TIME RESOLVED BEAM PROFILE MONITORING USING SYNCHROTRON RADIATION ON CTF3 DELAY LOOP AND THE COMBINER RINGS

by

GULNUR OZDEMIR

THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

THE ABANT İZZET BAYSAL UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF PHYSICS

JUNE 2011

Approval of the Graduate School of Natural and Applied Science

Prof. Dr. Nihat ÇELEBİ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Kıvılcım KILIÇ
Head of Physics Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Haluk DENİZLİ
Supervisor

Examining Committee Members

1. Assoc. Prof. Dr. Haluk DENİZLİ ....

2. Assis. Prof. Dr. Cihan PARLAK ....

3. Assis. Prof. Dr. Zafer NERGİZ ....

ABSTRACT

TIME RESOLVED BEAM PROFILE MONITORING USING SYNCHROTRON RADIATION ON CTF3 DELAY LOOP AND THE COMBINER RINGS

GULNUR OZDEMIR

Master of Science, Department of Physics

Supervisor: Assoc. Prof. Dr. Haluk DENIZLI

June 2011, 47 pages

In this work, using Synchrotron Radiation (SR) with Beam Profile Monitoring (BPM) time resolved measurement has been investigated on CLIC Test Facility (CTF3) in the Combiner Ring (CR). General information about CLIC and CTF3 has been presented in the first chapter of the work. The Compact LInear Collider (CLIC) is multi-TeV electron positron collider, it aims to provide high luminosity (10^{34} $cm^{-2}s^{-1}$) as an electron-positron colliders with centre of mass 3 TeV for research of particle physics. CLIC beam is based on two beam scheme. One of the beam is the main beam which accelerates for CLIC Test Facility number of 3 is aims to make exhaustive tests of the main CLIC parameters. CTF3 is composed of a 70 m long drive beam linac followed by two rings, a 42 m Delay Loop (DL) and a 84 m Combiner Ring (CR). The electrons are sent from thermoionic DC gun to Drive beam accelerator where accelerates the beam. And after this process, the bunches are sent to the rings. The frequency multiplication is done in two steps, a factor two combination is achieved in Delay loop and a factor 4 in the Combiner Ring. After these processes, the beam has 35 A current and 12 GHz bunch frequency. Drive beam is transported to this beam to experimental area. Aim of the drive beam is to produce 12 GHz RF power. When electrons pass the dipole magnets in the CR, radiation is produced which is called Synchrotron Radiation (SR). The properties of the radiation has been determined by SPECTRA simulation programme. And this radiation is captured by gated Camera. Beam size investigated for every turn with beam image. Analysis of beam image has been done by using Matlab Programme for each turn of the beam.

Keywords: CLIC, CTF3, CR, DL, Synchrotron, Beam Profile Monitor, Time Resolved.

ÖZET

CTF3'DEKİ GECİKTİRİCİ HALKA İLE BİRLEŞTİRİCİ HALKALARDA SİNKROTRON IŞINIMI KULLANARAK ZAMAN ÇÖZÜMLEMELİ DEMET PROFİLİNİN İNCELENMESİ

GÜLNUR OZDEMİR

Yüksek Lisans, Fizik Bölümü

Tez Danışmanı: Doç. Dr. Haluk Denizli

Haziran 2011, 47 sayfa

Bu çalışmada sinkrotron ışınımı kullanarak demet görüntüleme monitori (BPM) ile CTF3 Combiner Ring (CR) de zamana bağlı ölçümleri incelenmiştir. Çalışmanın ilk bölümünde CLIC ve CTF3 hakkında genel bilgi sunulmuştur. CLIC çoklu TeV elektron positron çarpıştırıcısıdır. Elektron pozitron çarpıştırıcısı olarak 3 TeV'lik kütle merkezi ile parçaçık fiziği aratırmalarında yüksek parlaklık üretmeyi amaçlar. CTF3, CLIC ana parametrelerini test etmeyi amaçlar. CTF3 70 m uzunluğunda sürücü linak ile 84 m CR ve 42 m DL oluşmuştur. Elektronlar termiyonik DC tabanca tarafından sürücü demet hızlandırıcılara(Drive beam accelerator) demeti hızlandırılmak üzere gönderilir. Bu işlemden sonra demet halkalara gönderilir. Geciktirici halka içinde (DL) frekans çarpımı (frequency multiplication) ve birleşimi (combination) 2 kez, birleştirici halka içinde (CR) 4 kez yapılır. Bu işlemlerden sonra demetin akımı 35 A, demet frekansı 12 GHzdir. Bu demet, sürücü demet (Drive Beam) tarafından deneysel alana (experimental area) transfer edilir. Sürücü demetin amacı 12 GHz lik RF gücü üretmektir. CR içerisinde elektronlar dipole magnetlerden geçerken ışınım üretilir , sinktrotron ışınımı olarak adlandırılır. Sinkrotron ışınımının özellikleri SPECTRA simulasyon programı kullanılara belirlenmiştir. Ve bu ışınım gated camera tarafından görüntülenir. Çekilen bu görüntü ile demetin her dönüşdeki boyutu incelenir. Her dönüş için demetin simulasyonu, MATLAB programı kullanılarak analizi yapılır.

Anahtar Kelimeler: CLIC, CTF3, CR, DL, Sinkrotron, Demet görüntüleme monitori, zaman çözümleme.

To My Family . . .

ACKNOWLEDGEMENTS

I would like to thank very much my first advisor, Dr. Haluk DENİZLİ for filling me with enthusiasm for Accelerator Physics who is always a sympathetic listener, helping me with discoussions and many suggestions.

My special thanks go to Thibaut LEFEVRE who is the leader of the beam diagnostic group at CERN. I am obliged for his continuous guidance and many suggestions, comments and valuable contributions. I would like to my sincere thanks to Benoit BOLZON for his trust in me. This thesis would not have been possible to finish without the help of them during my studies at CERN.

I would also want to express my sincere thanks to my close friends Gülnur and Gürcan YILDIRIM, Zeynep Liva ÖZTÜRK, Sümeyye TAKA, Ümeyye Taka AYDIN and Firdevs AKPEK for their unlimited moral support when I felt the need for it. They were always with me throughout the process of this study.

I would like to thank TAEK for financial support during my studies at CERN.

For memory of my dear friend DİLEK ÖZDEMİR died in 19.11.2010...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The global understanding of the Universe is, still nowadays, relatively uncomplete. Most of the Universe constituents are unknown and several theory have proposed to prove the existence of Dark matter and Dark energy.

High energy physics is dedicated to the study of matter, the knowledge of elementary particles and the forces existing between them. During the last 30 years, with the development of accelerators, new particles predicted by the Standard Model (Figure 1.1) have been found, like the Z and W bosons, responsible for the strong and the electro-weak forces respectively [1]. They were studied in details on the Large Electron-Positron Collider (LEP) [2] a 27 kms circumference collider at CERN.

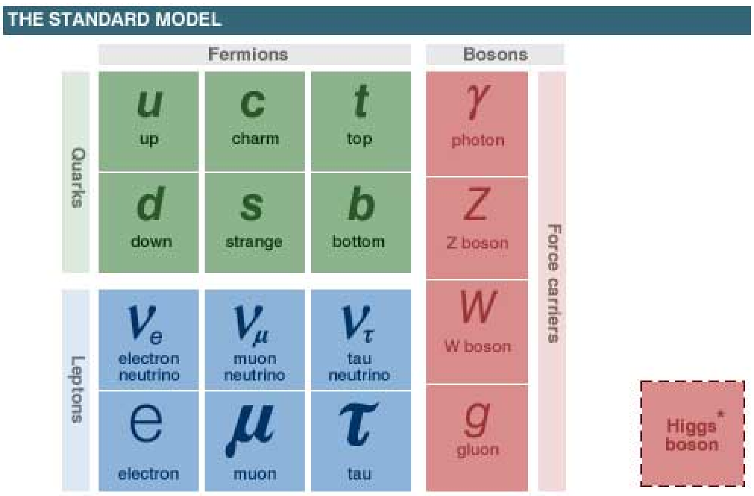


Figure 1.1: Figure of standart model.

LEP started to operate in 1989, it accelerated the electrons and positrons to the energy of 45 GeV and was upgraded in a second phase to reach the highest centre-of-mass energy in electron-positron collisions so far, 209 GeV. The collider was shutdown at the end of 2000 and dismounted in order to install the Large

Hadron Collider (LHC) in the same tunnel [3]. Energies higher than 1 TeV are currently available on TEVATRON [4] at Fermilab but so far the missing particle of the standard model has not been seen yet. Higgs boson is predicted by the standart model, and suggest that it should weight < 200 GeV [5]. Direct searches at the CERN LEP collider have set a 95 % confidence level limit on the Higgs boson mass of $m_H > 114.4$ GeV [6]. Taking into account this limit, precision electroweak measurements indirectly constrain the SM Higgs boson mass to be lower than 190 GeV at the 95 % Confidence Level [5].

LHC accelerates protons or heavy ions to a maximum center of mass energy of 14 TeV and the first collisions started in March 2010 [7]. Within this new energy reach, the LHC should confirm or refute the existence of the Higgs boson to complete the standart model predictions. In addition, the LHC experiments will explore the possibilities for physics beyond the Standart Model, such as supersymmetry, extra dimensions and new gauge bosons [8].

Hadron colliders are known as discovery machines, which are capable of providing high energy for a lower cost. But hadrons are not elementary particles, they are heavy, composite objects, constitued of quarks and gluons [10]. CLIC therefore, the hadron collisions are quite noisy. On the other hand e^-/e^+ colliders are by nature providing cleaner collisions and can be used for precision measurements. As a companion to the LHC experiments, many physicists agree that a new TeV center of mass energy (or multi- Tev) electron-positron collider should be built. It will complement LHC results with high precision and polarisation dependency. With its 27 km, LEP was already achieving the limit in size for circular lepton machine, were the energy gain per turn is just compensating the natural energy loss due to the emission of synchrotron radiation. To reach TeV beam energy, an electron-positron accelerator can not be based on a synchrotron and would be a linear machine.

During the last 20 years different technology have been studied in order to develop accelerating structures with the highest field and produce TeV beams in a ‘short’distance. At the moment there are two main projects : the International Linear Collider (ILC) and the Compact LInear Collider (CLIC) at CERN. The ILC [9] is 500 GeV center of mass energy collider based on the use of superconducting accelerating structures. The CLIC uses high field normal conducting cavities that can potentially deliver 3 TeV center of mass energy in the same distance as the ILC [10].

CTF3 is a prototype of CLIC. My work is to measure transverse beam size with synchrotron radiation.

The thesis has written in the following way.

• Firstly,genaral information about research LEP, TEVATRON, LHC, CLIC, ILC, CTF3 are given. Physics of these accelerators are explained briefly in chapter 1.

• In Chapter 2 detailed information about the CLIC and the CTF3 is included. Specially, beam manipulations were told in the CTF3 on CR.

• In chapter3 characteristic of synchrotron radiation was investigated on the CR by SPECTRA programme.

• In chapter 4, experimental set up discussed in detail

• In chapter 5 results are summarized.

CHAPTER 2

COMPACT LINEAR COLLIDER TEST FACILITY 3 PROJECT AT CERN

2.1 The Compact Linear Collider (CLIC)

CLIC (figure 2.1) is a linear collider, which accelerates electrons and positrons in opposite directions. CLIC is designed to provide the luminosity of $5.9 \times 10^{34} cm^{-2} s^{-1}$ [11], with a centre of mass energy of 3 TeV. Although the design study has been optimized for 3 TeV centre of mass energy, the collider could start operation at a lower energy and then be upgraded in stages. CLIC physicists designed accelerating structures at 12 GHz, capable of achieving a gradient of 100 MV/m. CLIC layout is presented in the Figure 2.1.

CLIC, like all linear collider, is composed of five main parts: electron and positron sources, damping rings, longitudinal bunch compressors, long linear accelerator and a final focusing system. The source of polarized electrons is based on the use of a photo-injector with a DC gun. Once extracted, the electrons are directly accelerated to 200 MeV in a linear accelerator using 2 GHz normal conducting cavities. Contrary to the electron source, the positron source relies on a much larger accelerator complex and requires a preliminary electron injector complex of 5 GeV. These electrons are then converted in a first target into high-energy photons, which then generate positron through e^{+}/e^{-} pair creation in a second target. The positron beam is then pre-accelerated to 200 MeV. Electrons and Positrons are finally accelerated in a common 2 GHz Linac up to 2.86 GeV. The transverse normalized beam emittances are then reduced from mm.rad to nm.rad in two consecutives rings, namely the Pre-Damping Ring and the Damping Ring. At the extraction of the DR, typical values are below 500 and 5 nm.rad respectively for the horizontal and vertical emittances. From the exit of the Damping rings, the beam enters in the Ring To Main Linac, where the bunches are compressed in a first magnetic chicane, accelerated to an energy of 9 GeV in a Booster Linac and transferred to the entrance of the Main Linac through more than 20 kms of beam line. After a final 1.5 km long turn-around loop, bunches are compressed even further in a second magnetic chicane to a final bunch length of 44um R.M.S. The beam is then accelerated in the Main Linac, composed of a succession of more than 70 thousands of 12 GHz accelerating structures. The beam energy rises up to 1.5 TeV over a total distance of 20.5 kms. In order to guarantee the design luminosity, the beam emittance growth from the Damping Ring to the Interaction Point must be kept below 20%. The beam is finally transported through the Beam Delivery System (BDS) to the Interaction Point. The BDS has three main functionalities. First, a beam diagnostic section characterizes the beam properties, measuring emittance, energy, and polarisation. The second part of the BDS is dedicated to the beam cleaning using a succession of collimation scheme and finally the last part provide the final optic adjustment, focussing the beam down to nanometres beam sizes at the Interaction Point. After collisions, the beam, highly disrupted is then guided towards a huge water dump. This part of the machine, called the Spent Beam line has also some crucial impact for instrumentation. The CLIC luminosity monitor is based on the detection of beamstrahlung photons generated at the IP.

Table 2.1 shows the nominal parameters for CLIC operating at its nominal design energy of 3 TeV [13].

Moreover, the CLIC technology relies on a so-called 'two-beam acceleration'scheme (see Figure 2.2) [14], where a second electron beam with a moderate energy but a high current and high bunching frequency is used to generate the radiofrequency power required to accelerate the main beam, as depicted on figure just below. The Drive Beam complex is nothing more than a power source, the equivalent of the 12 GHz relativistic klystron distributed over a distance of 48 kms, which is sup-

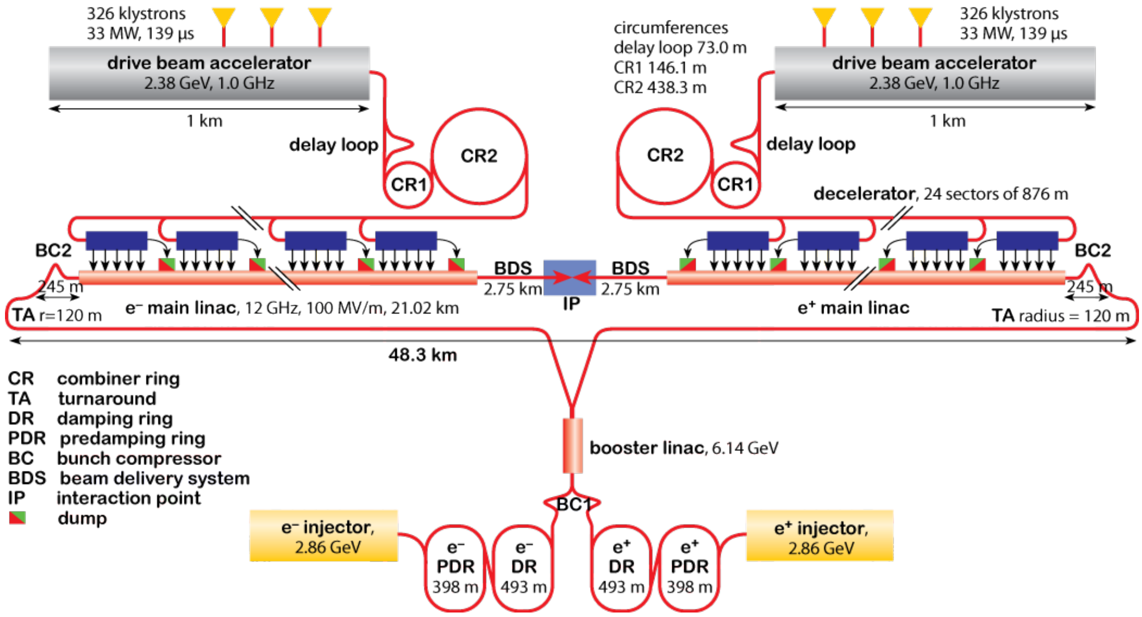


Figure 2.1: The CLIC layout, showing two-beam acceleration scheme.

posed to run with the best efficiency and reliability. Along the CLIC Main linac, both the drive and the main beam propagate the one close to the other. The drive beam is being decelerating by passing through 12 GHz resonating decelerating structures, named Power Extraction and Transfer Structures (PETS), and the RF produced is directly transferred to the CLIC main beam 12 GHz accelerating structures. Along the decelerator, the beam energy will be gradually transformed into 12 GHz RF power over some hundreds of meters. The beam energy spread rises up linearly to finally reach a value of 90%, before being dumped.

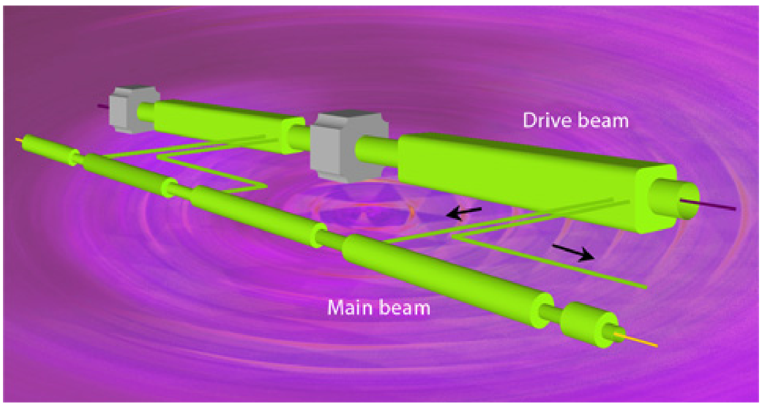


Figure 2.2: CLIC two beam acceleration concept.

More informations on the CLIC study can fe found here [15].

Overall Parameters	Symbol	Value	Unit
Centre of mass energy	E_{CMS}	3000	GeV
Main Linac RF Frequency	f_{rf}	11.994	GHz
Luminosity	L	5.9	$10^{34}\ cm^{-2}s^{-1}$
No. of particles / bunch	N	3.72	10^{9}
No. of bunches /	N_b	312	
Bunch separation	$\triangle t_b$	0.5 (6 periods)	ns
Bunch train length	τ_{train}	156	ns
Beam power / beam	P_b	14	MW
Unloaded / loaded gradient	G_{unl}/l	120/100	MV/m
Overall two linac length	l_{linac}	42.16	km
Total beam delivery length	l_{BD}	2 x 2.75	km

Table 2.1: Overall CLIC parameters [13].

Overall Parameters	Symbol	Value	Unit
Energy (decelerator injection)	$E_{in,dec}$	2.37	GeV
Energy (final, minimum)	$E_{fin,dec}$	237	Mev
Average current in pulse	I_{dec}	101	A
Train duration	t_{train}	243.7	ns
No. bunches / train	N_b	2922	
Bunch charge	Q_b	8.4	nC
Bunch separation	D_b	0.083	ns
Bunch length, rms	σ_s	1	mm
Normalized emittance, rms	$\gamma\varepsilon_{dec}$	150	μ m rad

Table 2.2: CLIC drive beam basic parameters [13].

2.2 The CLIC Drive Beam injector complex

As indicated in Table 1, the drive beam has a high current (100 A) and a high frequency (12 GHz) with a nominal beam energy of 2.4 GeV. Such a beam cannot be produced using classical electron gun and a scheme has been proposed back in 1999 [16]. In this scheme, the drive beams are produced by converting a low current (4.3 A) low frequency (500 MHz) but with a very long pulse of 140 μs (i.e. extremely high beam charge of 590 μ) into a serie of consecutives drive beams with high current (100 A), high frequency (12 GHz) but much shorter pulse duration (140 ns). This beam frequency multiplication is achieved by injecting and combining the bunches in one delay loop and two combiner rings. A small-scaled version of the CLIC drive beam, named 'CLIC Test Facility 3'[49] has been built at CERN during the last 8 years.

2.3 The CLIC Test Facility 3

CTF3 (figure 2.3) has been constructed at CERN by an international collaboration since 2003. The main goals of CTF3 are to prove the feasibility of the drive beam generation and to test and develop the main CLIC components, like 12 GHz cavities and PETS and state of the art beam instrumentation [15].

Table 2.3, 2.4, and 2.5 show the beam parameter of the CTF3 Machine and a comparison with the drive beam injector complex.

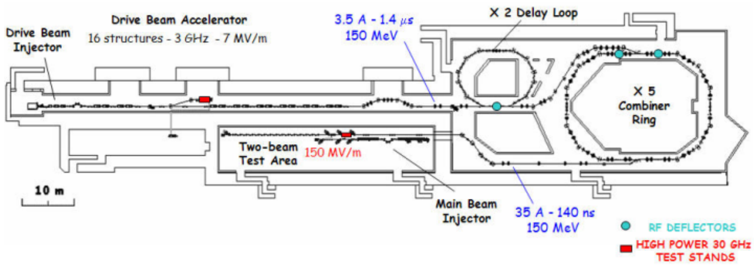


Figure 2.3: Layout of CTF3 [17]

Figure 2.3 shows schematic layout of CTF3 [18] which is composed of three main parts: drive beam linac, the rings, and the CLIC experimental area (CLEX). The electrons are generated out of thermoionic DC gun followed by a 1.5 GHz sub-harmonic bunching system and a coupled of 3 GHz linear accelerator. At the end of the injector, the beam energy is 20 MeV. It is followed by the drive beam accelerator, consisting of a seri of 3 GHz normal-conducting accelerating strucutures, which brought the beam energy to 150 MeV. At the end of the linac, the bunch length can be manipulated using a 4-bends magnetic chicane. Normally the bunches are lengthened before being sent in the rings. The frequency multiplication is done in two steps [19], a factor two combination is first achieved in the Delay Loop and a factor 4 can be done in the Combiner Ring. The Delay Loop and the Combiner Ring are 42 m and 84 m long respectively (Figure 2.4 and 2.5). After the combination process, the drive beam, with a 35 A beam current and a 12 GHz bunch frequency is finally transported to experimental area where it is used to produce 12 GHz RF power.

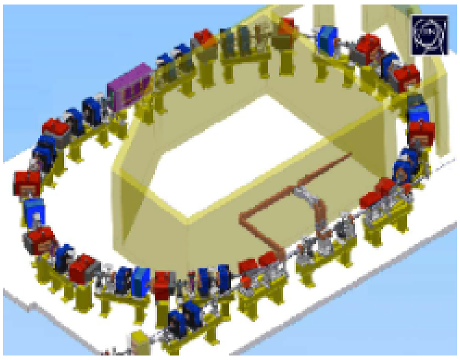


Figure 2.4: Delay loop layout.

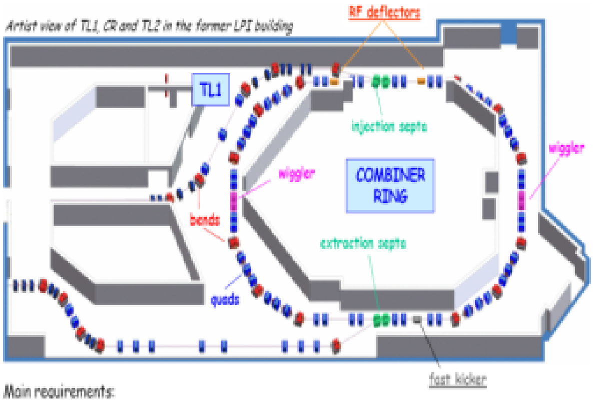


Figure 2.5: Combiner ring layout.

Finally, there are differences between the CTF3 beam and the CLIC drive beam which are the energy and current. The CLIC drive beam has a beam current of 100 A, whereas the CTF3 drive beam has a beam current of 35 A and energy of 150 MeV.

2.3.1 Delay loop (DL) and Combiner Ring (CR)

Drive beam linac aims to produce high current, high frequency electron beam using bunch multiplication frequency techniques with help of RF injection and RF manipulation which are used in Delay loop and Combiner Ring [20]. The aim of the DL and of the CR is to increase the current and the bunch repetition frequency. Just after the thermionic gun, a bunching system is used to obtain the time structure of the pulse. Among others, sub harmonic bunching cavities have their phase which are switched rapidly by 180° every 140 ns which is needed in order to perform bunch multiplication frequency in the DL and CR (Figure 2.7).

After the linac, beam manipulations such as electron pulse compression and bunch frequency multiplication are made in two steps. First beam manipulations are done by a factor two in the DL, 42 m long. Other beam manipulations are done by a factor four in CR, 84 m long.

Before the bunches are not transferred in DL, the bunches in the drive beam have a charge of 2.3 nC per bunch, an average current of 3.5 A, 1.4 μs long train , energy of 150 MeV. The first phase sub-pulse, called "even bunches" of 140 ns length, are sent into the DL using RF deflector which works at 1.5 GHz. Then, the next sub pulse, which is called "odd bunches" (also of 140 ns length), is not sent into the DL from the linac thanks to the phase difference of 180° between even and odd buckets made by the subharmonic bunching cavities. The 'even bunches 'are then coming back of odd bunches at the same time. So, each the odd buckets are recombined with each of the even buckets. The proccess is shown in Figure 2.6. So the resulting bunch compression structure and bunch frequency multiplication, at the DL output, shows four 140 ns long bunch trains spaced by 140 ns, (Figure 2.8).

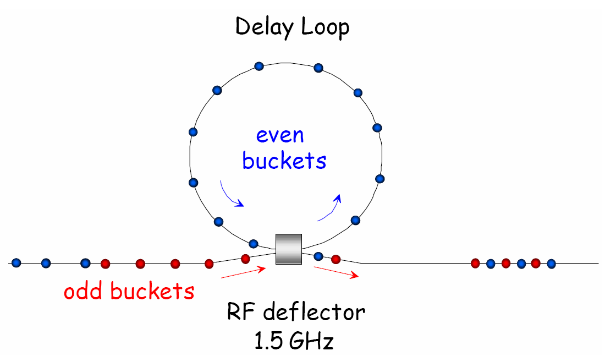


Figure 2.6: Operation of the Delay loop [20].

After the first pulse compression and frequency multiplication by a factor 2 completed, these electrons are transferred to the CR by two RF deflector working at 3 GHz [18]. Combiner Ring is used for second beam manipulations.

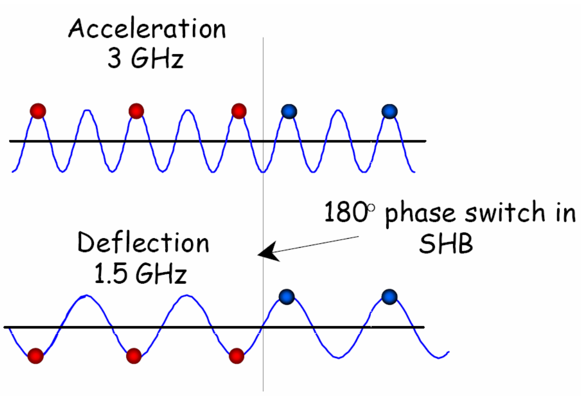


Figure 2.7: 180° phase switch between two batches of bunch trains [20].

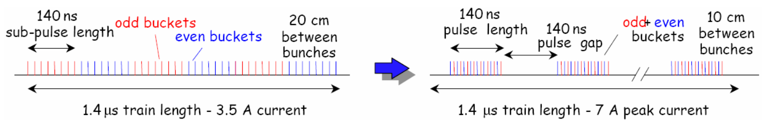


Figure 2.8: Bunch frequency multiplication [20].

Figure 2.9 shows operation of the injection using Rf deflector for a factor four inside the Combiner Ring.

1. Fist turn, incoming bunch trains are kicked by the RF deflector and circulate in the ring.

2. After the first turn is finished in the RF deflector at the zero crossing of RF field, the bunches arrive and kept on the equilibrium orbit. The second train is sent into the ring.

3. After a second turn, the first train is deflected in opposite direction. When the second bunch train arrive at the zero crossing of the RF field, the third train are injected into the ring.

4. After four turns, the drive beam pulse is 140 ns long and has an average current of 35 A with 2.3 nC bunches, and 2 cm space between bunches.

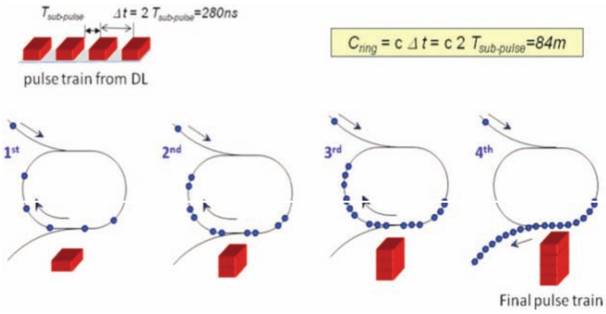


Figure 2.9: Operation of Combiner Ring [41].

Overall Parameters	Symbol	Value	Unit
Energy	E_i	20	MeV
Current	I_i	3.5	A
Pulse duration, total	τ_i	1.54	μs
Bunch charge (flat top)	Q_b	2.33	nC
Number bunches/pulse	N_b	2310	
Bunch separation	$\triangle_b$	0.67	ns
Bunch length, rms	σ_z	1.3	mm

Table 2.3: CTF3 drive beam injector exit [50].

Overall Parameters	Symbol	Value	Unit
Energy	E_a	150	MeV
Current	I_a	7	A
Pulse duration, total	τ	1.4	μs
Bunch charge (flat top)	Q_b	2.33	nC
Number bunches/pulse	N_b	2100	
Bunch separation	$\triangle_b$	0.33	ns
Bunch length, rms	σ_z	1.5-2.5	mm
Energy spread, rms single bunch	$\triangle E/E_{sb}$	~ 1	%

Table 2.4: CTF3 drive beam (Combiner ring injection) [13].

Overall Parameters	Symbol	Value	Unit
Energy	E_r	150	MeV
Current	I_r	35	A
Pulse duration, total	τ_r	140	ns
Bunch charge (flat top)	Q_b	2.33	nC
Number bunches/pulse	N_b	2100	
Bunch separation	$\triangle_b$	0.067	ns
Bunch length, rms, after compression	σ_z	0.5	mm
Energy spread, rms single bunch	$\triangle E/\ E_{sb}$	~ 1	%

Table 2.5: CTF3 drive beam(Combiner ring) [13].

CHAPTER 3

SYNCHROTRON RADIATION

Synchrotron Radiation (SR) is emitted when charged particles are accelerated under electromagnetic field. It was first observed in a synchrotron in 1947, so the name is called “SR” and it has been used the first time as a research tool in the mid-1960s [21].

The derivation of the fundamental equations describing the emission of Synchrotron Radiation can be found in electrodynamic books [22, 23]. There are also several articles on this topic [24, 25, 26]

When charged particles are moving on circular orbits, these particles lost a specific amount of energy. You can find more information about derivation of energy loss of synchrotron radiation in appendix A. The energy loss is given by

$$\triangle E = \frac{q^2}{3\epsilon_0(m_0.c^2)^4}\frac{E^4}{\rho} \tag{3.1}$$

In a circular accelerator, this equation (3.1) indicates that energy loss per turn scales like the fourth power of the particle energy and it is inversely proportional to the radius of the accelerator. Table 3.1 gives the energy loss per turn for several accelerators as well as the accelerator parameters allowing calculating this loss.

Accelerator	L[m]	E[GeV]	$\rho[m]$	B[T]	$\triangle E[keV]$
BESSY I	62.4	0.80	1.78	1.500	20.3
DELTA	115	1.50	3.34	1.500	134.1
DORIS	288	5.00	12.2	1.370	$4.53\text{X}10^3$
ESRF	844	6.00	23.4	0.855	$4.90\text{X}10^3$
PETRA	2304	23.50	195.0	0.400	$1.38\text{X}10^5$
LEP	$27\text{X}10^3$	70.00	3000	0.078	$7.08\text{X}10^5$
CTF3(CR)	84	0.15	1.09	0.46	0.04

Table 3.1: Parameters of a few circular electron accelerators [27].

Synchrotron light sources produce synchrotron radiation from a bending magnets or insertion devices (wiggler, undulator) over the spectral range from Infrared to the hard x-ray region which has maximum intensity, and also this sources would produce short pulses of radiation to reach higher performance levels.

3.1 Creation of Synchrotron Light

Dipolar magnets bend particles as they pass through (see Figure 3.1). This change in direction causes them to emit synchrotron radiation. It produces a broad- band smooth spectrum, where we can define a critical frequency ω_c such that equal power is radiated below and above ω_c.

$$\omega_c = \frac{3c\gamma^2}{2\rho} \tag{3.2}$$

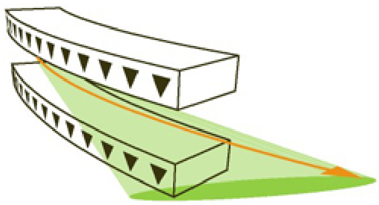


Figure 3.1: Emission of bending magnet.

Insertion device, like wiggler or undulator are made of a linear array of magnet poles (see Figure 3.3). The SR emitted by each magnet interfere with that from other ones. Depending on the wavellength the interfere will become constructive. An insertion device can be defined by a parameter called 'K'where K is given by

$$K = \frac{eB\lambda_u}{2\pi m_0 c} \tag{3.3}$$

where λ_u is the distance between poles, e is the electron charge, m_0 the particle mass, c the speed of light and B the peak magnetic field. The wavelength of the emitted light is then defined by

$$\lambda = \frac{\lambda_u}{2\gamma^2}(1 + \frac{K^2}{2} + \gamma^2\Theta^2) \tag{3.4}$$

where γ is the relativistic factor and Θ is the angle of observation in radians.

We can split the insertion devices in two kinds. when K<1, an observer would see radiation from everywhere in the device. It is called an Undulator (see Figure 3.2) magnet is a long periodic arrangement which causes particles to be a little deflected. It produces a narrow band radiation.

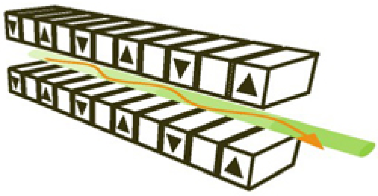


Figure 3.2: Emission of undulator magnet.

If K>1, it is a wiggler and it would appear as a broad spectrum of incoherent synchrotron light but with a higher photon flux compared to a dipolar magnet [28].

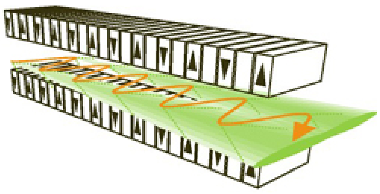


Figure 3.3: Emission of a wiggler magnet.

3.2 Synchrotron Radiation Applications

3.2.1 Light source

Synchrotron light source is one of the important types of accelerator which have become valuable tools for research. Light sources are research facilities which are built to produce and use of synchrotron radiaition, which are provides higher flux (photons/s/mrad/unit bandwith), higher brightness (photons /s/unit solid angle/unit solid area/unit bandwith) etc.

A)First Generation Light sources:

These sources constructed for high energy physics research, which have emittances of one hundred to several hundred mm radians [29]. With providing synchrotron radiation, it has constant spectrum. For example DORIS, SPEAR, TRISTAN, PETRA, PEP etc. were used as a first generation light sources . In generally, these sources give low electron beam emittances (about below 1 nm-rad) at lower energy.

B)Second Generation Light Sources:

They were designed as dedicated light sources, which were built from many bend magnets, and later insertion devices could be added [30]. And also second generation sources go below 100 nm-radians [29]. SOR ring is given as an example for second generation source.

C)Third Generation Light Sources:

These rings have many straight section for insertion devices and lower electron beam emittance (5-20 nmm) [31]. European Synchrotron Radiation Source (ESRF) [32], France (6 GeV, 844 m circumference, 32 straight sections) the Advanced Photon Source (APS) [33],USA (7 GeV, 1104 m circumference, 40 straight sections), SPring-8 facility [34], Japan (8 GeV, 1436 m circumference, 48 straight sections) are third generation hard X-ray sources.

D)Fourth Generation Light Sources:

These light sources in the wavelength range from the VUV to hard X-rays are lower emittance rings. And also they produce short pulse of radiation. For example, FELs offer sub-picosecond pulses compared to 10-50 ps for present storage rings, which used on both linac and ring as drivers [30].

So, these light sources have a broad range of wavelength from the infrared trough the visible and UV range and also X-ray which are part of electromagnetic spectrum. They are used research areas such as material science, electronic and comunications, radiology, x-ray tomograph, chemistry [35, 36].

3.2.2 Damping Ring

It was very rapidly observed that synchrotron radiation damping can produce very low emittance beam. For the next generation of linear colliders, like ILC and CLIC, as discussed in paragraph 2, very low emittance needs to be produced. They will be generated in special rings, named damping rings, where the beam is forced to emit SR to see its emittance shrinking.

3.3 Synchrotron Radiation as a Beam Diagnostic tool

Synchrotron radiation produce an good optical repliqua of particle beams. It can therefore be used to measure the beam properties such as transverse(beam size, emittance, intensity, dispersion etc.) and longitudinal profile measurements (bunch length, bunch spacing etc.) [37, 38]. It gives also information on the behaviour of the beam such as its position.

Beam diagnostics has a very big impact on the advancement of the accelerators and storage rings and it will give very useful informations for the next light source generations.

Beam diagnostic instrumentation is developed to check performance and stability, but also to control the wanted changes of parameters (like twiss parameters matching) in the accelerator setting [39].

Most of the diagnostic instrumentation is based on physical process like SR. This diagnostic is important for relativistic particles such as electrons. This is based among others on optical techniques [40] which is used in beam diagnostic instruments. Examples are synchrotron radiation monitors for beam profile and time measurements [39]. More detailed informations can be found in [39].

Synchrotron light is being used to diagnose the electron beam such as beam profile, emittance measurements, beam size, energy etc. on this rings [37, 38]. Incoming light is transported thanks to an optical line, which can be composed for example of lenses and mirrors.

The informations of position and intensity of the beam given by SR are usually correlated with the ones of essential instruments such as Beam Position and Intensity Monitors (the position of a beam is usually determined with pick - up plates) [41]. Design of the BPM pickup has been based on the time structure of the beam to provide the measurement of the average position of the whole bunch train at each turn in the combiner ring/delay loop [42].

3.4 Simulation of Synchrotron Radiation

In the Combiner Ring of CTF3, we are using bending magnets to produce synchrotron radiation in order to perform transverse and longitudinal beam profile measurements. The characteristic of this radiation (flux, power, brightness etc.) has been determined by using the SPECTRA simulation programme and the results are presented in this section. SPECTRA is an application software to calculate properties of synchrotron radiation emitted from bending magnets,wigglers and undulators [43].

Flux is defined by Eq.(3.5) as the number of photons per second, per unit bandwidth, per unit horizontal opening angle. In many experiments, it is important to produce high flux.

3.4.1 Radiation from a Bending magnet

The photon flux of the synchrotron radiation from the bending magnet is given by equation (3.5),

$$\frac{d\Phi(y)}{d\Theta} = 2.458 \cdot 10^{13} \frac{photons}{s \cdot 0.1\% BW \cdot mrad} \cdot (E/GeV).(I/A).(\Theta/mrad) \cdot G_1(\frac{\varepsilon}{\varepsilon_c}) \tag{3.5}$$

According to this equation, the photon flux is proportional to the beam current, the energy and the normalized function $G_1(\frac{\varepsilon}{\varepsilon_c})$ which depends only on the critical photon energy. The equation for the critical photon energy is given by

$$\varepsilon_c = 0.655keV \cdot (E/GeV)^2 \cdot (B/T) \tag{3.6}$$

Table 3.2 summarizes the main parameters of the Combiner Ring enabling such calculations, including also the parameter specifications of the dipole magnet. Table 3.3 gives the parameter specifications of the wiggler magnet, which is usually used to make the beam describing a closed orbit in the CR.

Parameter	Value	Unit
Energy	$0.15 - 1.79$	GeV
Bρ	0.60/1.09	T m
Circumference	84	m
Max. beta	11.1/11.1	m H/V
Max. Dispersion	0.72	m
Current	35	A
S_z	0.5	mm
Betatron Tune	7.23/4.14	H/V
Chromaticity	$-12.0/-8.8$	H/V
Bunches	2100	
Momentum compaction	$< 10^{-4}$	
Horizontal emittance	1/0.5	mm mrad
Vertical emittance	1/0.5	mm mrad
Energy spread	$-+1$	percent
Energy acceptance	$-+2.5$	percent

Table 3.2: Main parameters of the combiner ring.

The magnetic field of the bending magnet is 1.09 T for the electron beam energy of 0.15 GeV and the critical photon energy is 0.016 keV. The fluxes emitted within the bending magnet from a stored electron beam at different energies are presented in Figures 3.4, 3.5, 3.6 . These simulations show that the photon flux increases with the energy and these results are thus in agreement with equation (3.5). Due to the larger critical energy, photon flux is shifted to photon higher energies.

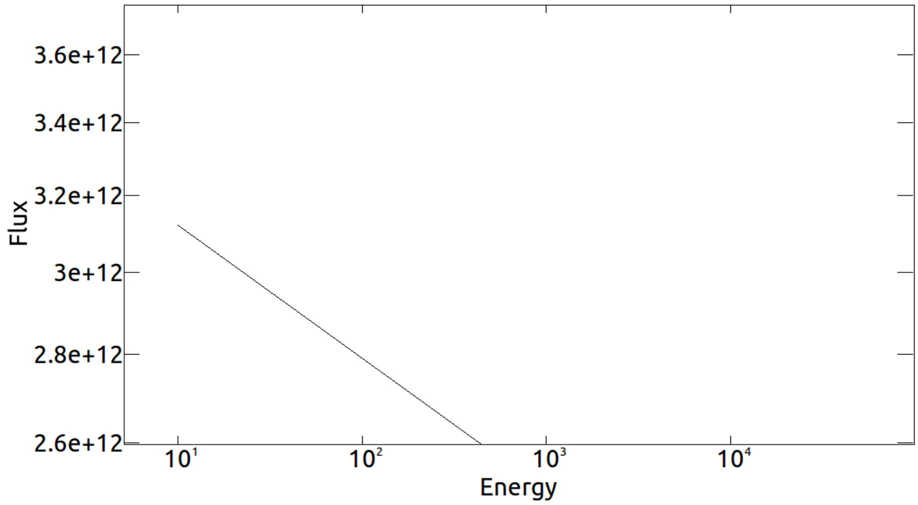


Figure 3.4: Flux vs photon energy (eV) from the dipole magnet in the CR for O.15 GeV.

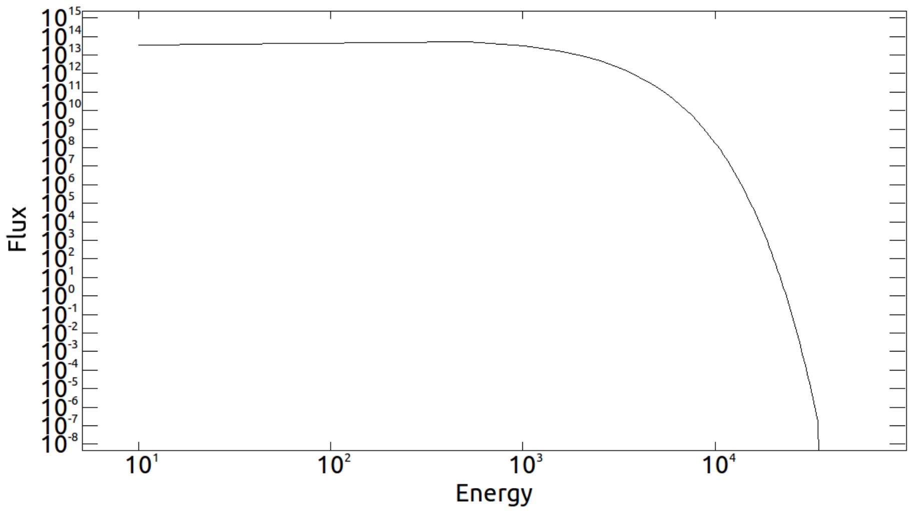


Figure 3.5: Flux vs photon energy (eV) from the dipole magnet in the CR for 1.5 GeV.

3.4.2 Radiation from a wiggler

Wiggler period λ_u, K parameter, ring energy E, number of magnetic poles are important parameters for the calculation of wiggler radiation characteristic. The photon flux of the radiation emitted by the wiggler is the same as the one of the

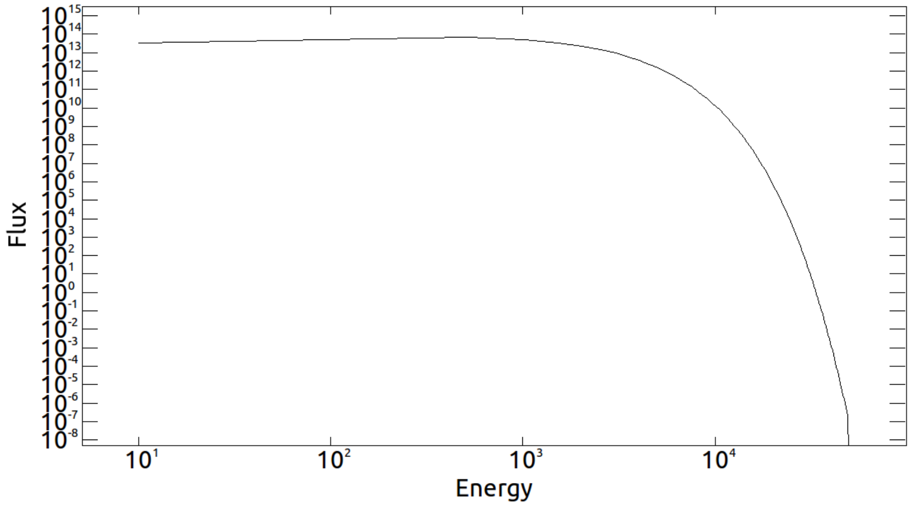


Figure 3.6: Flux vs photon energy (eV) from the dipole magnet in the CR for 1.79 GeV.

bending magnet but it is more intensive by a factor N_p, where N_p is the number of poles within the wiggler. It has a broader spectrum than the one of the bending magnet in Figures 3.7, 3.8, and 3.9.

wiggler Parameter	Value	Unit
Energy	$0.15 - 1.79$	GeV
Bρ	0.60	T m
Gap Value	40	mm
Periodic Length	160	cm
Max. Dispersion	1.708	m
Number of periods	1	

Table 3.3: Wiggler parameter of the Combiner ring.

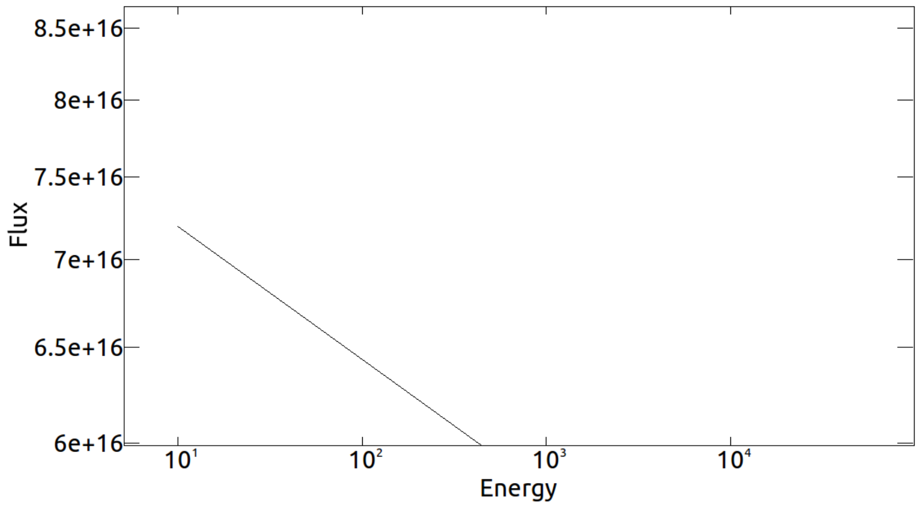


Figure 3.7: Flux vs photon energy from the wiggler magnet in the CR for 0.15 GeV.

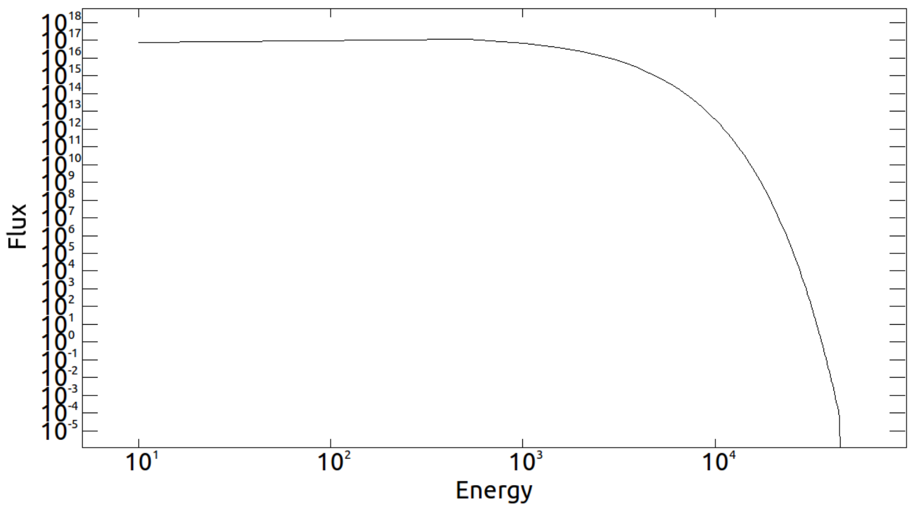


Figure 3.8: Flux vs photon energy from the wiggler magnet in the CR for 1.5 GeV.

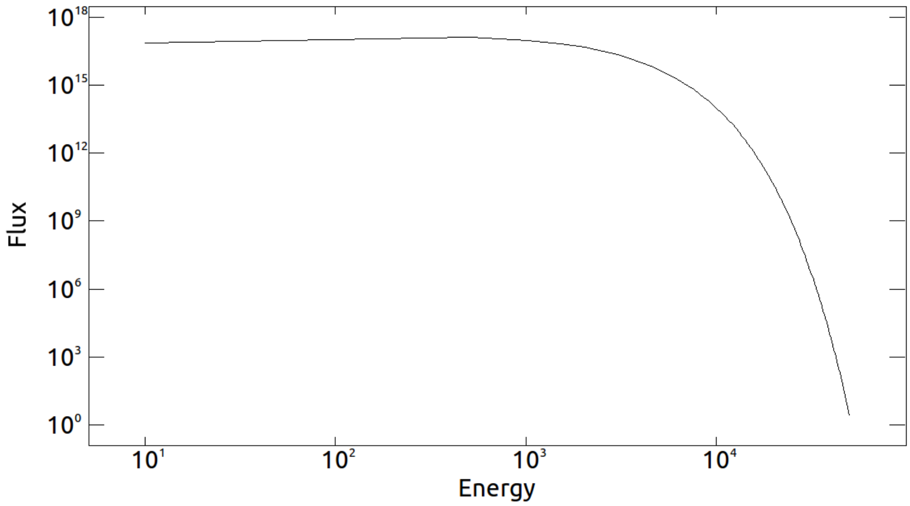


Figure 3.9: Flux vs photon energy from the wiggler magnet in the CR for 1.79 GeV.

3.5 Diffraction Limits

Diffraction is one of the phenomena which has an impact on the geometric properties of the photon beams and can thus impair the quality of the optical image [44], [45]. The diffraction phenomena of light is known as Fraunhofer diffraction and gives an expression of the distribution of the light diffracted through the optical system [46]. It concern travelling light that can propagate in free space, which needs a preferred plane such as lens.

A light source is called diffraction limited if the emittance of the electron beam is smaller than the one of the photon beam. Emittance of the light beam is given by

$\varepsilon_{photon} = \lambda \div 4\pi$

where λ is the wavelength of light. For instance the typical bending radiation peaks for a 0.15 GeV light source at wavelength of around 1.2 nm, or energy 1 keV. This means that for diffraction limited light source, the emittace would have to be below 0.1 nm rad. according to this emittance formula. Small beam sizes cannot be measured using direct imaging of the synchrotron light because of diffraction limitations [47]. The diffraction calculation shows also how the image contrast would be better.

So, the image resolution is limited by this diffraction and depends on the wavelength and on the geometry of the optical set up.

CHAPTER 4

EXPERIMENT AT CTF3

4.1 Experimental line

4.1.1 Description of the experimental set-up

The aim of this work is to make time resolved measurements with a good time resolution from different places in the ring. In CTF3, the CR is used to combine four 140 ns bunch trains where the individual trains are (or 140 ns) apart from each other. In bunch manipulation operations, the CR electron beam energy is 150 MeV.

There are two dipole magnets which produce SR via two different viewports for longitudinal and transverse beam profile measurements in the CR (see figure 4.1). One viewport has dispersion function while the other one has no dispersion function.

The emitted SR is extracted out of the vacuum system by a flat surface mirror. First, the mirror is located at 14 cm away from the first bending magnet and the other one is located at 17 cm from the second bending magnet.

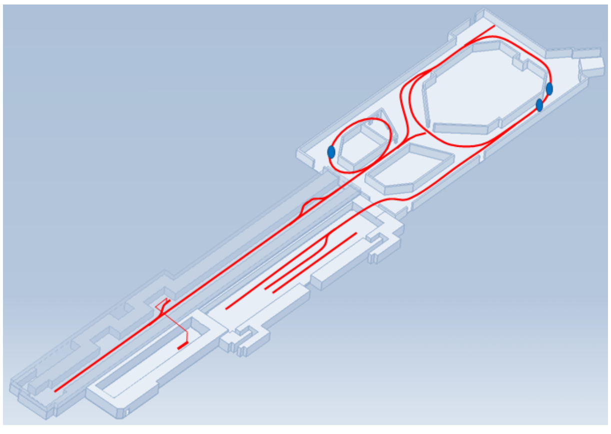


Figure 4.1: Synchrotron Radiation in the Delay Loop and Combiner Ring.

Incoming light from the two different viewports is transmitted to the synchrotron light laboratory using telescope arrangement (for instance lenses, mirrors) which is placed along the line. Each lens will absorb about 10 percent of the incident light. The camera is placed inside the laboratory in order to protect it from radiation.

The total length optical line is 10.848 m between the first dipole magnet and the camera room while the other optical line is 10.17 m between the second dipole magnet and the camera room. These two optical lines are called MTV0496 (for no dispersion measurement) and MTV0451 (for dispersion measurement).

Between the CR tunnel and the synchrotron light laboratory, a hole has been made to transmit the light from the CR to the laboratory. In front of the hole and inside the machine, a flipping mirror is used to select the light from either MTV0451 or MTV0496.

4.1.2 Modification of the design

The design of the optical line was made in the past thanks to ZEMAX software. The position of the mirrors and lenses are fixed by this software. These two optical lines were designed more for focusing than for imaging. In fact, they were planned to be used first for longitudinal beam profile measurements using a streak camera.

The design had then to be modified in order to do imaging and to get a magnification of 1 for good spatial resolution (see figure 4.2).

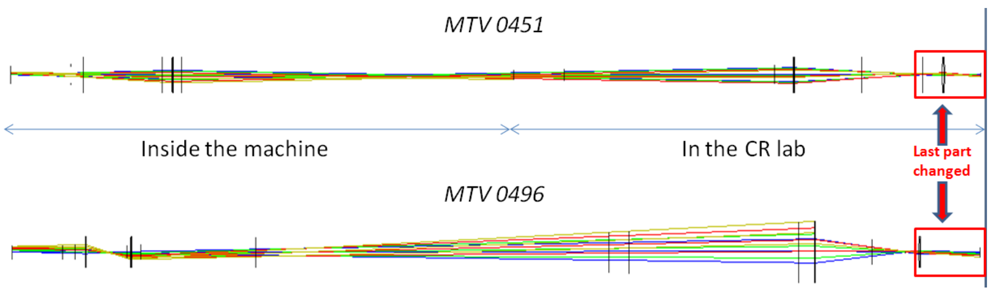
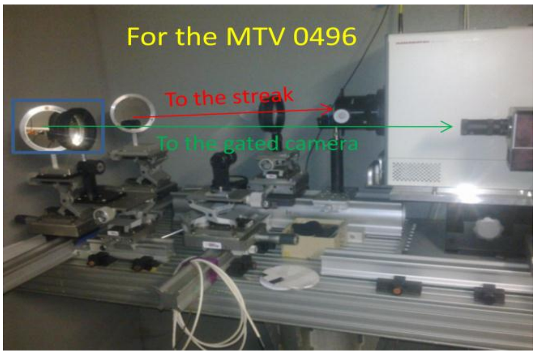


Figure 4.2: Figure of no dispersion and dispersion.

One mirror was thus added in order to transport the light through the gated camera instead of the streak camera. One lens (with a focal length of 160 mm) was also added upstream the camera in order to do proper imaging with the wanted magnification. Note that the distance of the last lens and of the camera is not the same for two optical lines in order to get a magnification of 1 for both of them (see figure 4.3)



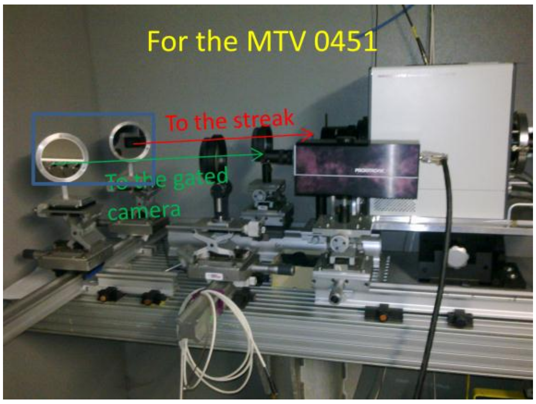


Figure 4.3: Synchrotron light laboratory with streak and gated cameras (left: set-up for the MTV0451 line and right: set-up for the MTV0496)

Object	Distance
Mirror	$25cm$
Lens (f'=160mm)	$65cm$
Mirror	$4cm$
Lens (f'=160mm)	$873.443mm$

Table 4.1: List of the elements of the optical line from MTV0451 to the Streak camera lab.

The alignment of the mirrors and lenses were performed thanks to a laser as it is shown in figure figure 4.3. The support of the lenses, of the mirrors and of the laser allow to move accurately the transverse and vertical positions (relatively to the direction of the synchrotron light) of these elements. Moreover, the support of the laser allows also to do pitch and roll motions.

4.2 Cameras in the laboratory

In the CTF3 CR, a streak camera is used to analyze in detail the longitudinal behavior of the electron bunches especially after the recombination process. The

Figure 4.4: Alignment of the mirrors and lenses with a laser.

time resolution of such a system is down a few picoseconds (see figure 4.5).

The operating principle of the streak camera is shown in figure 4.5. Incident photons from the radiation are transferred by the optics to a photocathode where the photons are converted to electrons via the photoelectric effect. These electrons are accelerated longitudinally and deflected using time synchronized in the streak camera. This deflecting field converts the time information much easier to analyze time resolved measurements. After this step, applying high voltage, the electrons enter the Micro Channel Plate (MCP) which produces a charge. Then, the signal from the electrons is sent to the phosphor screen where the light is converted again into a voltage signal.

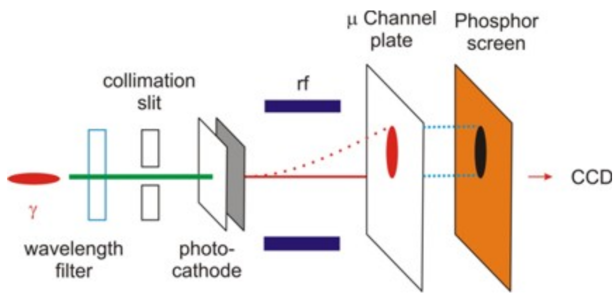


Figure 4.5: Principle of streak camera [20].

On the other hand, an intensified gated camera is used to analyze in detail the transverse profile measurements of the beam. It is called an intensified camera since it allows getting an image even with very low light. This can be the case when the optical lines are very long (around 10 m) since 10 percent of the incident light is lost on each lens.

It is called a gated camera since it allows taking an image integrated in a very small range of time. In our case, it is possible to take an image with a minimum gate of 100 ns, which enable to measure the transverse beam size turn by turn. A timing signal can be plugged into the camera in order to trigger it with the beam, that is to say every 1.2 s. Then, this timing can be delayed in order to measure the transverse beam size turn by turn.

4.3 Timing scheme

A fast gated, image intensified CCD camera is used to diagnostic the beam in CTF3. Although the system does not produce high resolution images, a minimum gating time of 5 ns provide good diagnostic potential. A fast gate module can pick up a signal of one small train along the bunch train. The gate module was attached to the adjustable measurement system.

An oscilloscope is often used for time controlled measurements, triggered by the gated camera timing system. Two pulses are prepared, which are complementary to each other. They are shifted by cable delay. The delay time determines the gated duration. The length of the delayed cable was adjusted to determine the duration of the gate.

The timing CX.SSTREAK2 was chosen to trigger our gated camera. In fact, it can be delayed from 0.2 ns up to several microseconds. This timing was controlled via Matlab by using the commands:

-JSetCoValueFesa('CX.SSTREAK2-CTTIM','Setting','delay', x) to control the rough timing where one steps corresponds to 50 ns

-JSetCoValueFesa('CX.SSTREAK2-F','CCV','value',x) to control the fine timing where one step corresponds to 0.2050 ns

For the rough and fine timing, it has been checked that one step corresponds well to respectively 50 ns and 0.2050 ns by controlling the timing via Matlab and looking at the delay observed at the oscilloscope (see Figures 4.6, 4.7, 4.8)

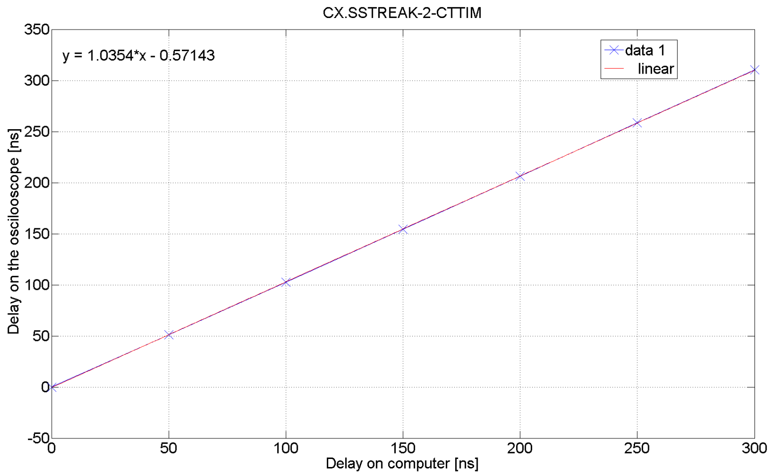


Figure 4.6: Rough timing of CX.SSTREAK2.

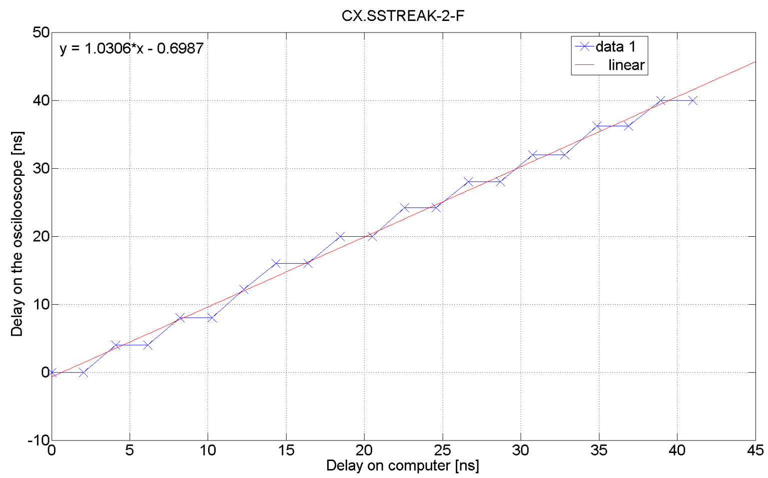


Figure 4.7: Fine timing of CX.SSTREAK2.

It was observed some jitter of the timing signal of about 5 ns, which is small compared to the minimal gate length of our gated camera which is of 100 ns. This timing could then definitively be used for our measurements.

4.4 Matlab GUI program

A Matlab program was made with a user interface (GUI) to scan the transverse beam size all along the pulse in the CR. For one measurement of the beam size, the process is the following. The program takes a picture of the beam, then

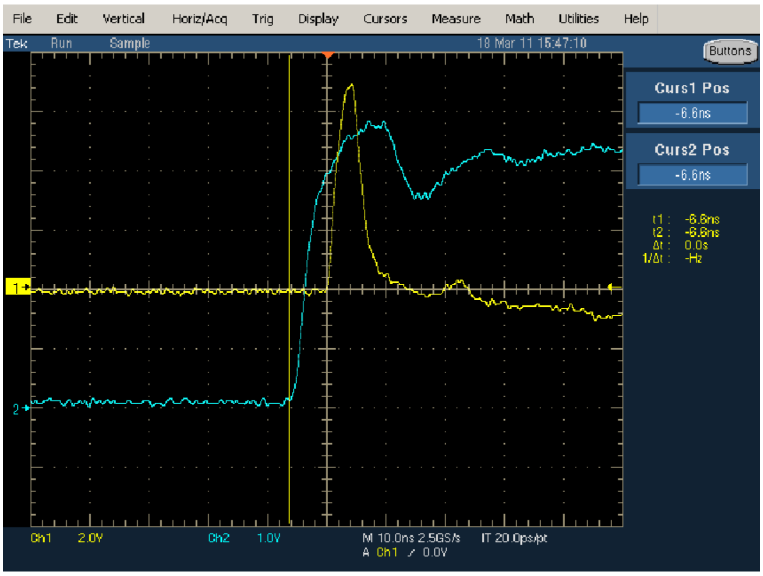


Figure 4.8: Timing signal observed on the oscilloscope.

performs horizontal and vertical projections of this picture versus the position of the beam, then performs a Gaussian fit of these projections in order to deduce the horizontal and vertical beam sizes (one sigma).

In the user interface, there are several terms which need to be explained in order to understand how this program works.

Firstly, the editable field 'gate length' refers to the delay (in ns) of the timing we want to apply in order to scan the beam size all along the pulse.

Secondly, the editable field 'delay number' refers to the number of time we want to apply this delay. Then, the editable field 'average number'refers to the number of averages we want to perform at every delay in order to have better statistics and to smooth the Gaussian of the horizontal and vertical projection.

Finally, the 'Back. Sub' button is used to perform a subtraction of the background from the raw data. For that, the gun of the machine was stopped in order not to have a beam and to thus be able take an image of the background. Thanks to this subtraction, the offsets of the horizontal and vertical projections are suppressed.

After that all the parameters are set by the user, the scan can be launched by pushing the 'Scan' button.

Moreover, the user can launch again another scan and select which turn he wants to scan. Like that, the different turns are displayed on the same graph where the program automatically delays the timing signal in order to go from a turn to another one.

Note that it is also possible to do only one acquisition of the beam size by pushing the 'One shot' button.

After that the scans are performed, it is possible to save automatically the data by pushing the button 'Save data'. The raw data (image of the beam) and the data of the table (horizontal and vertical beam size versus time) are saved. The program will ask to give the file name for this data and the repertory where the user wants to save the data. Then, he will automatically add incremented numbers after the file name in order to save all these data.

On the top of the GUI, there are also three 'Popupmenu' used for display. After that the scans are performed, it is possible to select again which turn ('Select turn'), which delay ('Select delay') and which average ('Select average') the user wants to display.

4.5 Measurements performed

We got a beam a little bit late and we thus decided to perform measurements only with the MTV0451 line. The gain of the intensified camera was adjusted at each measurement in order not to saturate the camera and to have in the same time enough light.

The gate of the camera was set to 100 ns and we configured the Matlab program to delay the timing signal of also 100 ns. In this way, we could measure the transverse beam size turn by turn. In fact, there is a time difference of 280ns between each turn while the pulse itself has a length of 140ns.

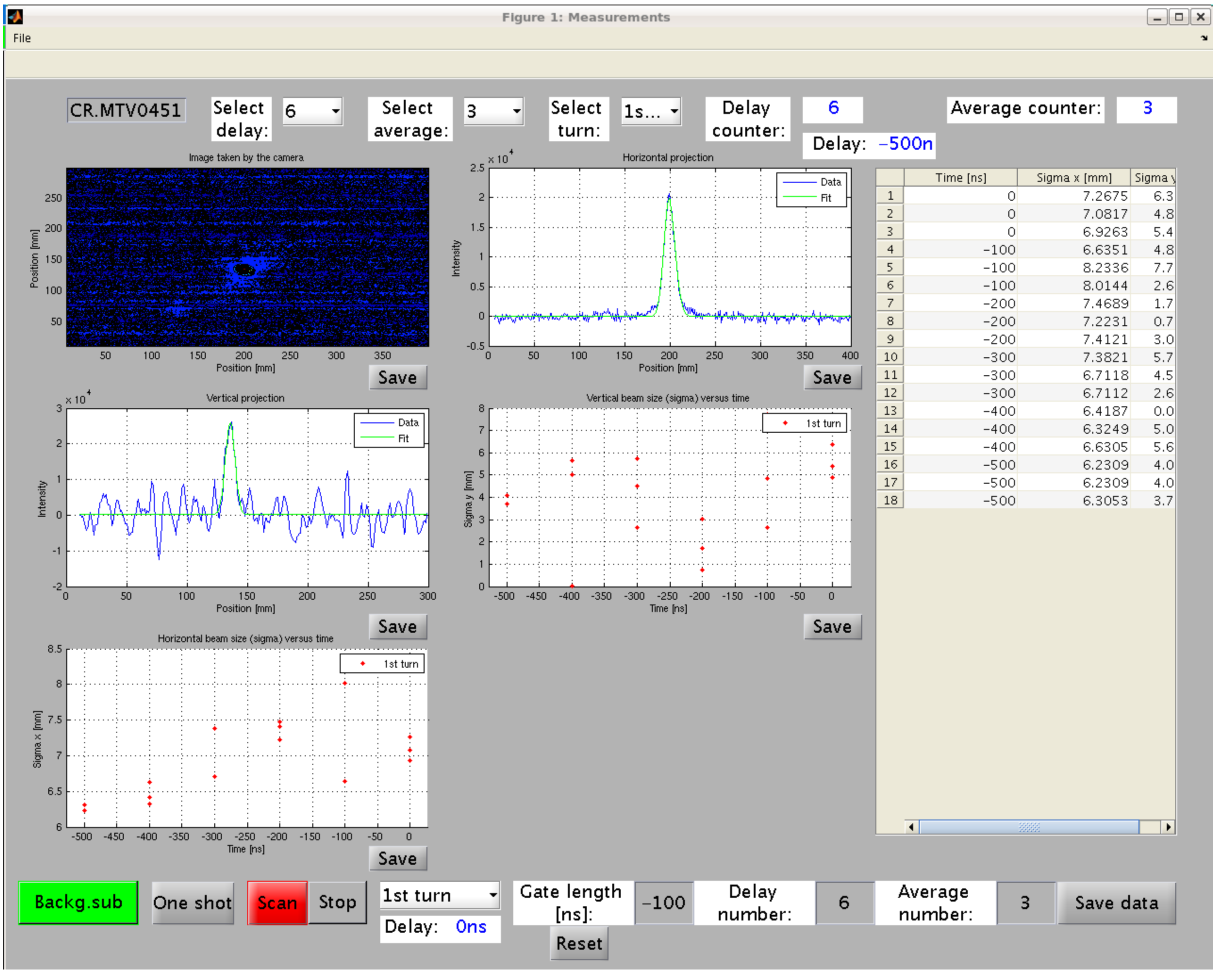


Figure 4.9: Principle of the Matlab GUI program for time resolved measurements.

4.5.1 Background subtraction

We have first checked that the background subtraction was well working. Data of the fourth turn was taken. Figures 4.10 and 4.11 show the horizontal and vertical projections respectively without and with background subtraction. The signal to noise ratio is much higher with background subtraction than without and consequently the fit is much more accurate. It is thus very important to perform this background subtraction.

Note that the noise is higher for the vertical projection since there are lines of noise observed in the image of the beam (see Figure 4.12). This noise comes from the electronics of the camera.

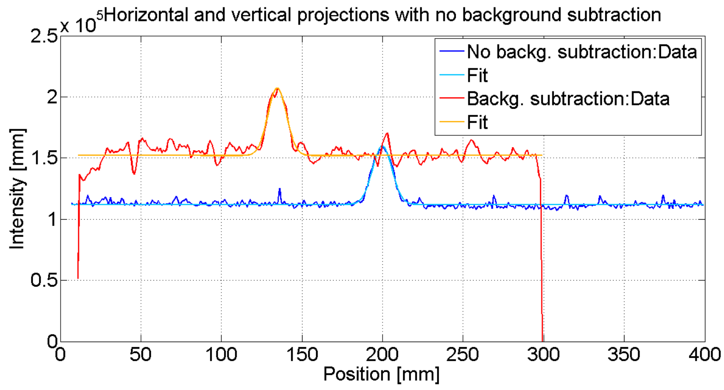


Figure 4.10: Horizontal and vertical projections (fourth turn) with no background subtraction

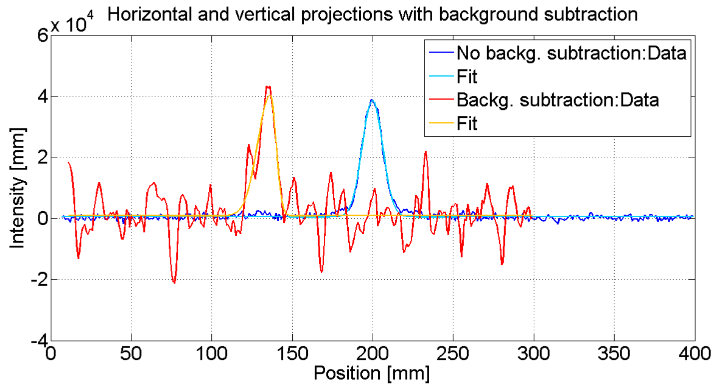


Figure 4.11: Horizontal and vertical projections (fourth turn) with background subtraction

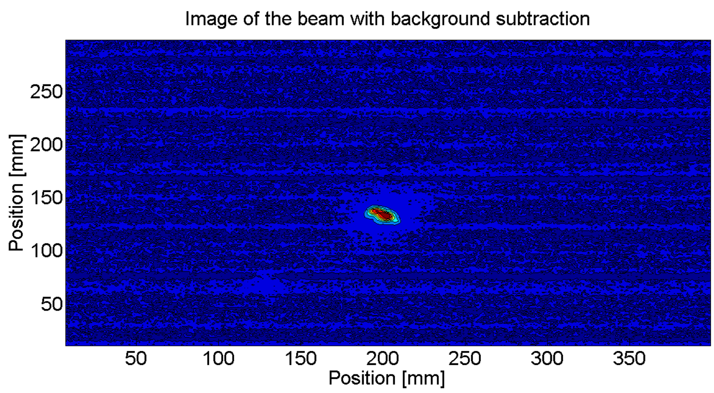


Figure 4.12: Image of the beam with background subtraction

4.5.2 Measurements

The beam image was obtained by averaging the CCD counts of the pixels. As shown in below figures,beam image obtained by the beam profile monitor for each turn. We did simulation by matlab programme. The beam were measured by changing the delay time with constant gate time of 140 ns. Beam size increases for every turn

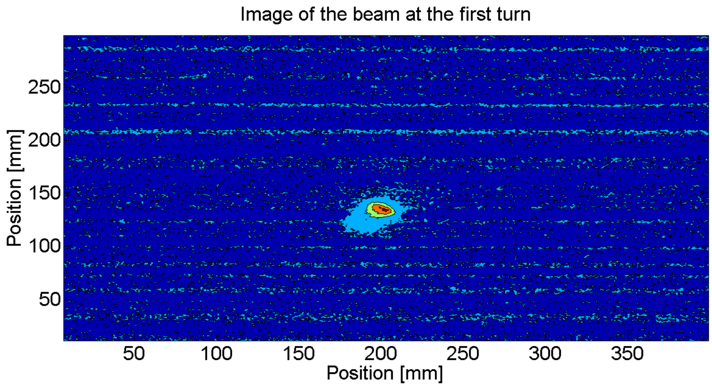


Figure 4.13: Beam image obtained by the beam prifile monitor for first turn.

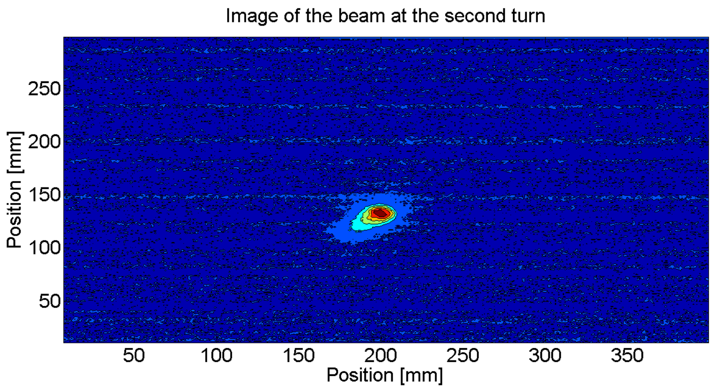


Figure 4.14: Beam image obtained by the beam prifile monitor for second turn.

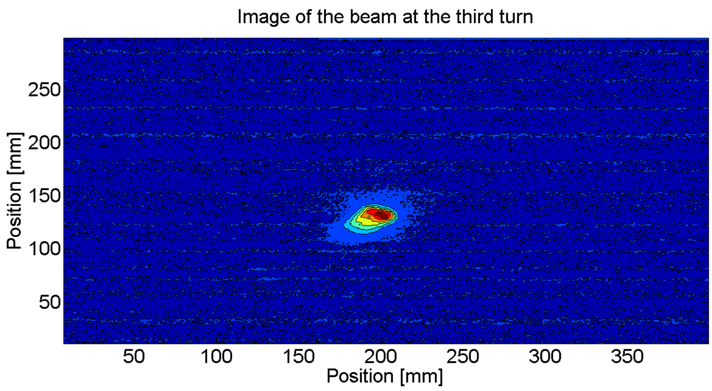


Figure 4.15: Beam image obtained by the beam prifile monitor for thirdh turn.

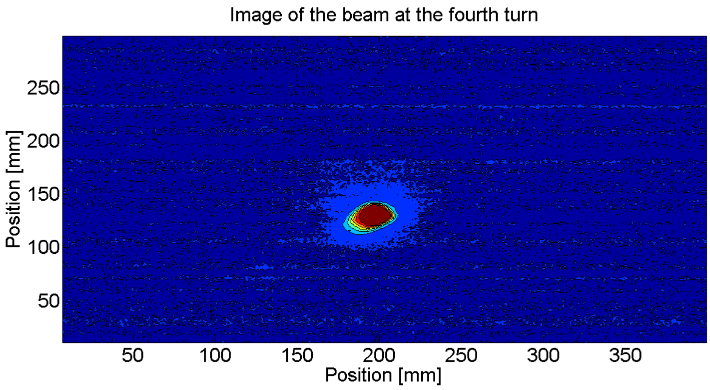


Figure 4.16: Beam image obtained by the beam prifile monitor for fourth turn.

The direct monitoring of the electron beam image is needed to decide the horizontal and vertical beam projection as well as the beam sizes and it can provide a lot of information on the electron beam (Figure 4.17 , 4.18).

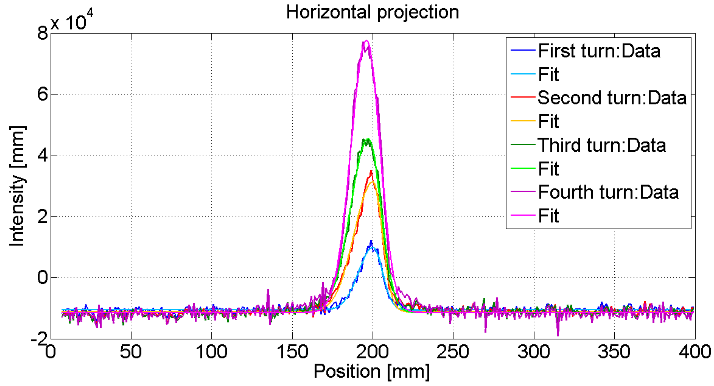


Figure 4.17: Horizontal projections for 4 turns.

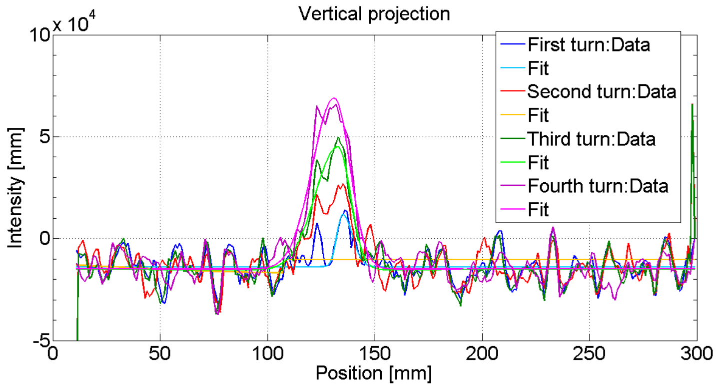


Figure 4.18: Vertical projection for 4 turns.

4.5.3 Measurements with uncombined beam

Firstly, the operators of the control room set the parameters of the machine in order to have an uncombined beam. The beam did not pass through the Delay Loop and there were no combinations done in the Combiner Ring (no increase of the current). However, the operators set the machine so that the beam did multi-turns in the combiner ring (five turns) and the goal of these measurements was to check that there was no significant increase of the transverse beam size from one turn to another one. In fact, the beam size can increase significantly if the beam trajectory does not describe a close orbit. Measurements were then performed with the gated camera in order to observe the five turns. For that, the timing signal was constantly delayed by 100ns up to 1700ns. Every 280ns, we passed from one turn to the next one. Figure 4.19 shows the results of these measurements for the horizontal and vertical beam sizes. No significant increase of the beam size was observed from one turn to the other one, which shows that the setting of the machine was good.

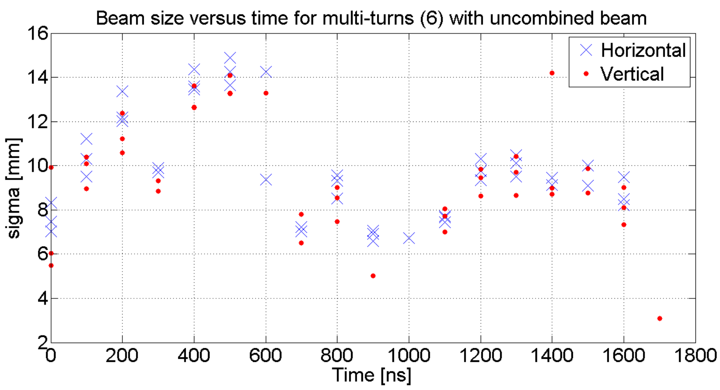


Figure 4.19: Beam size versus time for multi turns (6) with uncombined beam.

4.5.4 Measurements with combined beam

Measurements of the transverse beam size with a gated camera in the CR with combined beam (4 turns). These figures shows that the horizontal beam size seems to increase with the number of turns.

If investigated Figure 4.20, there are triple beams for the third turn and fourth

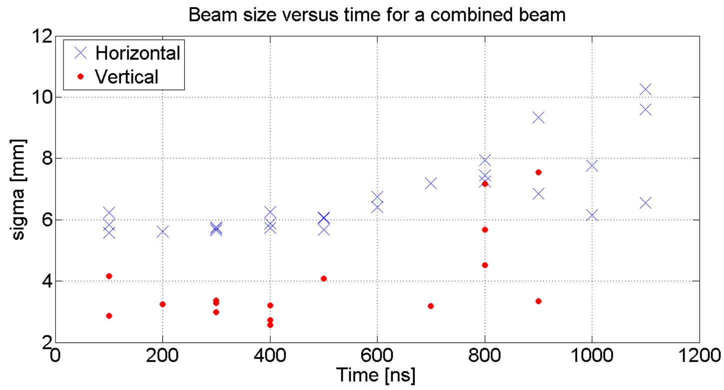


Figure 4.20: Beam size versus time for a combined beam.

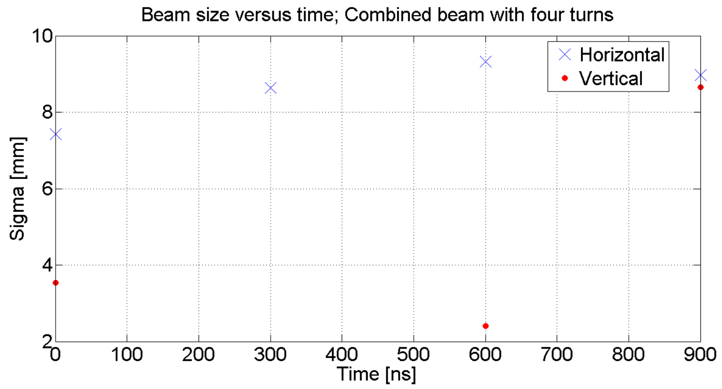


Figure 4.21: Beam size versus time for a combined beam.

turn. This isnot good result for beam size measurements. That is, recombination process is not good. Figure 4.21 is good result for recombination process.

When performing these measurements, it is very useful to look at the Beam Position Monitor (BPM) and Beam Position Intensity (BPI) in order to correlate these data with the ones of the gated camera. BPMs are more accurate for position measurements than BPIs while BPIs are more accurate for intensity measurements. Figure 4.22 shows the current (in Amperes) measured by the BPI located just downstream the bending magnet creating the synchrotron light for the line MTV0451. The five turns of the uncombined beam can be clearly seen. The current measured for each turn is almost the same, which confirms that the beam was well tuned in the combiner ring (no increase of the beam size turn by turn).

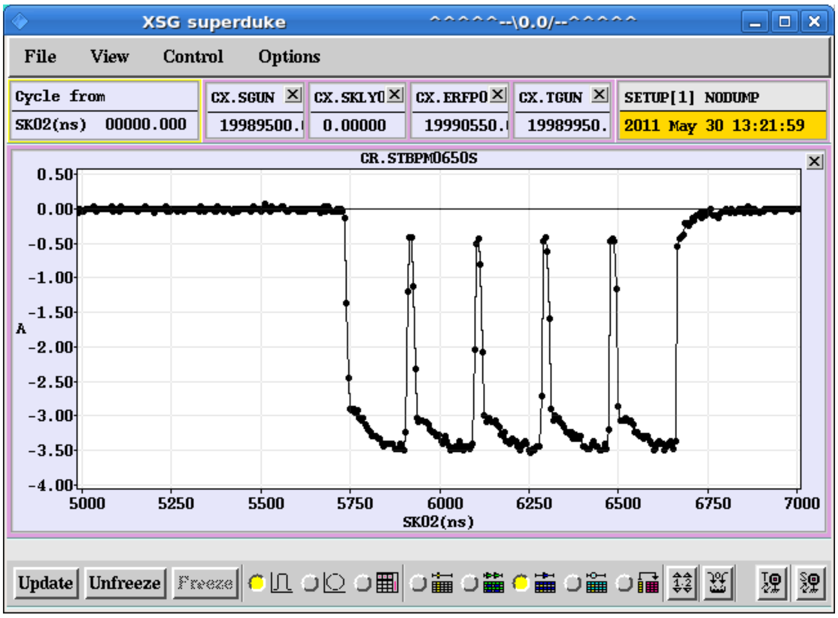


Figure 4.22: Beam position intensity located just downstream the bending magnet creating the synchrotron light for the line MTV0451.

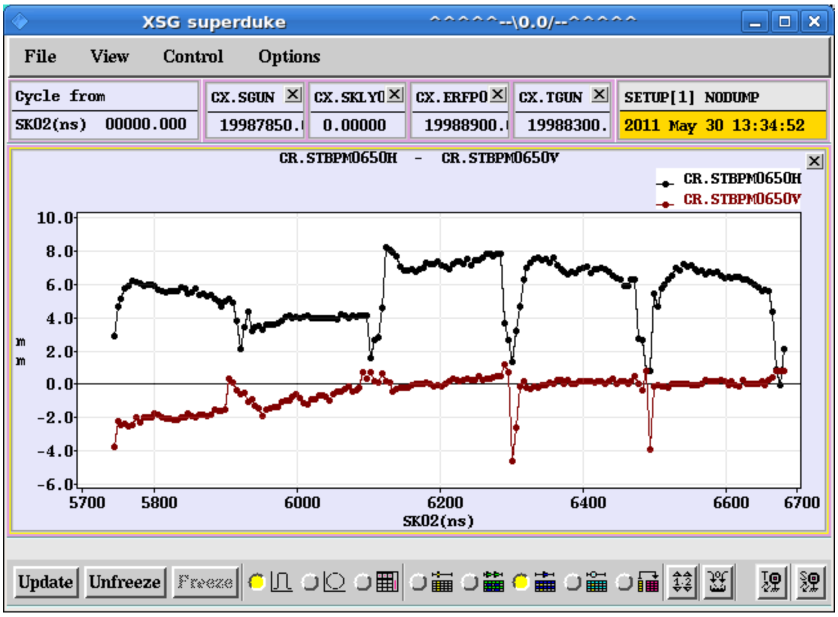


Figure 4.23: Shows the temporal evolution of the horizontal and vertical positions of the beam centroid as measured by the BPM0650 located close to our synchrotron light source.

Figure 4.23 shows the horizontal and vertical positions (in mm) measured by the BPM located at the same location than the BPM shown previously. The five turns of the uncombined beam can be also clearly seen. From one turn to another one, the beam orbit can varies up to 4 mm, which is reasonable but shows that the beam does not describe a perfect closed orbit in the CR. Some small optimizations of the machine have to be done by the operators. An improvement of the Matlab program will be to normalize the intensity measured by the gated camera by the one measured by the BPIs. Also, an online plot comparing the intensity and position measured by the gated camera with ones measured by the BPIs should be added in the user interface. This will provide a great tool for accurate analysis of the beam position and beam intensity in addition to the existing analysis of the beam size.

CHAPTER 5

CONCLUSION

We measured time resolved beam size with gated camera, with focusing method. By measuring the beam size at different time, we measured the time varying of beam. The beam size increased for each turn.

We have developed a new diagnostic tool for transverse beam size measurements in the CR with a gated camera. This allows measurements turn by turn thanks to the gate of the camera which can be as small as 100 ns. This allows also beam position and beam intensity measurements. Since these diagnostic tool was shown to work well, a gated camera with a gate as small as 5 ns will be bought in the future to enable performing measurements all along a pulse. We have also developed a Matlab program with a user interface allowing an automatization of these measurements. This program can be used by all the operators of the control room since it is automatized. Also, it can be used for measurements with other gated cameras which are planned to be installed after the combiner ring (in CLEX) for emittance measurements. One improvement of the matlab program will be to add correlations between the position and intensity measured by the gated cameras with the ones measured by the BPMs and BPIs. Then, this program will provide a very accurate analysis of beam position, intensity and size. These are very important informations for the operators of the machine.

Finally, for this experiment, high precision wave optics based calculations of synchrotron light characteristics were applied to determine the electron beam parameters to sufficiently high accuracy. The diagnostic beamline served as a good platform for beam profile. The lens method gave the possibility to measure beam size.

REFERENCES

[1] G. Arnison et al., "Experimental Observation of Lepton-Pairs of Invariant Mass around 95GeV/c^2 at the CERN SPS Collider", Physics Letters B 126, (1983), 398.

[2] S. Myers, "LEP2 : Present and Future Performance and Limitation", Proceeding of the Particle Accelerator Conference, Vancouver, BA, Canada, (1998), 1.

[3] L. Evans et al., "Large Hadron Collider Status and Plans", Proceeding of the Particle Accelerator Conference, Vancouver, BA, Canada, (1998), 56.

[4] V. Bharadwaj et al., "Status and Future of the TEVATRON", Proceeding of the Particle Accelerator Conference, Dallas, Texas, USA, (1996), 396.

[5] http://lepewwg.web.cern.ch/LEPEWWG/DOWNLOAD 0412251v1.

[6] G. Abbiendi, et al., "Search for the Standard Model Higgs Boson at LEP", (The ALEPH, DELPHI, L3, and OPAL Collaborations, The LEP Working Group for Higgs Boson Searches), Physics Letters B 565:61-75,2003, arXiv:hep-ex/0306033v1.

[7] S. Myers, "LHC Commissioning and First Operation", Proceeding of the International Particle Accelerator Conference, Kyoto, Japan, (2010), 6.

[8] K. Jakobs, "Physics at the LHC and sLHC", Nuclear Instruments and Methods in Physics Research A (K. Jakobs, Nuclear Instruments and Methods A (2010).

[9] K. Yokoda, "Status of the International Linear Collider", Proceeding of the International Particle Accelerator Conference, Kyoto, Japan, (2010), 2378.

[10] http://www.symmetrymagazine.org/cms/?pid=1000156.

[11] http://clic-study.org/ (date of access:20.06.2010).

[12] P.V. Baal, A course in a field theory, fall 1998.

[13] H.Braun et al., "for the CLIC Study Team", CERN-OPEN-2008-021, CLIC-Note 764, 2008, available from http://clic-meeting.web.cern.ch/clic-meeting/clictable2010.html.

[14] A.M. Sessler and S.S. Yu, "Relativistic Klystron Two Beam Accelerator", Physical Review Letters 54, (1987) 889, W. Schnell, "A Two Satge RF Linear Collider using a Superconducting Drive Linac", CLIC note 13, CERN LEP-RF 86/06 (1986).

[15] http://clic-study.web.cern.ch/CLIC-Study (date of access: 20.04.2011).

[16] I. Wilson, "Proposal for Future CLIC studies and a New CLIC Test Facility (CTF3)", CERN/PS 99-047 (LP), CLIC note 402 (1999).

[17] http://ctf3.home.cern.ch/ctf3/New_description. (date of access: 05.09.2010)

[18] R. Bossart et al., "ctf3 design report", slac-pub-9672.

[19] F. Sannibale, " Driving Beam Delay Loop Design for CTF3", CTF3 Note 023 (2001).

[20] C. P. Welsch, H.H. Braun, E. Bravin, R. Corsini, S. Döbert, T. Lefevre, F. Tecker, P. Urschütz, " Longitudinal Beam Profile Measurements at CTF3 using a Streak Camera", Journal of Instrumentation 1 PO9002 (2006).

[21] P. J. Duke, "synchrotron radiation production and properties", Oxford science publications, newyork, 2000, page:1.

[22] J.D. Jackson, "V Classical Electrodynamics", Second Edition., Chapter 14 ,John Wiley, Newyork, London 1962.

[23] D.J. Griffiths, "Introduction Electrodynamics", 3rd Edition, Chapter 11, 460 - 464, Prentice Hall, 1999.

[24] D.H. Tomboulian, P.L. Hartman: Physical Review Letters 102, 1423 (1956).

[25] D.H. Tomboulian, D. E Bedo: Journal of Applied Physics 29, 804 (1958).

[26] A.A. Sokolov, J. M. Ternov: Synchrotron Radiation Pergamon Press, London (1968).

[27] K. Wille, Joint Universities Accelerator School 2008, University of Dortmund, lecture notes, 2008 session.

[28] H. Wiedemann, "synchrotron radiation", 2003, springer, page 58.

[29] H. Winick, "High energy synchrotron radiaition sources", Stanford Linear Accelerator Center, Stanford Synchrotron Radiation Laboratory, Stanford CA 94309, U.S.A., SLAC- PUB-6281.

[30] H. Winick, "Fourth Generation Light Sources", Stanford Linear Accelerator Center Stanford, CA 94309, USA.

[31] H. Winick, "Synchrotron Radiation Sources, a primer" , World Scientific,1994 page 7.

[32] R. Haensel, "Status of ESRF at the end of 1990", Nuclear Instruments & Methods A303 (1991) pp 405-412.

[33] G. K. Shenoy, Status of Advanced Photon Source Project, Nuclear Instruments Methods A303 (1991) pp 413-420.

[34] H. Kamitsubo, "8 GeV Synchrotron Radiation Facility Project in Japan": JAERI-RIKEN Spring; 8 Project, Nuclear Instruments Methods A303 (1991) pp 421-434.

[35] S. K. Sinha, "Application of synchrotron radiation next term techniques to nanoscience", Radiation Physics and Chemistry 70 (2004) 633640.

[36] Å. Kvick, "our future light sources a brief introduction MAX lab", ESRF, Grenoble Prof. J. Als-Nielsen et al., University of Copenhagen, 2004.

[37] Å. Andersson, M. Eriksson, O. Chubar, "Beam profile measurements with visible synchrotron light on MAX-II, proceedings of the European Particle Accelerator Conference, EPAC 1996, Barcelona, Spain, 10-14 June 1996.

[38] A. Andersson, J. Tagger, "Beam profile measurement at MAX", Nuclear Instruments and Methods Vol. A364, 4 (1995).

[39] K. Wille, "The physics of particle Accelerators an introduction", Oxford university press,2000.

[40] M. Ferianis, "Optical techniques in beam diagnostic", Proceeding of the EPAC '98.

[41] T.Lefevre, "The CLIC Test Facility 3 Instrumentation",Proceedings of BIW08, Tahoe City, California WEIOTIO02.

[42] A. Stella, A. Ghigo, F. Marcellini, A. Zolla Frascati, "Design of a Beam Position Monitor for the CTF3 Combiner Ring", Accelerator Division, July 4, 2002 Note: CTFF3-008, INFN - LNF .

[43] http://radiant.harima.riken.go.jp/spectra/ (Date of access:17.03.2010)

[44] H. Winick, 4 Generation Light Sources, PAC97.

[45] A. Ogata, "On optical resolution of beam size measurements by means of synchrotron radiation", Nuclear Instruments and Methods in Physics Research A301 (1991) 596-598, North-Holland.

[46] A. Hofmann and F. Méot, "Optical resolution of beam cross-section measurements by means of synchrotron radiation", Nuclear Instruments and Methods in Physics Research Volume 203, Issues 1-3, 1 December 1982, Pages 483-493.

[47] P. Chevtsov et al., “Non-invasive energy spread monitoring for the JLAB experimental program via synchrotron light interferometers”, Nuclear Instruments and Methods in Physics Research A 557 (2006) 324327.

[48] H. Braun et al., “CLIC 2008 parameters”, European organization for nuclear research, cern AB department, CLIC-Note-764.

[49] G. Geschonke et al., “ CTF3 Design Report”, European organization for nuclear research, CERN/PS 2002-008 (RF),CTF3 Note 2002-047 LNF-02/008 (IR).

[50] R.W. Assmann et al., “ A 3 TeV e^+e^- Linear Collider Based on CLIC Technology”, CERN 2000-008.

[51] https://www.zemax.com (date of acces: 25.02.2011)

[52] M.Useaka et al., “Precise measurement of a subpicosecond electron single bunch by the femtosecon streak camera”, Nuclear Instruments Methods, A406 (1998) 371.

[53] http://verplant.org/history-geophysics/Wiechert.htm. (date of access: 25.02.2011)

[54] J.A. Clarke, “The science and technlogy of undulators and wigglers”, oxford science publications, 2004.

[55] D.D. Ivanenko, I. Pommeranchuk:physical reviews 65, 343 (1944).

[56] J.A. Clarke, “The science and technlogy of undulators and wigglers”, oxford science publications,2004, pg:2

[57] ESF Perspective Study 1977: Synchrotron Radiation a Perspective view for Europe. European Science Foundation, Strasbourg, France, 1977 (Joint ESRF-ILL Library).

[58] S.Y. Lee, “Accelerator Physics”, published by world scientific publishing Co. Pte. Ltd., 1999.

[59] H. wiedemann, “Particle Accelerator Physics 1”, second edition, springer 1965, page 29.

APPENDIX A
SYNCHROTRON RADIATION

A.1 Radiation by Moving Charges

The electromagnetic radiation is emitted when the charged particles are accelerated under electromagnetic field. This section describes the power radiated by moving charged particle using Lienard-Wiechert potentials in detail.

A.2 Lienard-Wiechert potentials of a moving charge

The derivation of the fundamental equations describing the emission of Synchrotron Radiation (SR) can be found in electrodynamic books [22, 23]. And also there are several articles on this topic[24, 25, 26].

The development of the theory of the Lienard - Wiechert potentials dates back to 1898 and it continued into the early 1900 [53]. The potentials were named after them. The Lienard-Wiechert potentials are used for the calculations of the energy loss of an electron moving on a circular path. It is solved directly from Maxwell's equations.

Before the radiated power is calculated, an electron moving with relativistic velocity on an arbitrary path is described. The description should include the field of a point charge on arbitrary motion.

A.3 Retarded Time and Position

The calculation of the electromagnetic radiation emitted by a moving charged particle needs a retarded time and position [27]. It was described by a system of coordinates shown in Figure A.1. t' is called retarded time which determines the time of emission of an electromagnetic waves according to observation time (t).

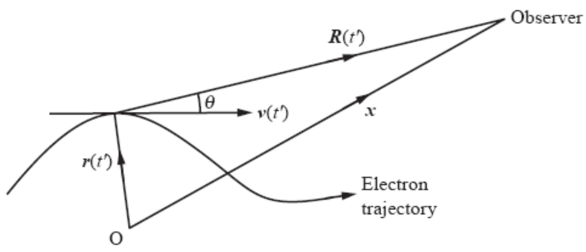


Figure A.1: Geometry of the electron and the observer [54].

These parameters are very important in order to explore how one changes with respect to the other. The relationship between the observation time and the retarded time is written according to figure (2.1).

$$
\begin{aligned}
\mid \overrightarrow{x} - \overrightarrow{r}(t') \mid &= c \mid t - t' \mid \\
t' &= t - \frac{\mid \vec{x} - \vec{r}(t') \mid}{c} \\
t' &= t - \frac{\mid \vec{R} \mid}{c}
\end{aligned} \tag{A.1}
$$

And where $\vec{R}$ is written;

$$\overrightarrow{R} = \overrightarrow{x} - \overrightarrow{r}(t') \tag{A.2}$$

According to this figure (1.1), the charge position is determined by the radius vector r(t). This vector is related to the radius vector of the observation $\overrightarrow{x}$.

$$\hat{n} = \frac{\vec{R}}{\mid \vec{R} \mid}$$

where $\hat{n}$ is the unit vector.

$$\overrightarrow{\beta}(t) = \frac{\dot{\vec{r}}(t)}{c} \tag{A.3}$$

$$\overrightarrow{\beta}(t') = \frac{\dot{\vec{r}}(t')}{c} \tag{A.4}$$

where β means the electron velocity relative to the speed of light. Also in Eq.(A.3) and (A.4). $\dot{\vec{r}}(t')$ is the velocity of the charged particle.

Let's define a scalar potential and a vector potential. These potentials are called Lienard-Wiechert potentials which are described using Maxwell's equations.

Scalar potential is ;

$$\vec{\phi}(\vec{x},t) = \frac{q}{4\pi\epsilon_0}\int \frac{1}{\mid \vec{x}-\vec{r}(t')\mid}\delta\left[f(t')\right]dt' \tag{A.5}$$

Vector potential is ;

$$\vec{A}(\vec{x},t) = \frac{\mu_0 q}{4\pi}\int \frac{\vec{V}(t')}{\mid \vec{x}-\vec{r}(t')\mid}\delta\left[f(t')\right]dt' \tag{A.6}$$

And since f(t') ;

$$f(t') = t' + \frac{\vec{R}(t')}{c} - t \tag{A.7}$$

The equations $(A.5)$ and $(A.6)$ are the Lienard-Wiechert potentials for a moving point charge.

Take the derivation of Eq.(A.7)

$$df/dt' = 1 + \frac{1}{c}(d\vec{R}/dt') \tag{A.8}$$

$$\begin{aligned} d\vec{R}/dt' = \frac{d\left[(\vec{x}-\vec{r}(t'))(\vec{x}-\vec{r}(t'))^{1/2}\right]}{dt'} &= \frac{1}{2\vec{R}}\left[-2(\vec{x}-\vec{r}(t'))\dot{\vec{r}}(t')\right] \\ &= -\frac{(\vec{x}-\vec{r}(t'))}{\vec{R}}\dot{\vec{r}}(t') \\ &= -\hat{n}\cdot(\vec{\beta}c) \end{aligned} \tag{A.9}$$

Equating result of this derivation with eqn (A.8) gives

$$df/dt' = 1 - \hat{n}(t')\cdot\frac{\vec{\beta}(t')c}{c} = \left[1-\hat{n}(t')\cdot\vec{\beta}(t')\right] \tag{A.10}$$

In Equations($A.5$) and ($A.6$) this integral,the delta function can be written as

$$\begin{aligned}\int g(t')\delta\left[f(t')\right]dt' &= \int g(t')\delta\left[f(t')\right]\frac{1}{\mid \frac{df(t')}{dt'}\mid}df \\ &= \frac{g(t')}{\mid \frac{df}{dt'}\mid} \end{aligned} \tag{A.11}$$

Now, electric field of a moving charge particle can find by using Eq.(A.12). E is built from Maxwell's first equation ($\vec{\bigtriangledown}\cdot$ B).

$$\vec{E} = -\vec{\bigtriangledown}\phi - \frac{\partial \vec{A}}{\partial t} \tag{A.12}$$

in order to find electric field, the first term in Eq.(A.12) is solved as the gradient of ϕ ($\vec{\bigtriangledown}\phi$)

$$\begin{aligned}\vec{\bigtriangledown}\phi &= \frac{q}{4\pi\epsilon_0}\int \vec{\bigtriangledown}(1/\mid x - r(t')\mid)\delta\left[f(t')\right]dt' \\ &= \frac{q}{4\pi\epsilon_0}\int \hat{n}\frac{\partial}{\partial \vec{R}}(\frac{1}{\vec{R}})\delta\left[f(t')\right])dt' \\ &= -\frac{q}{4\pi\epsilon_0}\int\left[\frac{\hat{n}}{\vec{R}^2}\delta[f(t')] - \frac{\hat{n}}{c\vec{R}}\frac{d}{df}\delta[f(t')]\right]dt' \end{aligned} \tag{A.13}$$

The second term in this Eq.(A.13) where I used integration by parts and also I have used the result of Eq.(A.10).Therefore

$$\begin{aligned} g(t') &= \frac{\hat{n}}{c\vec{R}} \\ \int g(t')\frac{d}{df}\delta\left[f(t')\right]dt' &= \int g(t')\frac{1}{df/dt'}\frac{d}{dt}\delta\left[f(t')\right] \\ &= -\left[\frac{1}{df/dt'}\frac{d}{dt'}\frac{g(t')}{df/dt'}\right] \\ \int g(t')\frac{d}{df}\delta\left[f(t')\right]dt' &= -\left[\frac{1}{(1-\hat{n}\cdot\vec{\beta}(t'))}\frac{d}{dt'}\frac{g(t')}{(1-\hat{n}\cdot\vec{\beta}(t'))}\right] \end{aligned} \tag{A.14}$$

Putting this result into Eq.(A.13) gives

$$\vec{\bigtriangledown}\phi = -\frac{q}{4\pi\epsilon_0}\left[\frac{\hat{n}}{(1-\hat{n}\cdot\vec{\beta}(t'))\vec{R}^2} + \frac{1}{c(1-\hat{n}\cdot\vec{\beta}(t'))}\frac{d}{dt'}\frac{\hat{n}}{(1-\hat{n}\cdot\vec{\beta}(t'))\vec{R}}\right] \tag{A.15}$$

Likewise, we can find the vector potential. Applying this to the time derivative of the vector. Second term($\frac{\partial \vec{A}}{\partial t}$) in this Eq.(A.12) where I have used the result of Eq (A.10)

$$
\begin{aligned}
\frac{\partial \vec{A}}{\partial t} &= \frac{\partial}{\partial t}\left[\frac{\mu_0 q}{4\pi}\int \frac{\vec{v}(t')}{\mid x - r(t') \mid}\delta\left[f(t')\right]dt'\right] \\
&= \frac{\mu_0 q}{4\pi}\int \frac{\partial}{\partial t}\frac{\vec{v}(t')}{\vec{R}}\delta[f(t')]dt
\end{aligned} \tag{A.16}
$$

Where $c = \frac{1}{\sqrt{\mu_0 \epsilon_0}}$, $\mu_0 = \frac{1}{c^2 \epsilon_0}$ in this expression the value of μ_0 is defined by the system of units and ϵ_0 is defined by the strength of the static electric field.

$$
\begin{aligned}
&= \frac{q}{4\pi\epsilon_0 c^2}\int \frac{\partial}{\partial t}\frac{\vec{v}(t')}{\vec{R}}\delta\left[f(t')\right]dt' \\
&= \frac{q}{4\pi\epsilon_0 c}\int \frac{\partial}{\partial t}\frac{\vec{v}(t')}{c\vec{R}}\delta\left[f(t')\right]dt' \qquad , \qquad with \qquad \vec{\beta} = \vec{v}(t')/c, \\
&= \frac{q}{4\pi\epsilon_0 c}\int \frac{\partial}{\partial t}\vec{\beta}(t')\delta\left[f(t')\right]dt' \\
&= \frac{q}{4\pi\epsilon_0}\frac{1}{c(1-\hat{n}\cdot\vec{\beta}(t'))}\frac{d}{dt'}\left(\frac{\vec{\beta}}{(1-\hat{n}\cdot\vec{\beta})\vec{R}}\right)
\end{aligned} \tag{A.17}
$$

A.4 The Electric Field of a Moving Charged Particle

Using the formula of electric field Eq.(A.12), we can derive the electric field at the point P. Finally, this results in this Eq.(A.15) and (A.17) can be combined to solve Eq.(A.12)

$$
\begin{aligned}
\vec{E}(\vec{r},t) = -\frac{q}{4\pi\epsilon_0}&\left[\frac{\hat{n}}{(1-\hat{n}\cdot\vec{\beta}(t'))\vec{R}^2} + \frac{1}{c(1-\hat{n}\cdot\vec{\beta}(t'))}\frac{d}{dt'}\frac{\hat{n}}{1-\hat{n}\cdot\vec{\beta}(t')\vec{R})}\right] \\
&- \left[\frac{1}{c(1-\hat{n}\cdot\vec{\beta}(t'))}\frac{d}{dt'}\left(\frac{\vec{\beta}}{(1-\hat{n}\cdot\vec{\beta})\vec{R}}\right)\right]
\end{aligned} \tag{A.18}
$$

which can be rewritten as

$$
\vec{E}(\vec{r},t) = \frac{q}{4\pi\epsilon_0}\left[\frac{\hat{n}}{(1-\hat{n}\cdot\vec{\beta}(t'))\vec{R}^2} + \frac{1}{c(1-\hat{n}\cdot\vec{\beta}(t'))}\frac{d}{dt'}\frac{\hat{n}-\vec{\beta}}{(1-\hat{n}\cdot\vec{\beta}(t'))\vec{R})}\right] \tag{A.19}
$$

In order to simplify the calculations in the Eq.(A.19) I define

$1 - \hat{n} \cdot \vec{\beta} = D$

and I rewrite this Eq.(A.19)

$$\vec{E}(\vec{r}, t) = \frac{q}{4\pi\epsilon_0} \left[\frac{\hat{n}}{D\vec{R}^2} + \frac{1}{cD} \frac{\partial}{\partial t'} \frac{\hat{n} - \vec{\beta}}{D\vec{R}} \right] \tag{A.20}$$

After replacing D, get derivative of second term in Eq.(A.20) where I used the result of Eq.(A.9).

$$\begin{aligned} \frac{1}{c} \frac{dn}{dt'} &= \frac{1}{c\vec{R}} \frac{\partial \vec{R}}{\partial t'} - \frac{1}{c\vec{R}^2} \vec{R} \frac{\partial \vec{R}}{\partial t'} \\ &= -\frac{1}{\vec{R}} [\vec{\beta} - \hat{n} \cdot (\vec{\beta}) \cdot \hat{n}] \end{aligned} \tag{A.21}$$

The quantity in bracket is written according to the triple vector product result

$$a \times (b \times c) = b(a \cdot c) - c(a \cdot b) \tag{A.22}$$

and so

$$\begin{aligned} \hat{n} \cdot (\vec{\beta} \cdot \hat{n} - \vec{\beta}) &= \hat{n} \cdot (\vec{\beta} \cdot \hat{n} - \hat{n} \cdot \hat{n} \cdot \vec{\beta}) = \hat{n} \times (\hat{n} \times \vec{\beta}) \\ \frac{1}{c} \frac{dn}{dt'} &= \frac{1}{\vec{R}} \hat{n} \times (\hat{n} \times \vec{\beta}) \end{aligned} \tag{A.23}$$

Likewise, get derivative quantitive in the bracket the Eq.(A.20)

$$\begin{aligned} \frac{d}{dt'} (\frac{1}{D\vec{R}}) &= -\frac{1}{(D\vec{R})^2} \frac{d}{dt'} \left[(1 - \hat{n} \cdot \vec{\beta}) \vec{R} \right] \\ &= -\frac{1}{(D\vec{R})}^2 \left[\vec{v}\vec{\beta} - (\hat{n} \cdot \dot{\vec{\beta}} \vec{R}) - c(1 - \hat{n} \cdot \vec{\beta}) \hat{n} \cdot \vec{\beta} \right], \quad with \quad \vec{v} = \frac{\vec{\beta}}{c} \\ &= -\frac{c}{(D\vec{R})^2} \left[\vec{\beta}^2 - \hat{n} \cdot \vec{\beta} - \frac{1}{c} \vec{R} (\hat{n} \cdot \dot{\vec{\beta}}) \right] \end{aligned} \tag{A.24}$$

Substitution of Eq.(A.21), (A.23) and (A.24) to Eq.(3.20) gives;

$$\begin{aligned}
\vec{E}(\vec{r},t) &= \frac{q}{4\pi\epsilon_0}\left[\frac{\hat{n}-\vec{\beta}}{(D\vec{R})^2} - \frac{\hat{n}-\vec{\beta}}{D(D\vec{R})^2}\left(\vec{\beta}^2 - \hat{n}\cdot\vec{\beta} - \frac{1}{c}\vec{R}(\hat{n}\dot{\vec{\beta}})\right) - \frac{\dot{\vec{\beta}}}{cD^2\vec{R}}\right] \\
&= \frac{q}{4\pi\epsilon_0}\left[\frac{\hat{n}-\vec{\beta}}{(D\vec{R})^2} - \frac{1}{D^3\vec{R}^2}[(\hat{n}-\vec{\beta})\beta^2 + (\hat{n}-\vec{\beta})\hat{n}\cdot\vec{\beta} + \frac{\vec{R}}{c}(\hat{n}-\vec{\beta})\hat{n}\cdot\dot{\vec{\beta}}]\right] \\
\vec{E}(\vec{r},t) &= \frac{q}{4\pi\epsilon_0}\left[\frac{(1-\vec{\beta})^2}{D^3\vec{R}^2}(\hat{n}-\vec{\beta}) + \frac{1}{CD^3\vec{R}}\hat{n}\times[(\hat{n}-\vec{\beta})\times\dot{\vec{\beta}}]\right] \qquad (A.25)
\end{aligned}$$

From the above relation I can conclude that the electric field has two terms,the first term is Coulomb field for relativistic effects, independent of the acceleration. It is proportional to $\frac{1}{\vec{R}^2}$ and The second term includes acceleration $\dot{\vec{\beta}}$ and it is proportional $\frac{1}{\vec{R}}$. When R increases the first term becomes less and less significant and therefore this first term does not contribute to radiation of energy. This is the radiation of electric field due to a moving charged particle. $\vec{\beta} = 0$ for nonrelativistic equation, because a particle is moving at non- relativistic speed whenever $\vec{\beta} << 1$. It is written in the simplified form of

$$\vec{E} = \frac{q}{4\pi\epsilon_0}\left[\frac{\hat{n}\times(\hat{n}\times\dot{\vec{\beta}})}{cD^3\vec{R}}\right] \qquad (A.26)$$

the quantity in brackets can written in following form;

$$| a\times b | = | a || b | sin\theta \qquad (A.27)$$

$$a\cdot b = | a || b | cos\theta \qquad (A.28)$$

translating cross product in Eq.(A.26) to Eq.(A.27) and (A.28);

$$\begin{aligned}
\hat{n}\times(\hat{n}\times\dot{\vec{\beta}}) &= \dot{\vec{\beta}}sin\theta \qquad (A.29) \\
\hat{n}\cdot\vec{\beta} &= \vec{\beta}cos\theta \qquad (A.30)
\end{aligned}$$

using Eq.(A.29) and (A.30), we rewrite Eq.(A.26)

$$\begin{aligned}
\vec{E}_{rad} &= \frac{q}{4\pi\epsilon_0 c}\frac{1}{\vec{R}}\frac{\dot{\vec{\beta}} sin\theta}{(1-\hat{n}\cdot\vec{\beta})^3} \\
\vec{E}_{rad} &= \frac{q}{4\pi\epsilon_0 c}\frac{1}{\vec{R}}\frac{\dot{\vec{\beta}} sin\theta}{(1-\vec{\beta} cos\theta)^3} \qquad \text{(A.31)}
\end{aligned}$$

Let's define the angular distribution of radiation power to find power radiation of non relativistic particle. The angular distribution of radiation power;

$$\begin{aligned}
\frac{d\vec{P}}{d\Omega} &= c\epsilon_0 \mid \vec{E}_{rad} \mid^2 \vec{R}^2 \\
&= c\epsilon_0 \left[\frac{q}{4\pi\epsilon_0 c}\frac{1}{\vec{R}}\frac{\dot{\vec{\beta}} sin\theta}{(1-\vec{\beta} cos\theta)^3}\right]^2 \vec{R}^2 \\
&= c\epsilon_0 \frac{q^2}{(4\pi)^2{\epsilon_0}^2 c^2}\frac{1}{\vec{R}}\frac{\dot{\vec{\beta}}^2 sin^2\theta}{(1-\vec{\beta} cos\theta)^6}\vec{R}^2 \\
\frac{d\vec{P}}{d\Omega} &= \frac{q^2}{(4\pi)^2\epsilon_0 c}\frac{\dot{\vec{\beta}} sin^2\theta}{(1-\vec{\beta} cos\theta)^6} \qquad \text{(A.32)}
\end{aligned}$$

In nonrelativistic limit $\vec{\beta} << 1$ therefore in Eq.(A.33) $\vec{\beta}$ goes to zero, if we translate to integrated form $d\Omega = sin\theta d\theta d\phi$

$$\begin{aligned}
d\vec{P} &= \frac{q^2}{(4\pi)^2\epsilon_0}\frac{\dot{\vec{\beta}}^2}{c}2\pi\int_0^{2\pi}\int_0^{\pi} sin^2\theta sin\theta d\theta d\phi \qquad \text{(A.33)} \\
&= \frac{q^2}{(4\pi)^2\epsilon_0}\frac{\dot{\vec{\beta}}^2}{c}2\pi\int_0^{\pi} sin\theta sin^2\theta d\theta \\
&= \frac{q^2}{(4\pi)^2\epsilon_0}\frac{\dot{\vec{\beta}}^2}{c}2\pi\int_0^{\pi} sin\theta(1-cos^2\theta)d\theta \\
d\vec{P} &= \frac{q^2}{(4\pi)^2\epsilon_0}\frac{\dot{\vec{\beta}}^2}{c}2\pi\left[\int_0^{\pi} sin\theta d\theta + \int_0^{\pi} cos^2\theta(-sin\theta)d\theta\right]
\end{aligned}$$

translating $x = cos\theta$, $dx = -sin\theta d\theta$

$$= \frac{q^2}{(4\pi)^2\epsilon_0}\frac{\dot{\vec{\beta}}^2}{c}2\pi\left[cos\theta|_0^{\pi} + \frac{x^3}{3}|_1^{-1}\right]$$

$$= \frac{q^2\dot{\vec{\beta}}^2}{6\pi\epsilon_0 c} \quad , \qquad with, \tag{A.34}$$

$$dot\vec{\beta} = \frac{\dot{\vec{v}}}{c}$$

$$d\vec{P} = \frac{q^2\dot{\vec{v}}^2}{6\pi\epsilon_0 c^3} \tag{A.35}$$

Putting the results of Eq.(A.35) into Eq.(A.34) gives

$$\dot{\vec{p}} = m_0\dot{\vec{v}} \ , \quad \frac{d\vec{p}}{dt} = m_0\dot{\vec{v}}, \tag{A.36}$$

$$\vec{P} = \frac{q^2}{6\pi\epsilon_0 c^3}\frac{1}{m_0^2}\left(\frac{d\vec{p}}{dt}\right)^2 --- > SI, , \quad unit \tag{A.37}$$

This is the Larmor radiation formula which is used to calculate the total power radiated by a nonrelativistic point charge. It was first derived by J.J.Larmor in 1987. To get an expression for relativistic particles I have to replace the time t by the Lorentz- invariant time $dt' = \frac{dt}{\gamma}$ and momentum $\vec{p}$ by the 4-momentum $\vec{P}_\mu$.

$$dt -- > dt' = \frac{1}{\gamma}dt \quad , \quad \gamma = \frac{E}{m_0c^2} = \frac{1}{\sqrt{1-\vec{\beta}^2}} \tag{A.38}$$

$$\left(\frac{d\vec{p}}{dt}\right)^2 --- > \left(\frac{d\vec{P}_\mu}{dt'}\right)^2 = \left(\frac{d\vec{p}}{dt'}\right)^2 - \frac{1}{c^2}\left(\frac{d\vec{E}}{dt'}\right)^2 \tag{A.39}$$

Get the radiated power in the relativistic invariant form

$$\vec{P}_s = \frac{q^2c}{6\pi\epsilon_0(m_0c^2)^2}\left[\left(\frac{d\vec{p}}{dt'}\right)^2 - \frac{1}{c^2}\left(\frac{d\vec{E}}{dt'}\right)^2\right] \tag{A.40}$$

where q is the electron charge, m its mass, c the velocity of light and $dt' = \frac{1}{\gamma}dt$ the element of proper time.

The radiated power depends on the angle between the direction of motion of the particle $\vec{v}$ and the direction of the acceleration $\frac{d\vec{v}}{dt'}$[3]. There are two kinds of acceleration which are linear acceleration and circular acceleration. Linear acceleration is motion of the particle $\vec{v}$ parallel to the acceleration $d\vec{v}/dt'$, circular acceleration is motion of the particle perpendicular to the acceleration.

A.5 Linear acceleration

We are dealing with a particle which may be moving at a speed close to that of light, we must use the relativistic expressions for the particle's energy and momentum, so energy of a particle is given by

$E = \gamma m_0 c^2$ and $p = \gamma m_0 v$.

The Particle energy is given as a function of particles rest mass energy and momentum;

$\vec{E}^2 = (m_0c^2)^2 + \vec{p}^2c^2$,

If we can apply the below formula according to above equtions;

$\vec{E}\frac{d\vec{E}}{dt'} = c^2\vec{p}\frac{d\vec{p}}{dt'}$

$$\begin{aligned} \gamma m_0 c^2 \frac{d\vec{E}}{dt'} &= c^2 \gamma m_0 \vec{v} \frac{d\vec{p}}{dt'} \\ \frac{d\vec{E}}{dt'} &= \vec{v}\frac{d\vec{p}}{dt'} \end{aligned} \tag{A.41}$$

now, we can derive the power radiated of linear acceleration in relativistic form using Eq.(A.4) and (A.39).

$$\begin{aligned} \vec{P}_s &= \frac{q^2 c}{6\pi\epsilon_0 (m_0c^2)^2}\left[\left(\frac{d\vec{p}}{dt'}\right) - \frac{1}{c^2}\left(\frac{d\vec{E}}{dt'}\right)\right] \\ &= \frac{q^2 c}{6\pi\epsilon_0 (m_0c^2)^2}\left[\left(\frac{d\vec{p}}{dt'}\right) - \frac{1}{c^2}\left(\frac{\vec{v}d\vec{p}}{dt'}\right)\right] \\ &= \frac{q^2 c}{6\pi\epsilon_0 (m_0c^2)^2}\left[1 - \left(\frac{\vec{v}}{c}\right)^2\right] \\ &= \frac{q^2 c}{6\pi\epsilon_0 (m_0c^2)^2}\left[\left(1 - \vec{\beta}^2\right)\left(\frac{d\vec{p}}{dt'}\right)^2\right] \end{aligned} \tag{A.42}$$

with $1-\vec{\beta}^2 = \frac{1}{\gamma^2}$ we can write

$$\vec{P}_s = \frac{q^2 c}{6\pi\epsilon_0(m_0c^2)^2}\left(\frac{d\vec{p}}{\gamma dt'}\right)^2 = \frac{q^2 c}{6\pi\epsilon_0(m_0c^2)^2}\left(\frac{d\vec{p}}{dt'}\right)^2 \tag{A.43}$$

The rate of change of momentum is equal to the rate of change of the energy of the particle per unit distance. Thus

$$\frac{d\vec{p}}{dt'} = \frac{cd\vec{p}}{cdt'} = \frac{d\vec{E}}{dx} \tag{A.44}$$

$$\vec{P}_s = \frac{q^2 c}{6\pi\epsilon_0(m_0c^2)^2}\left(\frac{d\vec{E}}{dx}\right)^2 \tag{A.45}$$

This equation shows that for linear motion, the power radiated indicates the rate of change of particle energy with distance.

A.6 Circular acceleration

$$\vec{P}_s = \frac{q^2 c}{6\pi\epsilon_0(m_0c^2)^2}\left(\frac{d\vec{p}}{dt'}\right)^2 = \frac{q^2 c\gamma^2}{6\pi\epsilon_0(m_0c^2)^2}\left(\frac{d\vec{p}}{dt}\right)^2 \tag{A.46}$$

Above equation which is used equation (A.36). On circular trajectory with radius ρ a change of the orbit angle $d\alpha$ .This causes a momentum variation

$$d\vec{p} = pd\alpha \tag{A.47}$$

$$\frac{d\vec{p}}{dt} = \vec{p}\vec{\omega} = \vec{p}\frac{\vec{v}}{R} = \frac{\vec{E}}{\rho} \tag{A.48}$$

insert this equation (A.44) and get with $\gamma = \frac{E}{m_0c^2}$

$$\vec{P}_s = \frac{q^2 c\left(\frac{\vec{E}}{m_0c^2}\right)^2}{6\pi\epsilon_0(m_0c^2)^2}\left(\frac{\vec{E}}{\rho}\right)^2 \tag{A.49}$$

$$= \frac{q^2 c}{6\pi\epsilon_0 (m_0 c^2)^4} \left(\frac{\vec{E}^4}{\rho^2} \right) \tag{A.50}$$

Eq.(A.49) shows that the power associated with this radiation P is a rapidly increasing function of the charging particle energy E. If I know the particle ebergy, It helps me to calculate the energy loss per turn of a particle in a circular accelerator by integrating the radiation power along the circular.

In circular accelerator the energy loss per turn is

$$\triangle\ E = \oint \vec{P}_s dt = \vec{P}_s t_b = \vec{P}_s 2\pi \frac{\rho}{c} \tag{A.51}$$

Here t_b is the time per revolution. Insert (A.48) into (A.49) and The energy radiated per orbit by each charged particle is

$$\triangle\ E = \frac{q^2}{3\epsilon_0 (m_0.c^2)^4} \frac{E^4}{\rho} \tag{A.52}$$

In a circular accelerator, this equation indicates that energy loss per turn scales like the fourth power of the particle energy and it is inversely proportional to the radius of the accelerator.

APPENDIX B

MATLAB M FILE FOR THE TIME RESOLVED

B.1 one_acq

```
clear all;
addpath('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir/mtv-scan-
prog/functions');
addpath('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir/mtv-scan-
prog/functions/gauss-fit');
%[MTVx,MTVxproj,MTVy,MTVyproj,MTVts,MTVp,MTVmaxpix] = Jgetim-
age('CR.MTV0451',0,2);
        %MTV.x = MTVx;
        %MTV.xproj = MTVxproj;
        %MTV.y = MTVy;
        %MTV.yproj = MTVyproj;
        %MTV.ts = MTVts;  % time stamp - not used at the moment
        %MTV.p = MTVp;
        %MTV.maxpix = MTVmaxpix;
   mtv_mtv=load('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir/
   mtv_scan_prog/data/combined_beam_2/scan_2/mtv_delay5_aver1.mat');
        mtv=mtv_mtv.mtv_save;
        mtv_x=mtv.x;
        mtv_y=mtv.y;
        mtv_p=mtv.p;
        mtv_xproj=mtv.xproj;
        mtv_yproj=mtv.yproj;
```

```
    contourf(mtv_x,mtv_y,mtv_p');
%save data
%data_x=['/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir/
mtv_scan_prog/data/MTV_data' '.mat'];
%save(data_x, 'MTV');
%data fit
    [xfit,x_q,x_dq,x_chisq_ndf]= agauss_fit(mtv_x,mtv_xproj,'','1');
    [yfit,y_q,y_dq,y_chisq_ndf]= agauss_fit(mtv_y,mtv_yproj,'','1');
sigma_x=x_q(4);
sigma_y=y_q(4);
    figure;
    plot(mtv_x,mtv_xproj,'b',mtv_x,xfit,'g');
    xlabel('Position [mm]');
    ylabel('Intensity [mm]');
    legend('Data','Fit');
    title('Horizontal projection');
    grid on;
    figure;
    plot(mtv_y,mtv_yproj,'b',mtv_y,yfit,'g');
    xlabel('Position [mm]');
    ylabel('Intensity [mm]');
    legend('Data','Fit');
    title('Vertical projection');
    grid on;
display(sigma_x);
display(sigma_y);
    %plot energy spread
    %Itot = 0;    % Total intensity
    %avgx = 0;    % averaged x
```

%avgy = 0; % averaged y

%Itot = sum(sum(MTVp));

%for i=1:size(MTVx,1)

%for j=1:size(MTVy,1)

%avgx = avgx + MTVx(i,1)*MTVp(i,j);

%avgy = avgy + MTVy(j,1)*MTVp(i,j);

%end

%end

%avgx = avgx/Itot;

%avgy = avgy/Itot;

%MTVx = MTVx - avgx;

%MTVy = MTVy - avgy;

%[dummy,hc]=contourf(MTVx,MTVy,MTVp',20); % The data is now centered

%MAXy = find(round(MTVy) == round(avgy)); % index where the mean intensity is

1cm%MAXy = MAXy(1,1); % very approximate

%MeanScreen = sum(MTVx.*MTVp(:,MAXy))/sum(MTVp(:,MAXy));

%SigmaScreen=sqrt(sum((MTVx-MeanScreen).*(MTVx-MeanScreen).*
(MTVp(:,MAXy)))/sum(MTVp(:,MAXy)));

%figure;

%plot(MTVx,MTVp(:,MAXy));

%hold on;

%plot(MTVx,max(MTVp(:,MAXy))*exp(-(MTVx-MeanScreen).*
(MTVx-MeanScreen)/(2*SigmaScreen*SigmaScreen)),'-k','Linewidth',3);

B.2 scan_mtv

clear all;

addpath('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir

```
/mtv_scan_prog/functions');
addpath('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir
/mtv_scan_prog/functions/gauss_fit');
    %MTV=load('/acc/oper/share/ctf/data/ctfmod/MatLab/Maja/ScreenScans
/data/2010/20100929/scan2/CCS.MTV0980_10_09_29_12_57_20_Cur_96.3265.mat');
    % MTVx=MTV.MTV.x;
    % MTVy=MTV.MTV.y;
    % MTVxproj=MTV.MTV.xproj;
    % MTVyproj=MTV.MTV.yproj;
    % MTVp=MTV.MTV.p;
gate_dur=400;
pulse_lenght=1200;
number_acquis=pulse_lenght/gate_dur;
time=0:gate_dur:pulse_lenght-gate_dur;
    %plot data
    for i=1:number_acquis
%reading of the data of the gated camera in the combiner ring
    [MTVx,MTVxproj,MTVy,MTVyproj,MTVts,MTVp,MTVmaxpix] =
Jgetimage('CR.MTV0451',0,2);
    MTV.x = MTVx;
    MTV.xproj = MTVxproj;
    MTV.y = MTVy;
    MTV.yproj = MTVyproj;
    MTV.ts = MTVts;    % time stamp - not used at the moment
    MTV.p = MTVp;
    MTV.maxpix = MTVmaxpix;
    contourf(MTVx,MTVy,MTVp');
%save data
data_x=['/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir
```

```
/mtv_scan_prog/data/MTV_data' num2str(i) '.mat'];
save(data_x, 'MTV');
[xfit,x_q,x_dq,x_chisq_ndf]= agauss_fit(MTVx,MTVxproj,'','1');
[yfit,y_q,y_dq,y_chisq_ndf]= agauss_fit(MTVy,MTVyproj,'','1');
    sigma_x(i)=x_q(4);
    sigma_y(i)=y_q(4);
    figure;
    plot(MTVx,MTVxproj,'b',MTVx,xfit,'g');
    figure;
    plot(MTVy,MTVyproj,'b',MTVy,yfit,'g');
    display(x_q(4));
    display(y_q(4));
% %plot energy spread
% Itot = 0;    % Total intensity
% avgx = 0;    % averaged x
% avgy = 0;    % averaged y
% Itot = sum(sum(MTVp));
% for i=1:size(MTVx,1)
% for j=1:size(MTVy,1)
% avgx = avgx + MTVx(i,1)*MTVp(i,j);
% avgy = avgy + MTVy(j,1)*MTVp(i,j);
% end
% end
% avgx = avgx/Itot;
% avgy = avgy/Itot;
% %MTVx = MTVx - avgx;
% %MTVy = MTVy - avgy;
% %[dummy,hc]=contourf(MTVx,MTVy,MTVp',20);    % The data is now centered
```

% MAXy = find(round(MTVy) == round(avgy)); % index where the mean intensity is

% MAXy = MAXy(1,1); % very approximate

% MeanScreen = sum(MTVx.*MTVp(:,MAXy))/sum(MTVp(:,MAXy));

% SigmaScreen = sqrt(sum((MTVx-MeanScreen).*(MTVx-MeanScreen).*(MTVp(:,MAXy)))/sum(MTVp(:,MAXy)));

%figure;

%plot(MTVx,MTVp(:,MAXy));

%hold on;

%plot(MTVx,max(MTVp(:,MAXy))*exp(-(MTVx-MeanScreen).*(MTVx-MeanScreen)/(2*SigmaScreen*SigmaScreen)),'-k','Linewidth',3);

%control the delay

%timing_delay(gate_length);

end

figure;

plot(time,sigma_x,'LineStyle','none','Marker','+','MarkerFaceColor','b','MarkerEdgeColor','b');

hold on;

plot(time,sigma_y,'LineStyle','none','Marker','+','MarkerFaceColor','r','MarkerEdgeColor','r');

legend('sigma x', 'sigma y');

title('Vertical and Horizontal beam size (sigma) versus time');

grid on;

% %reading of the streak camera timing (reference timing at the ns scale)

% streak_timing=JGetCoValueFesa('CX.SSTREAK2-CTTIM','Setting','delay');

%streak_timing2=JGetCoValueFesa('CX.SSTREAK2-F','Setting','delay');

% display(streak_timing);

% %reading of the fast delay (to scan the pulse)

% fast_delay_coarse=Jgetcontrolval('CL.ECLCNT1','CCV1');

%coarse timing

% fast_delay_fine=Jgetcontrolval('CL.ECLCNT1','CCV2');

%fine timing

% display(fast_delay_coarse);

% display(fast_delay_fine);

% %setting of the streak camera timing (reference timing at the ns scale)

% JSetCoValueFesa('CX.SSTREAK2-CTTIM','Setting','delay',streak$_{t}iming$);

% JSetCoValueFesa('CX.SSTREAK2-F','CCV','value',streak_timing2);

% %setting of the fast delay (to scan the pulse)

% JSetCoValueFesa('CL.ECLCNT1','CCV1','value',fast_delay_coarse); %coarse timing

% JSetCoValueFesa('CL.ECLCNT1','CCV2','value',fast_delay_fine); %fine timing

B.3 scan mtc turn graph

****

function scan_mtv_turn_graph_gui

****

%Note: uncomment the lines 1184 and 1226

%initial timing to see the 4th turn: 30107 (then decrease to go up to

%the first turn) or 30094 for multiturn of 6 turns; 30087

%CREATING THE DATA

mOutputArgs=;

addpath('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir/
mtv_scan_prog/functions/gauss_fit');

addpath('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir/
mtv_scan_prog/functions');

%CREATING GUI AND ITS COMPONENTS

```
%The main figure for GUI
{hMainFigure = figure(... % The main GUI figure}
            'MenuBar','none', ...
            'Toolbar','none', ...
            'HandleVisibility','callback', ...
            'Name','Measurements',...
            'Color', get(0,...
            'defaultuicontrolbackgroundcolor'));
%The file menu and its menu items
hFileMenu = uimenu(... % File menu
          'Parent',hMainFigure,...
          'HandleVisibility','callback', ...
          'Label','File');
hOpenMenuitem = uimenu(... % Open menu item
              'Parent',hFileMenu,...
              'Label','Open',...
              'HandleVisibility','callback', ...
              'Callback', @hOpenMenuitemCallback);
hSaveMenuitem = uimenu(... % Print menu item
              'Parent',hFileMenu,...
              'Label','Save',...
              'HandleVisibility','callback', ...
              'Callback', @hSaveMenuitemCallback);
hPrintMenuitem = uimenu(... % Print menu item
              'Parent',hFileMenu,...
              'Label','Print',...
              'HandleVisibility','callback', ...
              'Callback', @hPrintMenuitemCallback);
hCloseMenuitem = uimenu(... % Open menu item
```

```
                    'Parent',hFileMenu,...
                    'Label','Close',...
                    'HandleVisibility','callback', ...
                    'Callback', @hCloseMenuitemCallback);
%The Toolbar and Its Tools
hToolbar = uitoolbar(... % Toolbar for Open and Print buttons
             'Parent',hMainFigure, ...
             'HandleVisibility','callback');
% hOpenPushtool = uipushtool(... % Open toolbar button
                    % 'Parent',hToolbar,...
                    % 'TooltipString','Open File',...
                    % 'CData',iconRead(fullfile(matlabroot,...
                    % 'toolbox.mat')),...
                    % 'HandleVisibility','callback', ...
                    % 'ClickedCallback', @hOpenMenuitemCallback);
% hSavePushtool = uipushtool(... % Open toolbar button
                    % 'Parent',hToolbar,...
                    % 'TooltipString','Save File',...
                    % 'CData',iconRead(fullfile(matlabroot,..
                    % 'toolbox.mat')),...
                    % 'HandleVisibility','callback', ...
                    % 'ClickedCallback', @hSaveMenuitemCallback);
% hPrintPushtool = uipushtool(... % Print toolbar button
                    % 'Parent',hToolbar,...
                    % 'TooltipString','Print Figure',...
                    % 'CData',iconRead(fullfile(matlabroot,...
                    % 'toolbox.mat')),...
                    % 'HandleVisibility','callback', ...
                    % 'ClickedCallback', @hPrintMenuitemCallback);
```

```
%The axes for the image of the camera
hImageAxes = axes(... % Axes for plotting the selected plot
              'Parent', hMainFigure, ...
              'Units', 'normalized', ...
              'HandleVisibility','callback', ...
              'Position',[0.05 0.7 0.3 0.2]);
xlabel(hImageAxes,'Position [mm]');
ylabel(hImageAxes,'Position [mm]');
zlabel(hImageAxes,'Intensity [mm]');
title(hImageAxes,'Image taken by the camera');
%The axes for the horizontal projection
hHorizProjAxes = axes(... % Axes for plotting the selected plot
                 'Parent', hMainFigure, ...
                 'Units', 'normalized', ...
                 'HandleVisibility','callback', ...
                 'Position',[0.4 0.7 0.3 0.2]);
xlabel(hHorizProjAxes,'Position [mm]');
ylabel(hHorizProjAxes,'Intensity [mm]');
legend(hHorizProjAxes,'Data','Fit');
title(hHorizProjAxes,'Horizontal projection');
grid(hHorizProjAxes,'on');
%The axes for the vertical projection
hVertProjAxes = axes(... % Axes for plotting the selected plot
                 'Parent', hMainFigure, ...
                 'Units', 'normalized', ...
                 'HandleVisibility','callback', ...
                 'Position',[0.05 0.43 0.3 0.2]);
xlabel(hVertProjAxes,'Position [mm]');
ylabel(hVertProjAxes,'Intensity [mm]');
```

```
legend(hVertProjAxes,'Data','Fit');
title(hVertProjAxes,'Vertical projection');
grid(hVertProjAxes,'on');
%The axes for the plot of sigma versus time in the vertical direction
hSigmaTimeVerticalAxes = axes(... % Axes for plotting the selected plot
                              'Parent', hMainFigure, ...
                              'Units', 'normalized', ...
                              'HandleVisibility','callback', ...
'Position',[0.4 0.43 0.3 0.2]);
xlabel(hSigmaTimeVerticalAxes,'Time [ns]');
ylabel(hSigmaTimeVerticalAxes,'Sigma [mm]');
title(hSigmaTimeVerticalAxes,'Vertical beam size (sigma) versus time');
grid(hSigmaTimeVerticalAxes,'on');
%The axes for the plot of sigma versus time in the horizontal direction
hSigmaTimeHorizontalAxes = axes(... % Axes for plotting the selected plot
                                'Parent', hMainFigure, ...
                                'Units', 'normalized', ...
                                'HandleVisibility','callback', ...
                                'Position',[0.05 0.16 0.3 0.2]);
xlabel(hSigmaTimeHorizontalAxes,'Time [ns]');
title(hSigmaTimeHorizontalAxes,'Horizontal beam size (sigma) versus time');
grid(hSigmaTimeHorizontalAxes,'on');
%The background subtraction button
hBackgroundButton = uicontrol(... % Button for updating selected plot
                        'Parent', hMainFigure, ...
                        'Units','normalized',...
                        'HandleVisibility','callback', ...
                        'Position',[0.01 0.05 0.1 0.05],...
                        'FontSize',16,...
```

```
                    'String','Backg.sub',...
                    'Style','togglebutton',...
                    'Callback', @hBackgroundButtonCallback);
%The acquisition button (only one image)
hAcquisitionButton = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.12 0.05 0.07 0.05],...
                    'FontSize',16,...
                    'String','One shot',...
                    'Style','pushbutton',...
                    'Callback', @hAcquisitionButtonCallback);
%The scan button
hScanButton = uicontrol(... % Button for updating selected plot
               'Parent', hMainFigure, ...
               'Units','normalized',...
               'HandleVisibility','callback', ...
               'Position',[0.2 0.05 0.05 0.05],...
               'FontSize',16,...
               'String','Scan',...
               'Style','pushbutton',...
               'Callback', @hScanButtonCallback);
%The stop button
hStopButton = uicontrol(... % Button for updating selected plot
               'Parent', hMainFigure, ...
               'Units','normalized',...
               'HandleVisibility','callback', ...
               'Position',[0.25 0.05 0.05 0.05],...
```

```
                    'FontSize',16,...
                    'String','Stop',...
                    'Style','togglebutton',...
                    'Callback', @hStopButtonCallback);
%The pop-up menu for the selection of the turn of the combiner ring
hSelectTurnPopupmenu = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.31 0.05 0.1 0.05],...
                    'FontSize',16,...
                    'String','1st turn' '2nd turn' '3rd turn' '4th turn',...
                    'Style','popupmenu',...
                    'BackgroundColor','white',...
                    'Enable','on',...
                    'Callback', @hSelectTurnPopupmenuCallback);
%The text field for the delay when changing turn in the combiner ring
hDelayTurnText1 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.31 0.03 0.05 0.03],...
                    'FontSize',16,...
                    'String','Delay:',...
                    'BackgroundColor','white',..
                    'Style','text');
hDelayTurnText2 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
```

```
                    'HandleVisibility','callback', ...
                    'Position',[0.36 0.03 0.05 0.03],...
                    'FontSize',16,...
                    'String','0ns',...
                    'BackgroundColor','white',...
                    'ForegroundColor','blue',...
                    'Style','text');
%The text field for the length of the camera gate


hGateLengthText = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.42 0.05 0.1 0.05],...
                    'FontSize',16,...
                    'String','Gate length [ns]:',...
                    'BackgroundColor','white',...
                    'Style','text');
                                %'BackgroundColor','white',...
hGateLengthEdit = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.52 0.05 0.05 0.05],...
                    'FontSize',16,...
                    'String','400',...
                    'Style','edit',...
                    'Callback', @hGateLengthEditCallback);
%The text field for the number of delay change
```

```
hDelayNumberText = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.57 0.05 0.1 0.05],...
                    'FontSize',16,...
                    'String','Delay number:',...
                    'BackgroundColor','white',...
                    'Style','text');
%'BackgroundColor','white',...
hDelayNumberEdit = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.67 0.05 0.05 0.05],...
                    'FontSize',16,...
                    'String','3',...
                    'Style','edit',...
                    'Callback', @hDelayNumberEditCallback);
%The text field for the number of averages
hAverageNumberText = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.72 0.05 0.1 0.05],...
                    'FontSize',16,...
                    'String','Average number:',...
                    'BackgroundColor','white',...
```

```
                        'Style','text');
%'BackgroundColor','white',...
hAverageNumberEdit = uicontrol(... % Button for updating selected plot
                        'Parent', hMainFigure, ...
                        'Units','normalized',...
                        'HandleVisibility','callback', ...
                        'Position',[0.82 0.05 0.05 0.05],...
                        'FontSize',16,...
                        'String','3',...
                        'Style','edit',...
                        'Callback', @hAverageNumberEditCallback);
%The save button
hSaveButton = uicontrol(... % Button for updating selected plot
                        'Parent', hMainFigure, ...
                        'Units','normalized',...
                        'HandleVisibility','callback', ...
                        'Position',[0.87 0.05 0.1 0.05],...
                        'FontSize',16,...
                        'String','Save data',...
                        'Style','pushbutton',...
                        'Enable','off',...
                        'Callback', @hSaveButtonCallback);
%The text field for the number of acquistions (in the scan mode)
hDelayCounterText1 = uicontrol(...
                        'Parent', hMainFigure, ...
                        'Units','normalized',...
                        'HandleVisibility','callback', ...
                        'Position',[0.58 0.93 0.07 0.05],...
                        'FontSize',16,...
```

```
                    'String','Delay counter:',...
                    'BackgroundColor','white',...
                    'Style','text');
hDelayCounterText2 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.66 0.95 0.05 0.03],...
                    'FontSize',16,...
                    'String','0',...
                    'BackgroundColor','white',...
                    'ForegroundColor','blue',...
                    'Style','text');
%The text field for the delay when doing scan
hDelayValueText1 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.66 0.91 0.06 0.03],...
                    'FontSize',16,...
                    'String','Delay:',...
                    'BackgroundColor','white',...
                    'Style','text');
hDelayValueText2 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.72 0.91 0.05 0.03],...
                    'FontSize',16,...
```

```
                    'String','0 ns',...
                    'BackgroundColor','white',...
                    'ForegroundColor','blue',...
                    'Style','text');
%The text field for the number of averages (in the scan mode)
hAverageCounterText1 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.78 0.95 0.13 0.03],...
                    'FontSize',16,...
                    'String','Average counter:',...
                    'BackgroundColor','white',...
                    'Style','text');
hAverageCounterText2 = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.92 0.95 0.05 0.03],...
                    'FontSize',16,...
                    'String','0',...
                    'BackgroundColor','white',...
                    'ForegroundColor','blue',...
                    'Style','text');
%The edit button for the name of the camera
hCameraEdit = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
```

```
                    'Position',[0.05 0.95 0.1 0.03],...
                    'FontSize',16,...
                    'String','CR.MTV0451',...
                    'Style','edit');
%The pop-up menu for the selection of the graphs
hSelectGraphText = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.17 0.93 0.05 0.05],...
                    'FontSize',16,...
                    'String','Select delay:',...
                    'BackgroundColor','white',...
                    'Style','text');
hSelectGraphPopupmenu = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.23 0.93 0.05 0.05],...
                    'FontSize',16,...
                    'String','Select scan',...
                    'Style','popupmenu',...
                    'BackgroundColor','white',...
                    'Enable','off',...
                    'Callback', @hSelectGraphPopupmenuCallback);
%The pop-up menu for the selection of the turn for graph display
hTurnDisplayText = uicontrol(...
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
```

'HandleVisibility','callback', ...

'Position',[0.45 0.93 0.05 0.05],...

'FontSize',16,...

'String','Select turn:',...

'BackgroundColor','white',...

'Style','text');

hTurnDisplayPopupmenu = uicontrol(... % Button for updating selected plot

'Parent', hMainFigure, ...

'Units','normalized',...

'HandleVisibility','callback', ...

'Position',[0.51 0.93 0.05 0.05],...

'FontSize',16,...

'String','Select turn',...

'Style','popupmenu',...

'BackgroundColor','white',...

'Enable','on',...

'Callback', @hTurnDisplayPopupmenuCallback);

%The pop-up menu for the selection of the averages

hSelectAverageGraphText = uicontrol(...

'Parent', hMainFigure, ...

'Units','normalized',...

'HandleVisibility','callback', ...

'Position',[0.3 0.93 0.07 0.05],...

'FontSize',16,...

'String','Select average:',...

'BackgroundColor','white',...

'Style','text');

hSelectAverageGraphPopupmenu = uicontrol(... % Button for updating selected plot

'Parent', hMainFigure, ...

'Units','normalized',...

'HandleVisibility','callback', ...

'Position',[0.38 0.93 0.05 0.05],...

'FontSize',16,...

'String','Select average',...

'Style','popupmenu',...

'BackgroundColor','white',...

'Enable','off',...

'Callback', @hSelectAverageGraphPopupmenuCallback);

%The save button for the figure vertical sigma versus time

hSaveVertSigmaTimeButton = uicontrol(... % Button for updating selected plot

'Parent', hMainFigure, ...

'Units','normalized',...

'HandleVisibility','callback', ...

'Position',[0.66 0.37 0.05 0.03],...

'FontSize',16,...

'String','Save',...

'Style','pushbutton',...

'Callback', @hSaveVertSigmaTimeButtonCallback);

%The save button for the figure horizontal sigma versus time

hSaveHorizSigmaTimeButton = uicontrol(... % Button for updating selected plot

'Parent', hMainFigure, ...

'Units','normalized',...

'HandleVisibility','callback', ...

'Position',[0.3 0.11 0.05 0.03],...

'FontSize',16,...

```
                    'String','Save',...
                    'Style','pushbutton',...
                    'Callback', @hSaveHorizSigmaTimeButtonCallback);
%The save button for the figure vertical projection
hSaveVertProjButton = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.3 0.37 0.05 0.03],...
                    'FontSize',16,...
                    'String','Save',...
                    'Style','pushbutton',...
                    'Callback', @hSaveVertProjButtonCallback);
%The save button for the figure vertical projection
hSaveHorizProjButton = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.66 0.65 0.05 0.03],...
                    'FontSize',16,...
                    'String','Save',...
                    'Style','pushbutton',...
                    'Callback', @hSaveHorizProjButtonCallback);
%The save button for the figure horizontal sigma versus time
hImageButton = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.3 0.65 0.05 0.03],...
```

'FontSize',16,...

'String','Save',...

'Style','pushbutton',...

'Callback', @hImageButtonCallback);

%The table of sigma (horizontal and vertical) versus time

sigma$_{t}ime$ = [];

% Define parameters for a uitable (col headers are fictional)

colnames = 'Time [ns]', 'Sigma x [mm]', 'Sigma y [mm]';

% All column contain numeric data (integers, actually)

colfmt = 'numeric', 'numeric';

% Disallow editing values (but this can be changed)

coledit = [false false];

% Set columns all the same width (must be in pixels)

colwdt = 120 120;

% Create a uitable on the left side of the figure

hSigmaTimeTable = uitable(...,

'Parent', hMainFigure, ...

'Units', 'normalized',...

'FontSize',12,...

'Position', [0.72 0.15 0.25 0.75],...

'Data', sigma$_{t}ime, ...$

'ColumnName', colnames,...

'ColumnFormat', colfmt,...

'ColumnWidth', colwdt,...

'ColumnEditable', coledit,...

'ToolTipString',...

'Select cells to highlight them on the plot',...

'CellSelectionCallback',@select$_{c}allback$);

%The reset button

```
hResetButton = uicontrol(... % Button for updating selected plot
                    'Parent', hMainFigure, ...
                    'Units','normalized',...
                    'HandleVisibility','callback', ...
                    'Position',[0.45 0.01 0.05 0.04],...
                    'FontSize',16,...
                    'String','Reset',...
                    'Style','pushbutton',...
                    'Callback', @hResetButtonCallback);
%INITIALIZE THE GUI

gate_length=str2double(get(hGateLengthEdit, 'String'));
delay_number=str2double(get(hDelayNumberEdit, 'String'));
average_number=str2double(get(hAverageNumberEdit, 'String'));
mtv_name=get(hCameraEdit, 'String');
graph_string=get(hSelectGraphPopupmenu,'String');
graph_selection=get(hSelectGraphPopupmenu,'Value');
graph_chosen=graph_stringgraph_selection;
graph_average_string=get(hSelectAverageGraphPopupmenu,'String');
graph_average_selection=get(hSelectAverageGraphPopupmenu,'Value');
graph_average_chosen=graph_average_stringgraph_average_selection,1;
turn_string=get(hSelectTurnPopupmenu,'String');
turn_selection=get(hSelectTurnPopupmenu,'Value');
turn_chosen=turn_stringturn_selection;
turn_display_string=get(hTurnDisplayPopupmenu,'String');
turn_display_selection=get(hTurnDisplayPopupmenu,'Value');
turn_display_chosen=turn_display_stringturn_display_selection;
turn_selection_history=[];
turn_chosen_history=;
```

```
delay_number_history=[];
average_number_history=[];
current_graph_value=0;
current_graph_average_value=0;
time=[];
sigma_x=[];
sigma_y=[];
time_scatter=[];
sigma_x_scatter=[];
sigma_y_scatter=[];
if get(hBackgroundButton,'Value')==get(hBackgroundButton,'Max')
        background_sub='yes';
elseif get(hBackgroundButton,'Value')==get(hBackgroundButton,'Min')
        background_sub='no';
end
if get(hStopButton,'Value')==get(hStopButton,'Max')
        stop_scan='yes';
elseif get(hStopButton,'Value')==get(hStopButton,'Min')
        stop_scan='no';
end
MTV.x=[];
MTV.xproj=[];
MTV.xproj_average=[];
MTV.y=[];
MTV.yproj=[];
MTV.yproj_average=[];
MTV.ts='no date';
MTV.p=[];
MTV.p_average=[];
```

```
MTV.maxpix=0;
mtv.x=[];
mtv.xproj=[];
mtv.y=[];
mtv.yproj=[];
mtv.ts='no date';
mtv.p=[];
mtv.maxpix=0;
w_save=0;
scan_number=0;
%DEFINING THE CALLBACKS
%The toggle button for the background subtraction
function hBackgroundButtonCallback(hObject, eventdata, handles)
if get(hBackgroundButton,'Value')==get(hBackgroundButton,'Max')
            background_sub='yes';
            set(hBackgroundButton,'BackgroundColor','green');
elseif get(hBackgroundButton,'Value')==get(hBackgroundButton,'Min')
            background_sub='no';
set(hBackgroundButton,'BackgroundColor','default');     end
end
%The stop button for the scan
function hStopButtonCallback(hObject, eventdata, handles)
if get(hStopButton,'Value')==get(hStopButton,'Max')
            stop_scan='yes';
set(hStopButton,'BackgroundColor','red');
elseif get(hStopButton,'Value')==get(hStopButton,'Min')
            stop_scan='no';
set(hStopButton,'BackgroundColor','default');
    end
```

```
end
%The reset button
function hResetButtonCallback(hObject, eventdata, handles)
    MTV=[];
    scan_number=0;
    turn_selection_history=[];
    time=[];
    sigma_x=[];
    sigma_y=[];
    sigma_time=[];
    turn_chosen_history=;
    cla(hImageAxes);
    cla(hHorizProjAxes);
    cla(hVertProjAxes);
    cla(hSigmaTimeVerticalAxes);
    cla(hSigmaTimeHorizontalAxes);
    set(hResetButton,'BackgroundColor','red');
end
%The save button for the graph vertical beam size versus time
function hSaveVertSigmaTimeButtonCallback(hObject, eventdata, handles)
    % create a new figure for saving and printing
    fig = figure('visible','on');
    % copy axes into the new figure
    newax = copyobj(hSigmaTimeVerticalAxes,fig);
% If you would like to have the axes cover a larger area of the figure window
rather
% than the original size as in the subplot, then change it's Position value:
set(newax, 'units', 'normalized', 'position', [0.13 0.11 0.775 0.815]);
    % save the figure
```

```
        file_name=inputdlg('Enter file name for the selected data of the table');
        folder_name=uigetdir;
        directory_file_name=char(strcat(folder_name,'/',file_name,'.fig'));
        hgsave(fig,directory_file_name); % save it
        close(fig); % clean up by closing it
    end
    %The save button for the graph vertical beam size versus time
    function hSaveHorizSigmaTimeButtonCallback(hObject, eventdata, handles)
        % create a new figure for saving and printing
        fig = figure('visible','on');
        % copy axes into the new figure
        newax = copyobj(hSigmaTimeHorizontalAxes,fig);
    % If you would like to have the axes cover a larger area of the figure window
rather
    % than the original size as in the subplot, then change it's Position value:
        set(newax, 'units', 'normalized', 'position', [0.13 0.11 0.775 0.815]);
        % save the figure
        file_name=inputdlg('Enter file name for the selected data of the table');
        folder_name=uigetdir;
        directory_file_name=char(strcat(folder_name,'/',file_name,'.fig'));
        hgsave(fig,directory_file_name); % save it
        close(fig); % clean up by closing it
    end
    function hSaveVertProjButtonCallback(hObject, eventdata, handles)
        % create a new figure for saving and printing
        fig = figure('visible','on');
        % copy axes into the new figure
        newax = copyobj(hVertProjAxes,fig);
    % If you would like to have the axes cover a larger area of the figure window
```

```
rather
    % than the original size as in the subplot, then change it's Position value:
        set(newax, 'units', 'normalized', 'position', [0.13 0.11 0.775 0.815]);
        % save the figure
        file_name=inputdlg('Enter file name for the selected data of the table');
        folder_name=uigetdir;
        directory_file_name=char(strcat(folder_name,'/',file_name,'.fig'));
        hgsave(fig,directory_file_name); % save it
        close(fig); % clean up by closing it
    end
    function hSaveHorizProjButtonCallback(hObject, eventdata, handles)
        % create a new figure for saving and printing
        fig = figure('visible','on');
        % copy axes into the new figure
        newax = copyobj(hHorizProjAxes,fig);
    % If you would like to have the axes cover a larger area of the figure window
rather
    % than the original size as in the subplot, then change it's Position value:
        set(newax, 'units', 'normalized', 'position', [0.13 0.11 0.775 0.815]);
        % save the figure
        file_name=inputdlg('Enter file name for the selected data of the table');
        folder_name=uigetdir;
        directory_file_name=char(strcat(folder_name,'/',file_name,'.fig'));
        hgsave(fig,directory_file_name); % save it
        close(fig); % clean up by closing it
    end
    function hImageButtonCallback(hObject, eventdata, handles)
        % create a new figure for saving and printing
        fig = figure('visible','on');
```

```
        % copy axes into the new figure
        newax = copyobj(hImageAxes,fig);
    % If you would like to have the axes cover a larger area of the figure window
rather
    % than the original size as in the subplot, then change it's Position value:
        set(newax, 'units', 'normalized', 'position', [0.13 0.11 0.775 0.815]);
        % save the figure
        file_name=inputdlg('Enter file name for the selected data of the table');
        folder_name=uigetdir;
        directory_file_name=char(strcat(folder_name,'/',file_name,'.fig'));
        hgsave(fig,directory_file_name); % save it
        close(fig); % clean up by closing it
    end
    %The edit button for the length of the camera gate
    function hGateLengthEditCallback(hObject, eventdata, handles)
        gate_length = str2double(get(hObject,'string'));
        if isnan(gate_length)
        errordlg('You must enter a numeric value','Bad Input','modal')
        uicontrol(hObject)
        return
      end
    end
    %The edit button for the number of delay change
    function hDelayNumberEditCallback(hObject, eventdata, handles)
    delay_number = str2double(get(hObject,'string'));
    if isnan(delay_number)
        errordlg('You must enter a numeric value','Bad Input','modal')
        uicontrol(hObject)
          return
```

```
        end
    end
    %The edit button for the length of the beam pulse
    function hAverageNumberEditCallback(hObject, eventdata, handles)
    average_number = str2double(get(hObject,'string'));
    if isnan(average_number)
        errordlg('You must enter a numeric value','Bad Input','modal')
        uicontrol(hObject)
        return
    end
    end
    %The edit button for the length of the beam pulse
    function hCameraEditCallback(hObject, eventdata, handles)
        mtv_name = get(hObject,'string');
    end
    %The popupmenu for the selection of the graphs
    function hSelectGraphPopupmenuCallback(hObject,eventdata)
        graph_string=get(hObject,'String');
        graph_selection=get(hObject,'Value');
        graph_chosen=graph_string;
    if graph_selection  = current_graph_value(turn_display_selection)
    %in the case we push the button stop before it terminates the total number
of average requested first
        graph_average_string_num=1:average_number_history(turn_display_selection);
        graph_average_string_num=graph_average_string_num';
        graph_average_string=cellstr(int2str(graph_average_string_num));
        set(hSelectAverageGraphPopupmenu,'String',graph_average_string);
        set(hSelectAverageGraphPopupmenu,'Value',average_number_history
    (turn_display_selection));
```

```
graph_average_selection(turn_display_selection)=average_number_history
(turn_display_selection);
else
graph_average_string_num=1:current_graph_average_value(turn_display_selection);
graph_average_string_num=graph_average_string_num';
graph_average_string=cellstr(int2str(graph_average_string_num));
set(hSelectAverageGraphPopupmenu,'String',graph_average_string);
set(hSelectAverageGraphPopupmenu,'Value',current_graph_average_value
(turn_display_selection));
graph_average_selection(turn_display_selection)=current_graph_average_value
(turn_display_selection);
end
mtv_x=MTV(graph_selection,graph_average_selection,turn_display_selection).x;
mtv_y=MTV(graph_selection,graph_average_selection,turn_display_selection).y;
mtv_xproj_average=MTV(graph_selection,graph_average_selection,turn_display
_selection).xproj_average;
mtv_yproj_average=MTV(graph_selection,graph_average_selection,turn_display
_selection).yproj_average;
mtv_p_average=MTV(graph_selection,graph_average_selection,turn_display
_selection).p_average;
% a=0.9;
% b=1.1;
% r = a + (b-a).*rand(393,289);
% mtv_p=(MTV(graph_selection).p).*r;
%data fit
[xfit_average,x_q_average,x_dq_average,x_chisq_ndf_average] = agauss_fit(mtv_x,
mtv_xproj_average,'','1');
[yfit_average,y_q_average,y_dq_average,y_chisq_ndf_average] = agauss_fit(mtv_y,
mtv_yproj_average,'','1');
```

```
contourf(hImageAxes,mtv_x, mtv_y, mtv_p_average’);
plot(hHorizProjAxes,mtv_x,mtv_xproj_average,’b’,mtv_x,xfit_average,’g’);
plot(hVertProjAxes,mtv_y,mtv_yproj_average,’b’,mtv_y,yfit_average,’g’);
end
%The popup menu for the selection of the averages
function hSelectAverageGraphPopupmenuCallback(hObject,eventdata)
graph_average_string=get(hObject,’String’);
graph_average_selection=get(hObject,’Value’);
graph_average_chosen=graph_average_stringgraph_average_selection;
mtv_x=MTV(graph_selection,graph_average_selection,turn_display_selection).x;
mtv_y=MTV(graph_selection,graph_average_selection,turn_display_selection).y;
mtv_xproj_average=MTV(graph_selection,graph_average_selection,turn_display
_selection).xproj_average;
mtv_yproj_average=MTV(graph_selection,graph_average_selection,turn_display
_selection).yproj_average;
mtv_p_average=MTV(graph_selection,graph_average_selection,turn_display
_selection).p_average;
%data fit
[xfit_average,x_q_average,x_dq_average,x_chisq_ndf_average] = agauss_fit(mtv_x,
mtv_xproj_average,”,’1’);
[yfit_average,y_q_average,y_dq_average,y_chisq_ndf_average] = agauss_fit(mtv_y,
mtv_yproj_average,”,’1’);
contourf(hImageAxes,mtv_x, mtv_y, mtv_p_average’);
plot(hHorizProjAxes,mtv_x,mtv_xproj_average,’b’,mtv_x,xfit_average,’g’);
plot(hVertProjAxes,mtv_y,mtv_yproj_average,’b’,mtv_y,yfit_average,’g’);
end
function hTurnDisplayPopupmenuCallback(hObject,eventdata)
turn_display_string=get(hObject,’String’);
turn_display_selection=get(hObject,’Value’);
```

turn_display_chosen=turn_display_stringturn_display_selection;

graph_string_num=1:current_graph_value(turn_display_selection);

graph_string_num=graph_string_num’;

graph_string=cellstr(int2str(graph_string_num));

set(hSelectGraphPopupmenu,’String’,graph_string);

set(hSelectGraphPopupmenu,’Value’,current_graph_value(turn_display_selection));

graph_selection=current_graph_value(turn_display_selection);

graph_average_string_num=1:current_graph_average_value(turn_display_selection);

graph_average_string_num=graph_average_string_num’;

graph_average_string=cellstr(int2str(graph_average_string_num));

set(hSelectAverageGraphPopupmenu,’String’,graph_average_string);

set(hSelectAverageGraphPopupmenu,’Value’,current_graph_average_value

(turn_display_selection));

graph_average_selection=current_graph_average_value(turn_display_selection);

mtv_x=MTV(graph_selection,graph_average_selection,turn_display_selection).x;

mtv_y=MTV(graph_selection,graph_average_selection,turn_display_selection).y;

mtv_xproj_average=MTV(graph_selection,graph_average_selection,turn_display

_selection).xproj_average;

mtv_yproj_average=MTV(graph_selection,graph_average_selection,turn_display

_selection).yproj_average;

mtv_p_average=MTV(graph_selection,graph_average_selection,turn_display

_selection).p_average;

%data fit

[xfit_average,x_q_average,x_dq_average,x_chisq_ndf_average] = agauss_fit(mtv_x, mtv_xproj_average,”,’1’);

[yfit_average,y_q_average,y_dq_average,y_chisq_ndf_average] = agauss_fit(mtv_y, mtv_yproj_average,”,’1’);

contourf(hImageAxes,mtv_x, mtv_y, mtv_p_average’);

plot(hHorizProjAxes,mtv_x,mtv_xproj_average,’b’,mtv_x,xfit_average,’g’);

```
plot(hVertProjAxes,mtv_y,mtv_yproj_average,'b',mtv_y,yfit_average,'g');
%table sigma time
sigma_time=[time(:,turn_display_selection) sigma_x(:,turn_display_selection)
sigma_y(:,turn_display_selection)];
set(hSigmaTimeTable,'Data',sigma_time);
end
%The pop-up menu for the selection of the turn of the combiner ring
function hSelectTurnPopupmenuCallback(hObject,eventdata)
turn_string=get(hObject,'String');
turn_selection=get(hObject,'Value');
turn_chosen=turn_stringturn_selection;
end
%The button to save data
function hSaveButtonCallback(hObject, eventdata)
file_name_camera=inputdlg('Enter file name for the camera data');
file_name_sigma_time=inputdlg('Enter file name for the selected data of the
table');
folder_name=uigetdir;
display(average_number_history(scan_number));
display(current_graph_value);
display(turn_display_selection);
if w_save==2
    for i=1:current_graph_value(turn_display_selection)
    for j=1:average_number_history(turn_display_selection)
dir_file_name_camera=char(strcat(folder_name,'/',file_name_camera,'_delay',
num2str(i),'_aver',num2str(j),'.mat'));
    mtv_save=MTV(i,j,turn_display_selection);
    save(dir_file_name_camera, 'mtv_save');
    end
```

```
end
        dir_file_name_sigma_time=char(strcat(folder_name,'/',file_
name_sigma_time,'.mat'));
        save(dir_file_name_sigma_time, 'sigma_time');
else
        directory_file_name=char(strcat(folder_name,'/',file_name,'.mat'));
        save(dir_file_name_camera, 'mtv');
    end
end
%Push button "one acquisition"
function hAcquisitionButtonCallback(hObject, eventdata)
w_save=1;
set(hSelectGraphPopupmenu,'Enable','off');
set(hAcquisitionButton,'BackgroundColor','green');
set(hScanButton,'BackgroundColor','default');
pause(0.5);
%reading of the data of the gated camera in the combiner ring
[mtv_x,mtv_xproj,mtv_y,mtv_yproj,mtv_ts,mtv_p,mtv_maxpix]=
Jgetimage(mtv_name,0,2);
% if strcmp(background_sub,'yes')==1
[mtv_p, mtv_xproj, mtv_yproj] = background_subt_2(mtv_name, mtv_p,
mtv_xproj, mtv_yproj);
% end
mtv.x=mtv_x;
mtv.xproj=mtv_xproj;
mtv.y=mtv_y;
mtv.yproj=mtv_yproj;
mtv.ts=mtv_ts;
mtv.p=mtv_p;
```

```
mtv.maxpix=mtv_maxpix;
% mtv=load('/acc/oper/share/ctf/data/ctfmod/MatLab/BBolzon_GOzdemir
/mtv_scan_prog/CCS.MTV0980_10_09_29_12_57_20_
Cur_96.3265.mat');
% mtv_x=mtv.MTV.x;
% mtv_y=mtv.MTV.y;
% a=0.9;
% b=1.1;
% r_p = a + (b-a).*rand(393,289);
% r_x = a + (b-a).*rand(393,1);
% r_y = a + (b-a).*rand(289,1);
% mtv_xproj=mtv.MTV.xproj.*r_x;
% mtv_yproj=mtv.MTV.yproj.*r_y;
% mtv_p=mtv.MTV.p.*r_p;
contourf(hImageAxes,mtv_x, mtv_y, mtv_p');
xlabel(hImageAxes,'Position [mm]');
ylabel(hImageAxes,'Position [mm]');
zlabel(hImageAxes,'Intensity [mm]');
title(hImageAxes,'Image taken by the camera');
%data fit
[xfit,x_q,x_dq,x_chisq_ndf] = agauss_fit(mtv_x, mtv_xproj,'','1');
[yfit,y_q,y_dq,y_chisq_ndf] = agauss_fit(mtv_y, mtv_yproj,'','1');
sigma_x_one=abs(x_q(4));
sigma_y_one=abs(y_q(4));
sigma_time=[0 sigma_x_one sigma_y_one];
set(hSigmaTimeTable,'Data',sigma_time);
plot(hHorizProjAxes,mtv_x,mtv_xproj,'b',mtv_x,xfit,'g');
xlabel(hHorizProjAxes,'Position [mm]');
ylabel(hHorizProjAxes,'Intensity []');
```

```
legend(hHorizProjAxes,'Data','Fit');
title(hHorizProjAxes,'Horizontal projection');
grid(hHorizProjAxes,'on');
plot(hVertProjAxes,mtv_y,mtv_yproj,'b',mtv_y,yfit,'g');
xlabel(hVertProjAxes,'Position [mm]');
ylabel(hVertProjAxes,'Intensity []');
legend(hVertProjAxes,'Data','Fit');
title(hVertProjAxes,'Vertical projection');
grid(hVertProjAxes,'on');
% %plot energy spread
% Itot=0;
% avgx=0;
% avgy=0;
% Itot = sum(sum(mtv_p));
%       for i=1:size(mtv_x,1)
%     for j=1:size(mtv_y,1)
%     avgx = avgx + mtv_x(i,1)*mtv_p(i,j);
%     avgy = avgy + mtv_y(j,1)*mtv_p(i,j);
%   end
% end
% avgx = avgx/Itot;
% avgy = avgy/Itot;
% %mtv_x = mtv_x - avgx;
% %mtv_y = mtv_y - avgy;
% %[dummy,hc]=contourf(mtv_x,mtv_y,mtv_p',20);  %the data is now centered
% MAXy = find(round(mtv_y) == round(avgy));
% MAXy = MAXy(1,1);
% MeanScreen = sum(mtv_x.*mtv_p(:,MAXy))/sum(mtv_p(:,MAXy));
```

```
% SigmaScreen = sqrt(sum((mtv_x-MeanScreen).*(mtv_x-MeanScreen)
.*(mtv_p(:,MAXy)))/sum(mtv_p(:,MAXy)));
% figure;
% plot(mtv_x,mtv_p(:,MAXy));
% hold on;
% plot(mtv_x,max(mtv_p(:,MAXy))*exp(-(mtv_x-MeanScreen).*
(mtv_x-MeanScreen)/(2*SigmaScreen*SigmaScreen)),'k','Linewidth',3);
set(hAcquisitionButton,'BackgroundColor','red');
set(hSaveButton,'Enable','on');
end
function hScanButtonCallback(hObject, eventdata)
set(hResetButton,'BackgroundColor','default');
scan_number=scan_number+1;
turn_selection_history(scan_number)=turn_selection;
turn_chosen_historyscan_number,1=turn_chosen;
delay_number_history(scan_number,1)=delay_number;
average_number_history(scan_number,1)=average_number;
if scan_number > 1
    delay_turn=turn_delay(turn_selection_history,scan_number);
    timing_delay(delay_turn);
    set(hDelayTurnText2,'String',[int2str(delay_turn) 'ns']);
end
set(hDelayValueText2,'String', '0 ns');
set(hSaveButton,'Enable','off');
w_save=2;
set(hScanButton,'BackgroundColor','green');
set(hAcquisitionButton,'BackgroundColor','default');
set(hStopButton,'BackgroundColor','default');
set(hSelectGraphPopupmenu,'Enable','off');
```

```
set(hSelectAverageGraphPopupmenu,’Enable’,’off’);
set(hTurnDisplayPopupmenu,’Enable’,’off’);
graph_string_num=1:delay_number;
graph_string_num=graph_string_num’;
graph_string=cellstr(int2str(graph_string_num));
set(hSelectGraphPopupmenu,’String’,graph_string);
set(hSelectGraphPopupmenu,’Value’,delay_number);
graph_selection=get(hSelectGraphPopupmenu,’Value’);
graph_average_string_num=1:average_number;
graph_average_string_num=graph_average_string_num’;
graph_average_string=cellstr(int2str(graph_average_string_num));
set(hSelectAverageGraphPopupmenu,’String’,graph_average_string);
set(hSelectAverageGraphPopupmenu,’Value’,average_number);
graph_average_selection=get(hSelectAverageGraphPopupmenu,’Value’);
set(hTurnDisplayPopupmenu,’String’,turn_chosen_history);
set(hTurnDisplayPopupmenu,’Value’,scan_number);
turn_display_selection=get(hTurnDisplayPopupmenu,’Value’);
z=0;
for i=1:delay_number
pause(0.5);
mtv_p_average=0;
mtv_xproj_average=0;
mtv_yproj_average=0;
%control the delay
if i¿1
    timing_delay(gate_length);
    pause(2);
    set(hDelayValueText2,’String’,[int2str(gate_length*(i-1)) ’ns’]);
end
```

```
for j=1:average_number
        if strcmp(stop_scan,'no')
    z=z+1;
    time(z,scan_number)=(i-1)*gate_length;
    set(hDelayCounterText2,'String',int2str(i));
    set(hAverageCounterText2,'String',int2str(j));
    pause(0.5);
    %reading of the data of the gated camera in the combiner ring
[mtv_x,mtv_xproj,mtv_y,mtv_yproj,mtv_ts,mtv_p,mtv_maxpix]=
Jgetimage(mtv_name,0,2);
    if strcmp(background_sub,'yes')==1
    [mtv_p, mtv_xproj, mtv_yproj] = background_subt_2(mtv_name, mtv_p,
mtv_xproj, mtv_yproj);
end
    mtv_p_average=(mtv_p+mtv_p_average)/j;
    mtv_xproj_average=(mtv_xproj+ mtv_xproj_average)/j;
    mtv_yproj_average=(mtv_yproj+ mtv_yproj_average)/j;
    MTV(i,j,scan_number).x=mtv_x;
MTV(i,j,scan_number).y=mtv_y;
MTV(i,j,scan_number).xproj=mtv_xproj;
MTV(i,j,scan_number).yproj=mtv_yproj;
MTV(i,j,scan_number).p=mtv_p;
MTV(i,j,scan_number).xproj_average=mtv_xproj_average;
MTV(i,j,scan_number).yproj_average=mtv_yproj_average;
MTV(i,j,scan_number).p_average=mtv_p_average;
MTV(i,j,scan_number).ts=mtv_ts;
MTV(i,j,scan_number).maxpix=mtv_maxpix;
% mtv_mtv=load('/acc/oper/share/ctf/data/ctfmod/MatLab/
% BBolzon_GOzdemir/mtv_scan_prog/CCS.MTV0980_10_09_29_12_57_20_
```

% Cur_96.3265.mat');

% mtv=mtv_mtv.MTV;

% a=0.9;

% b=1.1;

% r$_p = a + (b - a). * rand(393, 289)$;

% r$_x = a + (b - a). * rand(393, 1)$;

% r$_y = a + (b - a). * rand(289, 1)$;

% mtv.xproj=mtv.xproj.*r_x;

% mtv.yproj=mtv.yproj.*r_y;

% mtv.p=mtv.p.*r_p;

% mtv_x=mtv.x;

% mtv_y=mtv.y;

% mtv_p=mtv.p;

% mtv_xproj=mtv.xproj;

% mtv_yproj=mtv.yproj;

% mtv_p_average=(mtv.p+mtv_p_average)/j;

% mtv_xproj_average=(mtv.xproj+ mtv_xproj_average)/j;

% mtv_yproj_average=(mtv.yproj+ mtv_yproj_average)/j;

% MTV(i,j,scan_number).x=mtv.x;

% MTV(i,j,scan_number).y=mtv.y;

% MTV(i,j,scan_number).xproj=mtv_xproj;

% MTV(i,j,scan_number).yproj=mtv_yproj;

% MTV(i,j,scan_number).p=mtv_p;

% MTV(i,j,scan_number).xproj_average=mtv_xproj_average;

% MTV(i,j,scan_number).yproj_average=mtv_yproj_average;

% MTV(i,j,scan_number).p_average=mtv_p_average;

% MTV(i,j,scan_number).ts=mtv.ts;

% MTV(i,j,scan_number).maxpix=mtv.maxpix;

%3D plot of the camera data

[a,b]=contourf(hImageAxes,mtv_x, mtv_y, mtv_p_average');

set(b,'LineStyle','none');

xlabel(hImageAxes,'Position [mm]');

ylabel(hImageAxes,'Position [mm]');

zlabel(hImageAxes,'Intensity [mm]');

title(hImageAxes,'Image taken by the camera');

%data fit

[xfit,x_q,x_dq,x_chisq_ndf] = agauss_fit(mtv_x, mtv_xproj,”,'1');

[yfit,y_q,y_dq,y_chisq_ndf] = agauss_fit(mtv_y, mtv_yproj,”,'1');

[xfit_average,x_q_average,x_dq_average,x_chisq_ndf_average] = agauss_fit(mtv_x, mtv_xproj_average,”,'1');

[yfit_average,y_q_average,y_dq_average,y_chisq_ndf_average] = agauss_fit(mtv_y, mtv_yproj_average,”,'1');

plot(hHorizProjAxes,mtv_x,mtv_xproj_average,'b',mtv_x,xfit_average,'g');

xlabel(hHorizProjAxes,'Position [mm]');

ylabel(hHorizProjAxes,'Intensity');

legend(hHorizProjAxes,'Data','Fit');

title(hHorizProjAxes,'Horizontal projection');

grid(hHorizProjAxes,'on');

plot(hVertProjAxes,mtv_y,mtv_yproj_average,'b',mtv_y,yfit_average,'g');

xlabel(hVertProjAxes,'Position [mm]');

ylabel(hVertProjAxes,'Intensity');

legend(hVertProjAxes,'Data','Fit');

title(hVertProjAxes,'Vertical projection');

grid(hVertProjAxes,'on');

pause(0.5);

sigma_x(z,scan_number)=abs(x_q(4));

sigma_y(z,scan_number)=abs(y_q(4));

sigma_time=[time(:,scan_number) sigma_x(:,scan_number) sigma_y

```
(:,scan_number)];
set(hSigmaTimeTable,'Data',sigma_time);
%plot sigma versus time
i_scatter=size(time,1)*scan_number;
time_scatter(i_scatter-size(time,1)+1:i_scatter)=time(:,scan_number);
sigma_x_scatter(i_scatter-size(time,1)+1:i_scatter)=sigma_x(:,scan_number);
sigma_y_scatter(i_scatter-size(time,1)+1:i_scatter)=sigma_y(:,scan_number);
szX= size(time);
group = reshape(repmat(1:szX(2),szX(1),1),[],1);
set(hMainFigure,'CurrentAxes',hSigmaTimeVerticalAxes);
h_gscatter_y=gscatter(time_scatter, sigma_y_scatter,group,[],[],[],[],
'Time [ns]','Sigma y [mm]');
legend(h_gscatter_y,turn_chosen_history:,1);
title(hSigmaTimeVerticalAxes,'Vertical beam size (sigma) versus time');
grid(hSigmaTimeVerticalAxes,'on');
hold all;
set(hMainFigure,'CurrentAxes',hSigmaTimeHorizontalAxes);
h_gscatter_x=gscatter(time_scatter, sigma_x_scatter,group,[],[],[],[],
'Time [ns]','Sigma x [mm]');
legend(h_gscatter_x,turn_chosen_history:,1);
title(hSigmaTimeHorizontalAxes,'Horizontal beam size (sigma) versus time');
grid(hSigmaTimeHorizontalAxes,'on');
hold all;
elseif strcmp(stop_scan,'yes')
    set(hScanButton,'BackgroundColor','default');
    set(hStopButton,'Value',0);
    stop_scan='no';
    if i¿ 1  j==1
    graph_string_num=1:i-1;
```

```
        current_graph_value(scan_number)=i-1;
        graph_average_string_num=1:average_number;
        current_graph_average_value(scan_number)=average_number;
else
        graph_string_num=1:i;
        current_graph_value(scan_number)=i;
        graph_average_string_num=1:j-1;
        current_graph_average_value(scan_number)=j-1;
end
graph_string_num=graph_string_num';
graph_string=cellstr(int2str(graph_string_num));
set(hSelectGraphPopupmenu,'String',graph_string);
set(hSelectGraphPopupmenu,'Value',current_graph_value(scan_number));
set(hSelectGraphPopupmenu,'Enable','on');
graph_average_string_num=graph_average_string_num';
graph_average_string=cellstr(int2str(graph_average_string_num));
set(hSelectAverageGraphPopupmenu,'String',graph_average_string);
set(hSelectAverageGraphPopupmenu,'Value',current_graph_
average_value(scan_number));
set(hSelectAverageGraphPopupmenu,'Enable','on')
set(hTurnDisplayPopupmenu,'Value', scan_number);
set(hTurnDisplayPopupmenu,'Enable','on');
turn_display_selection=scan_number; if (i==1  j¡=average_number)
set(hSaveButton,'Enable','on');
          end
        return;
        end
end
% %plot energy spread
```

```
% Itot=0;
% avgx=0;
% avgy=0;
% Itot = sum(sum(mtv_p));
% for i=1:size(mtv_x,1)
%           for j=1:size(mtv_y,1)
%       avgx = avgx + mtv_x(i,1)*mtv_p(i,j);
%       avgy = avgy + mtv_y(j,1)*mtv_p(i,j);
%       end
% end
% avgx = avgx/Itot;
% avgy = avgy/Itot;
% %mtv_x = mtv_x - avgx;
% %mtv_y = mtv_y - avgy;
% %[dummy,hc]=contourf(mtv_x,mtv_y,mtv_p',20);    %the data is now centered
% MAXy = find(round(mtv_y) == round(avgy));
% MAXy = MAXy(1,1);
% MeanScreen = sum(mtv_x.*mtv_p(:,MAXy))/sum(mtv_p(:,MAXy));
% SigmaScreen = sqrt(sum((mtv_x-MeanScreen).*(mtv_x-MeanScreen)
.*(mtv_p(:,MAXy)))/sum(mtv_p(:,MAXy)));
% figure;
% plot(mtv_x,mtv_p(:,MAXy));
% hold on;
% plot(mtv_x,max(mtv_p(:,MAXy))*exp(-(mtv_x-MeanScreen)
.*(mtv_x-MeanScreen)/(2*SigmaScreen*SigmaScreen)),'k','Linewidth',3);
end
set(hScanButton,'BackgroundColor','red');
set(hSelectGraphPopupmenu,'Value',delay_number);
```

```
set(hSelectGraphPopupmenu,'Enable','on');
set(hSelectAverageGraphPopupmenu,'Value',average_number);
set(hSelectAverageGraphPopupmenu,'Enable','on');
set(hTurnDisplayPopupmenu,'Value', scan_number);
set(hTurnDisplayPopupmenu,'Enable','on');
%hold(hSigmaTimeAxes,'off'); probably this command has to be deleted let's see...
current_graph_value(scan_number)=i;
current_graph_average_value(scan_number)=j;
set(hSaveButton,'Enable','on');
display(turn_display_selection);
end
function select_callback(hObject, eventdata)
% Get the list of currently selected table cells
sel = eventdata.Indices; % Get selection indices (row, col)
% Noncontiguous selections are ok
sel= unique(sel(:,1));
table = get(hObject,'Data'); % Get copy of uitable data
% Get vectors of x,y values for each column in the selection;
sigma_time=[table(sel, 1) table(sel, 2) table(sel, 3)];
end
function hOpenMenuitemCallback(hObject, eventdata)
% Callback function run when the Open menu item is selected
file = uigetfile('*.m');
if isequal(file, 0)
open(file);
end
end
function hSaveMenuitemCallback(hObject, eventdata)
```

```
[filename, pathname, filterindex]=uiputfile('','Save as');
file_path_name=char(strcat(pathname,filename))
saveas(hMainFigure,file_path_name);
end
function hPrintMenuitemCallback(hObject, eventdata)
% Callback function run when the Print menu item is selected
printdlg(hMainFigure);
end
function hCloseMenuitemCallback(hObject, eventdata)
% Callback function run when the Close menu item is selected
selection = ...
questdlg(['Close ' get(hMainFigure,'Name') '?'],...
['Close ' get(hMainFigure,'Name') '...'],...
'Yes','No','Yes');
if strcmp(selection,'No')
        return;
    end
    delete(hMainFigure);
end
end
```