



universität
uulm



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Building up a Frequency Stabilisation Based on a Wavemeter for a Diode Laser and Characterisation for a Digital Mirror Device for Artificial Potentials

Bachelor thesis

by

Aitor Beldarrain Ruiz de Azua

aitor.beldarrain@uni-ulm.de

abeldarrain005@ikasle.ehu.es

Supervisor: Prof. Dr. Johannes Hecker Denschlag

Supervisor of home University: Dr. Jon Urrestilla

Ulm University - Institute of Quantum Matter

Winter Semester 2014-2015

Statement of Authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Ulm, March 10, 2015

(Aitor Beldarrain Ruiz de Azua)

Contents

| | | |
|-----------|---|-----------|
| 1 | Introduction | 4 |
| 5 | | |
| 2.1 | Absorption, Spontaneous and Stimulated Emission | 5 |
| 2.2 | Population Inversion and Intensity | 6 |
| 2.3 | Resonators and Modes | 7 |
| 2.4 | External Resonator | 7 |
| 2.5 | Piezoelectric Element | 8 |
| 2.6 | Collimation | 9 |
| 3 | Characterisation of the Laser Diode | 10 |
| 3.1 | Backcoupling | 10 |
| 3.2 | Threshold Current and Output Power | 11 |
| 3.3 | Mode Hopping | 12 |
| 4 | Fabry-Perot Interferometer | 13 |
| 4.1 | Free Spectral Range | 13 |
| 5 | Wavelength Meter | 15 |
| 6 | Optical Setup | 16 |
| 7 | Setup for Wavemeter Lock | 17 |
| 8 | Arduino Due vs. NI USB-6008 | 18 |
| 8.1 | Resolution | 18 |
| 8.2 | Speed | 19 |
| 8.3 | Price | 20 |
| 9 | PID Controller | 21 |
| 9.1 | Programming a PID Controller | 21 |
| 10 | Experimental Data for the Wavemeter Lock | 22 |
| 10.1 | Long-term Measurements | 22 |
| 10.2 | Measurements for Step Function Response | 23 |
| 11 | Digital Mirror Device: LightCrafter 4500 | 26 |
| 12 | Arrangement of Mirrors in the DMD | 27 |
| 12.1 | Flipping of Mirrors | 27 |
| 13 | User Interface | 29 |
| 13.1 | Displaying Images | 29 |
| 13.1.1 | Firmware | 29 |
| 13.1.2 | Add Images to Firmware | 29 |
| 13.1.3 | Firmware Upload | 30 |
| 13.1.4 | Video Mode | 31 |
| 14 | Experimental Data | 32 |
| 14.1 | Data Analysis | 36 |
| 15 | Switching of Mirrors | 37 |

| | |
|--|-----------|
| A Appendix: Block Diagram of the LabVIEW program for the PID Controller | 40 |
| B Appendix: Front Panel of the LabVIEW program for the PID Controller | 41 |
| C Appendix: Arduino Program for PID Controller | 42 |

1 Introduction

I have divided this project in two parts. In the first one I have built a laser and programmed a PID controller. In the second, I have made the first steps on the understanding of a **Digital Mirror Device** (DMD).

The aim of the PID controller is to fix the wavelength of the laser light at 670,977 nm, which is the resonance wavelength of ${}^6\text{Li}$. This is the fermionic atom they are using in the experiments of laser cooling in the LiLaLab¹, in the group of Prof. Dr. Johannes Hecker Denschlag. In order to program the PID controller, I had to establish a communication system between a LabView program and Arduino, the Digital to Analog Converter used for this purpose.

The DMD used in the second part of the thesis was the LightCrafter 4500. This device allows us to project images which can be used in experiments in order to create artificial potential surfaces in order to arrange atoms in self-designed lattices. The main parts of the DMD are the LED's that emit light and the micrometer-order mirrors which act like pixels to form the wanted images. The DMD device is controlled by a software. I have learned how to use several features of the software in order to see how the already mentioned main parts of the LightCrafter 4500 work.

¹Lithium Lattice Laboratory

2 Diode Laser²

2.1 Absorption, Spontaneous and Stimulated Emission

An electron in an atom can reside in certain possible discrete states. The states in which the electrons are standing determine the energy of the atom. Only one of the states is stable, the ground state; this state is normally the most populated one (not in lasers). However, electrons can undergo transitions from one energy level to another by absorbing or emitting energy in form of photons, in this way, the excited states can be populated. The frequency of such a photon is

$$\nu = \frac{|E_f - E_i|}{h} \quad (1)$$

where E_i and E_f are respectively the initial and final energies and h is Planck's constant.

If E_f is smaller than E_i , the photon will be emitted. If the photon is absorbed, the energy of the electron will increase.

The interaction between photons and electrons gives rise to three different types of transitions. The first type of transition of an electron is the resonant transition. In this case, an electron is sitting in its ground state until it absorbs a photon with frequency $\nu = \nu_0$, where ν_0 is the transition frequency between the initial and final energy levels.

In the second type of transition, a electron resides in an excited energy level. After spending some time in that level -the lifetime of an electron in an excited state-, the electron decays to a lower level and radiates a photon of which energy $|E_f - E_i|$ corresponds to the difference between the two levels. A photon emitted in this way has a random direction and a random phase, so the light will be incoherent. Such a process is called spontaneous emission.

The third kind of transition is the stimulated emission. An electron is initially in the excited state. An incident photon of frequency $\nu = \nu_0$ forces the electron to jump from the excited state to the ground state, which generates a new photon in addition to the incident photon. The new photon has the same phase and direction as the incident photon, so the emitted light will be coherent.

²Laser Diode used: Thorlabs HL6756MG[1]

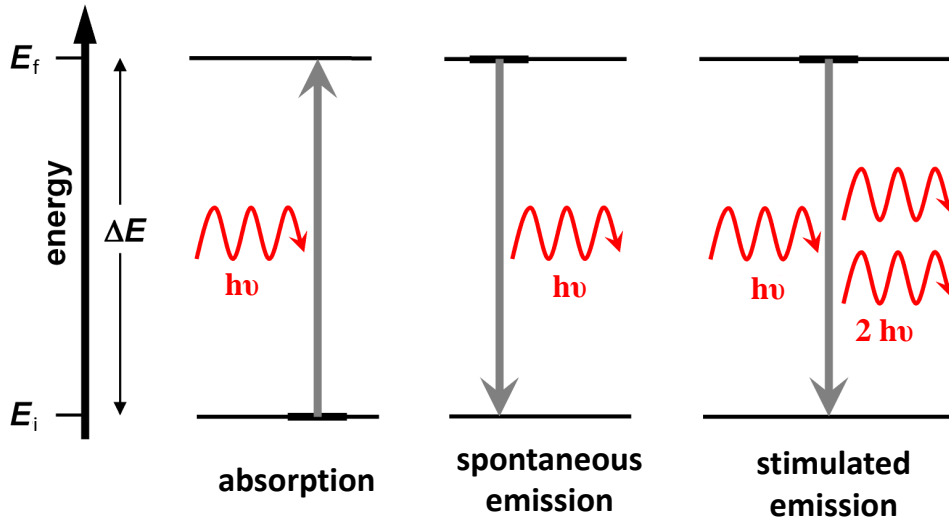


Figure 1: Three types of transition: absorption, spontaneous emission and stimulated emission. In the absorption, an electron in the ground state absorbs a photon of frequency ν_0 and jumps to an excited level. In the spontaneous emission, an excited electron emits a photon and decays to the ground state. For a stimulated emission to happen, an excited electron receives a photon that makes the electron jump from the excited state to the ground state, generating the same photon in as the incident photon.

2.2 Population Inversion and Intensity

In order to provoke a laser diode to lase, atoms have to undergo stimulated emission. In order to achieve that, laser diodes use as active zone a pn-junction device of a strongly doped semiconductor.

In order to maintain a stimulated emission in the laser diode, more atoms must reside in excited levels than in lower energy levels, i.e. there must be a population inversion between the energy levels in the junction. In order to accomplish that, atoms has to be externally pumped to higher energy states. The pumping consist in having more electrons in the conduction band than in the valence band. This can be fulfilled by making a strong electric current cross the pn-junction, the electrons in the current are in the conduction band, so thanks to the current, we are pumping electrons of the conduction band into the laser diode. When the current reaches a certain value, called threshold current, the laser diode starts to emit stimulated radiation.

When the current through the pn-junction is low, i.e, lower than the threshold current (see section 3.2), the spontaneous emission predominates in the diode, an emission similar to a LED.

Over the threshold current, the stimulated emission predominates and the diode starts to emit coherent laser light. The threshold intensity depends on the resonator and the temperature.

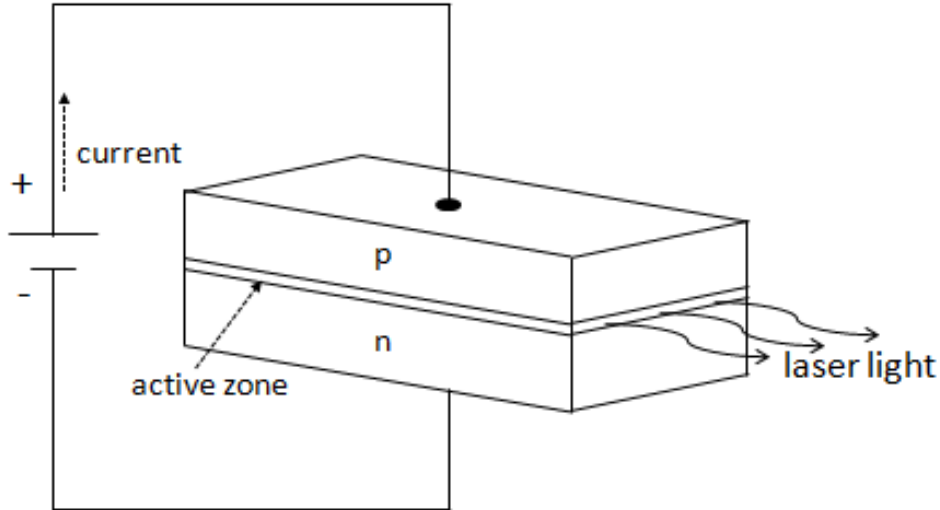


Figure 2: Diagram of a wired pn-junction used in laser diodes.

2.3 Resonators and Modes

Thanks to the polished ends of the laser diode and a grating (see section 2.4) a resonator is built in the diode laser. Several integer numbers of half the wavelength $\lambda/2$ must fit inside the length L of the resonator to enable the formation of standing waves. Only the wavelengths that fulfil the condition of resonance can be strengthened inside the resonator:

$$L = \frac{n \cdot \lambda}{2} : n \in N \quad (2)$$

These possible resonating waves are called modes of resonance. In the ideal case only one of these modes should be strengthened (*Single – Mode*). There are two ways to achieve this.

The first way is to have both ends of the laser diode polished, so that the laser diode itself becomes a resonator. This way we overlap two resonators (the laser diode and the resonator built by the grating). Both resonators must fulfil the condition (2). That way, the mode which fulfils (2) for all resonators is selected.

The other way is to exploit the fact that the gain of the laser depends on the wavelength. This way, one of the possible wavelengths of the resonator built by the grating (the only resonator in this case) is selected, the one for which the gain is maximum.

The wavelength for the laser light can be changed in different ways. First of all, the length of the resonator can be changed by moving the grating with a piezoelectric element (section 2.5). It can also be changed by changing the wavelength for which the gain of the laser diode is maximum, because this depends on the temperature, of which we have control, and the current that goes through the pn-juncture, of which we also have control. When using these methods to select a wavelength, it can happen that the laser has to change the strengthened modes in order to fulfil the equation (2). This phenomenon is called *mode hopping* (see section 3.3).

2.4 External Resonator

One of the most important types of diode lasers is the *External Cavity Diode Laser* (ECDL). In this kind of laser, a grating is used to form an external cavity. A blazed grating (see Figure 4) is a mirror which surface has the shape of a stairway. That enables the laser light to be back-reflected to the laser diode and simultaneously be forward-reflected to follow its way. This kind

of cavity is called *Littrow cavity*. The Littrow cavity imposes the following grating equation:

$$2d\sin\theta = m\lambda \quad (3)$$

where d is the distance between two neighbouring stairs in the grating, θ is the angle between the incident and the reflected beams, m is an integer, and λ the wavelength. In the zero order, the angle will be $\theta = 0$ and it will be back-reflected into the laser. The 1st order beam will be forward-reflected and will follow its path.

As it can be expected from equation (3), if we change the angle θ , the wavelength λ of the 1st order laser light will change. This feature can be exploited in order to control the wavelength of the laser light.

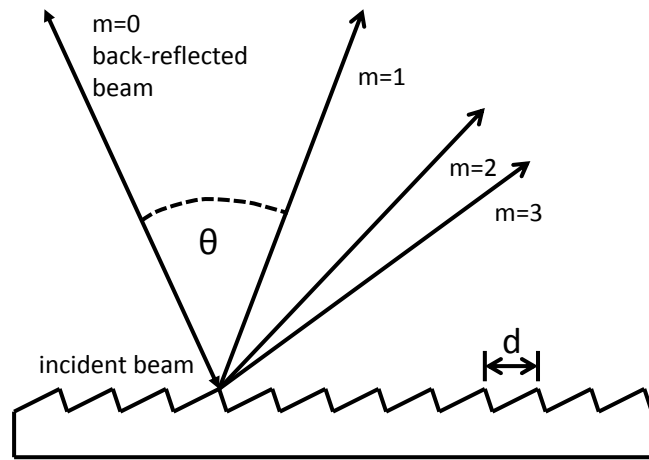


Figure 3: Diagram of a blazed grating with different order beams.

2.5 Piezoelectric Element

A piezoelectric element³ is a material of which length changes depending on the electrical voltage it is bound to. The electrical tension is given to the piezoelectric element by a High Voltage Amplifier. If the piezoelectric element is connected to the grating, it enables a very precise control of the angle of the grating. As the angle of the grating is related to the wavelength of the laser. The length of the resonator built by the grating can be finely tuned by changing the length of the piezoelectric element, i.e. by changing the voltage through it, this way, we can precisely control the wavelength of the laser light. However, this can only be controlled inside the limits of the maximum and minimum lengths of the piezoelectric element.

This will be an important element in this project, because our aim is to control the wavelength of the laser and the Piezoelectric element enables us to control that in a very precise way.

³Piezoelectric element used: Thorlabs AE0203D04F [4]

2.6 Collimation

The emitted light from the laser diode is divergent. In order to collimate it, a collimating lens is installed in the end of the laser. The focal length of this lens is 1,6 mm. The position of this lens can be finely changed with a screw. In a certain position, the beam will be collimated.

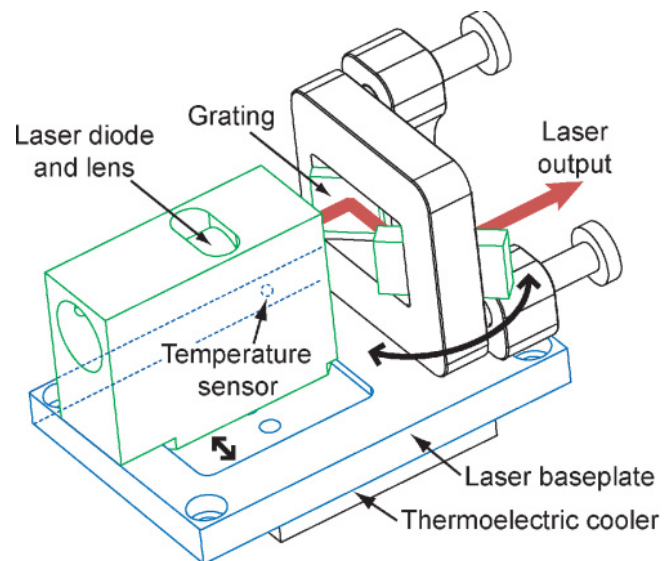


Figure 4: Diagram of an External Cavity Diode Laser.[2]

3 Characterisation of the Laser Diode

3.1 Backcoupling

We give a feedback to the laser diode with the help of the grating. When a certain fraction of the outgoing laser light reflects back into the laser diode, it will be called backcoupled. When the laser is backcoupled, the power given to the laser is the minimum needed so that the laser starts to emit a stimulated emission.

In order to backcouple the laser a triangular current signal has been given to the laser diode and its output intensity is measured continuously with a photodiode⁴. By carefully rotating the screws that move the mirror and the grating that are in front of the laser diode. Thanks to the setup that explained above, it can be seen whether the laser is nearer or further from being backcoupled. In Figures 5 and 6 the difference of the non-backcoupled and backcoupled laser can be seen.

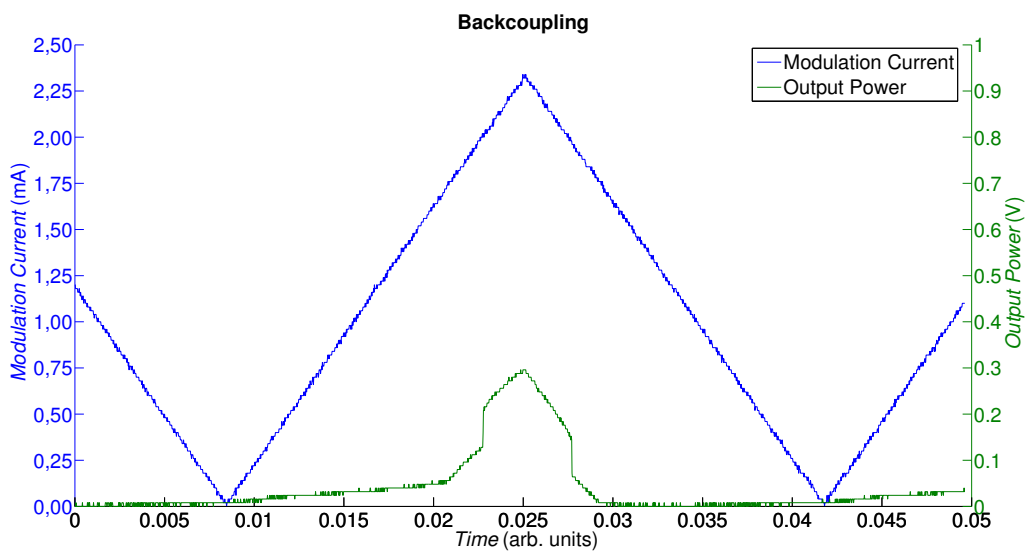


Figure 5: Backcoupling. In this Figure we can say that the laser is not properly backcoupled, because it starts to emit when the input power is very large.

⁴Photodiode used: Thorlabs DET-36A/M[5]

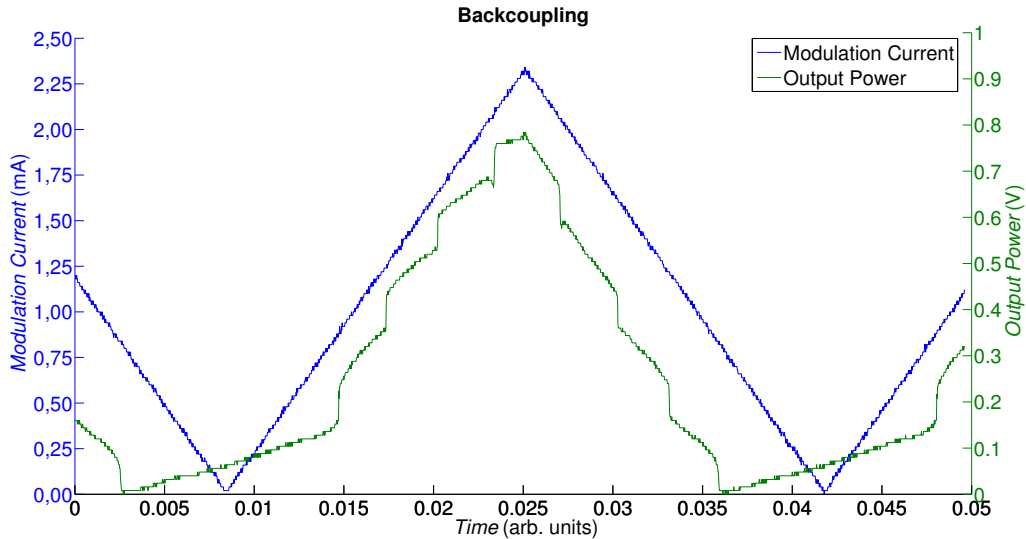


Figure 6: Backcoupling. In this case, the laser is properly backcoupled because the laser starts to emit already with a small input power given to the laser diode.

3.2 Threshold Current and Output Power

The threshold current is an important feature of laser diodes. For low currents, the laser diode emits spontaneously. As the current increases, the spontaneous emission increases gradually until the point in which it starts to emit stimulated radiation. The threshold current is the current value at which this phenomenon starts. If the threshold current is low, the laser will be more effective.

The output power has a correlation with the current injected into the laser. If the current is below the threshold, the laser diode will radiate spontaneous radiation, therefore the output power will be small. However, over the threshold current, the output power will be much bigger and will increase as the current increases.

In Figure 7, I have taken the two so far explained features and shown how the laser behaves according to them.

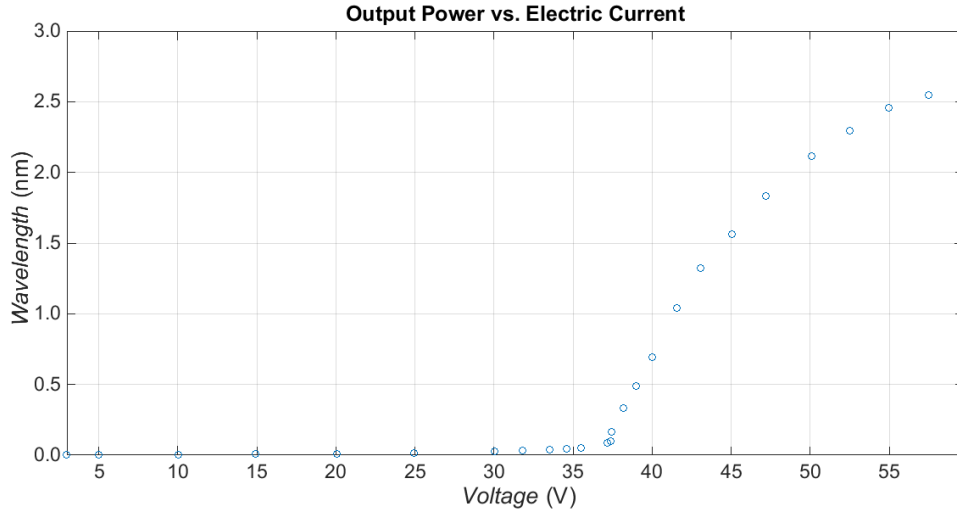


Figure 7: Experimental measurement made changing the input electric current to the laser and measuring the output power with a power-meter. The measured threshold current of this laser is $I = (37 \pm 1)$ mA. It can be seen that for big currents the output power saturates.

3.3 Mode Hopping

The wavelength of the laser light can be controlled by changing the voltage applied to the piezoelectric element. However, one has to be careful because big changes in the voltage can give rise to a mode hopping, i.e. jumping from a mode to another. It always has to be known which mode is wanted and also in which range the laser will hold this mode without jumping to another one.

Experimental measurements of the laser wavelength taken for different voltages for the piezoelectric element are shown in Figure 8. The wavelength has been measured with the WS5 wavelength meter from HighFinesse (5).

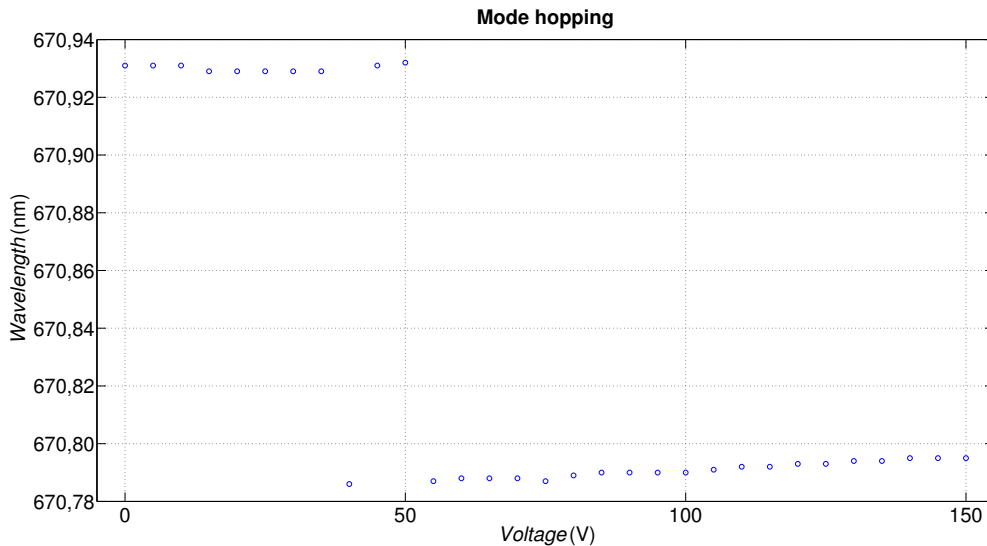


Figure 8: Mode hopping. The wavelength rises gradually with the voltage until the border of the mode is reached and the laser jumps to another mode.

4 Fabry-Perot Interferometer

A Fabry-Perot interferometer⁵ is an optical resonator which can be used as an optical spectrometer. This interferometer has two mirrors inside it. The laser light is incoupled with an optical fibre and it goes into a cavity formed by the two mirrors. The light bounces back and forth inside this cavity creating an interference. The mirror on the right is continuously moved back and forth by a system of piezoelectric elements, this way, the length of the resonator is continuously changed. There will be certain discrete lengths (multiples of half the wavelength) of the cavity where the condition of resonance will be fulfilled so that the light inside the cavity will interfere creating such a strong intensity that part of the light is able to escape through the mirror on the right (Figure 9), which becomes semitransparent for positive interference (resonance). Behind that mirror there is a photodiode that measures the intensity of the light and transmits the data to an oscilloscope⁶. When the condition of resonance is hit, a peak of intensity will appear on the oscilloscope. However, the piezoelectric elements will keep moving the mirror, so the resonance will be destroyed because the light will interfere destructively. That will lead to a drastic intensity loss, so the light will no more be able to get through the mirror, resulting in a null detection of intensity in the photodiode. The mirrors will keep on moving back and forth and letting the light pass when resonating. If the laser is running in single-mode, only one repeated peak will be seen on the oscilloscope if the interferometer is aligned properly.

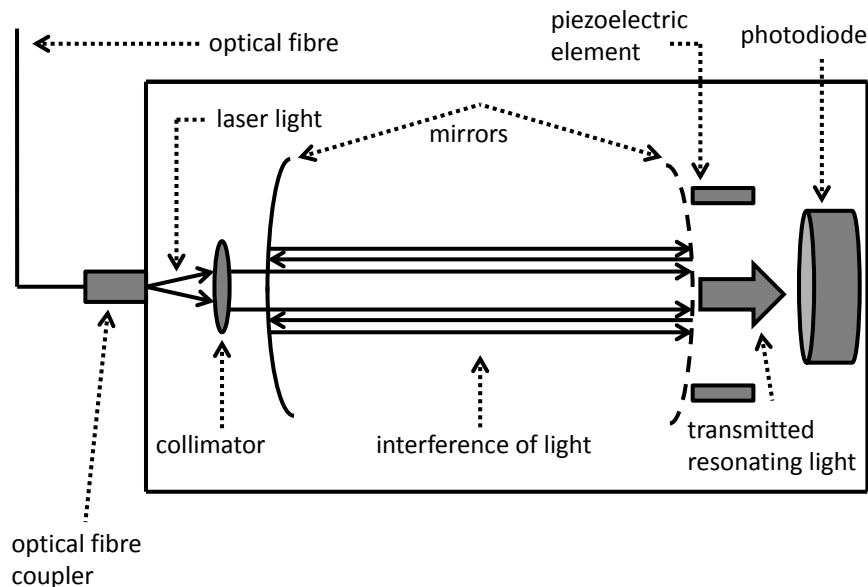


Figure 9: Diagram of Fabry-Perot interferometer.

4.1 Free Spectral Range

The free spectral range is the spacing in optical frequency or wavelength between two maxima in the intensity of the light of the two transmitted beams of the set of mirrors in the interferometer.

⁵Fabry-Perot interferometer used: Toptica FPI100

⁶Oscilloscope used: DWINSTER GDS-1072A-U

The free spectral range in a cavity of length L is:

$$\Delta\nu = \frac{c}{2n_g L} \quad (4)$$

where n_g is the group velocity refractive index of the media within the cavity and c the speed of light.

In wavelength, the free spectral range is expressed by:

$$\Delta\lambda = \frac{\lambda^2}{2n_g L} \quad (5)$$

An experimental measurement made with the laser running in single-mode can be seen in Figure 10.

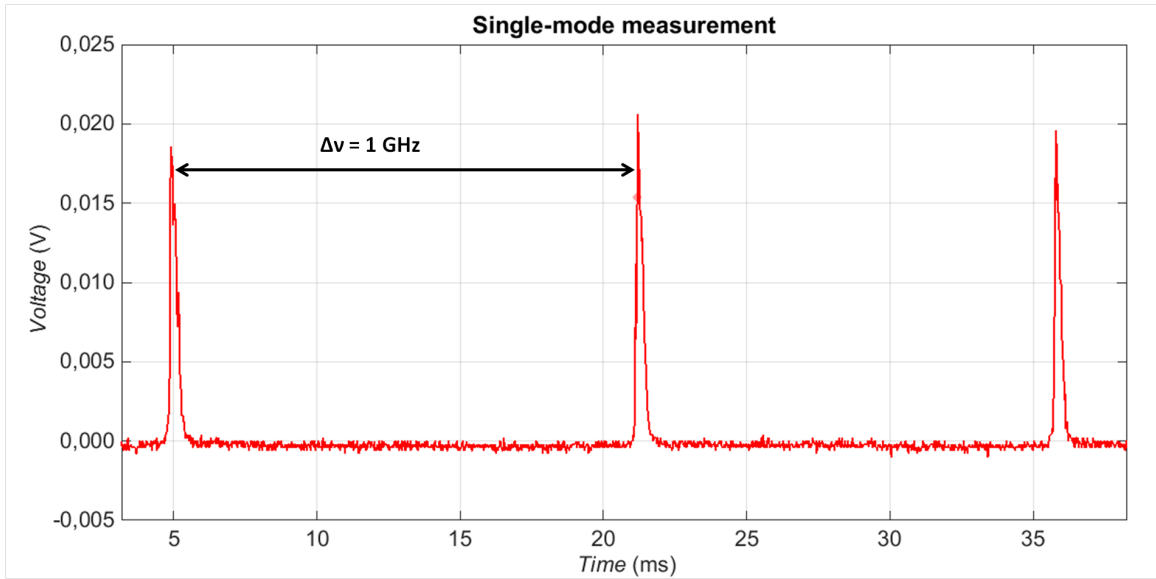


Figure 10: Single-mode measurement with Fabry-Perot interferometer. A single-mode signal can be seen in this capture of the oscilloscope. It can be said that the signal is single-mode because the different peaks are separated by a constant distance and no other mode can be seen in the gap between two peaks. The free spectral range of this interferometer is shown in the Figure and its value is $\Delta\nu = 1$ GHz.

5 Wavelength Meter

The wavelength meter is the main device of this project. Its aim is to provide us with accurate information of the exact wavelength of the laser. In order to measure that, the wavemeter uses an incorporated Fizeau interferometer.

The wavelength meter gets the light of the laser by an optical fibre that is coupled from our optical setup (section 6). Once the light gets into the wavemeter, it is collimated by a mirror (Figure 11). After that, it enters the Fizeau interferometer. This interferometer does not have mechanical moving parts, which is usual in other available interferometers. This is an advantage because it ensures a high accuracy (up to 2 MHz) and a better robustness for the device. The interference pattern created inside the Fizeau interferometer is projected by a cylindrical lens onto an array of CCD photodiodes, which record the interference pattern. The pattern is sent to the computer by USB cable. The wavemeter provides a software that can be installed on the computer. Thanks to this program, the wavelength of the laser can be calculated and transferred to another program -in this case, the LabVIEW program that contains the PID controller-.

In this project I started to use the WS5 wavemeter from HighFinesse[9], of which the precision is 3000 MHz. Later, after seeing that precision was not enough to set the laser in exactly the desired setpoint, we decided to change the wavelength meter to the WS7, which provides us with a better accuracy, an accuracy of 60 MHz according to HighFinesse; but we see that we are actually able to read the wavelength with a much higher accuracy of less than 10 MHz.

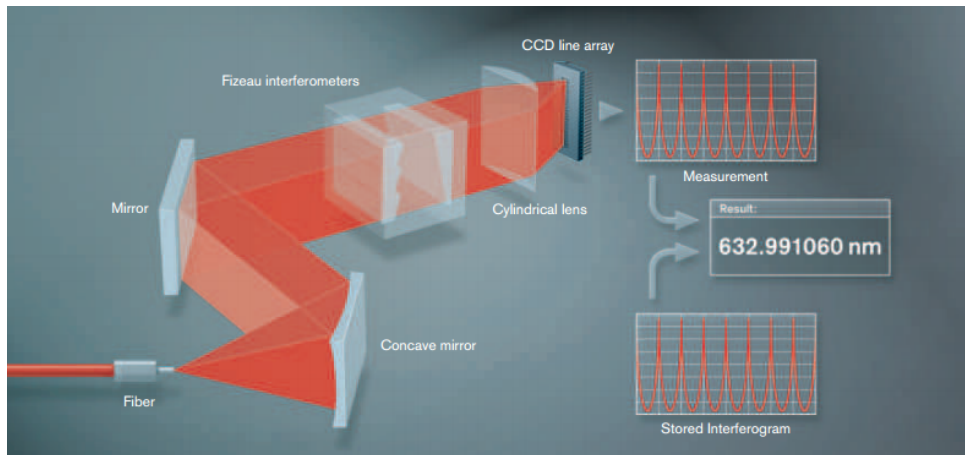


Figure 11: Diagram of the wavemeter. Here it can be seen how the light is coupled into the device by an optical fibre, that it is collimated by a mirror and interferes inside the Fizeau interferometer before sending the pattern to the CCD's.[9]

6 Optical Setup

The optical setup used in this project is shown in Figure 12.

A collimated laser beam comes out of the laser. Our optical setup requires an optical isolator, which allows the transmission of light only in one direction. That is necessary to avoid back-reflected beams that can cause unwanted feedback into the cavity of the laser diode. However, the optical isolator has also the disadvantage that it decreases the power of the beam. I have measured the power of the beam with a powermeter before and after the isolator⁷ in order to check this. After seeing that the power loss was too big, I decided to build a telescope in order to make the diameter of the beam smaller because of the power loss on the aperture of the isolator. That is done to couple the whole beam into the optical isolator, because its aperture is smaller than the section of the non-telescoped beam. This way the output power of the optical isolator will increase. The focal length of the lenses used for the telescope is $f=100$ mm.

The measured powers of the beam are shown in Table 1:

Table 1: Measurements of power around the optical isolator. The input power has been measured directly on the beam that comes out of the laser diode. The output power has been measured after the optical isolator. It can be appreciated that the telescope allows a bigger output power, nearly the double compared to the measurement without telescope.

| Telescope | Input Power (mW) | Output Power (mW) |
|-----------|------------------|-------------------|
| off | 2.2 | 0.8 |
| on | 2.2 | 1.4 |

After the telescope, the optically isolated beam encounters a $\lambda/2$ waveplate, which turns the polarisation plane of linearly polarised light. A beam splitter comes after that. This element splits the incident beam into two sub-beams. Depending on the polarisation of the incident light, the power in each sub-beam changes. One of the beams couples into the optical fibre which is connected to the wavelength meter. The other sub-beam hits a set of mirrors which are used to couple the beam into the Fabry-Perot interferometer.

⁷Optical isolator used: Thorlabs IO-3-671-LD

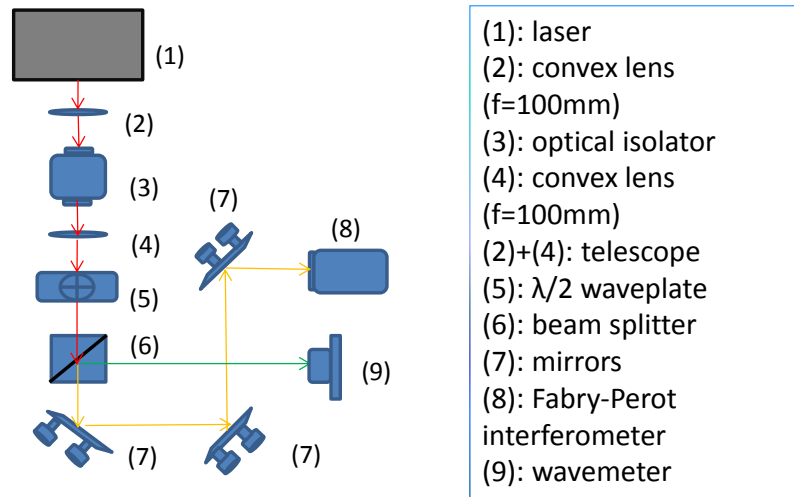


Figure 12: Diagram representation of the optical setup used in this project.

7 Setup for Wavemeter Lock

The setup for the wavemeter lock is shown in Figure 13.

Once the beams are coupled into the interferometer and the wavemeter, the wavemeter measures the wavelength and gives the information to the computer. Meanwhile, with the help of an oscilloscope, the Fabry-Perot interferometer shows whether the laser is running in single-mode or not. The program that connects the wavelength meter with the computer gives the information to a LabVIEW program that contains the PID controller. The LabVIEW program is able to generate an error signal which is transferred to the Arduino Due by USB. The error signal gives the difference between the measured value of the wavelength and a given setpoint. The Arduino is used as a **Digital-to-Analog Converter (DAC)** to convert the digital error to an analog voltage signal. The Arduino is equipped with a 12 bit DAC (compare with section 8 for further information). The signal from the Arduino is given to a HV-Amplifier, which generates the high voltage to drive the piezoelectric element in the laser. The signal from the Arduino is also connected to the Laser Diode Current Controller, which helps to keep the laser running in single-mode in a wider tuning range. Due to the high voltage, the piezoelectric element changes its size, which changes the angle of the grating, producing a slight shift in the wavelength. This process goes in a loop so that we are continuously measuring the wavelength and giving back an error signal to the laser via the piezoelectric element. This way, the wavelength of the laser light can be stabilised around a given setpoint.

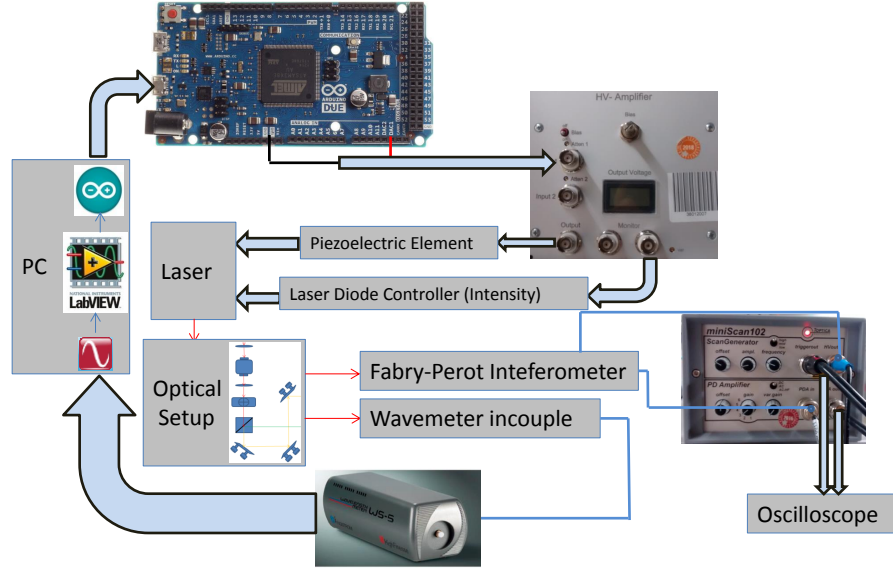


Figure 13: Diagram representation of the setup for the wavemeter lock

8 Arduino Due vs. NI USB-6008

We want to use a **Digital-to-Analog-Converter (DAC)** in order to take the error function from the LabVIEW program and convert it to an analog signal for the HV-Amplifier that controls the piezoelectric element and the laser current. I have two possible DAC's available, which are candidates to do this job. One of them is the NI USB-6008 I/O device from National Instruments and the other one is the Arduino Due. Both of these DAC's have USB connection, so they can be easily connected to the computer. I have taken into account the speed, resolution and price of these DAC's to choose one of them.

8.1 Resolution

The resolution of the NI device is 12 bits and its output-voltage range goes from 0V to 5V. So the voltage per step is:

$$\frac{\Delta U}{\text{steps}} = \frac{5 \text{ V}}{12 \text{ bits}} = \frac{5 \text{ V}}{2^{12}} = \frac{5 \text{ V}}{4096} = 1.2207 \text{ mV/step} \quad (6)$$

In the case of the Arduino, the resolution is also 12 bits and the theoretical output-voltage range goes from 0 V to 3,3 V. However, when I measured that, a strange offset could be seen in the voltage, which shrinks the range. The actual minimum voltage is 540 mV and the maximum 2800 mV. So that the actual width of the output voltage is:

$$\Delta U = 2.80 \text{ V} - 0.54 \text{ V} = 2.26 \text{ V} \quad (7)$$

Taking that into account, the voltage per step is:

$$\frac{\Delta U}{\text{steps}} = \frac{2.26 \text{ V}}{12 \text{ bits}} = \frac{2.26 \text{ V}}{2^{12}} = \frac{2.26 \text{ V}}{4096} = 0.5518 \text{ mV/step} \quad (8)$$

On the one hand, the potential output range of the NI device is wider than the one of the Arduino, which might be a positive point for the NI device. However, the voltage per step is smaller in the Arduino, which can result in a more precise tuning. Nevertheless, these features loose importance because the DAC's are connected to the HV-Amplifier, which allows to set a gain of the input signal.

The important feature here is the resolution, which is 12 bits for both devices, so in this case it is the same which one we use.

8.2 Speed

In order to check the speed of the DAC's I give them an input of a sine function programmed in either LabVIEW or in Arduino and use an oscilloscope to see their output. Figures 7 and 8 show measures made with the oscilloscope.

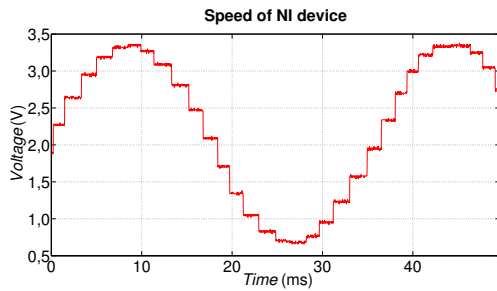


Figure 14: Oscilloscope capture for the analog output of the NI microcontroller.

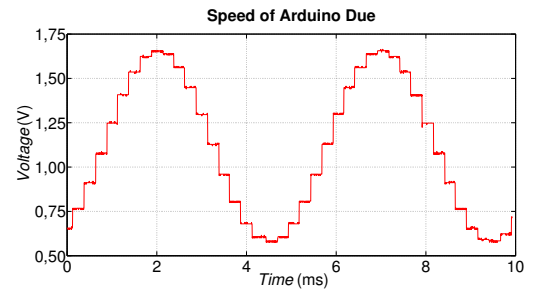


Figure 15: Oscilloscope capture for the analog output of Arduino Due.

The speeds of the NI device and Arduino can be calculated from Figures 14 and 15 respectively. Speed of the NI device:

$$\nu = \frac{5 \text{ values}}{8.4 \text{ ms}} = 0.595 \text{ kHz} \quad (9)$$

Speed of Arduino Due:

$$\nu = \frac{5 \text{ values}}{1.240 \text{ ms}} = 4.032 \text{ kHz} \quad (10)$$

It can be seen that the Arduino Due is much faster than the NI device.

The speed of the device is important for the wavemeter lock. The laser will constantly have perturbations. The frequencies of these perturbations can happen in different ranges. For example, the frequencies for perturbations due to changes in temperature are in the range of some Hz or lower. However, mechanical disturbances can create perturbation of some Hz to some kHz, which are more difficult to control if the speed of the DAC is not high enough.

In our case, the wavelength meter takes data with a frequency of around 100 Hz, so in principle both of the DAC's should be able to handle that. However, it is more convenient to have a fast device that really ensures an effective handling when regulating perturbations of high frequency.

There can even be perturbations with higher frequencies, noise for example. But these are impossible to control by these devices, because I am taking data and creating error signals in a smaller frequencies that those perturbations.

8.3 Price

The price of the Arduino Due is 36.00 Euro.

The price of the NI USB-6008 I/O device is 184.00 Euro.

Table 2: NI device vs. Arduino Due

| Device | Resolution (bit) | $\Delta V/\text{byte}$ (mV) | Speed (kHz) | Price (Euro) |
|---------------|-------------------------|---|--------------------|---------------------|
| NI USB-6008 | 12 | 1.2207 | 0.595 | 184.00 |
| Arduino Due | 12 | 0.5518 | 4.032 | 36.00 |

Considering the features of both DAC's, I have decided to use the Arduino Due because it is faster and cheaper. Moreover, the smaller voltage range can be compensated by the input gain of the HV-Amplifier.

9 PID Controller

A **Proportional-Integral-Derivative** Controller (PID) is a mechanism that calculates the error between a given setpoint and a measured variable and tries to minimise the error by giving a feedback to the system.

The PID controller method uses three control parameters: the proportional (P), integral (I) and derivative (D) values. It can be said that these terms control the errors in different periods of time: P takes care of the present error, I accumulates the past error and D predicts possible future errors. Each system requires a different set of these three parameters. There exist several different theoretical algorithms to find the most suitable parameters, but it is really hard to do that this way. That is why these parameters are usually fixed by trial and error. This can be made by setting some initial values to the parameters and noticing how the system reacts. Then, one can change the parameters depending of which features of the error signal ought to be strengthened and which suppressed. This is done repetitively until the error signal is satisfactorily optimised. At this point, the desired variable of the system should quickly take the setpoint value and ought to be finely stabilised around that value. Should the setpoint be changed, the PID should react and try to bring the system back to the setpoint. Should a change occur in the system, then the PID controller should take the system back to the desired setpoint.

Table 3 shows the features of each parameter of the PID controller:

Table 3: Features of P, I and D

| Control Type | Rising Time | Overshoot | Stabilisation Time | Error in Stationary State |
|---------------------|--------------------|------------------|---------------------------|----------------------------------|
| increase of P | decreases | increases | small change | decreases |
| increase of I | decreases | increases | increases | eliminates |
| increase of D | small change | decreases | decreases | small change |

9.1 Programming a PID Controller

In order to program the PID controller, I have used a combination LabVIEW and Arduino. When I did this, I faced some difficulties. There were programs available in internet that enabled the communication between LabVIEW and Arduino, but they did not work, so I had to program the connection myself. However, LabVIEW gave outputs to Arduino in form of characters, so taking into account the features of ASCII code I could translate them to numbers. The LabVIEW program does most of the work. It gets the information from the wavelength meter and gives it to the PID controller. I have used a built in PID controller *vi* from LabVIEW. This has the following main inputs: the measured value of the wavelength, the setpoint for the wavelength, and the parameters for P, I and D. Its output is an integer between 0 and 4095, which is the range of the output of the Arduino (12 bits). Then I have to convert this number into characters because that is required by the VISA.vi that connects this program to the Arduino program.

Once the connection is made, the Arduino program just translates the characters back to integers. Then, it sets a voltage into a given analog output pin depending on the value of the mentioned PID output.

10 Experimental Data for the Wavemeter Lock

A well programmed PID should have some certain characteristics. In this case, the PID controller should lock the wavelength of the laser diode in a given setpoint for a long time. Apart from that, if the system undergoes a perturbation, the PID controller should bring the value of the wavelength back to that of the desired setpoint.

To check that, I have done two main kinds of measurements. In the first one, I have let the laser run along the PID controller for a long period of time and I have set the setpoint in a constant value, so that the PID controller keeps the value of the wavelength around that value. On the second kind of measurement, I have made fast changes in on the setpoint in order to see whether the system reacts to these changes and how fast it reacts.

I have done different measurements for different values of the P, I and D parameters.

In the beginning, I started to use the WS5 wavelength meter, but its error was too big (± 0.001 nm) because it was the same as the wanted value of the wavelength (670.997 ± 0.001) nm. Because of this, we decided to use the WS7 wavelength meter, which error is $\pm 10^{-6}$ nm, much smaller than the error of the wanted wavelength, which enables a very accurate precision around the wanted value.

10.1 Long-term Measurements

In this section, I show the measurements for long-term measurements of the PID controller. Because of memory limitations, I have measured data every 30 seconds for some hours.

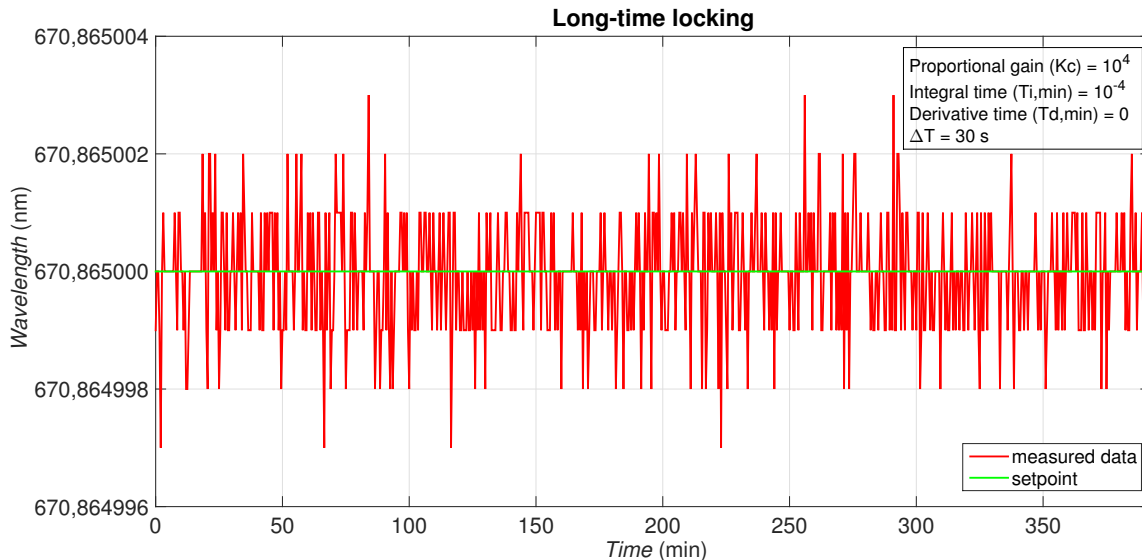


Figure 16: Long-time measurement for wavemeter lock. The setpoint is shown in green and the measured wavelengths in red. The P,I and D parameters used are shown too.

Even if the PID parameters used are different in each measurement, we can say that the two sets of parameters used in Figures 16 and 17 are appropriate for long-time wavelength locking because the system hold for a long time around the given setpoint. In Figure 17, there is a point where the measured wavelength is quite different to the setpoint. That could have happened because of some perturbation around the system. However, the PID brings the system back to the setpoint very fast. To sum up, we can say that my wavemeter lock works very fine. The wavelength is driven to the wanted setpoint and lock around it.

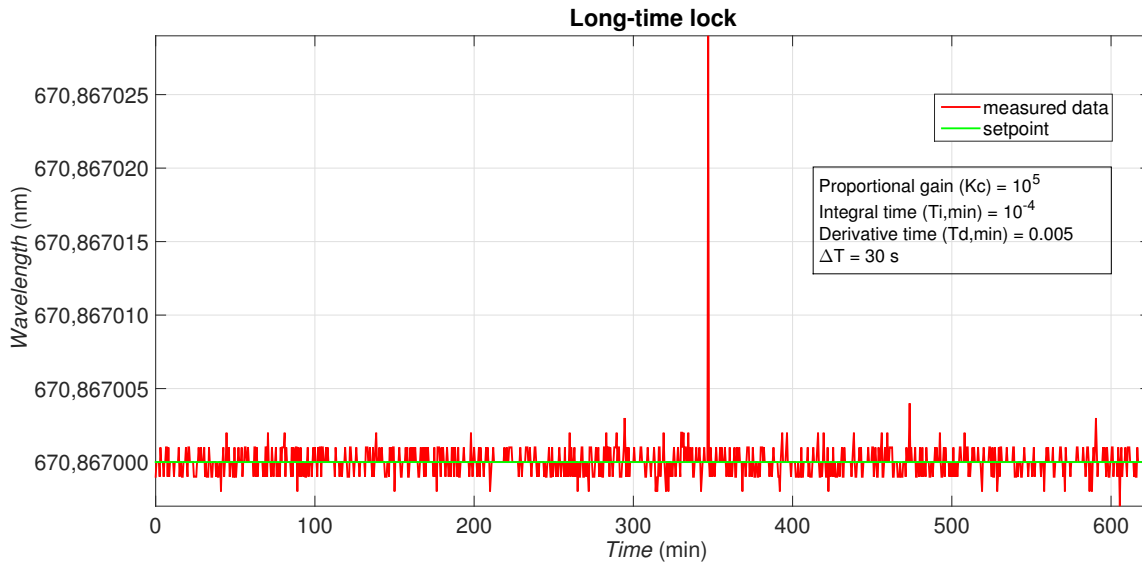


Figure 17: Long-time measurement for wavemeter lock. The setpoint is shown in green and the measured wavelengths in red. The P,I and D parameters used are shown too.

10.2 Measurements for Step Function Response

In this section, I show the measurements for fast changes in the setpoint in order to observe how the system reacts. In Figures 18, 19 and 20, I show the graphs of data taken for three different configurations of the P, I and D parameters. I will discuss how the different parameters affect the reaction of the system. These configurations have been the ones that adapt fastest the system to a change in the setpoint.

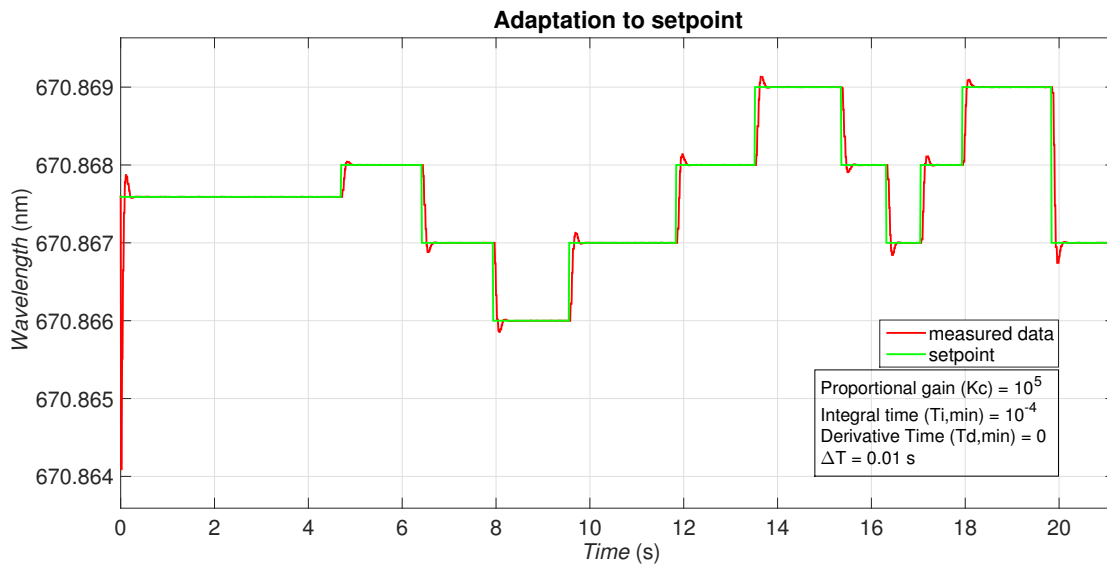


Figure 18: Fast adaptation to setpoint. The *proportional gain* is optimal and the *derivative time* is switched off.

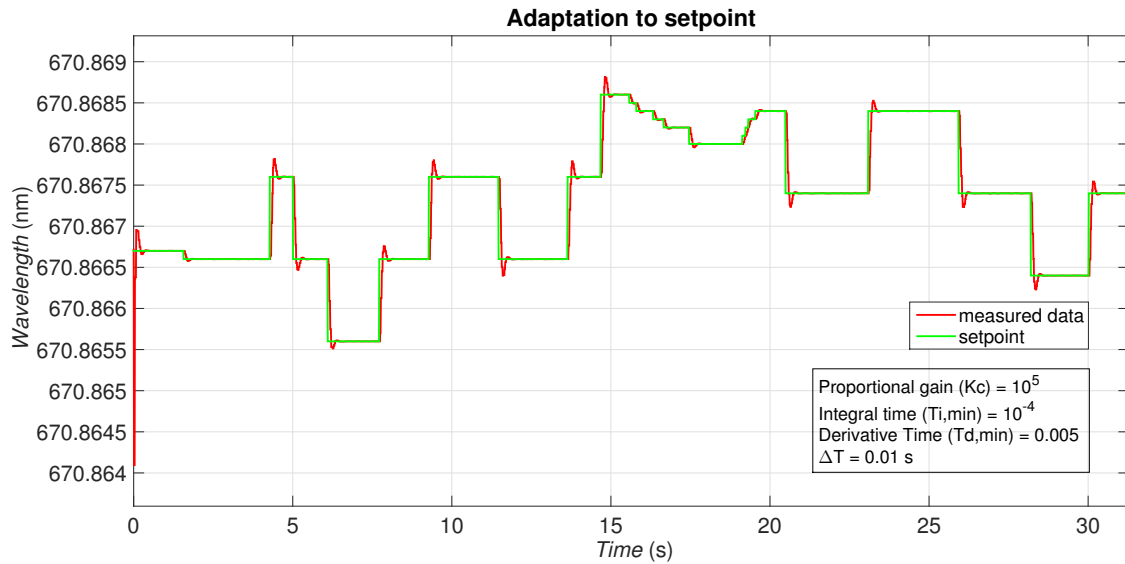


Figure 19: Fast adaptation to setpoint. The *proportional gain* is optimal and the *derivative time* is small.

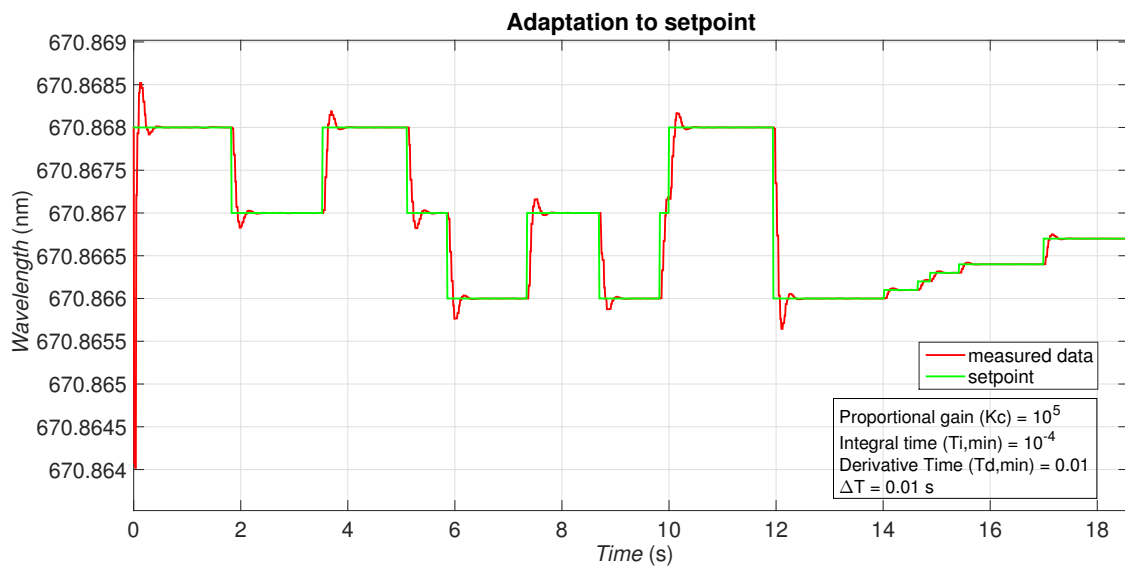


Figure 20: Fast adaptation to setpoint. The *proportional gain* is optimal and the *derivative time* is switched on.

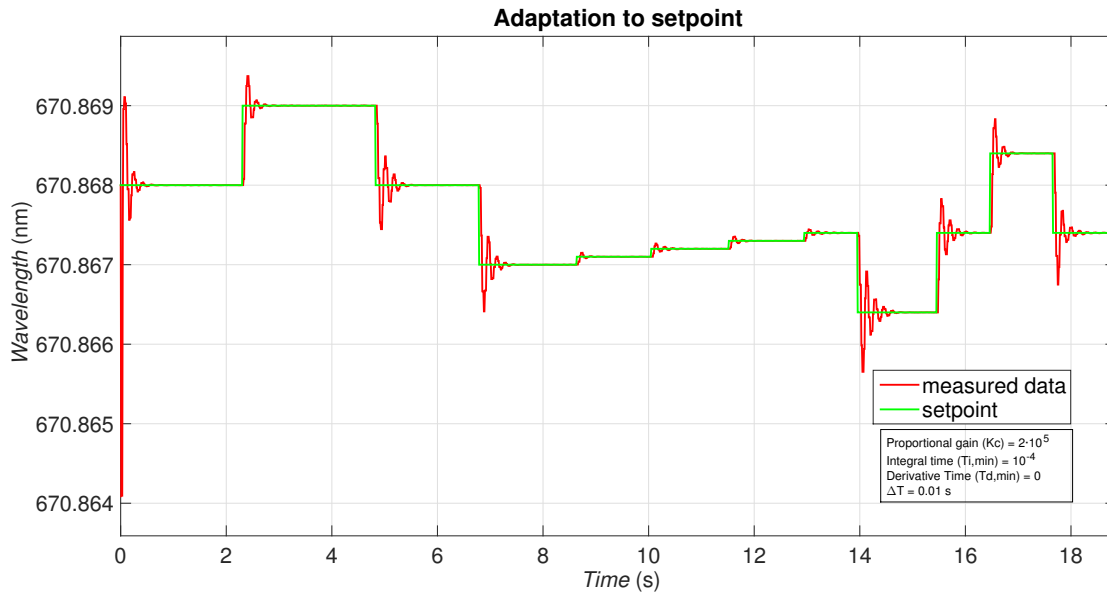


Figure 21: Fast adaptation to setpoint. The *proportional gain* is too big and leads to overshoots.

We can see that the different sets of parameters make the system react in different ways. In Figure 21, the *proportional gain* (K_c) is too big, and that leads to large overshoots, so it takes longer until the system stabilizes. This overshoot could be decreased if the other parameters were appropriate, but nevertheless, the reaction would be slow.

The K_c parameter in Figures 18, 19 and 20 is optimal for the system. It does not make sense to add the *derivative time* parameter because its aim is to prevent the errors that can be accumulated in the future, but as I change the setpoint at random times, it is impossible for this parameter to predict anything. As it can be seen in Figures 19 and 20, if we add that parameter, the system needs more time to change the wavelength, so even if there is no overshoot the PID controller is slower.

To conclude, we can say that the optimal set of parameters is the ones used in Figure 18 because the system reacts fast, there is no overshoot and after reaching the setpoint value, the wavelength sticks accurately to the setpoint.

Another remarkable fact that can be appreciated in these Figures is that in the beginning, the value of the wavelength is the one the laser diode has before switching on the PID controller. However, it reacts and changes fast to the given setpoint.

11 Digital Mirror Device: LightCrafter 4500

In this part of my Bachelor Thesis, I have learned how the LightCrafter 4500 Digital Mirror Device[7] works.

DMD's are used in commercial projectors. The projections are done by using three LED's of different colors (red, blue and green). The light of the LED's shine on arrangement of very small mirrors which act as pixels.

The experimental aim of using this device is being able to project potential surfaces, this way we would be able to put the atoms in the points where the minima of the potential surface. Thanks to that, we may display atoms in desired patterns in lattices created by those potentials.



Figure 22: Image of Digital Mirror Device [10]

12 Arrangement of Mirrors in the DMD

Each of the mirrors from the DMD acts as a unique pixel. Instead of being arranged in a matrix-like way, the mirrors are configured in a diamond-like way. There are 1039680 mirrors arranged in 912 columns by 1140 rows (Figure 23).

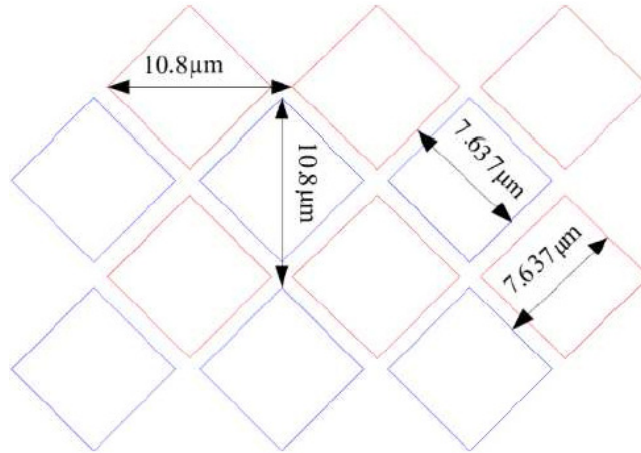


Figure 23: Arrangement of mirrors. The mirrors are arranged in diamond-like configuration. The length of the periodic arrangement from mirror to mirror is $10.8 \mu\text{m}$ in the direction of the main diagonals of the mirrors and $7.637 \mu\text{m}$ in the side-direction of the mirrors. These directions are easier to understand in the diagram.[11]

12.1 Flipping of Mirrors

The mirrors in the DMD can be individually switched on and off depending on their position. The mirrors can flip from on to off thanks to a torsion hinge (Figure 24). The position of the mirror is set by electrodes which create attractive or repulsive forces depending on the wanted position.

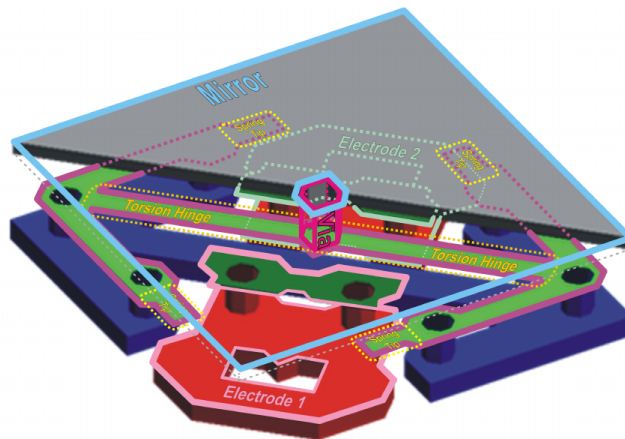


Figure 24: Diagram of mechanical setup of mirrors. Each mirror can turn around the torsion hinge, the hinge is located through one of the diagonals of the mirrors, so that diagonal is the torque axis. The position of the mirrors is set by electrodes which create attractive or repulsive forces. These electrodes are set in the opposite corners to the torque axis diagonal. The two operational positions(on- and off-states) are -12° and $+12^\circ$. When the DMD is off, the mirrors are positioned in the intermediate position 0° because of the lack of power supply.[11]

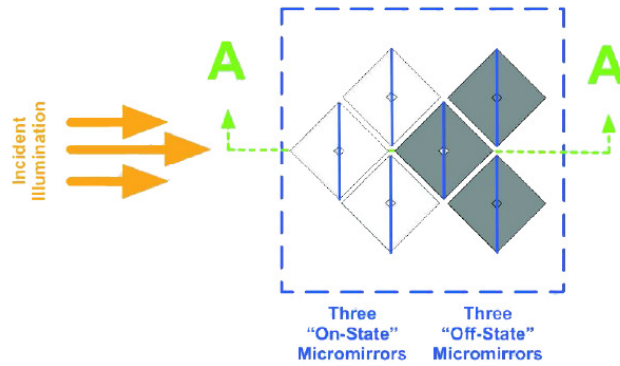


Figure 25: Thanks to the diagonal-like arrangement, each mirror can be flipped around the axis represented by a vertical line. As the mirrors are illuminated by the LED's, their position determines whether the reflected light will be projected on the screen.[12]

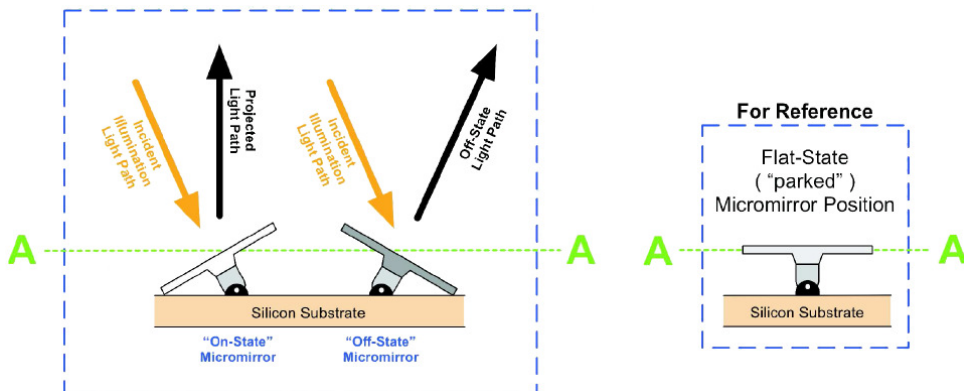


Figure 26: The incident LED light comes from the left side, so if a mirror is to be on, it will flip in the “on-state” direction. On the contrary, if a mirror is to be off, it will flip towards the “off-state direction”. When the power is off, the mirrors will be positioned in the “flat-state”.[12]

13 User Interface

In order to control the DMD device, a software called DLP LightCrafter 4500 Control Software is available. Using this I have taken several data in order to understand how the device works. In the **system control** area, one can see if everything is working fine. The DMD device has to be connected to the computer by USB connection. In order to have a feedback from the computer, the DMD must be connected to a power supply. The DMD must have a constant voltage supply of 12 V, so I have supplied the DMD with a fixed voltage of 12 V and a current of 10 A.

The LightCrafter 4500 device makes projections using three LED's: red, blue and green. The **LED Driver Control** box enables us to set the current going through each of the LED's, this way, one can strengthen or weaken each colour as desired. The values of the current go from 0 to 255. The default values are 104 for red, 134 for green and 130 for blue. This values are set this way because in order to achieve the perfect white balance. The white balance is the mixture of the primitive colours (red, blue and green) that gives white. If the proportion of the primitive colours differs a little from the white balance, more greenish, blueish or reddish images will be achieved. If we see that the projected light has a bad white balance, the proportions of the current can be reset.

13.1 Displaying Images

In this section I will explain how images can be projected by LightCrafter 4500. This could be used in experiments in order to create any desired potential distributions used for example to order atoms in space.

13.1.1 Firmware

Image files can be loaded in a firmware so that the DMD projects them. All that can be done inside the *Image/Firmware* nterface.

13.1.2 Add Images to Firmware

Once inside the *Image/Firmware* interface, go to the *Add Images to Firmware* section. There, a firmware file can be selected using the browser, or a new one can be created. Clicking *Add*, new images can be added to the firmware. The images have to be *bmp* files of dimensions 912 x 1140 -the same dimensions as the mirrors of the DMD-. After adding so many images as wanted, the firmware can be saved by clicking on *Save Updates* and selecting the desired firmware file -the extension of the firmwares is *.bin*-.

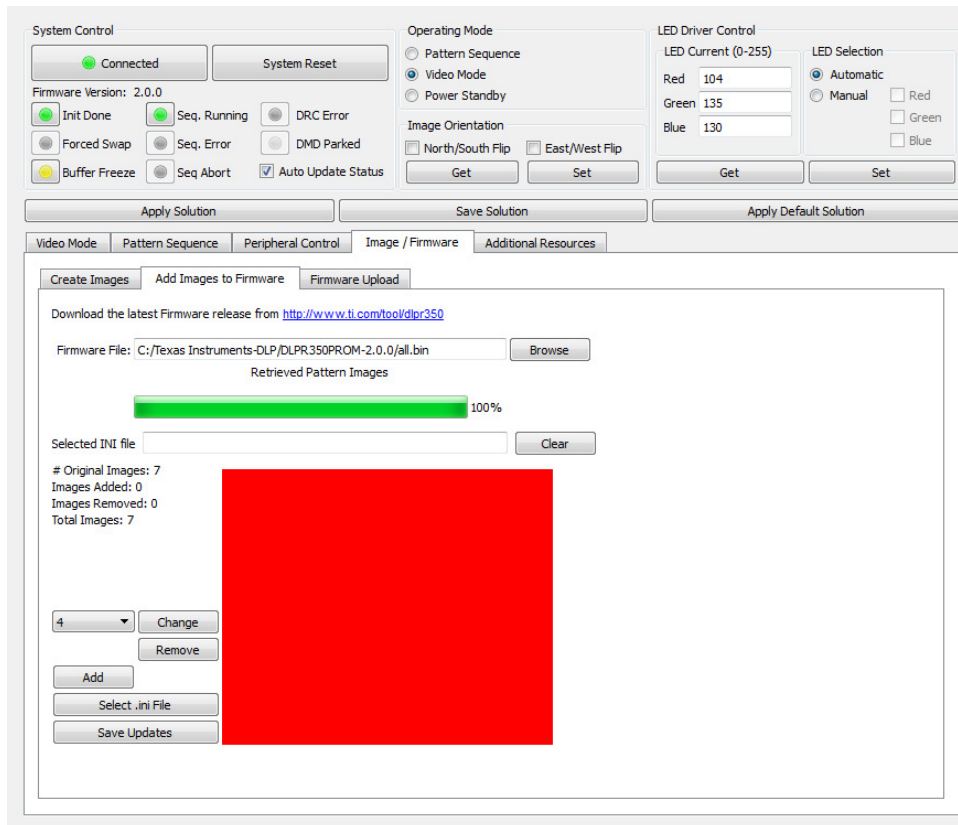


Figure 27: How to add images to firmware on the user interface of the DLP LightCrafter 4500 Control Software. *Image/Firmware* → *Add Images to Firmware* → *Add* → *Save Updates*.

13.1.3 Firmware Upload

The programmed firmware has to be transferred to the DMD device. The desired firmware file can be browsed and then it can be uploaded by clicking *Upload*. This process can take a couple of minutes.

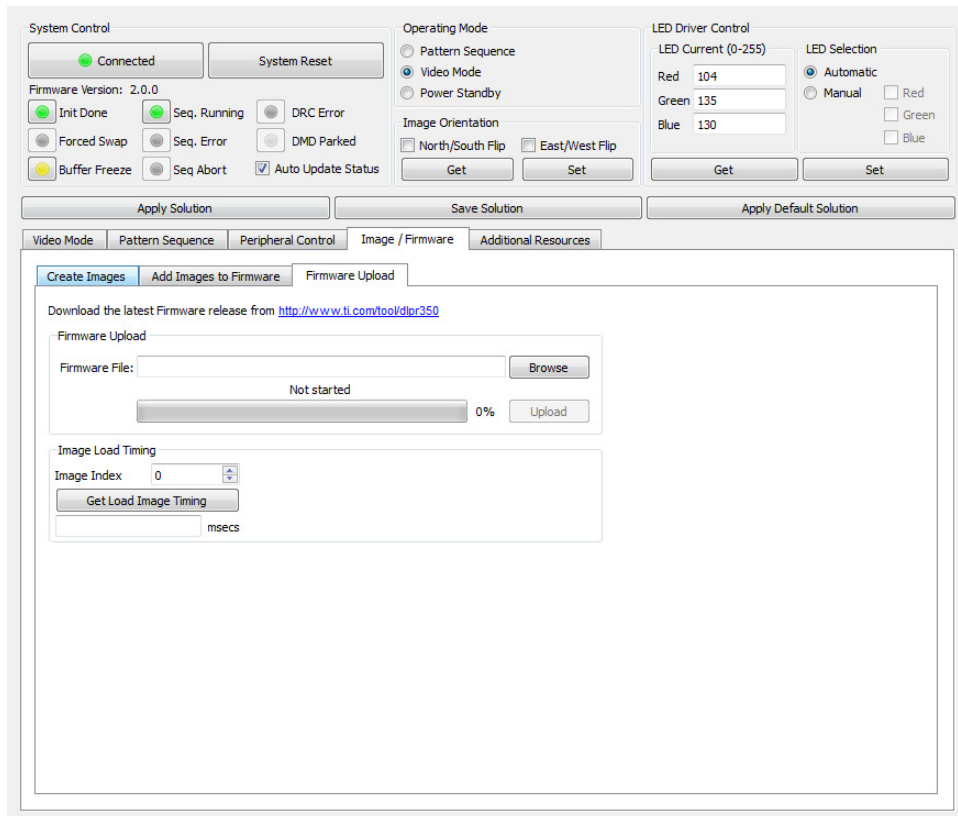


Figure 28: How to upload a firmware on the user interface of the DLP LightCrafter 4500 Control Software. *Image/Firmware* → *Firmware Upload* → *Browse* → *Upload*.

13.1.4 Video Mode

I have used the video mode so that the DMD device projects selected images from the firmware. In order to do that, I have selected *Images from Flash* from the *Source Select* box from the *Input Configuration*. Once that is selected, some integer numbers can be selected, each number corresponds to an image from the firmware.

14 Experimental Data

In order to learn more about how the LightCrafter 4500 work I have made measurements to see how the flipping of mirrors and the on and off switching of LEDs work. In order to do that, I have taken the following steps in the user interface.

I have used another feature of the **Video Mode**. I have selected the *Internal Test Pattern* and *Solid Field* mode from *Source Select*. After setting it, a single monochromatic colour will be projected. The colour can be chosen using the *Internal Test Pattern Color* box. For the purposes of my measurements, I have set different values -from 0 to 999- of the *Foreground* for the different LED's - red, green and blue-.

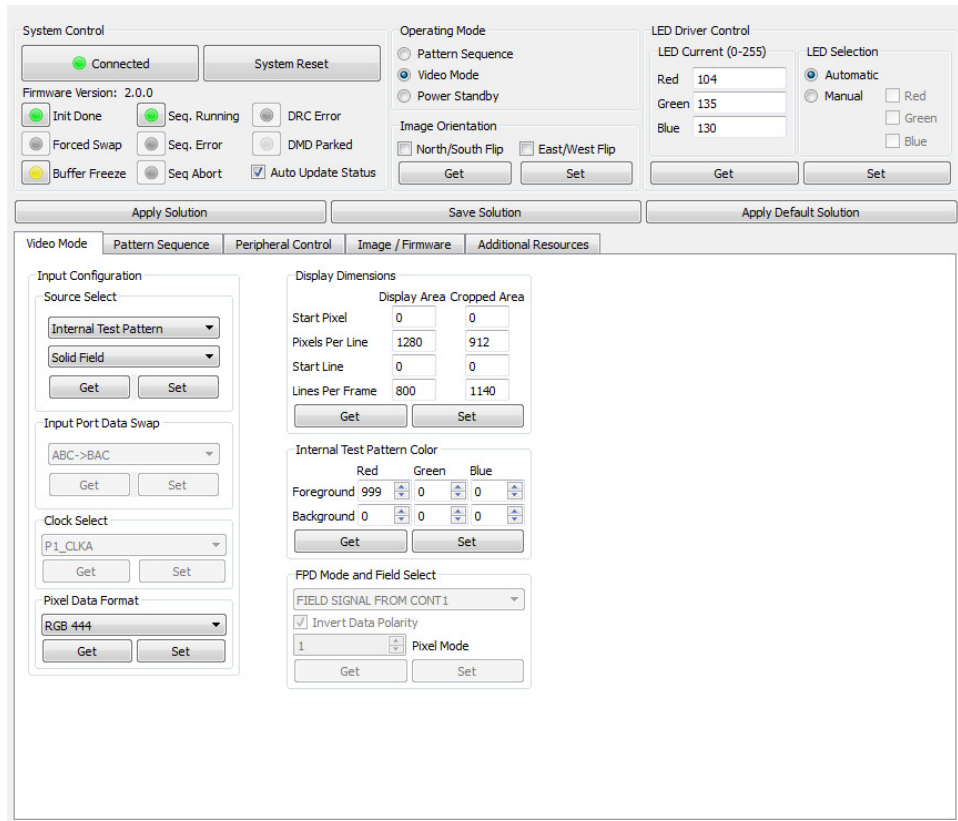


Figure 29: How to get a solid field of any wanted colour in the user interface. *Video Mode* → *Source Select* → *Internal Test Pattern* → *Solid Field* → *Internal Test Pattern Color* → *Foreground*.

I have measured the power of the light for different settings by using a photodiode and taken the data with an oscilloscope. The data taken in this way can be seen in the following Figures.

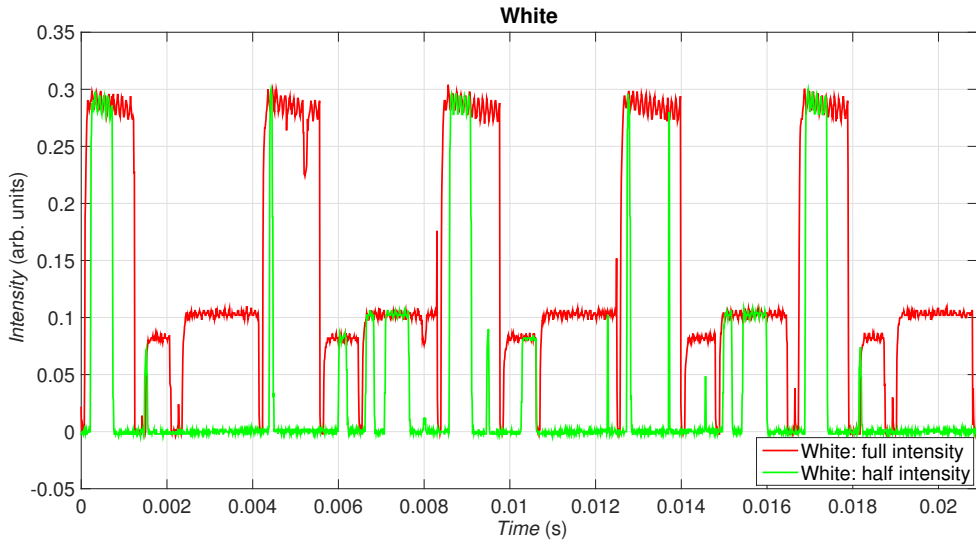


Figure 30: This data acquisition has been performed for white and greyscale light. In order to project white, I have set the power on the *Foreground* of the three LED's to the maximum, i.e. red=999, blue=999, green=999. In order to project the greyscale or half-white light, I have set the power of the LEDs to the half, i.e. red=500, blue=500, green=500. As we can see in the plot of the full-white light, the projection time is split in periodical intervals. Each interval is split into three subintervals, where each one corresponds to a primitive colour, the intervals of each colour are easily detectable because each colour has a different power. So it can be said that each LED has its own interval of time to be on. Looking at the plot of the half-white projection, it can be said that once each LED is on, a half-powered light can be obtained by switching the mirrors on and off.

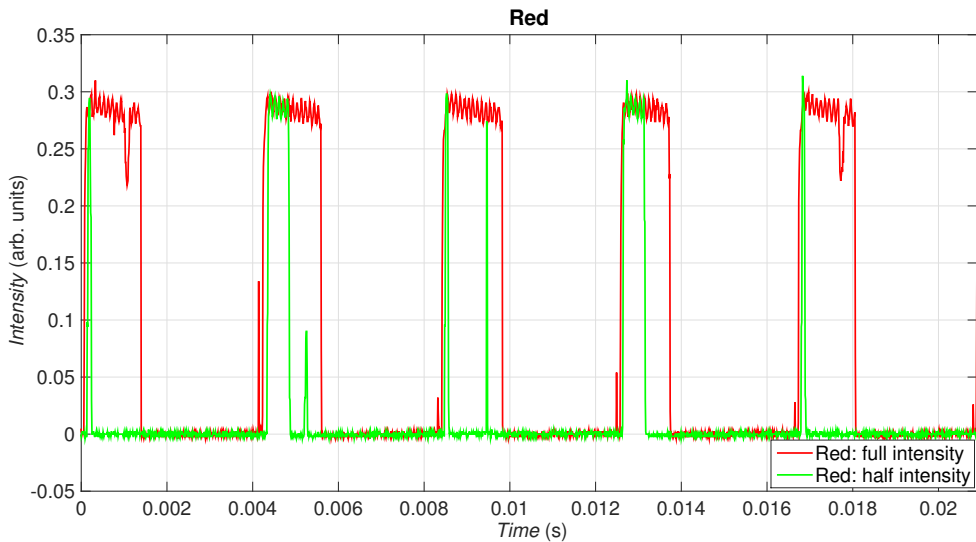


Figure 31: Measurement with full and half intensity for red.

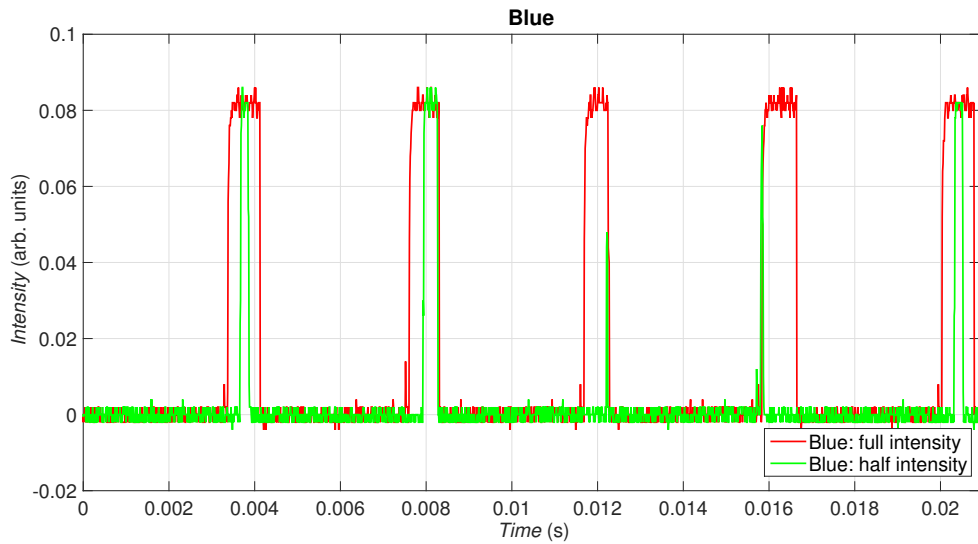


Figure 32: Measurement with full and half intensity for blue.

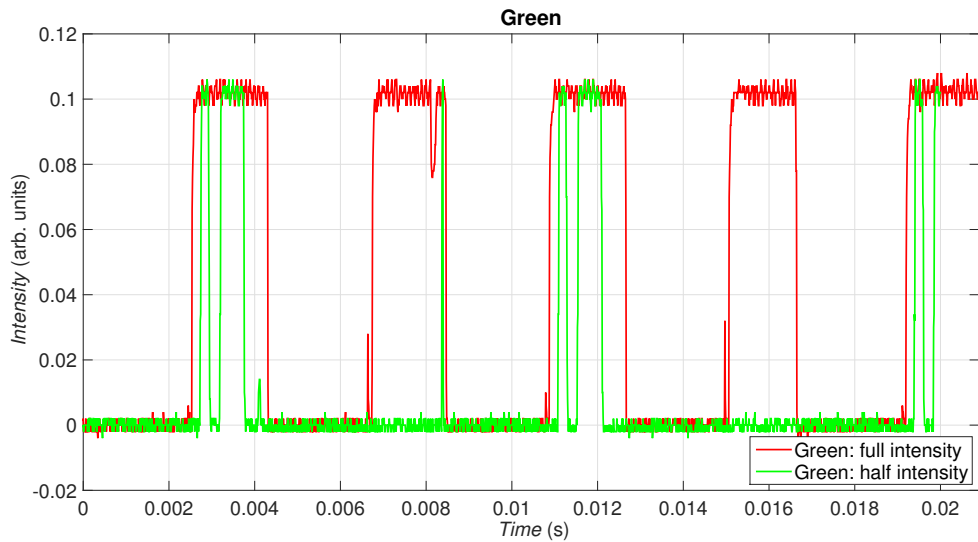


Figure 33: Measurement with full and half intensity for green.

Figures 31, 32 and 33 are plots of projections with pure primitive lights. As explained in Figure 30, each LED has its own allowed time to be on. It can again be seen that each colour has a certain maximum intensity: red has the biggest, then comes green and blue has the smallest.

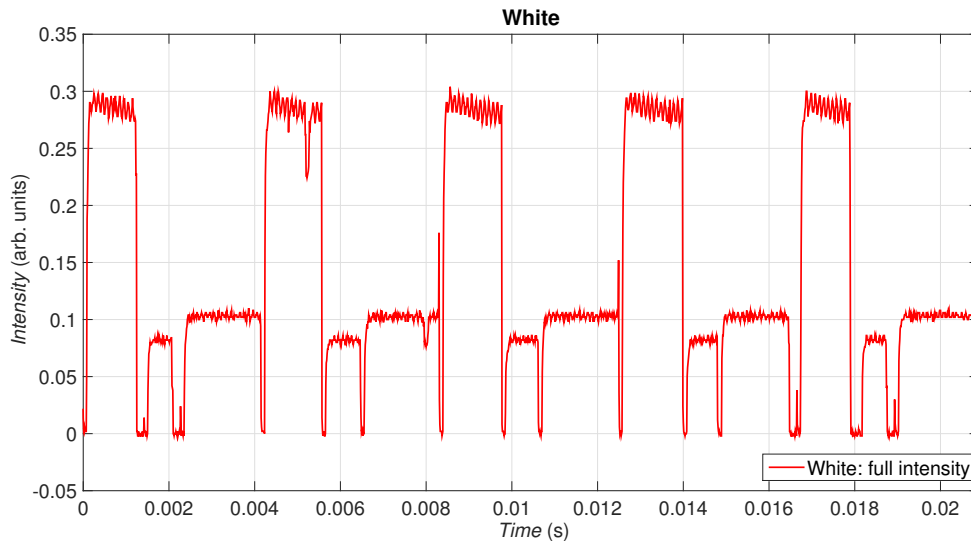


Figure 34: Measurement with full white. White colour is achieved by the coupling of the three primitive colours.

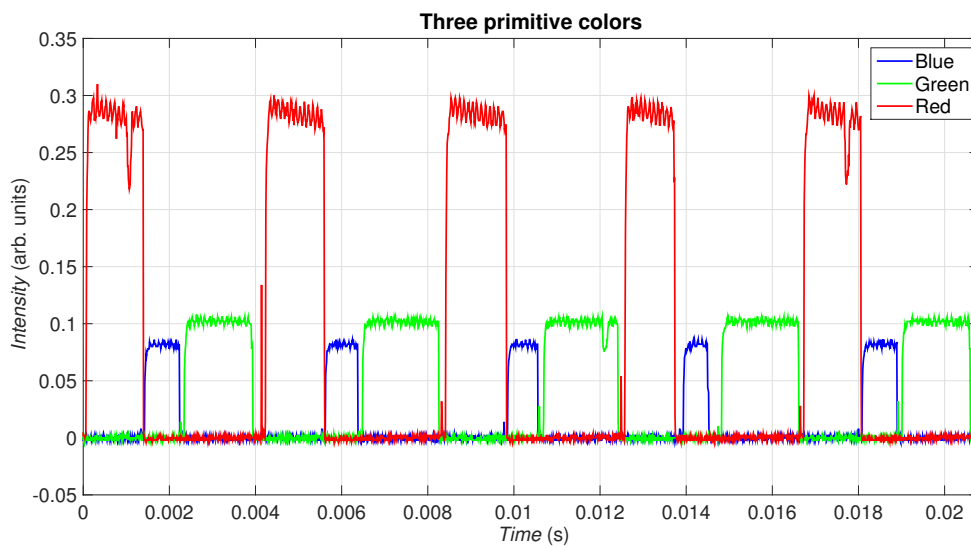


Figure 35: Independent measurements with the three primitive colours in full intensity plotted together.

In Figures 34 and 35 it can be proved the DMD really creates white light adding up the three primitive colours. Moreover, it can be seen that each LED has its own time, independent if the other LEDs are going to be on later or not.

14.1 Data Analysis

In this section, I analyse the plots shown in the previous section and discuss about the duty cycles of each LED, i.e. the time each LED is on.

Table 4: Different periods for the duty cycles from the different LED's have been measured.

| | Red on time (ms) | Blue on time (ms) | Green on time(ms) |
|----------------|-------------------------|--------------------------|--------------------------|
| | 1.42 | 0.80 | 1.88 |
| | 1.48 | 0.90 | 1.82 |
| | 1.45 | 0.86 | 1.89 |
| | 1.51 | 0.75 | 1.90 |
| | 1.39 | 0.90 | 1.87 |
| Average | 1.45 ± 0.04 | 0.84 ± 0.08 | 1.87 ± 0.03 |

Table 5: Different values of complete duty cycles starting from different points. The average of these values has been taken in order to have a reliable value.

| | Period (ms) |
|----------------|--------------------|
| | 4.10 |
| | 4.20 |
| | 4.16 |
| | 4.16 |
| | 4.17 |
| | 4.26 |
| | 4.04 |
| | 4.26 |
| | 4.13 |
| | 4.15 |
| | 4.19 |
| Average | 4.16 ± 0.06 |

Table 6: The fraction of each duty cycle (d.c.) comparing them with a complete cycle has been calculated.

| | Red d.c. (ms) | Blue d.c. (ms) | Green d.c. (ms) | Full cycle period (ms) |
|-------------------------|----------------------|-----------------------|------------------------|-------------------------------|
| Period of d.c. | 1.45 ± 0.04 | 0.84 ± 0.08 | 1.87 ± 0.03 | 4.16 ± 0.06 |
| Fraction of d.c. | 0.34 ± 0.03 | 0.20 ± 0.02 | 0.45 ± 0.02 | 1 |

Another phenomenon that I have seen in the plots of the previous section is that the pattern of the intensities repeat every 4 periods. Both patterns are being repeated, the patterns of the full intensities and the patterns of the half intensities.

15 Switching of Mirrors

It is interesting to know how much time the mirrors of the DMD device need to switch. In order to measure that, I have used the *Pattern Sequence* Operating Mode from The LightCrafter 4500 Control Software. We select the *Sequence Settings* option. In the *Individual Pattern Settings* we can choose the colour of the projection, we select white and we select *Internal Trigger*. The settings of *Bit Plane Selection* are not very important for this measurement. In this case I have chosen *Flash index* = 0, *Bit depth* = 1, then I click G0 and *Add Pattern to Sequence*, the chosen pattern will appear in the *Pattern Sequence* box. Between *Play Once* and *Repeat* I select *Repeat*. On the right side of the screen, I select *Flash* as Pattern Source and *Internal/External* as Trigger Mode. The timing controls how long each pattern is projected and also the time for triggering. The triggering resets the mirrors in every period and it needs at least $230 \mu\text{s}$ to do that. The *Internal Trigger* time is the time for a full period and the *Pattern Exposure* time, the time that the pattern is actually projected in each period. So the remaining time in each period is used for triggering. The values I have selected are *Internal Trigger* = $2000000 \mu\text{s}$ and *Pattern Exposure* = $1000000 \mu\text{s}$. That means that the pattern will be switching on and off every second. To upload the setting to the DMD device, I click *Send*, then *Validate Sequence* and finally *Play*.

Now that the pattern is being projected, I measure the power of the projection using a pho-

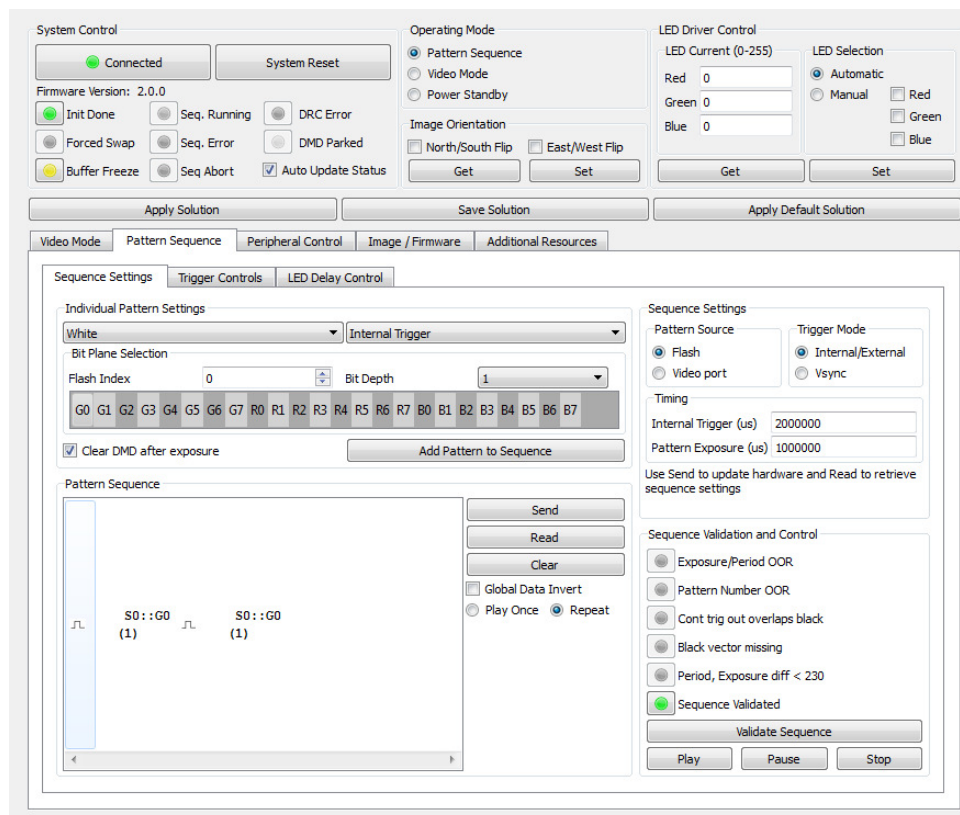


Figure 36: This is how the user interface look like when giving the explained external pattern to the DMD.

todiod and take the data with an oscilloscope. In my projection, the mirrors are switching on

and off every second, but what I have measured is the transition time they need to flip from on to off and vice versa (Figures 36 and 37).

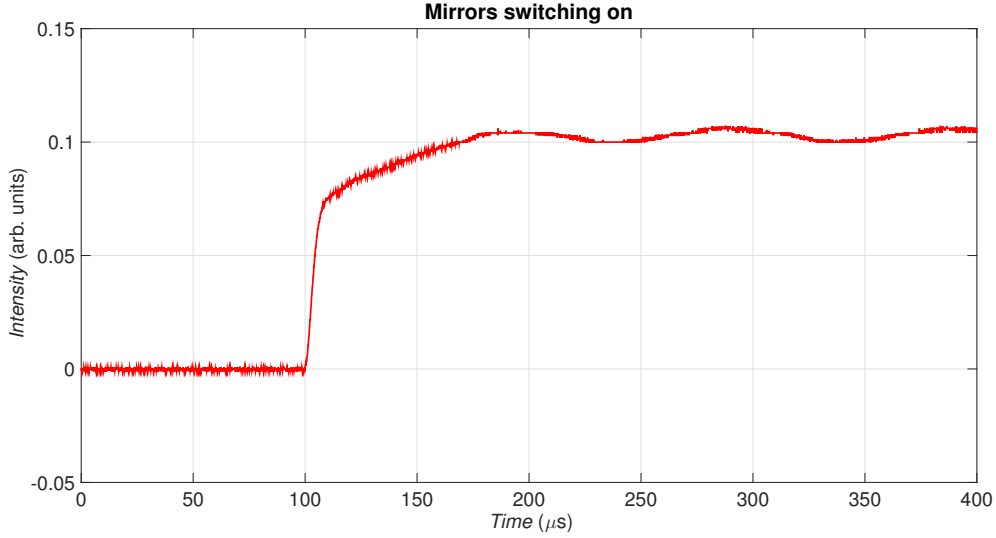


Figure 37: Measurement of the time the mirrors need to flip for switching the light on. I have set a threshold for the time needed to switch the light on: when the intensity is the 90% of the maximum value of the intensity. According to this criterion, the mirrors need $53.1 \mu\text{s}$ to switch on, and after they are switched on they bounce.

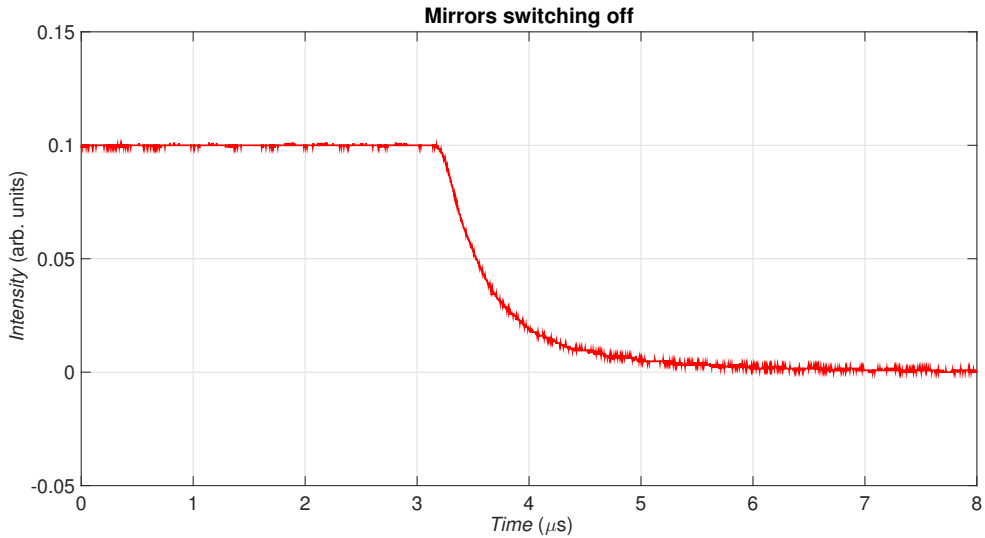


Figure 38: Measurement of the time the mirrors need to flip for switching the light off. Following the same criterion, I have set threshold for the time needed for switching the light off: when the intensity is the 10% of the maximum value of the intensity. The mirrors need around $1.36 \mu\text{s}$ to switch off. The reason that in this case the bounces can not be seen on the measurement is because already before reaching the “off-state” there is no projected light.

For the data taken from Figures 36 and 37, we can calculate the resulting maximum frequency for making on and off shining pulses in the projector:

$$\nu_{max} = (T_{on} + T_{off})^{-1} = (53.1 \mu\text{s} + 1.36 \mu\text{s})^{-1} = 18.36 \text{ KHz} \quad (11)$$

Conclusions

I have built and characterised a laser and I have learned what each of the components does. I have been able to run the laser in single-mode and stabilised it around a desired setpoint thanks to a PID controller that I have programmed. I have established a communication between the PID controller programmed in LabVIEW and Arduino Due, the DAC that I have chosen after comparing it with another candidate.

I have learned some basics of how the LightCrafter 4500 DMD works. The projected light is the composition of the lights of three LED's, each of which has a certain on-time. The mirrors chose to be in on- or off-stages depending to the amount of intensity that we want for each colour.

A Appendix: Block Diagram of the LabVIEW program for the PID Controller

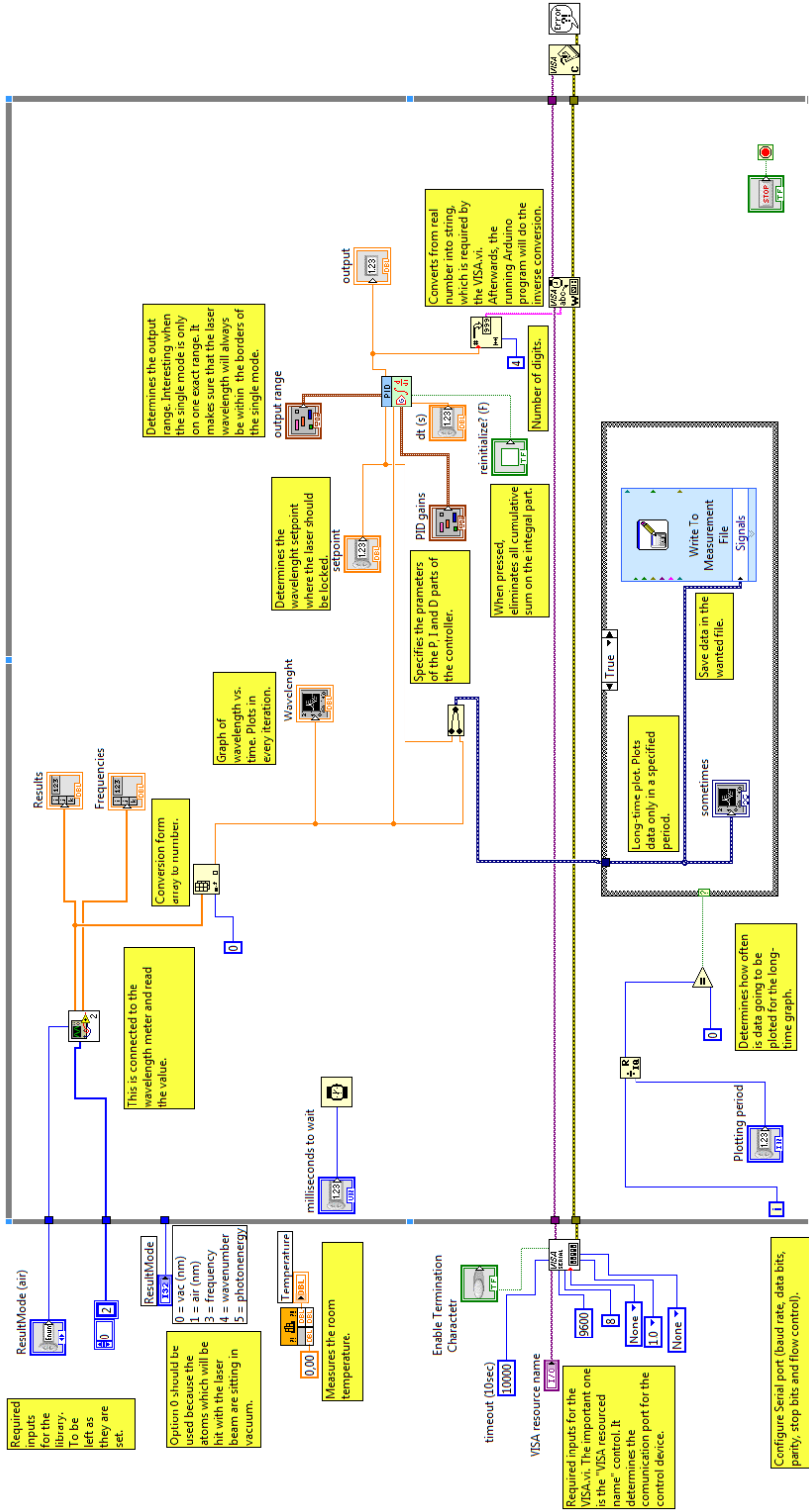


Figure 39: Block Diagram of the LabVIEW program for the PID Controller. Explanations available in the yellow boxes.

B Appendix: Front Panel of the LabVIEW program for the PID Controller

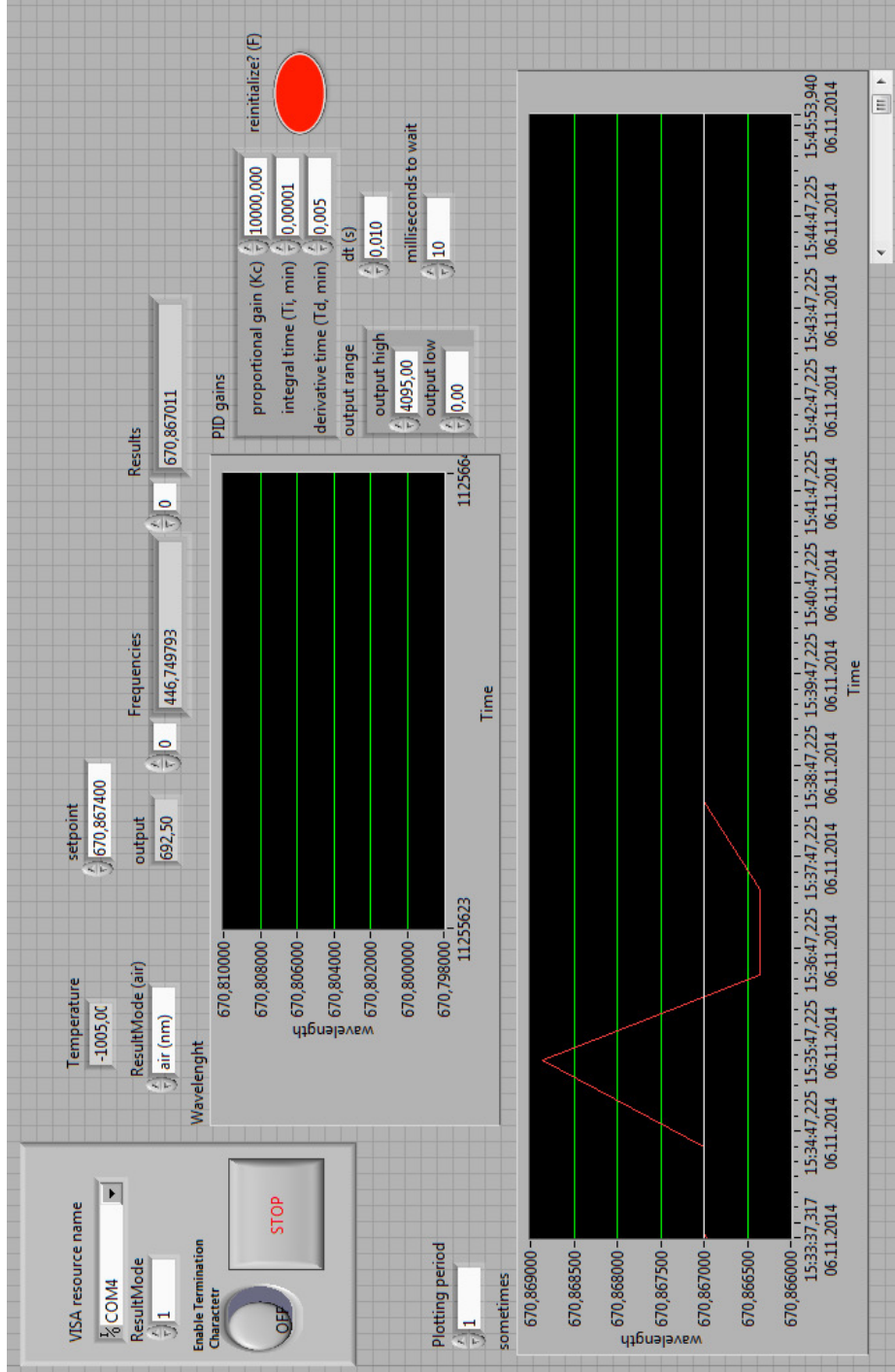


Figure 40: Front Panel of the LabVIEW program for the PID Controller.

C Appendix: Arduino Program for PID Controller

```
// THIS PROGRAM STABLISHES THE CONNECTION BETWEEN LABVIEW AND ARDUINO
//declarations:
int ii=0;
int serialData = 0;
int value = 0;
int bytes=4;           // Determines the number of digits to be read.
int ee=1000;
char buf[4];
void setup() {

  Serial.begin(9600);   // Inisialises the connection with LabView in the given speed.
  analogWriteResolution(12); // Resolution of the analog output 12 bits = 2E12 = 4096
}

void loop() {

  if (Serial.available() > 0 ){ //checks whether the VISA.vi of LabView program is giving an output
    Serial.readBytes(buf,4);    // reads and saves the output of the VISA in an array of 4 elements
                                // The VISA.vi give the output as a string, so "buf" will have the
                                // equivalent of the characters in ASCII code.

    for(ii=0;ii<4;ii++) {      // Converts each element of the array from ASCII code into a number.
                                // The ASCII analog of numbers from 0 to 9 go from 48 to 57

      if (int(buf[ii])<47){    // If the first number is "0", the VISA will read it as a space.
        buf[ii]='0';          // Here we convert the space into "0"
      }
      value=value+(int(buf[ii]-48))*ee;
      ee=ee/10;
    }
    analogWrite(DAC1,value);   // Gives the output to an analog output pin of the Arduino.

    value=0;                   // Erases memory before starting with the next loop.
    ee=1000;
  }
}
```

Figure 41: Arduino Program for PID Controler. Explanation available as comments.

References

- [1] Thorlabs www.thorlabs.de/thorcat/21700/HL6756MG-SpecSheet.pdf
- [2] <http://imagebank.osa.org/getImage.xqy?img=M3cubGFyZ2UsYW8tNDgtMzUtNjY5Mi1nMDAz>
- [3] <http://hyperphysics.phy-astr.gsu.edu/hbase/sound/beat.html>
- [4] Thorlabs www.thorlabs.de/thorcat/15600/AE0203D04F-SpecSheet.pdf
- [5] Thorlabs http://www.thorlabs.de/newgrouppage9.cfm?objectgroup_id=1295pn=DET36A/M
- [6] Master Thesis Fnu Traptilisa (2014): *Characterization and Development of an Extended Cavity Tunable Laser Diode*; San Jose State University
- [7] Bachelor Thesis Oliver Hermann (2014): *Erzeugung von Lichtfeldverteilungen mit Hilfe eines Digital Mirror Device (DMD)*; Ulm University
- [8] Bachelor Thesis Fabian Etzold (2010): *Laser Beam Shaping with a Digital Mirror Device*; Johannes Gutenberg University, Mainz
- [9] HighFinesse http://www.highfinesse.com/Brochure/LowRes/HighFinesse_Product_Brochure.pdf
- [10] Texas Instruments <http://www.ti.com/lscs/ti/dlp/advanced-light-control/microarray-greater-than-1million-lightcrafter4500.page?DCMP=dlp-lightcrafter4500-enHQS=dlplightcrafter4500>
- [11] Texas Instruments <http://www.ti.com/lit/an/dlpa008a/dlpa008a.pdf>
- [12] Texas Instrument: DLP®LightCrafter™DM365 Command Interface Guide (2012).
- [13] W. Brunner, K. Junge (1989): *Lasertechnik - Eine Einfuehrung*; Huethig Verlag Heidelberg; 4th Edition.
- [14] J.Eichler, H.J. Eichler (2006) *Laser - Bauformen, Strahlfuehrun*; Huethig Verlag Heidelberg; 6th Edition.

Acknowledgement

At this point, I would like to thank all the people who are part of the success of this work, specially Professor Dr. Johannes Hecker Denschlag for welcoming me in his team and for the facilities of the laboratory. Many thanks to M.Sc. Thomas Paintner for guiding me throughout my Bachelor Thesis, and helping me find solutions to the problems I had to face during this time in the lab. I would also like to thank M.Sc. Daniel Hoffman and M.Sc. Michael Griener for the given help and the rest of the LiLaLab-team for the nice atmosphere and welcome in the lab.