

# Multiplex wavelength measurements and control

Oliver Walton  
(Dated: April 30, 2021)

Free-running tunable continuous wave (CW) lasers will begin to drift in wavelength over time. This issue can be solved computationally by constantly tuning the laser to counteract the drift, essentially locking the wavelength. This was done with an external cavity diode laser (ECDL) and it was found to be locked to the desired wavelength within 0.02 pm.

## I. INTRO

Free-running tunable CW lasers drift in wavelength over time, which can lead problems in experiments that require a specific frequency of light over a prolonged period of time. A well applicable way to remedy this issue is to actively lock the cw laser to an optical frequency reference, such as wavelength meters, atomic transitions, or ultrastable optical cavities. The goal of this project is to create a setup to a commercial wavemeter that can read and lock the wavelengths of multiple lasers.

## II. DESIGN

For this, the basic design is to start with our ECDL, which then feeds into a fiber switch, and to a wavemeter. The wavemeter measures the current wavelength of light and sends the results to a computer that then generates an error signal by calculating the difference between the current wavelength and the targeted wavelength. A negative feedback loop that is composed by a DAC and a HV amplifier, as shown in fig[1]. This servo control is sent to the piezo driver (PZT) in the ECDL, where the diffraction angle is slightly adjusted, causing the laser's wavelength to be effectively locked to the desired frequency. adjust servo control

## III. COMPUTATIONAL METHODS

The computational framework is relatively simple however, the bulk of the work lies in attempting to communicate between multiple different machinery that all have different commands.

### A. Computational machinery

The main pieces of machinery that need an open communication with the program are the fiber switch, the wavemeter, and the DAC. However the speed at which the program can run is several times faster than the time it takes for each machine to receive and then complete a command. Due to this, the time it takes for each piece of equipment to begin and finish performing actions had to be individually measured. The program was then arti-

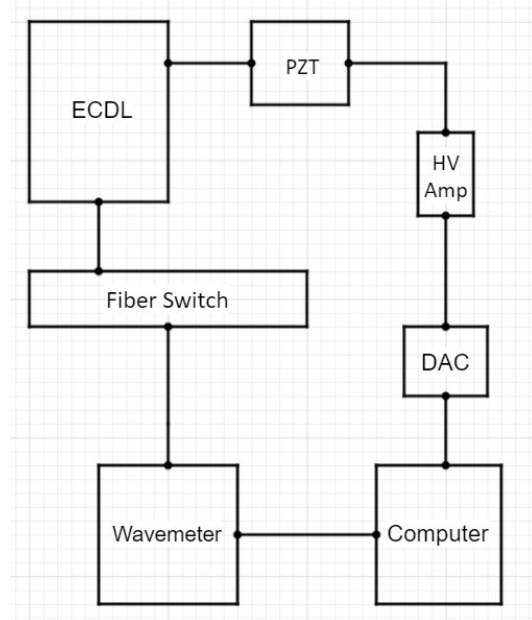


FIG. 1. A basic diagram of the feedback loop used in the experiment.

cially slowed down to ensure that the equipment would stay in sync.

The fiber switch which allows 12 inputs and 1 output. The switch itself allows a single input channel thus giving the user the option to open a single given input channel, switch to a new input channel, or close off the current channel, while also allowing the program to inquire which input channel is currently in use. Here there are 3 possible time losses, the time it takes to close a channel, the time it take to open a new channel, and the time it takes for a newly opened channel to be fully opened or more specifically; the time it takes for a newly opened channel to stabilize. As can be seen in fig[2] which depicts the switching back and forward between two channels, the fiber switch actually fully closes the current channel before opening the new channel. Fig[3] shows that certain channels also seem to oscillate slightly when switching to them, which slowly stabilizes. Each channel pair was timed to find that the closing and opening of each pair to be within [0.8]ms of each other, however the oscillations of certain channels where found to be unique from one another. Finally the response time was somewhat of an issue, and if two commands or a command and a query

where sent within too short of a time of each other, the switch would not receive the second command. To combat these timing problems a sleep function was created and was used to slow the program down to a speed that was reasonable for the switch to react to.

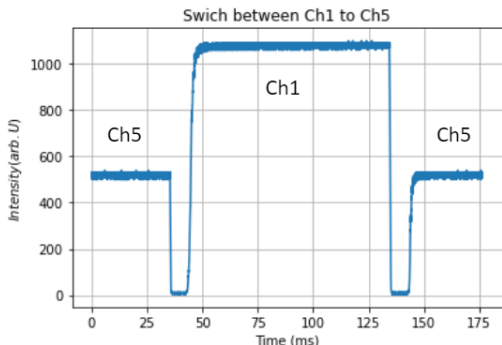


FIG. 2. An example switching back and forward from Channel 5 of the fiber switch to Channel 1.

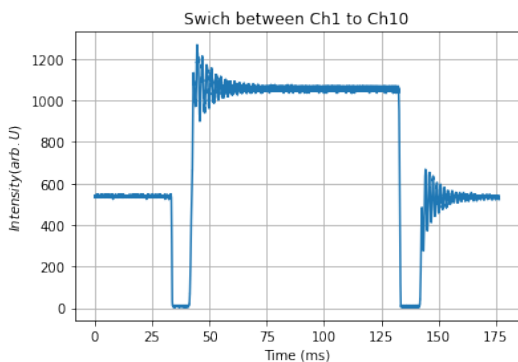


FIG. 3. An example of oscillation occurring during a channel switch.

The WS/7 wavemeter has the programmable capabilities to ask for the frequency of light and to set the exposure time. This wavemeter also allows for the ability to set an auto exposure mode, in which the wavemeter automatically attempts to find an exposure time that does not over or under saturate the wavemeter, notably this mode can find an adequate exposure time faster than could be manually programmed due to communication delay issues. The wavemeter is able to receive commands from the program relatively quickly, however the limiting factor is the response time, as if asked the frequency too quickly, the wavemeter will respond with the previous reading of the frequency. Meaning that there is a very slight updating time that the program must be delayed by, which was found to be  $[0.05]$ ms.

The laser wavelength is controlled by a PZT, that tunes the angle of a dispersive grating in the laser head. To drive the PZT, an HV analog signal is required. To generate such signal, we use a DAC. The max signal to come

from the DAC is approximately 10V, whereas the controller can take a maximum signal of 150V. In order to correct this voltage mismatch a high voltage amplifier (HV amp) must be used. Thus, the computer-controlled program can only send an arbitrary digital signal to the DAC, and then read the change in frequency measured by the wavemeter. Thus it was experimentally tested to determine the analog signal necessary to adjust the light's frequency by a given amount. Finding the speed at which the frequency changes is tricky as it only becomes obvious that a frequency change has occurred when the change is measured by the wavemeter, which itself requires an artificial delay in the program in order to read the current wavelength. The speed of frequency change was thus measured by observing the minimum artificial delay needed to measure a change in frequency subtracted by the delay needed for the wavemeter to measure a change in frequency, finding the minimum artificial delay needed to measure a change in frequency to be  $[0.1]$ ms.

## B. Computational Algorithm

The program's goal is to be able to lock a laser's wavelength to a given frequency for a long amount of time, while allowing for the fiber switch to switch signal channels whenever is needed. The signal channel and desired frequency is first initialized, then the locking process is begun. This process first asks the current light frequency, then sends a signal to the DAC for needed adjustments, then pauses for the minimum time required to measure a change in frequency by the wavemeter before looping again. This process is then continued for the desired time.

## IV. RESULTS

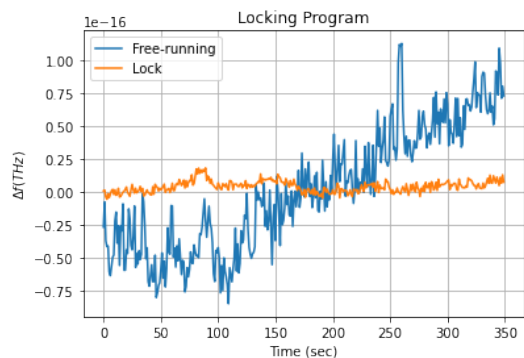


FIG. 4. free-running: ECDL being left to run for 5 minutes undisturbed. Lock: The laser being locked for 5 minutes.

The ECDL was left to run for 5 minutes undisturbed [fig.4], then the locking program is switched on. The

ECDL was then shown to be locked to the desired wavelength for the 5 minutes within to within 0.02 pm with a 1 second computational cycle time.

## V. CONCLUSION AND OUTLOOK

At the time of this experiment, there was only one working laser in our lab, however this could be easily ex-

panded to a multiple laser setup. This could be done by reducing the time of the locking cycles, allowing for the first laser to be put into a locked position, then the program must switch to the next laser and lock that one. With this method, up to 12 different lasers could be locked to the precision of the one laser shown in this experiment.