Ultrafast pulse duration characterization: Autocorrelation, spectrometers and Frequency Resolved Optical Gating (FROG)

Sandra Beauvarlet, University of Bordeaux, France

July, 19th - September, 8th 2016

Abstract

This report is giving a summary of my work time at DESY as a summerstudent using lasers, spectrometers and programming software. The main purpose of this work was to acquire knowledge about ultrafast pulse duration characterization methods such as Autocorrelation and FROG. For this purpose, I had to become familiar with LabVIEW and MATLAB environments, create a data acquisition program for the spectrometer and write a program which analyses the data obtained from the autocorrelation experiments.

MATLAB

Group: DESY FS-FL FLASH

Supervisors: Nikola Stojanovic & Torsten Golz

FLASH.
Free-Electron Laser FLASH

NATIONAL INSTRUMENTS LabVIEW

Contents

1 Introduction

Running since 2005, the DESY Free-electron LASer in Hamburg (known as FLASH) is the first free-electron laser for soft X-ray radiation.

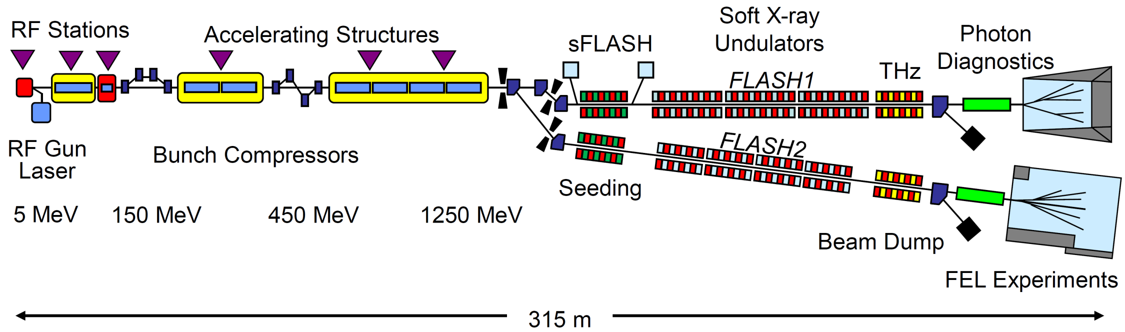

Figure 1: **Layout of FLASH facility. Here is shown the pathway of the electron bunches since their insertion and where the radiation produced goes. The electron bunches are being compressed, accelerated, forced to wiggle in the undulators and emit radiation, and then dumped. The radiation produced are diagnosed and then send to the different beam lines in the experimental halls.**

As we can see in **Figure 1**, FLASH works using electron beams which are accelerated, go through the undulators where the alternating electromagnetic field causes the electrons to undulate and thus to emit photons (**Figure 2**). Along the undulators, the electrons experience the Self-Amplified Spontaneous Emission (SASE) process producing intense and coherent radiation [5]. Once radiation is produced, the electron bunch is dumped and the photon beam is sent to one of the several beam lines that we can work with in the experimental halls.

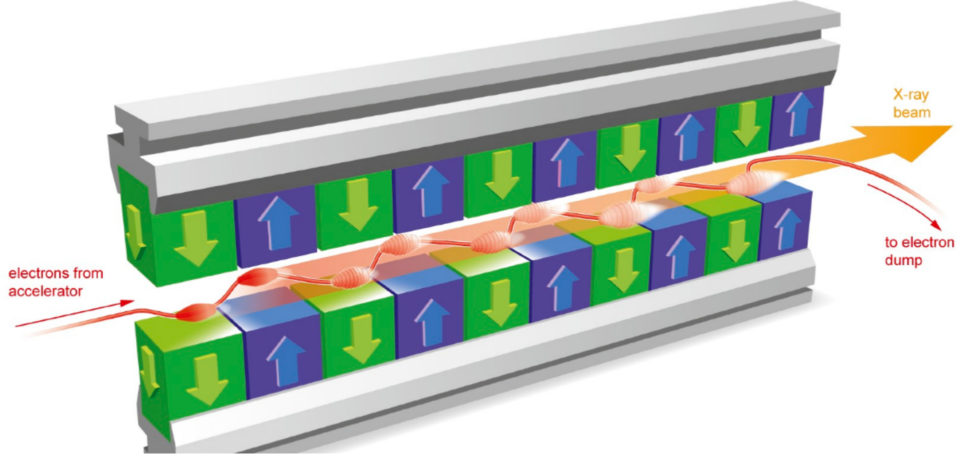

Figure 2: **Undulator principle: the alternative magnets (blue and green) create an alternating magnetic field which causes the electrons to wiggle and emit light along the undulator. Here is also represented the amplification of the intensity of the created radiation due to microbunching and SASE process.**

FLASH produces XUV pulses (duration: 30 to 300 fs / wavelength: 4.2 to 45 nm) using the first undulators, and THz pulses (duration: femtosecond (fs) to picosecond (ps) / wavelength: 10 to 230 μm) using the second undulator [4], the two pulses are naturally synchronized because they are generated by the same electron bunch [9]. The production of the THz pulses allows us to make pump-probe experiments using XUV pulses but we can also do THz/Laser pump-probe experiments.
However, pump-probe is a technique well used to study ultrafast processes such as femtochemistry. But to study such processes precisely, we need pulses shorter in time than the actual time length of the process. This comes from the well-known fact that to measure an event, we need a shorter event [2]&[5].
This is where being able to measure the duration of a pulse using ultrafast pulse duration characterization techniques is essential, thus where Autocorrelation and FROG are used.

2 Ultrafast Pulse duration measurement: Autocorrelation and FROG

To understand how FROG method work, we first have to introduce two essential notions called **Second Harmonic Generation** (SHG) and **Autocorrelation**.

2.1 Second Harmonic Generation (SHG)

The SHG process is also known as the frequency doubling. It occurs when two pulses of same frequency meet at the same time and same location in a specific non-linear material. These two pulses interact and generate a new pulse with twice the frequency of the inital pulses, thus the double of energy and half of the wavelength.
In our set up, the second harmonic is generated by overlapping two pulses (generated by the initial pulse equaly splitted) in a Beta Borate Bariyum (BBO) crystal.

2.2 Autocorrelation

The intensity autocorrelation set up (**Figure 3**) basically consist in a Michelson interferometer set up: We have one initial beam **(a)** divided by a beam splitter into two similar beams (50% transmitted and 50% reflected) propagating in 2 different part of the setup called arms. One arm is fixed **(b)** and the other one can be tuned in length **(c)**, what induce a difference in path length equivalent to a variable delay in time domain. The two pulses are then overlapped in the SHG crystal, generating the second harmonic pulse **(s)** from which we detect the intensity I(T)$\propto |E_{sig}(t,T)|^2$.

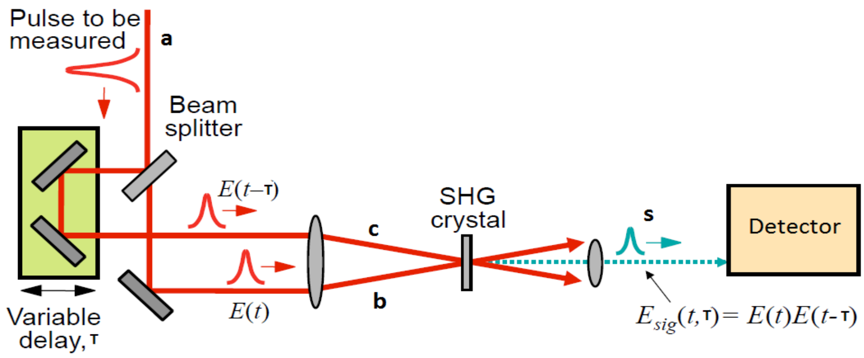

Figure 3: **Autocorrelation set up taken from [12]. The initial pulse is equally separated by the beam splitter and they follow different paths. One of the pulse is delayed and interfere with the other one in the non linear SHG crystal to produce second harmonic pulse, of which the intensity will be detected by the detector**

The intensity of the SH generated has to be measured as a function of the delay. From this measurement, one can extract the shape of the autocorrelation curve (more detailed in the **Autocorrelation results** section) which is the convolution of the two pulses **(b)** and **(c)**, in other words, the convolution of a pulse with itself-but-delayed, as follow:

$$I^{SHG}{}_{Auto}(T) = |\int_{-\infty}^{+\infty} E(t)E(t-T)\,\mathrm{d}t|^2 \tag{1}$$

where E(t) is the amplitude of the electric field of the pulse and T the delay.

In order to determine the pulse duration, few steps are still missing. We now have the shape of the autoccorelation of the pulse and we have to make assumptions on the shape that will fit the best. For simplest cases, one can try to fit the autocorrelation curve using the following functions: Gaussian, sech or Lorentzian ... Once the best-fit is found, we have to determine the Full Width at Half Maximum of the measured autocorrelation (FWHMac) which is related to the coefficients of the function used to fit the autocorrelation curve. And finally, knowing that the FWHMac is proportional to the FWHM of the initial pulse (FWHMp), by a factor that highly depends on the function we chose for the fitting, we can find the FWHM of the original pulse which is what characterizes the pulse duration [10].

2.3 FROG

As we can imagine, guessing shape is not a guarenteed bet and Autocorrelation does not give us enough information to be really accurate on the actual pulse shape. But FROG can solve this issue!
"The Frequency Resolved Optical Gating (FROG) simply involves spectrally resolving the signal beam of an autocorrelation measurement." said R. Trebino in his 1997 article *Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating*[1]. It is basically the same set up as the autocorrelation except that the dectector is replaced by a spectrometer. Now, the pulse representation is given as a function of two dimensions, not just the temporal dimension (delay) like in autocorrelation, but delay AND frequency (using the Fourier Transform of the signal (FFT)),which provides information concerning the phase of the pulse (see equation 2).

$$I^{SHG}_{FROG}(\omega, T) = |\int_{-\infty}^{+\infty} E(t)E(t-T)e^{-i\omega t}\,\mathrm{d}t|^2 \tag{2}$$

with w, the frequency and T, the delay.
From there, the next step is to use the Two-Dimensional Phase-Retrieval technique. Using Intensity and Phase as function of frequency and time, we can recreate the pulse shape with more accuracy and find its duration.

From this perspective, all our following work with spectrometers is essential for data acquisition in FROG and Autocorrelation.

3 USB 4k and RGB Photonics: Work on spectrometers

In this part, we had to become familiar with two spectrometers: The rgb photonics Qwave NIR (covering wavelength from 700nm-1040nm) and the Ocean Optics USB4000 (covering from 350nm to 1050nm). For the work explanation, we will focus on the rgb photonics' case.

3.1 LabVIEW integration

Starting with the continuous acquisition provided by rgb photonics, we had to understand how the VI works in order to modify and add functions to it. Our final goal was to have a program that grab and save data to study them afterwards (**Figure 4**).

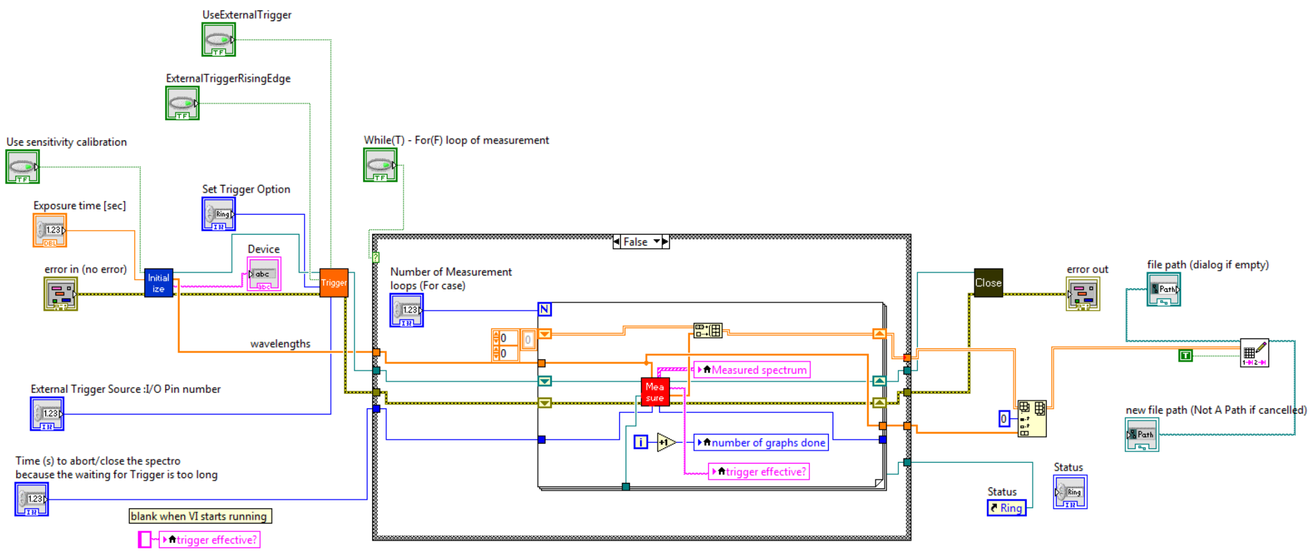

Figure 4: **Block Diagram of the data aquisition program in LabVIEW**

We thus built a program divided in 4 subVI:

1. Initialization, which opens and connects to the spectrometer. It also create an array with the wavelengths that the spectrometer is going to use and set the exposure time.

2. Triggering, which ask us if we want to use the external trigger (if it is the case one can choose the external I/O PIN to use) and offer several trigger options such as Free running or Hardware trigger

3. Measurement, which starts the exposure and the data aquisition once the trigger event occured. In addition, it plots the Intensity vs wavelength spectrum detected by the spectrometer. Through this subVI, we can also check the current status of the spectrometer ("Idle", "Waiting For Trigger"or "Taking Spectrum").
   ***Outside this subVI, we created a loop that controls the number of**

measurement cycle that we want and which saves each measurement data in an array that we save in spreadsheet at the end.* **The organization of the measurement subVI is presented in *Appendix 1***

4. Close, which simply closes the spectrometer once the acquisition is done and disconnects it.

In these 4 subVIs, a common error variable is used, which tells us if anything went wrong while the program was running, what kind of error it is and where to find it. Also, all of these information are controlled and/or indicated in the front panel. The organization of this front panel is presented in *Appendix 2*.

3.2 MATLAB processing

The next part was to create a MATLAB program able to plot every spectrum from the data saved in the spreadsheet and also one who plots the average of all theses data. Here, you can find the main code in MATLAB with the average plot that it gives.

```
%This program plots only the average of the whole set of data acquired
clear all
close all

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

mat=zeros(1,sdata(1,2)); % create a 0 row vector to save the average values

for c=1:sdata(1,2) % for each column
    s=0;            % Initiate the sum
    for l=2:sdata(1,1) % for each row
        s=s+data(l,c);  % sum the previous s with the value at index (l,c)
    end
    mean=s./(sdata(1,1)-1); % give the average
    mat(1,c)=mean;   % put the average value in the matrix
end

%create the graph
plot(data(1,:),mat(:,:),'-b')
xlabel('wavelength (nm)')
```

```
ylabel('Mean Intensity as normalized ADC value')
graph_title=sprintf('Average on %d measurement with rgb photonics ...
...spectrometer',sdata(1,1)-1);
  title(graph_title)
```

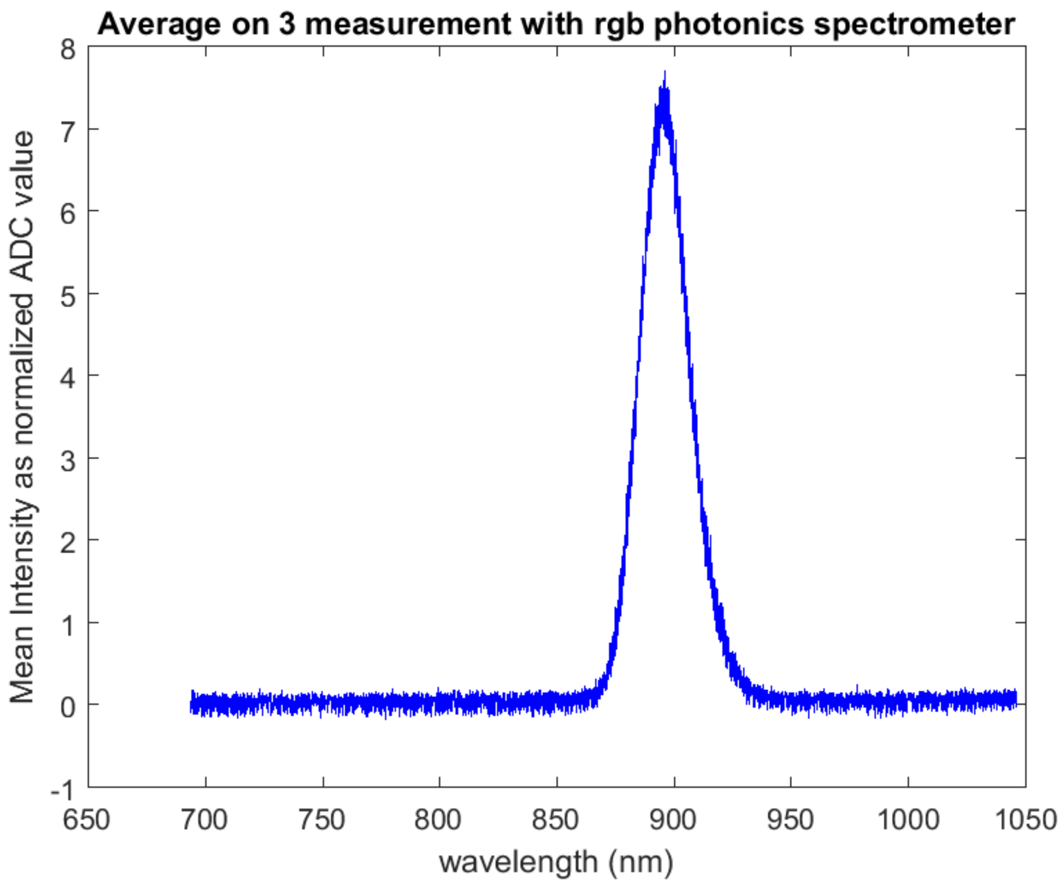

Figure 5: **figure given by the MATLAB data averaging program. On x axis, one sees the wavelengths and on y axis, the intensity normalized. This graph is an example using radiation from the computer screen detected by rgb photonics spectrometer**

4 Autocorrelation results

Using the autocorrelation set up presented in the previous section and the Ocean Optics usb 4k spectrometer, we were able to acquire some signal of the second harmonic generation data due to the overlaping of the two pulses in the BBO. Let's set here the parameters from the experiment:

4.1 Experimental parameters

- The pulse to be measured is given by a laser working at a wavelength= 860 nm
- The integration/exposure time of the spectrometer is set to 200 ms for the first experiment but can be changed
- The position of the delay stage range from 57mm to 57.2mm with a 0.5μm step

With these parameters fixed, we make 5 measurements that we analyse using the MATLAB program that we wrote (*Appendix 3*). From the data acquisition, we get a set of numbers organised in such way:

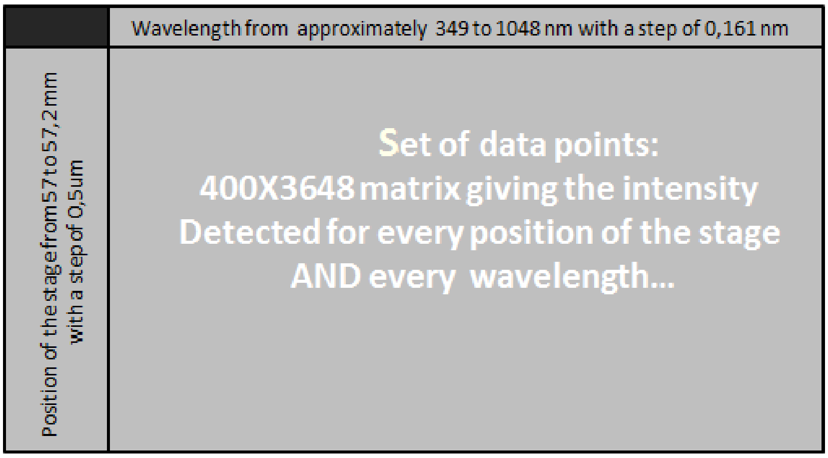

Figure 6: **Form of the acquired data to process**

4.2 MATLAB processing

From there, we have to find the wavelengths corresponding to the area of interest (**see in figure 7(a)**) using the colored 3D image. This selection will provide a clearer result when plotting the shape of the autocorrelation and also avoids using backgroung and cumulative noise (knowing that the SH is supposed to be equal to $SH = \frac{860}{2}nm = 430nm$ and that the spectrometer covers wavelength from 350 to 1050 nm roughly). Then, we average or sum (the two can be used because we only want the shape of the autocorrelation to appear but we chose to use the average) the values of the intensity over the previously selected range of interest in wavelengths and for each position of the stage (meaning that we average the data in a same row for a certain range of columns).

The next step is to plot this average against the delay (τ). In order to find the delay from the stage position, we have to find the maximum of intensity detected and find the corresponding position of the stage (x_{max}), then substract this value to the other stage positions values (x) and convert it in time, simply:

$$\tau = \frac{(x - x_{max})}{c} \tag{3}$$

with τ in s, c the speed of light $c \approx 3.10^8$m/s and x in m. We can now plot the autocorrelation curve vs the delay (**see in figure 7(b)**).

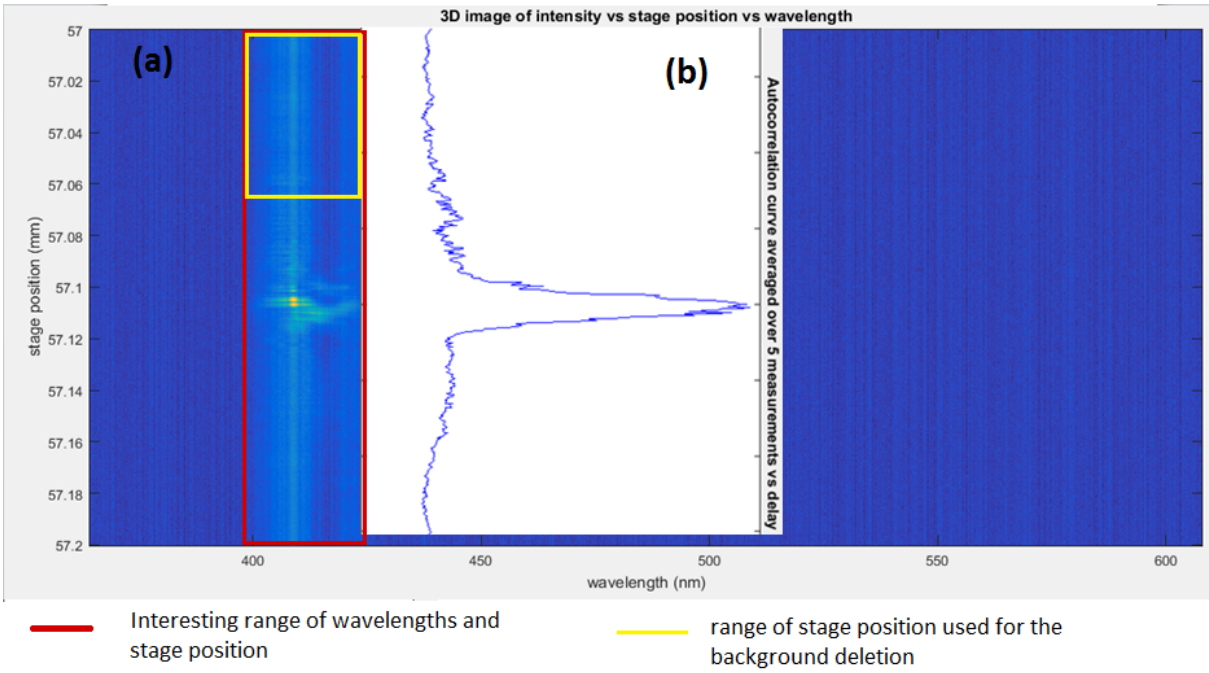

Figure 7: **(a) 3D image of the second harmonic generation at 408nm** (*blue to yellow* $\equiv$ *low to high intensity detected*) **for one of the 5 acquisition. We expected the second harmonic to be around 430 nm but that can be due to the spectral calibration of the spectrometer which was maybe not really accurate. It can also be due to the beam's angle of arrival into the entrance slit of the spectromter which can induce a shift of the spectrum with respect to the wavelengths. (b) Averaged intensity Autocorrelation curve vs stage position**

Now that the Autocorrelation curve is obtained, a fitting process is needed. We first

obtain the autocorrelation curve with a background (almost constant part on the y-axis that correspond to the light blue color in **figure 7(a)** in the yellow rectangle). This part should be subtracted in order to obtain a curve that start and end at approximately 0 in intensity (background free).
We finally have an autocorrelation curve that we can fit with a specific function. In our case we chose to fit it with a Gaussian function [11]:

$$f(x) = ae^{(\frac{x-b}{c})^2}$$

knowing that the parameter c is related to the FWHMac through:

$$FWHM_{ac} = 2c\sqrt{\ln 2} \approx 1.665c$$

The last step is to use the relation between the FWHMac and the FWHMp specific to Gaussian pulse shape [3]:

$$FWHM_p = \frac{FWHM_{ac}}{1.414}$$

4.3 Results

We actually did 2 sets of measurement (**table 1**) where we changed two parameters: the integration time of the spectrometer and the number of measurements withing the same set. Below is a table showing the results of the Gaussian fit compared to the direct calculations that we can make by using the actual experimental maximum and find the $FWHM_{ac}$ (as shown in **figure 8**).

Parameters	Experiment 1	Experiment 2
Number of measurements	5	7
Integration time	200 ms	101 ms
Results from the fit		
FWHM ac	40.41 fs	40.85 fs
FWHM pulse	28.58 fs	28.89 fs
Correlation coefficient R^2	0.9358	0.9153
Results from the experimental points		
FWHM ac	38.20 fs	38.34 fs
FWHM pulse	27.02 fs	27.11 fs
Relative error ($\lvert\frac{exp-fit}{exp}\rvert$)	5.8%	6.6%

Table 1: **Experimental parameters and results for Experiment 1 (FWHM experimental calculation in Figure 8) and Experiment 2 (same principle for the experimental calculation, figure in *Appendix 4*)**

The two experiments gave quite similar results (28.58 fs and 28.89 fs for the FWHM of the pulse which correspond to a 1% relative difference which is really low) which we could expect since every pulses are not supposed to be very different from each other.

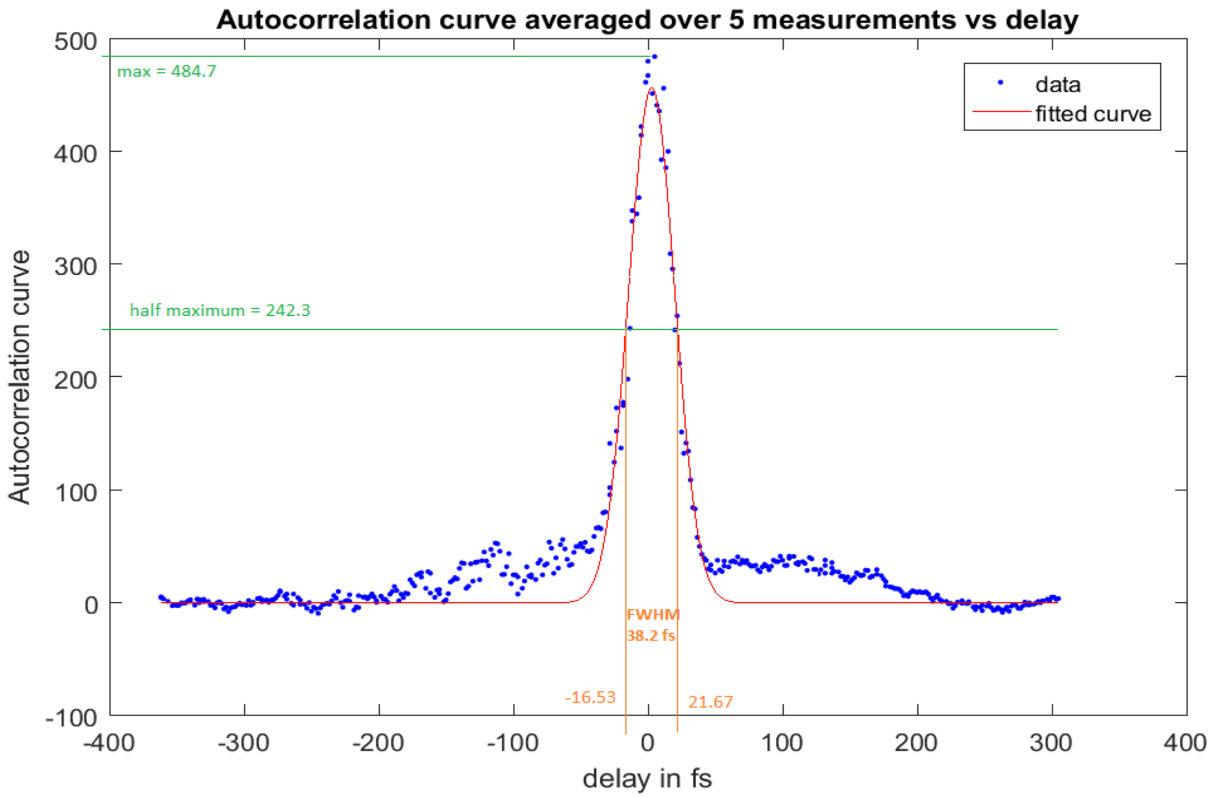

Figure 8: **Autocorrelation curve vs delay and its Gaussian fit for Experiment 1 : 5 measurements with an integration time of 200ms**

The errors between the experimental calculation and the Gaussian fit originate partly from the fact that in the two experiments, the Gaussian fit does not reach the maximum data points from the autocorrelation curve and it thus leads to a bad determination of the FWHM because the maximum considered is not the same. Also, the correlation coefficient is not very high because of the wings of our pulse (the parts in [-200; -50] fs and [+50;+200] fs), which are not well fitted by the Gaussian fit. However, the relative errors are quite small (less than 7 %) and with the correlation coefficient reasonably high ($R^2 > 0.9$), we can say that assuming a single Gaussian shape for the pulse is a sufficient approximation. Maybe fitting this pulse with 2 Gaussians will take care of the wings problem and give a better correlation coefficient and thus give us a more mathematically correct fit. But this approximation could be less physically correct, if the pulse is not a complex one.

5 Acknowledgement

- Nikola Stojanovic, thank you for allowing me to join your work group for this summer program and also for supervising me when you could.
- Torsten Golz, I am so grateful that you introduced me to LabVIEW! BEST SOFTWARE EVER! I really enjoyed every night shift that we had. Every part of the work you gave to us was really interesting to me, I really liked learning everything I could here, it was always a pleasure to ask you too many questions. And by the way, $1\mu m \equiv 3fs$ ... now I know.
- Marc Temme, I really had a great time sharing MY office with you but now it's all yours again, and Rui's too by the way. Thank you for everything you did to help me, thank your for all the jokes (Well, they are funny sometimes!), thank you for the secret sugar box for the coffee... You have made things really easy and pleasant for me while I was discovering work at DESY.
- Andreja Vladkovic, Thank you for allowing me to use your autocorrelation data, Thank you for all the fun we had in night shifts (the snapping "tutle" IS really scary!), thank you for the chocolate bar, for Berlin (flames!), for everything. I hope the corn starch will work!
- Jasmine Michalowska, I am really glad you were my partner in this whole program, it was so much fun! I hope you are going to keep good memories from LabVIEW and the "Waiting for Trigger" (that amazingly worked) and every other things we did together!!!

I am so glad to have met you all, I enjoyed working here so much because you all made it so interesting everyday. Thank you all for this amazing summer at DESY- FLASH...

6 References

[1] Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating,*Rick Trebino, Kenneth W. DeLong et al.*, (1997), chapters 1, 4, 5

[2] Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses, *Rick Trebino*, (2000)

[3] Ultrafast Laser Pulse Phenomena *Fundamentals, Techniques and Applications on a Femtosecond Time Scale*, Second edition, *Jean-claude Diels and Wolfgang Rudolph*, (2006), Chapter 9, section 9.4.3., page 477, table 9.1

[4] New infrared undulator beamline at FLASH, Infrared Phys. Technol. 51 , 423 425. (Article), *M. Gensch et. al.* , (2008)

[5] Diplom Thesis: Pulse duration measurement at 4th generation X-ray light sources - Frequency Resolved Optical Gating at FLASH,*Torsten Golz*, (2012)

Usefull websites

[6] *http://www.ni.com/labview/f/*

[7] *http://www.rgb-photonics.com/products/*

[8] *http://oceanoptics.com/product/usb4000-custom/*

[9] *http://photon-science.desy.de/facilities/flash/beamlines/thz_beamline_flash1/*

[10] *http://www.diss.fu-berlin.de/diss/servlets/MCRFileNodeServlet/FUDISS_derivate_00000000577*
- Pulse Characterisation Techniques

[11] *https://en.wikipedia.org/wiki/Gaussian_function*

[12] *http://www.swampoptics.com/assets/tutorials_autocorrelation-2015.pdf*

7 Appendix 1

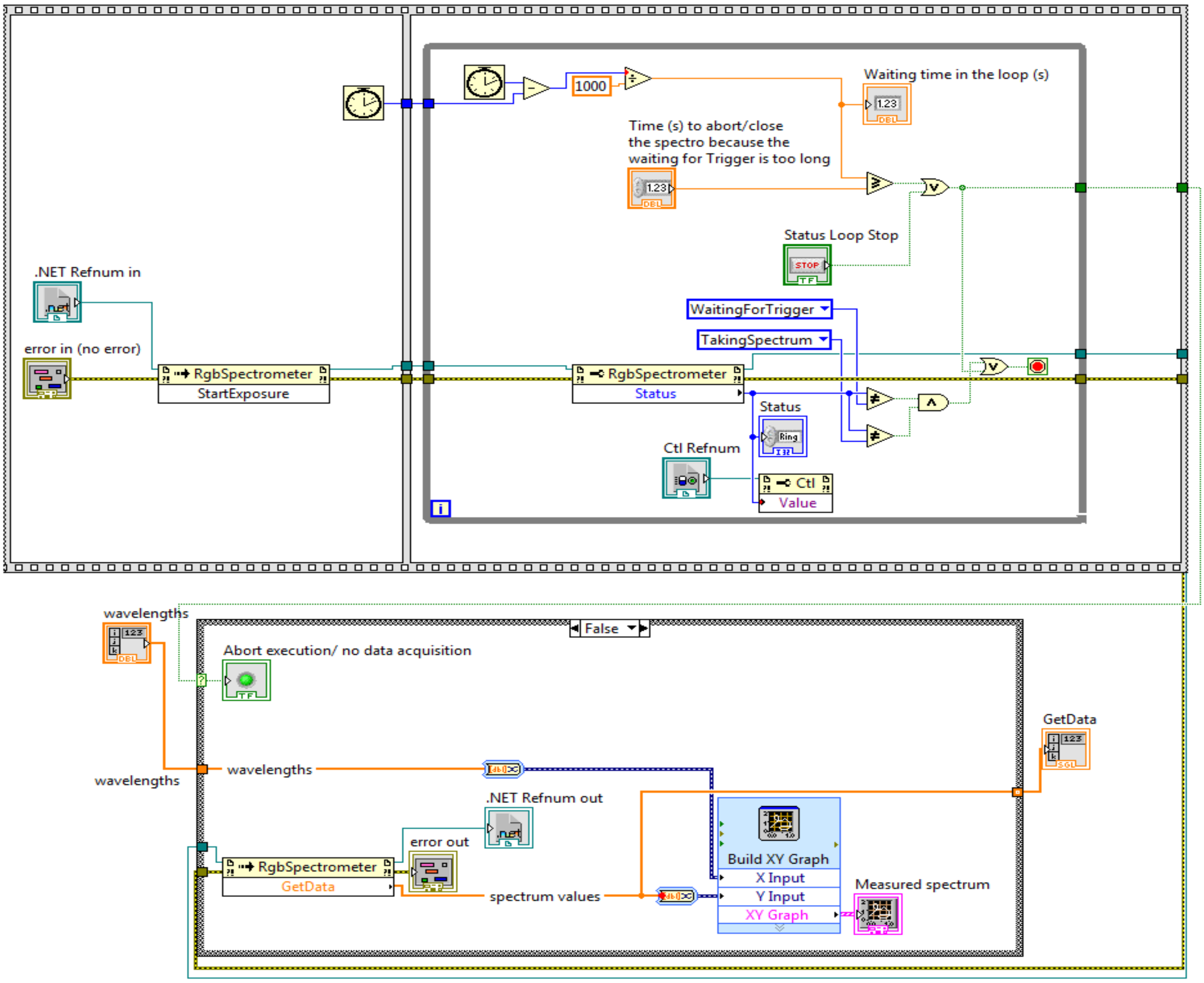

Figure 9: Measurement subVI starting the exposure, sending the current status of the spectrometer to the front panel and handling the condition to stop the acquisition. In teh false case presented, the data are sent to the front panel through the graph window but also saved in GetData array. In the True case, the spectrometer is closed without saving any data.

8 Appendix 2

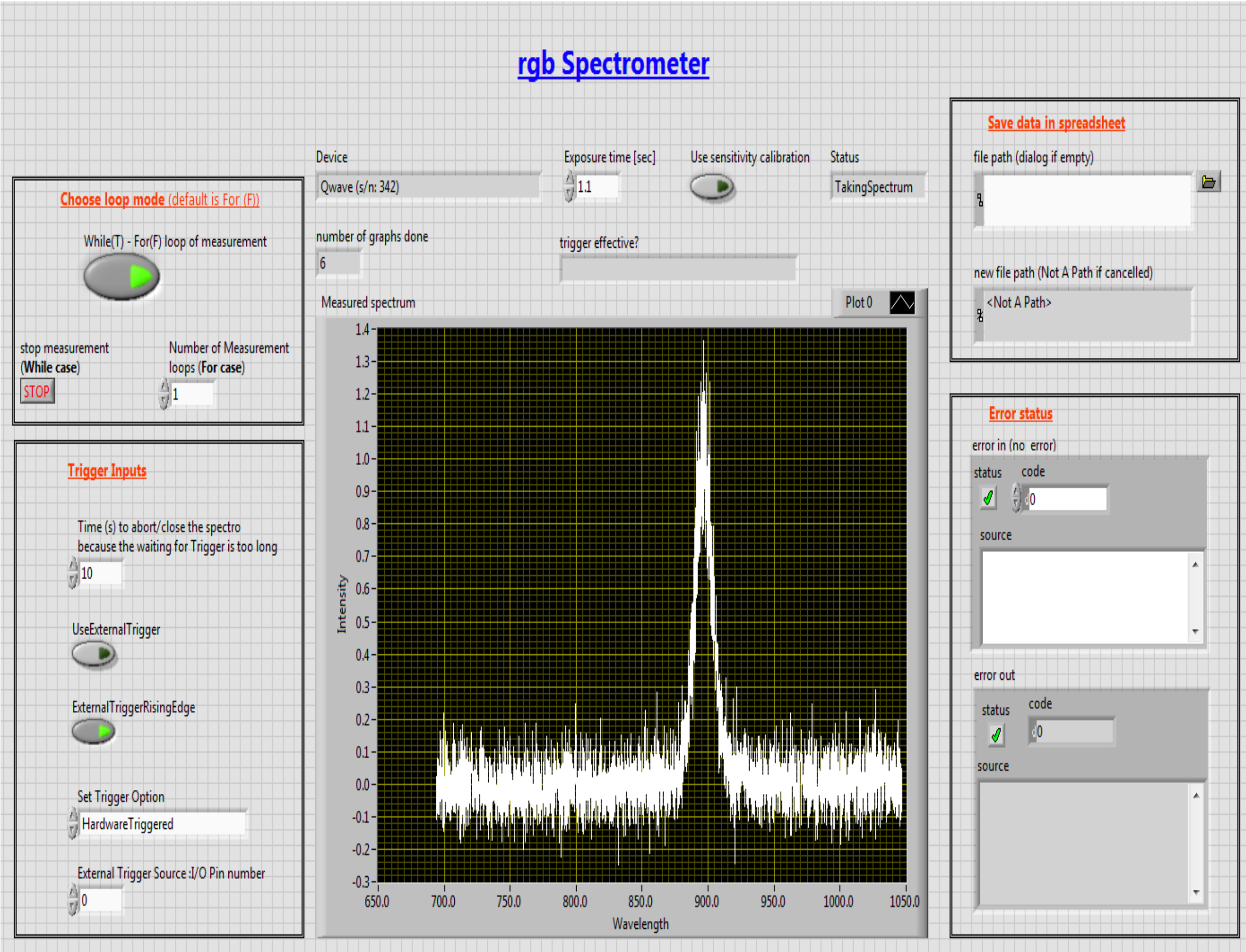

Figure 10: Main front panel of the LabVIEW data acquisition program for the rgb photonics spectrometer. Every control is done from this panel, the loop mode, the Trigger Option, the exposure/integration time and the file path for the saved data. We also have indication on the device used, its current status and if an error occured during the acquisition process. This intensity vs wavelength spectrum is again due to the computer screen radiation.

9 Appendix 3

MATLAB program for Autocorrelation, Experiment 1: give FWHMac, FWHMp, the Gaussian coefficients and the correlation coefficient R^2

```
%data set 10
clear all
close all

%Loading and creating parameters vectors
data100=load('autocorr_dots_data_10spectral_data_usb_4k__0');
data101=load('autocorr_dots_data_10spectral_data_usb_4k__1');
data102=load('autocorr_dots_data_10spectral_data_usb_4k__2');
data103=load('autocorr_dots_data_10spectral_data_usb_4k__3');
data104=load('autocorr_dots_data_10spectral_data_usb_4k__4');
wvlgth=load('autocorr_dots_data_10_0_spectral_values');

sdata=size(data100);
stage_position= data100(:,1);

%3D color image
figure()
imagesc(wvlgth,data100(:,1),data100(:,2:end))
xlabel('wavelength (nm)')
ylabel('stage position (mm)')
title('3D image of intensity vs stage position vs wavelength')

%background (bg) subtraction
%0 for the satge position colomn, average on first data rows
bg_ave=[0,mean(data100(1:100,2:end))];

%create 0 matrices to put the values once the bg is subtracted
subdata90=zeros(400,sdata(1,2));
subdata91=zeros(400,sdata(1,2));
subdata92=zeros(400,sdata(1,2));
subdata93=zeros(400,sdata(1,2));
subdata94=zeros(400,sdata(1,2));

for i=1:400 %sub for substracted
    subdata90(i,:)=data100(i,:)-bg_ave;
    subdata91(i,:)=data101(i,:)-bg_ave;
    subdata92(i,:)=data102(i,:)-bg_ave;
    subdata93(i,:)=data103(i,:)-bg_ave;
```

```
    subdata94(i,:)=data104(i,:)-bg_ave;
end

%Interesting wavelengths range selection
e101=250; %data edge 1 to retrieve the autocorr signal
e102=415;   %data edge 2 to retrieve the autocorr signal
%averaging values per rows for each file: autocorrelation curve creation
s100= mean(subdata90(:,e101:e102),2);
s101= mean(subdata91(:,e101:e102),2);
s102= mean(subdata92(:,e101:e102),2);
s103= mean(subdata93(:,e101:e102),2);
s104= mean(subdata94(:,e101:e102),2);
S10=[s100,s101,s102,s103,s104];

autocorr_curve_10=mean(S10,2); % average of the autocorrelation curves

%Set the delay
i0=find(s100==max(s100)); %find index of the maximum
i1=find(s101==max(s101));
i2=find(s102==max(s102));
i3=find(s103==max(s103));
i4=find(s104==max(s104));
index_max=round((i0+i1+i2+i3+i4)/5) %find the average index of maxima
c=3e+08;
delay=((stage_position-(stage_position(index_max,1)))*1e-03)./c ; % in sec
%put it in fs
delay=delay*1e+15;

% 1 Gaussian fitting process
[f,correl_coeff] = fit(delay,autocorr_curve_10,'gauss1')
gaussian_coef=coeffvalues(f);


%determine FWHM autocorr
FWHMac=gaussian_coef(3)*2*sqrt(log(2));
%determine FWHM pulse
FWHMp=FWHMac/(1.414);
sprintf('The FWHMac = %f fs and FWHMp = %f fs',FWHMac,FWHMp)

%plot autocorr + gausian fit
figure()
plot(f,'r',delay,autocorr_curve_10,'.b')
xlabel('delay in fs')
ylabel('Autocorrelation curve')
```

```
title('Autocorrelation curve averaged over 5 measurements vs delay')

% Find and plot the Half maximum of the experimental data for autocorr
hold on
HM=zeros(400,1)+(max(autocorr_curve_10)/2);
plot(delay,HM,'g')
hold off

%Plots only the  autocorr ( not the fit)
% figure()
% plot(delay,autocorr_curve_10,'b')
% xlabel('delay in fs')
% ylabel('Autocorrelation curve')
% title('Autocorrelation curve averaged over 5 measurements vs delay')
```

10 Appendix 4

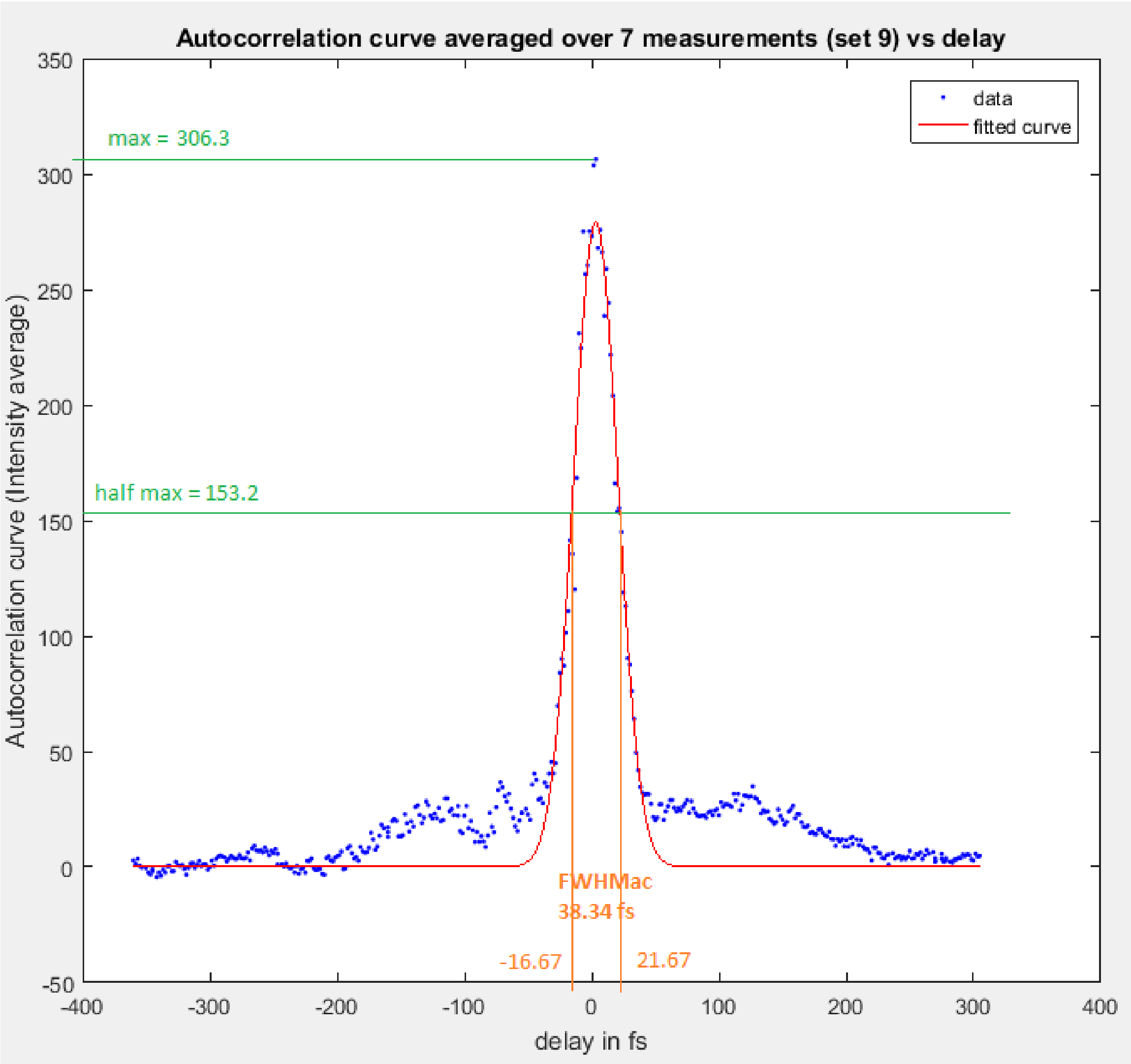

Figure 11: **Gaussian fit for Autocorrelation Experiment 2**

Once again, we can see that the Gaussian fit does not reach the maximum points of the autocorrelation curve, which explains the previous relative error calculated. And the wings problem still remains.