Active Stabilization of an Injection Lock

Ong Jing Hao
A0140236U
Department of Physics, Faculty of Science
National University of Singapore

Supervisor:

Travis Nicholson
Centre for Quantum Technologies
National University of Singapore

In partial fulfillment of the requirements for the degree of

Bachelor of Science (Hons)

2019

Acknowledgements

I would like to firstly thank my supervisor, Professor Travis Nicholson, for his limitless patience and invaluable help; without which my project would not have been possible. His unparalled problem solving skills saved me many a times when I was stuck. I would like to thank my fellow researchers, PhD students Wang Rui, Mo Jinchao and Yu Xianquan for always being ready to drop their work to help with any issues I encounter in this project. Lastly, I would like to thank the Department of Physics, NUS and the Centre of Quantum Technologies for the opportunity to delve into such a project, as well as the resources they have provided.

Special mention to Tan Ting You, a fellow undergraduate researcher, who has made valuable contributions to the creation of various graphs in this thesis, as well as being a good source of "intellectual" discourse.

Thank you mum, for telling me to always come home early, although I rarely did. Thank you for showering me with your love for so many years; I am finally graduating, and now it will be my turn to be the one taking care of you (for once).

Dads are rarely the ones to show too much emotion; my dad is a model example. But from his actions I can still feel the warmth of his love. Thank you dad, for being that silent yet unyielding pillar of support that I can always lean on when I am tired.

As for *that* bunch of friends, thanks for making the study of Physics less stressful. Although distracting, their antics never fail to take the edge off of those plentiful (painful) assignments. For those who guided me when I was confused, thanks for pulling up my grades.

Abstract

This paper aims to describe our development of a system for the active stabilization of an injection lock. For our group, we aim to cool thallium down into a Bose-Einstein condensate and to do so a laser operating in the ultraviolet region is required to pump thallium atoms into the cooling cycle. As it is rare to obtain a UV diode laser that produces a beam with both high power and narrow linewidth, we require an injection lock to produce these beams. Unfortunately, these locks drift over time and so require hourly manual tuning. Laser injection locks are generally maintained passively, but an active stabilization system would be more efficient in terms of labour and time. In this paper, we shall give an overview of how our system works, and our reasons for the various parameters and methods that we have set for the system.

Contents

1 Motivation **1**

2 Ultracold Atoms **3**

2.1 Bose-Einstein Condensates 3

2.1.1 What are they? 3

2.1.2 2001 Nobel Prize 3

2.1.3 Recent BEC News 4

2.2 Group III ultracold atoms 5

3 Injection Locking **7**

3.1 How injection locking works 7

3.2 Injection Lock Setup 8

4 Stabilization Procedure **10**

4.1 Basic Procedure 10

4.2 Experimental Setup 11

5 Firmware Overview **14**

5.1 Peak finding 15

5.2 Control loop 17

5.2.1 System Response Curve 17

5.2.2 Peak Lock Routine 20
5.2.3 Arduino Setup 22
5.3 Writing to the DAC 23
5.3.1 Measuring the background voltage 24
5.3.2 PID loop 25
5.4 Lock Acquisition Switch 27

A Appendix 1 - Cooling systems 28
A.1 Laser cooling 28
A.1.1 Doppler Cooling 28
A.1.2 Magneto-Optical Traps 30
A.2 Evaporative Cooling 31

B Appendix 2 33
B.1 Circuit Schematic 33
B.2 Firmware Code 35

References 44

Motivation

For many years, there has always been only 3 states of matter: solid, liquid and gaseous states. In 1879 plasma was discovered by Sir William Crookes, and people have gradually included that to become the 4th state of matter. Recently, there been growing interest in what some has come to call the 5th state of matter: Bose-Einstein condensates.

Bose-Einstein condensates, or BECs, were first postulated in 1924-1925 by Satyendra Nath Bose, who saw that this was one method of distribution for an ensemble of identical, indistinguishable particles. Albert Einstein later adopted and extended Bose' ideas, fully developing the concept of the Bose-Einstein Condensates.

To achieve a BEC state, we need to cool particles down to extremely low temperatures, to the tune of 100 nanokelvins. The first step in reaching these temperatures will be to perform laser cooling, which can efficiently cool atomic beam sources from temperatures on the order of a few hundred degrees Celsius to the few-μK level.

Laser cooling experiments often use diode lasers, which are relatively simple and low cost systems. Historically, most of these experiments required near-infrared diode lasers, which are now a reliable and mature technology. Our experiment requires ultraviolet lasers, and UV laser diodes are much newer and less reliable devices compared to their NIR counterparts. For UV diodes, the only available options are those with narrow linewidths but only a few mW of output power, and those with tens of mW of power but broad linewidths. Unfortunately, our experiment requires a laser with both a narrow linewidth *and* tens of mW of output power. One method of achieving a laser source with both of these properties is known as injection locking. In such a lock, we send the narrow linewidth beam from the master laser into the high power slave diode to produce an output beam that possesses both the original high power of the slave laser, and also the narrow linewidth of the master laser.

One major issue with such an injection lock is that the lock drifts over time, such that an hour is enough for the laser to fall out of lock. To keep the injection lock in an optimal state, there is a need to adjust and tune it hourly. The current standard is to adjust it manually, but we believe a more efficient method of stabilizing this lock is through an active stabilization system.

In this paper we shall describe our progress and insights obtained in the development of an active stabilization system for injection locking, such as the problems we faced and the reasoning behind our solutions to these problems.

Ultracold Atoms

2.1 Bose-Einstein Condensates

2.1.1 What are they?

First proposed by Satyendra Nath Bose in 1925, loosely speaking Bose-Einstein condensates are basically a collection of bosonic particles cooled down to such low temperatures that they lose their individual identity and settle down (often) into the lowest energy level, thus having such a wavelike nature that the whole system can be described by a single macroscopic wavefunction. Since all the particles are identical, we can also perceive the particles in the BEC as coherent matter waves, similar to the coherent light of lasers. If enough energy is removed from the gas of particles, they will begin to seek identical states due to their bosonic nature. Thus, a BEC is formed.

2.1.2 2001 Nobel Prize

Despite its early proposal by Bose, it took 7 decades until Eric A. Cornell and Carl E. Wieman managed to successfully produce a BEC in 1995, for which they won the Nobel prize in 2001 (sharing it with Wolfgang Ketterle, who also managed to produce a BEC in the same year).[12] The thermal de Broglie wavelength of a particle is proportional to the square root of its temperature, and so if a sufficiently dense gas of cold atoms can be produced, the matter wavelengths of the particles will be of comparable magnitude as the distance between them. It is at this point that their bosonic nature dominates and a BEC is formed[12]. However, gases usually condense into liquids when cooled, and this must be avoided when trying to produce BECs. All 3 Nobel laureates managed to achieve this by using akali atoms, where a BEC can be formed if the density of atoms (expressed as the number of atoms inside a λ-sided cube) exceeds 2.6, translating to a temperature of around 100 nanokelvin.

2.1.3 Recent BEC News

In 2017, a group of researchers managed to create a BEC in space and conduct a total of 110 experiments on it[2]. One main goal of this experiment was to "provide insights into conducting cold-atom experiments in space, such as precision interferometry", as these precision interferometry experiments can help test the universality of free-fall[1][21]. As the sensitivity of measuring inertial forces with matter-wave interferometers is proportional to the square of the time that the atoms spend in the interferometer[3], the accuracy of Earth-bound experiments is limited by the duration of free-fall, which can be greatly increased in any experiment performed in space due to micro-gravity conditions present there. This one example suitably illustrates the importance of BECs, not just in atomic physics but other fields of research too.

[1] Albert Einstein's insight that it is impossible to distinguish a local experiment in a freely falling elevator from one in free space, which subsequently led to the development of the theory of general relativity.

2.2 Group III ultracold atoms

As mentioned earlier, the first BEC was formed by cooling Group I atoms, which are generally elements belonging to the first column of the periodic table. They are convenient for cooling due to their inherent energy levels. In the ultracold regime, Group I atoms has an interesting property known as magnetic Feshbach resonances. This is the ability to precisely tune the scattering length using a magnetic field[5]. This ability of magnetic Feshbach resonances means that physicists are able to investigate exciting and important phenomena like the creation of bright matter solitons, the investigation of the crossover from Bardeen-Cooper-Schrieffer pairing to a Bose-Einstein condensate for strongly interacting Fermi gases, as well as the coherent formation of ultracold molecules.

In recent years, Group II atoms have gained much popularity due to their optical clock transitions. These have enabled the world's best atomic clocks[4], exotic quantum many-body problems[9], new quantum coupling prospects[8] and more.

Now, we get to the Group III atoms. This group of atoms have not been cooled to the ultracold regime yet, and our group aims to be the pioneer in the exploration of the field of ultracold Group III atoms. When we manage to cool Group III atoms down to the ultracold regime, it could enable novel atomic clocks, new systems for quantum simulation of many body physics, and studies of dipolar phenomena. Group III atoms have so much potential because they possess both magnetic Feshbach resonances and optical clock transitions, which do not both exist in Group I and II atoms.

For our group, we chose the element Thallium to work with. Laser cooling of Thalium requires a 378 nm laser to prepare it in its 6p $^2P_{3/2}$ F=2 cooling state.

The next chapter will reveal why we need to have an injection locked laser for this 378 nm transition.

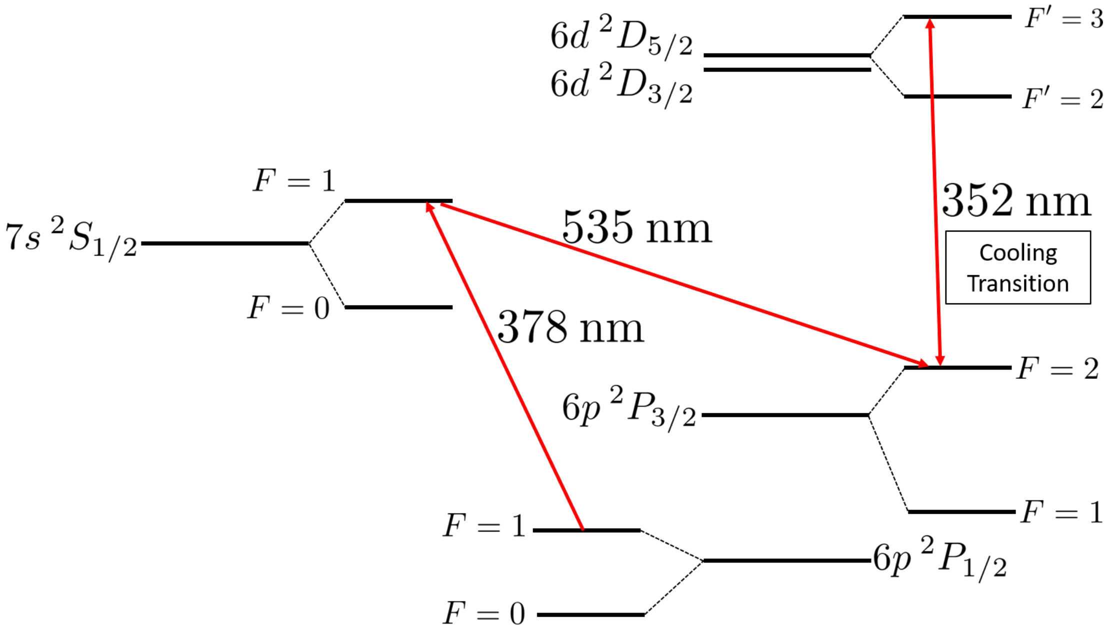


Figure 2.1: Energy level diagram of Thallium. We need to pump it to the 7s $^2S_{1/2}$ F=1 state where it will decay down to the 6p $^2P_{3/2}$ F=2 state, which is the ground state of the cooling cycle (where the excited state is the 6d $^2D_{5/2}$ F'=3 state, separated by a 352 nm transition).

Injection Locking

Fortunately for us, 378 nm diode lasers have become commercially available in recent years. Unfortunately, these diode lasers come in 2 types: either multimode diodes having high power or low power single frequency diodes; there are no suitable diode lasers that has both high power and narrow linewidth properties. Thus, we have to injection lock a high-power multimode diode with a low-power single frequency diode, to produce an output beam that has high power and a single frequency.

3.1 How injection locking works

Although used commonly in the field of optics, the discovery of such a phenomenon surprisingly orginated in a completely different field. It started from the discovery made by Christiaan Huygens in 1673, when he noted *"When we suspended two clocks so constructed from two hooks embedded in the same wooden beam, the motions of each pendulum on opposite swings were so much in agreement that they never receded the least bit from each other and the sound of each was always heard simultaneously. Further, if this agreement was disturbed by some interference, it reestablished itself in a short time. For a long time I was amazed at this unexpected result, but after a careful examination finally found that* *the cause of this is due to the motion of the beam, even though this is hardly perceptible. The cause is that the oscillations of the pendula, in proportion to their weight, communicate some motion to the clocks. This motion, impressed onto the beam, necessarily has the effect of making the pendula come to a state of exactly contrary swings if it happened that they moved otherwise at first, and from this finally the motion of the beam completely ceases."* In short, Huygens attributed this phenomenon of synchronised oscillations to the tiny vibrations travelling in the wooden beam, effectively coupling the 2 pendulums. Besides anti-phase, pendula can also oscillate in phase based on the same principle. Later, these types of synchronised oscillations were termed sympathetic oscillations.

In a similar fashion, lasers can be coupled to each other, and this method is known as injection locking. Carl E. Wieman's paper on the usage of diode lasers for atomic physics has also highlighted injection locking as a method to obtain a tuneable narrow linewidth laser.[20]

3.2 Injection Lock Setup

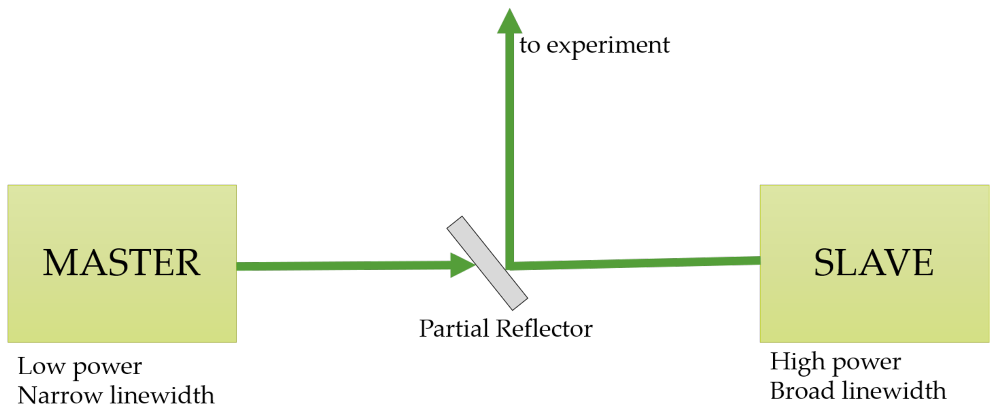


Figure 3.1: A simple schematic of an injection lock

The above is a simple schematic of an injection lock. We have 2 lasers here, the master and the slave laser. The master is the low power single frequency diode, while the slave is the high power multimode diode. We inject the narrow linewidth beam from the master into the slave laser, and based upon the principles of sympathetic oscillations the output beam of the slave laser will mimic the narrow linewidth of the master laser, effectively outputting a beam with high power and a single frequency, suiting our purposes for laser cooling. In order to exhibit this kind of resonance effect, the frequencies of the master laser and the slave laser must be sufficiently close; basically the closer the frequencies the better the injection lock.

There is, however, a problem with injection locks. The injection lock is optimal at a particular value of slave diode current, and this optimum drifts due to changes in temperature and beam alignment. Over the course of an hour, the optimum can drift so much that the laser loses lock. We would like the slave diode laser current to track the drifting optimum, thus maintaining a high quality injection lock at all times. The only published method that we found for achieving this is *Active Stabilization of a Diode Laser Injection Lock* by Gupta, et al. [18]. Our goal is to build an active stabilization system that improves upon the Gupta design.

Stabilization Procedure

4.1 Basic Procedure

We are able to determine if the laser is properly locked by measuring its power spectrum, which differs dramatically based on the status of the injection lock. We quantify the injection lock quality using a Fabry-Perot interferometer as described in the section below. The idea for our system is this: When the laser is not injection locked, the beam it emits is so spectrally broad that there will not be any fringes in the transmission signal of the Fabry-Perot cavity. When the laser is injection locked, the cavity will output fringes, and the fringe heights depends on the injection lock quality (Figure 4.2). Thus, our control loop will be measuring the fringe heights and providing the appropriate feedback to the slave diode current controller. Here, we are simply tuning the slave diode current so that we can match the frequency of the slave laser to that of the master laser so as to achieve optimal injection lock.

4.2 Experimental Setup

Figure 4.1 shows the schematic of our experimental setup. It comprises of the injection lock with its master and slave lasers, the beam splitter, the Fabry-Perot cavity and the circuit board. We use the beam splitter to divert a portion of the output beam into the Fabry-Perot cavity, where the cavity length is being varied by the triangular voltage signal we sweep across it. If there is injection lock and the cavity length is on resonance with the wavelength of the input laser beam, we see a sharp narrow spike in intensity, producing tall and narrow fringes (see Figure 4.2). Sub-optimal locking will lower the peaks of the cavity fringes. We will not see any fringes, however, if the system is not properly injection locked, as the input beam will be multimode, thus producing a broad, dispersed signal. The FP transmission signal goes into the circuit (Figure 4.3) where the Arduino Uno receives/reads the input signal, measures the peak height of the cavity fringes, generates a PID[1] correction to the Digital-to-Analog Converter (DAC) value that adjusts the voltage of the slave diode current controller, producing a corresponding change in the slave laser diode current.

To end off this chapter, we shall mention that although our scheme borrows the idea of using cavity fringes to discern the quality of an injection lock from Gupta et al, we have opted to design and build our own locking system and active stabilization algorithms, independent of any published work currently available.

[1]The Proportional-Integral-Derivative feedback loop is a common algorithm for generating a correction to minimise the described error signal.

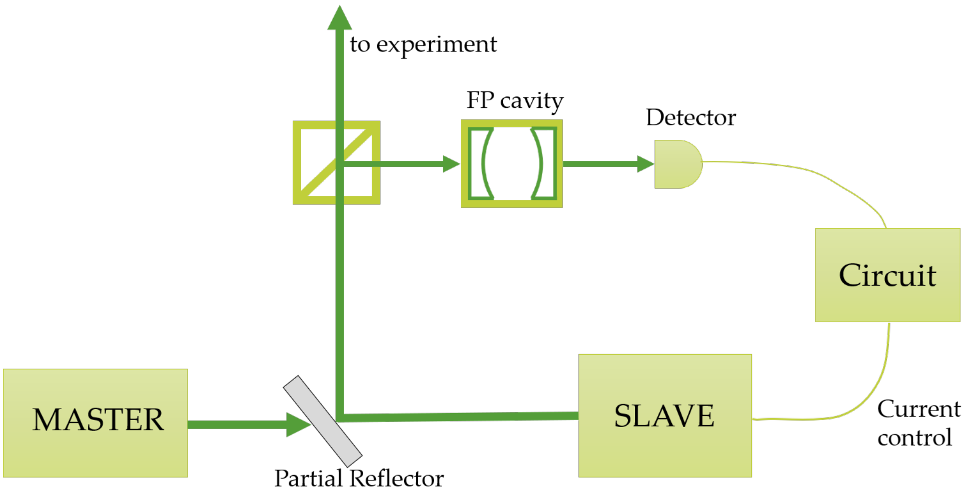


Figure 4.1: Setup of the feedback loop of our system.

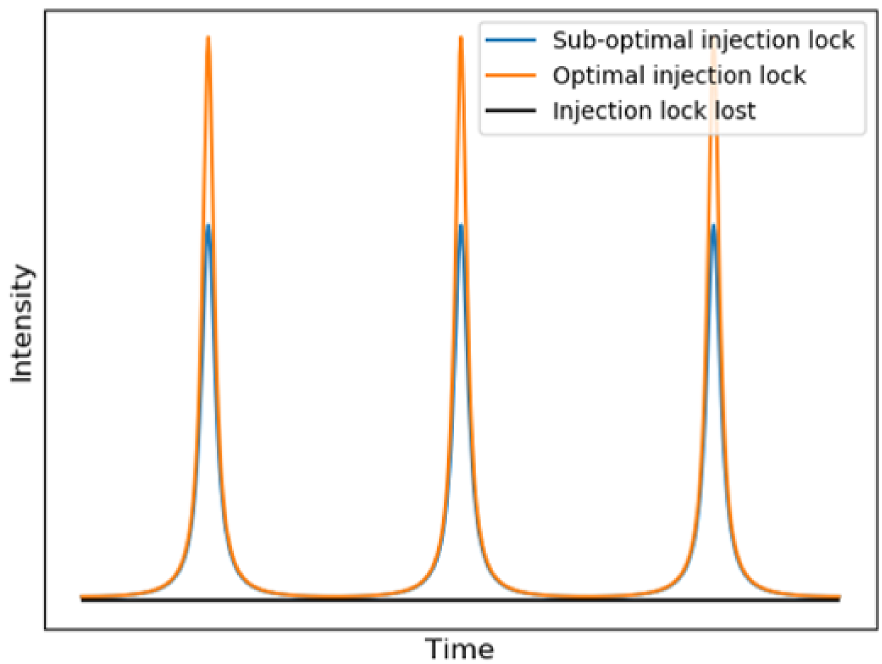


Figure 4.2: A diagram showing the FP cavity fringes for various injection lock qualities. As the cavity length is swept with a triangular voltage signal, the output signal is plotted with respect to time.

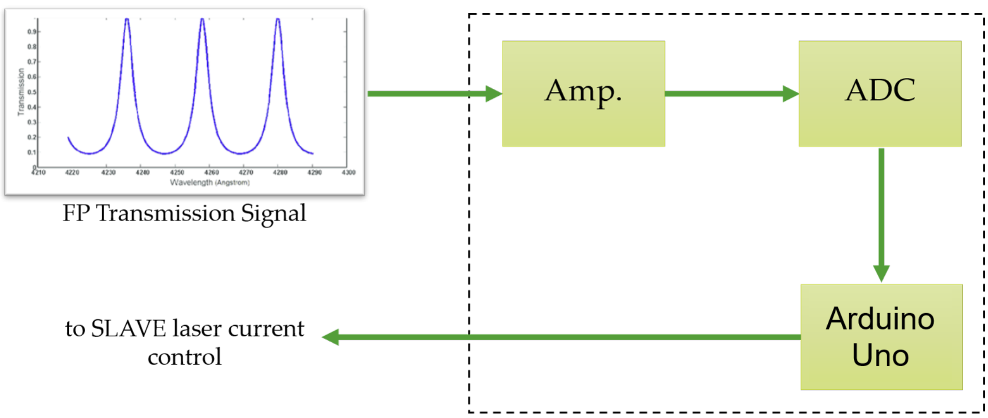


Figure 4.3: A block diagram of the components of our circuit board.

Firmware Overview

In this chapter we discuss the important aspects of our firmware and explore the reasoning behind the choice of some of our parameters. From understanding the circuit board's components to soldering them on, milling out a front panel for it and finally setting up the laser system and installing the board, before moving on to the alignment of lasers to get optimal data, one had to pick up quite a few things before even being able to start on the firmware. Once again I need to thank my various lab mates, and Professor Nicholson for being there to guide and help me. Although my project did involve building the electronics for this system, firmware development was the most labor intensive and complicated part of this project, and therefore I will focus mainly on the firmware from here on.

This chapter will be discussing the following aspects of our firmware: the two main features which are the peak finding and control loop algorithms, and other smaller (but no less crucial) functions like the lock acquisition code, how we measured the background values, and write to our DAC so we can output the correction current to the slave diode. To illustrate our point, we shall provide relevant snippets of our code, but to get the big picture we have attached our whole firmware code in the Appendix. All firmware codes are compiled using the Arduino IDE.

We shall begin by discussing the main features first.

5.1 Peak finding

Since we are determining the injection lock quality from the peak height of the cavity fringes in the FP transmission signal, we have a responsibility to provide a method of accurately measuring the peak height. *Numerical Receipes* by William H. Press et al suggested methods like Brent's method, parabolic interpolation and the cubic spline for interpolating the peak data, but we have come to the conclusion that it is indeed unlikely to be able to fully characterise the peak of any of our FP cavity signals obtained, especially since we would not want to compromise on the peak heights measured[1], as variations of these heights are what we shall be basing our feedback system on. Finally, our decision was to find the maximum peak value detected over 15 cavity peaks. The peak finding algorithm reads out the analog output of the cavity over one cavity fringe peak and chooses the max value detected. We want to sample enough cavity fringes to have a reasonable resolution of the entire fringe (especially the peak region), otherwise the peak finding algorithm's less-than-ideal analog read resolution means that we could have erroneously conclude that the peak height has decreased, when in actual fact we failed to accurately sample the peak due to the coarse sampling. However, we also need to keep in mind that increasing the sample size correspondingly increases the run time of our algorithm. After repeated trials, the number 15 was chosen here as a balance between timing considerations and worries about under-sampling.

To this end, we wrote the `find_peak(x)` function, taking the background reading as an argument[2]. It then begins to record peak values only when it detects a

[1]The above-mentioned methods still work well in providing a smooth approximation of the behaviour of the points at the peak, but come at a price of lowering the peak height obtained.

[2]How we find the background reading is covered in the later parts of this section.

peak height above 1.5 times the background reading. This is to make sure we do not expend memory saving redundant background readings.[1].

The relevant code is as such:

```
int find_peak(int x){ // input argument is the background reading
    while (peak_counter<15){
      if(analogRead(A0) > 1.5*x){
        for(int j=0; j<50; j++){
          peak_values[j] = analogRead(A0);
        }
        int peak0 = peak_values[0];
        for(int j=1; j<50; j++){
          if (peak_values[j] > peak0){
            peak0 = peak_values[j];
          }
        }
```

After recording 15 peak values, we then proceed to find the maximum out of these peaks. This will be the peak value that corresponds to the current DAC value. The `timeout_counter` is just a fail-safe in case something goes wrong and the peak values never go above the threshold we set.

```
        if(peak0 > peak){
          peak = peak0;
        }
        peak_counter++;
      }
```

[1]An Arduino chip has limited memory, so we avoid saving irrelevant readings of the background and any possible noise fluctuation of the background.

```
        timeout_counter++;
        if(timeout_counter >= 1000){
          break;
        }
      }
      return peak;
}
```

Finally it returns us the peak value it has detected.

5.2 Control loop

The next main feature of our firmware we shall be discussing is the control loop. This is arguably the most important part of our firmware, as the control loop is how we generate the error signal from the data we obtained.

5.2.1 System Response Curve

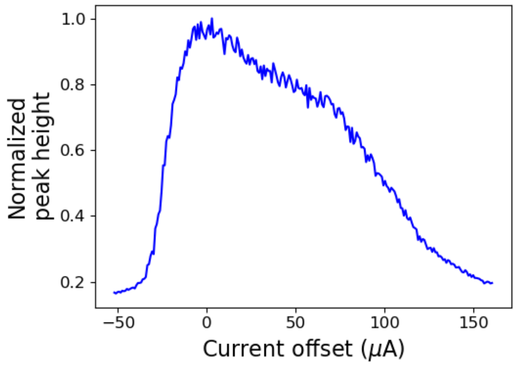
Firstly, we sweep over the output range, measuring the peak at each point. The DAC value is an integer between 0 to 65535[1] which we write to the DAC, eliciting a corresponding output voltage from the DAC. Since the current sent by the current controller to the slave diode is proportional to this DAC voltage set across the current controller, it would be fair to use the DAC value as a characteristic parameter of the output current sent to the slave laser diode. This generates a system response curve[2] showing us the behaviour of the injection lock quality with changing output current.

[1]The DAC we are using here is a 16-bit DAC.

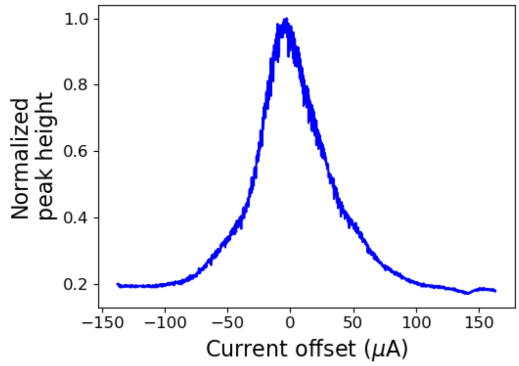
[2]Or a spectral purity curve if the reader has been referring to the Gupta paper.

```
for(unsigned int j = 28000; j <= 43000; j+=300){
  DACwrite(j);
  delay(10);
  find_max[max_counter] = find_peak(bg);
  max_counter++;
}
```

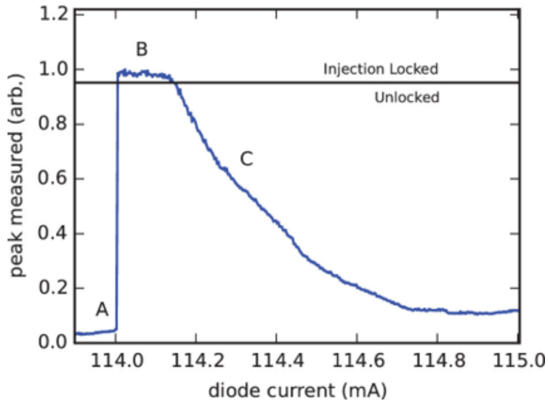
The system response curve is shown below. At first, we obtained the asymmetric curve on the left (Fig 5.1a). This is similar to that obtained by Gupta's group, who attributes the asymmetry to the combination of two thermal effects: the Joule heating of the laser and heating from seed light resonance. On the high current side, these 2 thermal effect act against each other and produce a gentler approach to the injection lock region; on the low current side they act with each other and produce a rapid change in peak height with current. Due to this asymmetry, the paper uses an active stabilization method that favours approaching the lock interval from the side with a gentler slope. After various adjustments we discovered that in our case the asymmetry of the curve can be removed by careful alignment of the laser system, and after doing so we obtained the symmetric curve on the right (Fig 5.1b). The asymmetry in the graph also appears to depend on seed power, such that an increase in seed power may cause an asymmetry in the curve obtained. We believe our symmetric curve is an improvement over Gupta's asymmetric curve for the following reasons: The asymmetry forces the Gupta group to fix a desired threshold peak value for injection lock. Unfortunately, the height of the system response curve can fluctuate due to changes in laser output power or cavity coupling. Thus their system can fail as the desired peak height is now physically unattainable. Our curve's symmetry allows us to use a peak locking algorithm that is insensitive to the fluctuations in the height of the system response curve.



(a) An asymmetric system response curve.



(b) A symmetric system response curve.



(c) The system response curve obtained by the Gupta group.

Figure 5.1: A comparison of the spectral purity curve obtained before tuning the system (a), and after tuning the system (b), characterising the behaviour of the injection lock quality to changes in diode current. For convenience, we normalised the peak height obtained such that the max height of the spectral purity curve is 1. Also for comparison, the system response curve obtained by the Gupta group is shown (c).

Looking at the peak region of Figure 5.1b, we see that there is quite a bit of noise present. This is likely because the injection lock is much more sensitive near its optimum, as can be seen from the increasing gradient of the spectral purity curve, and so will be much more susceptible to minute changes in the environment: mechanical vibrations, temperature fluctuations and even the inherent electrical noise in the circuit. One way to find the DAC value corresponding to the peak would be to search for the largest value among the data; however the noise in peak measurements could cause this largest recorded value to occur at a DAC value where the injection lock is not optimal. To minimise the impact of noise, we have introduced a way of smoothing the graph: a simple window filter (also known as a moving point average). We varied the number of average points (see Fig 5.2), finally choosing a moving point average of 5, as this generates a curve that is smoothest near the peak region without compromising on the peak height of our system response curve.

So the code for smoothing is pretty simple:

```
for(int j = 2; j<=48; j++){
  int avg=0;
  for(int i = -2; i<=2; i++){
    avg+=find_max[j+i];
  }
  smooth_max[j] = avg/5;
}
```

5.2.2 Peak Lock Routine

Now that we have a smoothed system response curve, we can now apply our peak lock routine. First, we find the maximum value of our system response curve and

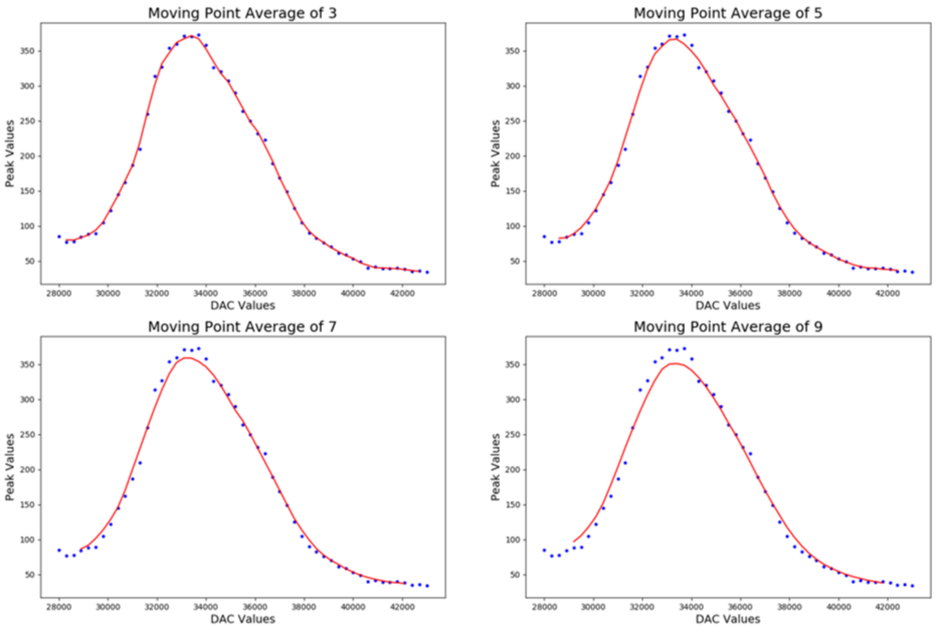


Figure 5.2: Application of window filters to data points. The data points are in blue and the smoothed curves are in red. We notice here the decreasing peak heights as we take the average of an increasing number of points.

find the corresponding DAC value (the `centre`). Then, we find the DAC values that correspond to 97% of the maximum value. We varied this threshold value and found 97% to be a good value for our needs of not varying too much from the peak value[1] as well as accounting for the noise present (even after smoothing). The difference between the peaks measured at these DAC values constitute our error signal.

```
for(int j = 0; j < 50; j++){
  float val1 = 0.97*max_value - find_max[j+1];
  float val2 = 0.97*max_value - find_max[j];
  if(val1 < 0 && val2 >= 0){
    left = centre - (28000 + 300*j);
  }
```

[1]To avoid introducing too much fluctuations into any experiment running at the same time.

```
    if(val1 >= 0 && val2 < 0){
      right = 28000 + 300*(j+1) - centre;
    }
  }
```

The reason we go to all this trouble of generating an error signal, instead of simply forcing the system to track the maximum value of the system response curve we found, is because the system response curve changes with time, as it is dependent on things like system alignment and seed power. For our active stabilization firmware to work regardless of the changes in system response curve, we used this idea of generating the error signal. Readers familiar with atomic clocks will recognise this method as similar to the atomic clock locking procedures. Compared to the system described in Gupta, et al., we are able to use this locking procedure because our system response curve is not highly asymmetric[1]. We then feed the error signal into the PID loop to compute the correction. This correction to the slave diode current should adjust the injection lock quality closer to the optimal value.

We have discussed the 2 main features of our firmware, now let us move on to the smaller (but no less important) functions that help make the features of our firmware work.

5.2.3 Arduino Setup

We quickly remind that we are using an Arduino Uno for this project, which communicates with a Digital-to-Analog chip to set a corresponding voltage across the slave diode current controller. For the Arduino, we set up the needed Serial

[1]Our firmware is still robust against some asymmetry, just not to the level observed in Gupta et al.

Peripheral Interface (SPI) setting as follows:

```
  void setup() {
    pinMode(2,INPUT);
    SPI.beginTransaction(SPISettings(14000000,MSBFIRST,SPI_MODE1));
    SPI.begin();
    Serial.begin(9600);
}
```

Parameters for the `SPISettings` were obtained from the DAC datasheet[1] and the baud rate for serial communication (so we can read out the data) is 9600.

5.3 Writing to the DAC

We need a command to write to the DAC the output voltage we wish to set. This voltage will draw from the slave diode current controller a corresponding slave laser diode current. We write it as a function so as to be able to layer our code. If we wish to output a specific current to the slave diode, we simply need to call it with an appropriate argument.

```
void DACwrite(unsigned int n){
  unsigned int n1 = floor(n/4096);
  unsigned int n2 = n - 4096*n1;
  PORTB &= B11111011;
  SPI.transfer(48+n1);
  SPI.transfer16(16*n2);
  PORTB |= B00000100;
```

[1] https://www.analog.com/media/en/technical-documentation/data-sheets/AD5683R_5682R_5681R_5683.pdf

```
}
```

Our DAC has a 16-bit resolution, so the output voltage resolution is (5/65535)V or 0.763μV. The input n will be an integer on the range of 0-65535, after which it is converted into binary and transferred to the DAC via SPI. Naturally, before and after we send any message we need to open and close the relevant ports, which we see is covered by the lines sandwiching the `SPI.transfer` functions.

5.3.1 Measuring the background voltage

```
// Measure background voltage
  if(state == 1){
    DACwrite(65535);
    delay(10);
    for(int j=0; j<5; j++){
      bg += analogRead(A0);
      delay(10);
    }
    bg = bg/5;
    DACwrite(28000);
    delay(100);
    state = 2;
    }
```

Here we take advantage of the narrow injection interval[1] and step directly to the end of the DAC value range (ie 65535) to break the injection lock. We can then measure the background values (we take a simple average of 5 to average

[1] The narrow range of currents over which injection lock holds.

away random error) before setting the DAC value back to near the injection interval.

5.3.2 PID loop

We are using the very common PID loop to generate our correction. The equation for such a correction is shown:

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt} \tag{5.1}$$

where K_p, K_i, K_d are the coefficients for the proportional, derivative and integral terms respectively, and $u(t)$ and $e(t)$ are the correction and error terms. Though the mechanism is simple, there is a need to tune the proportional, integral and differential coefficients to better fit our system. We manually tuned these parameters until the error signal has a good behaviour: it has a low steady state error[1] and a fast rise time. When we were tuning the PID parameters, we made some observations:

- A system with just proportional gain tended to have a large steady state error, as well as severe oscillations. (Fig 5.3)
- Adding in an integral gain damped down the steady state error, but added noise if the gain was set too high. (Fig 5.4)

We finally found satisfactory parameters for our PID loop to be optimised. Fig 5.5 is a data sample of the fluctuation of the error signal for a 8-hour period. We see the good behaviour mentioned earlier. When left running, the laser system stayed locked over a weekend, so that was a satisfactory test of our firmware.

[1] A low steady state error means it does not deviate too much from zero when the error signal is stable.

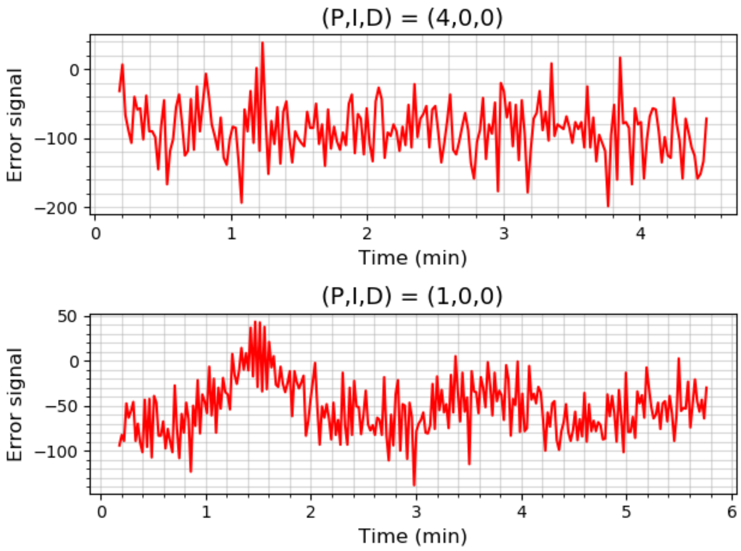


Figure 5.3: A comparison of the behaviour of the error signals when we vary the proportional gain. With a low proportional gain, there is a huge steady state error (top). We can remove this by turning up the proportional gain, but this comes at the cost of introducing oscillations (below).

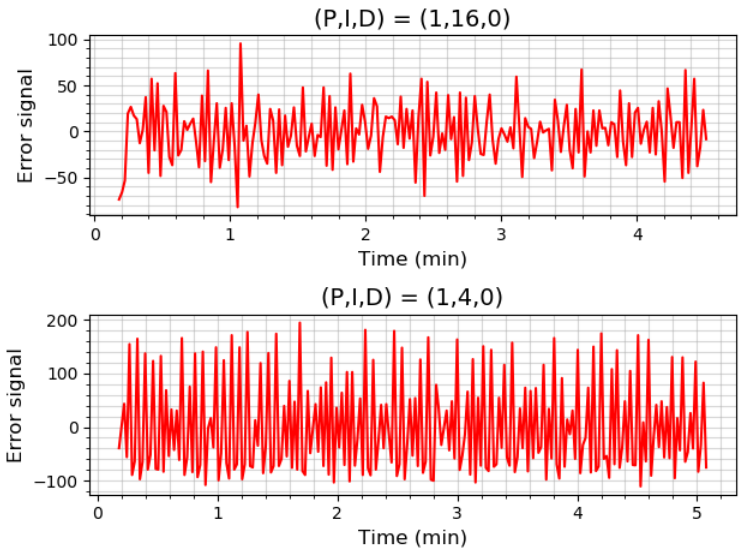


Figure 5.4: A comparison of the behaviour of the error signals when we vary the integral gain. We can see adding integral gain removes steady state error (top), but too much of it generates noise (below).

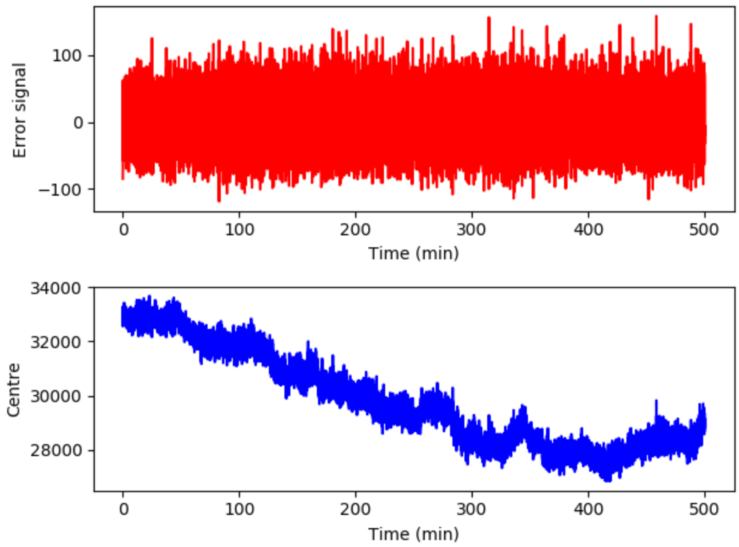


Figure 5.5: An 8-hour long behaviour of the error signal and DAC value of optimal lock.

5.4 Lock Acquisition Switch

Lastly, we would like to mention a feature that we have implemented for our firmware. We added a switch to the system, such that we are able to pause the running of our firmware (without unplugging the Arduino from the board). This is to allow us to easily reoptimise the system if there are any issues like if the master laser unlocks/mode hops. Once we flip the switch back on, the firmware restarts, once again going through the whole process of recording the background reading, sweeping out the system response curve and generating the error signal.

Appendix 1 - Cooling systems

As described in the earlier chapter, we need very low temperatures in order to condense a BEC: somewhere in the order of To reach such low temperatures, conventional cooling methods are insufficient. 2001 Nobel laureates Cornell, Ketterle and Wieman all used a combination of laser and evaporative cooling. In this chapter, we shall delve into such cooling methods, and how they lead to the need for our project.

A.1 Laser cooling

A.1.1 Doppler Cooling

The most common method of laser cooling, by far, is Doppler cooling. In Doppler cooling, six laser beams opposing each other in pairs are arranged in three directions and so are able to slow down and cool any atoms moving in the intersection of these beams.

The theory behind the feasibility of using lasers to cool particles is as follows: Light exhibits wave-particle duality, and so can be thought of as a stream of particles; photons. Although photons have no mass, they possess momentum, and so any impacts or collisions with other particles can be described by momentum conservation laws. As such, when colliding with atoms, if the photons involved have energies corresponding to the energy levels in the atoms, there is a chance for the atom to absorb the photon and gain its momentum. Thus, the atoms slows down if the photon has an opposing velocity. After a short de-excitation time, the atom emits a photon, and can now absorb a new photon from the beam of light. Upon emission the atom will experience a recoil energy, but since the direction of the recoil is random, over many cycles of absorption and emission the recoil is negligible compared to the reduction in velocity brought about by the absorption of the photon. The use of a laser beam should also be clear: the energy of the photons in the beam of light must be equal to that of the atom energy level, and a laser beam has coherent properties.

Since the atom is not stationary, we need to consider the Doppler effect. The Doppler effect, named after Christian Doppler who described this phenomenon in 1842, refers to the change in frequency or wavelength of a wave in relation to an observer who is moving relative to the wave source. Placed into perspective, an atom moving into our laser beam will see the frequency of the beam increased. Thus we suitably redshift the laser beams, or in other words decrease the frequency so that the approaching atom perceives the frequency of the laser to be increased to match its energy levels, thus allowing for absorption of the photon.

This method was developed by Steven Chu[13] in 1985 and was also known as Doppler cooling. However, this is not a real trap, as gravity causes them to fall out of the laser setup in about 1 second. The solution? Usage of magnetic fields.

A.1.2 Magneto-Optical Traps

The setup for the usage of magnetic fields on top of the lasers used for Doppler cooling are known as Magneto-Optical Traps, or MOTs for short. The critical idea behind how a MOT is able to successfully trap an atom is due to the Zeeman effect. This is the effect of the splitting of an energy level into several components in the presence of a static magnetic field. In a MOT, the spatially varying magnetic quadrupole field will cause a Zeeman shift to the m_f levels, which will increase as it gets further away from the center of the trap. As such, the atomic resonance is tuned closer to the frequency of the laser beams, and will be more likely to get kicked towards the center of the trap. Now the direction of the momentum kick the straying atoms receive from the beams can be left or right circularly polarised, and as such will have differing interactions with the different m_f levels. For a successful MOT to operate, care has to be taken to choose the appropriate polarisations of light so that photons from the laser beam that are moving towards the centre of the trap will always be on resonance with the atomic energy level, thus maintaining the effect of kicking the atoms to the center of the trap.

All that being said, not every atom is suitable to be magneto-optically trapped. There needs to be a closed optical loop for easy optical cooling. This closed optically is essentially a loop where the the excited atom will decay back to its original energy level. Take ^{8}5Rb as an example: It has a closed optical loop between the $5S_{1/2}$ F=3 and the $5P_{3/2}$ F=4 state. This is because once the atom is in the excited state, a decay to any of the $5P_{1/2}$ states is a forbidden transition as this would not conserve parity, and decaying to the $5S_{1/2}$ F=2 state is also forbidden because this would require an angular momentum change of 2, which cannot be supplied by a single photon. There is still hope for those atoms that do not have closed optical loops: we can use repump lasers. These repump lasers can re-excite atoms back into the optical loop when they have decayed to a state out of the optical loop. We shall again use ^{8}5Rb as an example: The detuning of the lasers for optical cooling will result in a small overlap with the $5P_{3/2}$ F=3 state. After a thousand or so cycles, the atom may be excited to this state, and is then free to decay to either the F=3 or F=2 state. If it decays to the F=2, the atom breaks out of the optical cycle and will no longer be cooled or trapped by the MOT. We need to use a repump laser which is on resonance with the $5S_{1/2}$ F=2 to $5P_{3/2}$ F=3 transition to re-excite the atom back into the closed optical loop.

A.2 Evaporative Cooling

This technique is present in our everyday life: the cooling sensation when sweat evaporates, how a hot cup of coffee cools. Surprisingly, we are able to use such a seemingly simple method of cooling to reach one of the coldest temperatures ever observed in the Universe: nano-Kelvin temperatures in atoms traps.

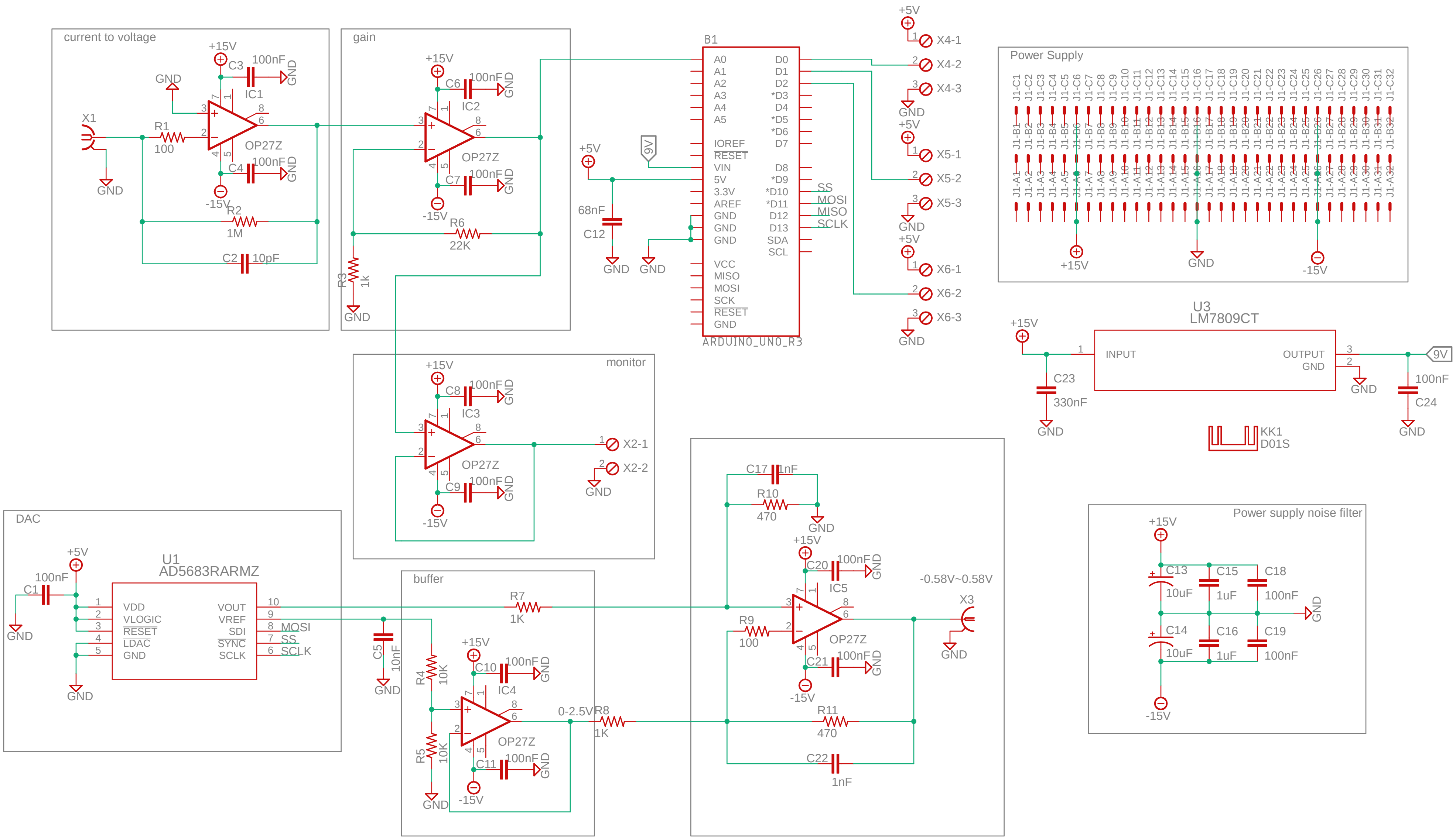
Evaporative cooling describes the process of energetic particles leaving a system with a finite binding energy.[10] This is a natural and automatic process as there are high energy particles at the tail end of the Maxwell-Boltzmann distribution. When these energetic particles evaporate and leave the system, they carry with them a certain amount of energy. Since energy is lost from the system, there is naturally cooling of the system.

In the case of BECs, it is noticed that this technique of evaporative cooling can increase the phase space density of the system to be cooled by up to six orders of magnitude.[10] This process was also used to reach temperatures that were unprecedented, and greatly exceeded what had been reached by laser cooling. Although laser cooling has broken the recoil limit in three dimensions, it is unlikely to work for larger densities of atoms. Thus evaporative cooling is a more favourable process as it can work over a large range of temperatures and densities.

One obvious disadvantage that others might be quick to point out is that evaporative cooling will result in a loss of atoms. Although this might be true, a six orders of magnitude increase in phase-space density only required losing a factor of about 1000 in the number of atoms. There are now various methods of increasing the load of a trap, and so evaporative cooling's transformation of higher initial load to higher final densities and lower temperatures will undoubtedly spur more efforts into obtaining larger samples of trapped atoms.

Appendix 2

B.1 Circuit Schematic


current to voltage
gain
monitor
DAC
buffer
Power Supply
Power supply noise filter
B1
ARDUINO_UNO_R3
A0
A1
A2
A3
A4
A5
IOREF
RESET
VIN
5V
3.3V
AREF
GND
VCC
MISO
MOSI
SCK
D0
D1
D2
*D3
D4
*D5
*D6
D7
D8
*D9
*D10
*D11
D12
D13
SDA
SCL
SS
SCLK
X1
X2-1
X2-2
X3
X4-1
X4-2
X4-3
X5-1
X5-2
X5-3
X6-1
X6-2
X6-3
IC1
IC2
IC3
IC4
IC5
OP27Z
R1 100
R2 1M
R3 1k
R6 22K
R4 10K
R5 10K
R7 1K
R8 1K
R9 100
R10 470
R11 470
C1 100nF
C2 10pF
C3 100nF
C4 100nF
C5 10nF
C12 68nF
C17 1nF
C22 1nF
C23 330nF
C24 100nF
C13 10uF
C14 10uF
C15 1uF
C16 1uF
C18 100nF
C19 100nF
U1
AD5683RARMZ
VDD
VLOGIC
RESET
LDAC
VOUT
VREF
SDI
SYNC
U3
LM7809CT
INPUT
OUTPUT
KK1
D01S
9V
+15V
-15V
+5V
0-2.5V
-0.58V~0.58V

B.2 Firmware Code

```
//Include the SPI Library
#include<SPI.h>

void setup() {

  pinMode(2,INPUT);

  SPI.beginTransaction(SPISettings(14000000,MSBFIRST,
     SPI_MODE1));
  SPI.begin();

  Serial.begin(9600);
}

void DACwrite(unsigned int n){
  unsigned int n1 = floor(n/4096);
  unsigned int n2 = n - 4096*n1;
  PORTB &= B11111011;
  SPI.transfer(48+n1);
  SPI.transfer16(16*n2);
  PORTB |= B00000100;
}

int find_peak(int x){ // input argument is the
   background reading
```

```
  int peak_values[50];
  int peak = 0;
  int peak_counter = 0;
  int timeout_counter = 0;
    while (peak_counter<15){
      if(analogRead(A0) > 1.5*x){
        for(int j=0; j<50; j++){
          peak_values[j] = analogRead(A0);
        }
        int peak0 = peak_values[0];
        for(int j=1; j<50; j++){
          if (peak_values[j] > peak0){
            peak0 = peak_values[j];
          }
        }
        if(peak0 > peak){
          peak = peak0;
        }
        peak_counter++;
      }
      timeout_counter++;
      if(timeout_counter >= 1000){
        break;
      }
    }
    return peak;
}
```

```
//Define global variables
int state = 0;
int bg = 0;
unsigned int centre = 32768;
unsigned int left;
unsigned int right;
int T = 1000;
float error1 = 0.0;
float error2 = 0.0;
float error3 = 0.0;

void loop() {

// If lock switch is on and state = 0, start lock
   aquisition
  if(state == 0 && digitalRead(2) == HIGH){
    state = 1;
  }

// Measure background voltage
  if(state == 1){
    DACwrite(65535);
    delay(10);
    for(int j=0; j<5; j++){
      bg += analogRead(A0);
      delay(10);
```

```
    }
    bg = bg/5;
    DACwrite(28000);
    delay(100);
    state = 2;
    }

// Find max value, stepping values
  if(state == 2){
    int find_max[51];
    int max_counter = 0;
    for(unsigned int j = 28000; j <= 43000; j+=300){
      DACwrite(j);
      delay(10);
      find_max[max_counter] = find_peak(bg);
      max_counter++;
    }
    int smooth_max[51];
    for(int j = 2; j<=48; j++){
      int avg=0;
      for(int i = -2; i<=2; i++){
        avg+=find_max[j+i];
      }
      smooth_max[j] = avg/5;
    }
    int max_value = 0;
    for(int j = 0; j < 51; j++){
```

```
      if(smooth_max[j]>max_value){
        max_value = smooth_max[j];
        centre = 28000 + 300*j;
      }
    }
    DACwrite(centre);
    delay(100);
    for(int j = 0; j < 50; j++){
      float val1 = 0.97*max_value - find_max[j+1];
      float val2 = 0.97*max_value - find_max[j];
      if(val1 < 0 && val2 >= 0){
        left = centre - (28000 + 300*j);
      }
      if(val1 >= 0 && val2 < 0){
        right = 28000 + 300*(j+1) - centre;
      }
    }
    state = 3;
  }

//Initialize error variables
  if(state == 3){
    int left_side;
    int right_side;

    // Measure left side of system response curve (for
       computing error signal)
```

```
DACwrite(centre-left);
delay(10);
left_side = find_peak(bg);

// Measure right side of system response curve (for
    computing error signal)
DACwrite(centre+right);
delay(10);
right_side = find_peak(bg);

// Initialize error signal variable
error2 = 1000.0*(right_side - left_side)/(right +
   left);

delay(1000);

// Measure left side of system response curve (for
   computing error signal)
DACwrite(centre-left);
delay(10);
left_side = find_peak(bg);

// Measure right side of system response curve (for
    computing error signal)
DACwrite(centre+right);
delay(10);
right_side = find_peak(bg);
```

```
    // Initialize error signal variable
    error1 = 1000.0*(right_side - left_side)/(right +
       left);

    delay(1000);

    state = 4;
  }

//Lock to peak
  if(state == 4){
    int left_side;
    int right_side;

    // Measure left side of system response curve (for
       computing error signal)
    DACwrite(centre-left);
    delay(10);
    left_side = find_peak(bg);

    // Measure right side of system response curve (for
        computing error signal)
    DACwrite(centre+right);
    delay(10);
    right_side = find_peak(bg);
```

```
// Store previous error measurements for PID
   correction
error3 = error2;
error2 = error1;

// Compute new value of error signal
error1 = 1000.0*(right_side - left_side)/(right +
   left);

// PID constants
float Kp = 1.0;
float Ki = 4.0;
float Kd = 1.0;
float dT = (T + 320.0)/1000.0;

// Compute correction to central value
int delta = round(Kp*(error1-error2) + Ki*dT*error1
    + (Kd/dT)*(error1-2*error2+error3));

// Prevent large output swings due to glitches
if(delta < 5000 && delta > -5000){
  centre += delta;
  DACwrite(centre);
}

// Transmit error signal data
Serial.print(millis());
```

```
    Serial.print(',');
    Serial.print(error1,4);
    Serial.print(',');
    Serial.println(centre);

    // Hold system at centre of response curve
    delay(T);

    // If the lock is switched off, reset the centre
       variable to the centre of the DAC range
    // and reset the state varible, which prepares the
       system to be relocked
    if(digitalRead(2) == LOW){
      centre = 32768;
      DACwrite(centre);
      state = 0;
    }
  }

}
```

Bibliography

[1] R. Adler. “A Study of Locking Phenomena in Oscillators”. In: *Proceedings of the IRE* 34.6 (1946), pp. 351–357. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1946.229930.

[2] Dennis Becker et al. “Space-borne Bose-Einstein condensation for precision interferometry”. In: *Nature* 562.7727 (2018), pp. 391–395. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0605-1. URL: https://doi.org/10.1038/s41586-018-0605-1.

[3] P. R. Berman. *Atom Interferometry.* Academic Press, New York, 1997. Chap. 9.

[4] B. J. Bloom et al. “An optical lattice clock with accuracy and stability at the 10-18 level”. In: *Nature* 506.7486 (2014), pp. 71–75. DOI: 10.1038/nature12941.

[5] Cheng Chin et al. “Feshbach resonances in ultracold gases”. In: *Rev. Mod. Phys.* 82 (2 Apr. 2010), pp. 1225–1286. DOI: 10.1103/RevModPhys.82.1225. URL: https://link.aps.org/doi/10.1103/RevModPhys.82.1225.

[6] S. Chu. “Laser Trapping of Neutral Particles”. In: *Scientific American* (1992), p. 71.

[7] C.N. Cohen-Tannoudji and W.D. Phillips. “New Mechanisms for Laser Cooling”. In: *Physics Today* (1997), p. 33.

[8] Andrew J. Daley. *Quantum Computing and Quantum Simulation with Group-II Atoms*. June 2011. URL: https://arxiv.org/abs/1106.5712.

[9] A. Goban et al. "Emergence of multi-body interactions in a fermionic lattice clock". In: *Nature* 563.7731 (2018), pp. 369–373. DOI: 10.1038/s41586-018-0661-6.

[10] Wolfgang Ketterle and N.J. Van Druten. "Evaporative Cooling of Trapped Atoms". In: ed. by Benjamin Bederson and Herbert Walther. Vol. 37. Advances In Atomic, Molecular, and Optical Physics. Academic Press, 1996, pp. 181–236. DOI: https://doi.org/10.1016/S1049-250X(08)60101-9. URL: http://www.sciencedirect.com/science/article/pii/S1049250X08601019.

[11] Krzysztof Kowalski et al. "Magneto-optical Trap: Fundamentals and Realization". In: (Jan. 2010).

[12] NobelPrize.org. *Information for the Public*. Last accessed 10 March 2019. 2001. URL: https://www.nobelprize.org/prizes/physics/2001/popular-information/.

[13] NobelPrize.org. *The Nobel Prize in Physics 1997*. Last accessed 12 March 2019. 1997. URL: https://www.nobelprize.org/prizes/physics/1997/press-release/.

[14] L. J. Paciorek. "Injection locking of oscillators". In: *Proceedings of the IEEE* 53.11 (1965), pp. 1723–1727. ISSN: 0018-9219. DOI: 10.1109/PROC.1965.4345.

[15] Helene Perrin. "Doppler cooling and magneto-optical trapping". Part of the *Les Houches lectures on laser cooling and trapping*. 2014.

[16] W.D. Phillips and H.J. Metcalf. "Cooling and Trapping Atoms". In: *Scientific American* (1987).

[17] B. Razavi. "A study of injection locking and pulling in oscillators". In: *IEEE Journal of Solid-State Circuits* 39.9 (Sept. 2004), pp. 1415–1424. ISSN: 0018-9200. DOI: 10.1109/JSSC.2004.831608.

[18] Brendan Saxberg, Benjamin Plotkin-Swing, and Subhadeep Gupta. "Active stabilization of a diode laser injection lock". In: *Review of Scientific Instruments* 87.6 (2016), p. 063109. DOI: 10.1063/1.4953589. eprint: https://doi.org/10.1063/1.4953589. URL: https://doi.org/10.1063/1.4953589.

[19] M R Tarbutt. "Magneto-optical trapping forces for atoms and molecules with complex level structures". In: *New Journal of Physics* 17.1 (Jan. 2015), p. 015007. DOI: 10.1088/1367-2630/17/1/015007. URL: https://doi.org/10.1088%2F1367-2630%2F17%2F1%2F015007.

[20] Carl E. Wieman and Leo Hollberg. "Using diode lasers for atomic physics". In: *Review of Scientific Instruments* 62.1 (1991), pp. 1–20. DOI: 10.1063/1.1142305. eprint: https://doi.org/10.1063/1.1142305. URL: https://doi.org/10.1063/1.1142305.

[21] T. van Zoest et al. "Bose-Einstein Condensation in Microgravity". In: *Science* 328.5985 (2010), pp. 1540–1543. ISSN: 0036-8075. DOI: 10.1126/science.1189164. eprint: http://science.sciencemag.org/content/328/5985/1540.full.pdf. URL: http://science.sciencemag.org/content/328/5985/1540.