

Université du Québec
INRS Énergie, Matériaux et Télécommunications

**Frequency Resolved
Optical Gating Setup
For Characterizing Amplitude and Phase Of Ultrafast Optical Pulses**

Par
Francisco Manuel Castro Fincheira

Mémoire présenté
pour l'obtention
du grade de Maîtrès en sciences (M.Sc.)
en sciences de l'énergie et des matériaux

Jury d'évaluation

Examineur externe	Denis Morris, Département de physique, Université de Sherbrooke
Examineur interne	François Vidal, INRS-Énergie, Matériaux et Télécommunications
Directeur de recherche	Tsuneyuki Ozaki, INRS-Énergie, Matériaux et Télécommunications

©DROITS RÉSERVÉS À FRANCISCO CASTRO FINCHEIRA, 2013

DEDICATION

Dedicated to my father and mother.

ACKNOWLEDGEMENTS

Thanks to everyone that was involve in my work giving me a serious advice or just encouraging words, to my Advisor Dr. Roberto Morandotti, my Co-Advisor José Azaña, my colleges François Blanchard, Katarzyna A. Rutkowska , Luca Razzari, David Duchesne, Rob Helsten, Yoav Linzon, YongWoo Park, Tommy Payette and the colleagues that were always in the laboratory with me taking care of the setup and helping me, Marco Peccianti and Alessia Pasquazi. To all the group UOP including Jae-Yeol Hwang, Antonio Malacarne, Mostafa Shalaby, Giulia Spina, Mohammad Hossein Asghari, Reza Ashrafi, Marcello Ferrera, Saju Thomas, Ho Sze Phing, Gargi Sharma, Paul-François Ndione, Michael Zaezjev, Chandrasekhar Manda, Tac-Jung Ahn, Shulabh Gupta, and Fangxin Li. Two persons were very important until the end, professors Tsuneyuki Ozaki, and François Vidal that helped me and encourage me to continue improving my thesis and without them I would not be able to finish, to professor Denis Morris that also helped me with the corrections. I also want to thanks all my friends and colleagues in INRS, as Régis Imbeault, Kanwarapal Singh, Jaeho Oh, Romain Dugas, Valeria Felice, Amel Tabet Aoul and many others, I want also thanks to Madame Hélène Sabourin from administration with her kind smile that always made my day.

ABSTRACT

The aim of the Frequency-Resolved Optical Gating (FROG) setup I developed was to be used as performance reference in a research project carried out by the researchers of our group, which we can see the outcomes in the article “Sub-picosecond phase-sensitive optical pulse characterization on a chip”, published in the journal NATURE PHOTONICS LETTERS [50]. In this paper, my colleagues show how they built a device capable of characterizing both the amplitude and phase of ultra-fast optical pulses with the aid of a synchronized incoherently related clock pulse. This device is based on a variation of Spectral Phase Interferometry for Direct Electric-field Reconstruction (SPIDER) that exploits degenerate four-wave mixing in a CMOS-compatible chip. Pulses were measured with a peak power of < 100 mW, a frequency bandwidth of > 1 THz, and up to 100 ps pulse widths, yielding a time bandwidth product of > 100 . My work was to build the FROG setup to be used in that research and as a consequence I will not venture any deeper into the work of my colleagues. In this thesis, I will explain how I built the FROG setup and how it works. I will also show examples of FROG measurements for pulses having propagated in spools of optics fibers of different lengths. FROG is a simple and practical technique for measuring time-dependent intensity, $I(t)$, and phase, $\phi(t)$, of arbitrary ultrashort pulses, i.e. the full complex electric field, $E(t)$, of an individual femtosecond pulse. This technique was developed in the early 90’s by Rick Trebino and Dan Kane. FROG operates on a multi- or single-shot basis and has measured pulses from UV to mid-IR, of many μJ , and from several ps to 9 fs. FROG involves splitting the

pulse to be measured into two replicas with variable relative delay, τ , and crossing them in any nonlinear optical medium. The signal field resulting from the crossing is given by :

$$E_{sig}(t, \tau) \propto E(t) | E(t - \tau) |^2$$

The spectral intensity of this signal (the “FROG trace ”) :

$$I_{frog}(\omega, \tau) \propto \left| \int_{-\infty}^{\infty} E_{sig}(t, \tau) \exp(-i\omega t) dt \right|^2$$

is then measured vs. the delay τ . Then, a numerical algorithm retrieves the pulse intensity and phase vs. time and frequency from this trace.

Francisco Manuel Castro Fincheira

Tsuneyuki Ozaki

ABRÉGÉ

Le but du système FROG (Frequency-Resolved Optical Gating) que j'ai développé était d'être utilisé comme référence dans un projet de recherche mené dans notre groupe, dont nous pouvons voir les résultats dans l'article "Sub-picosecond phase-sensitive optical pulse characterization on a chip", publié dans le journal NATURE PHOTONICS LETTERS [50]. Dans cet article, mes collègues montrent comment ils ont construit un appareil capable de caractériser l'amplitude et la phase d'impulsions optiques ultra brèves au moyen d'une impulsion de synchronisation incohérente. L'appareil est basé sur une variante de interférométrie spectrale de phase pour la reconstruction directe du champ électrique (Spectral Phase Interferometry for Direct Electric Field reconstruction, ou SPIDER) qui exploite le mélange dégéné de quatre ondes dans une puce CMOS-compatible. Ce dispositif a permis de mesurer des impulsions ayant une puissance crête < 100 mW, une bande de fréquence > 1 THz et des durées d'impulsion jusqu'à 100 ps, produisant un produit temps \times largeur de bande > 100 . Mon travail a consisté à construire un système FROG pour cette recherche. En conséquence, je n'irai pas plus profondément dans le travail de mes collègues. Dans ce mémoire, je vais expliquer comment j'ai construit le système FROG et comment il fonctionne. Je montrerai aussi des exemples de mesures FROG pour des impulsions laser après leur propagation dans des bobines de fibre optique de différentes longueurs. Développée au début des années 1990 par Rick Trebino et Dan Kane, la technique FROG permet de mesurer l'intensité, $I(t)$, et la phase, $\phi(t)$, d'impulsions ultra-brèves arbitraires, i.e. le champ électrique complexe, $E(t)$, d'une

impulsion femtoseconde. En dédoublant l'impulsion à mesurer avec un retard relatif variable, τ , puis en croisant les deux impulsions dans un milieu optique nonlinéaire, cette technique, qui utilise des tirs uniques ou multiples, a permis de mesurer des impulsions entre l'UV et le mi-IR, de plusieurs μ J et de plusieurs picoseconde à 9 fs. Le signal résultant du croisement des impulsions est donné par :

$$E_{sig}(t, \tau) \propto E(t) | E(t - \tau) |^2$$

L'intensité spectrale de ce signal (la trace FROG),

$$I_{frog}(\omega, \tau) \propto \left| \int_{-\infty}^{\infty} E_{sig}(t, \tau) \exp(-i\omega t) dt \right|^2$$

est alors mesuré en fonction du délai τ . Enfin, un algorithme numérique itératif permet de déterminer l'amplitude et la phase de l'impulsion en fonction du temps et la fréquence à partir de cette trace.

Francisco Manuel Castro Fincheira

Tsuneyuki Ozaki

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ABRÉGÉ	vi
LIST OF TABLES	xii
LIST OF FIGURES	xiii
1	PRESENTATION	1
1.1	INTRODUCTION TO THE DOMAIN OF ULTRASHORT LASER PULSES AND FROG	1
1.1.1	Ultrashort laser pulse measurement	1
1.1.2	Frequency-Resolved Optical Gating (FROG)	2
1.2	MOTIVATIONS	3
1.3	STRUCTURE OF THE THESIS	4
2	EXPERIMENTAL SETUP	5
2.1	OVERVIEW	5
2.2	COMPONENTS OF THE SETUP AND SOFTWARE	7
2.2.1	1550 nm Picosecond and Femtosecond Fibre Laser	7
2.2.2	Spectrometer ORIEL 1/4 <i>m</i> imaging spectrographs (MS260i)	8
2.2.3	Matlab codes	8
2.2.4	Motorized linear stage	8
3	MATHEMATICAL THEORY	12
3.1	THE TIME AND FREQUENCY-DOMAIN ELECTRIC FIELD	12
3.2	TAYLOR SERIES EXPANSION OF THE PHASES	13
3.3	DISPERSION OF GAUSSIAN PULSES	14

3.4	OBTAINING THE LENGTH OF A DISPERSIVE MEDIUM USING FROG	15
4	EXPERIMENTAL RESULTS	17
4.1	THE MAIN IDEA OF THE EXPERIMENT	17
	4.1.1 The experimental images	17
	4.1.2 The calibrated images	21
	4.1.3 The retrieved images from FROG	23
4.2	EXTRACTING THE PULSE PARAMETERS FROM FROG DATA	26
4.3	COMPARISON BETWEEN THE THREE PULSES RETRIE- VED FROM FROG : SIMPLIFIED REPRESENTATION . . .	32
4.4	OBTAINING THE SPOOL LENGTH	37
5	SUMMARY AND CONCLUSION	39
6	SOMMAIRE EN FRANÇAIS	41
6.1	IMPULSIONS LASER ULTRA BRÈVES	41
6.2	SYSTÈME FROG	42
	6.2.1 Spectrographe de formation d'images ORIEL 1/4m	43
	6.2.2 Les programmes	43
	6.2.3 Platine linéaire motorisée	44
6.3	MESURE DES IMPUSIONS OPTIQUES ULTRA BRÈVES . . .	44
6.4	FREQUENCY-RESOLVED OPTICAL GATING (FROG) . . .	45
6.5	MOTIVATIONS	46
6.6	STRUCTURE DU MÉMOIRE	47
6.7	THÉORIE MATHÉMATIQUE	47
	6.7.1 Séries de Taylor de la phase	48
	6.7.2 Impulsions gaussiennes	49
6.8	L'IDÉE PRINCIPALE DE L'EXPÉRIENCE	50
6.9	MÉTHODOLOGIE	51
	6.9.1 Traitement des spectres expérimentaux	51
	6.9.2 Les images expérimentales	51
	6.9.3 Les images résultant de l'analyse FROG	52
6.10	COMPARAISON DES RÉSULTATS POUR LES TROIS BOBINES	52
	6.10.1 Analyse FROG	52
	6.10.2 Calcul des paramètres	52
6.11	CONCLUSION	53

Bibliographie	55
Appendices	60
A MATLAB SCRIPT FOR FROG IMAGE ACQUISITION, USING LI- NEAR STAGE AND ORIEL SPECTROMETER SYNCHRONIZED	61
B MATLAB PROGRAM FOR CONTINUOUS SPECTRUM ACQUI- SION FROM THE CAMERA	78
C MATLAB PROGRAM FOR BACKGROUND ACQUISITION	83
D MATLAB PROGRAM FOR CALIBRATION OF THE SPECTROMETER	87
E MATLAB PROGRAM FOR CUT AND INTERPOLATE FROG EX- PERIMENTAL IMAGE	90
F BENCHTOP DC SERVO MOTOR CONTROLLERS	98
G OPTICAL SPECTRUM ANALYZER <i>AQ6317B</i>	101
H OPTICAL FIBER AMPLIFIERS	104
I PICOSECOND and FEMTOSECOND FIBER LASERS	106
J The Matlab program FROG 1.0	108
J.1 INSTALLATION	109
J.2 Parameters Settings	109
J.2.1 Type of Experiment	109
J.2.2 Grid Size	114
J.2.3 Center Frequency	116
J.2.4 Initial Guess of the field	116
J.2.5 Algorithm Sequence	118
J.3 FROG Algorithms	119
J.3.1 Basic "Vanilla" FROG	119
J.3.2 Overcorrection	120
J.3.3 Generalized Projection	120
J.4 Retrieval Process	120
J.4.1 Trace Windows	120
J.4.2 Amplitude and Phase Windows	120
J.4.3 Control Panel	122

	J.4.4 Results Panel	126
	J.4.5 Root Mean Squared (<i>RMS</i>) Error	126
J.5	Default Configuration File	128
K	The FROG Algorithm	129
	K.1 The Iterative-Fourier-Transform Algorithm Generalized projection	129
L	Oriel <i>MS260iTM</i> 1/4 m Imaging Spectrograph	133

LIST OF TABLES

<u>Table</u>	<u>page</u>
4-1 Interpolation coefficients for the calibration of the experimental CCD camera images in figures 4-1, 4-2 and 4-3. Coefficients d'interpolation pour la calibration des figures 4-1, 4-2 et 4-3.	21
4-2 Parameters for no spool. Paramètres obtenus sans bobine.	27
4-3 Parameters for the 10 m spool. Paramètres obtenus pour la bobine de 10 m.	32
4-4 Parameters for the 100 m spool. Paramètres obtenu pour la fibre de 100 m.	37
4-5 Physical parameters extracted from the fits. Paramètres physiques extraits des fits.	37
4-6 Calculated spool lengths. Longueur des bobines calculée.	37
5-1 Results.	39
6-1 Resultats	54

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 Experimental setup	7
2-2 Spectrometer	9
2-3 Inside spectrometer	10
2-4 Linear stage	11
4-1 FROG trace for no spool	18
4-2 FROG trace from 10 m spool	19
4-3 FROG trace from 100 m spool	20
4-4 Calibrated experimental image for no spool.	23
4-5 Calibrated experimental image for the 10 m spool.	24
4-6 Calibrated experimental image for the 100 m spool.	25
4-7 FROG retrieval for no spool	28
4-8 FROG retrieval for the 10 m spool	29
4-9 FROG retrieval for the 100 m spool	30
4-10 Amplitudes and phases in the time(delay) and frequency domains . .	31
4-11 FROG characterization and fit for no spool	33
4-12 FROG characterization and fit for the 10 m spool	34
4-13 FROG characterization and fit for the 100 m spool	35
4-14 Simplified amplitudes and phases in the time and frequency domains .	36
5-1 Real length vs. calculated length	40

F-1	Benchtop DC Servo Motor Controllers	99
F-2	Specifications for Benchtop DC Servo Motor Controllers	100
G-1	Spectrum Analyzer	102
G-2	Specifications for Spectrum Analyzer	103
H-1	Fiber Amplifier	105
H-2	Specifications for Fiber Amplifier	105
I-1	Fiber Laser	107
I-2	Specifications for Fiber Laser	107
J-1	View of the setting panel when the user starts the program	110
J-2	Type of Experiment	111
J-3	Theory Parameter Panel	112
J-4	Theoretical Shape Selection	113
J-5	Experiment Parameter Panel	115
J-6	Grid Size of the Trace	116
J-7	Initial Guess of the Field	117
J-8	Algorithm Sequence Panel	118
J-9	Retrieval process window	121
J-10	SHG FROG traces for linearly chirped pulses.	123
J-11	Control Panel	124
J-12	Result Panel	127
K-1	Schematic of a generic FROG algorithm	130
K-2	Solution space view of generalized projection.	131
L-1	Imaging Spectrometer	134

L-2 Specifications for Imaging Spectrometer 134

Chapitre 1 PRESENTATION

1.1 INTRODUCTION TO THE DOMAIN OF ULTRASHORT LASER PULSES AND FROG

1.1.1 Ultrashort laser pulse measurement

The first measurements of laser pulses were carried out in the 1960s using electronic detectors. The goal is to measure the pulse electric field vs. time, i.e. its intensity, $I(t)$, and phase, $\phi(t)$, given by :

$$E(t) = \sqrt{I(t)} \exp[i(\omega_0 t - \phi(t))] \quad (1.1)$$

The complex electric field in the frequency domain is obtained from the Fourier transform of $E(t)$, given by :

$$\tilde{E}(\omega) = \sqrt{S(\omega)} \exp[-i\varphi(\omega)] \quad (1.2)$$

where $S(\omega)$ is the pulse spectral intensity, and $\varphi(\omega)$ is the spectral phase.

In a second harmonic generation (SHG) crystal we get twice the frequency of the input light. When the two pulses overlap in time (see the ABSTRACT), the SHG crystal generates a signal with energy vs. delay given by [1]

$$\int_{-\infty}^{\infty} I(t)I(t - \tau)dt \quad (1.3)$$

where τ is the relative delay between the two pulses. The problem with this technique (autocorrelation) is that it is necessary to assume a pulse shape in order to obtain a pulse length. This problem is called the one-dimensional phase-retrieval problem¹ [2,3]. Moreover, we cannot obtain information about the phase $\phi(t)$. Even if we were able to identify all the possible pulses corresponding to a given trace, several of them would fit the measurements and we could not determine which one is correct. [4-9,46]

1.1.2 Frequency-Resolved Optical Gating (FROG)

FROG is an autocorrelation technique, but the pulse is spectrally resolved as a function of the delay, τ , between two pulses. From the spectrally-resolved measurement, we can use a phase-retrieval algorithm to retrieve the precise pulse intensity and phase vs. time. [10-12] The measured quantity required by the FROG technique is :

$$I_{FROG}(\omega, \tau) = \left| \int_{-\infty}^{\infty} E_{sig}(t, \tau) e^{-i\omega t} dt \right|^2 \quad (1.4)$$

where the signal field,

$$E_{sig}(t, \tau) = E(t)E_{gate}(t - \tau) \quad (1.5)$$

is a function of time and the delay. In FROG, the gate function, $E_{gate}(t - \tau)$, is a function of the unknown input pulse, $E(t)$, that we are trying to measure. When using SHG as the nonlinear-optical process, $E_{gate}(t - \tau) = E(t)$. However, in FROG, $E_{gate}(t - \tau)$ can be any known function acting as the reference pulse. In principle, $E_{sig}(t, \tau)$ can be any function of time and delay that contains enough information

1. Trebino, R., Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses. 2002, Boston : Kluwer Academic Publishers.

to determine the pulse characteristics. [13] As mentioned earlier (see ABSTRACT) pulse retrieval using FROG requires phase-retrieval numerical algorithms. [14-21] (See also <http://www.physics.gatech.edu/gcuo/subIndex.html>)

More information about early works on FROG can be found in Refs. [22-39] and on the web page <http://frog.gatech.edu/publications.html>.

It is worth mentioning that there are some opinions that FROG has nontrivial ambiguities, and thus cannot really retrieve unambiguously the amplitude and phase of the pulse. However, this is actually not the case. [40-42] In addition, FROG is more sensitive than autocorrelation to pulse variations. [43-49]

1.2 MOTIVATIONS

The interest of the researchers of our group was to characterize the pulses used in different experiments that were being conducted at this time. For that purpose, they needed to retrieve the precise amplitude and phase in function of time and frequency of the laser pulses used in their experiments.

The members of our group reported a device capable of characterizing both the amplitude and phase of ultrafast optical pulses with the aid of a synchronized incoherently related clock pulse. It is based on a novel variation of Spectral Phase Interferometry for Direct Electric field Reconstruction (SPIDER) technique that exploits degenerate four-wave mixing in a CMOS-compatible chip. It measures pulses with a peak power of < 100 mW, a frequency bandwidth of > 1 THz and up to 100 ps pulse widths, yielding a time \times bandwidth product of > 100 . More details are given in Ref. [50]. In this project, my work consisted in developnig a FROG

system in order to compare the pulse characteristics provided by the new device with those provided by the more conventional FROG technique.

1.3 STRUCTURE OF THE THESIS

In Chapter 1, we present the subject and the structure of the document, making a brief introduction to the domain of ultrashort laser pulses and FROG. We also explained the motivation of the subject in the framework of the research done in our group. In Chapter 2, we discuss the experimental setup, the material we used, provide information about the spectrometry, the program we used to obtain the images, and the motorized linear stage. In Chapter 3, we provide the mathematical bases of pulse characterization. In Chapter 4, we discuss the experimental results. We performed FROG for no spool, a 10 m spool and a 100 m spool. In Chapter 5, we conclude the thesis.

Chapitre 2 EXPERIMENTAL SETUP

2.1 OVERVIEW

The FROG experimental setup is based on an autocorrelator system, and as is always the case in autocorrelation, we have to use the pulse to measure the pulse itself. We must gate the pulse with itself, and, in order to make a spectrogram of the pulse, we have to spectrally resolve the gated piece of the pulse. In its simplest form, FROG is an autocorrelation-type measurement in which the autocorrelator signal beam is spectrally resolved. Instead of measuring the autocorrelator signal energy vs. delay Eq.(1.3), which yields an autocorrelation, we measure the signal spectrum vs. delay Eq.(1.4).

The laser beam comes from a “picosecond and femtosecond fiber laser”. In our case, we launched a beam with a wavelength of 1550 nm. We amplified the beam with an optical fibre amplifier, Pritel model HPFA-15 (see Figure H-1 and Figure H-2 in appendix H). After amplifying the beam intensity, we analyze the spectrum of the primary 1550 nm beam using a fibre spectrometer, model OSA AQ6317B (see Figure G-1 and Figure G-2 in appendix G). This is for checking the quality of the test beam. To test the FROG setup, right after the spectrometer we place a spool of optical fibre, and subsequently compare the real length of the spool with the length given by the FROG setup.

In a normal FROG experiment, the primary beam is spliced by a beam splitter. One of the spliced beams goes to a THORLABS moving stage, which is a variable optical delay line stage, and we produce a variable time delay with respect to the other beam $E(t - \tau)$. In order to perform FROG pulse characterization, we need a collection of different measurements with different delays. I will explain why later.

After that, both beams are redirected to the second harmonic generator (SHG) crystal βBaBO_4 (BBO), i.e. the one that passed through the moving stage, $E(t - \tau)$, and the other that merely passed through the beam splitter, $E(t)$. The two spliced beams have to interact inside an SHG crystal, but with a specific angle of 39.2° with respect to the normal to the surface to produce a 775 nm beam. Before the beams reach the crystal, we focus both beams in an optical fiber. The beams, which are now in the optical fibres, are easy to direct at the angle we want on the crystal.

In the SHG crystal, both beams interact and a new beam of double frequency is created. This means that the output wavelength is half of the input wavelength. In our case, the input wavelength is 1550 nm and it comes out at 775 nm. There are actually 3 outputs beams : two of 1550 nm at an angle of 39.2° and one of 775 nm at an angle of 0° , which we will call $E_{sig}(t, \tau)$. We have to send this 775 nm beam directly to the NEWPORT 1/4m Imaging Spectrographs model MS260i to obtain the spectrum. We then get a spectrum of intensity vs. wavelength that we then change to intensity vs. frequency (Eq.(1.4)).

This is repeated for different delays, τ , by changing the position in the THORLABS moving stage, using a MATLAB program that connects the stage with the computer. From this collection of spectra obtained from different delays, we obtain

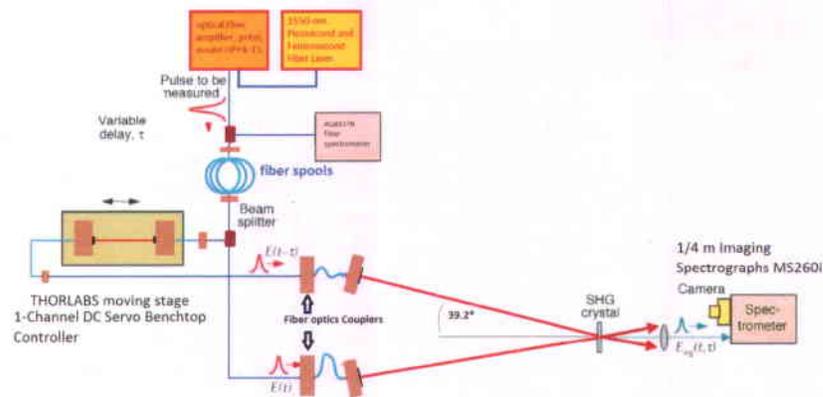


FIGURE 2-1 – The primary beam is spliced in two beams. One goes to a THORLABS moving stage, $E(t - \tau)$. $E(t)$ and $E(t - \tau)$ go to a SHG crystal, βBaBO_4 (BBO). They interact in a SHG crystal, with an angle of 39.2° . The input is 1550 nm and there are 3 outputs beams : two of 1550 nm at an angle of 39.2° and one of 775 nm at an angle of 0° , $E_{sig}(t, \tau)$ that goes to the NEWPORT 1/4m Imaging Spectrographs to obtain the spectrum. Le faisceau primaire est divisé. L'un des faisceaux divisés va vers la platine mobile de THORLABS, puis au cristal SHG. L'autre faisceau divisé va au cristal de SHG. Les faisceaux doivent interagir à l'intérieur du cristal de SHG avec un angle de 39.2 degrés pour produire un faisceau de 775 nm, ce faisceau est par la suite directement envoyé au spectrographe de formation d'image, ORIEL 1/4m (MS260i) pour en obtenir le spectre.

a plot of the intensity vs. frequency and delay (spectrum vs delay), and we get what we call a FROG image. The whole experimental setup is shown in Figure 2-1.

2.2 COMPONENTS OF THE SETUP AND SOFTWARE

2.2.1 1550 nm Picosecond and Femtosecond Fibre Laser

The PriTels FFL 1550 nm Picosecond and Femtosecond Fibre Lasers (Figure I-1) has a repetition rate of 10-100 MHz while the fixed average output power varies

with pulse width and pulse repetition rate (e.g., > 4 mW at 2 ps and 20 MHz). More detail are given in Appendix I (Figure I-2).

2.2.2 Spectrometer ORIEL 1/4m imaging spectrographs (MS260i)

The MS260i 1/4m Imaging Spectrograph is a multi-grating instrument. This dual output model uses dual grating mounts (Figures L-1 and L-2). It is an F/3.9 instrument with a flat 28 mm image plane for multichannel detectors. Stray light is negligible, re-entrant spectra are eliminated, throughput is high, and it has great resolution spatially and spectrally. A set of computer-optimized toroidal mirrors produce accurate images of the input slit in the flat output plane. The aberration limited spatial resolution is $40 \mu\text{m}$ (FWHM) for the dual grating instrument. It has a USB interface (see Figures 2-2 and 2-3).

2.2.3 Matlab codes

We developed a Matlab code for communication between the CCD camera of the spectrometer and the computer. This code is given in Appendices A, B, and C. The Matlab code for calibrating the Camera is given in Appendix D. After obtaining the FROG image we use a code to manipulate the image and adapt it to the Matlab program FROG 1.0 which executes the retrieval numerical algorithm. In Appendix B, we also present the Matlab code for continuous image acquisition from the camera.

2.2.4 Motorized linear stage

The Linear Stage is controlled also by the BENCHTOP DC SERVO MOTOR CONTROLLER shown in appendix F (see Figures F-1, F-2, and 2-4).

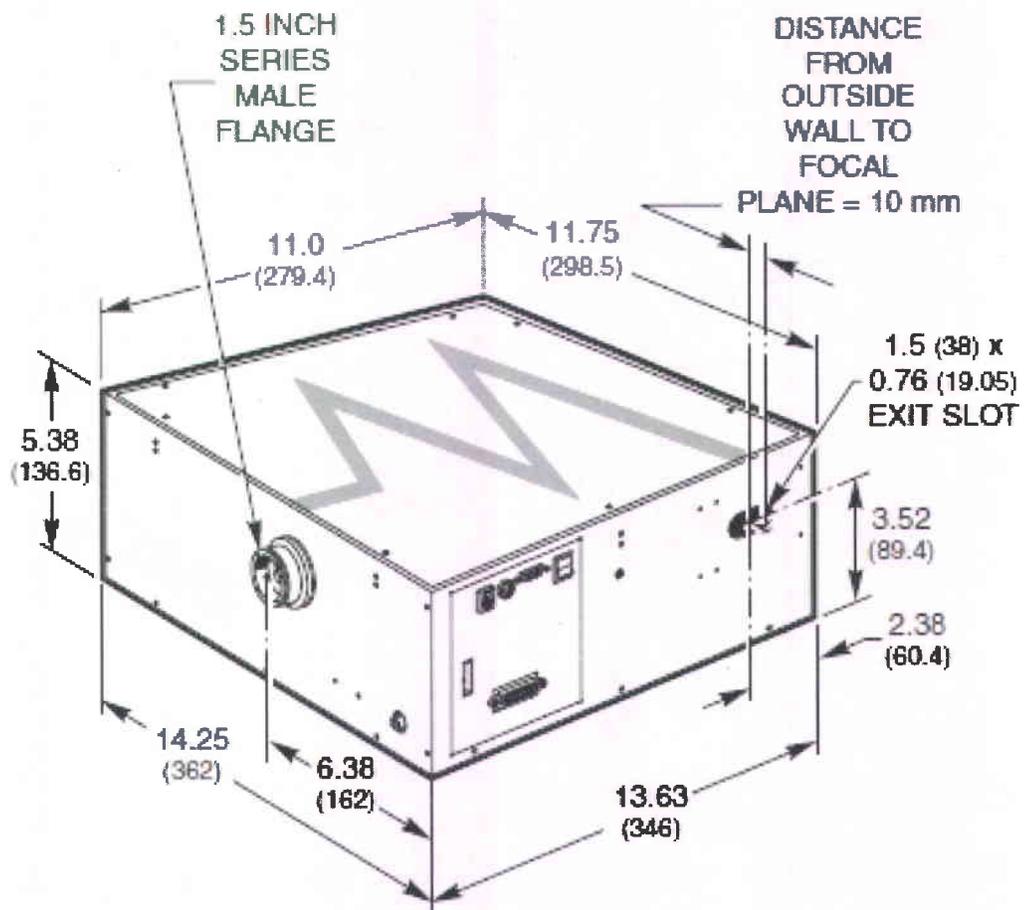


FIGURE 2-2 – Spectrometer ORIEL 1/4m imaging spectrographs (MS260i). Spectromètre

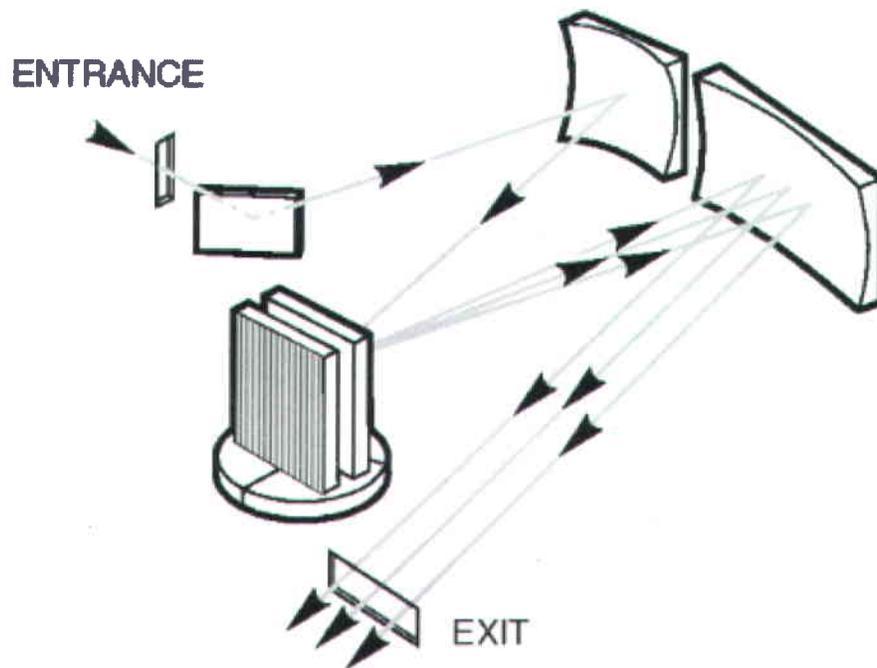


FIGURE 2-3 – A set of computer-optimized toroidal mirrors produce accurate images of the input slit in the flat output plane. The aberration limited spatial resolution is $40 \mu\text{m}$ (FWHM) for the dual grating instrument.. Intérieur du spectromètre



FIGURE 2-4 – The Linear Stage is controlled also by the BENCHTOP DC SERVO MOTOR CONTROLLER shown in appendix F . Platine linéaire.

Chapitre 3 MATHEMATICAL THEORY

3.1 THE TIME AND FREQUENCY-DOMAIN ELECTRIC FIELD

An ultrashort laser pulse is characterized by an amplitude or intensity and a phase vs. time. Neglecting the spatial dependence for the moment, the pulse electric field is given by :

$$\varepsilon(t) = \frac{1}{2} \sqrt{I(t)} \exp[i(\omega_0 t - \phi(t))] + c.c. \quad (3.1)$$

where $I(t)$ is the intensity, ω_0 is the central carrier frequency, $\phi(t)$ is the phase, and c.c. stands for the complex conjugate.

Fourier-transforming the pulse electric field given by Eq. (3.1) yields :

$$\tilde{\varepsilon}(\omega) = \frac{1}{2} \sqrt{S(\omega - \omega_0)} \exp[-i\varphi(\omega - \omega_0)] + \frac{1}{2} \sqrt{S(-\omega - \omega_0)} \exp[+i\varphi(-\omega - \omega_0)] \quad (3.2)$$

where $S(\omega)$ and $\varphi(\omega)$ are the frequency-domain equivalents of the intensity and phase in the time domain, respectively. $S(\omega)$ is called the spectral intensity or just the “spectrum”. One observes that the frequency-domain electric field $S(\omega)$ and phase $\varphi(\omega)$ have positive- and negative-frequency components. Actually the negative-frequency component $S(-\omega - \omega_0)$ contains the same information as the positive-frequency component $S(\omega - \omega_0)$. For that reason, it is generally more convenient to work with the complex field $E(t)$ instead of the real field, Eq. (3.1) :

$$E(t) = \sqrt{I(t)} \exp[i(\omega_0 t - \phi(t))] \quad (3.3)$$

In the frequency-domain, this becomes :

$$\tilde{E}(\omega) = \sqrt{S(\omega - \omega_0)} \exp[-i\varphi(\omega - \omega_0)] \quad (3.4)$$

3.2 TAYLOR SERIES EXPANSION OF THE PHASES

The pulse's variations in frequency are contained in the phase $\phi(t)$ or $\varphi(\omega)$. The frequency or color of the pulse at time t is :

$$\Omega(t) = -d\phi(t)/dt \quad (3.5)$$

The variation of the frequency with time is called "chirp". Positive chirp is an increasing frequency with time, and negative chirp is a decreasing frequency with time. But more complex chirps are also quite common. Indeed, understanding the complicated chirp present in ultrashort pulses plays an important role in modelocked lasers and is the key to making even shorter pulses.

We can write a Taylor series for the phase $\phi(t)$ around the time $t = 0$ as :

$$\phi(t) = \phi_0 + \phi_1 \frac{t}{1!} + \phi_2 \frac{t^2}{2!} + \dots \quad (3.6)$$

where :

$$\phi_n = \left. \frac{d^n \phi}{dt^n} \right|_{t=0} \quad (3.7)$$

Substituting this series expansion in Eq. (3.1) or Eq. (3.3), we see that the first term ϕ_0 , is just an overall phase without importance, while the second term ϕ_1 induces a shift in the central frequency ω_0 , which is assumed to be known. Thus these two first terms can be set to zero. Often, as in this work, only the next term

ϕ_2 is typically required to describe well-behaved pulses. Of course, for badly behaved pulses it is necessary to take into account the higher-order terms as well.

A similar Taylor series expansion can be performed for $\varphi(\omega)$ around $\omega = \omega_0$:

$$\varphi(\omega - \omega_0) = \varphi_0 + \varphi_1 \frac{\omega - \omega_0}{1!} + \varphi_2 \frac{(\omega - \omega_0)^2}{2!} + \dots \quad (3.8)$$

where :

$$\varphi_n = \left. \frac{d^n \varphi}{d\omega^n} \right|_{\omega=\omega_0} \quad (3.9)$$

Substituting this series expansion in Eq. (3.2) or Eq. (3.4), we see that the first term is only an overall phase. The second term only induces a time delay in the pulse, and is called group delay for this reason. Both terms can thus be set to zero. The third and following terms describe dispersion in the medium. The coefficient of the third term, ϕ_2 , is called is called the group-delay dispersion (GDD).

3.3 DISPERSION OF GAUSSIAN PULSES

A good first approximation for most laser pulses is the Gaussian pulse, given by :

$$\sqrt{I(t)} = E_0 \exp[-(t/\Delta\tau)^2] \quad (3.10)$$

where $\Delta\tau$ is the pulse duration and E_0 is the maximum pulse amplitude.

Substituting Eq.(3.10) and $\phi(t) = \phi_2 t^2/2$ (see Eq.(3.6)) in Eq.(3.3), we obtain :

$$E(t) = E_0 e^{-\frac{t^2}{\Delta\tau^2}} e^{i(\frac{\phi_2}{2} t^2)} e^{i\omega_0 t} \quad (3.11)$$

Fourier-transforming this pulse electric field we get :

$$\tilde{E}(\omega) = \tilde{E}_0 e^{-\frac{(\omega-\omega_0)^2}{\Delta\omega^2}} e^{-i(\frac{\varphi_2}{2}(\omega-\omega_0)^2)} \quad (3.12)$$

where \tilde{E}_0 is the maximum amplitude,

$$\Delta\omega^2 = \frac{\varphi_2^2 \Delta\tau^4 + 4}{\Delta\tau^2} \quad (3.13)$$

is the square of the pulse width in the frequency domain, and φ_2 is just the GDD (see Eq.(3.9)). We note that the spectral amplitude is given by :

$$\sqrt{S(\omega)} = \tilde{E}_0 e^{-\frac{(\omega-\omega_0)^2}{\Delta\omega^2}} \quad (3.14)$$

It is easy to show that :

$$\varphi_2 = \sqrt{\frac{\Delta\tau^2 \Delta\omega^2 - 4}{\Delta\omega^4}} \quad (3.15)$$

and, from Eqs. (3.13) and (3.15), that :

$$\phi_2 = -\frac{|\varphi_2| \Delta\omega^4}{4 + \varphi_2^2 \Delta\omega^4} \quad (3.16)$$

3.4 OBTAINING THE LENGTH OF A DISPERSIVE MEDIUM USING FROG

In order to check our FROG setup and software, we compared the actual length of an optical fiber, L , with the length calculated using the FROG measurements for the pulse parameters. This can be done through the simple relation :

$$L = \varphi_2 / b_2 \quad (3.17)$$

The FROG setup was used to measure the pulse width in time, $\Delta\tau$ (pulse duration), and in the frequency domain, $\Delta\omega$, to obtain the group delay dispersion, φ_2 , through Eq. (3.13). In Eq.(3.17), b_2 characterizes dispersion in a given optical material near the central frequency ω_0 and wavelength, λ_0 , of the laser pulse :

$$b_2 = \left. \frac{d^2k(\omega)}{d\omega^2} \right|_{\omega=\omega_0} \quad (3.18)$$

where $k = (2\pi/\lambda_0)n(\omega)$ is the wave number of the laser, $n(\omega)$ being the index of refraction of the material. For the optical fibers we used, the experimental value is approximately $b_2 = 22 \text{ ps}^2/\text{km}^1$ for 1550 nm. The dispersion of a medium is a consequence of the frequency dependence of the group velocity of the wave in the material. Indeed, since the group velocity is $v_g(\omega) = d\omega/dk$, we see that $b_2 = d(v_g^{-1})/d\omega$.

1. www.rp-photonics.com/chromatic-dispersion.html

Chapitre 4 EXPERIMENTAL RESULTS

4.1 THE MAIN IDEA OF THE EXPERIMENT

The idea is to test FROG setup using three different spools of optical fibers with different lengths, namely : $L = 0$ (no spool), 10 m and 100 m. The software FROG 1.0 was created in Matlab to retrieve the amplitude and phase of a short laser pulse from a measured FROG trace $I_{FROG}(\omega, \tau)$ using an algorithm which is explained in appendix K. The FROG trace can be considered as a matrix for discrete values of ω and τ . In order to use the retrieval algorithm the measured matrix must be set as a square matrix of dimension $N = 2^{ord}$. In this work we use $ord = 8$.

The spectrum of each spool is collected with a CCD camera with 1024 pixels in frequency, ω , and 100 pixels in delay, τ . We thus need to interpolate the measured image to respect the calibration of the frequency and time intervals. With the defined dimension of the matrix ($N = 2^{ord} = 256$), the temporal window (time window) and the spectral window, we use a Matlab program to properly resize and interpolate the experimental matrix (i.e. the two-dimensional image). We can see the Matlab program that performs this operation in the appendix E.

4.1.1 The experimental images

We can see the spectra collected with the CCD camera in Figures 4-1, 4-2 and 4-3 for the spool lengths $L = 0, 10, \text{ and } 100$ m, respectively. In those images,

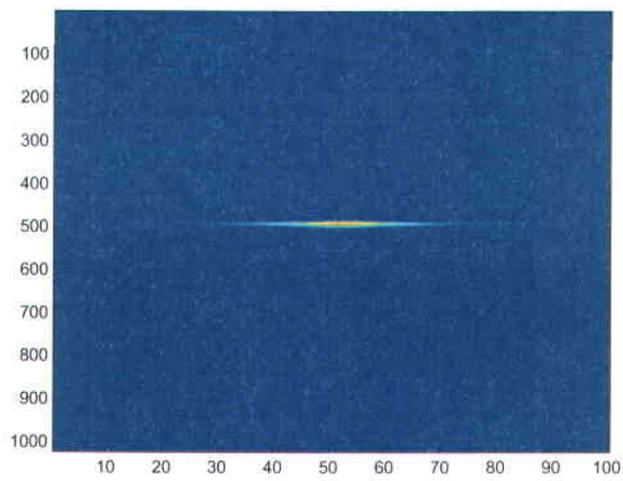


FIGURE 4-1 – FROG trace for no spool. The spectrum of each spool is collected with a CCD camera with 1024 pixels in frequency, ω , and 100 pixels in delay, τ . Trace FROG sans bobine.

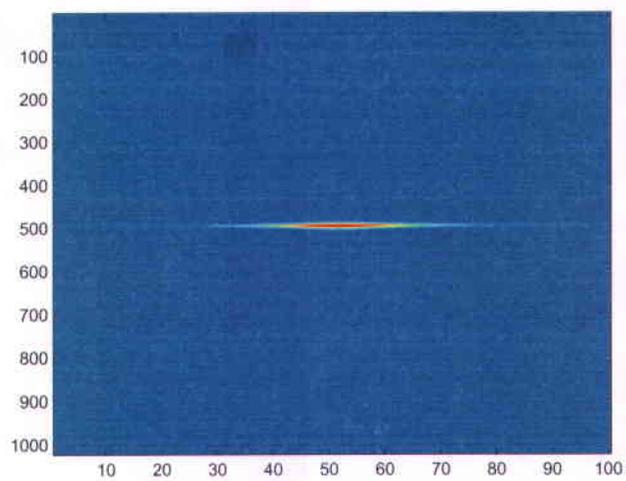


FIGURE 4-2 – FROG trace from 10 m spool. The spectrum of each spool is collected with a CCD camera with 1024 pixels in frequency, ω , and 100 pixels in delay, τ . Trace FROG de la bobine de 10 m.

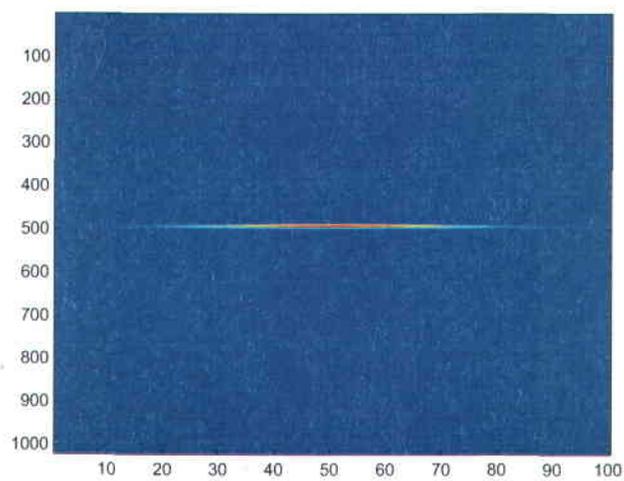


FIGURE 4-3 – FROG trace from 100 m spool. The spectrum of each spool is collected with a CCD camera with 1024 pixels in frequency, ω , and 100 pixels in delay, τ . Trace FROG de la bobine de 100 m.

the vertical axis represents the frequency or wavelength while the horizontal axis represents the delay τ .

4.1.2 The calibrated images

The experimental CCD camera images in figures 4-1, 4-2 and 4-3 consist in 1024×100 pixels. The pixels on the frequency, $\nu = \omega/2\pi$, and equivalent wavelength, $\lambda = c/\nu$, axis were labelled on the raw experimental images using the expressions :

$$\nu_i = zero_f + cal_f \times i \quad (4.1)$$

$$\lambda_i = zero_l + cal_l \times i \quad (4.2)$$

where i ($1 < i < 1024$) is the number of the pixel along the frequency or wavelength axis. The numerical values of the coefficients are given in Table 4-1. We note that the λ interval is centered near 775 nm since we are measuring the second harmonic of the 1550 nm laser. For a better management of data we fix zero at 775 nm or 387 THz as we can see in the figures 4-4, 4-5 and 4-6.

TABLE 4-1 – Interpolation coefficients for the calibration of the experimental CCD camera images in figures 4-1, 4-2 and 4-3. Coefficients d'interpolation pour la calibration des figures 4-1, 4-2 et 4-3.

Interpolation coefficients.		
$cal_f =$	-0.07779×10^{12}	frequency/pixel
$zero_f =$	425.6×10^{12}	frequency at pixel 1
$cal_l =$	0.1555×10^{-9}	wavelength/pixel
$zero_l =$	697.4×10^{-9}	wavelength at pixel 1

In figures 4-4, 4-5 and 4-6, the DELAY we get is from the linear stage, we can change the time that the laser beam hit the crystal changing the linear stage position. For example, a beam travelling free with a speed of light “c”, 3cm take a

time of $100ps$, so changing the position of output beam fiber in the linear stage we can change the time the beam reach the input beam fiber, so the DELAY. In the figures 4-4, 4-5 and 4-6, the DELAY is in seconds. The range of the DELAY axis is approx $3 \times 10^{-11}s$ it means $1cm$ of distance between the input and output optical fiber over the linear stage. We center the DELAY axis in a half of the time range so the range goes from $-1.5 \times 10^{-11}s$ to $1.5 \times 10^{-11}s$.

The experimental images in which the pixel numbers are replaced by physical parameters *frequency* and *delay* are shown in figure 4-4, 4-5 and 4-6 for the spool lengths $L = 0, 10$ and 100 m, respectively. By comparison with Figures 4-1, 4-2 and 4-3, we note that these new figures have been resized in order to zoom in the most relevant features of the images, keeping the calibration settings in the axis of frequency and delay.

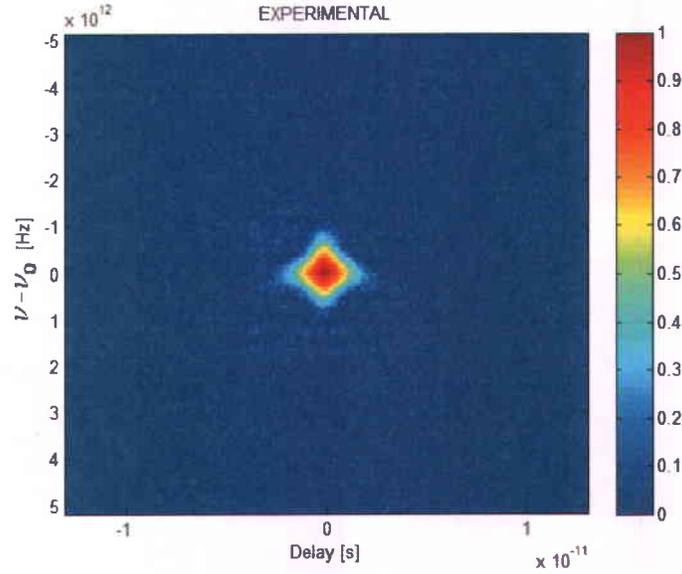


FIGURE 4-4 – Calibrated experimental image for no spool. By comparison with figures 4-1, 4-2 and 4-3, these new figures have been resized in order to zoom in the most relevant features of the images keeping, the calibration settings of the delay window and the spectral window centered at $\nu_o = 387$ THz. With a defined dimension of the matrix ($N = 2^{ord} = 256$). Image expérimentale calibrée sans bobine.

4.1.3 The retrieved images from FROG

Figures 4-7, 4-8 and 4-9 show the results of the FROG analysis for the $L = 0$, 10 and 100 m spool, respectively. The evolution of the error as a function the number of iterations of the numerical algorithm is shown at the bottom of the group of figures. One observes that few tens of iterations are required to minimize the error. The retrieved and experimental FROG traces are shown on the left of the top and middle panels, respectively. The pulse's amplitudes $\sqrt{I(t)}$ and $\sqrt{S(\nu)}$ are shown on the top panel while the phases $\phi(t)$ and $\varphi(\nu)$ are shown on the middle panel. The green curves are for iteration 1 while the other curves are

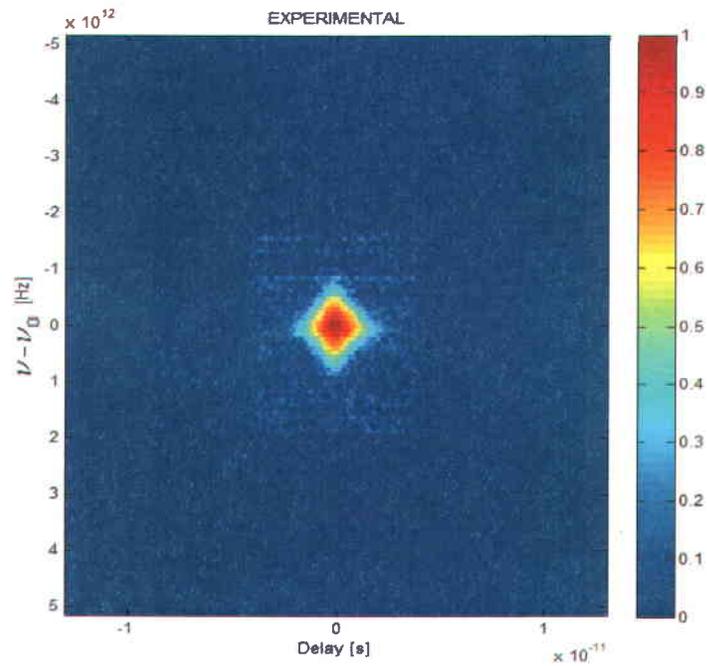


FIGURE 4-5 – Calibrated experimental image for the 10 m spool. By comparison with figures 4-1, 4-2 and 4-3, these new figures have been resized in order to zoom in the most relevant features of the images keeping, the calibration settings of the delay window and the spectral window centered at $\nu_o = 387$ THz. With a defined dimension of the matrix ($N = 2^{ord} = 256$). Image expérimentale calibrée pour la bobine de 10 m.

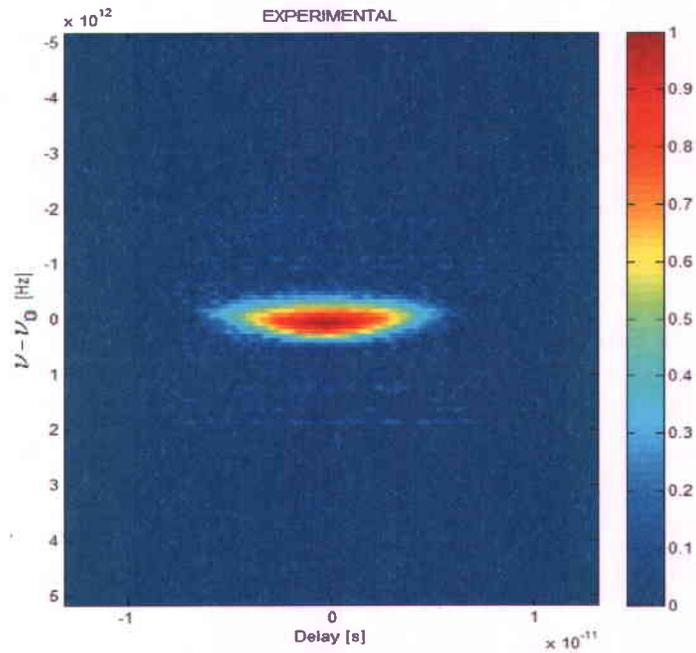


FIGURE 4-6 – Calibrated experimental image for the 100 m spool. By comparison with figures 4-1, 4-2 and 4-3, these new figures have been resized in order to zoom in the most relevant features of the images, keeping the calibration settings of the delay window and the spectral window centered at $\nu_0 = 387$ THz. With a defined dimension of the matrix ($N = 2^{ord} = 256$). Image expérimentale calibrée pour la bobine de 100 m.

the final results after 100 iterations. We note that the phases are defined up to a constant so that the global shifts are not significant. The program FROG we used is displayed in appendix J and the FROG algorithm is explained in appendix K.

4.2 EXTRACTING THE PULSE PARAMETERS FROM FROG DATA

Figures 4-7, 4-8, and 4-9 show the retrieved characterizations from the FROG algorithm for the three spool lengths. Figure 4-10 shows a comparison between the three cases.

In figures 4-7, 4-8, and 4-9 we have two images in the left side, they are the RETRIEVED FROG from FROG program and the EXPERIMENTAL FROG obtained from the experimental setup. The horizontal axis are different DELAYS and the vertical axis are different FREQUENCIES, both graphics just show the position of the PIXEL in the image, from 1 to 250. As explained above, to be able to use these image in our FROG program we need to convert them to an images of 256×256 pixels. To do that we resized the CCD camera image by respecting the calibration. The other four graphics in the right side are respectively, DELAY (in ps) vs AMPLITUDE (normalized), $\nu - \nu_o$ ($\nu_o = 387$ THz) vs AMPLITUDE (normalized), DELAY (ps) vs PHASE (rad), $\nu - \nu_o$ ($\nu_o = 387$ THz) vs PHASE (rad). And finally the graphic in the bottom is ITERATION vs ERROR.

We now need to identify the parameter φ_2 in order to calculate the spool length L given by Eq. (3.17).

For that purpose we first find the best Gaussian function that can fit the FROG amplitudes $\sqrt{I(t)}$ (Eq. (3.10)) and $\sqrt{S(\omega)}$ (Eq. (3.14)) in the time and frequency

domain, respectively. This was done using a Matlab fitting program. Having obtained approximate values of $\Delta\tau$ and $\Delta\omega$ (see Eqs. (3.11) and (3.12), respectively) by fitting the FROG amplitudes, one obtains φ_2 from Eq. (3.15) and ϕ_2 from Eq. (3.16).

We parametrize the FROG amplitude as follows :

$$A(t) = v_1 e^{-\frac{(t-v_2)^2}{v_3^2}} + v_4 \quad (4.3)$$

$$\tilde{A}(\mathbf{v}) = \tilde{v}_1 e^{-\frac{(\mathbf{v}-\tilde{v}_2)^2}{\tilde{v}_3^2}} + \tilde{v}_4 \quad (4.4)$$

where $\mathbf{v} = \nu - \nu_o$, $\nu = \omega/2\pi$ is the frequency and $\nu_o = 387$ THz is the center of the range of frequencies. Comparing with Eqs. (3.10) and (3.14), we see that $\Delta\tau = v_3$ and that $\Delta\omega = 2\pi\tilde{v}_3$. Then φ_2 and ϕ_2 are obtained from Eqs. (3.15) and (3.16), respectively.

The fitting parameters obtained for no spool, the 10 m spool and the 100 m spool are shown in Tables 4-2, 4-3 and 4-4, respectively.

TABLE 4-2 – Parameters for no spool. Paramètres obtenus sans bobine.

For frequency	
\tilde{v}_1	0.9171
\tilde{v}_2	0.0006
\tilde{v}_3	0.4547
\tilde{v}_4	0.0000
For time	
v_1	0.9856
v_2	0.0007
v_3	0.7000
v_4	0.0004

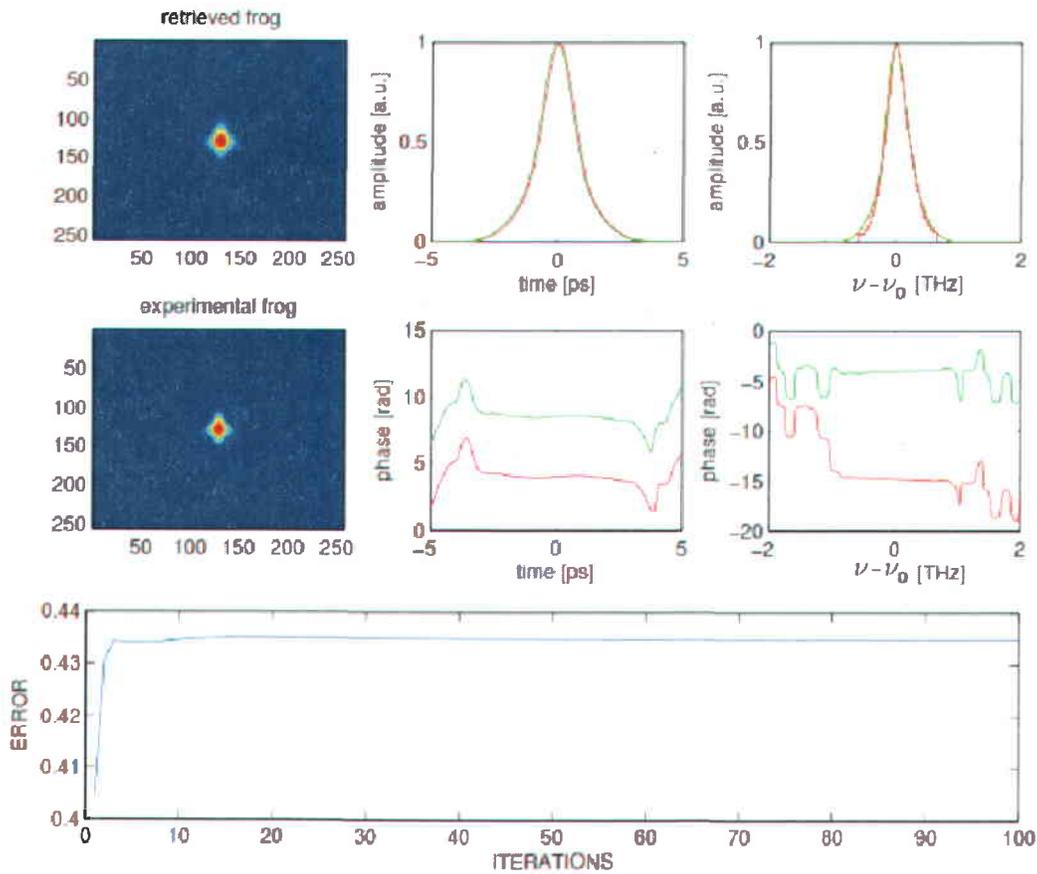


FIGURE 4-7 – FROG retrieval for no spool for 1 iteration (green curves) and 100 iterations (red curve). We have two images in the left side, they are the RETRIEVED FROG from FROG program and the EXPERIMENTAL FROG obtained from the experimental setup, the horizontal axis are different DELAYS and the vertical axis are different FREQUENCIES, both graphics just show the position of the PIXEL in the image, from 1 to 250. The other four graphics in the right side are respectively, DELAY (in ps) vs AMPLITUDE (normalized), $\nu - \nu_o$ ($\nu_o = 387$ THz) vs AMPLITUDE (normalized), DELAY (ps) vs PHASE (rad), $\nu - \nu_o$ ($\nu_o = 387$ THz) vs PHASE (rad). And finally the graphic in the bottom is ITERATION vs ERROR. Récupération FROG sans bobine pour 1 itération (courbes vertes) et 100 itérations (courbe rouge).

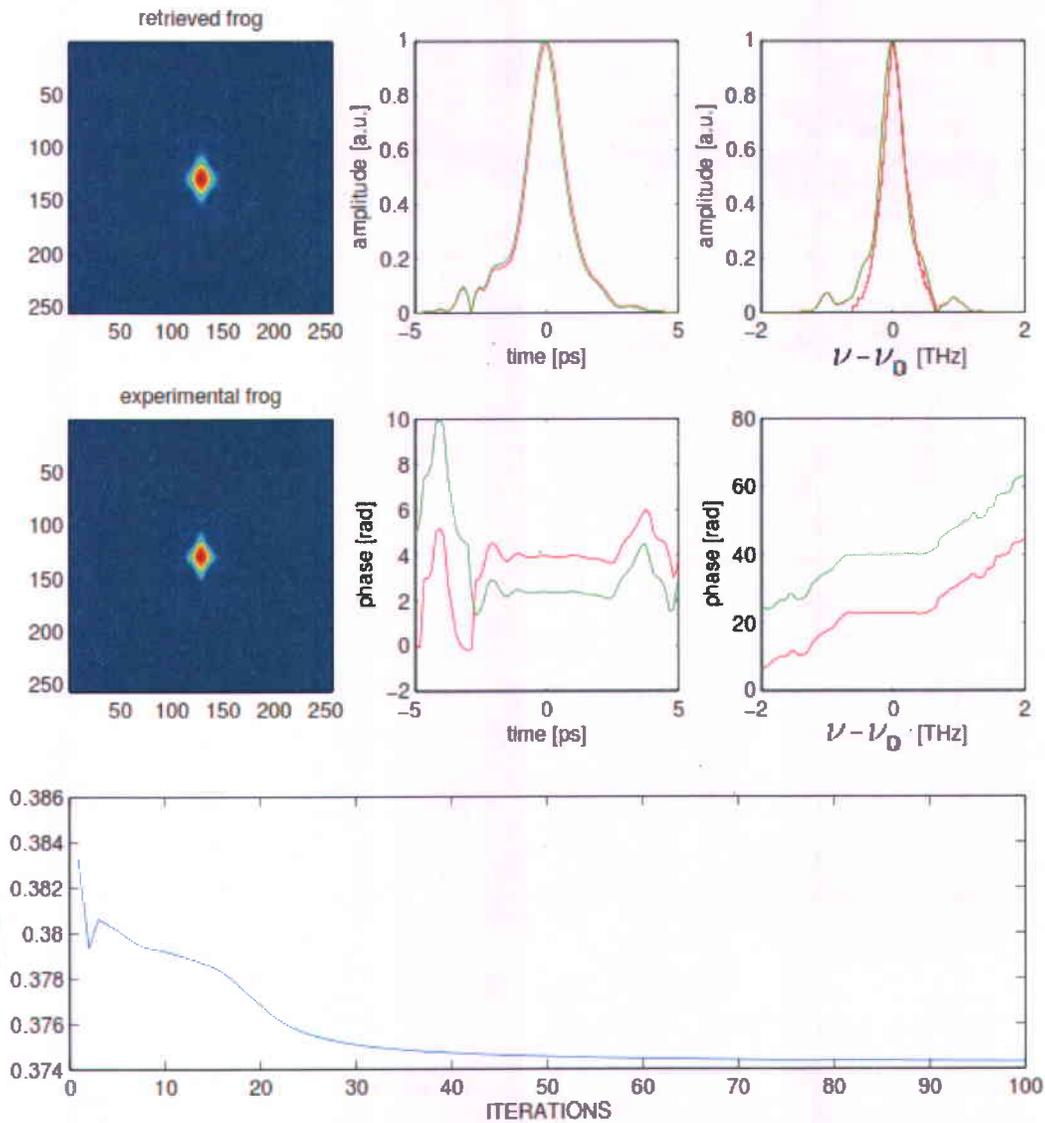


FIGURE 4-8 – FROG retrieval for the 10 m spool for 1 iteration (green curves) and 100 iterations (red curve). Récupération FROG pour la bobine de 10 m pour 1 itération (courbes vertes) et 100 itérations (courbe rouge).

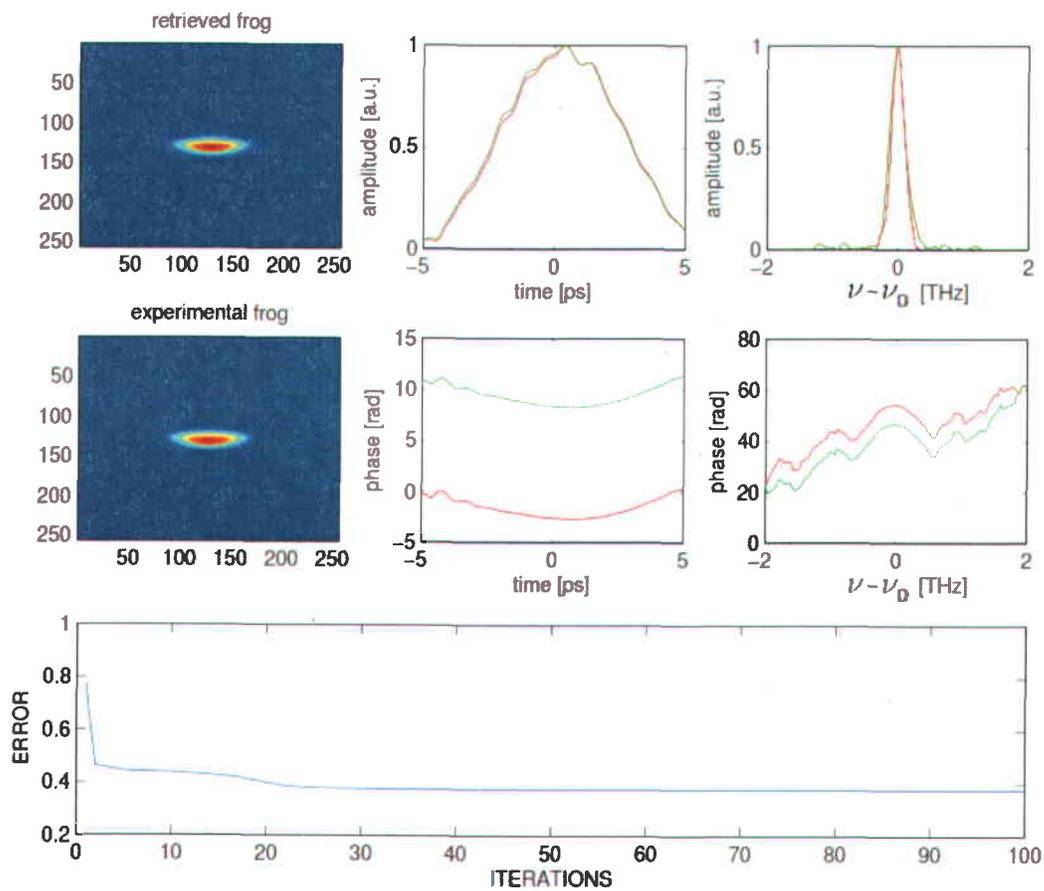


FIGURE 4-9 – FROG retrieval for the 100 m pool for 1 iteration (green curves) and 100 iterations (red curve). Récupération FROG pour la bobine de 100 m pour 1 itération (courbes vertes) et 100 itérations (courbe rouge).

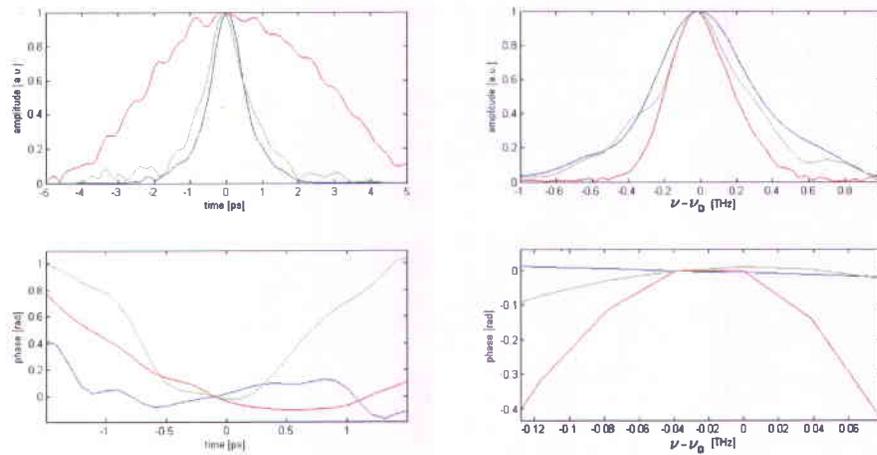


FIGURE 4-10 – Amplitudes and phases in the time(delay) and frequency domains ($\nu_0 = 387$ THz) for the 100 m (red), 10 m (green) and 0 m (blue) spools retrieved from FROG. Amplitudes et phases dans le domaine temporel(delay) et des fréquences pour des bobines de 100 m (rouge), 10 m (verte) et 0 m (bleu) obtenues par FROG.

TABLE 4-3 – Parameters for the 10 m spool. Paramètres obtenus pour la bobine de 10 m.

For frequency	
\tilde{v}_1	0.8033
\tilde{v}_2	0.0012
\tilde{v}_3	0.3979
\tilde{v}_4	0.0023
For time	
v_1	0.8057
v_2	0.0003
v_3	0.9250
v_4	0.0000

The pulses retrieved from FROG and the fitted pulses are compared in Figures 4-11, 4-12 and 4-13 for the $L = 0, 10$ and 100 m spool, respectively. From the fitting parameters we can retrieve $\Delta\tau$, $\Delta\omega$, ϕ_2 and φ_2 . The values of these parameters are given in Table 4-5 for the three spools.

4.3 COMPARISON BETWEEN THE THREE PULSES RETRIEVED FROM FROG : SIMPLIFIED REPRESENTATION

Using the fitting functions eqs. (4.3) and (4.4) and parameters obtained from the fits, it is thus possible to replot in a simplified way the amplitudes $\sqrt{I(t)}$ and $\sqrt{S(\omega)}$, as well as $\phi(t)$ and $\varphi(\omega)$. The fits for the amplitudes and phases for the three spools are shown in figure 4-14. The advantages of those curves is to provide a simplified representation of the pulses' features and allow an easier comparison between them.

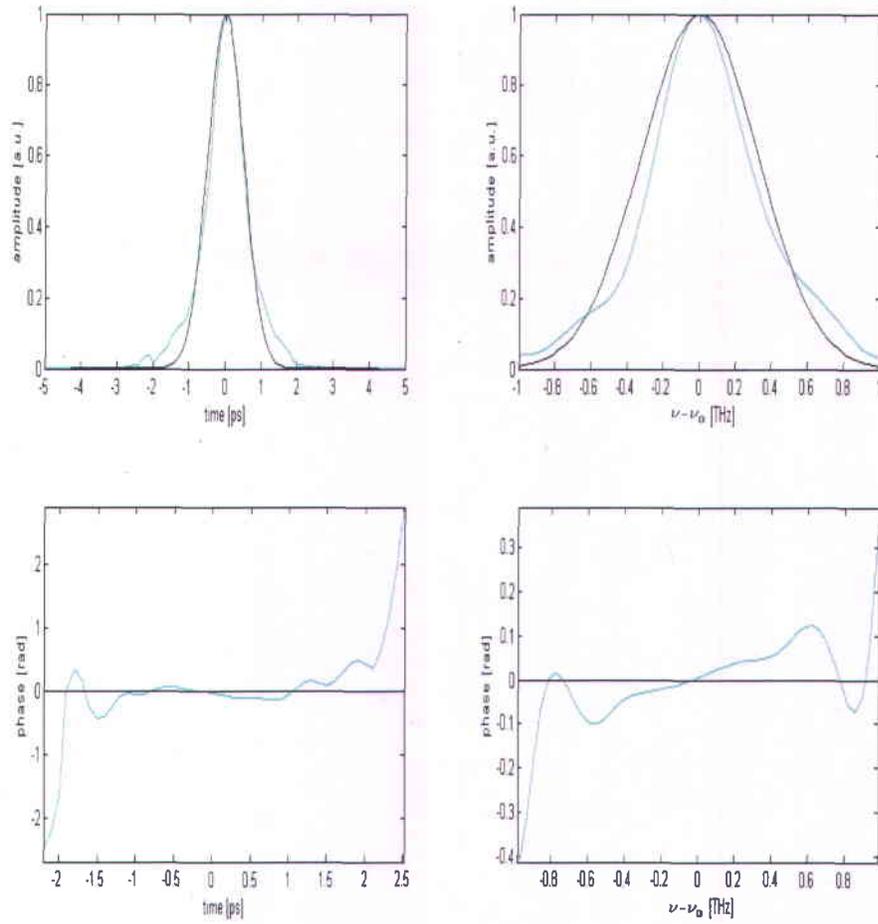


FIGURE 4-11 – Retrieved from FROG characterization (light blue) and the fitted pulses for no spool (black line) are compared. Caractérisation FROG (bleu claire) et fit sans bobine (noire).

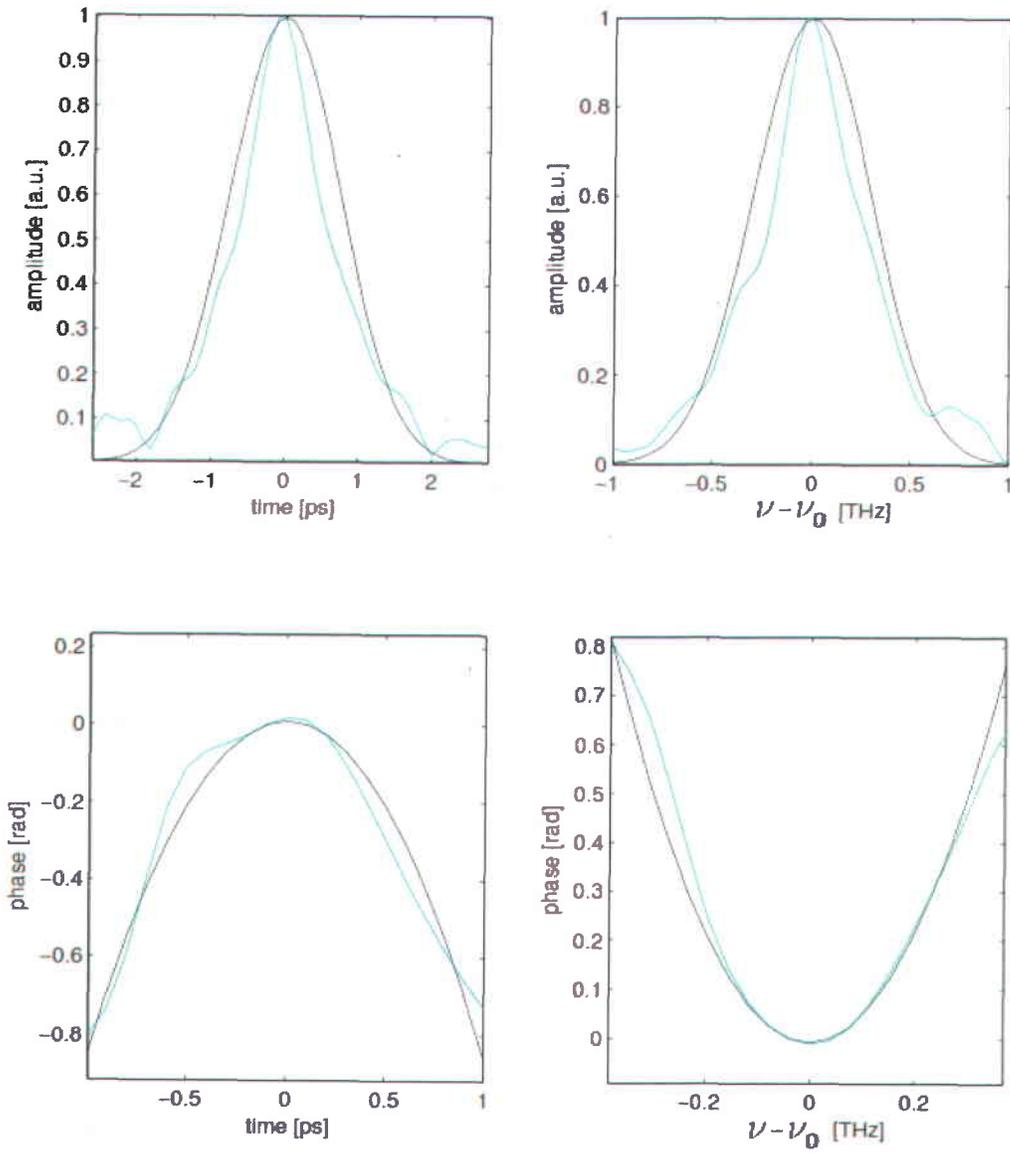


FIGURE 4-12 – FROG characterization (light blue) and fit for the 10 m spool (black line). Caractérisation FROG (bleu claire) et fit pour la bobine de 10 m (noire).

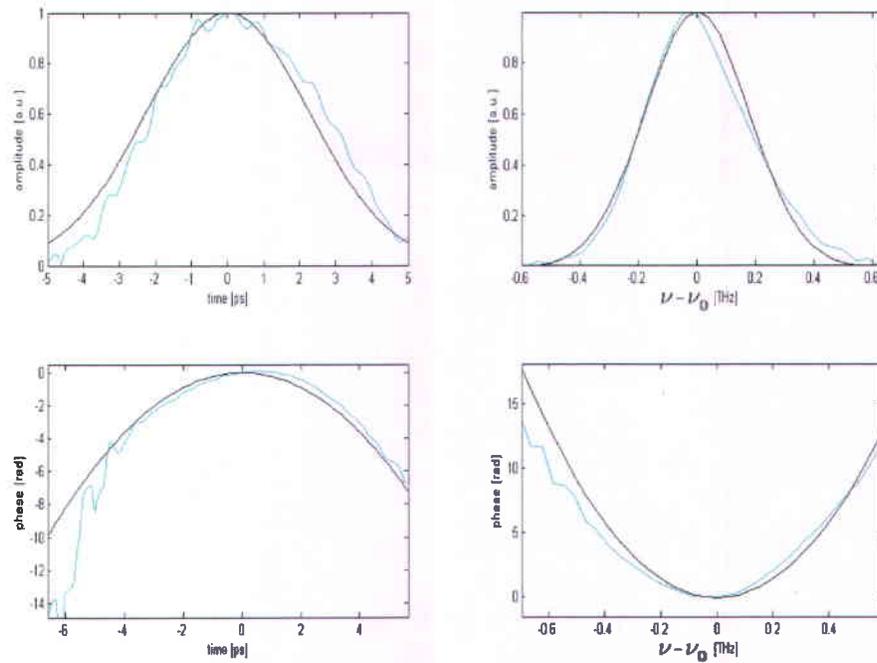


FIGURE 4-13 – FROG characterization (light blue) and fit for the 100 m spool (black line). Caractérisation FROG (bleu claire) et fit pour la bobine de 100 m (noire).

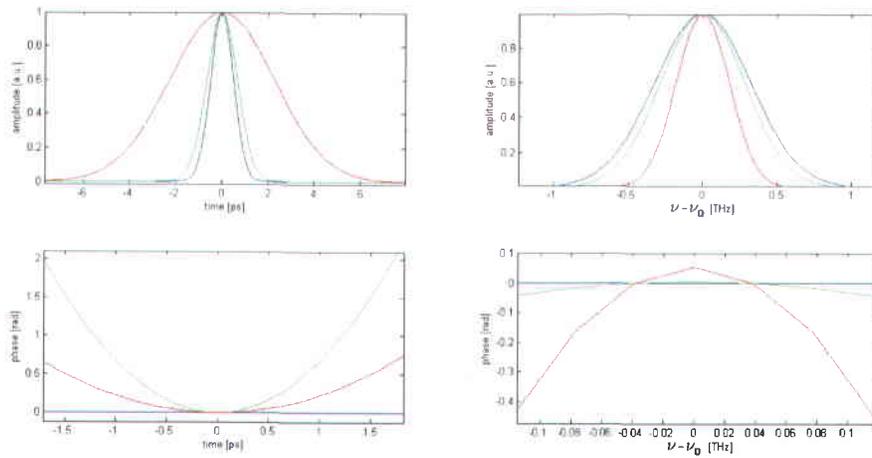


FIGURE 4-14 – Simplified amplitudes and phases in the time and frequency domains for the 100 m (red), 10 m (green) and 0 m (blue) spools, it seems that we are losing frequencies as the propagation distance increases. Amplitudes et phases simplifiées dans le domaine temporel et des fréquences pour des bobines de 100 m (rouge), 10 m (verte) et 0 m (bleu).

TABLE 4-4 – Parameters for the 100 m spool. Paramètres obtenu pour la fibre de 100 m.

For frequency	
\tilde{v}_1	0.9900
\tilde{v}_2	0.0028
\tilde{v}_3	0.2518
\tilde{v}_4	0.0006
For time	
v_1	1.0031
v_2	-0.0091
v_3	3.2003
v_4	0.0000

TABLE 4-5 – Physical parameters extracted from the fits. Paramètres physiques extraits des fits.

Spool length (m)	$\Delta\tau$ (ps)	$\Delta\omega$ (rad/s)	ϕ_2 (ps ⁻²)	φ_2 (ps ² /rad ²)
0	0.7	2.8571	0.	0.
10	0.9250	2.5	-1.7211	0.1857
100	3.2003	1.5824	-0.9213	1.8580

4.4 OBTAINING THE SPOOL LENGTH

From the parameter φ_2 given in Table 4-5, it is now possible to calculate the spool lengths using the expression $L = \varphi_2/b_2$. The results are displayed in Table 4-6 for two different values of the intrinsic dispersion coefficients of the fiber at 1550 nm, b_2 .

TABLE 4-6 – Calculated spool lengths. Longueur des bobines calculée.

Actual pool length (m)	Calculated pool length (m)
	$b_2 = 18.5 \text{ ps}^2/\text{km}$
0	0.
10	10.04
100	100.4

The calculated values for L are in excellent agreement with the actual spool lengths and the calculated value of the dispersion given by $b_2 = 18.5 \text{ ps}^2/\text{km}$ is close to the value given by books $b_2 = 22 \text{ ps}^2/\text{km}^1$. We believe that the later value is more accurate for the wavelength of 1550 nm and for our experiment.

1. www.rp-photonics.com/chromatic-dispersion.html

Chapitre 5 SUMMARY AND CONCLUSION

The purpose of this work was to build a FROG setup and to develop various software for taking measurements and running the FROG numerical algorithm for retrieving the pulse's amplitude and phase from the FROG trace. This FROG system was then validated by measuring the length of the optical fiber spools. Since we know the actual lengths of the spools, we can compare them to the lengths calculated by using the pulse features obtained from the FROG system and by using the intrinsic dispersion coefficient of the fiber b_2 at a wavelength of 1550 nm.

The length of the spool, given by $L = \varphi_2/b_2$, where φ_2 , given by Eq. (3.15), is obtained from the FROG analysis as explained in Chapter 4. The calculated lengths using the value $b_2 = 18.5 \text{ ps}^2/\text{km}$ give the values shown in Table 5-1.

Table 5-1 includes the results we were seeking,

TABLE 5-1 – Results.

Real Length	Retrieve Length
100 m	100.432m
10 m	10.038m
0 m	0.000 m

Using these data we can plot the curve shown in Figure 5-1, which represents the the calculated length vs. the real length. A linear fit through the three points

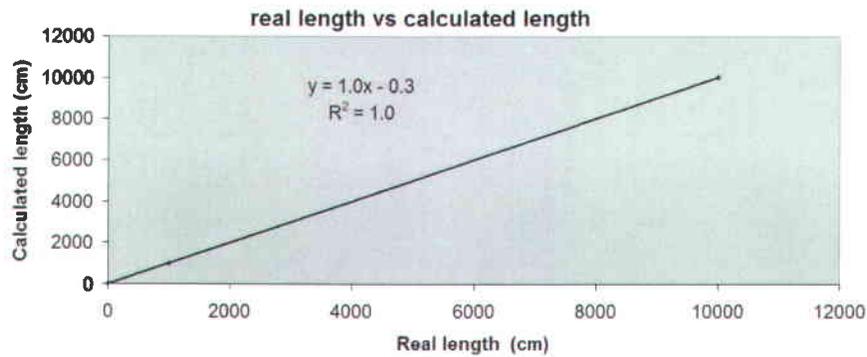


FIGURE 5-1 – Real length vs. calculated length. Vrai longueur vs. longueur calculé.

yields $y = 1.0x - 0.3$ cm and $R = 1$, indicating that the FROG system is working correctly and can be used with confidence. My FROG setup has already been used in an experiments and the results were published in the journal NATURE PHOTONICS LETTERS [50].

The pulse is significantly compressed in the frequency domain after 100 m of propagation in the fiber. There are temporal oscillations and asymmetry in this temporal pulse, which means that some third order dispersion appears, but it is not strong enough to create a big asymmetry in the pulse and it only appears for the 10 and 100 m fibers. The second order alone is enough strong to compress the pulse in the frequency domain.

Chapitre 6 SOMMAIRE EN FRANÇAIS

6.1 IMPULSIONS LASER ULTRA BRÈVES

Le “Frequency-Resolved Optical Gating”, plus couramment désigné par son acronyme anglais FROG, est une méthode de mesure des impulsions lumineuses ultra brèves (de l'ordre de la femtoseconde) par auto-corrélation. Son principe repose sur un montage de type Michelson, dans lequel deux impulsions issues d'une même impulsion laser convergent dans un cristal non-linéaire pour y générer un signal de seconde harmonique (i.e. dont la fréquence est double de celle des impulsions incidentes). C'est de ce signal dont on mesure le spectre. En faisant varier pas à pas la longueur de l'un des bras du Michelson, on peut former une trace FROG en deux dimensions (fréquence et délai). Un algorithme itératif est ensuite utilisé pour restituer l'amplitude et la phase de l'onde incidente.

Le problème de la mesure des impulsions courtes est pratiquement né avec le laser, dans les années 60. Comment mesurer un évènement plus rapide que le temps de réponse de l'instrument de mesure? Tous les progrès de l'électronique ne permettent pas de descendre en dessous de temps de réaction de l'ordre de la picoseconde. Pour contourner ce problème, l'auto-corrélation fut rapidement utilisée, employant par exemple un interféromètre pour diviser l'impulsion à mesurer et la faire réagir avec elle-même. Cette méthode (auto-corrélation), utilisant par sa conception même un instrument de mesure aussi rapide que la grandeur à

mesurer, produit de bons résultats, mais ne permet pas de déterminer la phase. On peut retracer au milieu des années 80 la première mention de l'utilisation de l'auto-corrélation associée à un élément non-linéaire pour obtenir le spectre du signal mesuré. Mais c'est en 1991, avec l'invention de FROG par Rick Trebino et Dan Kane, que la mesure très précise d'impulsions ultra brèves en intensité et en fréquence est devenu non seulement possible, mais relativement simple.

6.2 SYSTÈME FROG

Il est impossible de mesurer correctement un signal avec un autre signal plus long. L'auto-corrélation utilise un signal de la même durée, mais elle ne permet pas de résolution spectrale. Ce qui, dans FROG, permet la résolution spectrale, c'est l'utilisation d'un élément qui, a un temps de réponse plus rapide que la durée de l'impulsion lumineuse : le cristal non-linéaire. Il existe plusieurs types de FROG selon l'emploi fait du cristal non-linéaire : le plus simple est la génération de seconde harmonique à partir des deux ondes incidentes dans le cristal.

Comme dans l'auto-corrélation, FROG implique la mesure de l'impulsion avec avec elle-même. Sous sa forme la plus simple, FROG est n'importe quelle sorte d'auto-corrélation dans laquelle le faisceau du signal de l'auto-corrélateur est spectralement résolu. Au lieu de mesurer le signal d'énergie de l'auto-corrélateur par rapport au retard, qui rapporte une auto-corrélation, nous mesurons le spectre du signal par rapport au retard. Pour caractériser le faisceau, nous employons le spectrographe de formation d'image NEWPORT 1/4m modèle MS260i. Nous analysons le spectre du faisceau primaire de 1550 nm en utilisant le spectromètre de fibre, OSA, modèle

AQ6317B. Le faisceau primaire est ensuite divisé en utilisant un séparateur de faisceau. L'un des faisceaux divisés va vers la platine mobile de THORLABS, puis au cristal de génération de seconde harmonique (SHG). L'autre faisceau divisé va directement au cristal de SHG. Les faisceaux divisés doivent interagir à l'intérieur du cristal de SHG, mais avec un angle de 39.2 degrés pour produire un faisceau de 775 nm de longueur d'onde (seconde harmonique). Ce faisceau de 775 nm est par la suite directement envoyé au spectrographe de formation d'image, ORIEL 1/4*m* (MS260i) pour en obtenir le spectre.

À partir de différents délais entre les deux impulsions, nous obtenons différents spectres qui constituent les données d'entrées (trace FROG) du système FROG.

6.2.1 Spectrographe de formation d'images ORIEL 1/4*m*

Le spectrographe 74086 MS260i 1/4*m*, est un instrument multi-grille. Ce modèle de sortie double utilise une grille double. C'est un instrument F/3.9 avec un détecteur multicanaux plat, de plan d'image de 28 mm. La sortie possède une résolution élevée. Un ensemble de miroirs toroïdaux optimisés par ordinateur, dont l'orientation est optimisée.

6.2.2 Les programmes

Pour la communication entre la caméra CCD du spectromètre et l'ordinateur, nous avons développé un code en langage Matlab. Ce code se trouve dans les annexes A, B et C. Un code en Matlab pour calibrer la caméra est également donné dans l'annexe D. À la suite de l'obtention de la trace FROG, nous employons un code pour manipuler l'image et pour l'adapter au programme Matlab FROG 1.0 qui exécute l'algorithme permettant d'extraire l'amplitude et la phase de l'impulsion laser.

6.2.3 Platine linéaire motorisée

La platine linéaire motorisée est commandée par le BENCHTOP DC SERVO MOTOR CONTROLLER et permet une acquisition continue d'image de la caméra. Elle est illustrée dans l'annexe F.

6.3 MESURE DES IMPULSIONS OPTIQUES ULTRA BRÈVES

Le but du système FROG est de mesurer l'amplitude du champ électrique vs le temps, $\sqrt{I(t)}$, et la phase, $\phi(t)$. Le champ électrique complexe $E(t)$ est donné par :

$$E(t) = \sqrt{I(t)} \exp[i(\omega_0 t - \phi(t))] \quad (6.1)$$

où ω_0 est la fréquence angulaire centrale de l'onde. Le champ électrique dans le domaine de fréquence est obtenu par transformation de Fourier de $E(t)$:

$$\tilde{E}(\omega) = \sqrt{S(\omega - \omega_0)} \exp[-i\varphi(\omega - \omega_0)] \quad (6.2)$$

où $\sqrt{S(\omega)}$ est l'amplitude du spectre de l'impulsion et $\varphi(\omega)$ est la phase spectrale.

Après avoir divisé l'impulsion laser en deux parties, on superpose ensuite ces deux parties dans un milieu non linéaire optique, tel qu'un cristal de génération de seconde harmonique (SHG). On va ensuite retarder de façon variable une impulsion par rapport à l'autre.

L'énergie du signal résultant de la superposition des deux impulsions, qui est mesurée par la technique d'auto-corrélation, s'exprime :

$$\int_{-\infty}^{\infty} I(t)I(t - \tau)dt \quad (6.3)$$

où τ est le retard relatif entre les deux impulsions laser.

Le problème avec l'auto-corrélation simple est qu'il est nécessaire de supposer une forme de l'impulsion afin d'obtenir sa longueur. Ce problème s'appelle problème unidimensionnel de récupération de la phase¹ [2 – 3]. Par ailleurs, nous ne pouvons pas obtenir des informations sur la phase $\phi(t)$. Même si nous pouvions générer toutes les impulsions laser possibles correspondant à une trace donnée, nous obtiendrions plusieurs possibilités et nous ne pourrions pas déterminer lesquelles correspondent l'impulsion réelle.[4 – 9, 46]

6.4 FREQUENCY-RESOLVED OPTICAL GATING (FROG)

Le FROG est une méthode générale pour mesurer les impulsions laser ultra brèves qui s'étendent en durée de la sub-femtoseconde à environ une nanoseconde. Comme mentionné plus haut, FROG est une technique d'auto-corrélation, mais où l'impulsion est spectralement résolue en fonction du retard entre deux impulsions laser. Cette technique fait appel à un algorithme de récupération pour obtenir l'intensité et la phase de l'impulsion en fonction du temps.[10 – 12, 14 – 21]². Des informations sur les premiers travaux sur le système FROG sont disponibles à <http://frog.gatech.edu/publications.html>. [22 – 39]

1. Trebino, R., Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses. 2002, Boston : Kluwer Academic Publishers.

2. <http://www.physics.gatech.edu/gcuo/subIndex.html>

L'expression générale de la trace FROG mesurée est :

$$I_{FROG}(\omega, \tau) = \left| \int_{-\infty}^{\infty} E_{sig}(t, \tau) e^{-i\omega t} dt \right|^2 \quad (6.4)$$

où $E_{sig}(t, \tau)$ est une fonction du temps et du retard :

$$E_{sig}(t, \tau) = E(t)E_{gate}(t - \tau) \quad (6.5)$$

Dans la technique FROG, la fonction $E_{gate}(t - \tau)$, est une fonction de l'impulsion d'entrée inconnue, $E(t)$, que nous essayons de mesurer. La fonction $E_{gate}(t - \tau)$ peut être n'importe quelle fonction connue agissant en tant qu'impulsion de référence. Généralement, $E_{sig}(t, \tau)$ peut être n'importe quelle fonction du temps et du retard qui contient suffisamment d'information pour déterminer les caractéristiques de l'impulsion. [13]

Certains chercheurs sont d'avis que le FROG comporte quelques ambiguïtés non triviales. Ces mêmes auteurs avaient également conclu que l'auto-corrélation peut être plus sensible aux variations de l'impulsion que FROG. Ces deux énoncés sont erronés, comme démontré dans [40 – 49].

6.5 MOTIVATIONS

L'objectif des chercheurs de notre groupe est de caractériser les impulsions laser dans différentes expériences. Dans ce but, ils doivent mesurer avec précision l'amplitude et la phase en fonction du temps et de la fréquence du faisceau laser qu'ils utilisent en entrée ainsi que du faisceau laser en sortie du dispositif expérimental. Plus de détails sur l'utilité de cette caractérisation sont disponibles dans l'article

publié par notre groupe dans NATURE PHOTONICS LETTERS “Sub-picosecond phase-sensitive optical pulse characterization on a chip”. [50]

6.6 STRUCTURE DU MÉMOIRE

Dans le chapitre 1, nous présentons le sujet et la structure du document. Nous commençons par une introduction au domaine des impulsions laser ultra brèves et du FROG avec un bref historique de la mesure des impulsions laser ultra brèves. Nous expliquons également son intérêt pour notre groupe. Dans le chapitre 2 nous discutons l’installation expérimentale et le matériel que nous employons. Nous fournissons des informations courtes sur la spectrométrie, le programme que nous employons pour obtenir les images et contrôler la platine linéaire motorisée. Dans le chapitre 3 nous donnons une base mathématique à l’impulsion que nous essayons de caractériser. Au chapitre 4, nous discutons les résultats expérimentaux et les coefficients du calibrage nécessaires pour préparer les données d’entrée de l’algorithme itératif du FROG. Nous présentons les résultats de l’analyse FROG pour des bobines de fibres optiques de 100 m, 10 m et 0 m (aucune bobine). Dans le chapitre 5, nous concluons le mémoire, et discutons les valeurs expérimentales que nous recherchions, i.e. les longueurs des fibres optiques, et nous faisons la comparaison avec leurs longueurs réelles.

6.7 THÉORIE MATHÉMATIQUE

Comme mentionné plus haut, une impulsion laser ultra brève possède une intensité et une phase. Négligeant la dépendance spatiale, le champ électrique complexe

d'une impulsion est donné par l'Eq. (6.1). La transformée de Fourier du champ électrique, est donnée par l'équation Eq. (6.2).

La phase $\phi(t)$ détermine la variation de la forme de la fréquence de l'impulsion en fonction du temps. La fréquence, ou la couleur, de l'impulsion en fonction du temps est :

$$\Omega(t) = -d\phi(t)/dt \quad (6.6)$$

La variation de la fréquence avec le temps s'appelle le "chirp". Le chirp positif est une fréquence qui croît avec le temps et le chirp négatif est une fréquence qui décroît avec le temps. Mais des chirps plus complexes sont également un terrain tout à fait connu. En effet, la maîtrise des chirps complexes dans les impulsions laser ultra brèves joue un rôle important dans la génération de ces impulsions dans des lasers modelocked et est la clef vers la production d'impulsions laser encore plus courtes.

6.7.1 Séries de Taylor de la phase

La série de Taylor pour la phase dépendante du temps autour de $t = 0$ s'exprime :

$$\phi(t) = \phi_0 + \phi_1 \frac{t}{1!} + \phi_2 \frac{t^2}{2!} + \dots \quad (6.7)$$

où :

$$\phi_n = \left. \frac{d^n \phi}{dt^n} \right|_{t=0} \quad (6.8)$$

En substituant ce développement en série dans l'Eq. (6.1), on voit que le premier terme ϕ_0 est seulement une phase globale sans importance, alors que le second terme ϕ_1 induit un décalage dans la fréquence centrale ω_0 , supposée connue. Donc ces deux termes peuvent être posés égaux à zéro. Souvent, comme dans ce travail, seulement le terme suivant ϕ_2 est utile pour décrire les impulsions simples. Évidemment, pour

les impulsions complexes, il est nécessaire de prendre en compte les termes d'ordre supérieurs.

Un développement similaire en série de Taylor peut être écrit pour $\varphi(\omega)$ autour de $\omega = \omega_0$:

$$\varphi(\omega - \omega_0) = \varphi_0 + \varphi_1 \frac{\omega - \omega_0}{1!} + \varphi_2 \frac{(\omega - \omega_0)^2}{2!} + \dots \quad (6.9)$$

où :

$$\varphi_n = \left. \frac{d^n \varphi}{d\omega^n} \right|_{\omega=\omega_0} \quad (6.10)$$

En substituant ce développement en série dans l'Eq. (6.2), on voit que le premier terme φ_0 est seulement une phase globale. Le second terme φ_1 induit un retard dans l'impulsion et est appelé retard de groupe (group delay) pour cette raison. Donc ces deux termes peuvent être posés égaux à zéro. Le troisième terme φ_2 décrit la dispersion de l'impulsion dans le milieu et est appelé dispersion de retard de groupe (group delay dispersion ou GDD).

6.7.2 Impulsions gaussiennes

Une bonne première approximation pour l'amplitude des impulsions laser est la forme gaussienne, ici centrée en $t = 0$:

$$\sqrt{I(t)} = E_0 \exp[-(t/\Delta\tau)^2] \quad (6.11)$$

où τ est la durée d'impulsion et E_0 est l'amplitude maximum.

Substituant l'Eq.(6.11) et $\phi(t) = \phi_2 t^2/2$ dans l'Eq.(6.1), on obtient :

$$E(t) = E_0 e^{-\frac{t^2}{\Delta\tau^2}} e^{-i\frac{\phi_2}{2}t^2} e^{i\omega_0 t} \quad (6.12)$$

En prenant la transformée de Fourier de ce champ électrique on obtient :

$$\tilde{E}(\omega) = \tilde{E}_0 e^{-\frac{(\omega-\omega_0)^2}{\Delta\omega^2}} e^{-i\frac{\varphi_2}{2}(\omega-\omega_0)^2} \quad (6.13)$$

où \tilde{E}_0 est l'amplitude spectrale maximum,

$$\Delta\omega^2 = \frac{\phi_2^2 \Delta\tau^4 + 4}{\Delta\tau^2} \quad (6.14)$$

est la largeur de l'impulsion dans l'espace des fréquences et :

$$\varphi_2 = \sqrt{\frac{\Delta\tau^2 \Delta\omega^2 - 4}{\Delta\omega^4}} \quad (6.15)$$

est le GDD. On note que l'amplitude spectrale est :

$$\sqrt{S(\omega)} = \tilde{E}_0 e^{-\frac{(\omega-\omega_0)^2}{\Delta\omega^2}} \quad (6.16)$$

En utilisant les Eqs. (6.15) et (6.14), on obtient :

$$\phi_2 = -\frac{\varphi_2 \Delta\omega^4}{4 + \varphi_2^2 \Delta\omega^4} \quad (6.17)$$

Par conséquent, si nous obtenons $\Delta\tau$ et $\Delta\omega$ par l'analyse FROG, nous pouvons obtenir φ_2 et ϕ_2 , ce qui nous donne une caractérisation complète de l'impulsion en supposant que les termes d'ordre supérieurs dans la série de Taylor peuvent être négligés.

6.8 L'IDÉE PRINCIPALE DE L'EXPÉRIENCE

Dans le but de valider le montage expérimental FROG et les logiciels associés, nous avons utilisé trois bobines de fibres optiques ayant des longueurs de 0 (pas de

bobine), 10 et 100 m. En utilisant le dispositif FROG, nous pourrions calculer la longueur des bobines simplement à partir des caractéristiques de l'impulsion et en connaissant le coefficient de dispersion de la fibre optique, b_2 , à la longueur d'onde du laser. La longueur de la bobine est alors donnée par :

$$L = \varphi_2/b_2 \quad (6.18)$$

où φ_2 est le GDD.

6.9 MÉTHODOLOGIE

6.9.1 Traitement des spectres expérimentaux

Les spectres des bobines sont recueillis avec une caméra CCD de 1024×100 pixels. L'image à être utilisée par l'algorithme FROG doit être une matrice carrée de dimension $N = 2^{ord}$ (où nous avons utilisé $ord = 8$) dont les axes en temps et en fréquence sont calibrés en fonction des intervalles de délai et de fréquence mesurés. Afin de dimensionner et calibrer correctement la matrice carrée nous employons un programme en Matlab qui découpe et interpole la matrice expérimentale. Nous pouvons voir le programme Matlab correspondant dans l'annexe E.

Une fois que les amplitudes $\sqrt{I(t)}$ et $\sqrt{S(\omega)}$ ont été obtenues par l'analyse FROG nous pouvons extraire les largeurs en temps $\Delta\tau$ et en fréquence $\Delta\omega$ de l'amplitude de l'impulsion et en déduire les composantes de phase ϕ_2 et φ_2 .

6.9.2 Les images expérimentales

Nous pouvons voir les spectres bruts recueillis par la caméra CCD aux Figure 4-1, 4-2 et 4-3 respectivement pour les bobines de 0, 10 et 100 m. Ces images sont de

dimension 1024×100 pixels (1024 en fréquence et 100 en délai). Pour pouvoir les employer dans le programme FROG 1.0, nous devons transformer ces images dans le format 256×256 et les interpoler pour garder le calibrage. Les images résultant de cette opération sont illustrées sur les Figures 4-4, 4-5 et 4-6 qui correspondent respectivement aux bobines de 0, 10 et 100 m.

6.9.3 Les images résultant de l'analyse FROG

Les Figures 4-7, 4-8 et 4-9 représentent les traces FROG expérimentales et celles résultant de l'algorithme FROG, ainsi que les amplitudes et les phases en fonction du temps et de la fréquence. L'évolution de l'erreur en fonction des itérations dans le programme FROG 1.0 est aussi représentée.

6.10 COMPARAISON DES RÉSULTATS POUR LES TROIS BOBINES

6.10.1 Analyse FROG

Dans la Figure 4-10 nous comparons les résultats de l'analyse FROG obtenus pour les bobines de 0, 10 et 100 m. Ces figures démontrent l'augmentation attendue de la durée d'impulsion en fonction de la longueur de la bobine.

6.10.2 Calcul des paramètres

Comme mentionné plus haut, si nous connaissons τ et $\Delta\omega$ nous pouvons obtenir φ_2 et ϕ_2 . Pour extraire les paramètres τ et $\Delta\omega$ nous effectuons un fit des amplitudes obtenues par FROG (par exemple les amplitudes de la Figure 4-10) à l'aide des fonctions :

$$A(t) = v_1 e^{-\frac{(t-v_2)^2}{v_3^2}} + v_4 \quad (6.19)$$

$$\tilde{A}(\mathbf{v}) = \tilde{v}_1 e^{-\frac{(\mathbf{v}-\tilde{v}_2)^2}{\tilde{v}_3^2}} + \tilde{v}_4 \quad (6.20)$$

où $\mathbf{v} = \nu - \nu_o$, $\nu = \omega/2\pi$ est la fréquence et $\nu_o = 387$ THz est la fréquence central. En comparant ces expressions avec les Eqs.(6.11) et (6.16), on voit que $\Delta\tau = v_3$ et $\Delta\omega = 2\pi\tilde{v}_3$. Les phases φ_2 et ϕ_2 peuvent ensuite être obtenues respectivement des Eqs. (6.15) and (6.17). Les coefficients obtenus des fits sont donnés dans les Tableaux 4-2, 4-3 et 4-4. Les Figures 4-11, 4-12 et 4-12 montrent les amplitudes et les phases obtenues des fits pour chacune des trois bobines. La Figure 4-14 montre les amplitudes et les phases obtenues des fits pour les trois bobines.

6.11 CONCLUSION

Le but de ce travail a été de réaliser un système FROG incluant les différents logiciels pour prendre les mesures et exécuter l'algorithme numérique FROG permettant de trouver l'amplitude et la phase en temps et en fréquence. Ce système FROG a ensuite été validé en mesurant la longueur des fibres optiques dans des bobines de 0 (pas de bobine), 10 m et 100 m. Puisque nous connaissons la longueur des fibres optiques, nous pouvons les comparer avec celles qui sont calculées en utilisant les caractéristiques des impulsions extraites du système FROG et en utilisant le coefficient de dispersion intrinsèque de la fibre b_2 à une longueur d'onde de 1550 nm.

La longueur des fibres est donnée par $L = \varphi_2/b_2$, où φ_2 , est obtenue de l'analyse FROG comme expliqué plus haut. Les longueurs calculées en utilisant $b_2 = 18.5$ ps²/km sont affichées dans le Tableau 6-1.

Ces résultats indiquent que le système FROG fonctionne correctement et peut être utilisé avec confiance. Ce système FROG a déjà été utilisé dans des expériences

TABLE 6-1 – Resultats

Longueur réelle	Longueur calculée
100 m	100.432m
10 m	10.038m
0 m	0.000 m

dont les résultats ont été publiés dans le journal NATURE PHOTONICS LETTERS [50]. De plus, le système FROG a permis de montrer que le coefficient est plutôt 18.5 ps²/km. b_2 très proche a 22 ps²/km comme supposé initialement.

Bibliographie

- [1] Giordmaine, J.A., P.M. Rentzepis, S.L. Shapiro, and K.W. Wecht , Two-Photon Excitation of Fluorescence By Picosecond Light Pulses. *Appl. Phys. Lett.* **1967**, 216-218, *11(7)*
- [2] Akutowicz, E.J., On the Determination of the Phase of a Fourier Integral, I. *Trans. Amer. Math. Soc.*, 1956. 83(September) : p. 179-192
- [3] Stark, H., ed. *Image Recovery : Theory and Application.* **1987**, Academic Press : Orlando.
- [4] Diels, J.C., J.J. Fontaine, I.C. McMichael, and F. Simoni, Control and Measurement of Ultrashort Pulse Shapes (in Amplitude and Phase) with Femtosecond Accuracy. *Applied Optics*, **1985**. 24(9) : p. 1270-1282.
- [5] Diels, J.C. and J.J. Fontaine, Coherence Properties of Ultrashort Optical Pulses. *Journal of Optics (Paris)*, **1985**. 16(3) : p. 115-119.
- [6] Diels, J.C.M., J.J. Fontaine, N. Jamasbi, and M. Lai. The Femto-nitpicker. in *Conference on Lasers & Electro-Optics.* **1987**
- [7] Mead, C. A. ; Truhlar, D. G. *J. Chem. Phys.* **1979** 70, 2284.
- [8] Trebino, Rick *Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulse, Massachusetts* **2002**
- [9] Yan, C. and J.C. Diels, Amplitude and Phase Recording of Ultrashort Pulses. *Journal of the Optical Society of America B*, **1991**. 8(6) : p. 1259-1263.
- [10] Kane, D.J. and R. Trebino, Single-Shot Measurement of the Intensity and Phase of an Arbitrary Ultrashort Pulse By Using Frequency-Resolved Optical Gating. *Opt. Lett.*, **1993**. 18(10) : p. 823-825.
- [11] Kane, D.J. and R. Trebino, Characterization of Arbitrary Femtosecond Pulses Using Frequency Resolved Optical Gating. *IEEE Journal of Quantum Electronics*, **1993**. 29(2) : p. 571-579.

- [12] Trebino, R., E.K. Gustafson, and A.E. Siegman, Fourth-Order Partial-Coherence Effects in the Formation of Integrated-Intensity Gratings with Pulsed Light Sources *J. Opt. Soc. Amer. B*, **1986**. 3 : p. 1295.
- [13] Stark, H., ed. Image Recovery : Theory and Application. **1987**, Academic Press : Orlando.
- [14] Mairesse, Y. and F. Quéré, Frequency-resolved optical gating for complete reconstruction of attosecond bursts. *Phys Rev A*, **2005**. 71 : p. 011401.
- [15] Akturk, S., M. Kimmel, P. O'Shea, and R. Trebino, Measuring pulse-front tilt in ultrashort pulses using GRENOUILLE. *Opt. Expr.*, **2003**. 11(5) : p. 491-501.
- [16] Gu, X., L. Xu, M. Kimmel, E. Zeek, P. O'Shea, A.P. Shreenath, R. Trebino, and R.S. Windeler, Frequency-resolved optical gating and single-shot spectral measurements reveal fine structure in microstructure-fiber continuum. *Opt. Lett.*, **2002**. 27(13) : p. 1174-6.
- [17] Sansone, G., E. Benedetti, C. Vozzi, S. Stagira, and M. Nisoli, Attosecond metrology in the few-optical-cycle regime. *New J. Phys.*, **2008**. 10 : p. 025006.
- [18] Akturk, S., M. Kimmel, P. O'Shea, and R. Trebino, Measuring spatial chirp in ultrashort pulses using single-shot Frequency-Resolved Optical Gating. *Opt. Expr.*, **2003**. 11(1) : p. 68-78.
- [19] Zhang, J., A.P. Shreenath, M. Kimmel, E. Zeek, R. Trebino, and S. Link, Measuring of the intensity and phase of attojoule femtosecond light pulses using optical-parametric-amplification cross-correlation frequency-resolved optical gating. *Opt. Expr.*, **2003**. 11(6) : p. 601-609.
- [20] Trebino, R., Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses. **2002**, Boston : Kluwer Academic Publishers.
- [21] Trebino, R., Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses. **2002**, Boston : Kluwer Academic Publishers.
- [22] R. Trebino and D.J. Kane, "Using phase retrieval to measure the intensity and phase of ultrashort pulses : frequency-resolved optical gating," *J. Opt. Soc. Am. A*, vol. 10, pp. 1101-1111 (1993).

- [23] K.W. DeLong, R. Trebino, and D.J. Kane, "Measurement of the Intensity and Phase of a Single Ultrashort Pulse : A Simple and General Technique," in *Laser Spectroscopy XI*, ed. L. A. Bloomfield, T. F. Gallagher, and D. J. Larson, 1994.
- [24] D.J. Kane and R. Trebino, "Characterization of Arbitrary Femtosecond Pulses Using Frequency-Resolved Optical Gating," *IEEE J. Quan. Electron.*, vol. 29, pp. 571-579 (1993).
- [25] D.J. Kane, A.J. Taylor, R. Trebino, and K.W. DeLong, "Single-Shot Measurement of the Intensity and Phase of Femtosecond UV Laser Pulse Using Frequency-Resolved Optical Gating", *Opt. Lett.*, vol. 19, pp. 1061-1063 (1994).
- [26] Levine, A. M., E. Ozizmir, R. Trebino, C. C. Hayden, A. M. Johnson, and K. L. Tokuda, "Induced-Grating Autocorrelation of Ultrashort Pulses in a Slowly responding Medium", *J. Opt. Soc. Amer. B*, v. 11, p. 1609-1618 (1994).
- [27] Fourkas, J. T., R. Trebino, M. A. Dugan, and M. D. Fayer, 1993, "Extra Resonances in Time-Domain Four-Wave-Mixing Experiments", *Optics Letters*, v. 18, p. 781-783.
- [28] Kohler, B., V. V. Yakovlev, K. R. Wilson, J. Squier, K. W. DeLong, and R. Trebino, "Phase and Intensity Characterization of Femtosecond Pulses from a Chirped-Pulse Amplifier by Frequency-Resolved Optical Gating", *Optics Letters*, v. 20, p. 483-485 (1994).
- [29] Trebino, R., and I. A. Walmsley, eds., "Generation, Amplification, and Measurement of Ultrashort Laser Pulses", v. 2116 : Bellingham, WA, Society of Photo-Optical Instrumentation Engineers (1994).
- [30] DeLong, K. W., R. Trebino, J. Hunter, and W. E. White, "Frequency-Resolved Optical Gating With the Use of Second-Harmonic Generation", *J. Opt. Soc. Amer. B*, v. 11, p. 2206-2215 (1994).
- [31] Trebino, R., 1993 (appeared in 1995), "Calculating Expressions for Time and Frequency Domain Non-linear Spectroscopic Experiments or Feynman Diagrams Made Simple", *Lithuanian Journal of Physics*, v. 33, p. 358-362.
- [32] Trebino, R., K. W. DeLong, and D. J. Kane, 1993 (appeared in 1995), "Single-Shot Measurement of the Intensity and Phase an Arbitrary Ultra-Short Pulse Using Frequency-Resolved Optical Gating", *Lithuanian Journal of Physics*, v. 33, p. 247-252.

- [44] C. W. Siders, A. J. Taylor, and M. C. Downer, Multipulse Interferometric Frequency- Resolved Optical Gating : Real-time Phase-sensitive Imaging of Ultrafast Dynamics, *Opt. Lett.* *22*, 624-626 **1997**.
- [45] R. Trebino, Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses *Kluwer Academic Publishers*, Boston,**2002**.
- [46] J.-H. Chung, and A. M. Weiner, "Ambiguity of ultrashort pulse shapes retrieved from the intensity autocorrelation and power spectrum," *IEEE J. Sel. Top. Quant. Electron.* *7*, 656-666 ,**2001**.
- [47] D. Keusters, H.-S. Tan, P. O'Shea, E. Zeek, R. Trebino, and W. S. Warren, "Relativephase ambiguities in measurements of ultrashort pulses with well-separated multiple frequency components," *J. Opt. Soc. Amer. B* *20*, 2226-2237 **2003**.
- [48] E. Zeek, A. P. Shreenath, M. Kimmel, and R. Trebino, Simultaneous Automatic Calibration and Direction-of-Time-Ambiguity Removal in Frequency-Resolved Optical Gating,*Appl. Phys. B* *B74*, S265-271 **2002**.
- [49] L. Xu, E. Zeek, and R. Trebino, Simulations of Frequency-Resolved Optical Gating for measuring very complex pulses, *J. Opt. Soc. Am. B* *25*, A70-A80, **2008**
- [50] Alessia Pasquazi, Marco Peccianti, Yongwoo Park, Brent E. Little, Sai T. Chu, Roberto Morandotti, José Azaña and David J. Moss, "Sub-picosecond phase-sensitive optical pulse characterization on a chip", *Nature Photonics*, v. 5 ,p. 618623 (2011).
- [51] K.W. DeLong, R. Trebino, J. Hunter, and W.E. White, "Frequency-Resolved Optical Gating Using Second-Harmonic Generation," *J. Opt. Soc. Am. B*, vol. 11, pp. 2206-2215 (1994).
- [52] Ziyang Thesis *Ziyang Thesis*

Appendices

Annexe A
MATLAB SCRIPT FOR FROG IMAGE ACQUISITION, USING
LINEAR STAGE AND ORIEL SPECTROMETER SYNCHRONIZED

The image acquisition is one of the most important parts of the FROG process. We have two optical fibers one in front of the other, separated by a distance Δd that we can change using the present MATLAB script. The present script manage the camera and the linear stage as well, making them to work synchronised. In this process we need to acquire images of the spectrum coming from the SHG crystal for different Δd or DELAYS between the output and input fibers. We collect around 100 different spectrum for 100 different DELAY, the total amount of channels of frequencies measured are 1024. We see this in the figures 4-1, 4-2 and 4-3 images of 1024×100 pixels.

SCRIPT :

```
function frog;  
global runflag;  
global backaction;
```

```

% close camera
pvcamclose(0);

% open camera
disp('Opening camera');
h_cam = pvcamopen(0);
x_size = pvcamgetvalue(h_cam, 'PARAM_SER_SIZE');

% obtain size of serial register
y_size = pvcamgetvalue(h_cam, 'PARAM_PAR_SIZE');

% obtain size of parallel register
smode=0;          %enable the simulation model
sst=40123456;    %serial of the simulated stage

%I create a dummy spectrum for the simulation mode
if smode==1
    xxd=[1:1024];
    yyd=[1:124];
    [XXD,YYD]=meshgrid(xxd,yyd);
    w0=20;
    g=exp((- (XXD-512).^2)/w0^2);

```

```

end

predelay=0.01;
c=3E8;

%scanning range (pS) DELAY
Trange=[-1.5 2];

%Number of different DELAYS
Npoints=100;

%time axis
taxis=linspace(Trange(1),Trange(2),Npoints);

%position of the autocorrelation zero (mm)
center=54.6;
xsteps=center+3E-1*taxis;%position axis

dt=taxis(2)-taxis(1);

if exist('fig')==0

```

```

%init the motor interface

format long

%serial

%serialh1=90819137

if smode==1
    serialh1=sst;
else
    % this is the serial of the interface
    serialh1=90808754
end

fig = figure('Position', [100 70 800 500], ...
    'HandleVisibility', 'on', 'IntegerHandle', 'off', ...
    'Name', 'AC Interface', 'NumberTitle', 'off');

set(fig, 'Name', ['APT Interface, Handle Number '...
    num2str(fig, '%2.20f')'], 'Color', [0.9 0.9 0.9]);

drawnow;

stop_button = uicontrol(gcf, 'Style', 'pushbutton', ...
    'String', 'Stop', 'Interruptible', 'on', 'Callback', ...
    @stop_callback, 'Units', 'pixels', 'Position', [25 400 100 90]);

start_button = uicontrol(gcf, 'Style', 'pushbutton', ...

```

```

    'String', 'Start', 'Interruptible', 'on', 'Callback', ...
    @start_callback, 'Units', 'pixels', 'Position', [150 400 100 90]);
exit_button = uicontrol(gcf, 'Style', 'pushbutton', ...
    'String', 'Exit', 'Interruptible', 'on', 'Callback', ...
    @exit_callback, 'Units', 'pixels', 'Position', [275 400 100 90]);
back_button = uicontrol(gcf, 'Style', 'pushbutton', ...
    'String', 'Ripristinate Last Scan ', 'Interruptible', ...
    'on', 'Callback', @back_callback, 'Units', 'pixels', ...
    'Position', [400 400 150 90]);
text_field = uicontrol(gcf, 'Style', 'text', 'String', ...
    'Ripristinate Last Scan', 'Position', [575 400 150 90]);

ax3=axes('Units', 'pixels', 'box', 'on', 'Visible', 'on', ...
    'Position', [50 280 100 100], 'xtick', [], 'ytick', ...
    [], 'Color', [1 1 1]);
ax2=axes('Units', 'pixels', 'box', 'on', 'Visible', 'on', ...
    'Position', [50 50 300 150], 'xtick', [], 'ytick', [], ...
    'Color', [1 1 1]);
ax4=axes('Units', 'pixels', 'box', 'on', 'Visible', ...
    'on', 'Position', [400 50 300 150], 'xtick', [], ...
    'ytick', [], 'Color', [1 1 1]);
ax5=axes('Units', 'pixels', 'box', 'on', 'Visible', 'on', ...
    'Position', [200 280 500 100], 'xtick', [], 'ytick', [], ...

```

```

        'Color',[1 1 1]);
fig2 = figure('Position', [850 70 400 300],...
        'HandleVisibility', 'on', 'IntegerHandle', 'off', ...
        'Name', 'APT Interface', 'NumberTitle', 'off');

% Consult the functions actxcontrolselect,
%actxcontrollist, methodsview
h1= actxcontrol('MGMOTOR.MGMotorCtrl.1', [0 0 400 300], fig2);
h1.StartCtrl;
set(h1, 'HWSerialNum', serialh1);
h1.Identify
%h1.MoveHome(0,1)

[A1,B1,C1,D1,E1]=h1.GetHomeParams(0,0,0,0,0);
h1.SetHomeParams(A1,B1,C1,2,E1);
%mantain all the default setting end set the homing speed to 2
[A1,B1,C1]=h1.GetJogMode(0,0,0);
h1.SetJogMode(A1,1,C1);
%mantain all the default setting end set the jog mode to continuou

end

%FROG MEMORY

```

```

FMEM=zeros(1024,length(xsteps));
lmb=linspace(500,900,1024);

% this has to be the vector of the wavelength
axes(ax3) title('FROG memory ready');
figure(fig2) figure(fig)
axes(ax2) nsind=1; axes(ax4) plot(xsteps,xsteps.*0);
axes(ax2)
image(taxis,lmb,FMEM/(max(max(FMEM))*256);
colormap(jet(256))

% any translation command is blocking if the lastflag is 1
%   for example
%   h1.MoveAbsoluteEx(0,xsteps(1),0,1); is blocking
%   (the program flows stop here until the action is finished)
%   h1.MoveAbsoluteEx(0,xsteps(1),0,0);
%   is not blocking (the program flows
%   continue while the action is runing)

h1.SetVelParams(0,0,3,2);
colorind=1;
colors='k';
h1.SetVelParams(0,0,3,2)

```

```

runflag=0; %I stop the scan
speed=2; %speed mm/s
%runflag is 0 if you are stopped
%          1 if you want to start
%          -1 if you want to exit

while (runflag>-1) %exit condition
    jj=0;
    pause(0.5);

%CHECKIN THE EXISTENCE OF AN INVALID LAST SCAN
    backaction=0;
    if exist('last_scan_project.mat')==2

%I CHECK IF THE last_scan_project.mat file was created.
        load('last_scan_project.mat');

%carico ultimo namefile e xstep_sav
    end
    if exist(namefile_sav);

        load(namefile_sav,'xpos_sav','FMEM','ind');

```

```

%carico FMEM xpos_sav
    ind_sav=ind;
    Nst=length(xsteps_sav);
    if xpos_sav==xsteps_sav(Nst)
        set(back_button,'enable','off');
        banner=[''];
        set(text_field,'string',banner);
    else
        set(back_button,'enable','on');
        banner=[namefile_sav ' contains '...
            num2str(ind_sav) '/' num2str(Nst) ' valid lines'];
        set(text_field,'string',banner);
    end
end
else
    set(back_button,'enable','off');
end

while runflag==0 %wait for a start even
    pause(0.2);
end
if smode==0

```

```

        load('back_image.mat','backimage2','newscale');

end

if runflag==1 %here you perform the scan

    h1.SetVelParams(0,0,4,4)

    h1.MoveAbsoluteEx(0,xsteps(1),0,1);

    h1.SetVelParams(0,0,4,speed) %speed change

    ind=0;

    if smode==0

%I OPEN THE CAMERA

        banner2=[' -OPENING THE CAMERA-'];

        set(text_field,'string',banner2);

% open camera connection, it needs to be closed after one acquisition

        disp('il valore di h_cam');

%I DEFINE THE SCANNING AREA

        x1=0;

        x2=x_size-1;

```

```

        y1=0;
        y2=y_size-1;

        x_size0=x2-x1+1;
        y_size0=y2-y1+1;
        binx=1;
        biny=4;
        % set the ROI structure to full camera
        %array with full spatial detail (no binning)
        roi_struct = cell2struct({x1, x2, binx,y1, y2, biny},...
            {'s1', 's2', 'sbin', 'p1', 'p2', 'pbin'}, 2);

    end

%STORE THE SCANNING PARAMETERS
    if backaction==0

%DEFINE

        FMEM=rand(1024,length(xsteps));
        CC=clock;

        %I create a name from the computer clock

```

```

namefile=['FROG' num2str(CC(1)) '-' num2str(CC(2)) '-'...
        num2str(CC(3)) '-' num2str(CC(4)) '-' ...
        num2str(CC(5)) '-' num2str(CC(6),'%2.0f') '.mat'];
xsteps_sav=xsteps;
namefile_sav=namefile;
save('last_scan_project.mat','xsteps_sav','namefile_sav');
xsteps_real=xsteps;
banner2=[' -SCANNING- '];
set(text_field,'string',banner2);
else
namefile=namefile_sav;
xsteps=xsteps_sav;
ind=ind_sav;
xsteps_real=xsteps(ind:Nst);
banner2=[banner ' -RESUMING-'];
set(text_field,'string',banner2);
end

for xpos=xsteps_real;

ind=ind+1;
if runflag==0
break

```

```

end

h1.MoveAbsoluteEx(0,xpos,0,1);

if smode==1

    x=g;

%if i'm in the simulation mode I assign a dummy spectrum image
else

    image_data = pvcamacq(h_cam, 1, roi_struct, ...
        1000, 'timed'); %I acquire qn image at 200mS
    photo=reshape(image_data,x_size0/binx,y_size0/biny).';
    simage=double(photo);
    H=size(photo);
    simage_real=simage;
    %WE USE THE SCRIPT IN
    %APPENDIX B TO GET THE IMAGE simage
    for jj=1:H(1)
        simage_real(jj,:)=simage(jj,+)/newscale(jj);
    end
    x=simage_real-backimage2;
    %WE USE HERE THE BACKGROUND IMAGE WE
    %GET FROM SCRIPT IN APPENDIX C.

```

```

end

size(x)
axes(ax5);
pcolor(x);
set(gca,'XLim',[425 575]);
shading interp;

Vs=sum(x(14:15,:),1)/2;

FMEM(:,ind)=Vs';
axes(ax2)
title(' ');
FROGY=FMEM/(max(max(FMEM)))*256;
image(taxis,lmb,FROGY);
colormap(jet(256))

axes(ax4);
title(' ');
FROGY=FMEM/(max(max(FMEM)))*256;
pcolor(FROGY);shading interp;

colormap(jet(256))

```

```

set(gca,'YLim',[425 575],'Ydir','reverse',...
    'Ytick',[],'xtick',[])
drawnow;
xpos_sav=xpos;
ind_sav=ind;
save(namefile,'taxis','lmb','FMEM','xpos_sav',...
    'xpos','ind');
end

runflag=0; %I stop the scan
axes(ax2);
banner=[namefile ' SAVED!'];
title(banner);

end

if colors=='k'
    colors='r'
else
    colors='k'
end

end

```

```
% I CLOSE THE CAMERA
pvcamclose(0);

close all

end

function stop_callback(hObject,eventdata) global runflag;
%I make those variables global, i.e. visible to the callback fuctions

    runflag=0;

end

function start_callback(hObject,eventdata) global runflag;
%I make those variables global, i.e. visible to the callback fuctions

    runflag=1;

end

function exit_callback(hObject,eventdata) global runflag;
%I make those variables global, i.e. visible to the callback fuctions
```

```
runflag=-1;
```

```
end
```

```
function back_callback(hObject,eventdata) global runflag;
```

```
%I make those variables global, i.e. visible to the callback fuctions
```

```
global backaction;
```

```
runflag=1;
```

```
backaction=1;
```

```
end
```

Annexe B
MATLAB PROGRAM FOR CONTINUOUS SPECTRUM
ACQUISITION FROM THE CAMERA

This is an example how we get the spectrum using the PVCAM using a MATLAB script. MATLAB has the possibility to manage different interfaces, CAMERAS for example. Also this script subtract the image of the background we got in the script in annexe C.

SCRIPT :

```
% clear all;
close all; clc pvcamclose(0);
h_cam = pvcamopen(0); % open camera connection

% pvcamclose(h_cam); % close camera
% pvcamsetvalue(h_cam, 'PARAM_EXP_TIME', 100);
% set camera to max readout speed

x_size = pvcamgetvalue(h_cam, 'PARAM_SER_SIZE');
% obtain size of serial register
```

```

y_size = pvcamgetvalue(h_cam, 'PARAM_PAR_SIZE');
% obtain size of parallel register
pvcamsetvalue(h_cam, 'PARAM_SPDTAB_INDEX', 0);
% set camera to max readout speed

% win=8;
% x1=0;
% x2=x_size-1;
% y1=62-win;
% y2=62+win-1;

x1=0;
x2=x_size-1;
y1=0;
y2=y_size-1;

x_size0=x2-x1+1;
y_size0=y2-y1+1; binx=1;
biny=4;

%we call the background image
load('back_image.mat','backimage2','newscale');

```

```

% set the ROI structure to full camera array
%with full spatial detail (no binning)
roi_struct = cell2struct({x1, x2, binx,y1, y2, biny}, {'s1', 's2',
'sbin', 'p1', 'p2', 'pbin'}, 2); figure(100);
    set(gcf,'Position',[100 100 800 600]);
    subplot(2,2,1);
    imagesc(backimage2);
    title('back');
    colorbar

% IMAGE_ACQ (save as separate script file)
% acquire a single image
for jj=1:10000
    tic
    image_data = pvcamacq(h_cam, 1, roi_struct, 600, 'timed');
    % acquire image
    photo=double(reshape(image_data,x_size0/binx,y_size0/biny).');

% for jj=1:N
%     photo(jj,:)=photo(jj,:)/H(jj);
% end

```

```
toc
```

```
subplot(2,2,2);  
    imagesc(photo);  
title('acquired');  
colorbar
```

```
H=size(photo);
```

```
subplot(2,2,3);  
simage=double(photo);
```

```
simage_real=simage;
```

```
for jj=1:H(1)  
    simage_real(jj,:)=simage(jj,:)/newscale(jj);
```

```
end
```

```
plot(newscale,1:H(1));
```

```
colorbar
```

```
title('descaled');
```

```
subplot(2,2,4);
```

```
cimage=simage_real-backimage2;
%subtraction the image of the background
%we got in the script in appendix C.
imagesc(cimage);
title('normalized');
colormap(jet(256));
colorbar
drawnow;
end

pvcamclose(0); % close camerac
```

Annexe C
MATLAB PROGRAM FOR BACKGROUND ACQUISITION

Before we can use a CCD images in our FROG program, we need to get a background image, it means an spectrum image with no laser beam passing thought the setup. We then can subtract that image to the images we get from our setup, figures 4-1, 4-2 and 4-3.

SCRIPT :

```
% clear all;
close all;
clc pvcamclose(0);

% open camera connection
h_cam = pvcamopen(0);

% set camera to max readout speed
x_size = pvcamgetvalue(h_cam, 'PARAM_SER_SIZE');
```

```

% obtain size of serial register
y_size = pvcamgetvalue(h_cam, 'PARAM_PAR_SIZE');

% obtain size of parallel register
pvcamsetvalue(h_cam, 'PARAM_SPDTAB_INDEX', 0);

% set camera to max readout speed
% win=8;
x1=0;
x2=x_size-1;
y1=0;
y2=y_size-1;

x_size0=x2-x1+1; y_size0=y2-y1+1; binx=1; biny=4;

% set the ROI structure to full camera
% array with full spatial detail (no binning)
roi_struct = cell2struct({x1, x2, binx,y1, y2, biny}, {'s1', 's2',
'sbin', 'p1', 'p2', 'pbin'}, 2);

```

```

% IMAGE_ACQ (save as separate script file)

tic

% acquire image
image_data = pvcamacq(h_cam, 1, roi_struct, 1000, 'timed');
photo=double(reshape(image_data,x_size0/binx,y_size0/biny).');
H=max(photo'); N=length(H);

toc figure(100) imagesc(photo); set(gca,'Clim',[70 180]);
colormap(jet(256)); colorbar

% close camera
pvcamclose(0);
backimage=double(photo); H=size(backimage);
scale=sum(backimage')/H(2);

win=5; center_cord=round(H(1)/2);
bottom=sum(scale((center_cord-win):(center_cord+win)))/(2*win+1);

newscale=scale/bottom;
backimage2=backimage;
backimage_orig=backimage;

```

```
for jj=1:H(1)
    backimage2(jj,:)=backimage(jj,:)/newscale(jj);
end

imagesc(backimage2);

save('back_image.mat','backimage_orig','backimage2',...
'newscale','x1','x2','binx','y1','y2','biny');
```

Annexe D
MATLAB PROGRAM FOR CALIBRATION OF THE
SPECTROMETER

Oriel Spectrograph from annexe L, is not calibrated, the image is just intensity and number of pixel. For calibrating the Oriel Spectrograph we need to use OPTICAL SPECTRUM ANALYZER, OSA, from ANNEXE G. We then compare the pixel image from Oriel Spectrograph with spectrum range from OSA. In this way every pixel from Oriel will have a frequency representation. And this script build this representation.

SCRIPT :

```
close all clear all clc c=299792458;

%time in fs e frequency in THz
nomeF1='spectrum_text_file_OSA_1';
nomeF2='spectrum_text_file_OSA_2';
nomeF3='spectrum_text_file_OSA_3';
center=[0 0 0];
```

```

center_la=[0 0 0];
figure;
col=jet(3);
xx=1:1024;
for num=1:3

    nome0=['W000' num2str(num) '.TXT'];
    fid=fopen(nome0);
    lettura=textscan(fid,'%n%n%*[\n]', 'delimiter', ',', 'headerlines', 3);
    fclose(fid);
    la=lettura{1};
    da=lettura{2};
    da=da-min(da);
    da=da/max(da);
    omega_frog=-(2*pi*c./(la*1e-9))*1e-12;
    switch(num)
        case (1)
            nome=nomeF1;
            center_la(num)=sum(la.*da)/sum(da);
        case (2)
            nome=nomeF2;
            center_la(num)=sum(la.*da)/sum(da);
        case (3)

```

```

        nome=nomeF3;
        center_la(num)=sum(la(1:300).*da(1:300))/sum(da(1:300));
    end

    LL=load(nome);
    a=sum(LL.FMEM,2).';
    a=a/max(a);
    a=a.*(a>.1);
    a(550:1024)=0;
    center(num)=sum(xx.*a)/sum(a);
    mat(num,:)=a;

end

center_sh=center_la/2;
omega_sh=c./(center_sh*1e-9)*1e-12;
cfun= fit(center_sh.',center.', 'poly1');
cfun2 =fit(omega_sh.',center.', 'poly1');
plot(center_sh,center,'o') hold on
xlabel('\lambda [nm]') ylabel('pixel')
plot(center_sh,center_sh*cfun.p1+cfun.p2,'r')

```

Annexe E
MATLAB PROGRAM FOR CUT AND INTERPOLATE FROG
EXPERIMENTAL IMAGE

The image frog retrieved must be a square matrix, with compatible frequency and temporal axis. Defined the dimension of the matrix ($N = 2^{ord}$) and the temporal window the program cuts and interpolate properly the CCD experimental matrix.

SPECTROMETER PROPERTY.

The spectrum is collected with a CCD camera with 1024 pixel.

Interpolation function for the calibration :

calibration in wavelength.

x : pixel

y : wavelength [nm] $y = p1*x + p2$

Coefficients (with 95% confidence bounds) :

$p1 = 0.1555$ (0.1321, 0.1789)

$p2 = 697.4$ (685.9, 709)

calibration in frequency.

x : pixel

y : frequency [THz]

$y = p1*x + p2$

Coefficients (with 95% confidence bounds) :

p1 = -0.07779 (-0.0907, -0.06487)

p2 = 425.6 (419.2, 432)

we use the value 0.07779 THz per pixel

SCRIPT :.

```
close all clear all clc.
```

```
%FILE NAMES
```

```
name_frog_exp='input_file.mat';
```

```
%name of the measured file
```

```
name_frog_interp='output_file.mat';
```

```
% name of the output file (to be used in the program for the pulse retrieval
```

```
c=299792458;          %speed of light
```

```
%SPECTROMETER PROPERTY: the spectrum is collected with a CCD camera with
```

```
%1024 pixel. interpolation function for the calibration:
```

```
% %calibration in wavelength:
```

```
%          x: pixel
```

```
%          y: wavelength [nm]
```

```

%      y = p1*x + p2
%      Coefficients (with 95% confidence bounds):
%      p1 =      0.1555 (0.1321, 0.1789)
%      p2 =      697.4 (685.9, 709)
%calibration in frequency:
%      x: pixel
%      y: frequency [THz]
%      y = p1*x + p2
%      Coefficients (with 95% confidence bounds):
%      p1 =     -0.07779 (-0.0907, -0.06487)
%      p2 =      425.6 (419.2, 432)
%we use the value 0.07779 THz per pixel
cal_f=-0.07779*1e12;      %frequency/ pixel
zero_f=425.6*1e12;      %frequency at the pixel 1
cal_l= 0.1555*1e-9;      %wavelength/ pixel
zero_l= 697.4*1e-9;      %wavelength at the pixel 1
pixel_spectrum=1:1024;      %pixel axis for frequency staring from 1

%DEFINE THE AXIS FOR THE OUTPUT MATRIX
ord=7;      % matrix dimensions: 2^ord X 2^ord
N=2^ord+1;      %matrix dimension
df=abs(cal_f)/2;      %angular frequency step
ax=(-2^(ord-1)):1:(2^(ord-1));%generic axis with dim 2^ord centered in zero

```

```

time_window=1/df;      %time window of the output matrix
dt=time_window/N;     %time step
t=ax*dt;              %temporal axis
frequency_window=1/dt; %angular frequency window of the output matrix
f=ax*df;              %angular frequency axis

%LOAD the experimental file
LL=load(name_frog_exp);

%LOAD experimental time axis
time=(LL.taxis*1e-12); %experimental temporal axis
dt_exp=time(2)-time(1); %experimental time step
pixel_time=1:length(time); %pixel axis for time staring from 1

%LOAD and clean the experimental matrix
frog_exp=(LL.FMEM); %experimental matrix
bk=sum(frog_exp(:,1:5),2)/5; %background (in spectrum)
%clean the experimental matrix from noise (in spectrum dimension)
for num=1:size(frog_exp,2)
    frog_exp(:,num)=frog_exp(:,num)-bk*1.0;
end frog_exp=(frog_exp.*(frog_exp>0));
frog_exp=frog_exp/max(max(frog_exp)); ff=fft2(frog_exp); win_s=500;
win_t=floor(length(time)/4); ff(win_s:size(ff,1)-win_s,:)=0;

```

```

ff(:,win_t:size(ff,2)-win_t)=0; frog_exp=real(iff2(ff));
%CALCULATE THE NUMBER OF PIXEL NECESSARY TO COVER THE SPECTRAL-TEMPORAL
%WINDOW
num_pixel_f=floor(frequency_window/abs(cal_f))+4;
%number of pixel needed to cover the spectral window
f_exp=(-floor(num_pixel_f/2):floor(num_pixel_f/2))*cal_f;
%pixel axis for frequency centered in zero and matching num_pixel_f
num_pixel_t=floor(time_window/dt_exp)+4;
%number of pixel needed to cover the temporal window
t_exp=(-floor(num_pixel_t/2):(floor(num_pixel_t/2)))*dt_exp;
%pixel axis for time centered in zero and matching num_pixel_t

%CUT THE MATRIX IN SPECTRUM AXIS
%find the central pixel
int=sum(frog_exp,2).'; int=int/max(int); int=int.*(int>0.5);
center_pixel_spectrum=round(sum(int.*pixel_spectrum)/sum(int));
%calculate the central frequency (needed for the method and back converted
%form the sh to ff)
central_frequency=(zero_f+cal_f*center_pixel_spectrum)/2;
central_wavelength=(zero_l+cal_l*center_pixel_spectrum)*2;
%put to zero some lines
n_clean=25; frog_exp(1:center_pixel_spectrum-n_clean,:)=0;
frog_exp(center_pixel_spectrum+n_clean:size(frog_exp,1),:)=0; if

```

```

length(pixel_spectrum)>length(f_exp)
    frog_exp=frog_exp(center_pixel_spectrum-floor(num_pixel_f/2)...
    :center_pixel_spectrum+floor(num_pixel_f/2),:); % cut the matrix
else
    dim1=floor(length(f_exp)/2-center_pixel_spectrum);
    dim2=floor(length(f_exp)/2-(size(frog_exp,1)-center_pixel_spectrum))+1;
    zeros1=zeros(dim1,size(frog_exp,2));
    zeros2=zeros(dim2,size(frog_exp,2));
    frog_exp=[zeros1.' frog_exp.' zeros2.'].';
end
%CUT THE MATRIX IN TIME AXIS
%find the central pixel
int=sum(frog_exp,1); int=int/max(int); int=int.*(int>0.5);
center_pixel_time=round(sum(int.*pixel_time)/sum(int)); if
length(pixel_time)>length(t_exp)
    frog_exp=frog_exp(:,center_pixel_time-floor(num_pixel_t/2)...
    :center_pixel_time+floor(num_pixel_t/2));% cut the matrix
else
    dim1=floor(length(t_exp)/2-center_pixel_time);
    dim2=floor(length(t_exp)/2-(size(frog_exp,2)-center_pixel_time))+1;
    zeros1=zeros(size(frog_exp,1),dim1);
    zeros2=zeros(size(frog_exp,1),dim2);
    frog_exp=[zeros1 frog_exp zeros2];

```

```

end

%INTERPOLATE THE MATRIX TO OBTAIN THE OUTPUT MATRIX

frog_exp=frog_exp/max(max(frog_exp)); frog_exp=frog_exp-.01;
frog_exp=frog_exp/max(max(frog_exp));
frog_exp=((frog_exp.*(frog_exp>0)));

FROG=interp2(t_exp,f_exp.',frog_exp,t,f.');
```

```

for num1=1:size(FROG,1)
    for num2=1:size(FROG,2)
        if isnan(FROG(num1,num2))
            FROG(num1,num2)=0;
        end
    end
end
end

FROG_AUX=FROG(:,size(FROG,2):-1:1); FROG=FROG+FROG_AUX;
FROG=FROG/max(max(FROG)); figure; subplot(1,2,1) imagesc(t,f,(FROG))
title('INTERPOLATE') subplot(1,2,2) imagesc(t_exp,f_exp,(frog_exp))
title('EXPERIMENTAL')
% figure
% imagesc(LL.FMEM)
% hold on

```

```
% plot(center_pixel_time,center_pixel_spectrum,'wo')
time_array=t*1e15; div=N; frequency_array=f*1e-12; best_trace=FROG;
central_frequency=c/1550e-9;
save(name_frog_interp,'time_array','frequency_array',...
'best_trace','div','central_frequency','df','dt')
```

Annexe F
BENCHTOP DC SERVO MOTOR CONTROLLERS

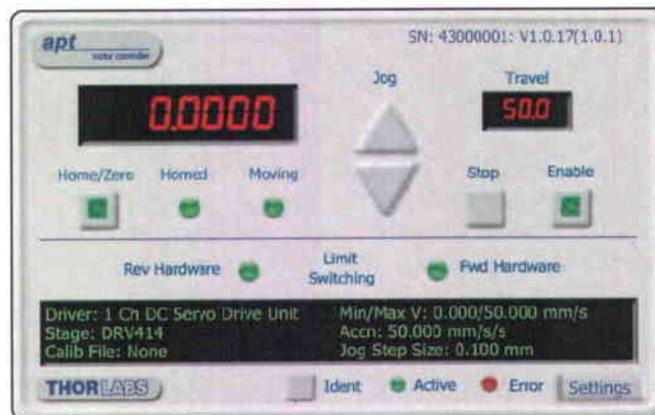


FIGURE F-1 – Benchtop DC Servo Motor Controllers. Contrôleurs de moteur servo Benchtop DC.

Specifications

Drive Connector: 15-Pin D-Type Female
(Drive Outputs, QEP Inputs, Limit
Switch Input)

Motor Output Power:
Up to 48 V/50 W(Peak)

Output Type: 10-bit Sign/Magnitude PWM

Operating Modes: Position, Velocity

Control Algorithm: Digital PID Filter,
(16 Bit)

Velocity Profile: Trapezoidal

Position Count: 32-Bit

Position Feedback: Incremental Encoder -
Differential QEP Inputs

Encoder Feedback Bandwidth:
500,000 Counts/Second

Motor Speeds: Up to 6000 RPM
(for 4096 Count Encoder)

Encoder Supply: 5 V

User Control Connector: 15-Pin D-Type
Female (Jog Inputs, Enable Interlock,
RS-232, Trigger In/Out (TTL), Brake Out,
Safety Stop, User Digital I/O, 0 to 10 V
Analog Input)

Input Power Requirements

Volts: 85 to 264 VAC

Power:
BDC102, BDC103: 200 W
BDC101: 100 W

Fuse: 3.15 A

Dimensions:
BDC101: 6.0" x 9.6" x 4.1"
(152 mm x 244 mm x 104 mm)
BDC102, BDC103: 9.5" x 14.2" x 5.2"
(240 mm x 360 mm x 133 mm)

Weight:
BDC101: 7 lbs (3.18kg)
BDC102, BDC103: 14.75 lbs (6.7 kg)

FIGURE F-2 – Specifications for Benchtop DC Servo Motor Controllers. Car-
actéristiques pour des contrôleurs de moteur servo Benchtop DC.

Annexe G
OPTICAL SPECTRUM ANALYZER AQ6317B



FIGURE G-1 – Spectrum Analyzer. Analyseur de spectre.

Applicable fibers	SM, GI (50/125 μ m)
Measurement wavelength range ¹	600 to 1750 nm
Wavelength accuracy ^{1,3}	± 0.02 nm (1520 to 1580 nm, after calibration with build-in reference light source) ± 0.04 nm (1580 to 1620 nm, after calibration with build-in reference light source) ± 0.5 nm (600 to 1750 nm)
Wavelength linearity ^{1,3}	± 0.01 nm (1520 to 1580 nm) ± 0.02 nm (1580 to 1620 nm)
Wavelength repeatability ^{1,3}	± 0.005 nm (1 min)
Wavelength resolution ^{1,3}	Max. resolution: 0.015 nm or better (1520 to 1620 nm, RESOLN: 0.01 nm) Resolution setting: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0 nm
Resolution accuracy ^{1,3}	$\pm 5\%$ (1300 to 1650 nm, RESOLN: 0.05 nm or more, resolution correction: ON)
Measurement level range ^{2,3}	-90 to +20 dBm (1200 to 1650 nm, SENS: HIGH 3) -80 to +20 dBm (1000 to 1200 nm, SENS: HIGH 3) -60 to +20 dBm (600 to 1000 nm, SENS: HIGH 3)
Level accuracy ^{2,3,4}	± 0.3 dB (1310/1550 nm, input level: -30 dBm, SENS: HIGH 1 to 3)
Level linearity ^{2,3}	± 0.05 dB (input: -50 to +10 dBm, SENS: HIGH 1 to 3)
Level flatness ^{2,3}	± 0.1 dB (1520 to 1580 nm), ± 0.2 dB (1580 to 1620 nm)
Polarization dependency ^{2,3}	± 0.05 dB (1550/1600 nm), ± 0.05 dB typ. (1310 nm)
Dynamic range ⁴	60 dB (1523 nm, peak ± 0.2 nm, resolution: 0.01 nm) 70 dB (1523 nm, peak ± 0.4 nm, resolution: 0.01 nm) 45 dB (1523 nm, peak ± 0.2 nm, resolution: 0.1 nm)
Sweep time	Approx. 500 ms (SPAN: 100 nm or less, SENS: NORM HOLD, AVR: 1, SMPL: 501, resolution correction: OFF) Approx. 0.5 min (SPAN: 100 nm or less, SENS: HIGH 2, AVR: 1, SMPL: 501, No signal)
Function	Automatic measurement Long term measurement function
Setting of measuring conditions	Span setting: 0 to 1200 nm Measuring sensitivity setting: NORMAL, HOLD/AUTO, MID, HIGH 1/2/3 Number of averaging setting: 1 to 1000 times Sample number setting: 11 to 20001, AUTO Automatic setting function of measuring conditions Sweep between-marker function 0 nm sweep function Pulse light measurement function Air/vacuum wavelength measurement function TLS synchronized measurement function

Function	Trace display	Level scale setting: 0.1 to 10 dB/div, linear Simultaneous display of 3 independent traces Max./Min. hold display Roll averaging display Calculation between-traces display Normalized display Curve-fit display 3D display Split display Power density display, % display, dB/km display Frequency display of horizontal axis scale
	Data analysis	WDM waveform analysis (Wavelength/Level/SNR list display) Optical fiber amplifier analysis (GAIN/NF, Single/Multi channel) PMD analysis, Optical filter analysis, DFB-LD analysis FP-LD analysis, LED analysis, SMSR analysis Peak search, bottom search, spectral width search, notch width search Delta marker (max. 200), line marker (analysis range specification) Graph display of long-term measurement result
	Others	Self-wavelength calibration function (using built-in reference light source)
Memory	Build-in FDD	3.5-inch 2HD
	Internal memory	32 traces, 20 programs
	File format	Trace file, program file, measuring condition file Text file (trace, analysis data, etc.) Graphics file (BMP, TIFF)
Data output	Printer	Built-in high speed thermal printer
Interface	Remote control	GP-IB (2 ports) TLS control interfaces (TTL)
	Others	Sweep trigger input (TTL) Sample enable input (TTL) Sample trigger input (TTL) Analog output (0 to 5 V) Video output (VGA)
Display	9.4-inch color LCD (Resolution: 640 x 480 dots)	
Optical connector	FC (Standard)	
Power requirement	AC 100 to 120/200 to 240 V, 50/60 Hz, approx. 200 VA	
Environmental conditions	Operating temperature: 5 to 40 °C Storage temperature: -10 to +50 °C Humidity: 80 %RH or less (No condensation)	
Dimensions and mass ⁵	Approx. 425 (W) x 222 (H) x 450 (D) mm, approx. 30 kg	
Accessories	Power cord: 1, printer paper roll: 2, floppy disc: 2, instruction manual: 1	

FIGURE G-2 – Specifications for Spectrum Analyzer. Caractéristiques pour l'analyseur de spectre.

Annexe H
OPTICAL FIBER AMPLIFIERS



FIGURE H-1 – Fiber Amplifier. Amplificateur de fibre.

Model	HPPFA-XX	
Saturated output power	18 dBm for HPPFA-18 20 dBm for HPPFA-20	
Peak Power (before pulse breakup)	>10,000 W into free space output	>5,000 W Fiber coupled output
Peak Power (Minimal SPM)	> 2,000 W into free space output	>1,000 W Fiber coupled output
Small signal gain	>27 dB	
Optical noise figure	< 6 dB	
Input power range	-20 to +13 dBm	
Wavelength range	1527–1560 nm	
Polarization sensitivity	< 0.3 dB	
Dimensions	10 cm x 26 cm x 28 cm	
Optical		
Gain medium	Er co-doped silica fiber	
Pump source	Diode laser	
Input	FC/APC (single mode fiber, isolated)	
Output	Free space, no isolation	SMF-28 fiber, with output isolator
Environmental		
Operating temperature	+15 to 30°C	
Storage temperature	-20 to 50°C	
Electrical/ Mechanical		
Operating Voltage	85-264 VAC at 47-63 Hz	
Power consumption	<125 W	

FIGURE H-2 – Specifications for Fiber Amplifier. Caractéristiques pour l'amplificateur de fibre.

Annexe I
PICOSECOND and FEMTOSECOND FIBER LASERS



FIGURE I-1 – Fiber Laser . Laser de fibre.

FFL			
Pulse repetition rate	5-100 MHz, fixed		
Average output power	Varies with pulsewidth and pulse repetition rate (e.g., >4 mW at 2 ps and 20 MHz)		
Timing jitter*	<1 ps		
Pulsewidths available**	Spectral width (nm)	Tunable wavelength (nm)	Peak Power (W)
<10 ps	>0.3	1530-1560	>50
<5 ps	>0.4	1530-1560	>50
<2 ps	>1.0	1530-1560	>100
<600 fs	>3.5	1530-1560	>100
<100 fs	>25	Fixed wavelength	>100
Optical			
Gain medium	Er-doped silica fiber		
Pump source	980 nm diode laser		
Connectors	FC/APC (other connectors available on request)		
Environmental			
Operating temperature	+15 to 30 °C		
Storage temperature	-20 to 50 °C		
Electrical/ Mechanical			
Operating Voltage	85-264 VAC at 47-63 Hz		
Power consumption	<125 W		
Dimensions (2U)	10 cm x 26 cm x 36 cm		
Weight	9 kg		

FIGURE I-2 – Specifications for Fiber Laser. Caractéristiques pour le laser de fibre.

Annexe J

The Matlab program FROG 1.0

The Matlab program FROG 1.0¹ can be used to retrieve the amplitude and phase of a signal from its trace. A trace is a two-dimensional matrix containing intensity points of a time-frequency coordinate. The time coordinates is on the horizontal axis and the frequency coordinates on the vertical axis. This trace can be produced from a real experiment in the lab or created synthetically in the program. The various algorithms used assume that the experiment is done for second harmonic generation pulses (SHG) that are determined by the type of non-linear interaction in the crystal and the geometry of the apparatus. The program was completed by **Tommy Payette** during the summer 2006 under the supervision of professor José Azaña of INRS-EMT in Montreal based mainly on the algorithms from the book Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses[8][51]² by Rick Trebino³ . The version of Matlab that was used to run this program is 7.0.0.19920 (*R14*). If

-
1. ©Droits réservés à **Tommy Payette**, 2006
 2. Trebino, Rick, Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulse, Massachusetts, 2002.
 3. Rick Trebino Professor Georgia Research Alliance-Eminent Scholar Chair of Ultrafast Optical Physics

you have any problems or concerns about this program or if you find errors, you can contact him at tommy.payette@gmail.com.

J.1 INSTALLATION

To install the FROG program, unzip the file and copy the directory *FROG1.0* at the location you want in the computer. Open Matlab, and change the current directory for the location where you saved the FROG directory or use the command path to include the location into the current search path as follows :

```
>> path(path,'FullPathToTheFile');
```

After having done the later method, it will be possible to call the FROG program from any directory. To start the FROG program use the command *frog* in the command line :

```
>> frog
```

J.2 Parameters Settings

The first window that the user sees in the Frog program is the settings panel.

Figure J-1

Note that the actual look of the parameter and trace can change depending of the actual default values.

Through the panel on the left, the user can select different settings.

J.2.1 Type of Experiment

Using the pop-up menu, the user can select between two types of source for experiment input. Figure J-2

-THEORY

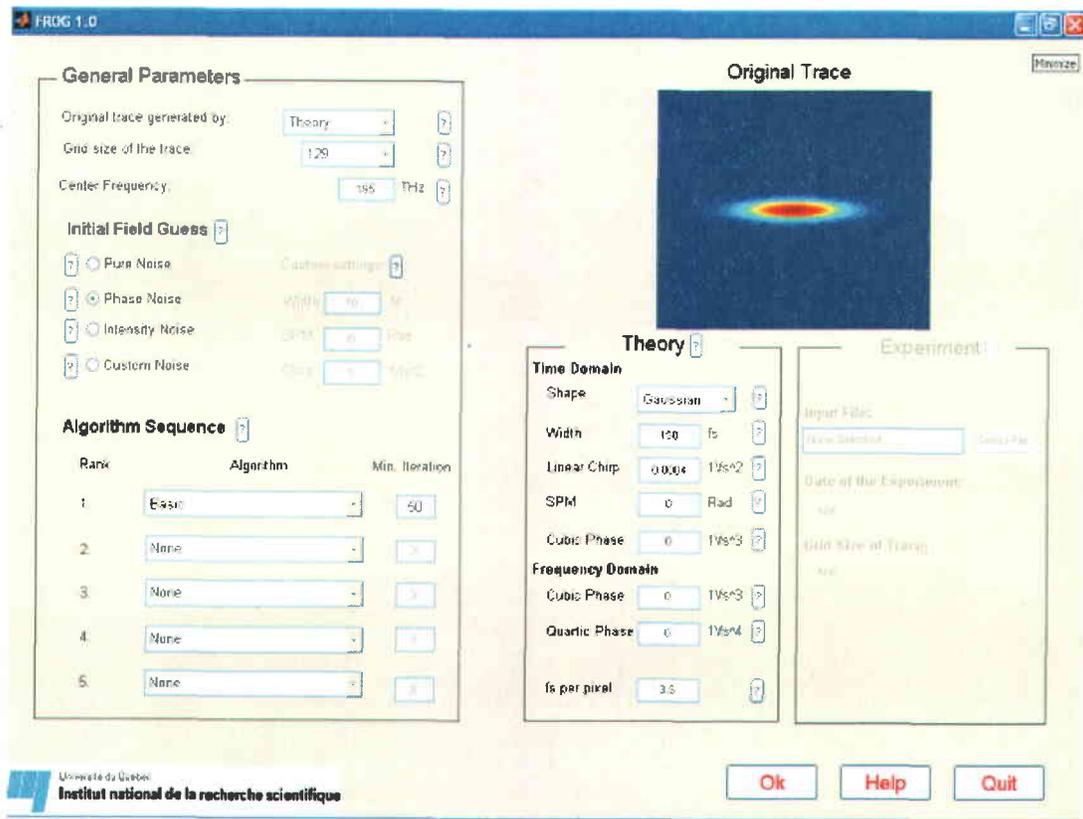


FIGURE J-1 – View of the setting panel when the user starts the program. Vue du panneau quand l'utilisateur commence le programme

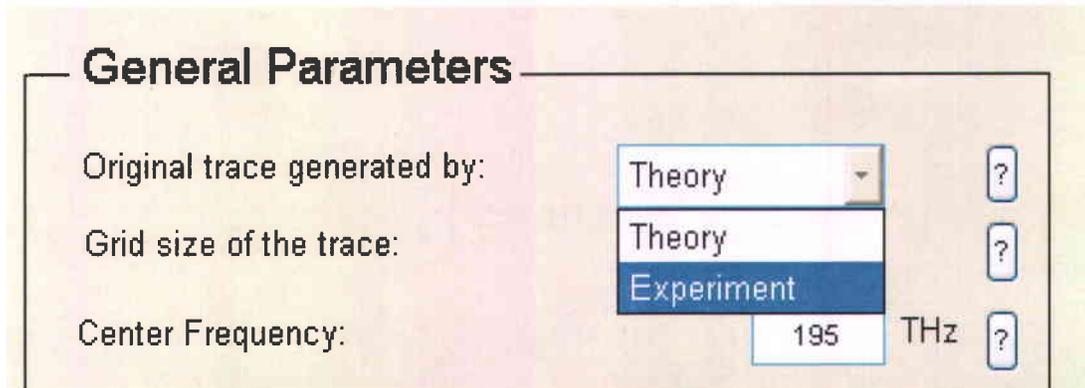


FIGURE J-2 – Type of Experiment. Type d'expérience

If the user chooses Theory, a panel under the trace will be enabled to select parameters to in order to create a theoretical field. Figure J-3

On this panel, the user can select many parameters like the shape, the width, the linear chirp, the self-phase modulation (SPM), the cubic phase (in both domain) and the quartic phase in the frequency domain. Also the femtosecond per pixel can be set and used to fit the trace into the entire window trace. Note that if this parameter is not set correctly the trace will not be seen correctly and the retrieval will fail. For the shape, there is two possibility : Gaussian or sech^2 . Figure J-4

The width is the full with at half maximum (FWHM) of the field. The linear chirp (chirp) is the quadratic factor of the phase. The self-phase modulation (SPM) adds a phase proportional to the intensity. The time cubic phase (TCP), the frequency cubic phase (SCP) and frequency quartic phase (SQP) are defined by their names.

Theory ?

Time Domain

Shape ?

Width fs ?

Linear Chirp $1/\text{fs}^2$?

SPM Rad ?

Cubic Phase $1/\text{fs}^3$?

Frequency Domain

Cubic Phase $1/\text{fs}^3$?

Quartic Phase $1/\text{fs}^4$?

fs per pixel ?

FIGURE J-3 – Theory Parameter Panel. Panneau de paramètre de théorie

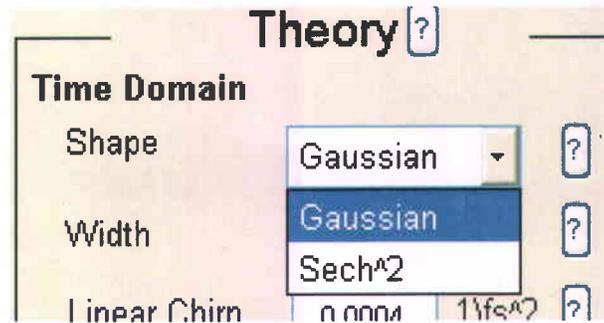


FIGURE J-4 – Theoretical Shape Selection. Choix théorique de la forme.

Based on the selection of those parameters, the field will be created similarly to the definition in the FROG program help file by Femptosoft Technologies⁴ demo version 3.08a :

$$E(t) = A(t) \exp[i(\text{chirp} \cdot t^2 + SPM | A(t) |^2 + TCP \cdot t^3)] \quad (\text{J.1})$$

where t represents the time and $A(t)$ is defined by the shape selected as :

$$A(t) = \exp[-4 \log_2(2) \left(\frac{t}{\text{width}}\right)^2] \quad (\text{J.2})$$

for gaussian pulse and :

$$A(t) = \text{sech}^2\left[1.76 \frac{t}{\text{width}}\right] \quad (\text{J.3})$$

for sech^2 pulses.

4. Femptosoft Technologies, <http://www.femptosoft.biz>

After the field $E(t)$ is created as above, the field is Fourier transformed in the frequency domain and other phase distortion are added.

$$E_{new}(\varpi) = E_{old}(\varpi) \exp[i(SCP \cdot t^2 + SQP \cdot t^4)] \quad (J.4)$$

where $E_{old}(\varpi)$, is the field in equation 2.1. Finally, the new field is Fourier transformed back to the time domain and this results in the theoretical pulse.

The femtosecond per second has to be set correctly by a trial and error until the entire trace is present in the window and at the user will.

-EXPERIMENTAL

If the user chooses Experiment, a panel under the trace will be enabled to select parameters to in order to create a trace to be loaded from a file. Figure J-5

The user will use the Select File button to locate a valid experiment to be loaded. A valid file should be a Matlab® .m file with parameter needed saved. Those parameters are the time array, the frequency array with the trace. They should be saved respectively in the Matlab variable *time_array*, *frequency_array* and *best_trace*. Also an optional parameter can be defined : the date in the variable *experiment_date*. But this last parameter is not necessary to the retrieval.

If the date is the experiment is defined, when the user select the file it will appear on the panel. The grid size of the arrays is also displayed for information.

J.2.2 Grid Size

This represents the number of point in the trace in the horizontal and vertical axis. From the pop-up menu, the user can choose from different size from 17 to 2049 pixel points per line. Figure J-6

Experiment [?]

Input File:
None Selected

Date of the Experiment:
N/A

Grid Size of Trace:
N/A

FIGURE J-5 – Experiment Parameter Panel. Panneau de paramètre de la expérience.

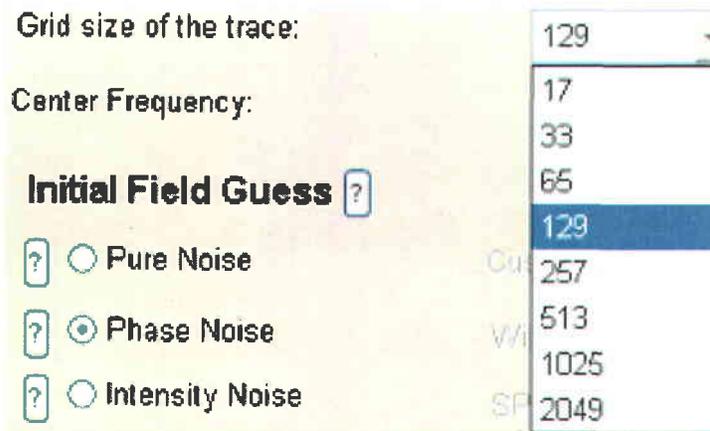


FIGURE J-6 – Grid Size of the Trace. Taille de grille de la trace

The more points N the trace has, the more definition the retrieval has. But as the trace gets more points, the retrieval process becomes slower. The help manual of the demo version indicates the speed of the algorithm decreases as $N \cdot N \cdot \ln(N)$, so that doubling the number of points reduces the speed of the algorithm by more than a factor of 4.2 Usually, 65 or 129 is enough for a good retrieval.

J.2.3 Center Frequency

The center frequency (in THz) is the frequency at which the frequency domain field amplitude and phase will be centered. This parameter is not used if the trace comes from a file (Experiment type).

J.2.4 Initial Guess of the field

Every iterative algorithm has to start with an initial guess for the field, Figure J-7. In FROG, there are four possibilities :

Pure noise is a field where real and imaginary parts of the initial field are generated randomly. Phase noise is a gaussian distributed function with the grid size

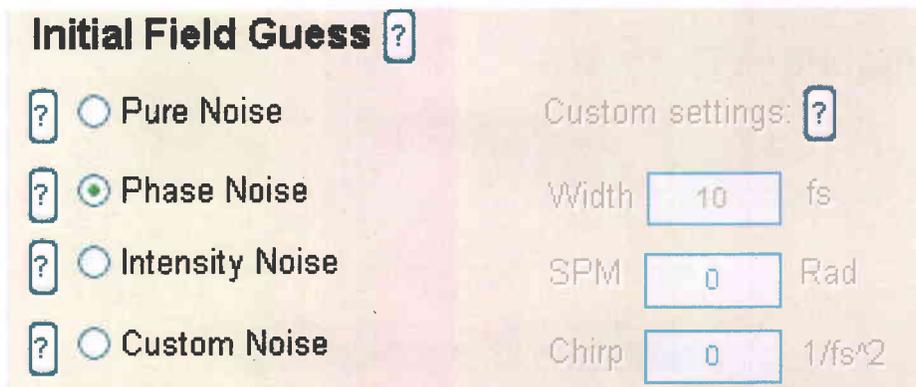


FIGURE J-7 – Initial Guess of the Field. Conjecture initiale du champ.

as the *FWHM* and with random phase. For example if the grid is set to 65, the *FWHM* of the pulse will be 8 (rounded down). Intensity noise is a field with no phase and a random intensity. When custom noise is selected, the custom settings options become enabled and the user can choose the width the self-phase modulation and linear chirp of the gaussian pulse as the initial field.

The user manual of the Femtosoft program states that :

*“In most cases, the selection of the initial field will not have a large effect in how the pulse is retrieved or in the final answer. On rare occasions, however, we have noticed an effect for SHG FROG. There are certain pulses (even theoretically generated ones) that appear to stagnate for certain initial guesses, yet retrieve with ease for other initial guesses.”*⁵

5. Femtosoft Technologies, <http://www.femtosoft.biz>

Algorithm Sequence ?

Rank	Algorithm	Min. Iteration
1.	Basic	50
2.	Basic	X
	Overcorrection	X
	Generalized Projections	X
3.	None	X
4.	None	X
5.	None	X

FIGURE J-8 – Algorithm Sequence Panel. Panneau d’algorithme.

Therefore the user should keep in mind a dependency on the initial field for the speed and results of the retrieval.

J.2.5 Algorithm Sequence

The user can choose from up to three different algorithm for retrieval : the basic (“Vanilla”) algorithm, the overcorrection algorithm and the generalized projection.

Figure J-8

The panel selection indicates a list of algorithm in order by rank that the user can choose. Each algorithm is also associated with a value for the minimum number

of iteration. Once the user has selected a list of algorithm and minimum number of iteration N for each of them, the retrieval process will use this as follow :

- i The first algorithm is chosen and at least N iteration using this algorithm is done.
- ii If the *rms* error has not decreased by at least 0.5% from the tenth iteration before, the retrieval will still use this method until this condition is not met.
- iii In the case, the condition is met, that is the *rms* has not decrease by 0.5%, the second algorithm will be used and the same analyse will be done.
- iv Once all five algorithms from the list are completed, there is a loop to the first one and the process will continue at the user needs.

J.3 FROG Algorithms

There are three available algorithm choices in the FROG program : the basic ("Vanilla") algorithm, the overcorrection algorithm and the generalized projection.

J.3.1 Basic "Vanilla" FROG

This algorithm was adapted from the FROG book⁶ from Rick Trebino and the JOSA paper from Delong and Trebino⁷.

6. Trebino, Rick, Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulse, Massachusetts, 2002.

7. Delong Kenneth W., Trebino Rick, Improved ultrashort pulse-retrieval algorithm for frequency-resolved optical gating, Journal of Optical Society of America, Vol 11, No 9/September 1994.

J.3.2 Overcorrection

From the same reference above, this algorithm is essentially the same as the basic algorithm except that it corrects the field in order to speed up the convergence. A parameter is set (usually between 1.0 and 1.5) to speed up the convergence. This parameter is set by default in the *default_option* file.

J.3.3 Generalized Projection

From the same reference again and with the help of the PhD thesis of Wang Ziyang the method of generalized projection was implemented.

J.4 Retrieval Process

After all parameters have been set correctly and the original trace to be retrieved is known, the user press the Ok button and the retrieval window appear. Figure J-9

J.4.1 Trace Windows

Part of the retrieval window is the original trace view (top left), that is the trace from which the amplitude and phase has to be retrieved. Also the retrieved trace view (bottom left) represent the current iteration of the trace updated at each five iterations.

J.4.2 Amplitude and Phase Windows

In the top center, there are two other windows that represent the amplitude and phase retrieved. Depending on the type of experiment, there will be one or two curves for each graph. In the experimental type, only one curve per windows since the original function is not known. In the theoretical experiment, two curves are present, blue and red, for the original and retrieved function respectively. Note that the amplitude is the normalized amplitude and that the phase is shown using

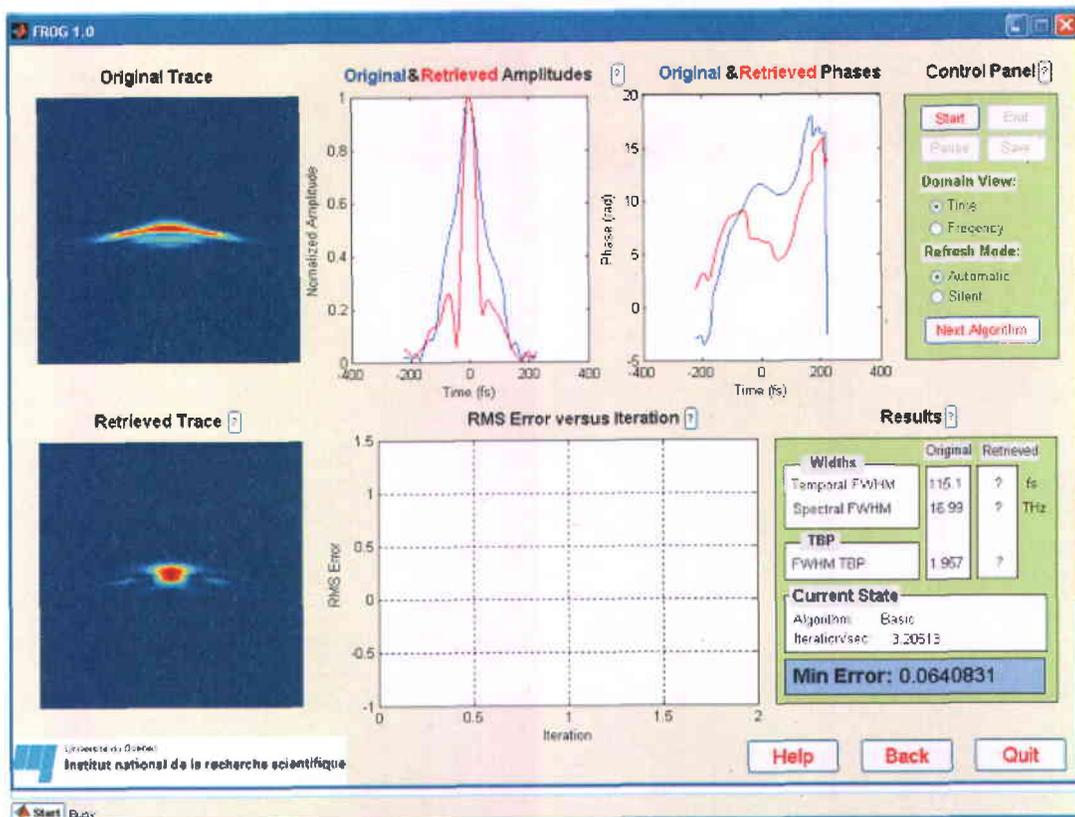


FIGURE J-9 – Retrieval process window. Fenêtre de processus de récupération.

the unwrap Matlab® command. These windows indicated by the default the time domain amplitude and phase but using the control panel at the top right, the user can press the appropriate 'Domain View' radio button to alternate between the frequency and time domain.

Keep in mind some important aspect about phase retrieval : *"In any case, it is common to see the phase jump around apparently randomly due to these undetermined, but not very important, quantities. Please don't interpret this to mean that the FROG algorithm isn't operating properly. Also, by definition, the phase becomes undetermined when the intensity goes to zero. So you'll see the phase jumping around in the pulse wings, where the intensity is nearly zero, too. This is also as it should be."*⁸

Furthermore, the SHG non-linear optical non-linear interaction produces a well know ambiguity in the direction of time and hence produces symmetrical trace (See figure 10). Figure J-10

However, this ambiguity can easily be removed in one of several way explain by Trebino and the reader is referred to his book (p128).Figure J-11⁹

J.4.3 Control Panel

The control panel on the top right is where the user can choose the operation to be done. There he can Start/Pause/End/Save the retrieval. Once the user has

8. Swamp Optics, http://www.swampoptics.com/tutorials_FROG.htm

9. Trebino, Rick, Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulse, Massachusetts, 2002.

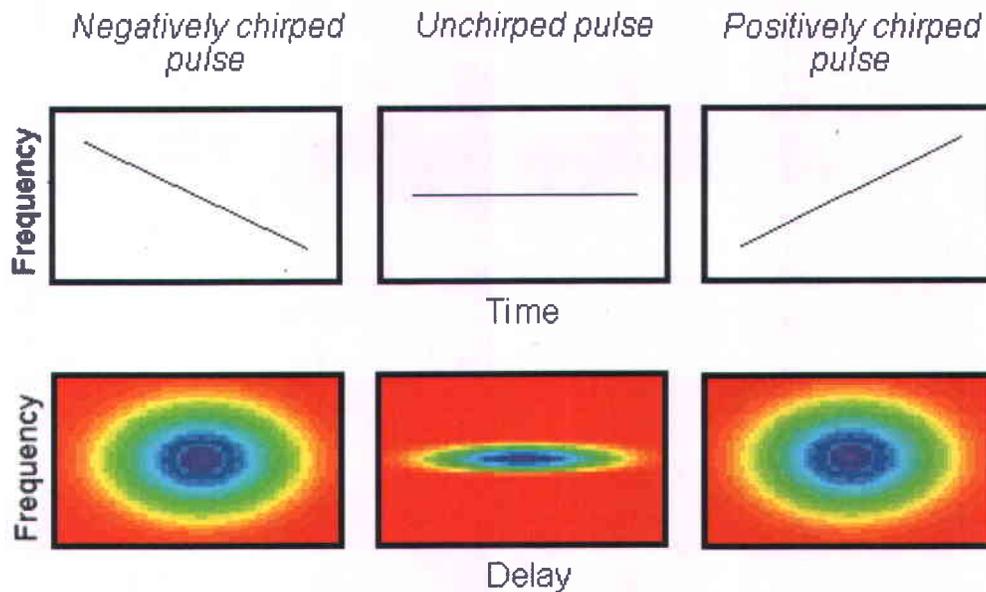


FIGURE J-10 – SHG FROG traces for linearly chirped pulses. Note that the traces are necessarily symmetrical, so the direction of time is not determined. This and a few "trivial" ambiguities are the only known undetermined parameters in SHG FROG. Traces SHG de FROG pour des impulsions linéairement gazouillées. Notez que les traces sont nécessairement symétriques, ainsi la direction du temps n'est pas déterminée. Ceci et quelque "trivial" ambiguïtés sont les seuls paramètres indéterminés connus dans la FROG de SHG.

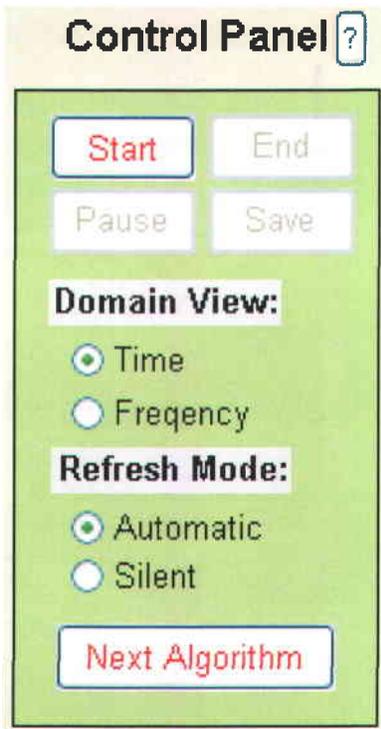


FIGURE J-11 – Control Panel. Panneau de commande

pressed the End button, the process cannot be continued. All windows are updated and the results appear in the result windows according to that action. As mentioned, the user can also switch between the frequency or time domain view for the amplitude and phase.

The Save option records many parameters of the retrieval in a Matlab® file that the user choose. It will save the original trace (*original_trace*), the frequency array in the frequency domain (*frequency_array*), the time array (*time_array*), the best retrieved field (*best_field*) and trace (*best_trace*), the amplitude and phase in the time domain (*time_amplitude,time_phase*) and frequency domain (*frequency_amplitude, frequency_phase*) with the date (*experiment_date*) and minimum error achieved (*minimum_error*). Furthermore, if the experiment is generated by theory, it will also save the original field (*original_field*) with the original amplitude and phase in the time domain (*original_time_amplitude,original_time_phase*) and frequency domain (*original_spectral_amplitude, original_spectral_phase*).

The “Refresh Mode” option let the user choose to refresh or not the windows during the retrieval process. On the 'Automatic' option, every graphs and windows are updated according to the default refresh rates (more on this later). On the 'Silent' mode no graphs and windows are updated. The calculation is done the same in the two modes, but since the speed to display the image on the graphical user interface is slow, it is better to stop it and view it just at the end. See the result panel for the iteration average per second. Finally, the Next Algorithm button, if pressed during retrieval, will change the algorithm for the next one in the list according to their

rank. This will force the change even if the minimum number of iteration has not been achieved.

J.4.4 Results Panel

If the type of experiment is theoretical, the panel will indicate the FWHM in the frequency and time domain with its time bandwidth product (TBP), otherwise the information is not displayed. When the user terminates the process, the FWHM in the temporal and spectral domain appears. Note however that if the retrieval is not good or the pulse shape is too complex, these informations are meaningless and should not be reported. Figure J-12

At each iteration, the current algorithm used is displayed with the iteration/second speed from the average of the ten last iterations by default.

J.4.5 Root Mean Squared (RMS) Error

The root mean squared error is displayed in the graph at the center button graph. The error of the last 50 iterations is shown. The equation for the error is based on the equation :

$$G^{(k)} = \sqrt{\frac{1}{N^2} \sum_{i,j}^N | I_{FROG}(\varpi_i, \tau_i) - I_{FROG}^{(k)}(\varpi_i, \tau_i) |^2} \quad (J.5)$$

where N is the number of point, $G^{(k)}$ is the current error, $I_{FROG}(\varpi_i, \tau_i)$ is the original FROG trace and $I_{FROG}^{(k)}(\varpi_i, \tau_i)$ is the k-th iteration of the retrieved FROG trace.¹⁰

10. Trebino, Rick, Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulse, Massachusetts, 2002.

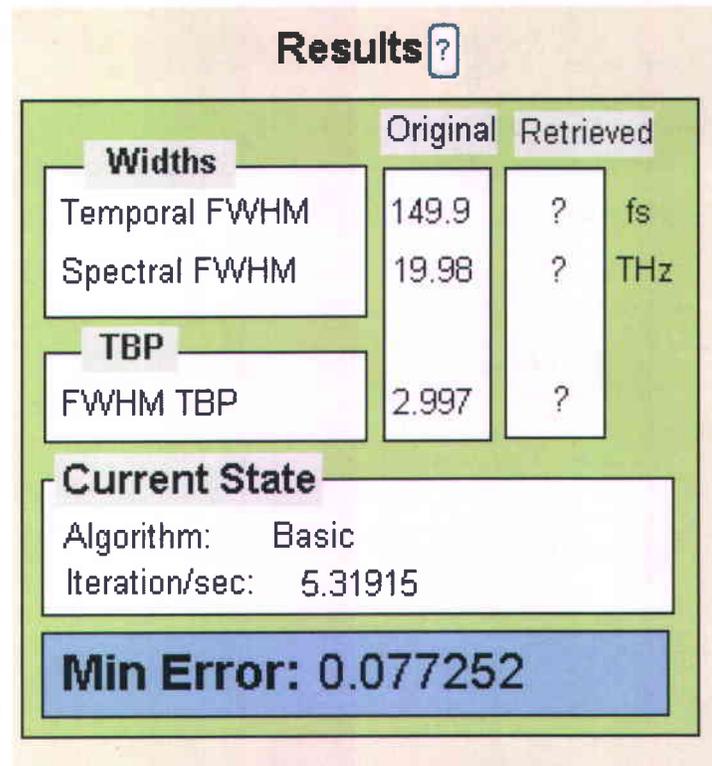


FIGURE J-12 – Result Panel. Panneau de résultat.

J.5 Default Configuration File

The FROG program has been implemented to easily set the default values, the parameter settings when the user starts the program. Every default settings can be changed in the file *default_options*. For example the width, chirp, type of experiment, etc. can all be changed in this file so that when the user starts the program, he sees those settings. Keep in mind however that bad values for those parameters could results in program error i.e. : initial negative width.

There are even some options that can only be set in this file. Those include the alpha parameter of the overcorrection algorithm (set to 1.3 by default), the refresh rate of the Amplitude/Phase graph, the refresh rate of the *rms* error graph, the refresh rate of the retrieved trace graph (all set to 5 iterations).

Also there is the cut-off intensity parameter that is the phase blanking threshold. By default set to zero, this could be a small fraction of the field amplitude. All the phase associated with amplitude smaller than that fraction is set to zero.

Finally, there is the average iteration number from which the iteration/second is built.[52]

Annexe K The FROG Algorithm

The task of the FROG algorithm is to retrieve the complex electric Field $E(t)$ from the measured FROG trace $I_{FROG}(\omega, \tau)$.

K.1 The Iterative-Fourier-Transform Algorithm Generalized projection

A generic FROG algorithm is designed as shown in Fig. K – 1. It works in the following way :

Step 1. Starting with an initial guessed field $E_{initial}(t)$, a random guess is used to start with.

Step 2. Calculating the nonlinear signal field, $E_{sig}(t)$. In case of SHG it is $E(t)E(t - \tau)$.

Step 3. Fourier transforming $E_{sig}(l, \tau)$ with respect to t to get the signal field in the frequency domain, $\tilde{E}_{sig}(\omega, \tau)$

Step 4. Replacing the amplitude of $\tilde{E}_{sig}(\omega, \tau)$ by the square root of the measured FROG trace, $I_{FROG}(\omega, \tau)$, to get an improved $\tilde{E}_{sig}(\omega, \tau)$,

Step 5. Inverse Fourier transforming $\tilde{E}_{sig}(\omega, \tau)$ back to time domain, $E_{sig}(t, \tau)$

Step 6. Minimizing $E_{sig}(t, \tau)$ to find out a better guess of $E(t)$ for the next iteration.

Step 7. Calculating the termination condition. If the condition is satisfied, algorithm returned, else go back to step 2 and continue.

The steps are summarized in Fig. K-1.

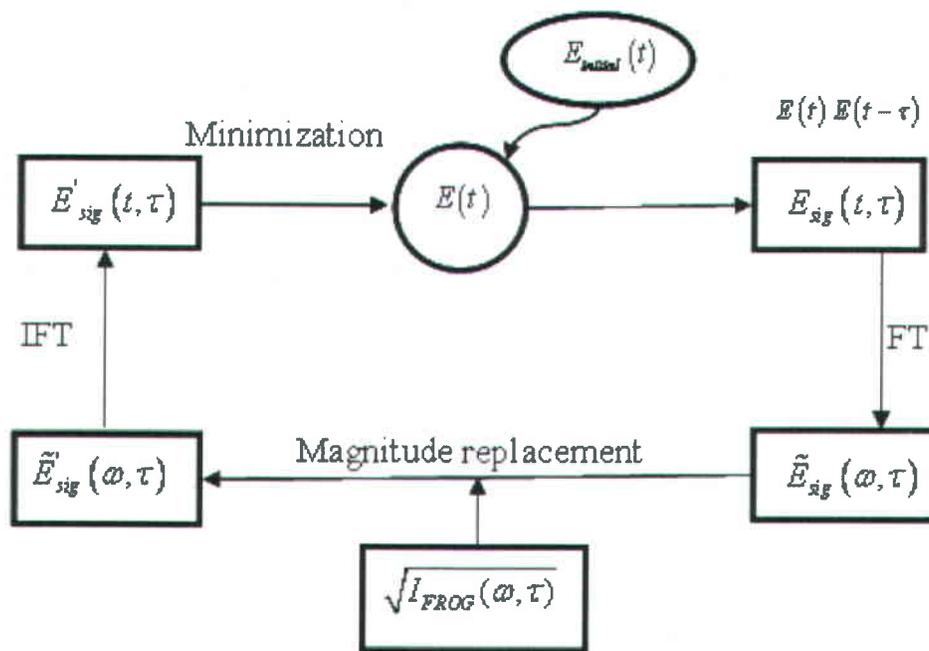


FIGURE K-1 – Schematic of a generic FROG algorithm

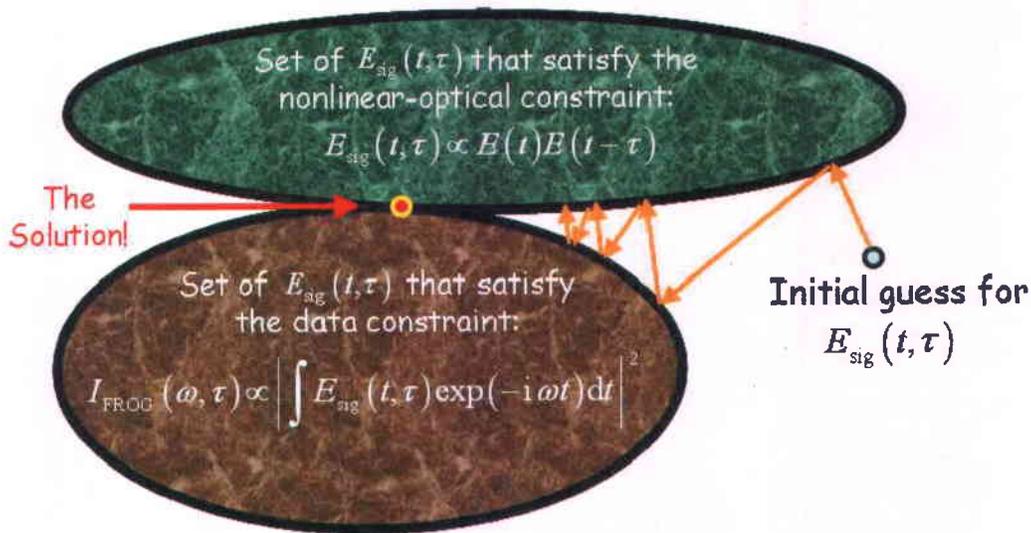


FIGURE K-2 – Solution space view of generalized projection.

In the iteration loop, two constraints are applied in the FROG pulse retrieval algorithm. The first constraint is the data constraint. This constraint indicates that the squared magnitude of $\tilde{E}_{sig}(\omega, \tau)$ should be equal to the measured FROG trace $I_{FROG}(\omega, \tau)$. This constraint is enforced with the magnitude replacement in step 4. The second constraint is the mathematical-form constraint or nonlinear constraint. This constraint requires the desired pulse field must obey the mathematical form with the nonlinear signal field, such as $E(t)E(t - \tau)$ in the SHG FROG. The mathematical form constraint is applied when doing minimization in step 6. To clearly understand the two constraints scheme in the FROG algorithm, the concept of generalized projection will be introduced. The idea of generalized projection is shown in Fig K-2. The lower elliptical region represents all the signal fields satisfying the

data constraint. The upper elliptical region indicates set of signal fields satisfying the mathematical-form constraint. The overlapping point of two regions is the pulse field satisfying both constraints, therefore the solution pulse field we are looking for. In the FROG algorithm, two constraints are applied to the target field alternatively. Therefore the guessed solution is projected between two constraint sets back and forth while approaching the real solution. The technique is so-called generalized projection. In practice, the generalized projection algorithm works very efficiently in FROG pulse retrieval.

Annexe L
Oriel *MS260*™ 1/4 m Imaging Spectrograph



FIGURE L-1 – Imaging Spectrometer. Spectromètre de formation image.

Focal Length	257 mm
F/#	F/3.9
Reciprocal Dispersion (nm/mm)	3.22 nm/mm
Usable Wavelength Range	180 nm to 24 μm , with interchangeable grating
Spectral Resolution (triple grating instruments) (triple grating instruments)*	0.20 nm, 1200 l/mm grating, 10 μm x 2 mm slits, 26 μm wide pixels
Spatial Resolution (triple grating instruments) *	~50 μm (FWHM)
Spectral Resolution (dual grating instruments)(dual grating instruments)*	0.15 nm, 1200 l/mm grating, 10 μm x 2 mm slits, 26 μm wide pixels
Spatial Resolution (dual grating instruments)*	~40 μm (FWHM)
Wavelength Accuracy	0.35 nm
Wavelength Precision	0.08 nm
Maximum Slew Rate	193 nm/s
Horizontal Magnification	1:1
Vertical Magnification	1.6
Weight	21 lb (11 kg)

FIGURE L-2 – Specifications for Imaging Spectrometer. Caractéristiques pour le spectromètre de formation image.