



Pulse duration techniques and measurement using FROG and autocorrelation at FLASH

Jasmine Michalowska, University of Leicester, UK

September 7, 2016

Abstract

Pump-probe experiments can be used to investigate the evolution of ultrafast reactions, producing 'molecular movies'. In these experiments it is crucial to be able to measure the length of the laser pulses being used and confirm that the resolution is appropriate for the process being studied. This can be done using a technique derived from second harmonic autocorrelation, FROG (Frequency Resolved Optical Gating). This technique employs the basis of the Michelson-Morley interferometer and use of spectrometers that can be integrated into the laboratory setup and in our case controlled through the LabView virtual interface. In this report the integration and triggering of one RBG Photonics Qmini spectrometer into LabView is described. Autocorrelation analysis using Matlab to find the pulse duration for 15 data sets is also conducted.

Contents

1	Introduction	3
1.1	FLASH and producing THz radiation	3
1.2	Autocorrelation and FROG techniques	3
2	Integrating spectrometers into LabView	5
2.1	RBG spectrometer	5
2.2	Integration into LabView	6
3	Determining pulse length	8
3.1	Autocorrelation analysis	8
3.2	Results	10
4	Acknowledgements	11
5	References	13
6	Appendix	13
6.1	A	13
6.2	B	14

1 Introduction

1.1 FLASH and producing THz radiation

FLASH, the free electron laser situated at the DESY site in Hamburg, Germany, produces radiation in the extreme ultraviolet and soft x-ray range, 4.1-44nm in wavelength. Electron bunches are accelerated to relativistic speeds in 800 microsecond bursts and drift towards an undulator where neodymium magnets arranged to create an alternating magnetic field induces a Lorenz force upon the electrons. This creates a small cone of intense radiation in the forward direction.

Inside the undulator the electrons propagate the same path as the radiation they create. Hence a change in electron density occurs due to the electric field of the high intensity radiation interacting with the electron charge. This leads to microbunching where individual bunches radiate coherently, in turn causing an exponential growth of radiation energy until saturation is reached. This process, SASE, or Self-Amplified Spontaneous Emission is the principle behind light generation at FLASH. (Flash.desy.de, 2016)

At FLASH there is the possibility of a second undulator to be used in order to produce THz radiation. With this high frequency radiation coherent, femtosecond-picosecond laser pulses specifically for pump-probe experiments can be created. (Photon-science.desy.de, 2016)

The BL3 beamline allows both the THz and XUV pulses to overlap, providing a way to utilize the low frequency of the THz pulse in pumping where the XUV would then probe the samples (Photon-science.desy.de, 2016). When conducting pump-probe experiments the pump laser provides the light for excitation whereas the probe laser provides light to measure, for instance, the absorbance and hence any changes in the sample that occurs after the probe laser has disturbed it. (Pubs.acs.org, 2016)

1.2 Autocorrelation and FROG techniques

One of the major limitations in any measurement technique is the requirement of a smaller scale than the object being measured. In the femtosecond regime this is particularly difficult to achieve. Optical autocorrelation, and its derivative FROG (Frequency Resolved Optical Gating) is a technique that can bypass this problem.

In the late 1800s Michelson and Morley carried out a series of experiments originally designed to measure the motion of the earth through a substance they called 'ether' to further investigate the properties of light. No change in the speed of light was seen however the experimental set up they created can be utilised to measure the length of ultrafast laser pulses.

Figure 1 displays an example of a setup they used. Here light falls onto a beam splitter where part of the light is reflected from the silver surface onto a moveable mirror which returns light to the eye or detector. The light not reflected passes through the beam splitter and instead travels to the fixed mirror, where it is reflected back to the beam splitter then reflected to the eye. When the distance L_1 and L_2 are equal the two beams recombine, an interference pattern can be seen that disappears when one arm is lengthened,

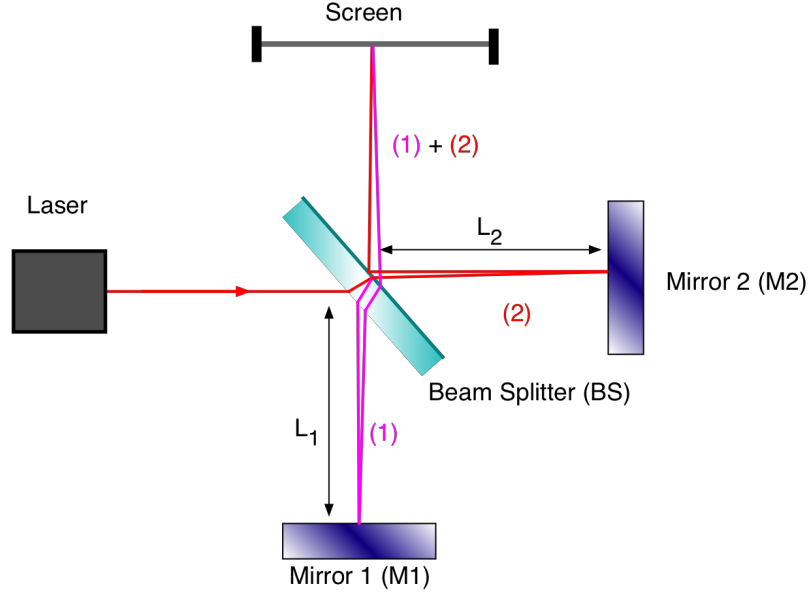


Figure 1: Diagram displaying the typical Michelson-Morley interferometer set up. The mirror M1 is moveable where M2 is fixed. When path lengths L_1 and L_2 are equal the beams combine an interference pattern can be seen on the screen. (Wanda.fiu.edu, 2016)

and hence the wavelength of the light can be determined through the use of equation (1). Here d is the path difference, λ the wavelength of light and m is an integer.

$$d = \frac{m\lambda}{2} \quad (1)$$

Autocorrelation techniques use a similar set up to the Michelson interferometer where a signal such as a light pulse is correlated with itself at different points in time, hence measuring the ultrashort pulse against itself, overcoming the measurement scale limitation.

When autocorrelation is used in an optical setting it can be experimentally realised as either field or intensity autocorrelation. Field autocorrelation is used to calculate the spectrum of a source of light and is measured by using the typical Michelson interferometer set up, instead placing a spectrometer at the output. Intensity autocorrelation has a similar set up but involves two non collinear beams with a variable delay generated then focused on a second harmonic generation crystal. As shown in Figure 1 FROG also uses this arrangement, with a spectrometer placed where the screen would be.

Using FROG to characterize optical pulses allows for measurement of not just the pulse parameters but also the optical spectrum.

2 Integrating spectrometers into LabView

2.1 RGB spectrometer

The basic function of a spectrometer is to take light in through an opening and disperse in order to separate into its spectral components before then digitizing the signal as a function of wavelength. Figure 2 displays a simple schematic of the diffraction grating, optical table and detector found in a typical spectrometer.



Figure 2: Example of the inside of a typical spectrometer. Light enters at the point labelled 1 and follows the path shown, onto a mirror and diffraction grating before being reflected onto the detector. (BW Tek, 2016)

To make pulse duration measurements using FROG the spectrometer being used needs to be integrated into the laboratory set up. Typically the easiest way to do this is by incorporating it, alongside the ability to trigger and write data to file, into a LabView virtual interface.

The spectrometer a Labview VI was created for is a RGB USB Qmini Infrared spectrometer. It has a focal length of 75mm and specifically can detect Near Infrared wavelength ranges as shown in Table 1 below.

Table 1: Possible Wavelength ranges for RGB Qmini Spectrometer

Spectrum component	Wavelength range (nm)
UV	220-400
Visable	350-880
Near Infared	700-1040

2.2 Integration into LabView

In order to use the spectrometer in the THz beamline for the purpose of autocorrelation and FROG it needs to be easily controlled independently. Figure 3. displays the LabView front panel that was created in order to achieve this for the RGB Qmini Spectrometer.

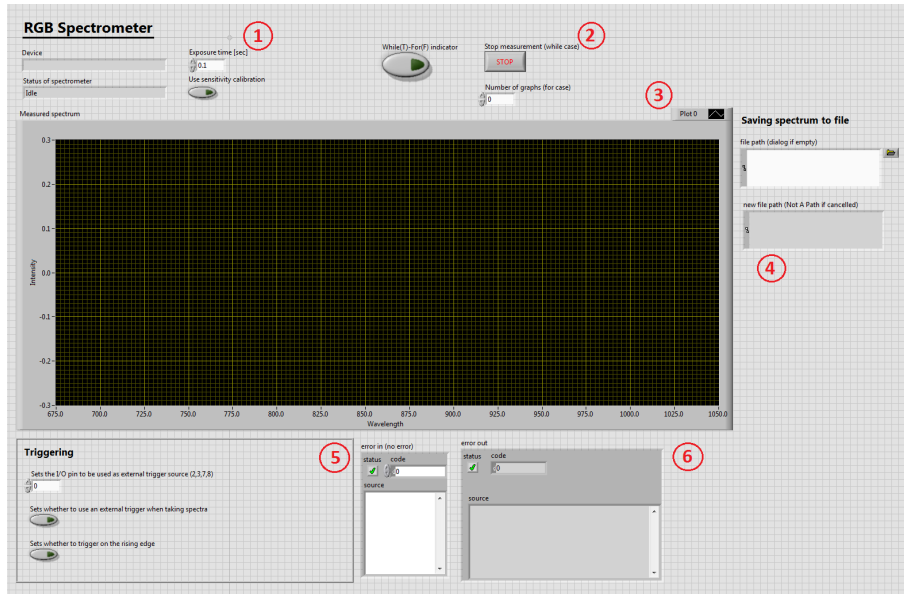


Figure 3: The front panel of the Labview VI for the RGB spectrometer, including the ability to externally trigger the spectrometer, save spectrum to file and run the spectrometer either for a pre-specified number of spectra or indefinitely until the program is manually stopped.

Referring to Figure 3, area 1, marked with the red circled number, displays the general controls for using the spectrometer. This involves the device name and status which is displayed when the spectrometer is connected to the set up, and by default displays 'idle'. When taking spectra or triggering this changes to 'taking spectra' or 'waiting for trigger', respectively. Controls for exposure and sensitivity calibration are also available to the user.

Area 2 displays the controls for running the spectrometer, with the options to run either in the 'True' case, where a while loop in the block diagram (displayed in Figure 4.) allows for an indefinite number of spectra to be taken until the user manually stops the program, or in the 'False' case, where a for loop allows for a specified number of spectra to be taken using the controls seen.

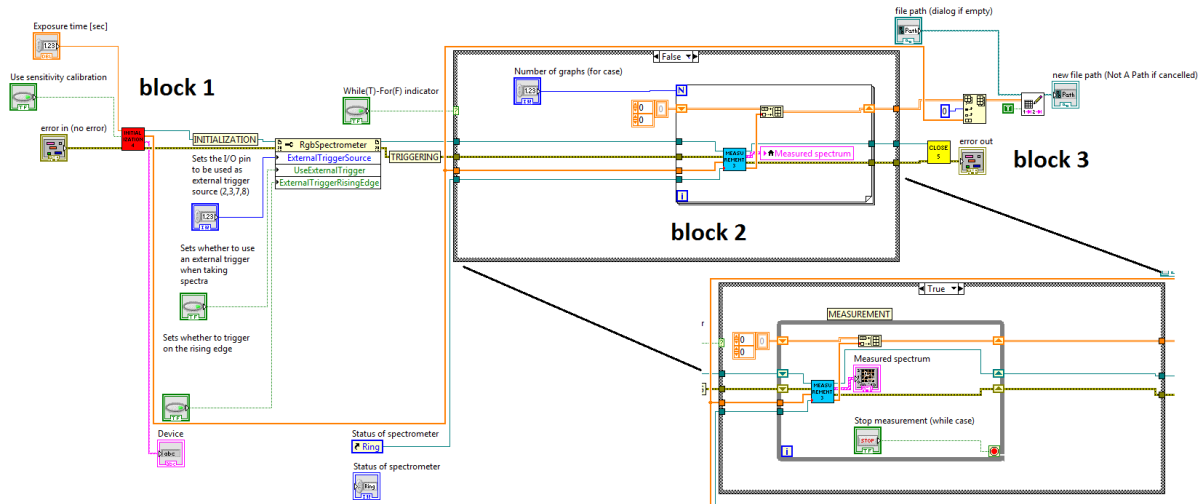


Figure 4: The main block diagram for the labview program showing both the true and false case for the two different options for taking spectra. Block 1, initialization includes the subVI for searching for and opening the spectrometer. Block 2, the measurement sub VI, with the option to take spectra continuously or for a set number of aquisitions is connected to the ability to write data to file and Block 3, the closing of the spectrometer sub VI.

Area's 3 and 4 display the spectrum to the user, plotting a wavelength vs intensity graph and also allow the user to save these spectra to a specified file location for future analysis.

Area 5 displays the control options for triggering the spectrometer. This is useful as in the normal mode of running the spectrometer is 'free'. This means it is continuously scanning, aquiring and transferring data to the computer. While doing this there is no way to synchronise the scanning, aquisition or transfer of data with an external event, this being particularly important when dealing with short time scales such as an ultrafast laser pule, and so in order to sync the aquisition an external triggering device can be connected so the spectrometer is triggered before the software recieves the data. This can be done in a number of ways including external triggering, in which an event outside the sampling system triggers the voltage levels on the spectrometer and hence initiates the data aquisition. it can also be done by triggering an extrnal event. Although the names are similar this second method involves the spectrometer instructing an external device, typically a lamp, to illuminate prior to spectral aquisition.

With external triggering, the mode accounted for here, edge triggering is used by a signal generator attached to a pin of the spectrometer. The spectrometer hence begins acquisition when it sees the rising edge of a voltage peak. If the trigger is continuously applied this mode is equivalent to operating in a normal mode.

Area 6 displays any errors that run through the program.

After spectra is taken and saved it can then be opened in Matlab to be analysed, including averaging if more than one spectra is taken. An example of the codes used to do this are included in Appendix A.

3 Determining pulse length

3.1 Autocorrelation analysis

As described earlier, autocorrelation can be used to determine pulse length. Consequently three sets of measurements are taken, scanning from 57 to 57.2mm of the stage position, with 0.5 micron step size and 1 micron delay stage step size and 101ms integration time, and one set of measurements scanning from 57 to 57.2mm with 0.5 micron step size and 1 micron delay stage step size and 200ms integration time.

The data obtained in the laboratory includes the stage position increments in half-micron steps and the corresponding intensities measured by the spectrometer for each spectral component. To integrate over the signal `imagesc()` can be used to display the data as a raw FROG trace in an array file that assigns a colour to each of the matrix positions alligning to each intensity value, with wavelength against stage position delay in the first two dimensions and intensity in the third (displayed as colour). Here, as seen in Figure 5, the second harmonic can be viewed as a bright yellow and green contrast against the blue background stretching across array positions 290-400. This range of wavelengths can then be averaged across all wavelength ranges and plotted against stage position to obtain the autocorrelation curve, also seen in Figure 6, with the peak alligning with the second harmonic on the raw FROG trace.

The background noise is already subtracted by the spectrometer prior to taking measurements. The background on the autocorrelation graph is also subtracted to reduce error due to blur on the FROG trace by taking the first 50 measurements of each data set, averaging them and individually subtracting this from each line in the main intensity vs stage position data set. This results in the graph overlayed on Figure 5, where the intensity begins at 0.

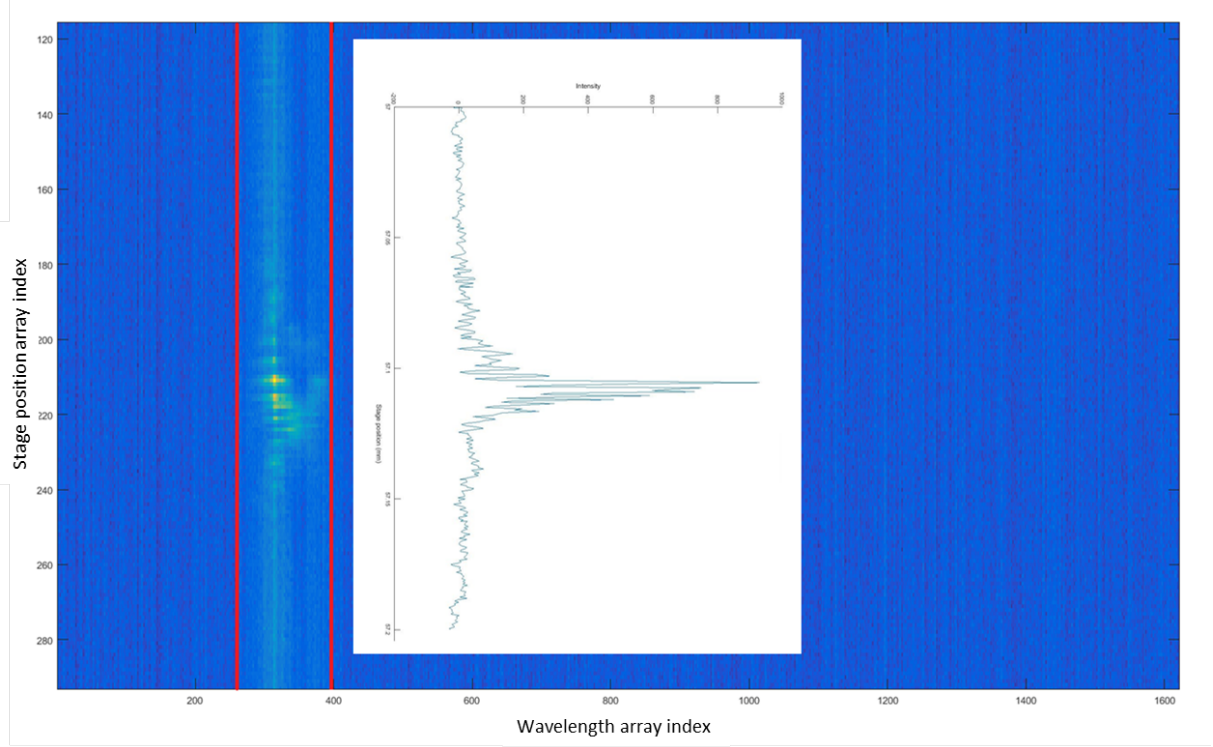


Figure 5: The raw FROG trace for 0.5micron step size and 101ms integration time. Plotted with wavelength index array on the x-axis, stage position index array on the y-axis and the values of intensity plotted on the z-axis and displayed as colour. The data between the two red lines are the index wavelength values data is taken from in order to produce the autocorrelation curve. Here the background-subtracted-autocorrelation curve is overlayed, with stage position plotted on the x-axis and intensity plotted on the y-axis. The peak of this graph corresponds to the second harmonic on the FROG trace.

From this autocorrelation peak the axis can be changed to time delay as opposed to stage position by using equation (2), where t equals time delay in femto seconds, X is the original stage position, X_{max} the stage position corresponding to the maximum intensity index number and c is the speed of light.

$$t = \frac{(X - X_{max}) \times 10^{12}}{c} \quad (2)$$

This is necessary in order for the pulse length to be calculated later on using the gaussian model seen in Figure 6. This is produced using the `fit(, , 'gauss1')` function in Matlab. From this the $FWHMac$, or the full width at half maximum of the autocorrelation pulse can be calculated using equation (3).

$$FWHMac = 2c_1 \sqrt{\log(2)} \quad (3)$$

Here the variable c_1 refers to the constant in the gaussian fit as seen in equation (4). Once the $FWHMac$ is obtained, the relationship seen in equation (5), (Diels and Rudolph, 2006) can then be used to obtain $FWHMp$, or the full width at half maximum of the pulse. This value directly corresponds to the length of the pulse, in femtoseconds.

$$f(x) = a_1 \exp(-[\frac{x - b_1}{c_1}]^2) \quad (4)$$

$$FWHMac = 1.414FWHMp \quad (5)$$

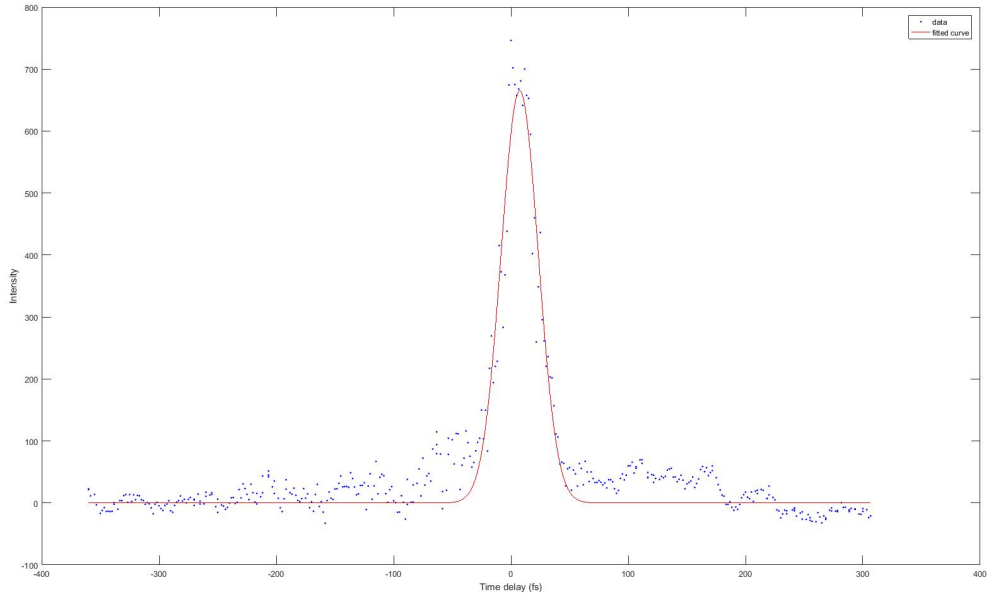


Figure 6: The gaussian fit for the 0.5 micron step size and 200 ms integration time data set is overlayed over the original data. Time delay in femtoseconds on the solid line is plotted against intensity on the dotted line. From the gaussian fit a $FWHMp$ of 27.70 is given.

The Matlab program to find the length of the pulse, as described above, can be found in Appendix B.

3.2 Results

The results for the 15 data sets taken are displayed in Table 2.

Table 2: Pulse lengths (FWHMP) in femtoseconds for all the separate data sets taken. 7-9,6 are in the scanning range 57 to 57.2mm with 0.5 micron step size and 101ms integration time. 10,0-10,4 in the scanning range 57 to 57.2mm with 0.5 micron step size and 200ms integration time.

Data set	FWHMP(fs)
7	28.24
8	28.57
9,0	29.41
9,1	29.56
9,2	29.52
9,3	30.45
9,4	30.14
9,5	26.72
9,6	24.94
10,0	27.70
10,1	29.18
10,2	26.38
10,3	26.65
10,4	28.10

There is some variation in the pulse lengths found in Table 2, this potentially being affected by the index intensity numbers chosen in the matlab code in Appendix B, as smaller ranges can potentially miss the second harmonic in the `imagesc()` plot.

In possible future work a Matlab program that averages over multiple data sets could have been completed, as this was begun, and would be useful for reducing the error in the pulse length due to noise in the data and extreme data points. An example which can be seen in Figure 6, where the centre of the gaussian is slightly shifted away from 0. The 'wings' in the underlying data can also be seen, potentially effecting the accuracy of the value obtained for the FWHMP. Depending on the pulse shape a two-gaussian fit, one for the lower intensity data collected, and one for the central peak, could potentially provide a much more accurate FWHMP.

4 Acknowledgements

I would like to thank Torsten Golz for being a fantastic supervisor: for making the internship really enjoyable and teaching me a great deal, not just about the science being conducted at FLASH but also helping me to program and figure out the solutions to problems myself- even when they seemed impossible. Also for the endless biscuits and cake provided at coffee breaks and for introducing me to Beginner's Advanced Chemistry album!

Marc Temme for all his help throughout the project, his patience in teaching us how to screw the vacuum chamber and for all his (terrible) jokes.

Andrea for letting us use his data in this report and for solving all the LabView problems I had (there were many).

Nikola Stojanovic for welcoming us into the FLASH group.

And finally, Sandra Beauvarlet, for being the best lab partner and friend I could wish for, whose intelligence and help I otherwise would not have been able to learn and progress so much in this project without and whose extraordinarily, endless supply of excitement and kindness contributed to what has been a really amazing summer.

5 References

- [1] DESY - FLASH - Accelerator. Available at: desy.de/accelerator [Accessed 7 Sep. 2016]
- [2] Photon-science.desy.de. (2016). THz Undulator Beamline @ BL3. [online] Available at: photon-science.desy.de/facilities/flash/beamlines/thz_beamline_flash1/index_eng.html [Accessed 7 Sep. 2016]
- [3] BW Tek. (2016). Part 1: The Slit - BW Tek. [online] Available at: bwtek.com/spectrometer-part-1-the-slit [Accessed 7 Sep. 2016]
- [4] Pubs.acs.org. (2016). Femtosecond Time-Resolved Photoelectron Spectroscopy - Chemical Reviews (ACS Publications). [online] Available at: pubs.acs.org/doi/abs/10.1021/cr020683w [Accessed 7 Sep. 2016]
- [5] Wanda.fiu.edu. (2016). [online] Available at: wanda.fiu.edu/teaching/courses/Modern_lab_manual/images/michelson.png [Accessed 7 Sep. 2016]
- [6] Diels, J. C and Rudolph, W (2006), Ultrashort Laser Pulse Phenomena, Second Edition, London, Elsevier Inc, pp 447

6 Appendix

6.1 A

Matlab code for plotting of the average of all graphs acquired with the RGB 4000 usb spectrometer and accompanying laview program.

```
%Plots only the average
%clear all
%close all

%enter the FILE NAME + extension if needed
data=load('NEW.txt');
sdata=size (data);

mat=zeros(1,sdata(1,2));

for c=1:sdata(1,2)
s=0;
for l=2:sdata(1,1)
s=s+data(l,c);
end
mean=s./(sdata(1,1)-1);
```

```

mat(1,c)=mean;
end

plot(data(1,:),mat(:,:),'-r')
xlabel('wavelength (nm)')
ylabel('Mean Intensity as normalized ADC value')
graphtitle=sprintf('Average on (\%d measurement with rgb
photonic spectrometer',sdata(1,1)-1);
title(graphtitle)

```

6.2 B

Matlab code for obtaining the image(sc) graph, autocorrelation peak and length of the laser pulse, for one data set at a time.

```

%obtaining the autocorrelation function for data set 7, scan from 57 to
%57.2mm, 0.5microm resolution and 101ms integration time
spectraldata_7 = load ('autocorr_data_7spectral_data_usb_4k__0.txt');
%stage position(row 1), then intensity for each wavelength in var2
spectralvalues_7 = load ('autocorr_data_7_0_spectral_values.txt');
%wavelengths data taken for each stage position
size(spectraldata_7);
size(spectralvalues_7);
only_intensity=spectraldata_7(:,2:end); %getting rid of stage position column
%figure;
%imagesc(spectraldata_7);
%displaying data to obtain bandwidth range to do autocorrelation with
selectbandwidth=spectraldata_7(:,300:340);
%obtaining select bandwidth range where second harmonic is
stageposition=spectraldata_7(:,1);
%obtaining only the stage position
%figure(2);
%plot(stageposition,selectbandwidth);
%plotting the range the second harmonic is thought to be seen in imagesc(var1)
averageofbandwidth=mean(selectbandwidth,2);
%average each row (intensity for a specific stage position over a range of
wavelengths) in the bandwidth range in variable a
%figure(3);
%plot(stageposition,averageofbandwidth);
%plotting stage position against average of bandwidths to get the
autocorrelation curve

%subtracting the background
bkgground=spectraldata_7(1:50,2:end);

```

```

%first 50 intensity over spectrum measurements
averageofbkgground=mean(bkgground);%averaging
var1_minus_background=bsxfun(@minus,only_intensity,averageofbkgground);
%minus the background from the intensity values
bandwidth_range=var1_minus_background(:,300:340);
averageofbandwidth_minusbackground=mean(bandwidth_range,2);
% figure(4);
% plot(stageposition,averageofbandwidth_minusbackground);
% xlabel('Stage position (mm)')
% ylabel('Intensity')%plotting new autocorrelation curve minus background

%changing the stage position axis to delay
stageaverage=[stageposition, averageofbandwidth_minusbackground];
[maxintensity,maxindex]=max(stageaverage(:,2));
%finding the max intensity and corresponding index number
maxstageposition=stageposition(maxindex,:);
%finding the max stage position corresponding to max intensity index number
stageposition_minusdelay=bsxfun(@minus,stageposition,maxstageposition);
%stage position-stage position(max)

timedelayseconds=(stageposition_minusdelay*(1.0e-3))/(3.0e8);
%changing to seconds
timedelayfemtoseconds = timedelayseconds/(1.0e-15);
% figure(5);
% plot(timedelayfemtoseconds, averageofbandwidth_minusbackground);

%fitting a gauss to the data
Timedelayfs=timedelayfemtoseconds;
Intensity=averageofbandwidth_minusbackground;
f = fit(Timedelayfs ,Intensity ,'gauss1')
%figure(6);
plot(f, Timedelayfs, Intensity)
xlabel('Time delay (fs)')
ylabel('Intensity')

%finding the FWHM
%maximum_Timedelayfs = f.b1 %this is one way to achieve FWHM but not the easiest
%maximum_Intensity = f.a1
%HalfMax= (maximum_Intensity)/2

sigma=(f.c1/2)
FWHMauto=sigma*(2*(2*log(2))^(1/2))
FWHMPulse=FWHMauto/1.414

```