Properties of a Solid-State Laser
A Computer Controlled Experiment

339 - Measurements Lab
Due March 12th, 2018

February 20, 2018

Professor: David Cooke[1]
Teaching assistants: Aaron Mascaro[2], Andreas Spielhofer[3], Eli Martel[4], Ata Madanchi[5], Ziggy Pleunis[6]
Technician: Saverio Biunno

1 Introduction

In this experiment, you are going to examine some of the properties of a solid-state laser. You will measure the direction of its polarisation, if any. You will measure the non-linear response of the laser to current. You should also be able to determine Brewsters angle for a glass plate by measuring the variation in the transmission of polarised light through it when the angle between the beam and the plate is varied.

To prepare yourself for this experiment from the physics point of view, you need to refresh your knowledge of optics. In particular, you should understand polarisation and Brewsters angle. You can find these in most of the elementary optics text books. We have included some reading material/technical ramblings in another file. It will be helpful to read them before, during and after you do the experiment (although most of the **experiment-related** parts of this document are no longer applicable due to hardware upgrades in recent years).

For this set of experiments, you will use the computer to set the conditions for your measurements. You need to rotate a polariser or a glass plate, or to move a knife edge under computer control, and measure the intensity of the light reaching a phototransistor after each movement.

The motions are achieved using a stepper motor. Unlike the electric motors you are familiar with, these turn in discrete steps in response to pulses applied at their input; one step per pulse. Your Arduino has been fitted with a shield (an accessory board basically) that allows it to power a stepper motor with very little technical input from the user. For more information on this shield, check out: `https://www.adafruit.com/product/1438`.

Always try out the experiment quickly before polishing the programs. Get a feel for what you have to do, what accuracy is needed, what range of parameters you need. As always, the description below should be taken as a guide. If it does not work, try something else, don't blame us (well, you can blame Andreas, that's generally encouraged).

[1] david.cooke2@mcgill.ca
[2] aaron.mascaro@mail.mcgill.ca
[3] andreas.spielhofer@mail.mcgill.ca
[4] eli.martel@mail.mcgill.ca
[5] ata.madanchi@mail.mcgill.ca
[6] ziggy.pleunis@mcgill.ca

2 The Measurements

Before you attempt to load the Arduino sketch (`Laser_2018.ino`) we first have to make sure the proper libraries for the stepper motor shield are installed. In the toolbar of the Arduino IDE click on `Sketch -> Include Library -> Manage Libraries`. The Library Manager should open. It should take a few seconds (up to a minute) for it to load all the available libraries. Once it does and the search bar is click-able, type in '*Adafruit Motor Shield V2*' and click on the first result of the same name. If the **Install** button is not greyed out, click it and wait for the install to finish. Now you should be able to load the sketch without any errors.

Before you make any measurements you should characterize the stepper motor. How many steps make a full rotation. Given the gear ratio, is this consistent with the motor having quantized stable positions?

Before you can calibrate your stepping motor, you will need to find the range of stepping frequencies which your motor is most efficient and reliable at. For most of the motors this falls in the range 200 - 500 Hz. Too high a frequency generates too little torque, too low a frequency will excite mechanical resonances.

To measure the polarization of the laser place a polarizer plate in the beam, and measure the variation of the laser beam intensity as one rotates relative to the other. The results should then be compared (fitted) to a sinusoidal curve. This measurement gives you a chance to write a data acquisition program and to actually fit your results. Pay attention to the linear response requirement of the phototransistor. The goodness of your sinusoidal fit to the data will be an indication.

You will observe that the intensisty of the light emitted by the laser is a very non-linear function of the current supplied. Is the polarization affected?

Determine Brewsters angle for the glass plate provided. The simplest way to do this is to manually (i.e. hold it in your hand) vary the incident angle of the laser beam on the glass plate and observe when the reflected intensity goes to zero. DO NOT LOOK INTO THE BEAM. This method works far too well and does not use a computer, it is therefore unacceptable for this course. In principle, Brewsters angle can also be deduced by monitoring the transmitted intensity of light polarised in or perpendicular to the incident plane as a function of the incident angle. Make this measurement and compare your results with the reflection measurement. Which is more accurate? Which is easier? How would you improve the method used? This measurement demonstrates that computers do not always give superior results, and often do not make the task any easier.

In your lab report, you should include a brief description of the principle of operation of a solid-state laser.

3 Software

You are provided with an Arduino sketch, `Laser_2018.ino`, and a sample client Python script, `Laser_2018.py`. There is very little of note about the Python client, it is the same interface given in the Introduction lab. It uses the `Arduino` class contained in the file `Arduino.py`, which must be in the same folder and/or your path. The Arduino sketch provides the following commands:

- `LASER` - This sets the current to the laser, via an all new I^2C DAC. Please ask for assistance about how to calibrate this.
- `START` - Switches from mode 0 (off) to mode 1 (on), see below.
- `STOP` - Switches to mode 2, which requests the motor to stop at the end of the current loop. If `STEPS` is equal to the number of steps required for a full rotation the motor will stop in the same position it started in.
- `ABORT` - This switches immediately to mode 0, stopping the motor. The original position will be lost.
- `DELAY` - This sets the wait time in `ms` between the motor step and the `analogRead()` call. A few `ms` should be adequate, but don't wait longer than the timeout in the Arduino sketch (1s default).

- `STEPS` - The number of steps per loop.

The modes are as follows:

- Mode 0 - Default, the arduino waits for a command. It will send `'Timeout!'` if no commands are sent within 1 second.
- Mode 1 - This is the data collection mode. The motor is stepped, the delay is waited, then the input is read and the value sent to the serial port (i.e. your Python script).
- Mode 2 - This is the same as Mode 1, except once the current loop is completed the mode changes to Mode 0 and the motor is halted.

The format of data returned in Mode 1/2 is as follows: `aaaa:bbbb`, where `aaaa` is the 4 digit representation of the current step number with leading zeros as necessary, and `bbbb` is the 4 digit representaion of the ADC result also with leading zeros. The reason for the fixed width of the output is that printing of the result back to the computer is part of the inter-step delay. If this were not done, small numbers would cause shorter delays and larger numbers longer delays, which would result in a jerky motion.

4 Suggestions

You can tune the frequency of the motor by opening the Arduino serial monitor and using the commands START, STOP and DELAYS commands. If you choose to move multiple steps using the `myMotor->step(1, FORWARD, SINGLE)` method (where the first input is the number of steps), the frequency will be set by the initialization of the `Adafruit_StepperMotor` object: `AFMS.begin()`. The default is 1kHz, as indicated.

You can tune the laser current by shining the beam onto the phototransistor and using the `LASER` command to adjust the current to the laser until the phototransistor output is in the range 2-4 V. Later you will probably want to write some Python code to measure the intensity as a function of current, but initially just set a usable curent.

Having determined useful values for the `DELAY` and `LASER` parameters, you can update the Python example in order to calibrate the number of steps per rotation. If you set up the laser to shine through the rotating polaroid, the pattern will repeat each rotation. If the `STEPS` parameter is correctly set, the two traces shown by the program will coincide. If the parameter is not correctly set, the pattern will creep to the left (too small), or to the right (if too large).

5 Notes

1. Discussions of Brewsters angle usually involve the reflected intensity (R) as a function of angle for parallel polarisation. You will be measuring transmission (T). Recall: R+T=1.
2. Some lasers may have polarised outputs. You should determine (How?) which type you have before proceeding with the measurements.
3. The phototransistor is powered by the fixed 5 V output at the right side of the power supply. There is no display or control for this voltage.