designed so the rectangular "key" goes in the slot, then you screw it in.

[2] This is called a BNC connector.

interferometer. Although you can no longer see the signal, the photodetector can see it. Well, not the one you've been using. Instead, we will need to use an avalanche photodiode (APD) and its signal conditioner, which is much more sensitive to low light signals. Instead of measuring a voltage, the script counts the rate that individual photons hit the detector.

Load the python program SinglePhoton.py from and **read the comment at the top of SinglePhoton.py for instructions for how to reconnect your LabJack**. Follow the instructions.

We need to share the APD because we only have one. After you have reconnected the LabJack, call the instructor, replace your photodiode with the APD, block any extraneous photons, and run SinglePhoton.py. Save a copy of your graph and data for your records and print it for your lab notebook. Contemplate the implications of this result. Take another run (looped twice) with one arm blocked.

Cleanup

Make sure everyone has a copy of the data, python script and graphs. Please turn off the photodiode, oscilloscope, and laser, and disconnect all the cables and adapters.

LAB ASSIGNMENT

1) Write Results and Conclusion sections for the single photon interference experiment:
    - Include a copy of your python script as an appendix.
    - Include graphs of your interference scans with your photodetector, with axes labeled and captions.

---

[i] https://www.thorlabs.com/thorproduct.cfm?partnumber=PK2FMP1
[ii] https://labjack.com/products/u3
[iii] https://labjack.com/pages/support?doc=/datasheets/u3-datasheet/2-hardware-description-u3-datasheet