Laser & HHG Diagnostic for sFLASH

Cagil Cinar

Istanbul University, Istanbul - Turkey

Supervisor: Manuel Mittenzwey

D E S Y
N O T K E S T R A S S E 8 5
2 2 6 0 7 H A M B U R G , G E R M A N Y

INDEX

ABSTRACT

This project implements the development of new tools for High Harmonic Generation (HHG) and driving laser diagnostic for the seeding project at the Free electron Laser in Hamburg (sFLASH). One of those tools is an Ultra high vacuum (UHV) mirror switch between near and far field diagnostic for the HHG. The second one is a feedback loop for the near and far field of the 800nm driving laser.

A seeding free-electron laser (FEL) experiment at XUV wavelengths, known as "sFLASH", is being prepared at the existing SASE FEL user facility FLASH. Beyond a proof-of-principle demonstration in the XUV, the emphasis will be on high stability in terms of intensity and timing. Seed pulses at wavelengths around 30 nm from high harmonic generation (HHG) of an 800nm driving laser will interact with the electron beam in the newly installed undulators upstream of the existing SASE undulator section. The seeded FEL radiation will be directed into a dedicated photon beamline for timing experiments.

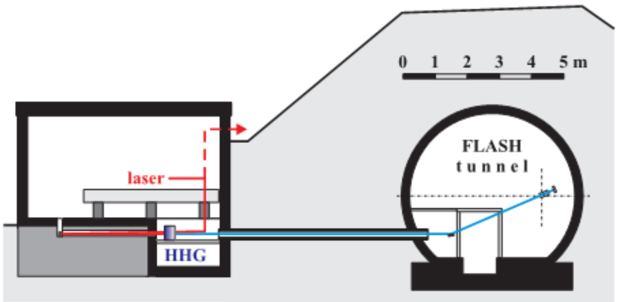

Figure 1: Cross section of the FLASH tunnel and the adjacent laser laboratory. Under the laboratory floor, the HHG source and the incident laser beam (red) are aligned in a tube, through which the HHG pulses (blue) enter the tunnel.

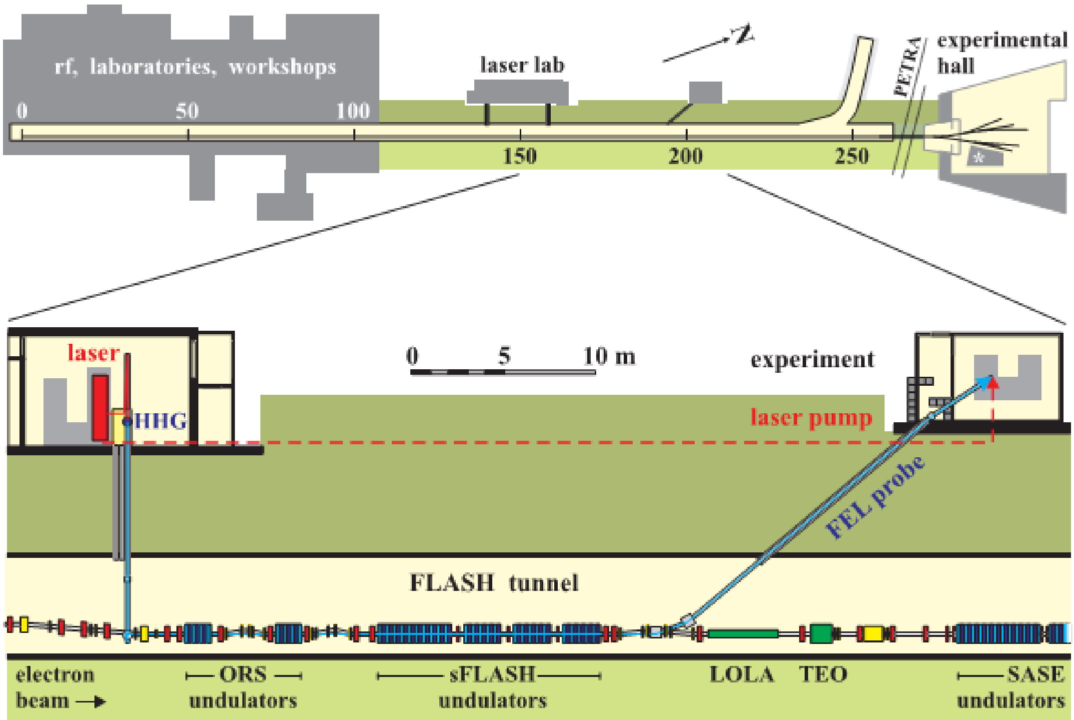

Figure 2: FLASH Facility (top) with sFLASH section (bottom)

Changing a plane and a focusing mirror in the HHG diagnostic section in order to switch between near and far field, respectively, needs venting and adjusting of the setup. This is very time consuming. Therefore, a mirror switch (Figure 2) has been designed to make this application easier.

The challenge was to design a compact switch because we have very limited area. Therefore we changed the motor position to the back side and set the switch on the translation stage in an unusual way. (See Figure 3)

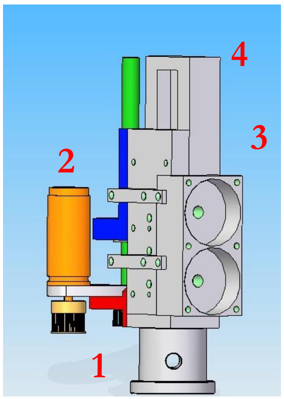

Figure 3: Switch on a translation stage.1: 25mm post, 2: Motor, 3: Mirror holder, 4: Translation stage

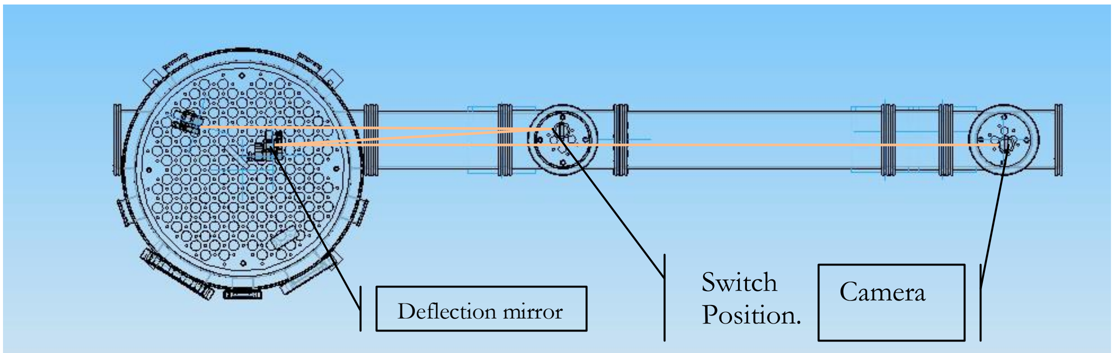

Figure 4: Diagnostic section of the HHG. the Switch position marked in the figure. Orange line: HHG path.

The lack of space in the HHG diagnostic UHV chamber is shown in Figure 4 and 5. The dimensions are 53 - 27 - 18 mm³. The HHG height is about 75 mm above the ground of the chamber floor. Therefore, a 25 mm post is used to elevate the translation stage (See Fig. 3). Thus, the upper mirror is on the right beam height and the second mirror can be selected and aligned by the translation stage.

All components of the switch are cleaned for UHV application. The motor screw is lubricated with UHV grease and the electric feed through for the motor was sealed. Then, the vacuum seal has been checked with a helium leak tester.

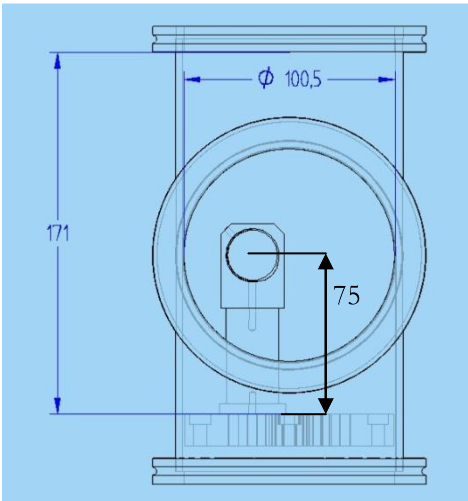

Figure 5: Dimensions of the UHV chamber where the mirror switch will be installed.

At the end, the complete setup was successfully tested in air and is ready to be installed in the UHV Chamber

SOFTWARE

The design of the switch is made with Solid Edge, a 3D CAD modeling software. It provides solid modeling, assembly modeling, and drafting functionality. Through third party applications it has links to many other Product Lifecycle Management (PLM) technologies. It is originally developed and released by Intergraph in 1996 using the ACIS geometric modeling kernel and is later changed to a Parasolid kernel. From 1998 it was purchased and further developed by the UGS Corp (the purchase date correspond to the kernel swap).[1]

[1] Dean, Al (2006.09.14). "Solid Edge V19 Review". MCAD Magazine.
Dean, Al (2006.06.18). "Solid Edge V20". MCAD Magazine.
Henry, Michael (1998.09). "A tale of two CAD systems". Mechanical Engineering Magazine.

The driving laser of the HHG source is an 800nm NIR laser. It is manually aligned with a far and near field camera In order to compensate drifts of the 800nm laser, a feedback system is needed. The developed program reads the near and far field cameras and moves the motorized mirrors (New Focus) to fix the laser path.

FRAMEWORK

.net Framework 2.0 is used for this project and AForge.net framework is used for image processing. The program uses a minimal telnet connection class for the New Focus Ethernet controller.

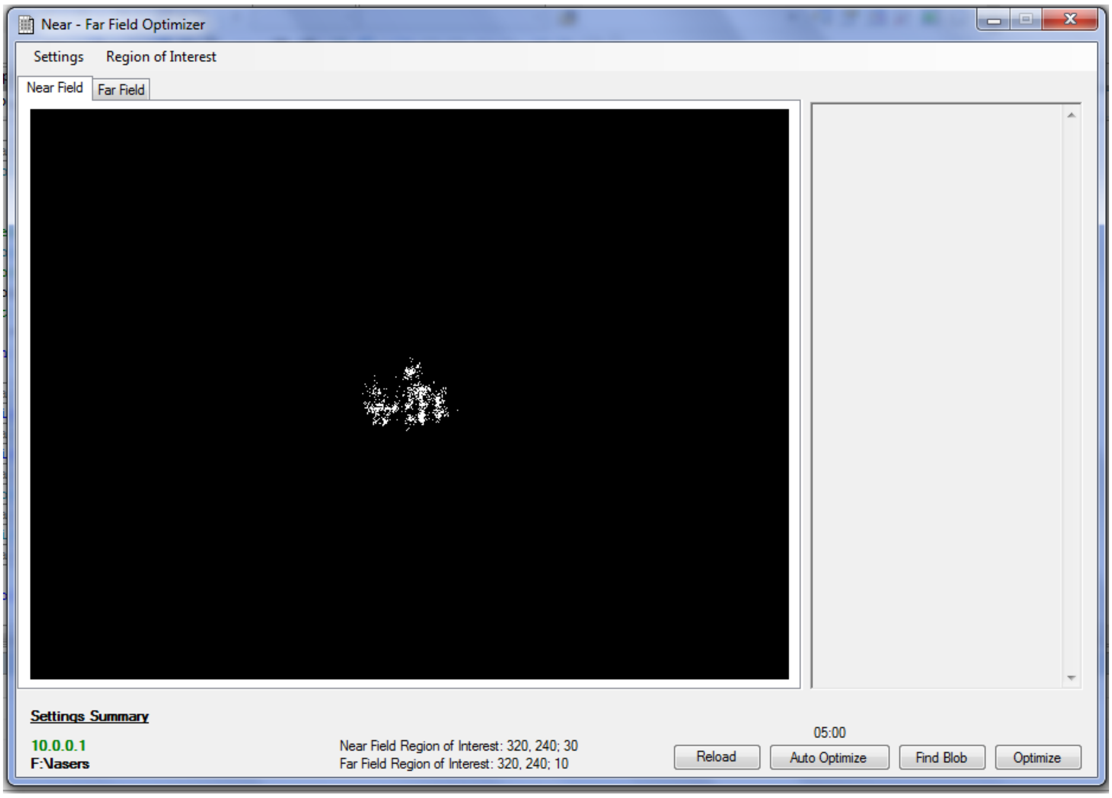

Figure 6: General state of the front panel.

AMS.profile.dll is used for saving the settings. This library saves settings in several ways like XML, Registry, Config file. This program saves the setting via XML (settings.xml)

HOW IT WORKS

The near and far field cameras are connected with a Unix server. A TCP/IP image client/server has been installed on both, Unix server and Windows PC. The client program installed on the Unix system pushes the images to the server. The server program installed on the Windows PC with CygWin[2] has a listening port to capture the images. After the program receives the images, it applies a threshold filter to binarize the image. With AForge.net framework's BlobFinder function, it constructs a circle on the image, and with corresponding analysis, it determines the center of the laser spot.

According to the center of the spots, the program calculates the distances from the set point and moves the motorized mirrors.

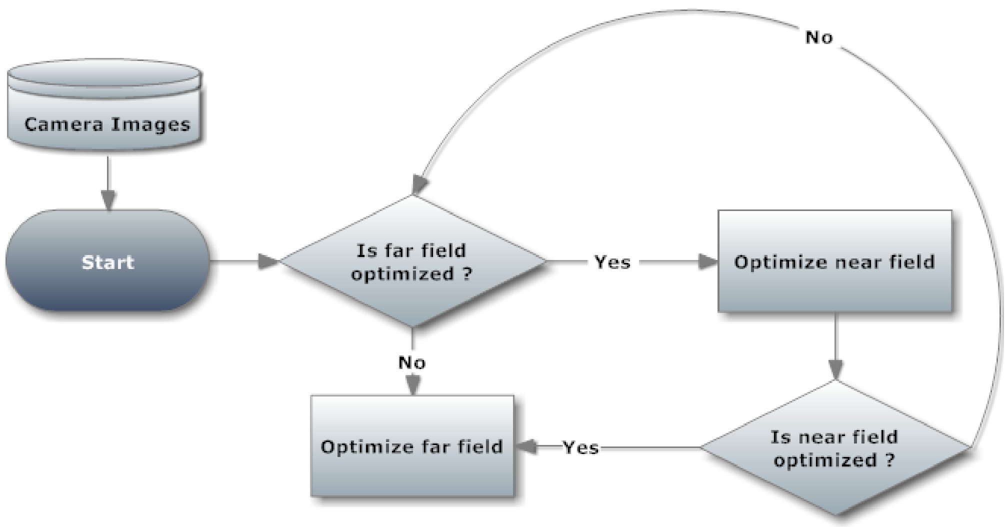

Figure 7: Flow chart of the program.

The general algorithm of this program: it starts from one field (near or far field), optimizes until it fits to the region of interest and starts to optimize the other field. This is an infinite feedback loop and it will optimize both fields until it is closed. (See Fig. 7)

FUNCTIONS

- **Set IP Address**: IP address of New Focus Ethernet Controller

[2] An emulator of Unix platform.

- **Motor Settings**: There are two motors for each mirror. This function set the motors' driver and motor numbers.

- **Step Settings**: Approximation of step number that moves the laser spot 1px from its location. This setting has two values, one for each field.

- **Laser Images**: This sets the directory of the laser images.

- **Region of Interest**: This sets the region of interest. X, Y, and radius for each field.

CODES

```
        /// <summary>
        /// Find Circles on the image.
        /// </summary>
        /// <param name="image">An image that must be <c>Bitmap</c></param>
        /// <returns></returns>
        public Blob[] FindBlob(Bitmap image)
        /// <summary>
        /// Moves specified motor in the direction and amount of steps.
        /// </summary>
        /// <param name="driver">Select the driver called A1, A2 ...</param>
        /// <param name="motor">Select the motor 0,1,2</param>
        /// <param name="steps">Select steps with direction !</param>
        /// <example>
        /// try{ MoveMotor(1,2,-300000)}
        /// catch(Exception ex){MessageBox.Show(ex.Message)}</example>
        public void MoveMotor(int driver, int motor, int steps)
        /// <summary>
        /// Moves 2 motors at the same time. Must be seperate drivers !
        /// </summary>
        public void MoveMotor(int driver1, int motor1, int step1, int
driver2, int motor2, int step2)
        /// <summary>
        /// Returns the blob that higher than 20px. If there is none it
returns first one.
        /// </summary>
        /// <param name="blobs"></param>
        /// <returns></returns>
        public Blob BigBlob(Blob[] blobs)
        /// <summary>
        /// Returns the center of circle...
        /// </summary>
        /// <param name="blob"></param>
        /// <returns></returns>
        public Point Center(Blob blob)
        /// <summary>
        /// Moves the motors (X, Y) according to <c>ApprStep</c> number.
        /// It can moves 2 motors at the same time if necessary.
        /// </summary>
        /// <param name="center">Point of center of circle</param>
        public bool OptimizeFieldNear(Point center)

        /// <summary>
        /// Moves the motors (X, Y) according to <c>ApprStep</c> number.
        /// It can moves 2 motors at the same time if necessary.
        /// </summary>
        /// <param name="center">Point of center of circle</param>
        public bool OptimizeFieldFar(Point center)
```

CONCLUSIONS

With this two project, optimizing the laser system becomes more sufficient and easy. The program is very important to stabilize the laser state without human power.

I have to thank the people who helped me with these projects in a limited time. Without them, I could not do anything. I have to thank Manuel Mittenzwey, Theophilos Maltezopoulos, Oliver Becker, and all workshop staff who helped me to find solutions quickly.