September, 2012

DESY Summer Student Report

MAMA @ FLASH

Mean momentum and momentum spread analysis at the FLASH photo-injector (IDUMP)

Luca Somaschini
Università degli Studi di Milano

Abstract

The free-electron laser FLASH (Free-Electron Laser in Hamburg) at DESY is currently one of the most important Free Electron Lasers (FELs) all over the world. In order to initiate the Self-Amplified Spontaneous Emission (SASE) process of the FEL it is necessary to stress on the beam quality. In particular a precise control of the transverse emittance, a high peak current and a small energy spread are fundamental requirements. To measure and monitor the momentum spread after the laser driven photo-injector a dipole magnet was installed and operated as a spectrometer. During this summer student program a Matlab code with a graphical interface was developed, in order to acquire the image of the screen within the dipole spectrometer to perform measurements of mean momentum and momentum spread in the longitudinal direction of the electron bunch.

Contents

1 The FLASH facility

In 1991 the international TESLA collaboration of 55 institutes in 12 countries started a new project to push the performances of superconducting cavities to ultimate gradients, in order to demonstrate the feasibility of a electron-positron linear collider. It was decided to build the TESLA Test Facility (TTF) at DESY [5]. Meanwhile scientists started to focus on the possibility of using a new generation LINAC to generate X-Ray laser pulses. As a result a working group, at DESY, started to investigate the possibility of using the TTF as a FEL light source. Positive results of some tests and upgrades on the TTF turned the initial test facility into a FEL user facility with a two times longer accelerator complex and a new experimental hall (fig. 1).

The facilities has been renamed to FLASH and it has been updated in 2010. It's now able to deliver an electron beam with an energy up to $1.25\,GeV$ providing laser-like light pulses with a temporal length of $50 - 200\,fs$ FWHM and a wavelength of $45 - 4.1\,nm$ [5].

1.1 FLASH Layout

The FLASH facility[3] consists of a $256\,m$ long electron linac (fig. 2), followed by a photon transport line equipped with photon diagnostics and 5 experimental beam lines for users.

Electron bunches are generated by photo emission and then accelerated by RF

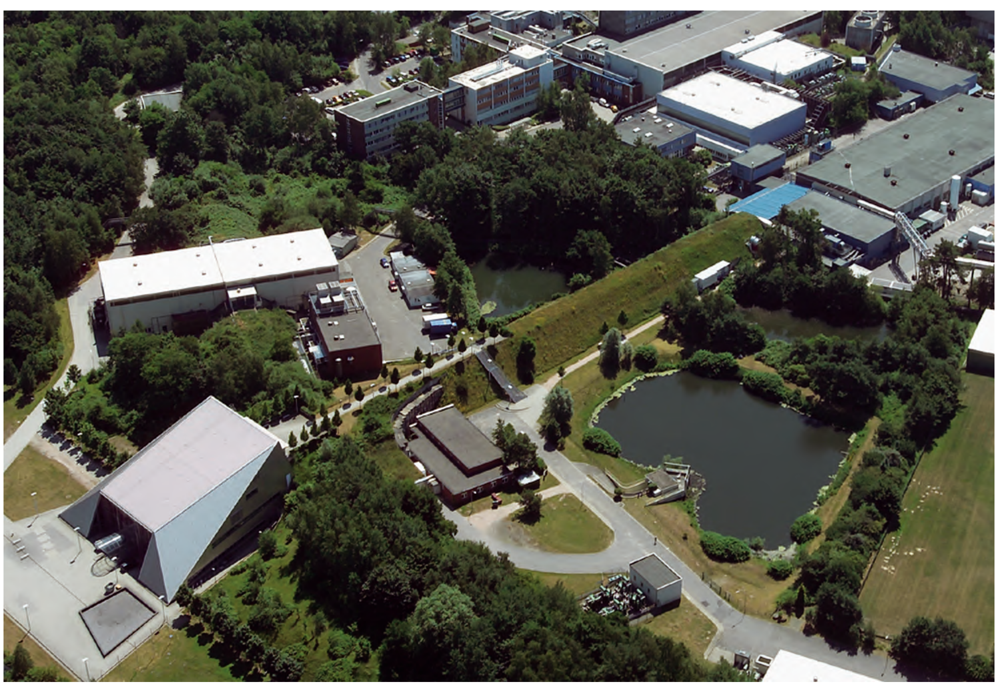


Figure 1: Aerial view of FLASH site and the experimental hall

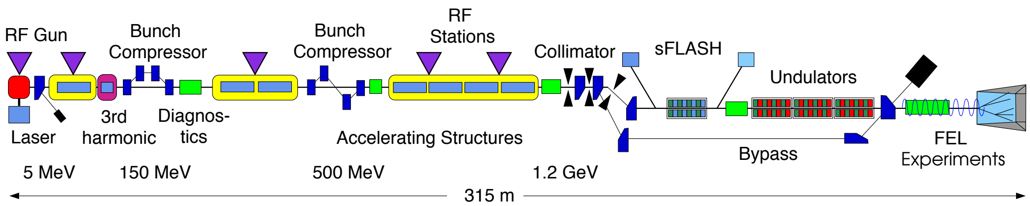


Figure 2: FLASH layout

cavities: all these ones are typically pulsed at a frequency of $10\,Hz$. FLASH consists of a normal-conductive photo-injector and seven $1.3\,GHz$ superconducting accelerating modules, with eight nine-cell niobium cavities each. The theoretical maximum energy gain per module ranges from $180\,MeV$ to $240\,MeV$ according to the module.

In order to reduce space-charge effects in the low energy regime[1] and since high peak currents together with small emittances cannot be preserved at the photocathode, the bunches have to be generated significantly longer (with peak current of about $50\,A$) and compressed at higher energies, to reach typical intensities of about $1\,kA$. However non-linearity of the RF causes a non-linearity in the longitudinal phase space distribution, leading to a curvature of the longitudinal phase space distribution. A third harmonic module ($3.9\,GHz$) consisting of four cavities produced at Fermilab is used in order to linearize the longitudinal phase space distribution and to achieve an optimized compression.

The electrons in the LINAC have speeds very close to the speed of light and the velocity difference between different electrons is far to small for a trailing electron to catch up with a leading one. For this reason two electro-magnetic chicanes have been introduced: in the cavities before the bunch compressors an energy slope is imprinted to bunch. Once they have an energy slope, provided by cavities, bunches are forced to pass into an electro-magnetic chicane, such that particles with different energy travels along different paths, in particular higher energetic particles, in the tail of the bunch, travel a shorter distance than the particles at the head of the bunch.

In order to reach SASE saturation it is important to achieve a high peak current [4].

After the acceleration and the compression of the beam, the electron bunch travels through a system of 6 undulators of $4.5\,m$ of length each, providing the desired laser light. The electron beam is finally deflected by a dipole magnet and sent to a beam dump, while the FEL light is sent to the experimental hall.

[1]Space charge effect is less important at higher energy

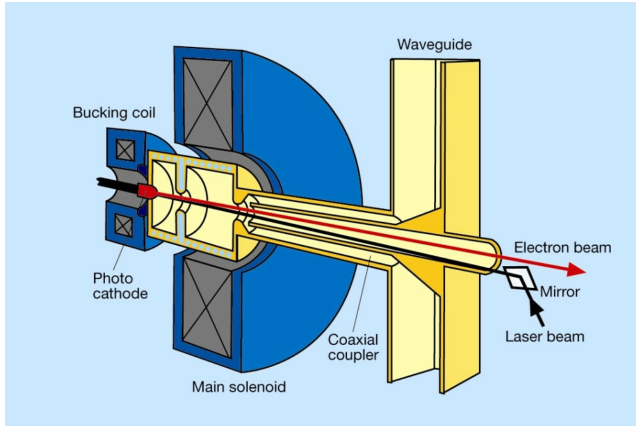


Figure 3: FLASH GUN

1.2 GUN

An important requirement on the beam condition is a small transverse emittance. A small emittance, that can naively be imagined as the product of the radial beam size and the beam divergence, means that it is possible to maintain the beam size small over a long distance. This turns out to be fundamental in the undulators of a FEL, which can be longer than $100\,m$. The emittance of the electron bunch is determined in the gun and can only degrade afterwards. It is also necessary to have a small energy spread within an electron bunch, in order to initiate the SASE process.

The high charge density in the bunches can be accomplished by means of photocathodes illuminated with short ultraviolet laser pulses. The injector (fig. 3) consists of a cathode of molybdenum coated with a Cs_2Te layer to achieve a high efficiency for the photoelectric emission. UV pulses with a wavelength of $262\,nm$ are generated by a solid state laser system, creating a spot on the cathode, which is approximately flat transversally, with a diameter of typically $1.2\,mm$. The longitudinal laser distribution is a Gaussian with a RMS duration of $6.5\,ps$. The cathode is placed a the end of a normal conducting RF cavity. It is important to accelerate the particles immediately, to avoid Coulomb repulsion called space charge effect. The accelerating field is provided by a 1.5 cell accelerating cavity made by copper which is operated at $1.3\,GHz$. The laser pulse and the cavity are synchronized with a precision better than $100\,fs$. The current laser is capable to produce 800 pulses at $1\,MHz$ for each macropulse of $10\,Hz$ operation. Moreover several frequencies between $1\,MHz$ and $40\,kHz$ can be provided.

In table 1 typical FLASH parameters are shown.

2 Beam Analysis

The requirements on the electron beam quality are very demanding: for this reason high resolution diagnostic instruments are essential for the operation.

e^- :		
emittance $\beta\gamma\epsilon_{x,y}$		
(1 nC, on–crest, 90% rms)	1.4	mm mrad
charge	0.2 - 1	nC
beam energy	375 - 1250	MeV
bunches / train	1 - 800	
bunch spacing	1 - 25	μs
train repetition frequency	10	Hz
γ :		
wavelength (fundamental)	4.1 - 45	nm
average single pulse energy	10 - 400	μJ
pulse duration (fwhm)	50 - 200	fs
spectral width (fwhm)	0.7 - 2.0	%
peak power	1 - 3	GW
peak brilliance	10^{29} - 10^{31}	(+)
average brilliance	10^{17} - 10^{21}	(+)
(+) : photons/(s mm^2 $mrad^2$ 0.1%bw)		

Table 1: Typical FLASH parameters [3]

It turns out to be extremely important to have information on the longitudinal momentum distribution after the photo-injector: on the one hand at low energy the beam is strongly afflicted by space charge effect and, on the other hand, the emittance can only degrade afterwad.

To map the momentum distribution after the electron gun, a spectrometer system is used, providing a measurement of the momentum distribution.

The most important component of the spectrometer is a magnetic dipole (fig. 4). When the dipole magnet is turned on, electrons accelerated by the photo-injector are deflected from their standard, straight trajectory toward a screen contained in the dispersive section (called IDUMP). Here the spatial distribution which represents the momentum distribution of the particles is measured.

The main effect of the dipole magnetic field is to impress different paths to electrons with different longitudinal momenta, translating the momentum spread into a spatial distribution: this property of the dipole magnet is called dispersion. In order to have a precise measurement, it is important to reduce the beam size, such that the figure obtained on the screen only depends on the longitudinal momentum differences and not on the transverse beam size and divergence. This could be achieved by means of a slit, cutting the beam before it enters the dipole; the remaining influence of the beam size is minimized by optimizing the edge angles of the dipole.

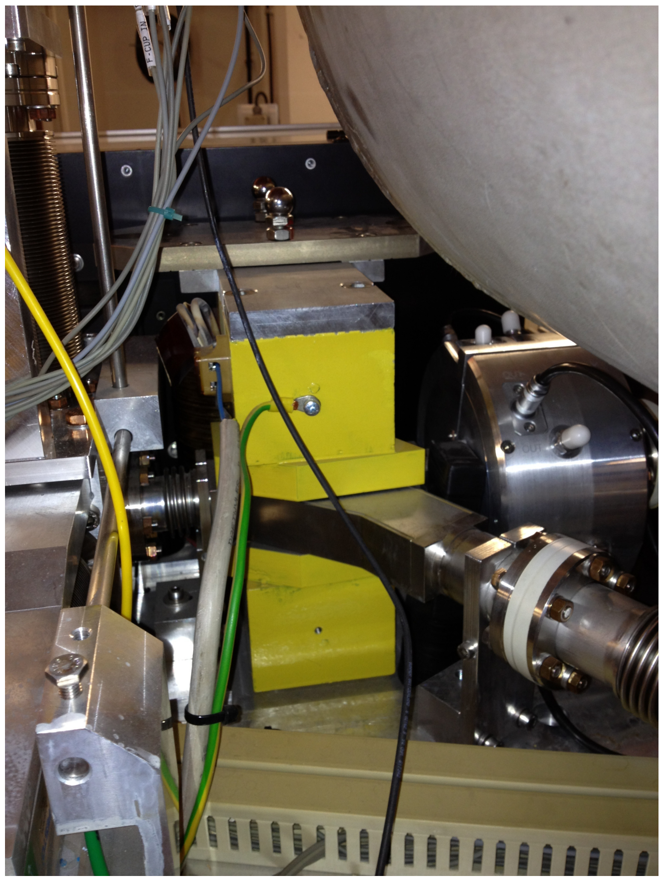


Figure 4: FLASH gun spectrometer dipole magnet

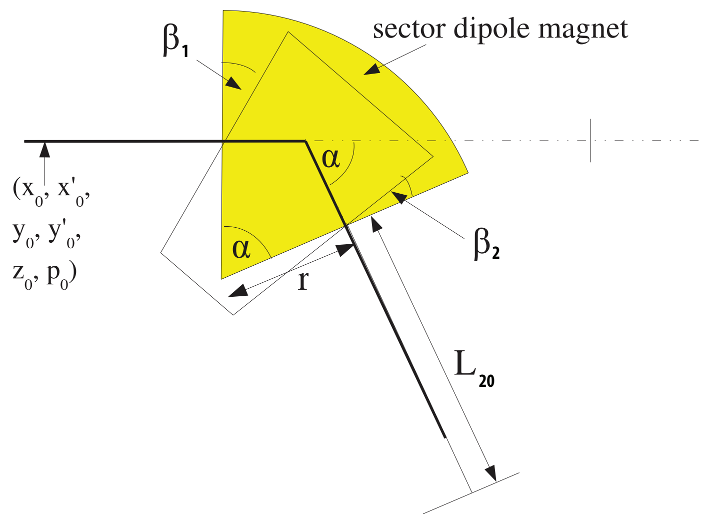


Figure 5: Dipole magnet scheme

2.1 Dipole Magnet: Hysteresis Cycle

In order to provide precise momentum measurements, as shown in the following paragraphs, it is important to have an accurate knowledge of the field produced by the dipole (fig. 7). The dipole field can't be measure every time a momentum analysis is running, since it requires access to the dipole gap. However the magnetic field is a fixed property of a dipole, for a specific current, so it was measured once before the operation of the dipole magnet.

Figure 5 shows a schematic view of the dipole magnet.

The momentum at the center of the screen in the dispersive section can be computed as:

$$p_0 = \frac{e}{\alpha} \left| B_{dipole} \right| l_{eff} \tag{1}$$

where e is the electron charge, α is the deflection angle impressed by the dipole (60°, for FLASH injector) and B_{dipole} is the magnetic field strength between the pole shoes of the dipole magnet, in the center. l_{eff} is the effective length, defined as the integral of the magnetic field over the particle trajectory [2]:

$$l_{eff} = \frac{1}{B_{max}} \int B\left(l\right) dl$$

The effective length takes into account the fact that the field is not uniform and not perfectly confined into the dipole magnet. This allows us to compute p_0 simply in terms of the field at the center of the dipole (fig. 6).

In the FLASH IDUMP dipole magnet, the effective magnetic length is given by

$$l_{eff} = 148.79 \pm 0.56\, mm$$

.

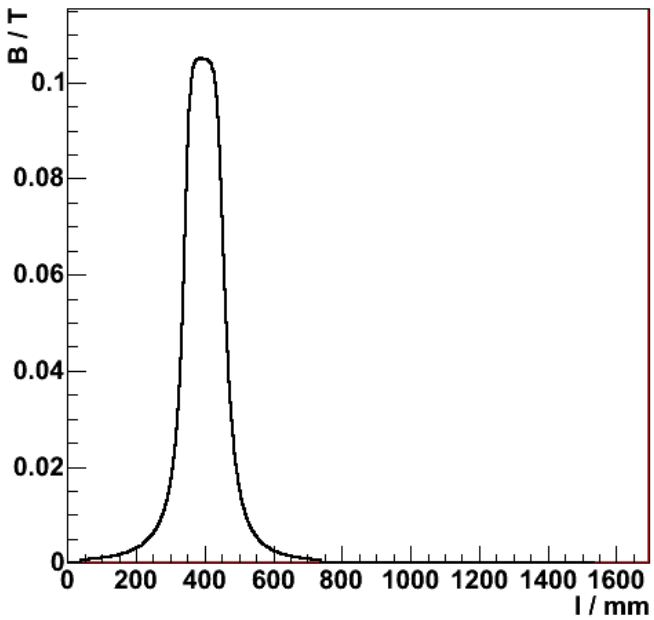


Figure 6: Magnetic field along the particle trajectory

Since the magnet has a ferromagnetic joke, it features an hysteresis cycle that, even if it is small, it is not negligible. To compute the effective length it is important to map the magnetic field inside the dipole magnet in 3 dimensions and for different currents along the cycle. The field map is obtained by means of a hall probe that can be moved in 3 dimensions. The magnetic field shows edge effects that have to be considered when computing the momentum associated to a certain current (in particular into the computation of the effective length).

The magnetic field at the center of the dipole magnet was measured at different currents with steps of $0.5\,A$ in the range from $-3.5\,A$ to $3.5\,A$. For each current step eight measurements where recorded, cycling anti clock-wise: four with increasing current and four with decreasing current (fig. 8).

We also computed the error of the field, considering the standard deviation on the four different sets of measurements. The error coming from the deviation is extremely small. Further improvements (but non essential, as shown later) could be achieved by taking into account the precision of the hall probe used to make the measurements.

To have an accurate knowledge of the field at the center of the dipole B_{dipole}, it is necessary to fit the hysteresis cycle and extrapolate a function to describe the behavior of magnetic field as a function of current.

We performed the fit with four different functions:

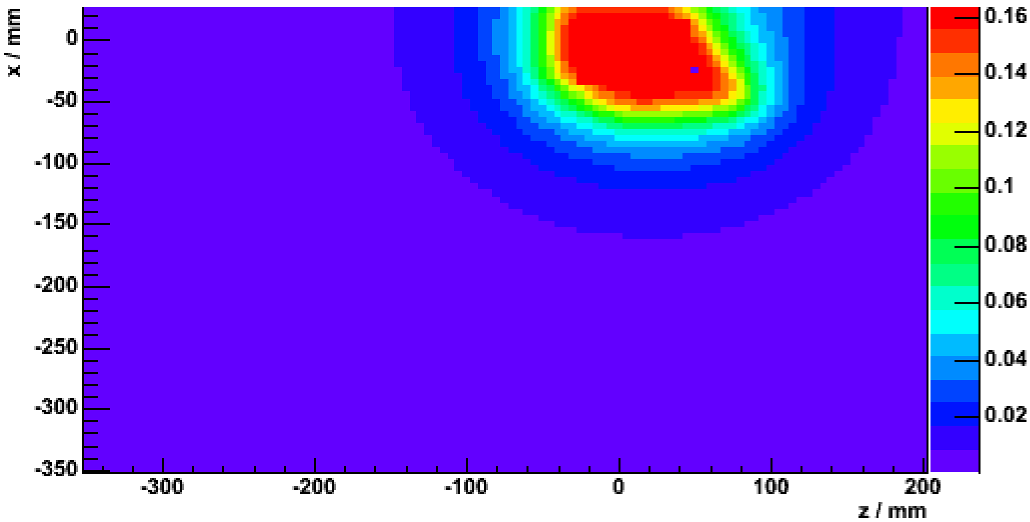


Figure 7: Magnetic field map of the IDUMP dipole magnet

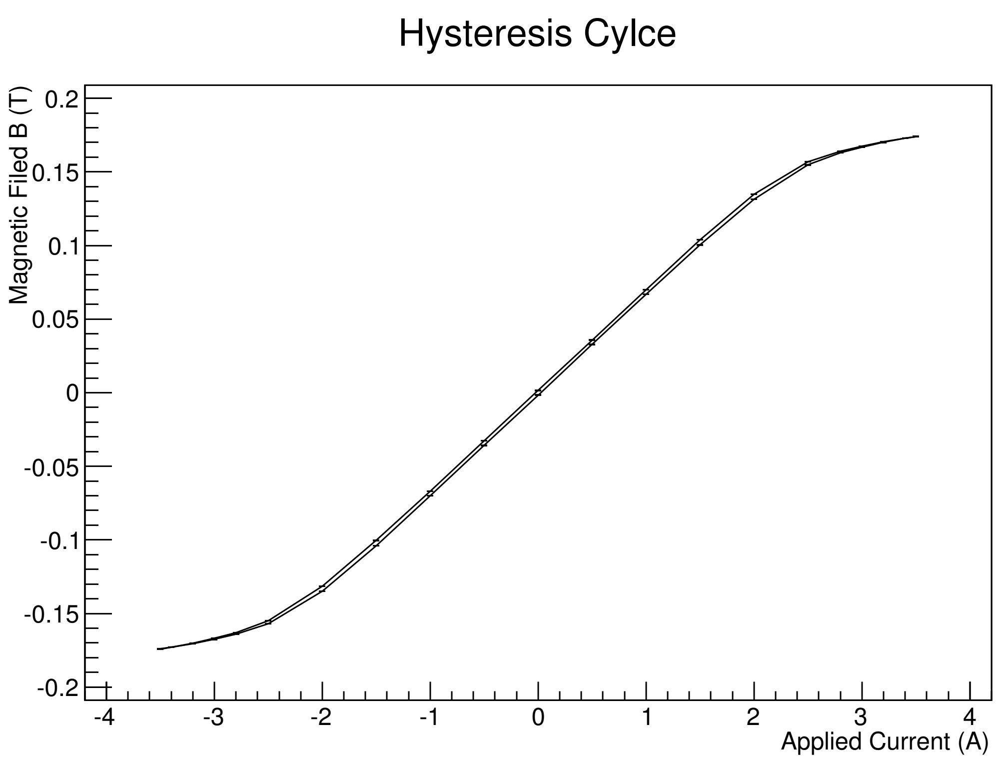


Figure 8: Hysteresis Cycle of the IDUMP dipole magnet. *The picture shows the hysteresis cycle of the dipole magnet. The magnetic field difference, even if small, translates into a relevant difference in momentum: it is fundamental to distinguish the case of increasing current from the case of decreasing current.*

- An third degree polynomial

$$B = a + bI + cI^2 + dI^3$$

- A sixth degree polynomial

$$B = a + bI + cI^2 + dI^3 + eI^4 + fI^5 + gI^6$$

- An exponential with three parameters

$$B = \frac{a\,(I-b)}{1 + e^{-\frac{c}{(I-b)^2}}}$$

- An exponential with four parameters

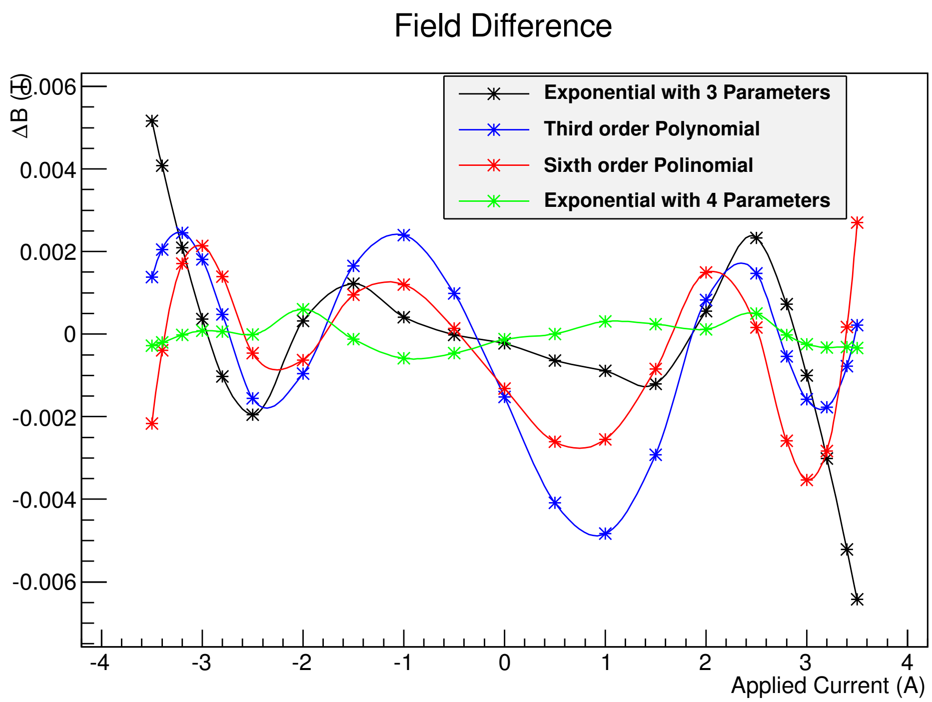
$$B = \frac{a\,(I-b)}{d + e^{-\frac{c}{(I-b)^2}}}$$

In [1] it was proposed to fit the magnetic field as a function of the dipole current (in the case of decreasing current) with an exponential with three parameters, but this could lead to an error in the mean momentum up to $0.3\,MeV/c$. For this reason we decided to check the improvements brought by other fitting functions.

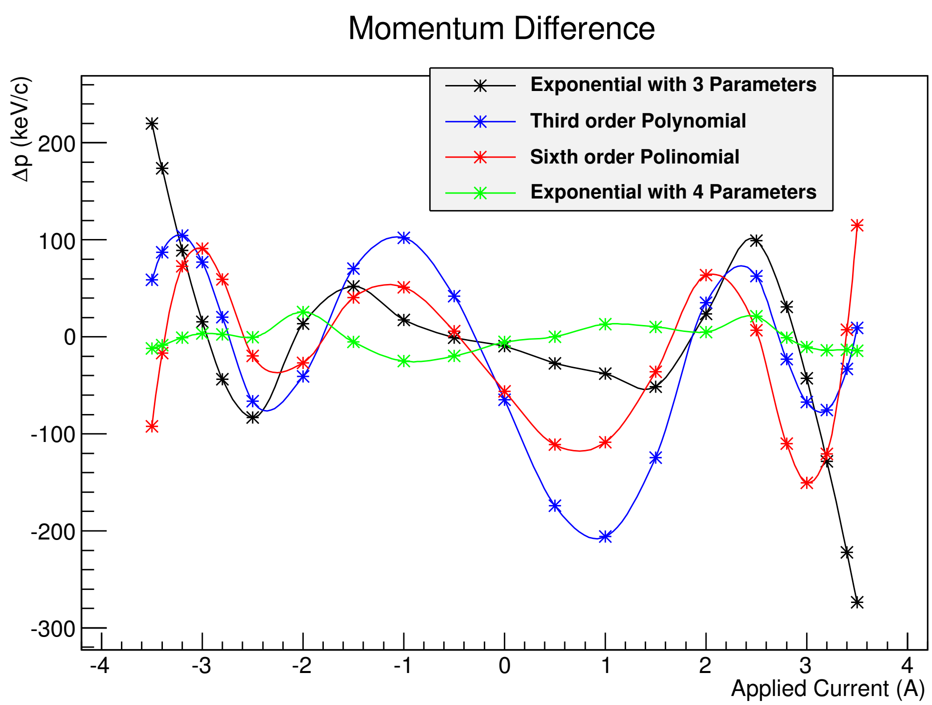
We computed the fitting parameters by means of a ROOT fit and then we computed the difference between the measured field and the one extracted from the fit, only for currents corresponding to measurements. Then we plotted the difference of field as a function of current. Since what really cares is the momentum we also converted these differences of magnetic field in terms of momentum differences (fig. 9). It can be seen from the graphic that the difference between the measured field and the fitted field is smaller than $5\,mT$, and depending on the current and on the fitting function, they vary between a few percent and a few per mil $\frac{\Delta B}{B}$. However this turns into an error of the momentum (using the formula 1), in order of hundreds of keV/c, which is larger than the required resolution. The best fitting was obtained using the exponential function with four parameters.

We decided also to plot separately the behavior of this best fitting function with the parameters given in table 2, including the error bars (fig. 10). Here it comes the usual question: is it correct to add parameters to a fitting function to get better results? How many parameters can be added? It is clear that the more parameters you add the higher the chances to get a better fitting results increase. But it is also true that in this case we are dealing with an empirical function: we don't want to extract a theory on the dipole field, we simply want to have the function that gives us the smaller error while computing the magnetic field, independently on any theoretical foundation.

For this reason we decide to use the four parameters exponential. We also decide to compute the error on the difference.



(a) Field Difference



(b) Momentum difference

Figure 9: Difference between measured and fitted values of IDUMP dipole magnet magnetic field (a) and of the resulting momentum (b)

Parameter	Value	Error
a	$4415 \cdot 10^{-5}$	$7 \cdot 10^{-5}$
b	$251 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
c	17.56	0.02
d	0.649	0.001

Table 2: Fitting Parameters and errors for the four parameters exponential fitting for decreasing current

The difference, according to formula 1, is computed as:

$$\Delta p_0 = \frac{e}{\alpha} l_{eff} \left(B_{meas} - B_{B_{fit}} \right)$$

So the error on the difference is:

$$\left(\frac{\sigma_{\Delta p_0}}{p_0} \right)^2 = \left(\frac{e}{\alpha} \right)^2 \left[\left(\frac{\sigma_{l_{eff}}}{l_{eff}} \right)^2 + \left(\frac{\sigma_{\Delta B}}{\Delta B} \right)^2 \right]$$

where:

$$\sigma^2_{\Delta B} = \sigma^2_{B_{meas}} + \sigma^2_{B_{fit}}$$

The error on the measured field is computed as described above. The error on the fit is computed taking into account the fitting function and the error on the fit parameters a_i is derived from the usual error propagation.

$$\sigma^2_{B_{fit}} = \sum_i \left(\frac{\partial B_{fit}}{\partial a_i} \right)^2 \sigma^2_{a_i}$$

As we can see from the detailed plot the error bars are almost compatible with zero and the difference between the measured and the fitted field are considerably smaller than the ones for other fitting functions. The plots is compatible with zero even if the error bars doesn't include the uncertainty deriving from the precision of the hall probe: this precision is not available now, but could be included in the future, once this information is made available.

3 The MAMA Code for momentum analysis

The main aim of my stay here at DESY was to develop a Matlab code to analyze the mean momentum and the momentum spread of the electron beam after the photocathode. The software is based on a code developed at PITZ [1]. As we said before, the beam is deflected by a dipole magnet and sent to a screen in the dispersive section. Here the longitudinal momentum distribution is translated into a spacial distribution.

The developed software is structured into four files:

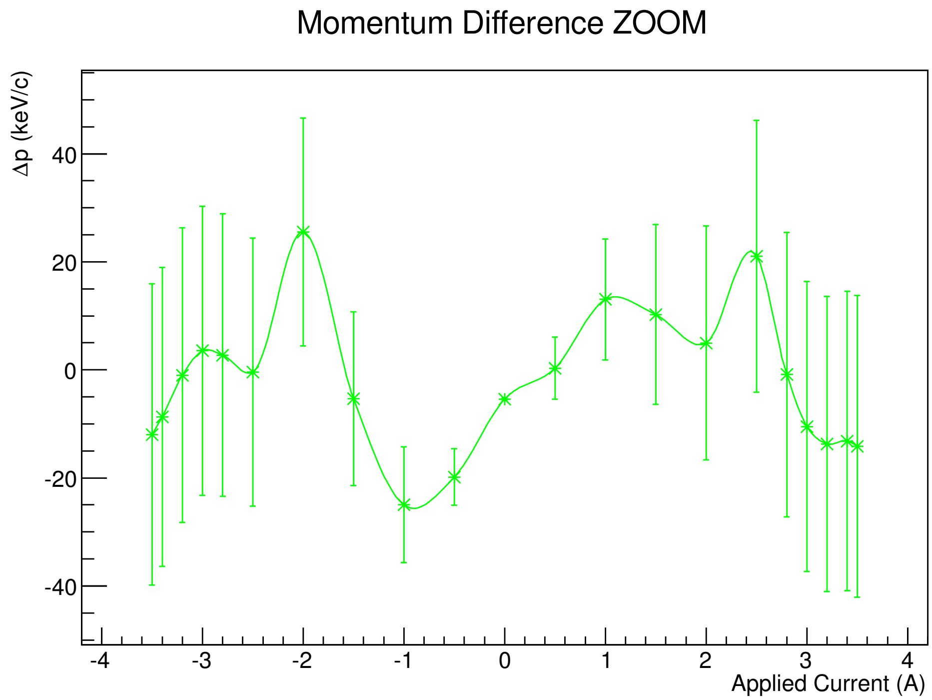


Figure 10: Zoom of the behavior of the best fitting function including error bars

- MAMAmain.m: to runs simple tests of the code
- data.m: containing all the physical properties and methods
- MAMA_GUI.m: containing the graphic interface callbacks
- MAMA_GUI.fig: containing the graphical properties

The basic idea behind the program structure is that the work and the code developed in this two months are only a small part of what would become a much more complex software. For this reason we wanted to create a program that can be easily changed and modified at different times and by different people, without too many efforts. First of all we tried to add as many comments as possible inside the code, on the other hand we tried to separate different functions and features into different files.

The core of the physics is contained into the data.m file. Here we have a class with several methods and properties that we have to call to make the physics analysis. Basically all the functions to compute the mean momentum, to make calibrations, to compute errors etc. are contained here. The MAMAmain.m file contains a simple Matlab script that includes the data class and calls data methods to perform a simple analysis. This file does not offer a graphical interface and all the parameters, such as the calibrations have to be set manually as input arguments of the fitting functions.

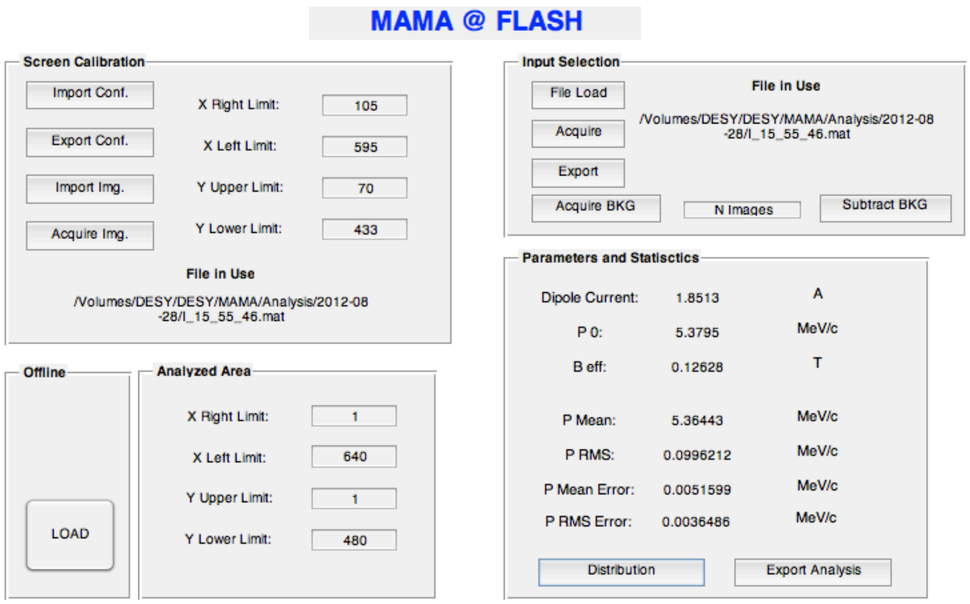


Figure 11: MAMA graphic interface

This is not a good solution for future operation but it is a great tool for developers, since it allows to test the data methods separately from the graphical interface.

The MAMA_GUI.m is one of the two files containing the information for the graphical interface. All the layout information are directly stored into a figure (MAMA_GUI.fig) that is directly opened by the .m file when the program is running and it can be separately be opened in GUIDE by future developers to make further changes. The .m file contains, on the other hand, all the callback, which are the function used to define the action of each button or knob. This callbacks basically recalls one or more methods of the data class performing all the physics analysis required by the specific button.

The great bonus of such a graphic interface is not just the simplicity of figures instead of a code but also the fact that the user can change settings such as calibrations, cropped areas and acquired image online and on the go, with no need to relaunch the program if a setting has to be changed. Every relevant parameters can be reset and the analysis is immediately updated.

A picture of the graphic interface is shown in figure 11.

3.1 The MAMA structure

The first step of the analysis is to acquire an image of the screen and to make a calibration between the screen and the pixels. Basically what we have to do is to understand how the screen is centered in the image and to understand the relation between the position (mm) and pixels. The length and width of the screen are known parameters and the user can define on the figure the corresponding pixel of the edges

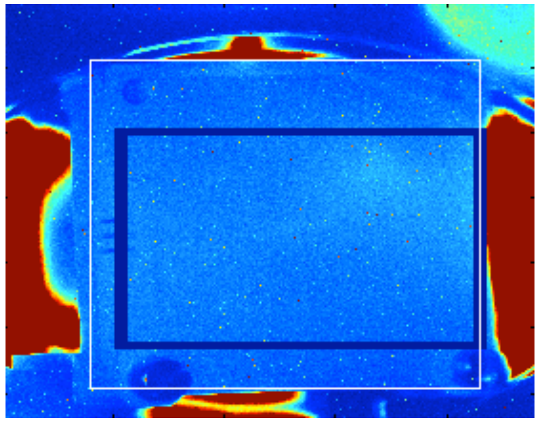


Figure 12: Screen acquisition

of the screen (fig. 12). When the program is started the last image of the screen used and the respective calibration are loaded. A solid line rectangle is impressed over the picture and it should fit exactly into the screen. If the rectangle does not fit the screen the user can recalibrate the screen size by specifying different pixels values. It is obviously fundamental that the rectangle in not bigger nor smaller than the screen. Once the calibration is completed the user can export the configuration so that it can be used when the program is started the next time. The configuration and the image file path are stored both into a current configuration file and into a folder with an archive of all previous configurations.

The user can also load a new image from file or simply acquire a new image of the screen pressing on the dedicated buttons.

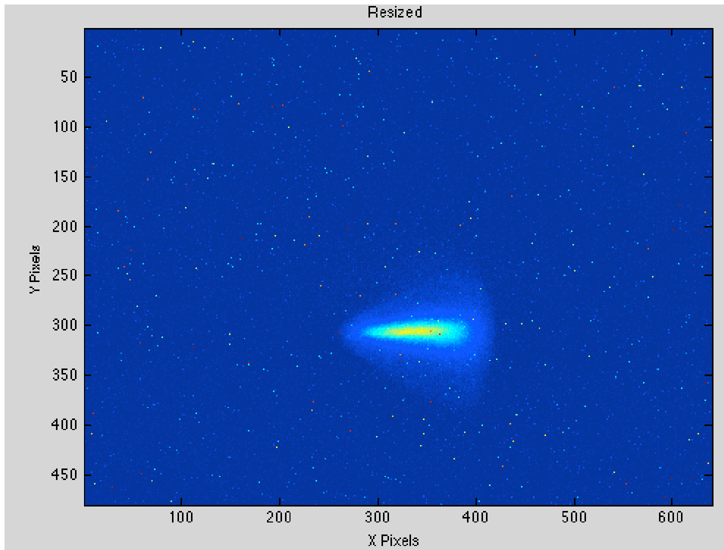
Once the calibration of the screen is completed the user can load an image of the beam from a file or acquire a new beam image, pressing on the proper button. At this point the physics analysis begins.

The dispersion of the dipole magnet acts only in one direction of the screen, while in the other one the spot size is defined by the beam size and by the transverse momentum. For this reason a projection of the beam in the direction perpendicular to the dispersion is performed. Let's suppose that we are acquiring an image as a matrix of counts of each pixel H_{ij}, where i is the index in x direction and j is the index in y direction. If the dispersion of the dipole magnet is performed along the x direction we can add up the contribution in y direction $h_i = \sum_j H_{ij}$.

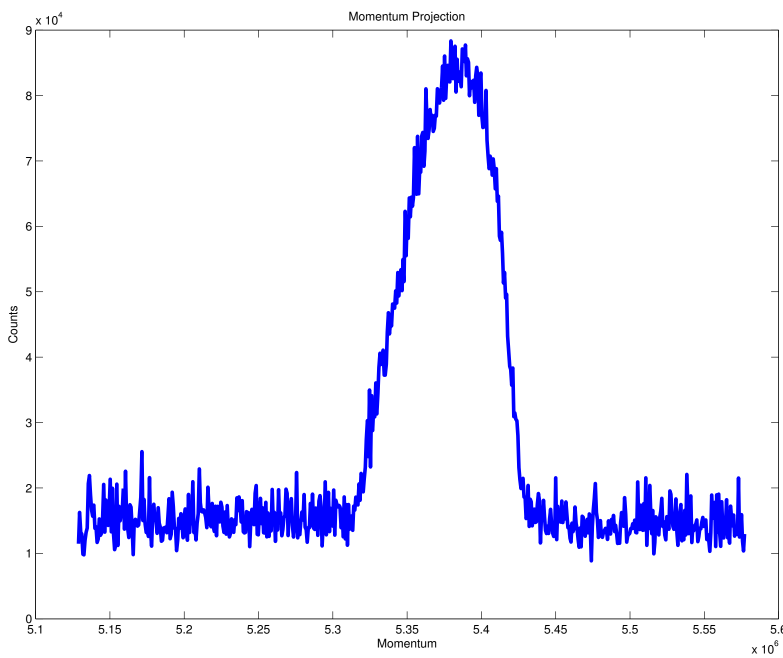
The user has the possibility to resize the area of interest such that if the beam occupies only a small region of the screen. The projection can be made only using the chosen part of the screen, reducing the contribution of noise.

In figure 13 it is possible to see an image of the beam on the screen and also the momentum distribution deriving from that image.

Once the projection is done, the program automatically computes the mean mo-



(a) Image of the beam on the screen



(b) Momentum distribution

Figure 13: Image of the beam (a) and related momentum projection (b)

mentum, momentum spread and its errors.

First of all it is necessary to compute the magnetic field generated by the dipole: the programs loads the fitting parameters, saved into a file named MAMA_FitPar.txt and evaluates the four parameter exponential at the required current value.

Then the momentum in the central position of the screen is computed, using formula 1. The momentum value in another point of the screen and particularly in another point of the projection, which is different from the center is computed with the formula [1]:

$$p = p_0 \left(1 + \frac{x_d}{D(s_D)}\right)$$

where x_d represents the coordinate on the screen computed from the center and $D\,(s_D)$ is the dispersive function. This equation is valid for point-like beams, without divergence. Therefore the transverse beam size and divergence before the dipole results into a smearing of the momentum distribution. The smearing function is generally unknown, so this results into a systematic uncertainty which has to be simulated.

The dispersive function describes the relation between the deflection of different particle and their energy. It's expressed by the formula [2]:

$$D = \left(\frac{l_{eff}}{\alpha} + l_2 \tan\beta_2\right)(1 - cos\alpha) + l_2 \sin\alpha$$

where β_2 is the exit angle difference to 90°. The length l_2 is defined as [2]:

$$l_2 = l_{20} - \frac{l_{eff} - l_{geom}}{2} \pm y_D$$

where l_{20} is the distance between the dipole exit and the screen in the dispersive arm and l_{geom} is the geometrical length of the electron trajectory within the dipole shoe.

At the FLASH IDUMP-dipole magnet they are:

$$l_{20} = 473.24\,mm$$

and:

$$l_{geom} = 470\,mm$$

From the projection of the image on the screen we can compute the mean momentum:

$$\bar{p} = \frac{\sum_i p_i h_i}{\sum_i h_i}$$

where p_i is the momentum computed at the position x_d of the $i{-}pixel$ of the projection[2]. h_i represents the total count of the $i - pixel$ of the projection and acts as a weight for the mean.

[2]For this reason it is fundamental to establish a precise relation between positions and pixels, with an accurate calibration of the screen.

The program also computes the momentum spread:

$$p_{RMS} = \sqrt{\frac{\sum_i (\bar{p} - p_i)^2 h_i}{\sum_i h_i}}$$

Once one has computed the mean value and the momentum spread, it's necessary to understand the uncertainty associated with that measurement.

Assuming that all single errors are the same and assuming that they are equal to p_{RMS} it results that :

$$\sigma_{\bar{p}} = p_{RMS} \frac{\sqrt{\sum_i h_i^2}}{\sum_i h_i}$$

and:

$$\sigma p_{RMS} = \frac{\Delta \bar{p}}{\sqrt{2}}$$

All this results are displayed on the graphical interface and can be exported into a Matlab file in the form of Matlab structures. This allows to create, in future, arrays of this structures to make, for examples, studies on the evolution of momentum as a function of time or current.

It's also possible to acquire several images without the beam to create a background pattern and subtract it to the acquired beam image.

3.2 Further Improvements

Some further improvements could be achieve to extend the operability and the performance of the MAMA software. Here is a list of suggested future improvements:

- Saturation control: it could be useful to monitor the status of saturated pixels and to specify a maximum number of allowed pixels in saturation. To have pixels in saturation means that the average mean is performed with wrong weights.
- Scan: it may happen that the beam doesn't fit into the screen. One could think to develop a routine that performs an analysis by taking into account more than one image by scanning the dipole current.
- Current and Screen control: The program could be able to move the screen and to set the dipole current directly, without requiring the usage of other control interfaces.

4 Conclusions

The knowledge of the beam status in terms of momentum and momentum spread, charge and emittance is fundamental to obtain a shorter saturation length and to reach the SASE (Self Amplified Spontaneous Emission) conditions. For this reason it is important to develop a code that fast and easily allows physicists to check the beam status. The code developed in Matlab tries to be a good interface for this analysis: on the one hands it provides all the tools to complete a simple analysis, on the other hand it can be easily modified or integrated in more complex Matlab softwares.

Further improvements to the code are, anyway, recommended to make it complete and reliable, also in much more complex operation conditions.

5 Acknowledgments

I'd like to thanks DESY staff and particularly Olaf Behnke for organizing the summer student program and for offering us such a great opportunity.

Many thanks to my supervisor, Juliane Ronsch Schulenburg who guided me through the project and gave me new inspirations and ideas for the future studies.

Thanks to Dirk Lipka, Sven Ackermann and Christoph Lechner for their help with my code.

References

[1] D. Lipka. Momentumand momentumspread analysis. *PITZ Note*.

[2] D. Lipka. Spectrometer dipole for gun at flash. *DESY Note*.

[3] Josef Feldhaus Katja Honkavaara Siegfried Schreiber Rolf Treusch Mathias Vogt, Bart Faatz. Status of the free electron laser flash at desy. *Proceedings of IPAC2011*.

[4] J. Ronsch. Investigations on the electron bunch distribution in the longitudinal phase space at a laser driven rf electron source for the european x-fel. *PhD. Dissertation*.

[5] DESY Staff. Flash brochure.

Appendix I

The main purpose of this appendix is to give further details on the code we used to make the analysis of the dipole magnetic field, describing the general structure and the role of each function.

The program to analyze the magnetic field is divided into three files: a main file, named Hysteresis_Plotter.C; a file containing the data and the methods to acquire them from external files, named Take.C and finally a file containing a class to make the analysis of the data, named Analysis.C.

The users should not care about the constitution of the last two and can perform different analysis simply calling the methods we are going to describe from the Hysteresis_Plotter file. The folder also comes with a makefile: calling make Hysteresis_Plotter from a shell the program will be directly compiled and linked to the required ROOT libraries.

To avoid double declaration all the required root libraries are inclosed inside a precompiler if condition: if a library is added to one file has to be added also to the oder two file.

Take.h

- Take: the default constructor builds up a class acquiring the data from a file called infile and storing the current and the magnetic field inside two arrays.
- SingleBrach: this function plots the measured magnetic field as a function of the applied dipole current. In each file only one of the two branches of the hysteresis cycle is contained, so only one branch is plotted with this method.

Analysis.h

- Analysis: the default constructor simply reads the number of tests on the magnetic field response. Pay attention: each test is composed by two files.
- BranchPlots: these function plot the responses of the dipole magnet in each test for each branch of the cycle and on the same canvas. In other words it returns a plot of the hysteresis cycle of each test.
- MeanAnalyzer: this function computes the mean value of the magnetic field for each test for each branch of the cycle.
- ErrorAnalyzer: this function computes the standard deviation of the data from the different tests.
- MeanPlot: this function plots the trend of the hysteresis cycle. It accepts a string as input. This can have the following values: empty, Poly3, Poly6, Exp3, Exp4.

Depending on the value of the string the function will fit the data with different curves (see paragraph 2.1) and will plot the fit on the same canvas. The fitting function is stored inside a class variable, in oder to make it available to other methods.

- DifferencePlot: This function evaluates the fitting function in a point of the current array and computes the difference between the real value and the fitted value. It accepts a string as input (Momentum or Field) and makes a plot of these differences as a function of the current: it can plot the difference of momentum or of the magnetic field. The difference is evaluated in the upward direction of the hysteresis cycle.

- ZoomDifferencePlot: this function makes zoom on the plot before, giving a detailed view over the difference between the measured momentum and the one fitted with the best fitting function, which, in that specific case, is represented by the four parameters exponential. It also plots some errors, taking into accounts both the uncertainties on the measurement and the uncertainty of the fitting parameters. For further details on the error computation, please refer to paragraph 2.1.

- BtoP: this function computes a conversion factor from field to momentum. Please refer to formula 1 for further details.

- SGMCalculator: This function computes the error associated with the evaluation of the momentum from the fitting function.

- ParFit: this function creates the .txt file with the fit parameters used by the MAMA code.

Appendix II

In this appendix we will give a more detailed description of the graphic interface (fig. 14), so that the user can better understand how to proceed in the analysis of the longitudinal beam status.

Screen Operations

- The program can be started opening the file MAMA_GUI.m or the file MAMA_GUI.fig and pressing on the Matlab run button.

- Meanwhile the operator can move in the screen and turn on the light to illuminate the screen.

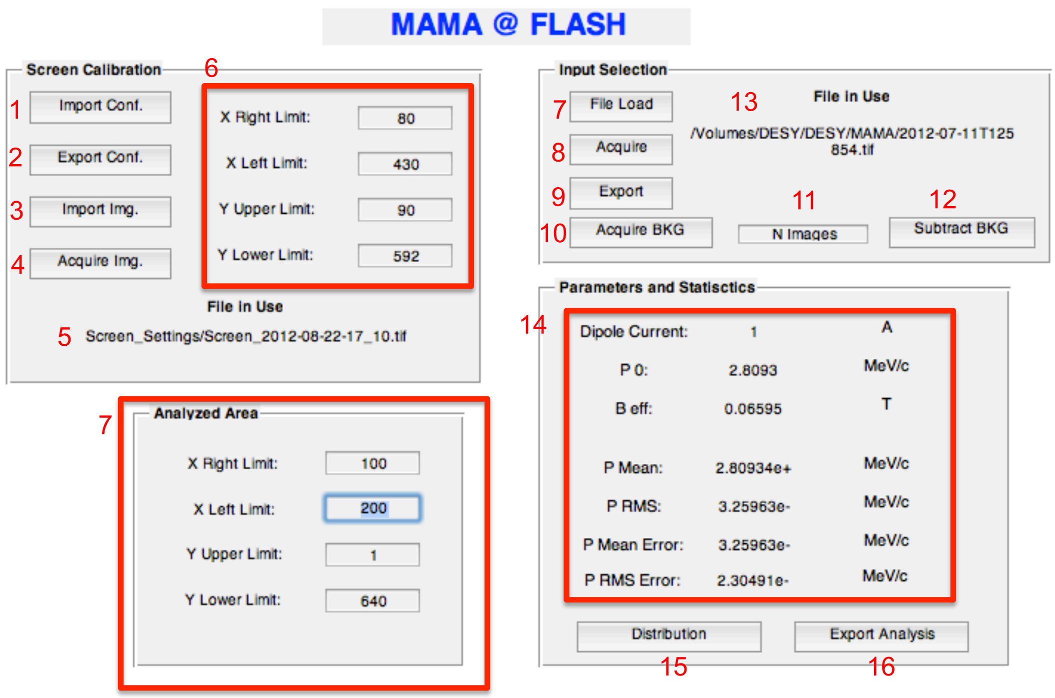


Figure 14: MAMA GUI

- The screen can be acquired pressing button 4. The screen position can be determined changing the values of parameters in box 6, so that the screen exactly fits into the solid line rectangle. It is important that the fitting is correct, since this calibration is used to compute the momentum.
- The light can be switched off.
- The user can export the acquired image and the calibration settings pressing the button 2. These will be saved in a folder called Screen_Settings and in a file called Screen_Setting_Current. When the program is re-launched the last configuration (picture and calibration), saved in the file Screen_Setting_Current will be automatically reloaded.
- The user can import an old configuration with the button 1 or import just an old image (tif format, monochromatic) pressing button 3.
- From panel 5 the user can see if he is using an ONLINE image or a saved image (the image path is displayed)

Image Operations

- The user can load an old image from a file with button 7 or acquire a new image from the camera with button 8.
- The acquired images can be exported with button 9. In panel 13 the user can check which image is he working on.
- The user has also the possibility to perform a background reduction independent from the one embedded into the camera software. He can insert the number of images he wants to sample for the background reduction in box 11. Then he can acquire the background pressing button 10. After that it is possible to turn on and off the background reduction pressing toggle button 12. Every time the image is changed or a background setting is changed the analysis results are refreshed.
- From panel 7 the user can resize the area where the momentum computation is performed. This can give precise results, getting rid of more background noise. Every time the analyzed area is changed the analysis results are refreshed.

Analysis Operations

- On panel 14 the user can see the results of the analysis.
- He can also have a close up view of the momentum distribution pressing button 15
- It's possible to Export all the data, including pictures, settings, paths etc by pressing button 16. The exported data are saved into a Matlab.mat file, and particularly as members of a structure. This allows further analysis, for example creating a vector of structures and studying the behavior of the momentum distribution as a function of the dipole current. The exported files are saved in a folder called Analysis, grouped by day in sub folder and named according to the day and time.
- The user can upload an old data-set from the analysis folder pressing the button load, in the section offline (fig. 15).

Figure 15: Button to load offline analysis