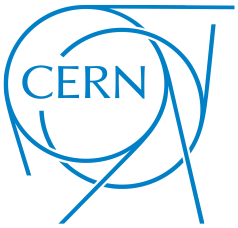
SAPIENZA
UNIVERSITÀ DI ROMA



Dipartimento di Ingegneria Astronautica, Elettrica ed Energetica
PhD in Engineering and Applied Science for Energy and Industry

Characterization, Control and Alignment of Crystal Collimators of the High Luminosity Large Hadron Collider (HL-LHC)

Supervisors:

Prof. **Andrea Mostacci**

Dr. **Eloise Matheson**

Dr. **Enrica Soria**

Dr. **Mario Di Castro**

Dr. **Stefano Redaelli**

Student:

Gianmarco Ricci

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy.

Academic Year 2024/2025

Contents

List of Figures

List of Tables

Abstract

The Large Hadron Collider (LHC), the world's most powerful particle accelerator, is undergoing a significant upgrade to increase both its luminosity and operational lifespan. The LHC upgrade project is referred to as the High-Luminosity Large Hadron Collider (HL-LHC). A central aspect of this upgrade is the implementation of crystal collimation technology, which enhances beam loss management by using bent crystals to improve collimation efficiency for heavy-ion beams. This work aims to advance the characterization, alignment, operational supervision, and control of crystal collimators for the HL-LHC by developing novel software, algorithms, and machine learning models to streamline these processes.

To address these challenges, an innovative X-ray diffractometer was designed to simplify and automate the traditionally complex task of crystal characterization. Advanced programming techniques were employed to control the diffractometer's mechatronic components and X-ray beam, while algorithms were developed for post-processing and automation of measurements and setup. Furthermore, cutting-edge convolutional neural networks and real-time monitoring systems were designed to optimize crystal alignment and supervision during LHC operations, ensuring more efficient setup and beam collimation. This thesis also introduces a new test bench for the evaluation of a piezo-goniometer controller algorithm for crystal collimators.

The overall goal of this work is to provide efficient solutions for automating particle accelerator setup and improving the performance of the LHC collimation system through advanced technologies. The findings presented in this thesis offer valuable insights into future research directions for crystal collimation systems and their integration with modern machine learning techniques.

Chapter 1

Beam collimation in high beam power particle colliders

The Large Hadron Collider (LHC) is the world's largest and most powerful particle collider. Constructed by the European Organization for Nuclear Research (CERN), it involves collaboration from over 8,500 scientists, hundreds of universities and laboratories, and more than 100 countries. Situated in a 27-kilometer circular tunnel beneath the France-Switzerland border near Geneva, the LHC accelerates two high-energy, counter-rotating protons or heavy ions beams of particles and brings them into collision [35, 36]. To extend its operational lifespan, the LHC is undergoing a major upgrade known as Hi-Lumi LHC. This aims to increase both its luminosity (the rate of collision events) and its integrated luminosity (the total number of collision events produced) [37]. A crucial aspect of this upgrade is enhancing the existing collimation system, which manages beam losses within the accelerator, safely and effectively.

The importance of beam collimation systems has grown significantly in modern particle accelerators as they seek to achieve higher beam energies and intensities. The CERN LHC has achieved a stored energy (total energy contained within the particle beams circulating in the accelerator) of 430MJ at 6.8TeV. The LHC's high-luminosity upgrade (HL-LHC) has a goal of 700 MJ. These stored energies are more than a billion times larger

than the necessary energy to quench[1] superconductive magnets. For this reason, effective beam collimation [38] is essential for controlling inevitable losses during standard beam operation in high-power accelerators. The complexity and performance of beam collimation in the LHC have reached unprecedented levels, allowing for efficient "cleaning" of beam losses before they reach the small apertures of superconducting magnets [1, 39].

1.1 Purposes of the collimation system and its layout

As formulated in [1], particles with transverse amplitudes or energy deviations significantly larger than those of the reference particle are referred to as *beam halo particles*. In particle beams, the transverse positions of the particles (relative to the beam axis) often are approximated as a Gaussian (normal) distribution, where the majority of particles are clustered around the mean, with fewer particles further away from the center. Halo particles are located in the tails of the distribution—far from the central region where most particles are (Figure 1.1). Halo particles may arise due to intra-beam scattering, which involves multiple scatterings between particles that increases bunch dimensions and energy spread; beam-beam effects, where particles scattered during collisions between bunches populate the halo; or residual gas scattering, where particles scatter off residual gas molecules in the beam pipe. Beam collimation is achieved by placing collimator jaws, which are blocks of material (black boxes in Figure 1.1), close to the circulating beams to constrain particles that stray outside the core.

Particles in the beam halo can cause a magnet to transition from its superconducting state, which can damage the equipment. Therefore, it is crucial to dispose of these losses with a robust collimation system.

The primary functions of the collimation system include cleaning betatron and off-momentum beam halos, which are formed when particles at the edges of accelerated bunches follow wider orbits than the core, potentially causing undesired beam losses. These halos must be intercepted and safely disposed of before they can damage sensitive equipment. Additionally, the system provides passive machine protection by positioning collimators closest to the circulating beam, serving as the first line of protection during normal or abnormal beam loss events.

[1] The transition of a magnet from a superconducting state to a resistive state.

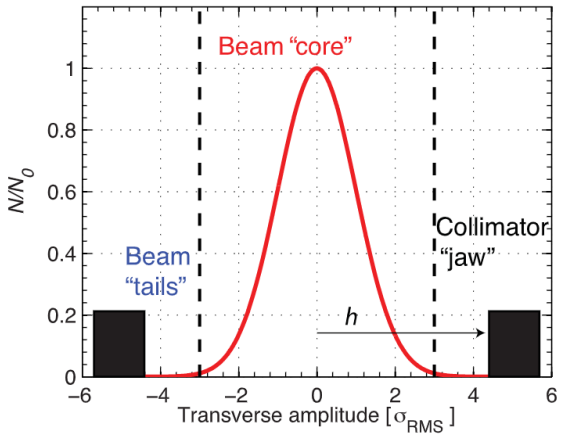


Figure 1.1: Gaussian distribution, which is typically adequate to model the particle distribution of the beam core (red line). Overpopulated tails may be intercepted by collimator jaws, which constrain particle motion at a given transverse amplitude, from [1].

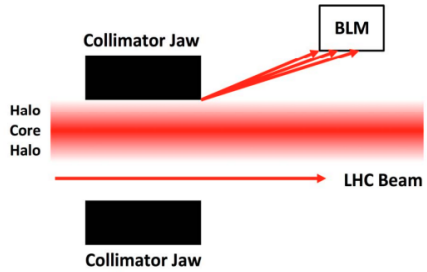


Figure 1.2: Scraping of the beam halo when the collimator jaw is aligned to the beam. The scattered secondary particles are detected by a beam loss monitor (BLM) downstream and result in a time-varying signal in units of Gy/s, from [2].

The collimation system also cleans up collision products by using movable collimators placed in the outgoing beam paths of experiments to capture post-collision particles. Another crucial role is optimizing the experimental background by minimizing halo-induced noise in detector measurements, achieved through beam tail scraping or local shielding at detector sites. This helps reduce spurious signals in detectors.

The system further provides local protection for equipment, enhancing its longevity by shielding it from radiation effects with dedicated collimators. Finally, the system facilitates beam halo scraping and diagnostics. While not always a primary design criterion, the ability to scan and shape the beam distribution is a valuable feature. The LHC's robust primary collimators efficiently perform this scraping and shaping function [1].

The collimation layout of the LHC (Figure 1.4) is composed of 110 mov-

able devices. The collimation system comprises various types of collimators, each with a specific function, as illustrated in Figure 1.3. Primary collimators (TCP) are designed to intercept the primary halo particles, and they are constructed from carbon fiber composite (CFC) to withstand the highest energy dissipation. Secondary collimators (TCSG) follow the primary stage, while absorber collimators (TCLA), made of tungsten alloy, are responsible for absorbing hadronic showers generated in the first two stages of collimation. According to [1], these absorber collimators are less robust than the CFC collimators and should not intercept primary beam losses, with a setting hierarchy in place to ensure this. Tertiary collimators (TCT) capture the tertiary halo, and the target collimator dump quadrupole (TCDQ) is a single-jaw collimator used to intercept particles during beam dumps.

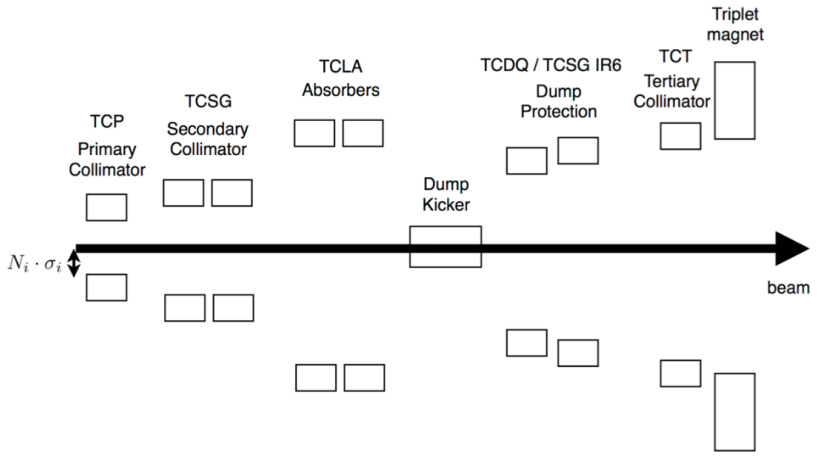


Figure 1.3: Collimator Hierarchy in the LHC, from [3].

1.2 The collimation system upgrade

Despite the lower total energy of stored beam with ions compared to protons, the multi-stage collimation system in the LHC is less effective for ion beams due to nuclear fragmentation processes. These processes generate ion fragments with varying magnetic rigidities that do not receive adequate transverse kicks to be directed towards secondary collimators. As a result, these fragments pass through the betatron cleaning region (Figure 1.4), which is responsible for disposing of transverse beam losses. This increases the risk of magnet quenches. Although the ion-collimation limitation is a well-known concern for the LHC, the standard system has performed adequately for nominal LHC ion-beam parameters. However, the HL-LHC

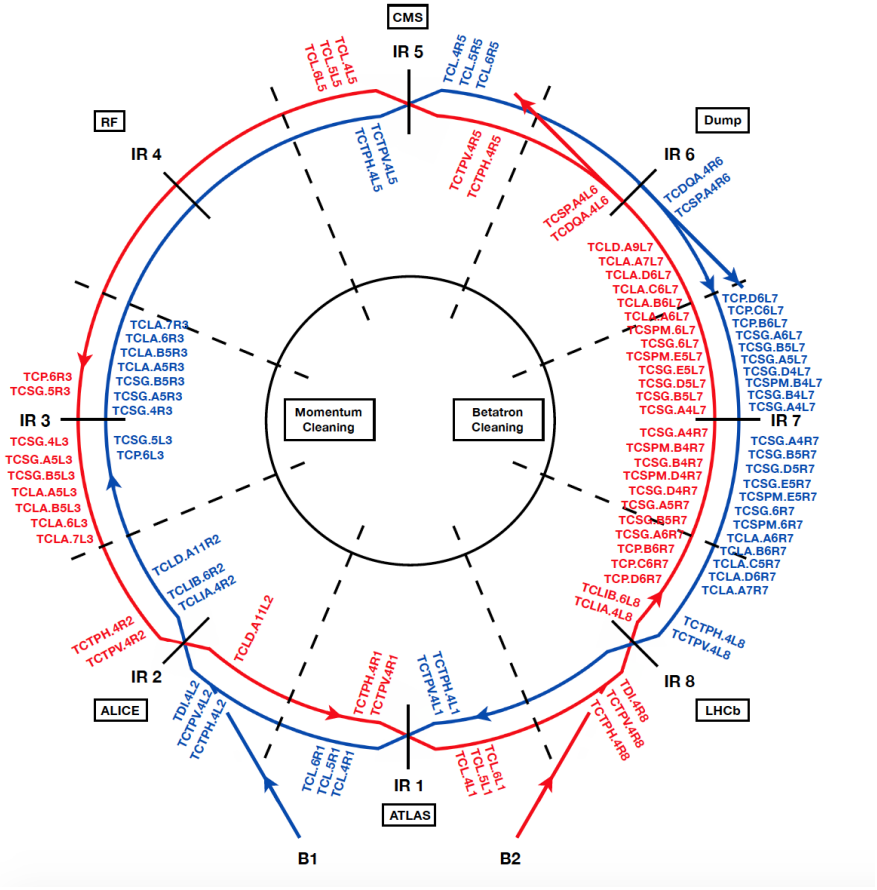


Figure 1.4: Collimator Layout in the LHC at the start of the activity in 2022, from [3].

upgrade presents additional challenges, and improved betatron collimation is necessary to ensure sufficient operational margins [40–42].

One of the upgrades designed to make the LHC collimation system compatible with the target Pb beams is the use of crystal collimation. This concept relies on the use of bent crystals to efficiently steer high-energy particles (Figure 1.5) that become trapped between the electrostatic potential of parallel lattice planes, phenomenon called *planar channeling* [43]. Equivalent bending fields of up to hundreds of Tesla can be achieved in crystals with a length of only 3–4 mm, enabling, in principle, halo particles to be steered to a well-defined point, with obvious potential applications to beam collimation. Particles that stray from the ideal beam orbit are therefore channeled and forced to follow the curvature of the crystalline planes. As opposed to standard primary collimators based on amorphous materials, which require several secondary collimators and absorbers to catch the products developed through the interaction with matter, one single absorber per collimation plane is, in theory, sufficient in a crystal-based collimation

system [10]. The crystal collimation scheme on insertion region 7 (IR7), a section of the LHC accelerator complex specifically designed for beam collimation.

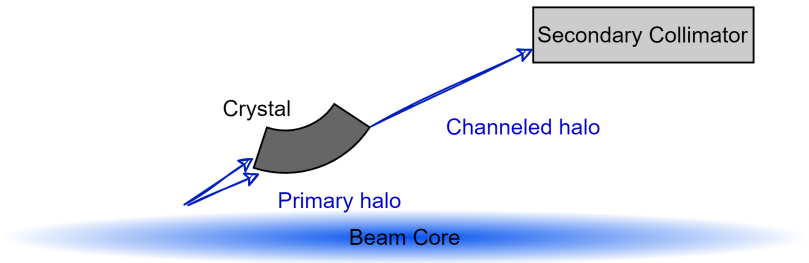


Figure 1.5: Illustrative view of the crystal channeling primary halo particles onto secondary collimator. The optimal channeling orientation can be found by observing losses on the crystal and secondary collimator, from [4].

As detailed in [44], crystal collimation provides two primary benefits:

- The likelihood of inelastic interactions within a crystal collimator is significantly lower compared to a standard collimator. This reduction is due to channeled particles traveling through the relatively empty space between lattice planes, which results in fewer losses from interactions with atomic nuclei.

- Crystal collimation leads to a notable decrease in the impedance budget of particle accelerators. Impedance refers to the collective effects of electromagnetic interactions between the circulating particle beam and the accelerator structure. This is due to the secondary collimator used as an absorber that is positioned further from the circulating beam.

These advantages make crystal collimation an attractive option for beam collimation in high-energy particle colliders [5, 39]. Currently, CERN is at the forefront of developing this innovative collimation technique, with four crystals installed in the machine for operational use in insertion region 7 during ion physics runs.

As formulated in [5], to evaluate the effectiveness of the halo cleaning performance in the LHC collimation system, a key method involves using the Active Transverse Damper (ADT) [45] to deliberately stimulate the beam and induce controlled losses throughout the machine. The signals produced by these losses are captured by the beam loss monitors located around the ring. The data from these signals is then presented in a format known as a

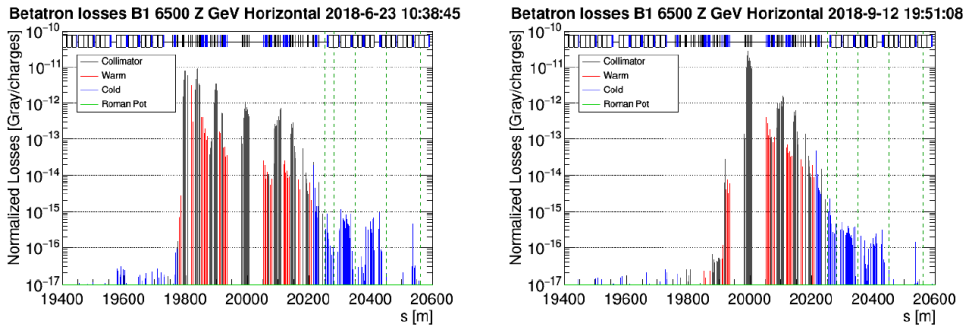


Figure 1.6: The typical IR7 loss patterns for the standard (left) and crystal (right) collimation systems are shown, measured on the horizontal plane of Beam 1 with protons at 6.5 TeV. The losses are normalized to the beam flux, from [5].

loss map. These tools facilitate the comparison between the standard and crystal collimation systems. For instance, Figure 1.6 illustrates normalized loss maps for the horizontal plane of Beam 1. In the left frame, the typical IR7 hierarchy of the standard collimation system is visible, showing the highest loss peak at the primary collimator. Conversely, the right frame displays a distinctly different loss pattern with crystal collimation, where the highest peak is found at the secondary collimator that captures the channeled halo [5]. Thus, a distinct loss pattern becomes evident when the crystal is positioned in its optimal orientation.

1.3 Thesis aim and organization

The objective of this thesis is to improve the performance of crystal collimation, from the initial geometric characterization of crystals to their optimized use in the machine.

The first aim is to enhance crystal characterization that measures the geometric parameters of the devices before being installed in the machine by developing novel software and algorithms for an innovative custom-built X-ray diffractometer. This integration aims to streamline and simplify the traditionally time-consuming process of crystal characterization using X-ray beams. Advanced programming has been utilized to control the diffractometer's mechatronic components, while new algorithms have been developed to allow post-processing and automation.

Another key goal of this thesis is to explore cutting-edge machine learning models for operational deployment and monitoring of crystals during LHC operations. Optimizing these processes is crucial for automating particle accelerator setup and improving the efficiency of the collimation system.

Additionally, the thesis aims to develop a novel test bench for evaluating crystal collimators control for the LHC machine.

The application of advanced technologies and operational tests in the LHC has led to the development of innovative solutions for measuring crystal collimators, automating device setup during beam operations, and efficiently managing these devices.

The theoretical knowledge necessary for understanding the topics covered in this thesis, along with foundational information about the crystal collimation system and crystallography, is provided in the initial chapters.

Specifically, Chapter 1 provided an overview of the LHC's collimation system, while Chapter 2 delves into the principles underlying the crystal collimation system, which plays a central role in the LHC collimation upgrade. Chapter 3 covers the theoretical foundations of X-ray diffraction, including the principles of X-ray beams, diffraction from crystalline materials, and Bragg's law. It also provides an overview of the custom X-ray diffractometer used in this work and its components, along with details on the system's validation and the development of software used to control and automate the diffractometer, including its graphical user interface.

Chapter 4 describes the development and implementation of a new piezo-goniometer controller, using next-generation electronics to manage mechatronic devices at the LHC.

Chapter 5 focuses on the development of innovative convolutional neural networks for optimizing crystal alignment and providing real-time crystal monitoring during LHC operations. Finally, Chapter 6 summarizes the key findings of this thesis and outlines future research directions, emphasizing the characterization, operational supervision, and setup optimization of bent crystals at CERN using advanced deep learning techniques.

Chapter 2

Crystal collimation

2.1 Coherent phenomena in crystals

As reported in [46], a crystal is composed of atoms arranged in a regular, ordered pattern in three dimensions. This arrangement is known as the *crystal structure*. In a single crystal, this order extends throughout the entire volume of the material, while in a poly crystalline material there are many small single-crystal regions (called grains) separated by grain boundaries. A point lattice is an arrangement of points or nodes in three-dimensional space. The point lattice can be partitioned into smaller units, which can be arranged in three dimensions to recreate the original point lattice. A unit cell (Figure 2.1) may be described by the *lattice parameters* i.e. the interrelationship between the lengths (a, b, c) of its sides and the inter-axial angles (α, β, γ) between them. Where, α is the angle between the b and c axes, β is the angle between the a and c axes, and γ is the angle between the a and b axes.

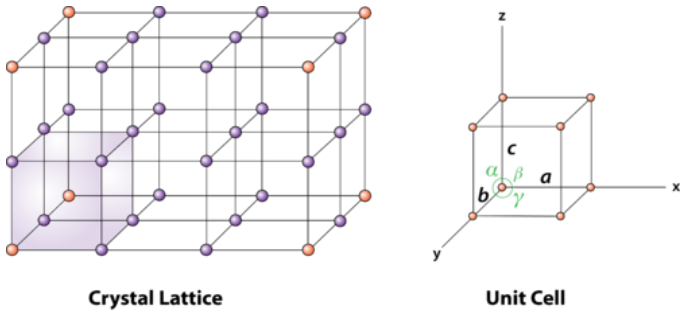


Figure 2.1: Crystal lattice (left) and a unit cell (right) with the inter-axial angles (α, β, γ) between the lattice parameters (a, b, c), from [6]

A crystal plane (Figure 2.2) is a two-dimensional surface within a crystal lattice that is defined by a set of parallel lattice points. In order to describe the orientation of crystal planes in a crystal lattice Miller indices are used. They are defined as the reciprocals of the intercepts of that plane with the three crystallographic axes. These intercepts are multiplied by a common denominator to obtain integers. For example, if a plane intersects the a-axis at 1/2 of the unit cell length, the b-axis at 1/3 of the unit cell length, and the c-axis at 1/4 of the unit cell length, the Miller indices for that plane would be (234) [46]. If a plane is parallel to an axis, the reciprocal intercept is taken to be infinity, so the Miller index for that direction is zero.

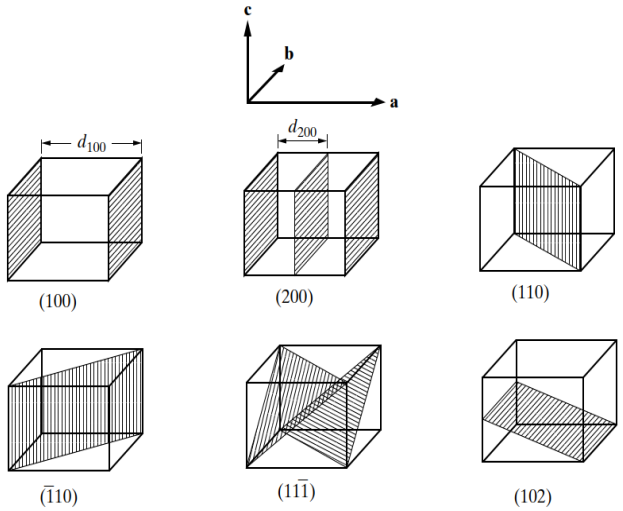


Figure 2.2: Miller indices of lattice planes. The distance d is the plane spacing, from [7].

For crystal collimation studies the (110) and (111) planes are commonly used due to their high atomic density[1] and low planar spacing[2]. The higher the atomic density, the more atoms are packed closely together, resulting in a denser material.

In this section, the theory of channeling in bent crystals is reviewed. In particular, some of the coherent phenomena that can take place when charged particles pass through a crystal are planar channeling, volume reflection, and dechanneling (Figure 2.3).

[1] High atomic density refers to the number of atoms per unit volume in a crystal

[2] Low planar spacing refers to the distance between adjacent planes of atoms in a crystal.

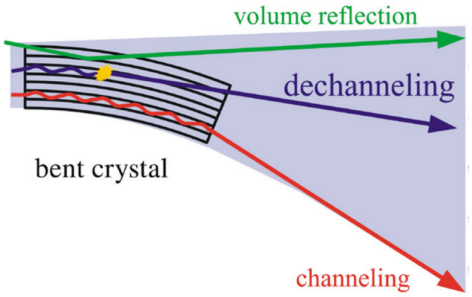


Figure 2.3: Illustration of coherent phenomena that can take place when charged particles penetrate a crystal, from [8].

2.1.1 Planar channeling

Planar channeling as proposed in [8] refers to the coherent phenomenon in which charged particles impinge a crystal almost parallel to its axes or planes, resulting in under-barrier motion along the crystal planes. The charged particles experience the electric field of atoms and undergo transverse oscillations, known as planar channeling oscillations, induced by the interplanar electric field (Figure 2.4). The wavelength of the planar oscilla-

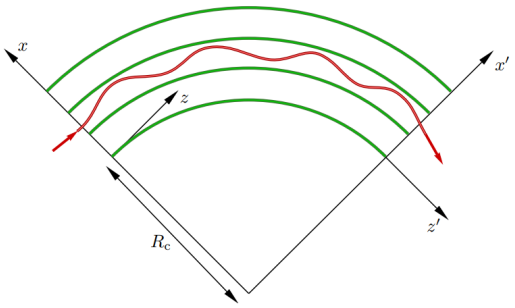


Figure 2.4: Illustration of deflection of a positively charged particle by planar channeling in the field of curved crystalline planes, from [9].

tion can be estimated using a harmonic approximation:

$$\lambda = \pi d_0 \sqrt{\frac{pv}{2U_0}}, \tag{2.1}$$

where d_0 is the interplanar distance, p the particle momentum, v the particle velocity and U_0 the potential well height for a straight crystal. During the process of planar channeling, charged particles undergo transverse oscillations as they traverse a crystal lattice. These oscillations occur under conditions of an interplanar potential U(x) and result in an oscillatory

under-barrier motion of the particles. The latter is depicted in Figure 2.5 for both (110) and (111) planes of a straight silicon crystal. Particles are there-

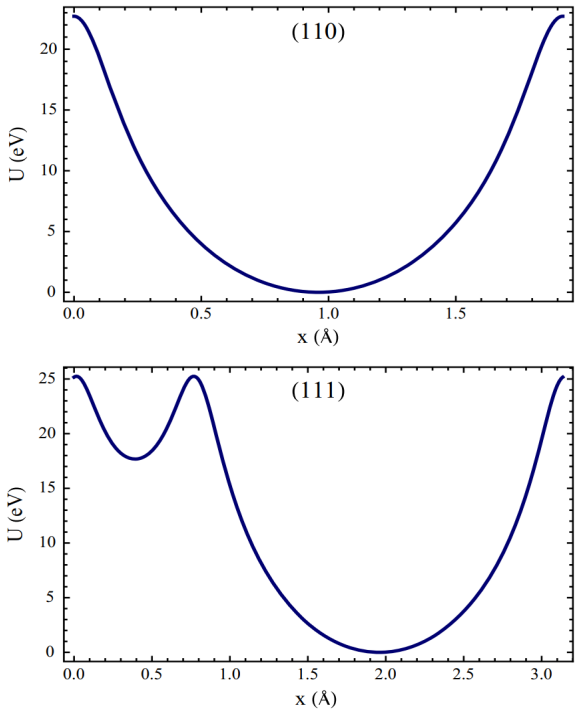


Figure 2.5: Potential well in a straight silicon crystal for (110) (top) and (111) (bottom) planes for positive particles as functions of the transverse coordinate, from [8].

fore trapped within the potential well generated by the adjacent crystalline planes and forced to follow the path of the crystal lattice. As the particles continue along this path, they experience periodic oscillations. The amplitude of these oscillations depends on the angle of incidence of the particles and the strength of the interplanar potential. In general, particles with smaller angles of incidence and lower kinetic energies experience stronger channeling effects and are more likely to be trapped within the potential well of the crystal lattice. Therefore the main condition to obtain channeling is the angle of impact of a particle with the crystal that must be less than the critical angle $\theta_{critical}$:

$$\theta_{critical} = \sqrt{\frac{2U_0}{pv}}. \tag{2.2}$$

2.1.2 Volume reflection

According to [10] when a particle collides with a bent crystal at an incident angle within the range of $\theta_c < \theta < \theta_b$ where θ_c is the critical angle and

θ_b is the crystal bending angle they may undergo volume reflection (VR) (Figure 2.6). During this process, protons are essentially reflected by the averaged potential of the crystalline planes upon impact. The reflection occurs when particles collide with a crystalline plane at a tangential angle to their momentum.

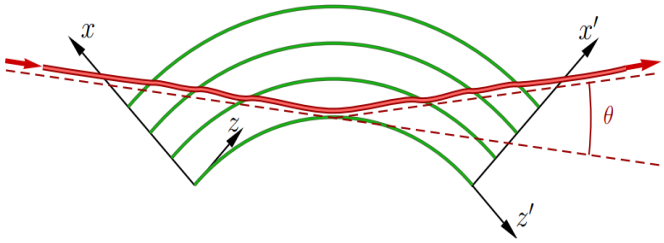


Figure 2.6: Illustration of volume reflection, from [9].

2.1.3 Dechanneling

As formulated in [10, 43] when particles are channeled by a bent crystal they are subjected to a centrifugal force pv/R (where R is the bending radius). As a result of a crystal bend, the channeled particles are shifted towards the atomic plane and may experience changes in their transverse energy with each interaction. If the total transverse energy exceeds the maximum of the potential well, the particle may lose its channeling condition (Figure 2.7). This process is known as dechanneling (DC) and can lead to a decrease in the initial population of channeled particles. In other words, dechanneling contributes to the reduction of the number of particles that remain in the aligned path through the crystal lattice.

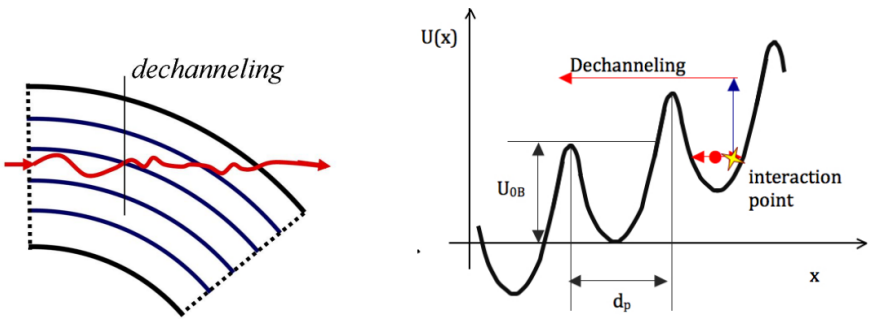


Figure 2.7: Illustration of the dechanneling process. Particles with trasnverse energy greater than the maximum of the pontential well lose the channeling condition, from [10].

2.1.4 Axial channeling and skew planes

Other symmetries known as Skew Planes (SK) appear diagonally to the planes used for channeling, as illustrated in Figure 2.8 (Left) for a silicon crystal oriented through the axis <111>. They can trap charged particles but with lower efficiency due to the lower potential well they can generate, and with a lower deflection angle compared to channeling. In the presented case in Figure 2.8 (Left), the stereogram shows that the skew planes are oriented at 30° and 60° with respect to the planar channeling. The crystal rotational axis is identified as pitch, yaw, and roll, as shown in Figure 3.11 (Right). In this setup, the incident particles are oriented along the x-axis, and the crystalline plane direction is along the roll axis. By orienting the crystal on the yaw axis, channeling can be achieved [11]. Encountering skew planes during the rotation of the crystal on the yaw axis can still result in particle deflection, but with significantly lower efficiency compared to planar channeling. Consequently, locating these planes can be challenging, and it is crucial to avoid them.

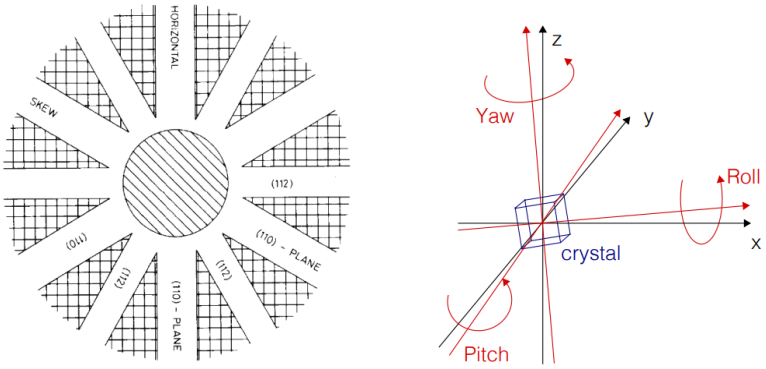


Figure 2.8: Stereogram of a crystal aligned to Si axis <111> (left). Reference system for orientation axis (right), from [11].

2.2 Crystal Collimators at the LHC

As reported in [12], collimation is a delicate operation for a circular accelerator machine, and its efficiency is fundamental to prevent component damage without reducing beam intensity. The separation of beam halos from the beam core with the use of bent crystals improves the collimation system, and only material with perfect lattice quality is used. These devices are composed of mono-silicon with very low or zero dislocations in the crystalline lattice placed on a metal holder that applies a primary curvature to

the device. The nature of the deformation can be either of elastic or plastic origin. Elastic primary curvature may be imposed by a mechanical application of forces to generate a bending momentum using the metal bender. Plastic deformation may be applied by generating a thermal gradient or elemental variation of crystal composition. The collimators are then aligned with the circulating beam with the use of special Target Collimator Primary Crystal (TCPC). This section will present the types of crystals that can be used in the LHC, along with an explanation of how these crystals are characterized—a crucial task that directly impacts their performance in high-energy physics applications. Additionally, the process of maneuvering and aligning collimators during operations with high-energy beams will be discussed.

2.2.1 Anticlastic curvature

The anticlastic curvature is an elastic reaction of the material to a primary curvature, transforming a flat surface into a saddle-shape (Figure 2.11). This bending occurs on the same plane of the primary curvature but along the perpendicular direction and with opposite convexity. An important

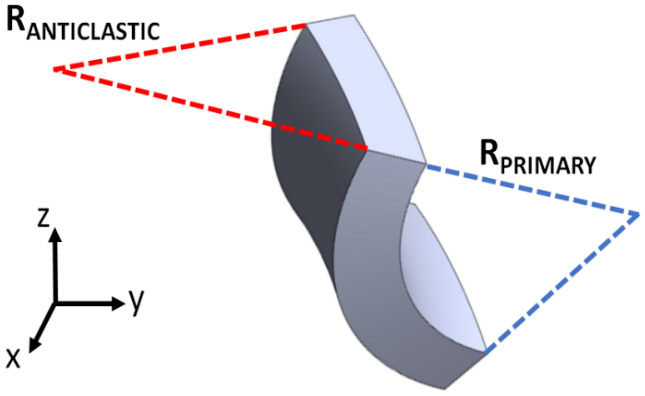


Figure 2.9: Illustration of anticlastic bending, from [12].

advantage of this geometry is the possibility to minimize the quantity of material close to the beam. As it can be seen in Figure 2.10, the central portion of the crystal used for channeling is the very far from the holder structure. This feature can be of critical importance in case of beam collimation, where the least amount of material should be placed near the beam halo in order to avoid perturbation of the main beam dynamic [12]. Furthermore, the anticlastic curvature is obtained on the (110) planes that are equidistant. Crystal collimators bent with this technique are called *strip crystals* (ST).

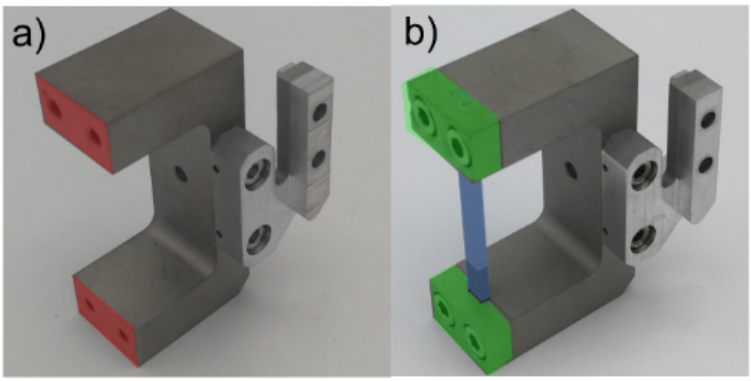


Figure 2.10: Holder for LHC crystal collimator (a); silicon sample (highlighted in blue) mounted on the red surfaces by clamps (highlighted in green) in order to force an arched shape, from [12].

2.2.2 Quasi-mosaic curvature

The quasi-mosaic (QM) curvature (Figure 2.11) is another secondary deformation effect of a crystal to a primary curvature. The planes bent by the QM effect are orthogonal to the main surface of the plate. A QM crystal highlights an interesting feature for channeling experiments, it allows the interception of a broad beam because the largest face of the crystal is exposed to the beam [47].

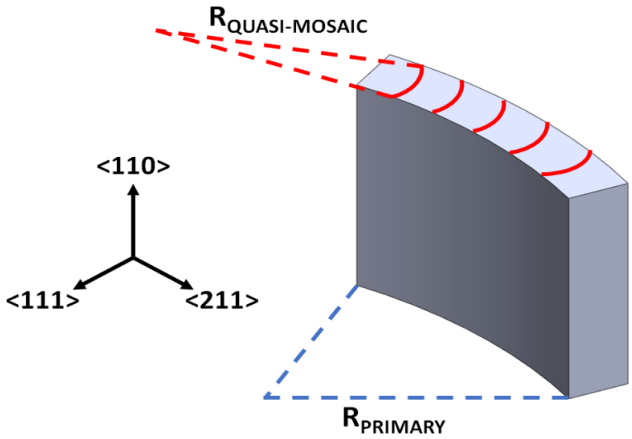


Figure 2.11: Illustration of quasi-mosaic bending. The curvature is induced along the (111) plane, from [12].

As it can be seen in Figure 2.12, as opposed to strip crystals, quasi-mosaic crystals are bent along the larger face of the material inducing a curvature on the (111) planes.

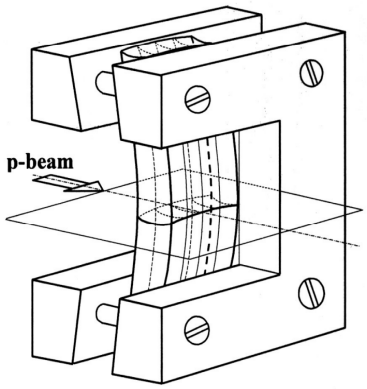


Figure 2.12: Sketch of a quasi-mosaic crystal holder. By clamping the crystal an anticlastic curvature along the larger face of the material is obtained, from [13].

2.2.3 Crystal characterization

Characterizing a crystal collimator refers to measuring the deviation of the crystal from its ideal shape and orientation, which can affect the collimation of high energy beams. The three properties measured displayed in Figure 2.13 are:

1. **Bending angle**: The bending angle of a crystal collimator is defined as the angular deviation of the crystal's surface from a perfect plane, measured as the ratio of the crystal length in the beam direction (l) to its bending radius (R), expressed as $\theta_b = \frac{l}{R}$. Channeled particles passing through the crystal collimator will follow its curvature, resulting in the deflection of particles by an angle equal to the bending angle (θ_b). This deflection angle is known also as the channelling angle (θ_{ch});

2. **Miscut angle**: defined as the angle between the orientation of the lattice planes utilized for channelling and the orientation of the external surface of the crystal;

3. **Torsion angle**: defined as the angle of twisting or rotation deformation of the crystal along the vertical axis [14].

Silicon crystals in the form of "strips", designed for studies of coherent interactions between crystals and charged particle beams, can be fabricated using silicon anisotropic etching techniques [48], though other techniques also exist. Although this method of crystal production enables high crystalline quality, the devices still need to be characterized to evaluate

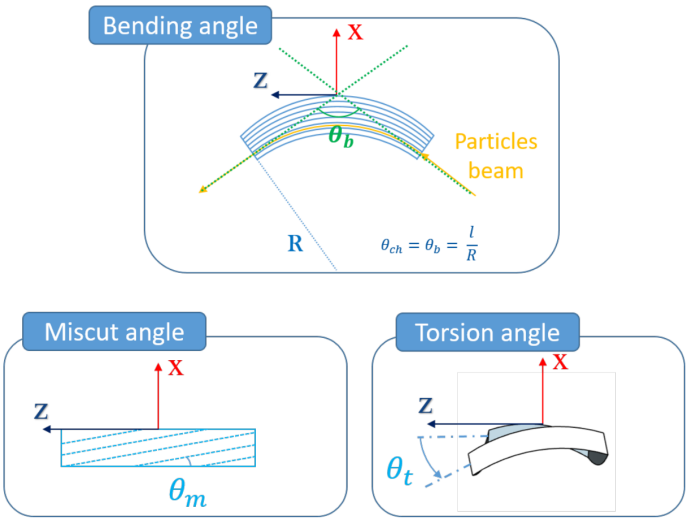


Figure 2.13: Schematic illustration of the parameters used in crystal characterization, from [14].

their feasibility for operational use in the LHC machine. Therefore, crystals are characterized using X-ray diffraction [21], and this was done with the X'Pert machine from Malvern Panalytical [49] at CERN, which offers high-resolution analysis of crystalline materials. However, the machine was not initially designed to accommodate crystal collimators, requiring several modifications for this purpose. Additionally, the machine's vendor-locked software cannot be customized for optimal measurement, necessitating an expert operator for data acquisition and additional post-processing steps.

Thus, crystal characterization becomes a tedious and time-consuming endeavor. This study addresses these challenges with a custom X-ray diffractometer, optimized specifically for automated and efficient crystal characterization. This innovative approach significantly streamlines the process, enhancing flexibility in crystal characterization.

The procedures for crystal characterization were previously developed in [29]. This work aims to adapt these procedures to the new X-ray diffractometer used in this endeavor.

2.2.4 Target Primary Collimator Crystal

Precise goniometers [50] are utilized to hold bent crystals, which include both a linear and rotational stage. The linear stage enables the crystal to be adjusted until it makes contact with the beam halo, while the rotational stage allows for accurate positioning to locate the channeling orientation. As formulated in [51], bent crystals must be placed at distances from circu-

lating beams as small as 2 mm. The angular orientation of the crystal must be finely adjusted, with a precision below 1 μrad in order to ensure that the channelling conditions are respected. This called for the development of a special assembly, named Target Primary Collimator Crystal (TCPC). In consists of a replacement beam pipe chamber that is used when the crystal collimator is in parking position, a linear positioning system, an interferometry goniometer used for angular adjustments of the crystal, mounted on the bender; and a vacuum tank that includes all these sub components (Figure 2.14).

Currently four TCPC devices are installed in the LHC, all located in Insertion Region 7, two in horizontal and two in vertical orientation. Figure 2.15 depicts the schematic view of the actuation of the rotational axis with respect to the horizontal or vertical orientation of the crystal. Two crucial components within the TCPC include a rotational stage governed by a piezoelectric actuator for ensuring precise and controlled crystal rotation, and an interferometry system designed to measure angular movements.

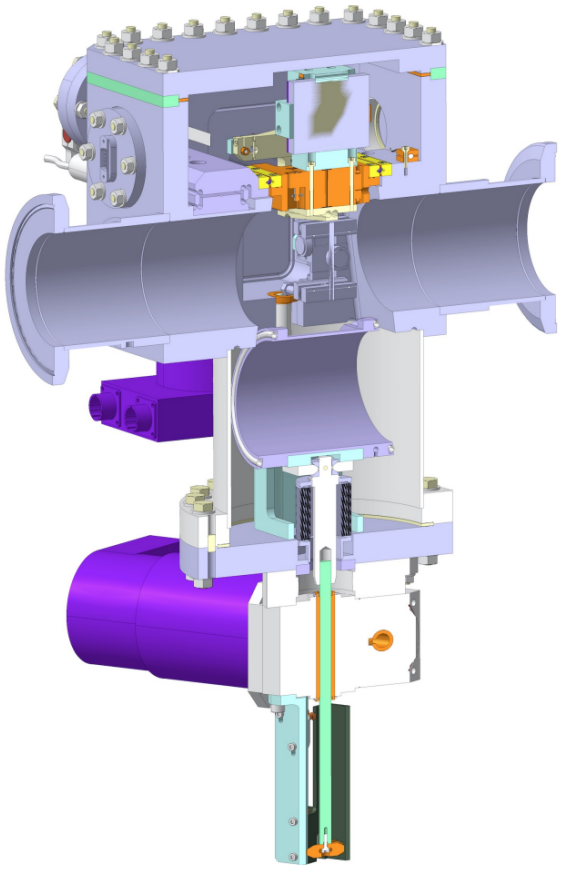


Figure 2.14: Cross section of the TCPC. In this configuration the beam pipe is in parking position, from [15].

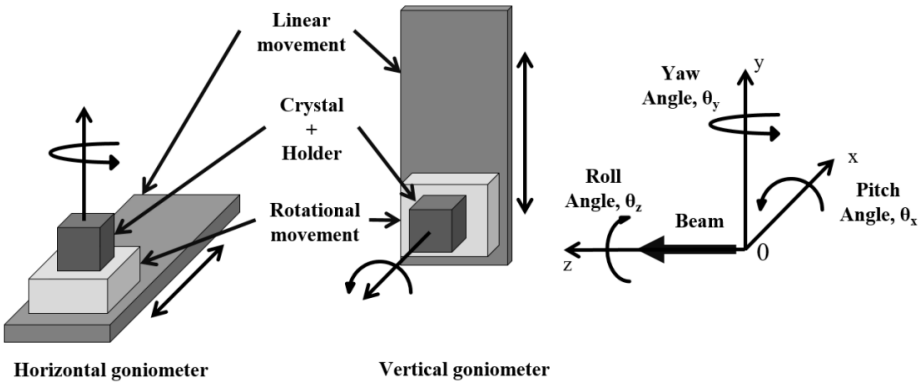


Figure 2.15: Actuated angle for horizontal and vertical orientation with respect to beam axis, from [16].

Piezo actuators serve to directly convert electrical energy into mechanical energy and vice versa, enabling precise motions within the subnanometer range. The crystal rotational stage's actuator utilizes a piezoelectric piston, complemented by a spring for preload to prevent backlash. The pivot function is accomplished through the flexion of thin, flexible blades within a single part, eliminating any contact friction between elements (Figure 2.16). Piezo actuators offer numerous advantages: no moving parts such as gears or bearings, displacement based on crystalline solid-state dynamics with negligible wear and tear, and no abrasion or lubrication requirements. These actuators are well-suited for ultrahigh vacuum applications. Additionally, piezo actuators exhibit rapid response times in the order of microseconds. The current design of the rotational stage presents a nominal angular range of 20 mrad, coupled with a linear stroke of the piezoactuator measuring 40 μm [52].

The Fabry-Pérot sensor (FPS) [53] interferometer from Attocube, based on fiber optics, enables the measurement of a target's relative displacement with picometre resolution and provides real-time position output with a 10 MHz bandwidth [31]. In various industrial and scientific applications, precise position detection of a target at high velocities is essential. One such application involves measuring the displacement between the interferometer heads and the crystal collimator. As depicted in Figure 2.17, the interferometer captures linear motions of a target relative to the interferometer head. By utilizing readings from the two stationary interferometric heads, the angular (yaw) movement can be determined using straightforward trigonometric relationships [18].

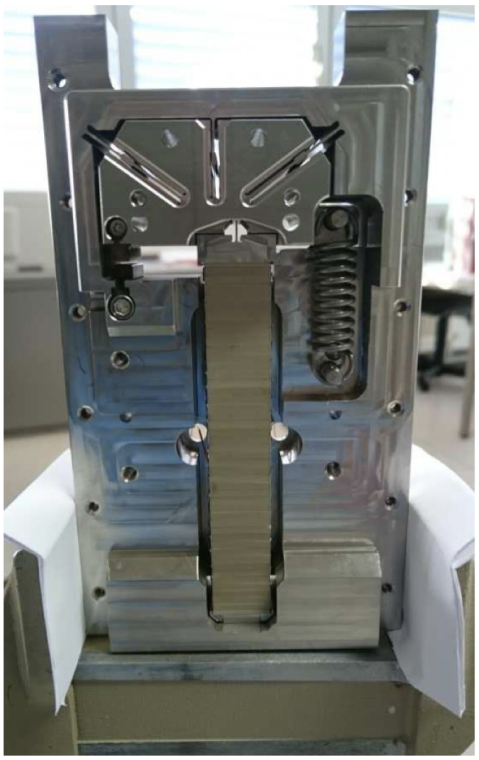


Figure 2.16: Rotational stage of the goniometer. It can be seen the piezo actuator, the spring and the blades that allow the rotation of the crystal, from [17].

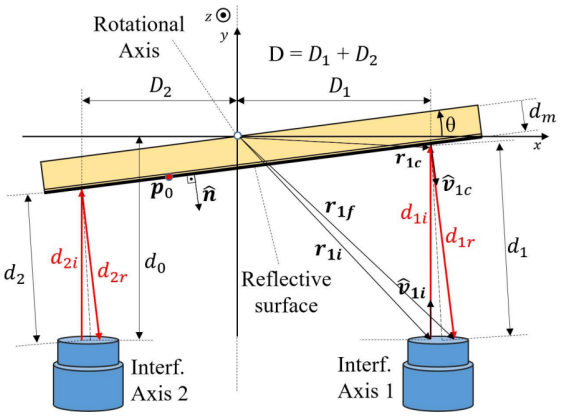


Figure 2.17: Schematic of the Interferometer Head and Reflector. This diagram shows the two independent interferometer axes mounted parallel to each other on a stationary support. The beams are reflected by a reflective surface attached to the rotating component of the system, from [18].

Beam Intercepting Devices (BIDs), such as the LHC collimators, present challenges due to their stringent requirements for high positioning precision. Currently, the acquisition and control systems for these collimators use commercially available PXI cards, controllers, and chassis from National

Instruments [54].

As part of the control system update planned for Long Shutdown 3 (LS3)[3], a high-performance field Sensors Acquisition and Motion Control framework (SAMbuCa) [55] has been conceived. SAMbuCa is an in-house solution offering a modular and flexible hardware and software framework designed to meet the demanding requirements of motion control applications at CERN.

A key application of this framework is the control of the Target Primary Collimator Crystal. The upgrade also involves redeveloping the control system for the Target Primary Collimator Crystal (TCPC) using customizable firmware electronics devices like field programmable gate arrays (FPGAs). Employing a System on Chip (SoC) that integrates FPGA and CPU provides greater flexibility, enabling efficient handling of complex control algorithms and data processing tasks simultaneously. Additionally, the FPGA and CPU combination in an SoC enhances communication and data exchange between system components, improving overall performance and responsiveness [56].

This work is embedded within the SAMbuCa framework to establish a new testbench for evaluating the innovative crystal piezo goniometer controller algorithm presented in [56]. The focus is on leveraging FPGA design and ensuring seamless integration of hardware and software, thus optimizing the testing process and advancing the development of the control system.

2.2.5 Crystal alignment

Essential devices used in the alignment of standard and crystal collimators in the LHC are Beam Loss Monitors (BLMs). According to [57, 58], the main purpose of the BLMs is to provide continuous surveillance of particle losses by converting particle shower information into electrical signals, which are then compared with limits. If the limits are exceeded, extraction of the LHC beam from the ring is initiated to stop the irradiation of equipment. In the case of the LHC, the protection function is often linked to the quench prevention of the superconducting magnets. Another important function of BLMs is to give a feedback of losses produced when high-energy hadrons (such as a protons or neutrons) collide with collimators producing the so-called *hadronic shower*. For this reason BLMs are normally located downstream of the collimators. The loss signal is then used to align stan-

[3]Long Shutdown 3 (LS3) is a scheduled shutdown of the LHC for upgrading technology and instrumentation in preparation for the Hi-Lumi LHC

dard and crystal collimators jaws with the beam envelope. The detectors used for beam loss monitoring (Figure 2.18) are filled with a gas, such as nitrogen or argon, and operate based on the principle of ionization. When a high-energy hadronic shower interacts with the gas, it creates electron-ion pairs, which are then detected by the ionization chamber. The detection is achieved by one or more high voltage gaps along the length of the chamber. The resulting electrical current generated by the electron-ion pairs is measured and directly proportional to the amount of beam loss that occurred at the location of the beam loss monitor and it's expressed in units of Gy/s.

Figure 2.18: Beam Loss Monitor (BLM) in the LHC (left) and ionization chambers (right), from [19]

The crystals collimator, installed into the TCPC described in Section 2.2.4, is utilized during ion beams operation and aligned with the circulating beam envelope to search for the optimal channeling orientation. The latter is identified with the use of Beam Loss Monitors signals as feedback. When aligning a crystal collimator the setup depicted in Figure 2.19 is used, in which it can be seen a standard primary collimator, a crystal collimator and a secondary collimator.

The key steps that need to be taken to find the channeling orientation of a crystal collimator are: movement of the jaw of the primary collimator towards the beam to form a reference cut in the beam halo (Figure 2.20) until an alignment spike (Figure 2.21) is observed; movement of the crystal collimator in steps towards the beam until it touches the reference beam halo created by the primary collimator (Figure 2.22); unti the primary collimator is then retracted to it's parking position (Figure 2.23);

Finally, a procedure called an *angular scan* [11] (Figure 2.24) is initiated. During this procedure, the crystal is slowly rotated (on average the step size during this process is 1μrad/s). At the end of the scan, a signature as the one showed in Figure 2.25 is observable from the beam loss monitor data located in proximity of the crystal and of the secondary collimator. From the

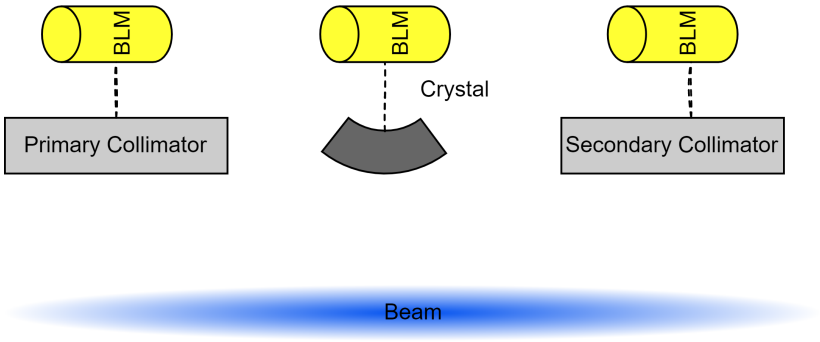


Figure 2.19: Scheme of the initial collimators configuration during crystal studies. The primary collimator is used to form a reference halo that is then used to align the crystal collimator. The secondary collimator absorbs the channeled particles by the cystal.

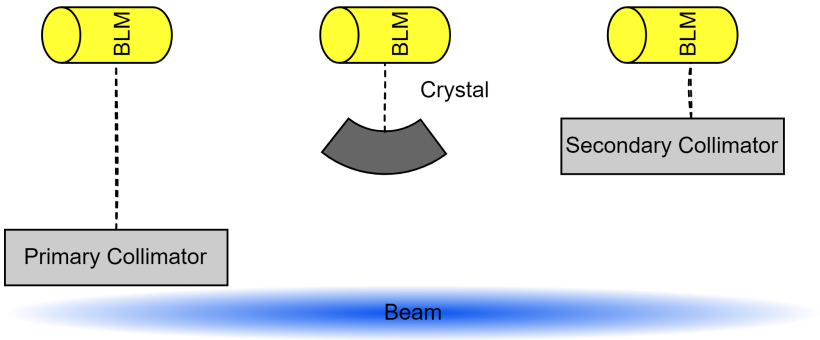


Figure 2.20: Illustration of the primary collimator creating a reference cut in the beam halo.

signature, 3 main patterns can be observed: (1) *Amorphous shoulders*. The channeling orientation is far away that the crystal behaves like a standard collimator; (2) *Channeling well*. A minimum in the loss pattern observed at the crystal location and increase of losses in correspondence of the secondary collimator due to the beam particle being channeled by the crystal; (3) *Volume reflection plateau*. Particles bounce from the crystalline planes instead of being channeled [59].

During the angular scan, to better see the characteristic signature of channeling from the BLM signal, white noise is introduced into the beam

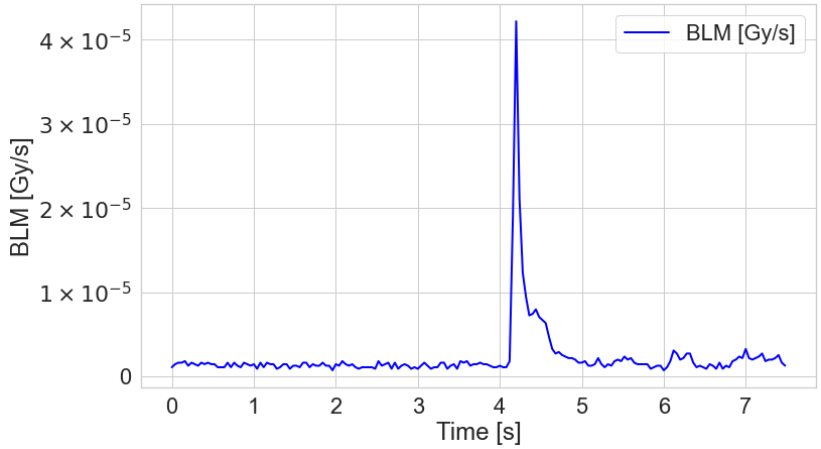


Figure 2.21: BLM signal while jaw is moving. This is identified as an alignment spike: the rise of loss corresponds to the moment when the jaw touches the beam halo.

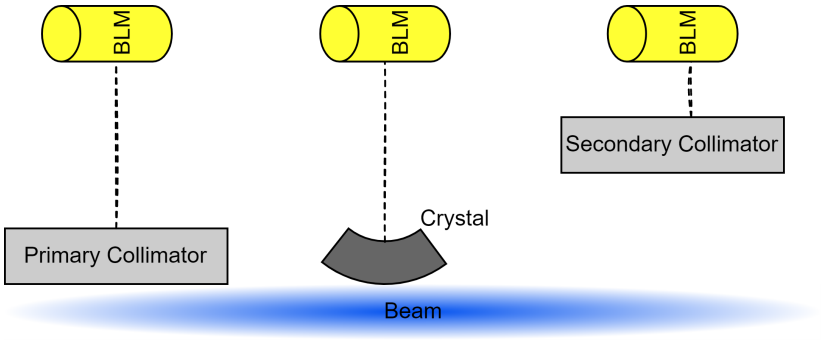


Figure 2.22: Illustration of the primary collimator and crystal collimator touching the halo.

using the Active Transverse Damper (ADT) [45]. As formulated in [4], achieving the optimal performance with crystal collimators necessitates an angular alignment of the crystal with respect to the beam envelope, requiring precision on the order of μrad—a highly challenging task. Traditionally, this alignment was done "manually" by relying on human visual feedback to identify the channeling signature through signals from beam loss monitors during crystal rotational scans. Given the goniometer's angular range of 20 mrad and an average scan step size of 1 µrad/s, locating the channeling orientation is a complex and time-consuming task. This study focuses on leveraging deep learning techniques to automate the detection of main planar channeling conditions in crystal collimators during angular scans, aiming to streamline and enhance the efficiency of the alignment process.

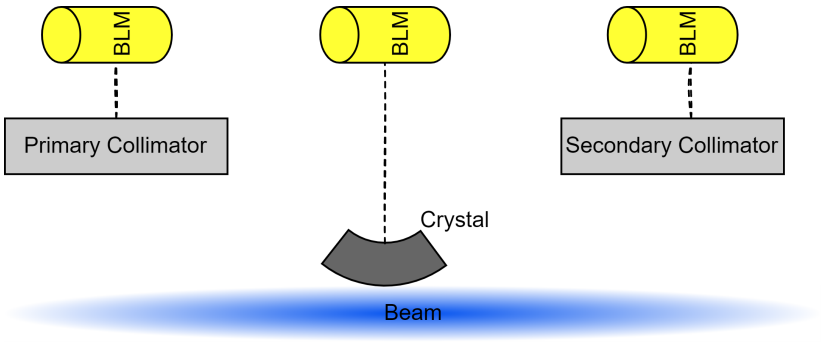


Figure 2.23: Illustration of the primary collimator retracted in it's parking position while the crystal is touching the halo.

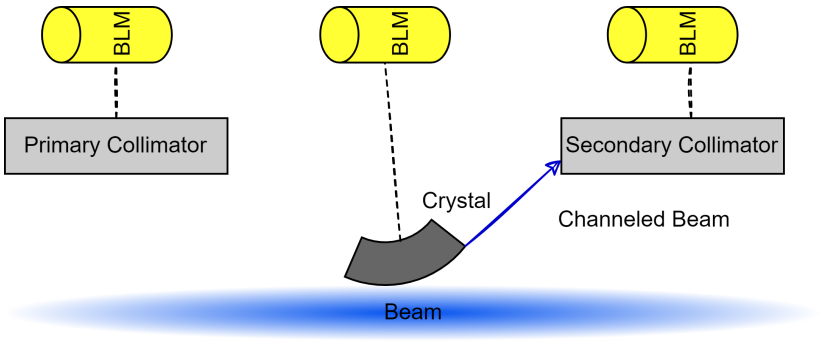


Figure 2.24: Illustration of the crystal rotated and channeling halo particles towards a secondary collimator.

2.2.6 Main goals and contributions

This thesis presents several key contributions aimed at enhancing the performance of the crystal collimation in the Large Hadron Collider (LHC). The primary goals of the research include:

- **Development of a software framework:** A modular software framework has been developed and integrated into the in-house X-Ray Diffractometry system. This framework facilitates efficient control and monitoring of experimental setups, enhancing the measurements.
- **Optimisation of control techniques for crystal collimators:** The research focuses on the optimization of control techniques for the piezo-goniometer rotational stage of the TCPC device. Leverag-

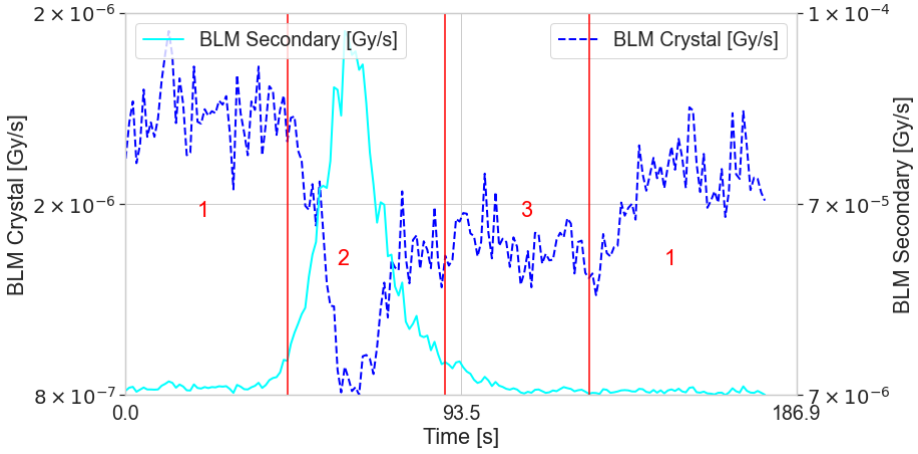


Figure 2.25: BLM signal as a function of time while crystal is rotating identified as a *"Channeling well"* (dark blue line). **BLM signal** as a function of time observed in correspondence of the secondary collimator, while crystal is rotating (light blue line), from [4].

ing the SAMbuCa framework, advanced control algorithms have been implemented.

- **Autonomous understanding of crystal channeling:** This thesis investigates methods for the autonomous detection of channeling conditions in crystal collimators during angular scans. Machine learning algorithms have been developed and tested, allowing for the automation of the alignment process, promising to significantly reduce the time and complexity traditionally associated with manual alignment techniques.

- **Real time crystal monitor:** The research integrates machine learning techniques to continuously monitor crystal collimators during operations. This approach enhances the understanding of crystal behavior and contributes to the development of more efficient utilization of crystals.

- **Experimental validation:** Comprehensive experimental validation of the proposed methodologies has been conducted, demonstrating the feasibility and effectiveness of the developed techniques in real-world applications at CERN.

Chapter 3

Crystal characterization with X-ray beam

The precise characterization of crystal collimators is crucial for understanding their structural properties and performance in the LHC. X-ray diffraction (XRD) is a fundamental technique used to investigate the atomic arrangement within crystals, providing invaluable insights into material structure and crystallographic parameters. This chapter explores the principles of X-ray diffraction and Bragg's Law. It also examines the operational aspects of the custom X-ray diffractometer used in this study, including its components, validation, software, algorithms, and graphical user interface (GUI). The design of a custom diffractometer, coupled with modular and advanced software, addresses the need to streamline and enhance the complex task of characterizing crystal collimators. This chapter details the semi-automatic X-ray diffractometer used in this study, advancing the validation of beam-intercepting devices, specifically crystal collimators.

3.1 Concepts of X-ray diffraction

X-rays are high-energy electromagnetic radiation. They have energies ranging from about 200 eV to 1 MeV; which puts them between γ-rays and ultraviolet (UV) radiation in the electromagnetic spectrum. X-rays show wave nature with wavelength ranging from about 10 to 10^{-3} nm. According to the quantum theory, the electromagnetic wave can be treated as particles called photons or light quanta. In the electromagnetic spectrum (Figure

3.1) x-rays therefore occupy the region between gamma and ultraviolet rays [20].

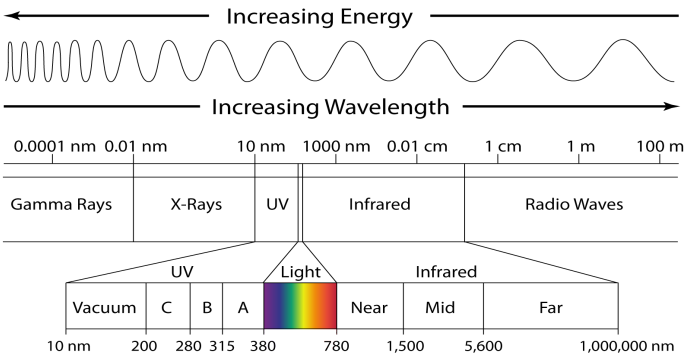


Figure 3.1: Electromagnetic spectrum, X-rays have a wavelength ranging from about 10 to 10^{-3} nm, from [20].

Diffraction is a general characteristic of all waves and can be defined as the modification of the behavior of a wave by its interaction with an object. If a photon collides with a loosely bound electron (Figure 3.2) and the collision between photon and electron is considered an elastic one the electron is knocked aside at an angle ϕ and the direction of the incident X-ray photon is altered by an angle 2θ. In this collision process, a part of the energy hv_0 of the incident X-ray photon is converted to kinetic energy of the electron. As a result, the energy of the incident X-ray photon after collision becomes hv which is smaller than hv_0. This process of absorption and remission of electromagnetic radiation is known as scattering [21, 22].

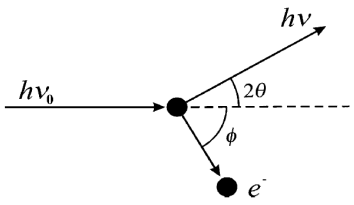


Figure 3.2: Scattering of x-rays by an electron, from [21].

In the case of interaction of a X-Ray beam with the electrons that orbit the nucleus of an isolated atom the waves that are scattered are multiples. Let's consider for simplicity two waves incident on two electrons of an atom (Figure 3.3) [21, 22]. The upper wave is scattered by electron A in the forward direction. The lower wave is scattered in the forward direction by

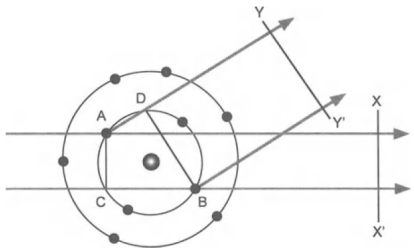


Figure 3.3: Scattering of x-rays by an atom, from [22].

electron B. The two waves scattered in the forward direction are said to be in phase (peaks and troughs align with each other at a given point in time resulting in a wave with a greater amplitude) across wavefront XX' because there is no path (or phase) difference. The other scattered waves in Figure 3.3 will not be in phase across wavefront YY'. If these two waves across wavefront YY' are added, the amplitude of the scattered wave is less than the amplitude of the wave scattered by the same electrons in the forward direction. Thus can be defined the atomic scattering factor (f) to explain how efficient an atom is scattered in a given direction:

$$f = \frac{\text{Amplitude of wave scattered by an atom}}{\text{Amplitude of wave scattered by one electron}} \tag{3.1}$$

When scattering is in the forward direction (i.e. scattering angle $\theta = 0°$) f is equal to the total number of electrons (the atomic number). But as θ increases, the waves become more and more out of phase because they travel different path lengths and, therefore, the amplitude, or f, decreases. A diffracted beam may be defined as a beam composed of a combination of multiple waves that have been scattered or reflected from a material or object [21, 22].

3.1.1 Diffraction from crystalline materials. The Bragg's law

The law that relates the wavelength of X-Rays, the angle of incidence and the spacing of atoms in a crystal lattice is the the Bragg's law. It can be derived considering an incident wave with an angle of incidence of θ on a crystal medium (Figure 3.4).

Now the criterion for the existence of the diffracted wave with an angle of 2θ (diffraction angle) from the transmitted one is that the scattered x-rays should all be in phase across a wavefront such as BB'. For this to be

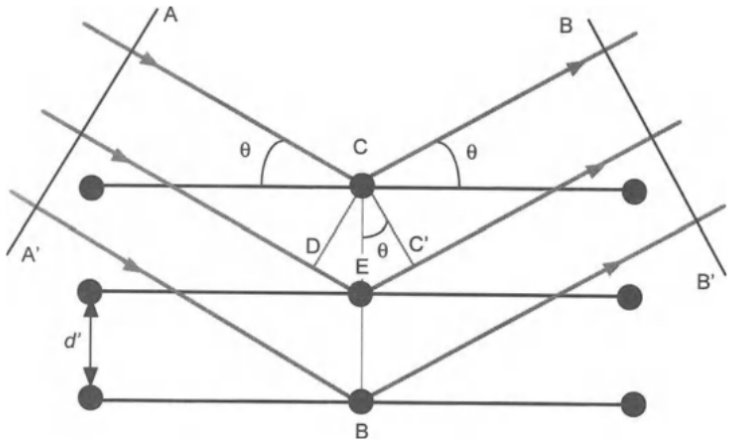


Figure 3.4: Diffraction of X-ray from a crystal, from [22].

so, the path lengths between wavefronts AA' and BB' for the rays shown must differ by exactly an integral number (n) of wavelengths λ. Therefore the path difference, δ, must be:

$$\delta = n\lambda, \tag{3.2}$$

where n is an integer. Equation 3.2 can be rewritten as:

$$\delta = \mathrm{DE} + \mathrm{EC}' = 2\mathrm{EC}' = 2\mathrm{CE}\sin\theta \tag{3.3}$$

where CE in 3.3 is the interplanar spacing *d'*. Therefore we can rewrite:

$$\delta = 2d'\sin\theta. \tag{3.4}$$

By combining Equations 3.2 and 3.4:

$$n\lambda = 2d'\sin\theta. \tag{3.5}$$

The parameter n is known as the order of reflection and represents the number of X-ray wavelengths that fit into the path difference between two successive atomic planes in the crystal lattice. For example, if n = 1, it means that only one X-ray wavelength fits into the path difference between two successive atomic planes, and this corresponds to the first order diffraction peak. Similarly, if n = 2, it means that two X-ray wavelengths fit into the path difference, and this corresponds to the second order diffraction

peak, and so on. Equation 3.5 can be rewritten as:

$$\lambda = 2d \sin\theta, \tag{3.6}$$

where $d = d'/n$. In conclusion Equation 3.6 states that constructive interference between scattered X-rays occurs when the path difference between the waves is equal to an integer multiple of the X-ray wavelength. In other words, the X-rays scattered from different atomic planes in the crystal lattice will be in phase with each other if the path difference between them is equal to an integer number of X-ray wavelengths [22].

3.1.2 Diffractometer and diffraction peak intensity

This section will demonstrate how structural information can be obtained from crystalline samples using a diffractometer, which is the most common method for measuring the intensity of a scattered X-ray beam from such materials. The diffractometer measures X-ray diffraction intensity from the sample at various scattering angles, allowing for analysis of the measured intensity data to extract relevant structural information. The basic design of a diffractometer (Figure 3.5) consists of an X-ray source, a sample holder, and a detector.

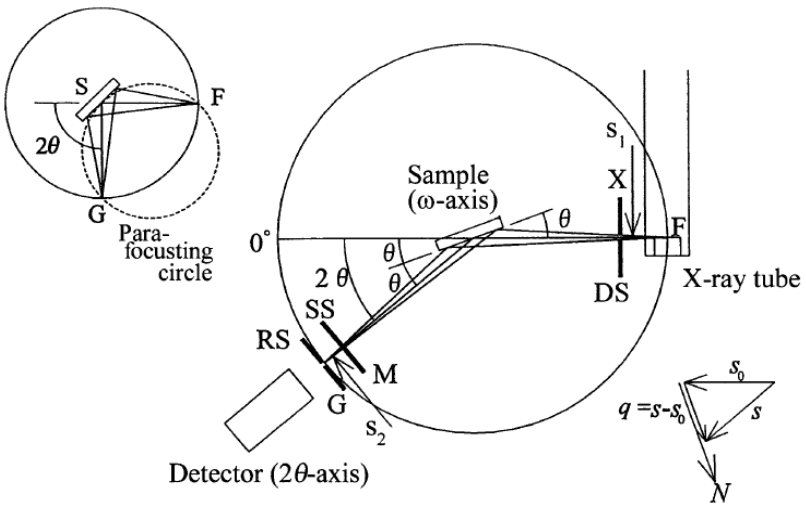


Figure 3.5: Schematic illustration of a diffractometer. The x-ray source, the sample and the detector lie in a circumference known as the focusing circle, from [21].

The three components lie in a circumference known as the focusing circle. By a rotation of the sample a set of point representing the intensities of the

diffracted beam can be computed. The incoming radiation is diffracted also at angles that deviate slightly from the Bragg angle, however, the highest intensity of diffracted X-rays occurs when the exact diffraction angle satisfies the Bragg law. The intensities of the diffracted x-ray beam counted by the detector as a function of the angle of the sample holder are referred to *rocking curve* (Figure 3.6) [21].

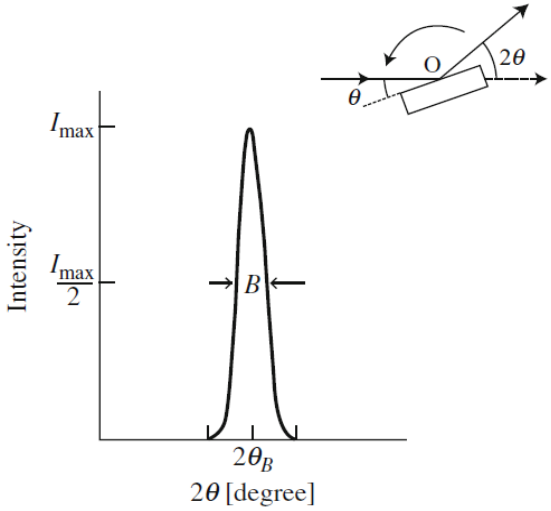


Figure 3.6: Intensity diffracted from a crystalline sample as a function of the sample angle, from [21].

Every lattice plane within a crystal produces a diffraction peak, and the combination of these peaks creates a diffraction pattern. This pattern arises from the interaction between a sample and X-rays and provides important insights into the internal structure of the sample. Specifically, it can be utilized to determine the arrangement and spacing of atoms within a crystal lattice, as well as the orientation of the crystal. Moreover, the intensity of diffraction peaks can furnish information about the composition, crystalline quality, and size of the crystal domains. In summary, the diffraction pattern is an effective method for examining the properties and structure of materials [21].

3.2 Instrumentation of the custom X-ray diffractometer

The X-ray diffractometer used in this work, depicted in its Computer-aided Design (CAD) in Figure 3.7, includes several key components: an X-ray source that can be rotated using a stepper motor; a monochromator, which

is movable in two degrees of freedom via stepper motors (as shown in Figures 3.8 and 3.9); a slit, also movable in two degrees of freedom with stepper motors; a sample stage that allows movement in six degrees of freedom through a hexapod robot and crystal rotation via a stepper motor; an autocollimator that rotates with the aid of a stepper motor; and an X-ray detector, which can be rotated using a stepper motor (illustrated in Figure 3.10).

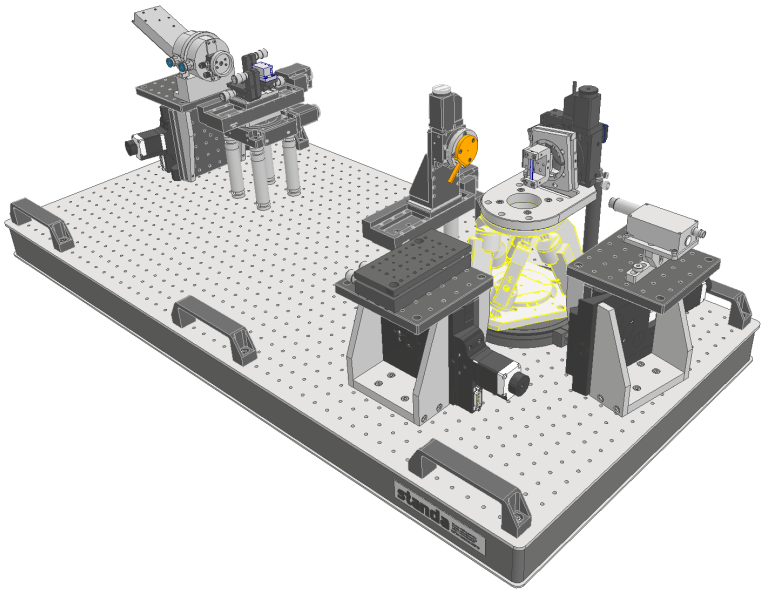


Figure 3.7: Visualization of the X-ray diffractometer design using Computer-Aided Design (CAD). The diagram illustrates the layout, with the X-ray source positioned on the left, followed by the monochromator. The crystal is positioned atop the hexapod, located between the slit and X-ray detector.

All stepper motors utilized in this work provide control with a precision of 0.01 mm or degrees. The sample stage, which is operated by a hexapod robot, offers even higher precision, exceeding 0.0001 degrees/mm. Detailed functionality of the sample stage will be described in Section 3.2.3.

3.2.1 X-ray source

X-ray sources typically consist X-ray tube, and a power supply. The X-ray tube is composed of a cathode and an anode, which are housed within a vacuum-sealed envelope. The voltage of the power supply creates an electric field that exerts a force on the electrons, causing them to move towards the anode (Figure 3.12). The X-ray source used in this work is an Amptek Mini-X2 [25]. It has a cathode composed of Silver (Ag) and emits electrons through a process called thermionic emission. This process involves the transfer of energy from the cathode to its constituent electrons, which results

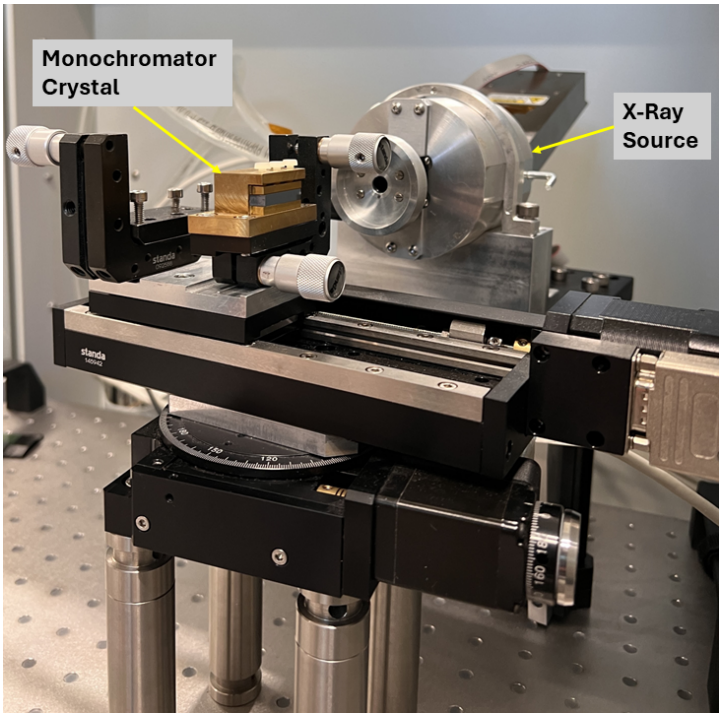


Figure 3.8: Photo showing the side view of the assembly of the X-Ray Source and of the Monochromator Crystal.

in the liberation of these particles from the surface of the cathode. The liberation of electrons is obtained heating the cathode filament introducing an electrical current into it. The free electrons created in this process are then accelerated towards the anode by a high voltage applied across the X-ray tube. In an atom, electrons occupy different energy levels or shells, which are defined by the principal quantum number (n). The lowest energy level, or shell, is the "K" shell (n=1), followed by the "L" shell (n=2), "M" shell (n=3), and so on as seen in Figure 3.11. As an incoming electron interacts with the atoms of the target in the X-ray source, it may undergo a collision with one of the target atoms. Such a collision may result in the ejection of an electron from the target atom, leaving behind a vacant electron shell. Subsequently, an electron from an outer shell of the atom may transition into the vacant shell, thus releasing an X-ray photon with a discrete energy level. This photon is referred to as a characteristic X-ray and possesses a specific wavelength, which is unique to the atom from which it is emitted. Two distinct types of radiation can be emitted when bombarding the anode with electrons. $K\alpha$ radiation is emitted when an electron from an outer "L" shell fills the hole left by an electron of the innermost "K" shell. While $K\beta$ radiation is emitted when an electron from the "M" shell falls into the K shell [23].

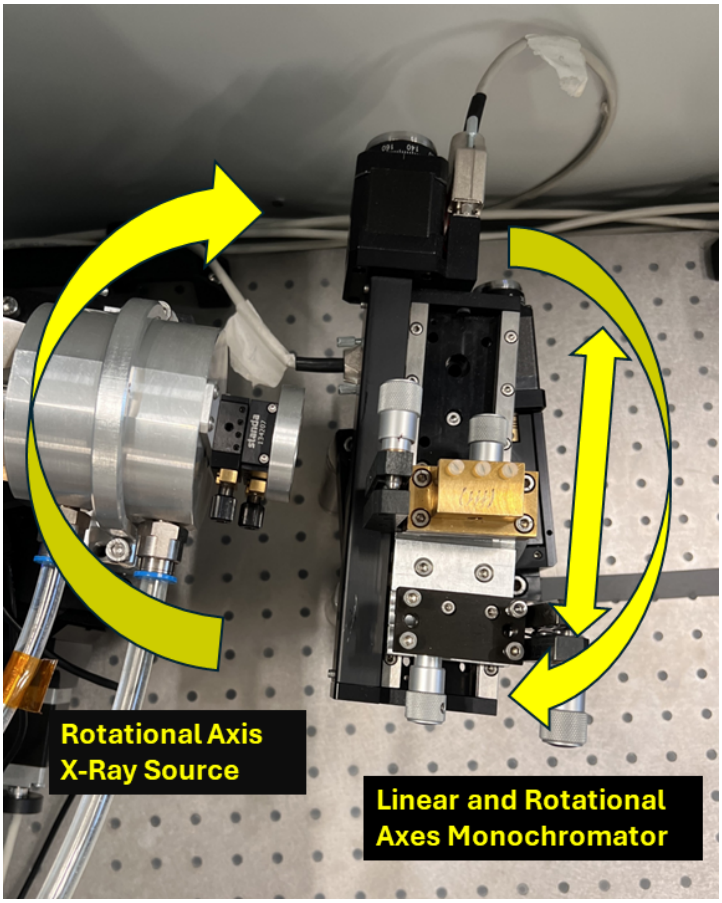


Figure 3.9: Photo showing the top view of the assembly of the X-Ray Source and of the Monochromator Crystal, including the specification of axes.

The anode material used in the X-ray tube is Silver (Ag), as this material has a high melting point and can withstand the high temperatures generated by the electron collisions. In addition, the source is water cooled to help dissipate heat and prevent damage to the material [24].

The spectrum of an X-ray beam is characterized by two distinct peaks known as K-alpha and K-beta, as previously described. An example of an X-ray spectrum obtained with the X-ray source used in this work is shown in Figure 3.13. As observed, the K-alpha and K-beta peaks appear at approximately 22 keV and 25 keV, respectively.

For the purposes of this study, we will focus exclusively on the K-alpha radiation because it has a higher intensity compared to the K-beta peak, providing more robust data for analysis. Additionally, the K-alpha radiation is often used in X-ray diffraction analysis because it allows better penetration and interaction with the sample material. This characteristic makes K-alpha radiation more suitable for detailed material characterization and quantitative analysis. Furthermore, in this study we will focus only in the

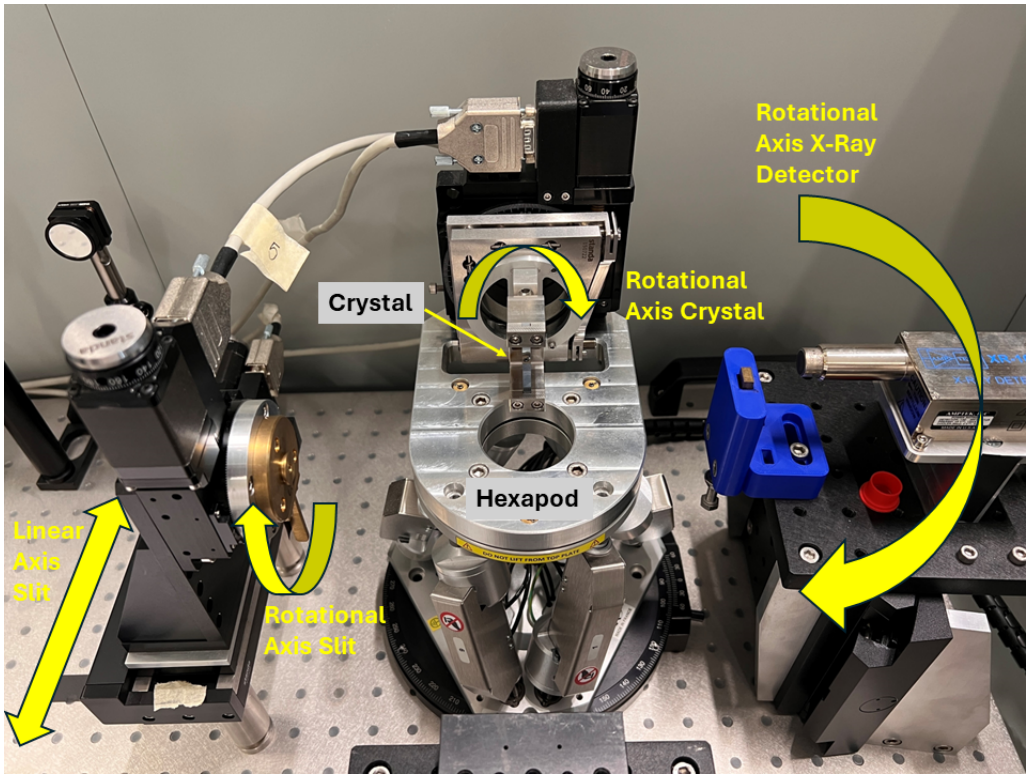


Figure 3.10: Photo showing the assembly of the Slit, Hexapod robot with crystal, and X-ray detector, including the specification of axes.

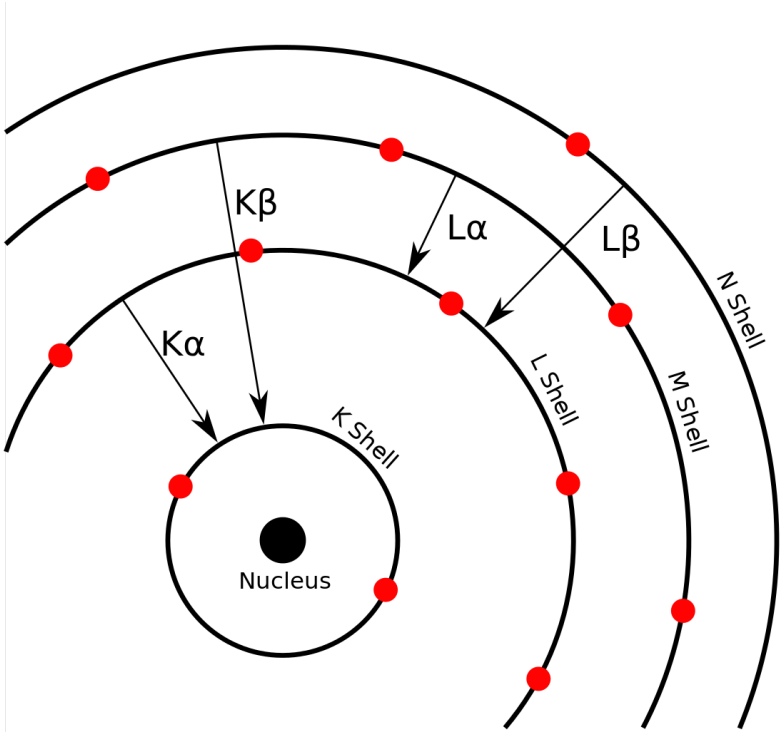


Figure 3.11: Atom shells, the lowest energy level, or shell, is the "K" shell (n=1), followed by the "L" shell (n=2), "M" shell (n=3), from [23]

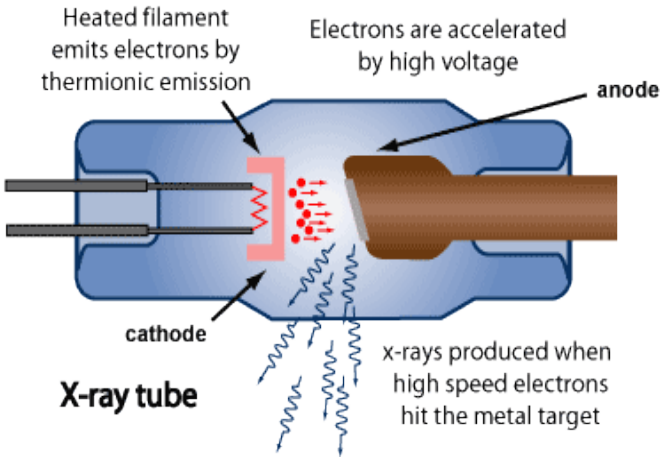


Figure 3.12: Illustration of the X-ray source working principle. An heated filament emits electron by thermionic emission. accelerated towards the anode by a high voltage applied across the X-ray tube. The collision between the accelerated particles and the anode releases x-rays, from [24].

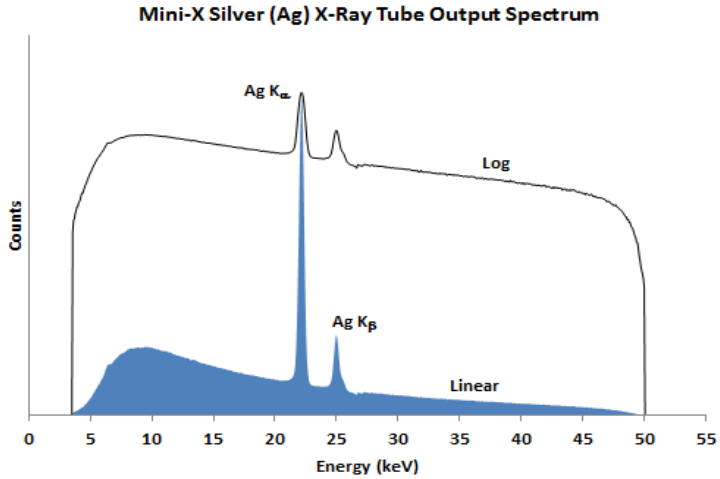


Figure 3.13: Mini-X2 Silver (Ag) X-Ray Output Spectrum at 50 kV, from [25].

use (111) and (110) of Silicon crystal's planes whose Bragg angles are 5.1 deg and 4.17 deg respectively according to [60]. To calculate the Bragg angles (θ_B) for silicon crystals with the (111) or (110) plane using a silver anode X-ray source the Bragg's Law can be used:

$$n\lambda = 2d_{hkl}\sin(\theta_B) \tag{3.7}$$

Where:

- n is the order of diffraction;
- λ is the X-ray wavelength. A silver anode typically emits K_α radiation with a wavelength of approximately 0.55×10^{-10} m;

- d_{hkl} is the interplanar spacing for a crystal plane;
- θ_B is the Bragg angle.

For silicon, the lattice constant a is 5.431×10^{-10} m. The interplanar spacing d_{hkl} for a cubic crystal is given by:

$$d_{hkl} = \frac{a}{\sqrt{h^2 + k^2 + l^2}}. \tag{3.8}$$

Using Equation 3.8 we can compute the interplanar spacing for the 110 plane ($h = 1$, $k = 1$, $l = 0$):

$$d_{110} = \frac{5.431 \times 10^{-10}\text{m}}{\sqrt{1^2 + 1^2 + 0^2}} = \frac{5.431 \times 10^{-10}\text{m}}{\sqrt{2}} \approx 3.840 \times 10^{-10}\text{m}. \tag{3.9}$$

While for the 111 plane ($h = 1$, $k = 1$, $l = 1$):

$$d_{110} = \frac{5.431 \times 10^{-10}\text{m}}{\sqrt{1^2 + 1^2 + 1^2}} = \frac{5.431 \times 10^{-10}\text{m}}{\sqrt{3}} \approx 3.13 \times 10^{-10}\text{m}. \tag{3.10}$$

Using Bragg's law (Equation 3.7) for first-order diffraction ($n = 1$):

$$\sin(\theta_B) = \frac{\lambda}{2d}. \tag{3.11}$$

Substituting the known values of interplanar spacing (Equation 3.9) and λ in 3.11, for (110) plane we have:

$$\sin(\theta_B) = \frac{0.5594 \times 10^{-10}\text{m}}{2 \times 3.840 \times 10^{-10}\text{m}} \approx 0.0728 \tag{3.12}$$

$$\theta_B = \arcsin(0.0728) \approx 4.17^\circ. \tag{3.13}$$

While substituting the interplanar spacing (computed in 3.10) and λ in 3.11, for (111) plane we have:

$$\sin(\theta_B) = \frac{0.5594 \times 10^{-10}\text{m}}{2 \times 3.13 \times 10^{-10}\text{m}} \approx 0.0893 \tag{3.14}$$

$$\theta_B = \arcsin(0.0893) \approx 5.10^\circ \tag{3.15}$$

3.2.2 Monochromator

A crystal monochromator is a key component in X-Ray diffraction. It is used to to select and isolate a specific wavelength of X-rays from a polychromatic beam. A monochromating crystal when rotated with respect to the incident beam it will diffract a spectral component in accordance with Bragg's Law. The diffracted beam can also be focused by a toroidal shape of the monochromator. The crystal used in the monochromator is chosen based on its lattice structure and spacing between the atoms, which determines the angle of diffraction for a particular wavelength. The crystal monochromator used in this work is cut and polished such that a particular atomic plane of the crystal is parallel to the surface of the crystal as Si(111) or Si(110). As the X-rays pass through the crystal, they are diffracted at specific angles, with different angles corresponding to different wavelengths in such a way that it only diffracts the $K\alpha$ radiation, and not the $K\beta$ radiation. By selecting the appropriate crystal and adjusting its position, a narrow range of wavelengths can be isolated and separated from the rest of the beam. A slit is then used to cut the monochromatic beam geometrically such that only a well-collimated beam with the desired spatial properties is used (Figure 3.14). Additionally, using a slit can also help to improve the resolution of the monochromator by restricting the angular and spatial spread of the beam. By limiting the size of the beam incident on the crystal, the slit helps to ensure that only a small region of the crystal is illuminated by the beam, which can improve the quality of the monochromatic beam that is produced.

3.2.3 Sample stage

In X-ray Diffraction, achieving precise alignment of the sample rotation axis is crucial in at least five degrees of freedom concerning the source, the monochromator, the slit, and detector. Additionally, high stiffness and stability are essential. For this purpose, a hexapod-type parallel kinematic machine (PKM) is highly suitable. It offers six degrees of freedom, allowing the selection of an arbitrary pivot point. Furthermore, compared to a serial kinematic stack of stages, the hexapod-type PKM exhibits superior stiffness and compactness. The hexapod used in this work is a Newport HXP50-MECA (Figure 3.15) [26] with a 125 mm diameter platform, 5 kg load capacity and sub-millimeter or radian precision depending of the motion. In order to accurately and uniquely define the position of the moving

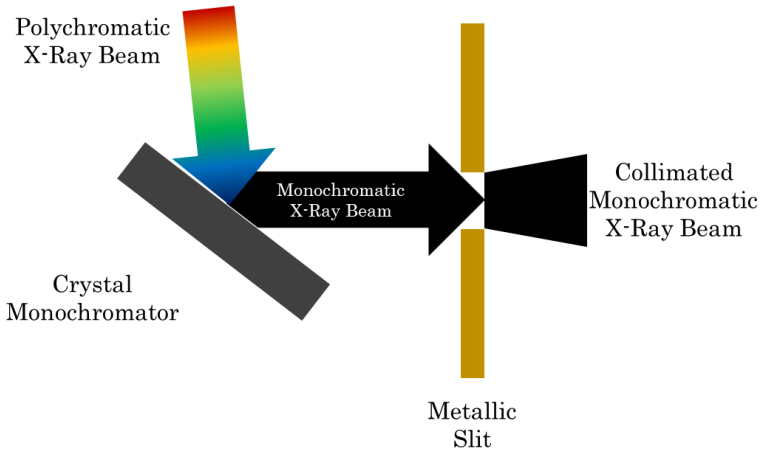


Figure 3.14: Illustration of the X-ray beam being diffracted by the crystal monochromator. Based on the angle of diffraction a particular wavelength can be selected. The metallic slit cut the beam geometrically. After passing the aperture the wave propagating tends to spread out.

platform in three-dimensional space, it is necessary to provide precise information regarding both its spatial location and angular orientation. This entails specifying three linear coordinates to determine its position in space and three rotational coordinates to define its orientation through angular rotations. For this reasons a Cartesian coordinate system is used to represent the position of the platform in the space and Bryant angles to represent the angular orientation.

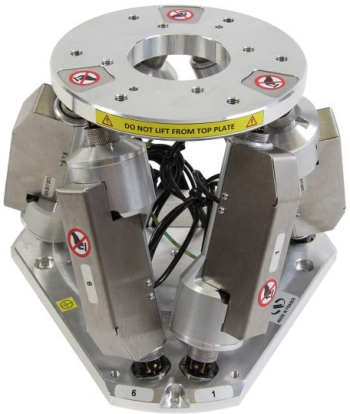


Figure 3.15: Newport HXP50-MECA with six degrees of freedom used in this work, from [26]

Figure 3.16 represents the coordinate set (X, Y, Z, U, V, W) of the hexapod. Where *forward* and *backward* describe movements along the Y-

axis, *left* and *right* denotes movements along the X-axis and *up* and *down* denotes movements along the Z axis. In addition, for the rotational axes V represents the angle of rotation around the X-Axis (*pitch*); U represents the angle of rotation around the Y-Axis (*roll*); and W represents the angle of rotation around the Z-Axis (*yaw*).

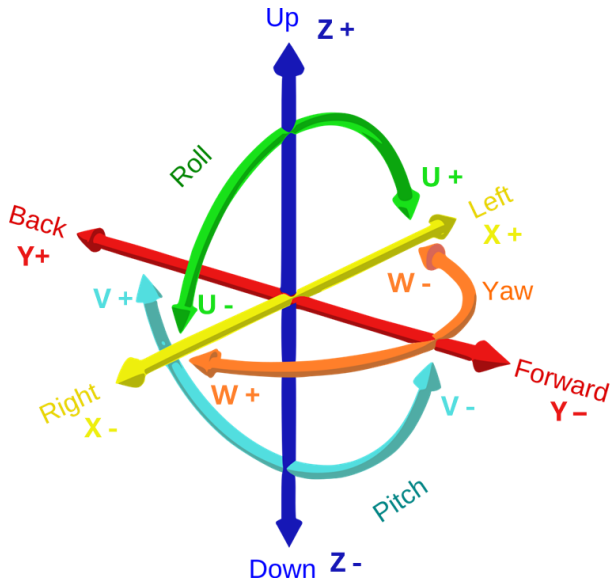


Figure 3.16: Coordinate system of the hexapod illustrating its six degrees of freedom. The system is defined by the coordinates (X, Y, Z, U, V, W), adapted from [27].

In order to determine uniquely the position of the hexapod in the 3-D space the *base*, *tool* and *work* coordinate systems are used. The base coordinate system is a fixed reference frame or point of origin for the robot. It serves as a stable starting point for all measurements and calculations. The work coordinate system defines a specific location or position in space where the robot needs to perform a task or work. It acts as a target or goal for the robot's movements. The tool coordinate system is associated with the end effector or tool of the robot, which is typically located at the end of each leg, on the platform. The base coordinate system provides a reference frame for the robot's position and orientation [26].

3.2.4 X-Ray detector

Solid-State X-ray Detectors (SDDs) have revolutionized X-ray detection and imaging technology, offering improved performance, enhanced sensitivity, and higher spatial resolution compared to traditional detectors. SDDs are semiconductor-based devices that convert X-ray energy into electrical

signals. In this work an Amptek solid state x-ray detector is used. The main components of an SDD include a collimator, which serves as a selective passage for X-rays from the specific area being excited by the electron beam while excluding stray X-rays; an electron trap, a permanent magnet assembly that deflects passing electrons to prevent background noise; a window, which maintains vacuum within the detector while remaining transparent to low-energy X-rays; a sensor, a semiconductor device that converts X-ray energy into an electric charge through ionization; and a FET amplifier, directly connected to the sensor, which amplifies the charge released due to incident X-rays into a voltage output.

3.2.5 X-ray sensor

This study utilizes a FAST SDD® Ultra High Performance Silicon Drift Detector [61], in conjunction with the Digital Pulse Processor PX5, both manufactured by Amptek [62]. The silicon drift detector (SDD) sensor is fabricated from high purity silicon with a large area contact on the entrance side facing the incoming X-rays. On the opposite side there is a central, small anode contact, which is surrounded by a number of concentric drift electrodes (Figure 3.17);

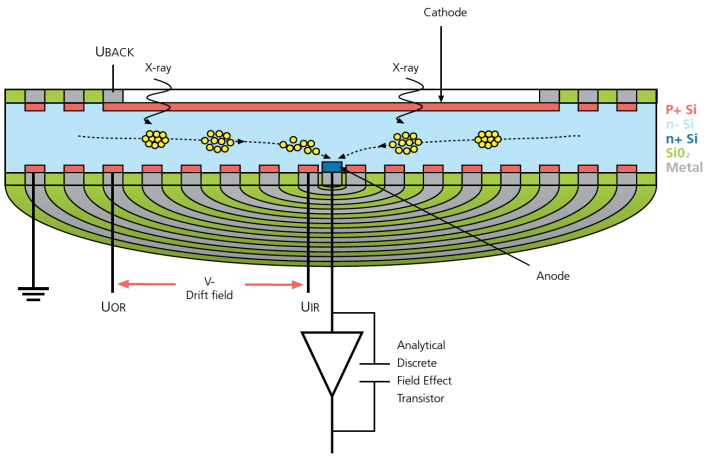


Figure 3.17: Construction and Operation of an SDD detector, from [28].

When the energy of a particle or radiation interacting with a material is increased, it can provide sufficient energy to remove electrons from the atoms or molecules in the substrate, resulting in a higher degree of ionization. Applying a bias to the SDD detector chip and exposing it to X-rays results in the conversion of each detected X-ray into an electron cloud. The

charge of this electron cloud is directly proportional to the characteristic energy of the X-ray. The higher is the energy the greater is the degree of ionization that occurs in the substrate. These electrons are elevated to the conduction band of the silicon semiconductor, leaving behind holes that act as freely moving positive charges within the sensor. By applying a field gradient between the drift rings, the electrons are guided or 'drifted' towards the anode, where they are collected [28].

FET amplifier

The charge that accumulates at the anode is converted to a voltage signal by the FET pre amplifier. The circuit (Figure 3.18) is composed of a mosfet transistor whose gate is connected to the anode of the substrate, while the drain is connected to an operational amplifier and to a coupling capacitor. During operation, charge is built up on the latter. The output of the pre amplifier shown in Figure 3.19 is a voltage ramp over time with shorts capacitor discharges to prevent saturation of the circuit. The preamplifier output is sampled continuously by an analogue to digital converter (ADC) and X-ray pulse heights are typically measured by subtracting the average of one set of values, measured before an X-ray event, from that for another set, measured after the event. The resultant value of the step measurement will be count rate of the X-rays [28].

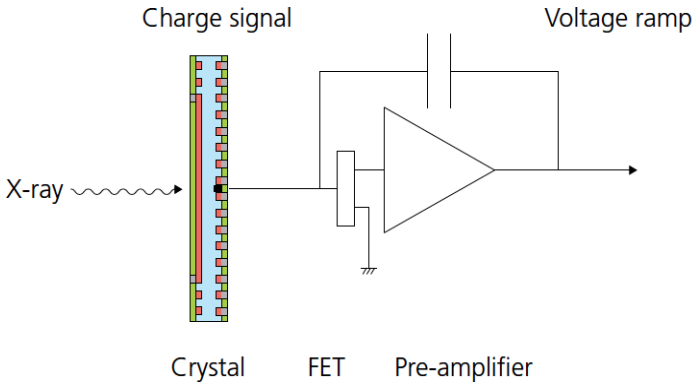


Figure 3.18: Schematic of SDD connected to FET pre-amplifier and feedback capacitor, from [28].

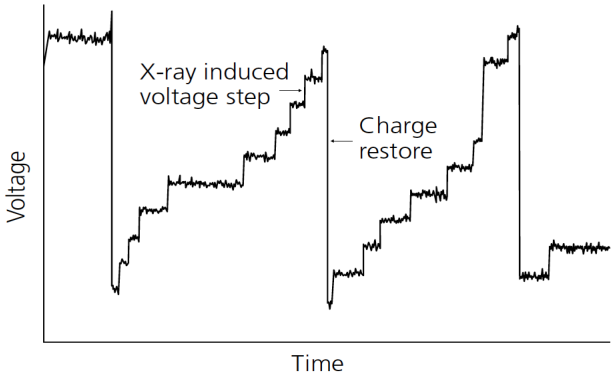


Figure 3.19: Pre-amplifier output voltage pulse, from [28].

3.3 X-Ray diffractomer software

This section provides a comprehensive account of the innovative and novel C++ software designed for managing the X-ray diffractometer. The principal objective of this software is to offer an efficient mechanism for controlling all the instruments within the measurement setup and to establish a post-processing unit that can optimize the alignments and measurements described in Section 3.4.

The software development process adopted a modular approach, where each module performs a specific activity. Furthermore, the code structure follows a bottom-up approach, as depicted in Figure 3.20, in which the lower modules are responsible for controlling the stepper motors and hexapod robot. The intermediate module is used for finite-state machines control. The upper modules are utilized for post-processing analysis and graphic user interface. In order to allow greater flexibility and extensibility of the code, interface classes[1] [63] have been used. Interfaces allow the abstraction of the implementation details of a class and provide a higher-level view of its functionality. This allows the use of mock objects [64] for testing purposes without affecting the client code.

In addition, interfaces facilitated modularity by separating the definition of behavior from its implementation. This promoted code organization, maintainability, extensibility, testability and reusability that are the main pillars of this code structure. Another key feature used in developing this framework is the use of shared pointers [65]. In C++, shared pointers are smart pointers that manage the lifetime of dynamically allocated objects

[1] An interface is a class that lacks an implementation. Its sole purpose is to serve as a base for other classes to inherit its methods, fields, and properties. By introducing an interface class, a level of abstraction is established that defines the object's behavior before its actual instantiation.

through reference counting. They automatically deallocate the object when the last shared pointer owning it is destroyed, helping to prevent memory leaks by ensuring proper deletion when the object is no longer needed.

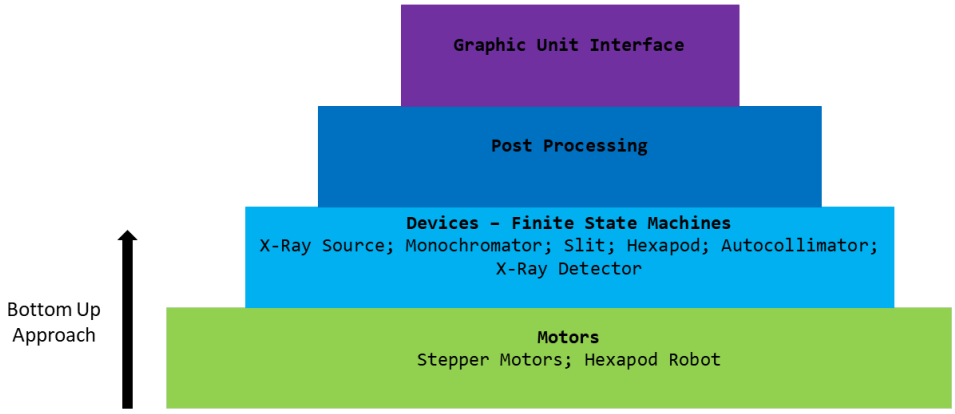


Figure 3.20: Illustration of the bottom-up approach utilized for developing the code modules for X-Ray Diffractometer control. Lower-level modules handle device control, while higher-level modules manage the integration of post-processing algorithms and graphical user interface control.

The system architecture depicted in Figure 3.21 is modularized into several key components, each fulfilling distinct roles to ensure effective operation and integration.

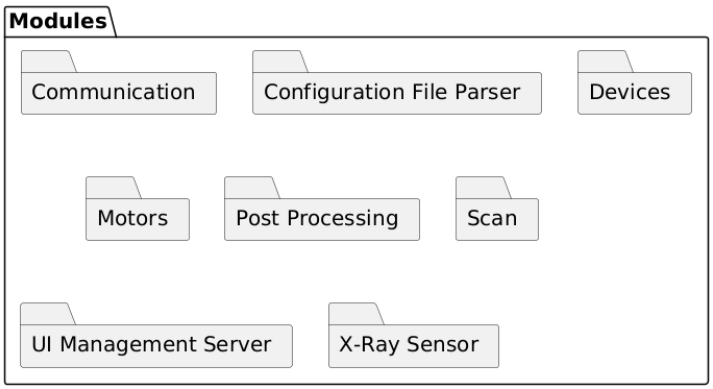


Figure 3.21: Organization of system modules illustrating the Communication, Configuration File Parser, Devices, Motors, Post Processing, Scan, UI Management Server, and X-Ray Sensor modules. Each module is represented as a distinct folder to denote its functional grouping and role within the system architecture.

3.3.1 Low-level control module

The low-level control module, named "Motors", is organized into two main class interfaces, *IMotor* and *IHXP*:

1. *IMotors*: The class interface "IMotors" (Figure 3.22) acts as an interface for interacting with stepper motors and defines functions for calibrating the motors, move the devices to specific positions, stop the motion ecc..

2. *IHXP*: The class interface "IHXP" (Figure 3.23) contains a set of pure virtual functions that define the required methods for controlling the hexapod robot. These methods include setting the IP address and port number of the hexapod controller, establishing a connection, moving to the home position, disconnecting from the hexapod, setting the position manually or incrementally, retrieving the current position, stopping the movement, and manipulating the coordinates of the hexapod. The code section concludes with virtual getter and setter methods for accessing and manipulating the individual coordinates of the hexapod. These methods allow for fine-grained control over the position of the hexapod in each axis.

The file structure of the Motors module is depicted in Figure 3.24, showing two primary folders: include and src. The include folder contains the header files that define the class interfaces and their implementations, while the src folder includes the source files that implement the behavior of the motors.

Overall these interfaces provide a standardized set of functions for controlling stepper motors and the hexapod robot used in the custom build X-ray diffractometer allowing for different implementations to be created based on specific motor controllers or configurations.

3.3.2 Configuration module

The Configuration File Manipulation module is designed to facilitate the manipulation of configuration files in the ".ini" format. These files serve as essential repositories for storing device configurations, including critical parameters such as IP addresses, motor IDs, and devices alignment positions. Additionally, the configuration files contain information required for conducting scans[2] to align devices accurately or measure crystals under

[2]Motion of a device over a specified range in steps.

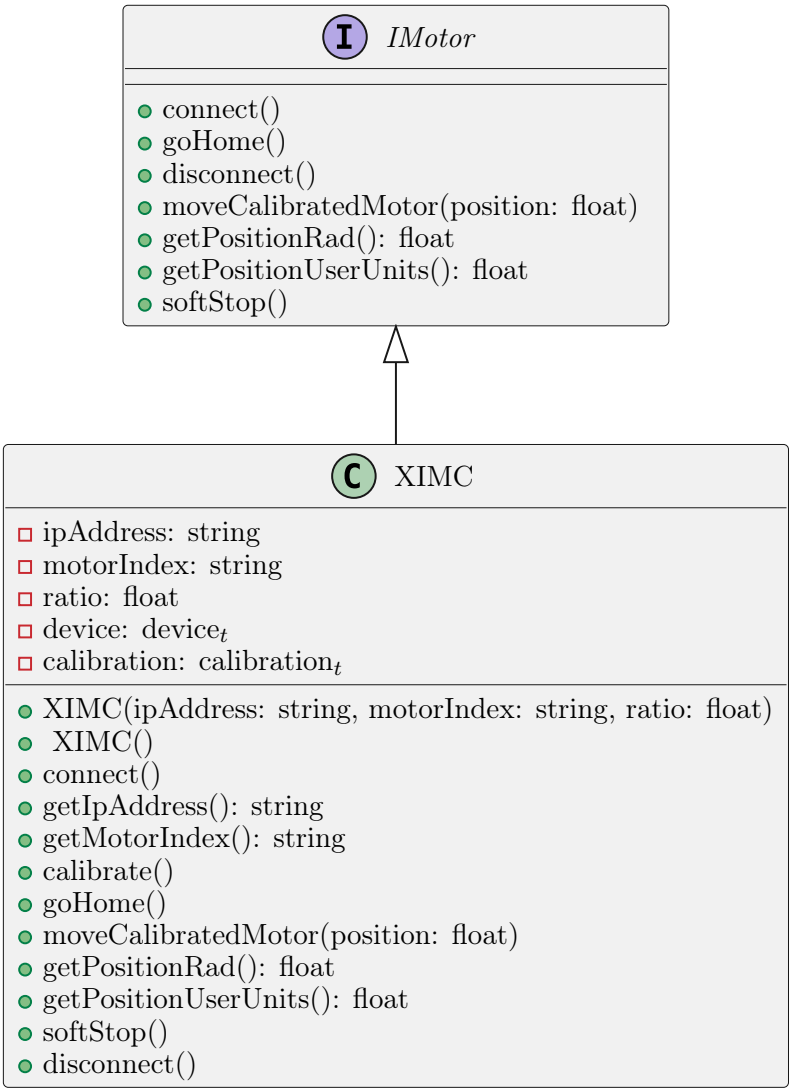


Figure 3.22: Simplified Unified Modeling Language (UML) diagram showing the IMotor interface and its implementation in the XIMC class, representing a motor controller.

test. To enable efficient handling of these configuration files, the module leverages the "mINI" external library [66].

As illustrated in Figure 3.26, the module is organized into two main directories: include and src. The include directory houses the header file that define the class interface and implementation, whereas the src directory contains the source files that implement the functionality of these classes. The project directory contains also two key configuration files. The first file serves as a repository for both device configurations (e.g., IP addresses, stepper motor IDs) and alignment configurations (i.e., computed alignment positions and measured crystal parameters). The second configuration file contains comprehensive information essential for conducting scans to align

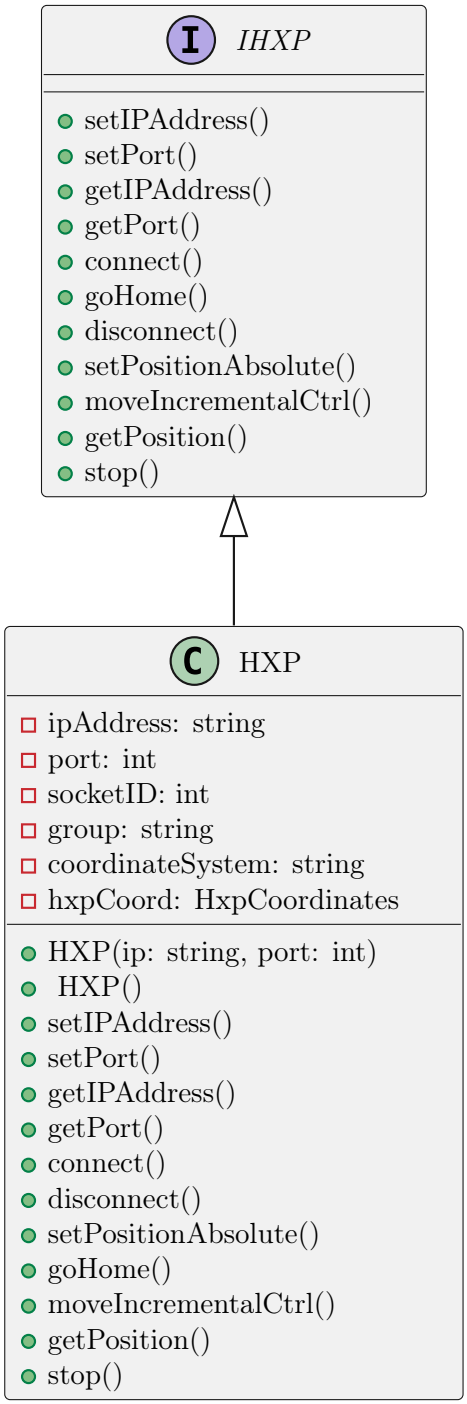


Figure 3.23: Simplified Unified Modeling Language (UML) diagram showing the IHXP interface and its corresponding implementation in the HXP class, representing a controller for Hexapod devices. IHXP interface defines basic control methods, while the HXP class provides the implementation for communication with Hexapod devices.

devices and measure crystals accurately.

The core functionalities of the Configuration module are encapsulated in the abstract interface class "*IConfiguration*" (Figure 3.25). This class

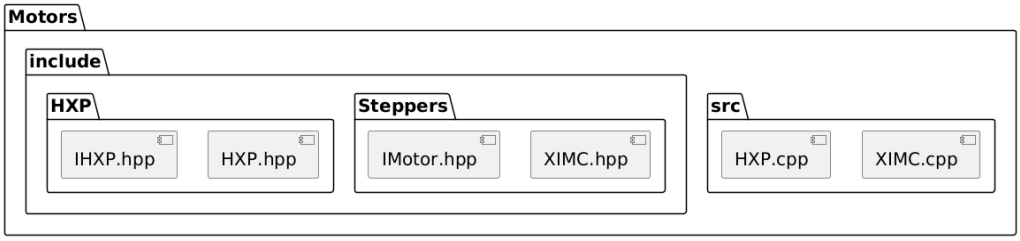


Figure 3.24: Folder organization of the Motors module. The include folder contains the header files, with separate subfolders for HXP (hexapod control) and Steppers (stepper motor control). The src folder holds the corresponding implementation files. The HXP folder includes IHXP.hpp and HXP.hpp, which define and implement the interface for hexapod control, while the Steppers folder contains IMotor.hpp and XIMC.hpp, providing the interface and implementation for stepper motor control. The src folder contains the source files HXP.cpp and XIMC.cpp that implement the functionality defined in the header files.

defines a set of pure virtual functions, establishing a contract that concrete implementations of the module must adhere to. The main functionalities of the interface are: creation of empty configuration files with specified filenames in designated paths, writing string values to specific keys within designated sections of a configuration file, determine the existence of a configuration file with a given filename in a specified path, check for the presence of a specific section within a configuration file, determine the existence of a configuration file with a given filename in a specified path and capability to read string, integers and boolean values from the configuration file.

The presented C++ module serves a dual purpose in the context of this work. Firstly, it facilitates the organization of vital parameters related to the device by storing them in separate configuration files, such as '.ini' files. Secondly, the module optimizes the compilation process by enabling direct reading of values from the configuration files. This approach eliminates the need for recompilation when configuration parameters are modified, as the updated values are dynamically retrieved from the files during runtime. This dynamic reading mechanism reduces compilation times significantly, streamlining the development workflow. Consequently, the module enhances the maintainability and customizability of the codebase, providing a more efficient and organized framework for handling parameters of the diffractometer.

3.3.3 Communication module

This section presents the implementation of a Transmission Control Protocol (TCP) communication module utilizing the Asio C++ library [67]. The

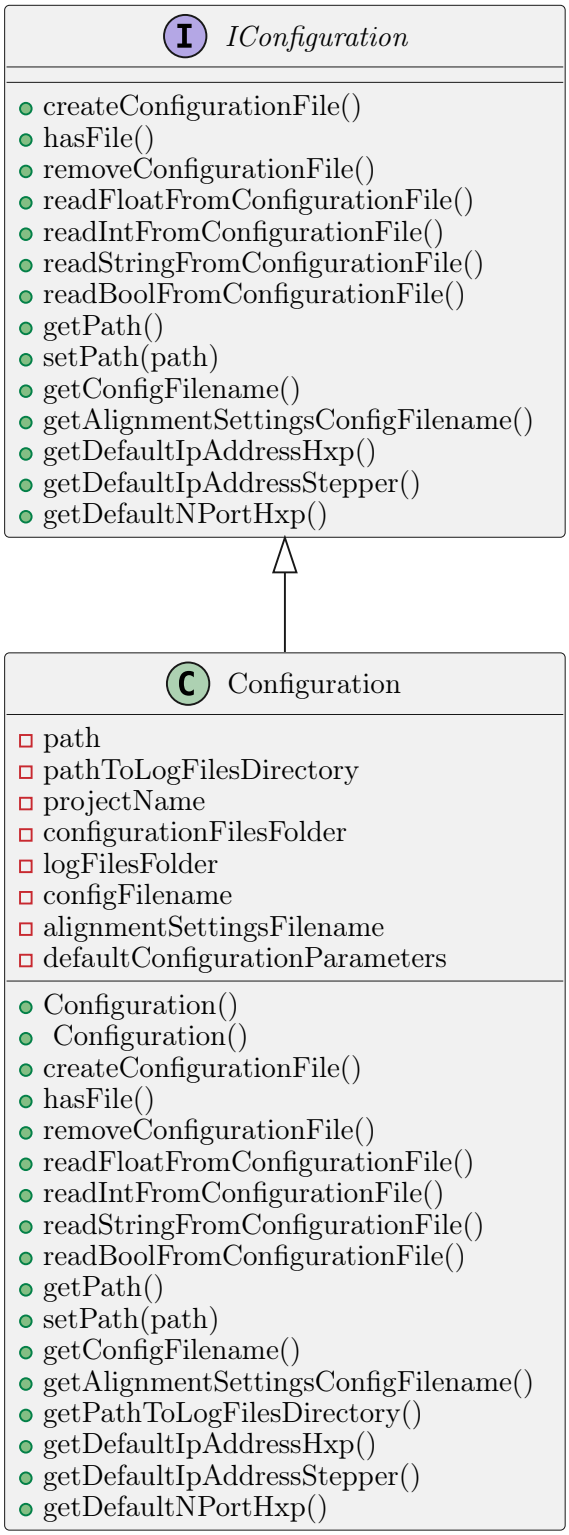


Figure 3.25: Overview of the IConfiguration interface and its implementation in the Configuration class, designed for managing configuration files. IConfiguration defines methods for file manipulation and retrieval of configuration settings, while the Configuration class provides concrete implementations and storage for configuration-related data.

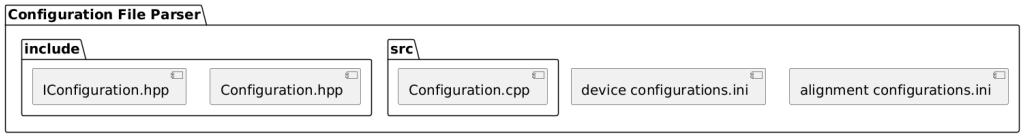


Figure 3.26: The diagram illustrates the structure of the Configuration File Parser module. It is divided into two main directories: include and src. The include directory contains header files defining class interface, while the src directory includes the source file that implements the class behavior. Additionally, the module handles configuration files essential for storing device settings and alignment parameters.

server module facilitates bidirectional communication over TCP sockets and enables the establishment of connections with multiple clients. The module comprises two essential interfaces, namely *ISocket* and *ISocketServer*, which serve as the foundation for managing TCP socket control and server operations, respectively.

- *ISocket*: The main functions defined in the ISocket class include the following: the ‘open‘ method, which is used to open a TCP socket and takes two function objects, ‘messageHandler‘ and ‘errorHandler‘, as parameters to handle incoming messages and errors; the ‘close‘ method, responsible for closing the TCP socket and terminating the connection when called; the ‘write‘ method, which allows data to be sent over the socket; and the ‘read‘ method, used to receive incoming messages by reading data from the socket.

- *ISocketServer*: The main functions defined in the ISocketServer class include the following: the ‘open‘ method, used to open the server and start listening for incoming connections; the ‘close‘ method, responsible for shutting down the server and terminating active connections; and the ‘acceptConnection‘ method, which handles accepting new TCP connections from clients. This method is typically called recursively after each new connection is accepted and includes functionality to manage events such as receiving messages from clients.

As shown in Figure 3.27, the Communication module is organized into two primary directories: include and src. The include directory contains header files for TCP communication (TCPServer.hpp, TCPSocket.hpp) and general socket interfaces (ISocket.hpp, ISocketServer.hpp). The src directory includes the corresponding implementation files (TCPServer.cpp, TCPSocket.cpp) that define the behavior of the TCP communication classes. Overall, this TCP communication server module, based on the provided

interfaces, allows bidirectional communication with multiple clients. The module's design allows for easy integration with the GUI, enabling the GUI to send and receive JSON files to and from the server, thereby achieving effective communication between the GUI and the server module.

Figure 3.27: The diagram depicts the organization of the Communication module. It is divided into two main directories: include and src. The include directory contains header files for TCP communication, as well as general socket interfaces. The src directory includes the corresponding source files that implement the functionality of the TCP communication classes.

3.3.4 Scanning module

The scan operation of every instrument of the diffractometer is a critical aspect of the X-ray sensor data acquisition process. The scan procedure involves initiating the stepper motor's (or hexapod robot axis) motion from a defined starting position to a predetermined stop position. The scan is carried out in incremental steps, each determined by the predefined step size. The selection of an appropriate step size is crucial, as it influences the resolution and accuracy of the scan. Larger step sizes may lead to faster scans, but at the expense of reduced positional accuracy. The range of the scan is defined as the total displacement the stepper motor must cover during the scanning process. It encompasses the distance between the starting and stop positions. To facilitate the control and the execution of the scans by every diffractometer device, the "IScanning" interface (Figure 3.28) has been developed. This interface provides a set of methods and parameters to initiate, configure, and monitor the scanning process. The core functionality of the "IScanning" interface includes the initiation of scans and the management of scan parameters.

The module structure depicted in Figure 3.29 organizes the source files and headers that implement the scanning functionality of the diffractometer. The Scan folder contains two subfolders: include and src. The include folder holds the header files, which define the interfaces and class declarations necessary for controlling the scan process. The src folder contains the corresponding source files that implement the methods declared in the

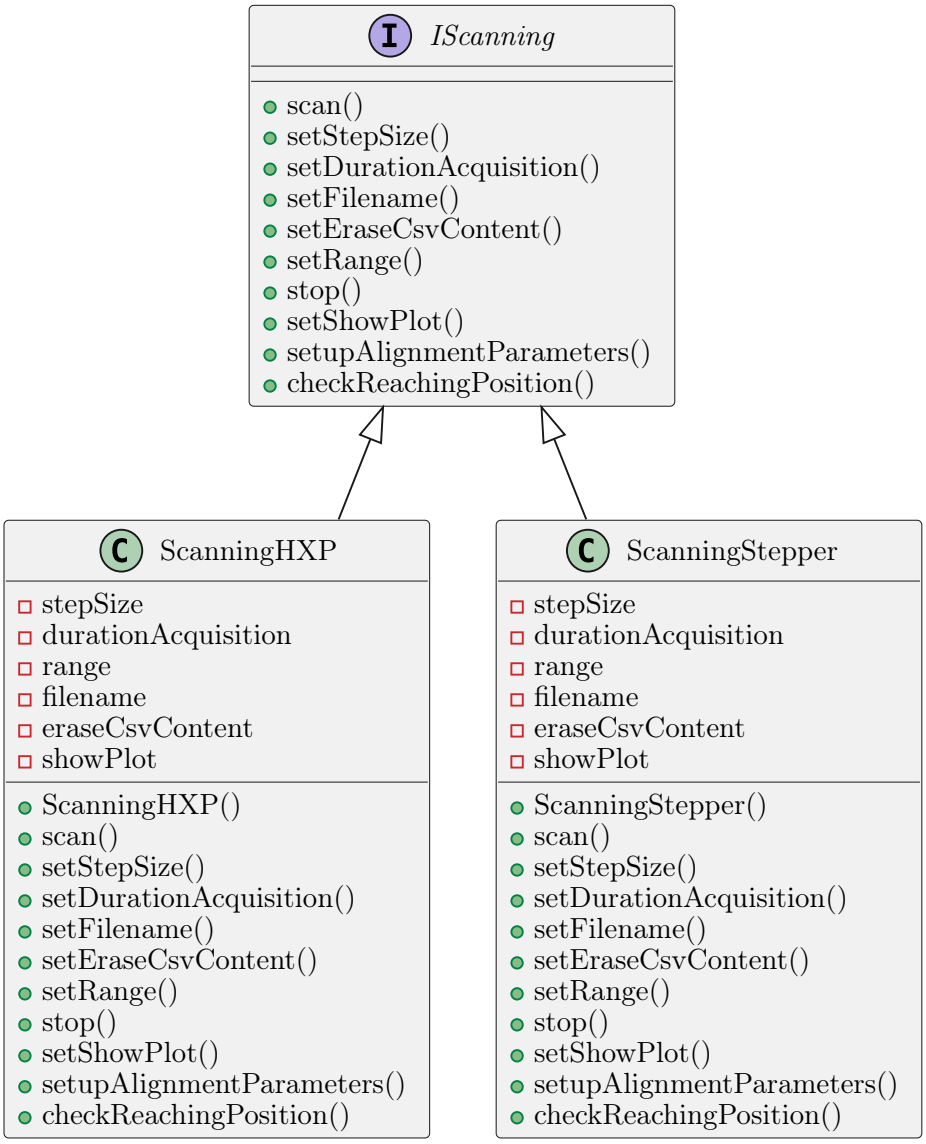


Figure 3.28: Simplified Unified Modeling Language (UML) diagram showing the IScanning interface and its implementations: ScanningHXP and ScanningStepper. The IScanning interface defines methods for scanning operations, while the ScanningHXP and ScanningStepper classes provide implementations tailored for the Hexapod and stepper motor devices, respectively.

headers.

3.3.5 X-ray sensor module

The X-ray sensor module's interface represents a crucial abstraction for managing communication with X-ray sensors. It establishes a standardized set of operations necessary for interacting with such sensors, ensuring consistency and reliability in sensor management and data acquisition. The 'XRaySensor' class, which implements the 'IXRaySensor' interface (Figure

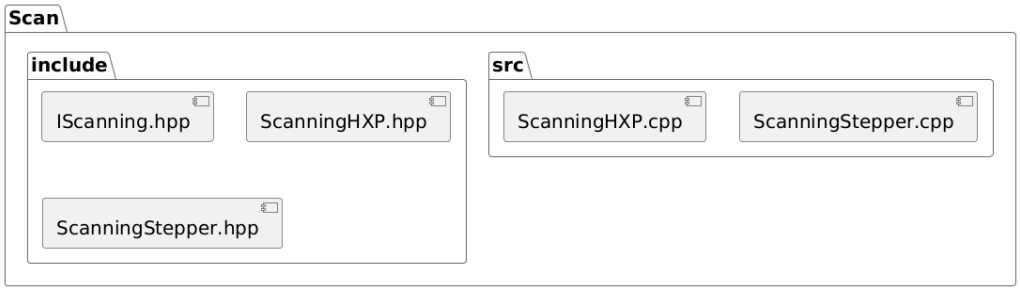


Figure 3.29: The Scan module is divided into two main directories: include and src. The include directory contains the header file of the interface as well as the implementations. The src directory includes the corresponding source files that implement the functionality of the classes.

3.30), adheres to this contract and provides concrete implementations of these methods. This design ensures that any interaction with X-ray sensors through the XRaySensor class is standardized, facilitating reliable and consistent sensor communication and data handling.

The module structure organizes the source files and headers related to the X-ray sensor functionality (3.31). The X-ray Sensor folder contains two subfolders: include and src. The include folder contains the header files, which define the interface and class for handling X-ray sensor operations. The src folder contains the corresponding source file.

The interface specifies operations for establishing and terminating connections with an X-ray sensor. It provides a mechanism to attempt a connection using default device settings and returns a boolean value to indicate the success or failure of this operation. Conversely, it includes functionality to disconnect from the sensor, ensuring proper resource release and maintaining system stability. An important feature of this interface is its capability to query and interpret the sensor's status. It includes a method for sending a status request to the sensor, processing the received data, and updating internal state indicators based on the status response. This functionality is essential for verifying the operational state of the sensor and for diagnosing potential issues. In terms of data acquisition, the interface supports gathering various forms of radiation data. It includes methods for measuring specific radiation counts related to K-alpha and K-beta radiation, requiring a time parameter to define the duration of the measurement. These methods return integer counts representing the detected levels of these specific types of radiation, which are crucial for applications requiring precise measurements of these radiation types. Additionally, the interface provides functionality to acquire a comprehensive spectrum of radiation counts. This capability involves configuring the sensor hardware,

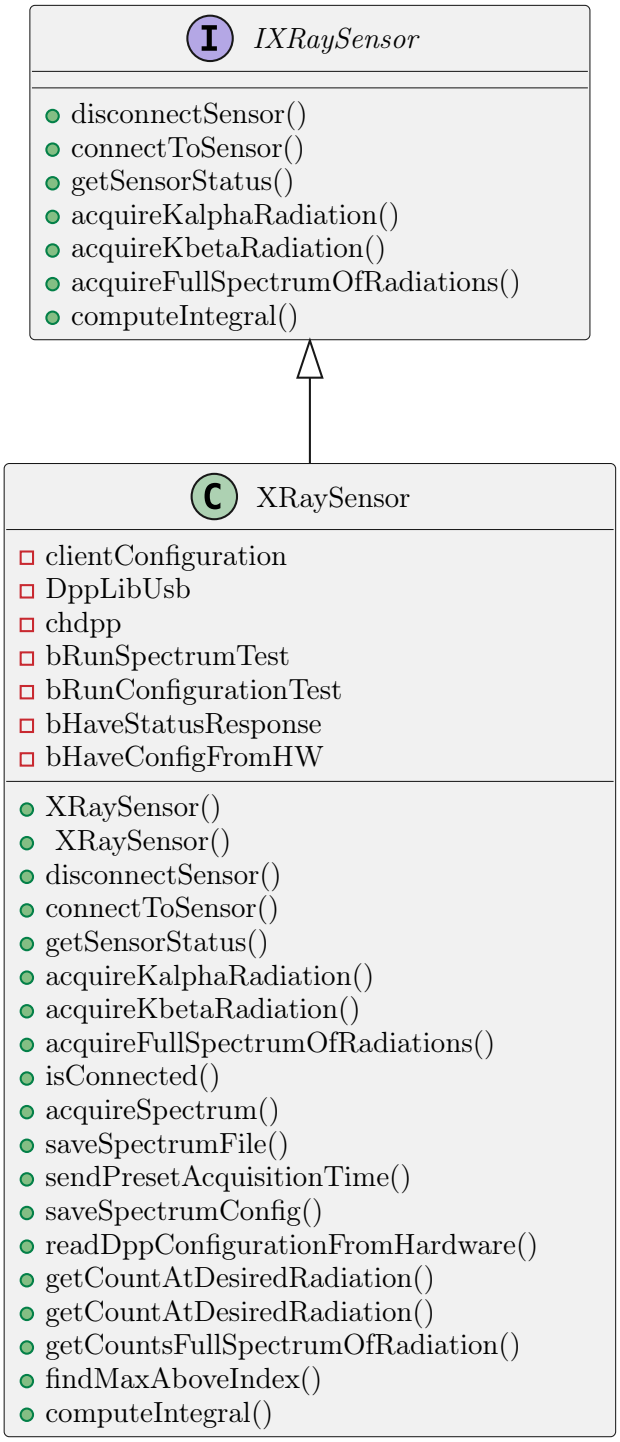


Figure 3.30: Simplified Unified Modeling Language (UML) diagram showing the IXRaySensor interface and its implementation XRaySensor.

performing data acquisition over a specified duration, and returning a vector of integers that represents the full spectrum of detected radiation. Such detailed data is essential for thorough spectral analysis and for applications needing a complete radiation profile.

Finally, the interface includes a method for computing the integral of a

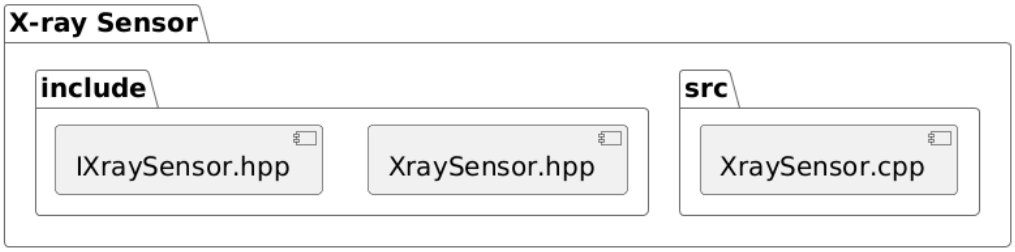


Figure 3.31: Module organization for the X-ray sensor functionality, illustrating the division between header files in the include directory and the source file in the src directory.

specified range within the spectrum data set. This functionality allows for the quantitative assessment of k_α or k_β radiations by calculating the total radiation within a defined range, based on specified start and stop indices within the spectrum data vector.

3.3.6 High-level control module

The higher-level module *"Devices"* is designed to facilitate control over stepper motor axes and a hexapod robot, providing an abstraction layer that enables high-level commands and behavior. The implementation incorporates the use of finite state machines[3] (FSMs), which are constructed and managed using the Boost State Machine Language (SML) [68] library. The latter is an extension to the Boost C++ Libraries [69] that provides a powerful framework for implementing FSMs in a concise and expressive manner. As shown in Figure 3.32 the Devices module organizes the headers and source files for various device controllers within the diffractometer system. The structure is divided into include and src subfolders, each responsible for specific components. The include folder contains headers for general device controllers. Within the include folder, subfolders organize headers for specific device categories. The src folder is intended for the implementation of these device controllers.

The class interfaces that this module contains are:

- *ISingleStepperDeviceController*: The interface (Figure 3.33) represents an abstraction for controlling devices that are moved by a single stepper motor (x-ray source, autocollimator, x-Ray detector) encapsu-

[3]Computational models used to represent systems with discrete states and transitions. FSMs provide a structured approach to modeling and controlling the behavior of systems that exhibit different states and undergo state transitions based on external events or conditions.

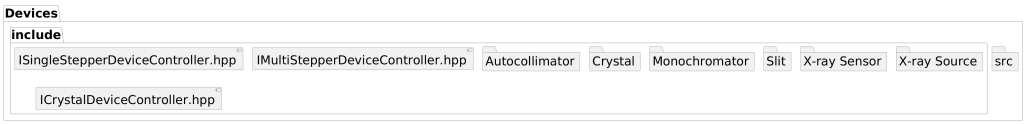


Figure 3.32: Device module folder organization, showing interfaces headers for general controllers and specific devices, with subfolders dedicated to each device type.

lating the common functionality and operations that can be performed on these devices.

- *IMultiStepperDeviceController*: The interface (Figure 3.34) represents an abstraction for controlling devices that are moved by two stepper motors (crystal monochromator and slit) and serves as a blueprint for classes that control multi-stepper devices and manage their FSM-based behavior.

- *ICrystalDeviceController*: The interface (Figure 3.35) represents a control interface for the devices allowed to control the pose of the crystal in the diffractometer (stepper motor and a hexapod robot).

The purpose of these interfaces is to provide a contract for implementing classes that can control the devices and manage their FSM behavior. One of the main functionalities of these classes is to automatize the motion of the devices during tasks as the beam alignment, crystal alignment or crystal measurements.

Finite-state machine design

In an FSM, the system's behavior is defined by a finite number of states, each representing a distinct condition or mode of operation. The transitions between states are triggered by events or specific conditions, leading the system from one state to another. FSMs are particularly useful for modeling systems with well-defined and predictable behavior, such as game logic, user interfaces and control systems. A basic FSM consists of the following components:

- States: The states represent the distinct conditions or modes of operation that the system can be in. Each state defines a specific set of behaviors or actions that occur when the system is in that state. The system can be in only one state at a time.

- Transitions: Transitions define the conditions under which the system moves from one state to another. Transitions are triggered by events

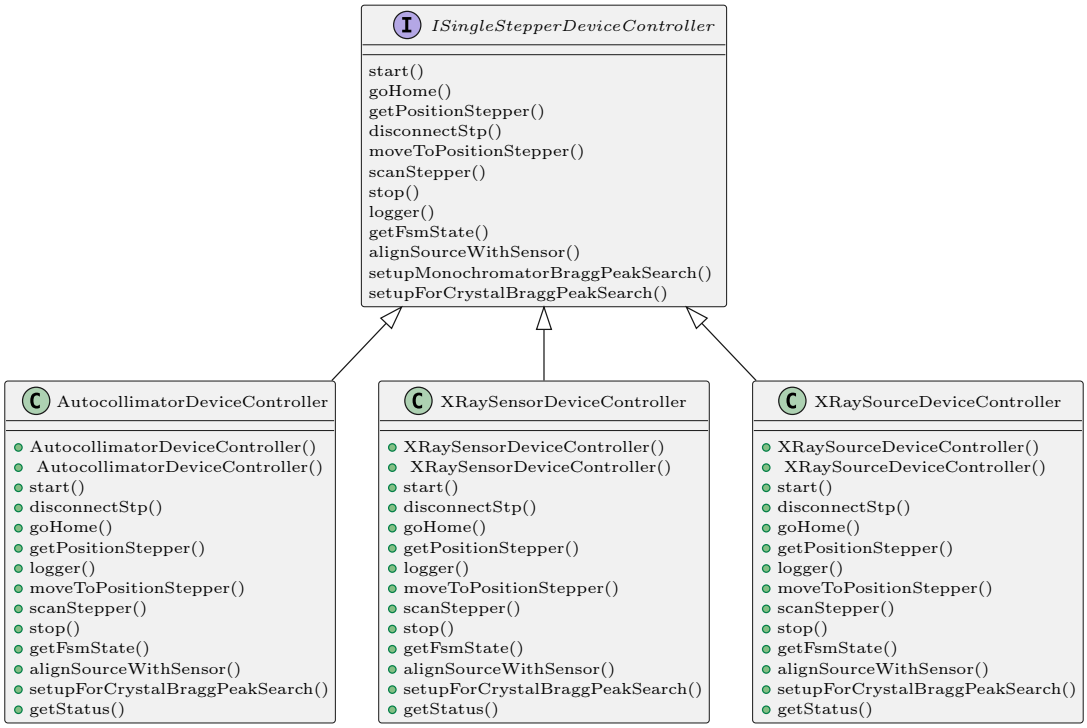


Figure 3.33: Unified Modeling Language (UML) diagram illustrating the ISingleStepperDeviceController interface and its implementations in the AutocollimatorDeviceController, XRaySensorDeviceController, and XRaySourceDeviceController classes. The interface outlines methods for controlling single-stepper devices, while each class provides specific implementations tailored for autocollimators, X-ray sensors, and X-ray sources, including alignment and scanning functionalities.

or conditions that occur within the system or externally. When a transition occurs, the system undergoes a change in state and performs any associated actions or behaviors defined for that transition.

- Events: Events are the triggers that cause state transitions to occur. They represent changes or occurrences that affect the system's behavior. In the FSMs developed in this work the events are always triggered by users.

- Actions: Actions are the behaviors or operations associated with state transitions. When a transition occurs, specific actions may be executed, such as updating variables, performing calculations, sending messages, or invoking functions. Actions define the behavior of the system during state transitions.

FSMs provide several benefits in software development as: Modularity, enabling the decomposition of complex systems into simpler states, making it easier to understand and manage system behavior; Readability and Main-

Figure 3.34: Unified Modeling Language (UML) diagram presenting the IMultiStepperDeviceController interface and its implementations in the MonochromatorDeviceController and SlitDeviceController classes. The interface outlines methods for controlling multi-stepper devices, while the classes provide specific implementations for controlling monochromator and slit, including alignment and scanning functionalities.

tainability, the structured nature of FSMs promotes code readability and simplifies maintenance and debugging, as the system's behavior is explicitly defined in states and transitions; Scalability, FSMs can be expanded

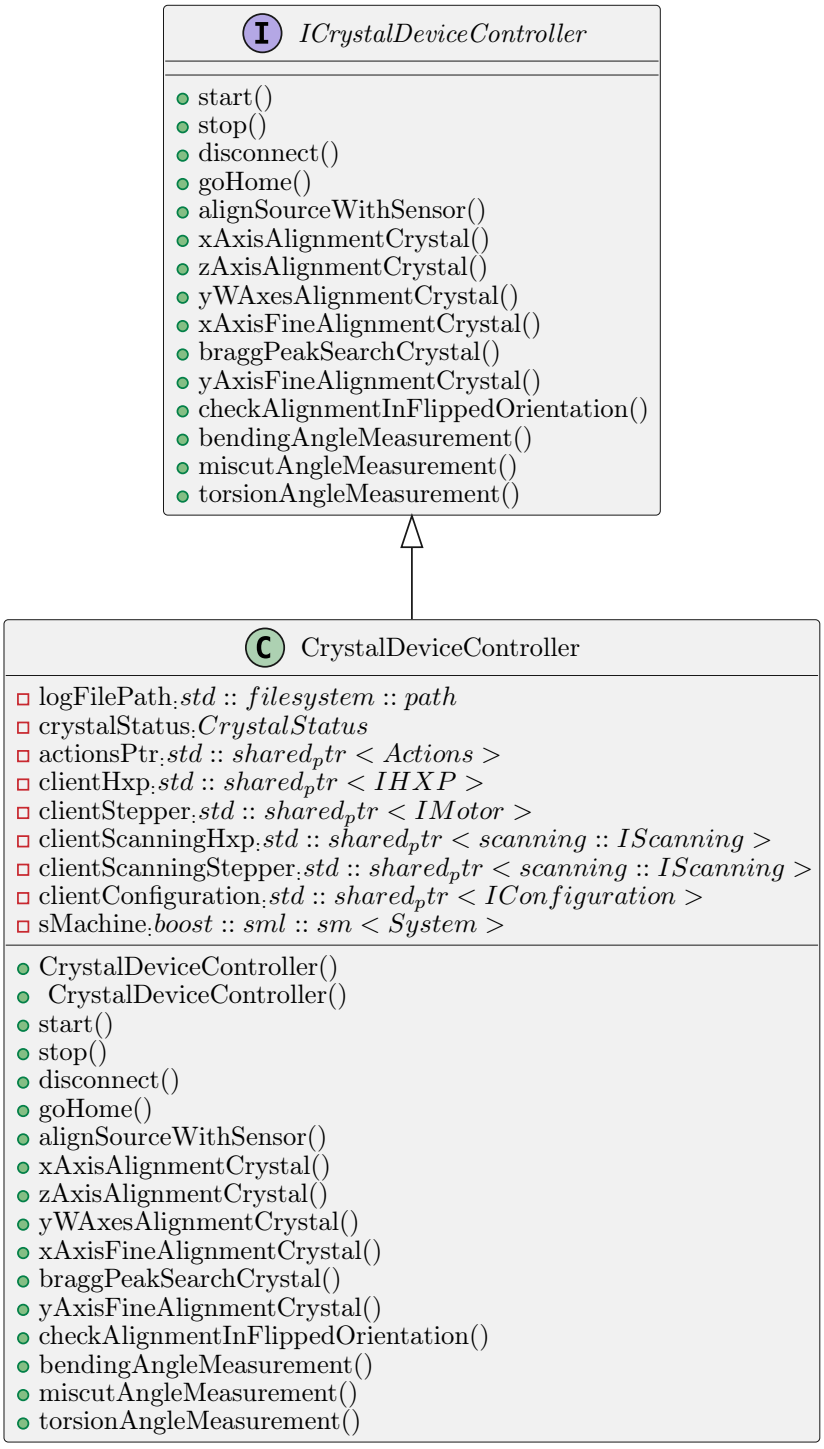


Figure 3.35: Unified Modeling Language (UML) diagram illustrating the ICrystalDeviceController interface and its implementation in the CrystalDeviceController class, designed for controlling crystal devices. The interface defines various methods for crystal device manipulation and control, while the CrystalDeviceController class provides the concrete implementation, including interaction with hardware components and sensor alignment.

by adding new states and transitions to accommodate additional system

behavior or features; Flexibility, FSMs allow for the specification of guard conditions on transitions, enabling fine-grained control over state transitions based on specific criteria.

The basic structure of the FSM developed to control each device of the diffractometer is depicted in Figure 3.36. It consists of five states: "SystemNotInitialized", "SystemConnected", "SystemInMotion", "SystemHome", and "SystemError". The FSM transitions between these states based on specific events and actions. Events trigger state transitions, and actions are performed when these events occur in a particular state. In the "SystemNotInitialized" state, the system's motors are not connected. The FSM transitions to the "SystemConnected" state upon connecting the motors. Once in the "SystemConnected" state, the system is operational with motors connected. The FSM moves to the "SystemInMotion" state when setup actions such as "Setup Home", "Move Device", or "Start Scan" are performed. If the motors are disconnected, the FSM returns to the "SystemNotInitialized" state. In the "SystemInMotion" state, the system is actively scanning or moving. The FSM transitions to the "SystemHome" state when the motion is completed. If the scan finishes, the FSM returns to the "SystemConnected" state after completing the motion. If an error occurs, it moves to the "SystemError" state. If the system is disconnected during motion, it returns to the "SystemNotInitialized" state. In the "SystemHome" state, the motors are in their home position. The FSM transitions to the "SystemInMotion" state to continue moving the device or returns to the "SystemNotInitialized" state if disconnected. In the "SystemError" state, an error is indicated. The FSM remains in this state if the error persists or moves to the "SystemNotInitialized" state upon system disconnection.

Six finite state machines have been developed, each corresponding to a specific device in the diffractometer. These FSMs are constructed based on the previously mentioned structure to effectively manage the device states. While the number of states remains consistent across all FSMs, variations occur in the specific events and actions that trigger state transitions for each device. A detailed description of a FSM can be found in Appendix A.

Integration with the low-level control module - "Actions"

The integration of finite state machines with the low-level control module ("Motors") is achieved using the "Actions" classes. This integration defines the behavior of each diffractometer device in response to specific events during FSM state transitions. The code implementation involves

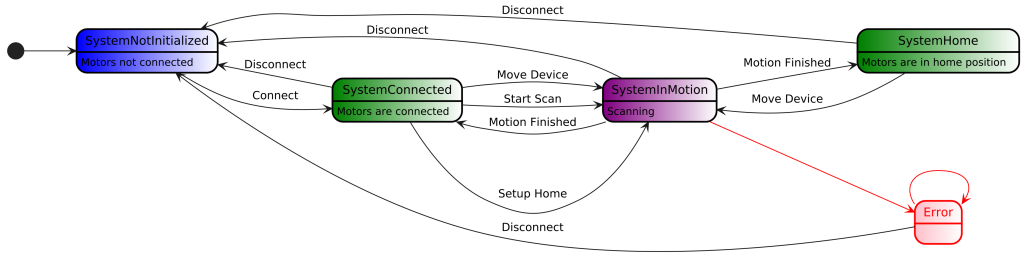


Figure 3.36: The simplified version of the finite state machine engineered. It's composed of five states and events that are the triggers that cause state transitions to occur.

passing five shared pointers to the class constructor as input parameters: a shared pointer to the "IMotor" class, which controls device movement; a shared pointer to the "IScanning" class, responsible for scanning operations; a shared pointer to the "ISensors" class, used to interact with the X-ray sensor; a shared pointer to the "IConfiguration" class, which handles reading and writing configuration parameters such as IP addresses and scanning parameters; and a shared pointer to the "IPostProcessing" class, which handles post-processing and data analysis from scanning operations.

The "Actions" classes provide functionality for connecting and disconnecting the devices, performing homing and movement operations, starting and stopping scans, and executing alignment procedures. These actions are performed using the provided shared pointers to interact with the corresponding modules and components of the device. The "Actions" class serves as a bridge between the finite state machine and the low-level control modules. It allows for precise control and coordination of the devices movements and scanning operations based on specific events during state transitions. This integration is crucial for achieving accurate and reliable functioning of the devices in various scenarios. Allowing the automation of the beam alignment, crystal alignment and crystal measurement procedures. The implementation provides a robust foundation for further development and utilization of the system.

3.3.7 Post-processing unit

The post-processing unit introduces a comprehensive Python module designed for the post-processing computations of beam alignment, crystal alignment and measurements. The module encompasses five subdirectories, each focusing on specific aspects related to the diffractometer device:

- Crystal: contains all the scripts needed to compute the crystal align-

ment positions along the X, Y, Z and W axes. In addition it contains the scripts to calculate the crystal's parameters (bending, miscut and torsion angles);

- Monochromator: contains the scripts to compute the device alignment positions on the rotational and linear axes. Furthermore it contains a script to calculate the Bragg's peak angle of the monochromator crystal;
- Slit: contains the scripts to compute the device alignment positions on the rotational and linear axes;
- X-ray source: contains the script to compute the alignment position of the device with the X-ray sensor.

In summary, this unit incorporates a collection of scripts designed to automate the post-processing of signals acquired by the X-ray sensor while different devices of the diffractometer are in motion. Integrating this post-processing unit within the diffractometer's code offers significant flexibility and streamlines the measurement of crystal parameters. Moreover, can enhance measurement repeatability and accuracy.

3.3.8 Graphical user interface

This section provides an overview of the graphical user interface (GUI) developed to enhance the usability of the x-ray diffractometer. The GUI is built using the QT framework and utilizes the code described in previous sections to control the diffractometer. Various control functions have been defined and implemented within the GUI to streamline the process of crystal characterization, thereby reducing the operator's workload.

The GUI is designed to fulfill specific requirements, acting as a bridge between the operator and the diffractometer workspace to facilitate seamless communication and interaction. It simplifies the execution of device alignment and measurements by automating and streamlining these processes for enhanced efficiency. Additionally, it automates the logging and post-processing of acquired data, reducing manual efforts and enabling more efficient analysis. The interface ensures ease of use, even for operators with limited expertise in the field, allowing them to navigate and utilize the diffractometer effectively. Furthermore, it provides the operator with comprehensive information about the states of the finite state machines and the positions of the devices, enhancing their understanding of the system's

current status and configuration. By incorporating these features, the GUI enhances the user experience, optimizes the utilization of the x-ray diffractometer, and simplifies the characterization of crystals.

The GUI has been tested with four users, with adjustments made based on their feedback to support ongoing improvements.

System architecture

The graphical user interface (GUI) is interconnected with the diffractometer framework through TCP communication, enabling bidirectional transmission of commands. JSON files are employed for transmitting commands between the GUI and the server. Four types of JSON files are utilized for communication purposes:

1. Motor Controls Json (Figure 3.37): This file enables control over the connection, disconnection, and movement of diffractometer devices. Each motor control is represented by a key-value pair, where the key denotes the control's name, and the value is an object containing relevant information. The object properties, such as "connect," "disconnect," "go home," and "relative motion," can be set to true or false. Additionally, the "stepper position" (or "axes positions" for the hexapod) property specifies the target position of the motor if the "Relative Motion" (or "Absolute Motion") button is activated;

2. Device Scans Json: This file facilitates the execution of scans. It consists of sub-keys representing individual devices, such as "Autocollimator Rotational," "Crystal Rotational," "Hexapod," and others. Each device sub-key contains properties related to data acquisition and scanning, including file names for data logging, acquisition duration, clearing of the data log file, scan range, initiation of scans, step sizes, and scan termination;

3. Beam Alignments Json (Figure 3.38): This file governs the execution of beam alignment tasks. It comprises three primary keys: "Monochromator," "Slit," and "X-Ray Source Rotational." These keys contain sub-keys indicating actions such as computing alignment positions, searching for the Bragg peak, and initiating alignment processes for respective devices;

4. Crystal Alignments Json (Figure 3.39): This file manages the execution of crystal alignment procedures. The top-level key, "Crystal

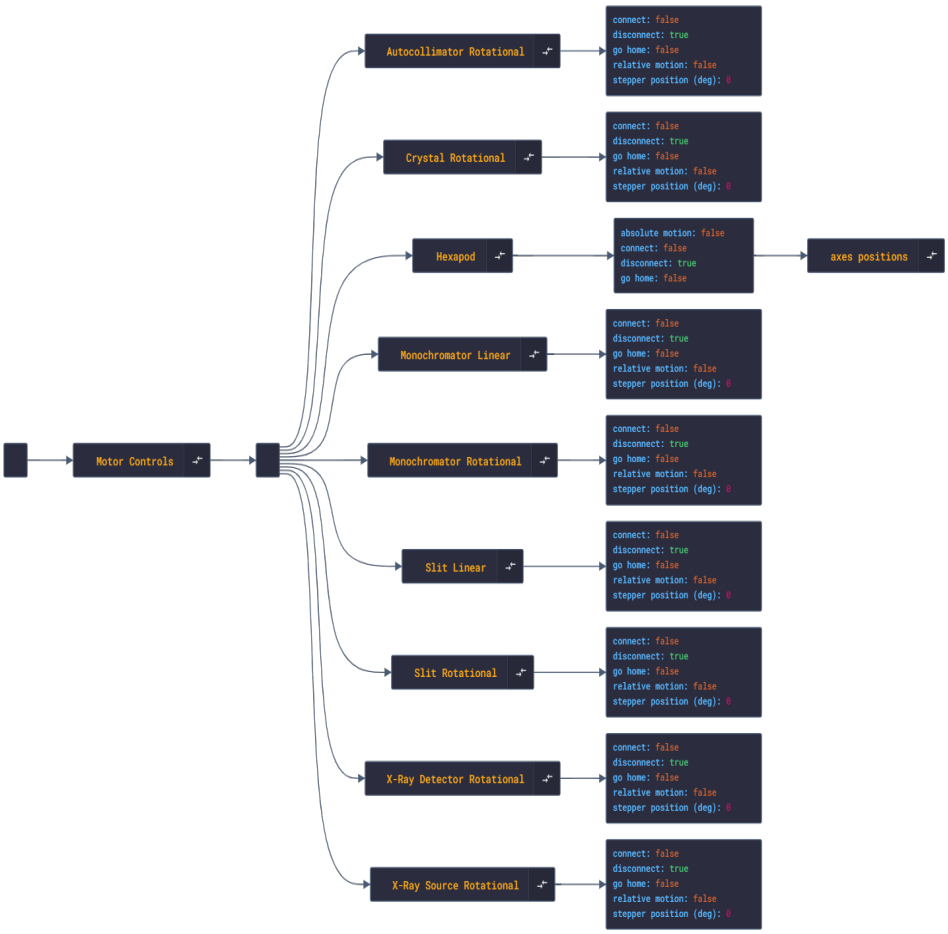


Figure 3.37: Motor Controls JSON: This configuration file manages diffractometer devices by specifying motor actions.

Alignments," encompasses sub-keys for various alignment procedures, including "Bragg Peak Search," "Check Alignment in Flipped Orientation," "X-Axis Alignment," "X-Axis Fine Alignment," "Y-Axis Fine Alignment," "YW-Axes Alignment," "Z-Axis Alignment," and others. These sub-keys possess a "start alignment" property, initially set to false, which can be activated to commence the alignment procedure;

5. Crystal Measurements Json (Figure 3.40): This file controls the execu-

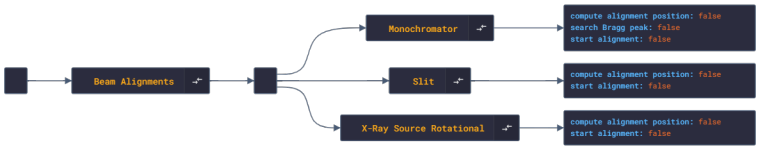


Figure 3.38: Beam Alignments Json: This file orchestrates beam alignment tasks.

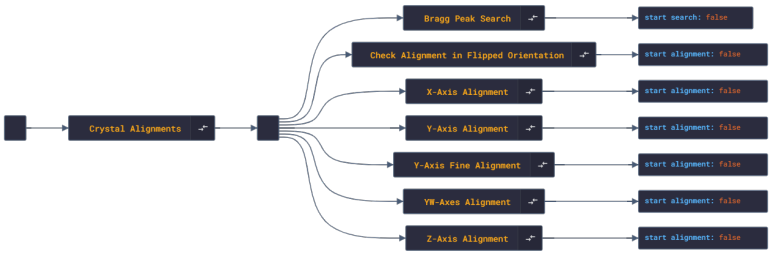


Figure 3.39: Crystal Alignments Json: This file directs crystal alignment processes.

tion of crystal measurements. It includes properties such as "Bending Angle," "Miscut Angle," and "Torsion Angle." Each property incorporates a single boolean property, "start measurement," initially set to false.

Through TCP communication and the utilization of these JSON files, the GUI facilitates seamless control and interaction with the diffractometer, enhancing the management of motor controls, device scans, beam alignments, crystal alignments, and crystal measurements.

3.3.9 Framework management server

The 'IUIManagementServer' interface (Figure 3.41) enables communication between the graphical user interface (GUI) and the system, acting as a man-

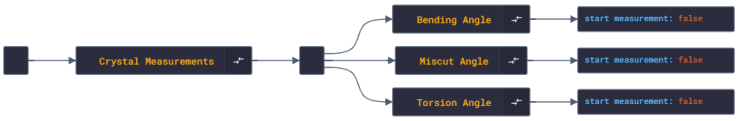


Figure 3.40: Crystal Measurements Json: This file directs crystal measurements processes.

agement server to control all diffractometer devices. It handles the creation of objects and the initialization of the JSON structure for data transmission to the client. Additionally, it defines server-related functions necessary for server setup, operation, and closure. The interface interprets messages from the client (GUI) by deserializing JSON messages, validating their structure, and executing device controller methods based on the received values. It also sends status messages to the client, providing information on the states of Finite State Machines (FSMs) and devices positions.

GUI design

As can be seen in Figure 3.42 the main window of the GUI presents two control panels:

The left control panel provides direct control over the motion of each axis within the diffractometer. It consists of two tabs, one for controlling the stepper motor and another for controlling the hexapod. Within the stepper motor tab (Figure 3.42), users can select the axis to move, input the desired target position and initiate a movement to reach that position. Additionally, there is an option to directly initiate movement to a predefined "home position". Similarly, within the hexapod tab (Figure 3.43), users can input the desired target position for each axis and initiate movement accordingly. The control panel enables precise control over the positioning of the diffractometer's axes, allowing for convenient and accurate adjustments.

The right control panel is divided in 5 tabs. The "Scans" tab (Figure 3.43) provides users with the capability to select the desired axis for scanning operations and input the necessary scan parameters. These parameters include the scan range, acquisition duration, step size, and the filename for

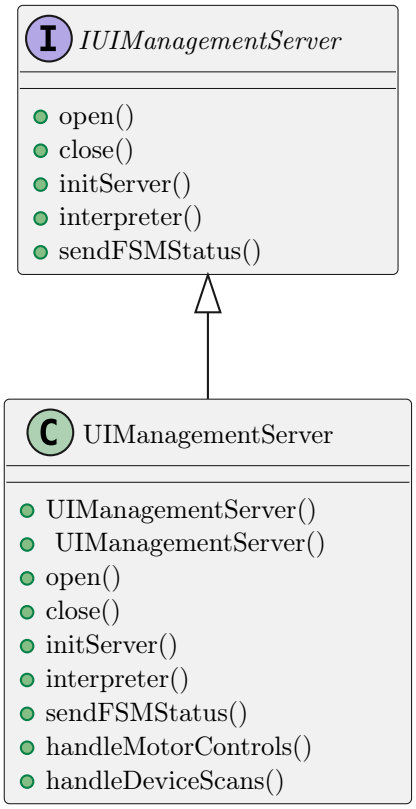


Figure 3.41: Unified Modeling Language (UML) diagram showcasing the IUIManagementServer interface and its implementation in the UIManagementServer class. The interface defines methods for managing a UI server, while the UIManagementServer class provides concrete implementation, including additional functionalities for handling motor controls and device scans.

logging the positions and x-ray detector values. Additionally, once the scan is completed, a plot displaying the count values gathered from the x-ray sensor as a function of the scanned axis is showed. This feature allows users to conveniently visualize and analyze the collected data, aiding in the interpretation and understanding of the scan results empowering users to effectively explore and analyze the behavior of the scanned axis in relation to x-ray sensor count values. By utilizing the "Scans" tab, users can easily configure and customize the scanning process, tailoring it to their specific requirements and check it's status. The tab streamlines the setup of scans, simplifying the input of essential parameters for precise data acquisition and analysis.

Within the "Beam Alignment" tab (Figure 3.44), users can access buttons corresponding to various beam alignment tasks. Clicking on any of these buttons initiates the motion to the predefined alignment position. Additionally, users have the option to enable the "Compute Alignment Position" checkbox, allowing them to restart the measurement of the alignment

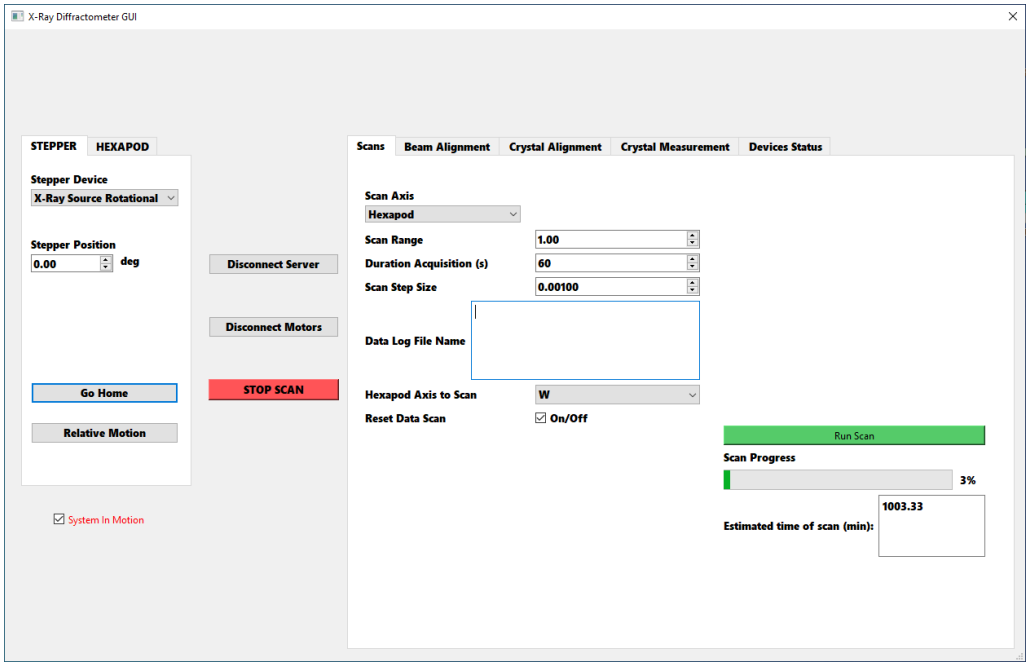


Figure 3.42: Screenshot of the left GUI control panel of the diffractometer, highlighting the tab for stepper motor control. This interface enables users to choose an axis from a drop-down menu, enter a target position, and start the movement, including returning to a predefined 'home position'.

position. When the "Compute Alignment Position" checkbox is selected, a scan of the target device is executed using configured parameters such as step size, range, and more. At the conclusion of the scan, the data recorded by the X-Ray sensor is plotted, providing a visual representation of the alignment process. The "Beam Alignment" tab streamlines the execution of beam alignment tasks, enabling users to easily initiate motions to predefined positions and perform alignment measurements with real-time data visualization;

Within the "Crystal Alignment" tab (Figure 3.45), users can find buttons that initiate specific crystal alignment tasks. Clicking on any of these buttons triggers a scan of the target hexapod axis, incorporating predefined parameters such as step size and range. Upon completion of the scan, the data recorded by the X-Ray sensor is plotted, providing a visual representation of the alignment process. The "Crystal Alignment" tab streamlines the execution of crystal alignment tasks, enabling users to initiate scans, visualize the resulting data, and make informed decisions based on the plotted information.

Within the "Crystal Measurements" tab (Figure 3.46), users have access to buttons that initiate specific crystal measurement tasks. Clicking one of these buttons triggers a motion of the target hexapod axis, following

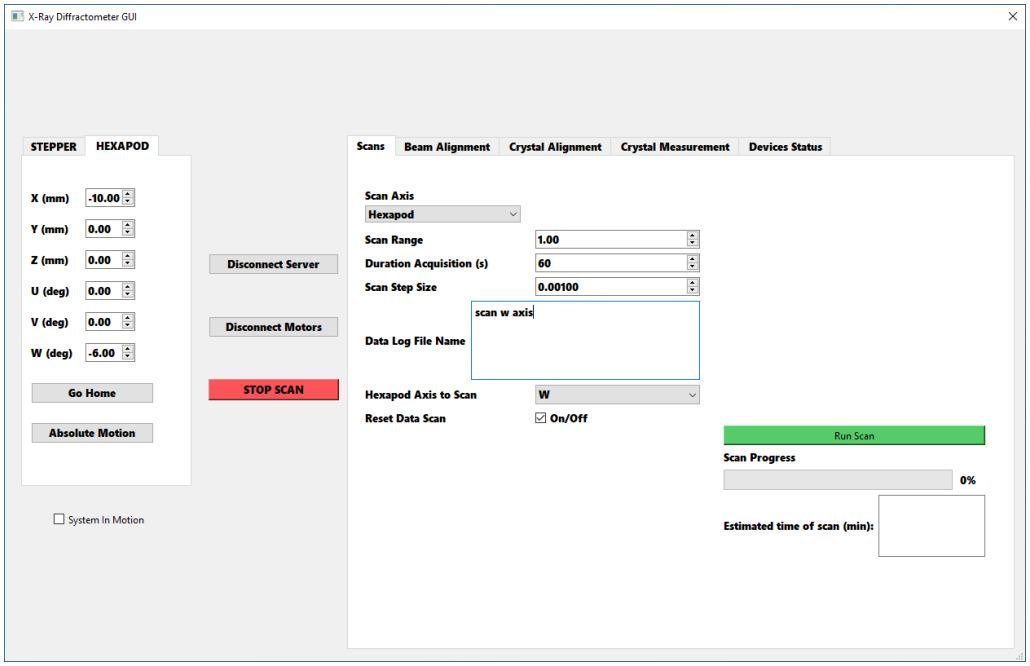


Figure 3.43: Screenshot of the left GUI control panel of the diffractometer, showcasing the tab for hexapod control. This interface provides precise control over the hexapod axes, facilitating convenient and accurate positional adjustments. The right control panel enables users to select the axis for scanning, input key parameters like range, duration, step size, and logging filename, and check the status of the scan.

predefined parameters such as step size and range. Upon completion of the measurement, the recorded data from the X-Ray sensor is plotted, providing a visual representation of the measurement results. The plotted data enables users to analyze and interpret the collected information effectively. The "Crystal Measurements" tab offers a streamlined interface for conducting crystal measurements. By integrating motion control, data acquisition, and visualization, it facilitates precise and efficient measurements of the crystal under examination.

The "Devices Status" tab (Figure 3.47) displays the current states of the Finite State Machines (FSMs) responsible for controlling the devices. Additionally, it presents the current positions of the stepper motor or hexapod axes. By accessing the "Device Status" tab, users can obtain real-time information regarding the operational states of the FSMs governing the devices. This includes the status of each device and its corresponding mode of operation. Furthermore, the tab provides an overview of the precise positions of the stepper motor or hexapod axes, allowing users to monitor and track their current positions. The "Device Status" tab serves as a comprehensive resource for users to stay informed about the operational states

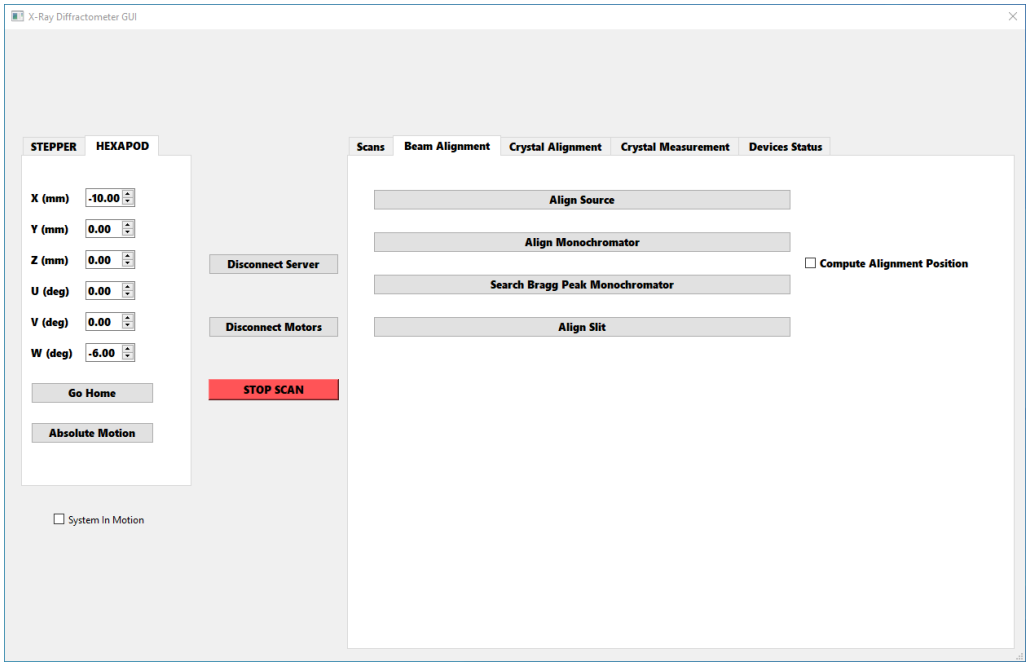


Figure 3.44: The "Beam Alignment" tab offers users a set of buttons to initiate motion to predefined beam alignment positions. By selecting the "Compute Alignment Position" checkbox, users can restart the measurement process, executing a scan. This tab simplifies beam alignment tasks, facilitating motion control and data visualization for precise alignments.

and axis positions, enabling effective management and monitoring of the system's devices.

3.4 X-ray Diffractometer Benchmarking

In this section, the diffractometer components described in Section 3.2 will be validated through measurements and alignment procedures using X-ray beams.

3.4.1 X-Ray sensor validation

Before aligning the diffractometer instrumentation, we first validated the X-ray sensor used in our study by collecting spectra at various acquisition times (Figure 3.48) while maintaining stable the X-ray source at 50 kV and 10 µA. The x-axis of this figure represents channels, which are discrete intervals that categorize the detector's output based on X-ray energy levels. It is crucial to introduce the concept of "acquisition time," which refers to the duration over which the X-ray detector collects data during an exposure. The image shows that longer acquisition times lead to higher recorded

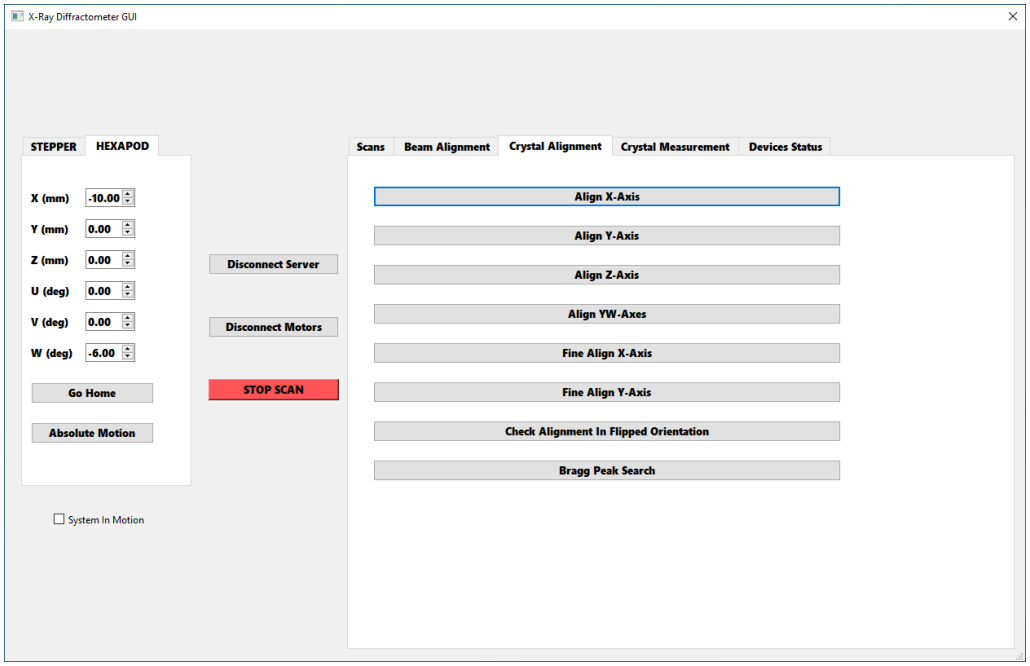


Figure 3.45: The "Crystal Alignment" tab provides users with buttons to initiate crystal alignment tasks, triggering scans of the target hexapod axis using predefined parameters. Upon scan completion, the X-ray sensor data is plotted, offering a visual representation of the alignment process. This tab simplifies crystal alignment by allowing users to execute scans and analyze results.

counts, especially evident in the peaks corresponding to K-alpha and K-beta radiation. Additionally, it can be confirmed that the X-ray spectra closely resemble the spectrum from the silver anode (Ag) X-ray source detailed in Section 3.2.1.

It has been further analyzed the centroids of the K-alpha radiation peaks in relation to their maximum indices across different acquisition times (Figure 3.49). With longer acquisition times, it has been observed a stabilization in the K-alpha centroid values, indicating a convergence between centroids and the positions of maximum intensity for K-alpha radiation.

Moreover, it can be noticed in Figure 3.49 a minimal difference (maximum of 2 channels) between the calculated centroid of K-alpha radiation and its positions of maximal intensity. This discrepancy underscores the effectiveness of both methods used to determine the K-alpha radiation characteristics. Based on these findings, our study focuses solely on K-alpha radiation, conducting data acquisitions at varying acquisition times ensuring optimal counting at the K-alpha peak.

To further validate the X-ray sensor, an additional analysis was conducted to assess the uncertainty in count measurements. This involved acquiring 10 distinct spectra at different acquisition times. The focus was

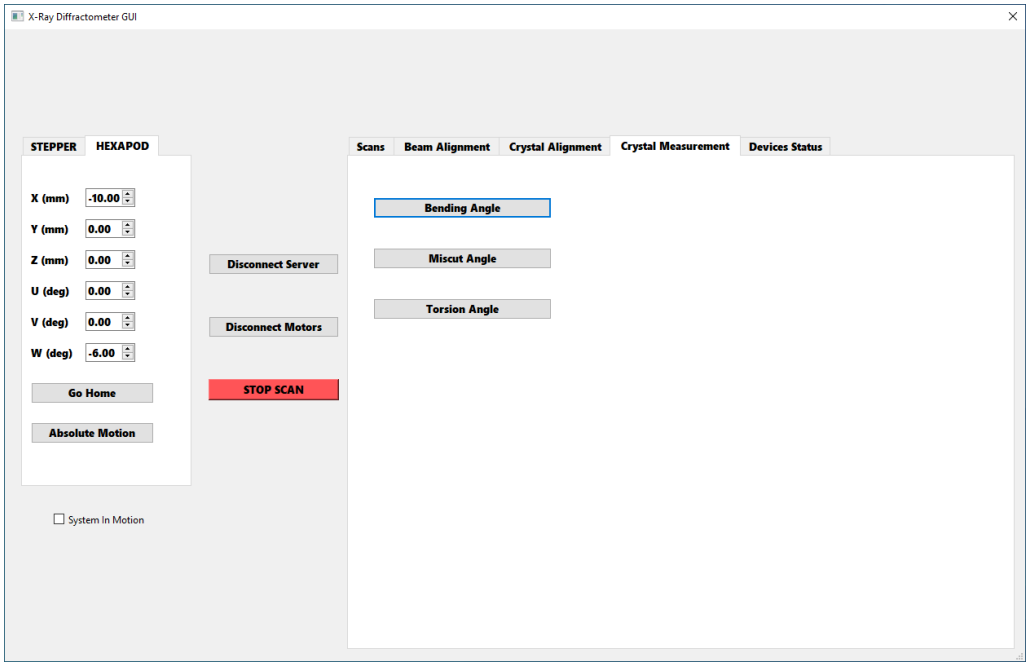


Figure 3.46: The "Crystal Measurements" tab provides an interface for performing crystal measurement tasks. With dedicated buttons to initiate hexapod axis motions based on predefined step size and range parameters, this tab streamlines the measurement process. Once a measurement is completed, data from the X-ray sensor is automatically plotted, offering users a clear visual representation of the results. This integration of motion control, data acquisition, and visualization allows for precise and efficient analysis of the crystal under examination.

on examining variations in k-alpha counts detected at the centroid of the radiation, at the peak, and in calculating the integral of the chosen radiation (Figure 3.50).

Figure 3.51 displays the distribution of X-ray counts measured at the centroid, peak and integral of the bell-shaped spectra of K-alpha radiation from a silver anode (Ag) source. It also shoes that the distribution remains spread out even with extended accumulation times.

Additionally, Figure 3.52 depicts the trend of the standard deviation of K-alpha counts detected at the centroid, peak, and by integrating the K-alpha radiation, plotted against accumulation time. This trend indicates that as accumulation time increases, the standard deviation of the counts also rises, resulting in noisier measurements.

Figure 3.53 illustrates the relationship between accumulation time and the percentage standard deviation of X-ray count measurements. This measure normalizes the standard deviation by the mean (Equation 3.16), providing a scale-independent metric that facilitates comparisons across different accumulation times and signal types. As accumulation time increases, the

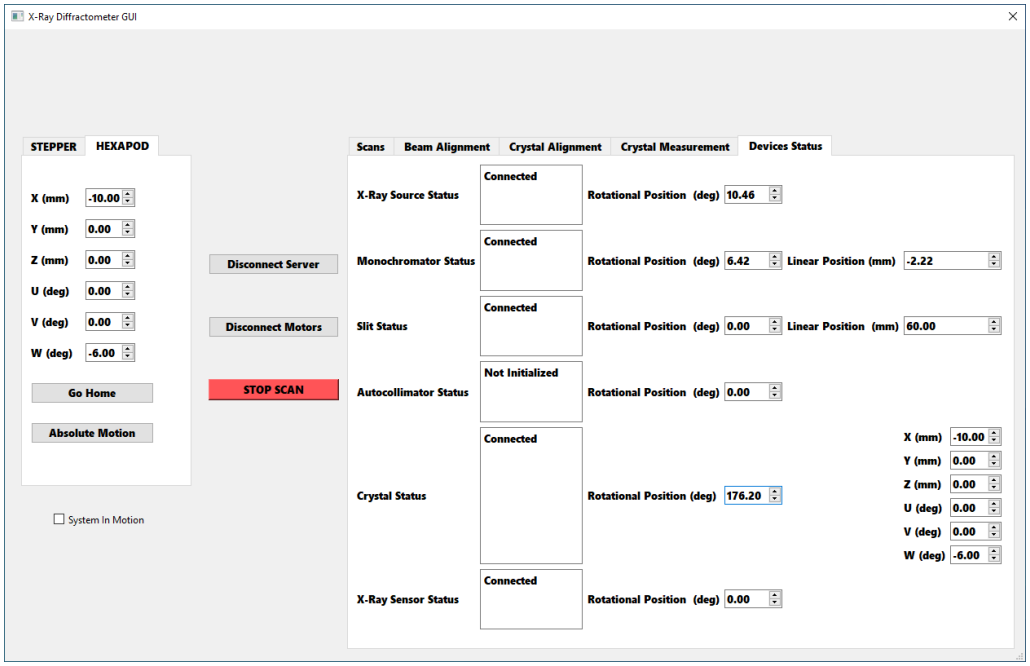


Figure 3.47: The "Devices Status" tab offers real-time updates on the states of the Finite State Machines (FSMs) controlling the devices, along with the current positions of the stepper motor or hexapod axes. This tab allows users to monitor device statuses, operational modes, and precise axis positions, facilitating efficient system management and oversight.

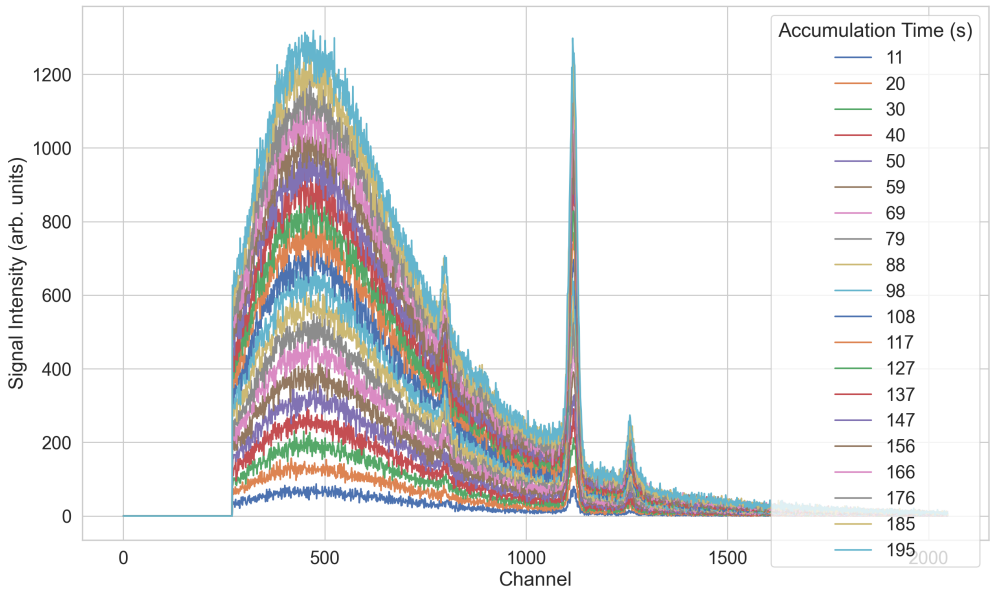


Figure 3.48: The X-ray spectra from the Ag anode X-ray source detected by the sensor change with acquisition time, with longer acquisition times resulting in higher recorded counts.

percentage standard deviation for counts at the centroid, peak, and integral tends to decrease. Additionally, even at shorter accumulation times, the

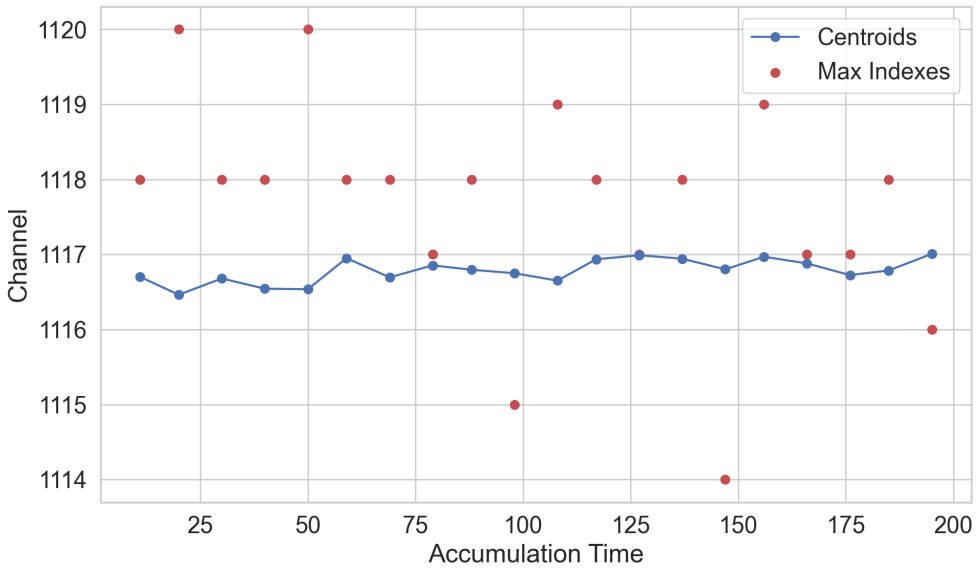


Figure 3.49: Channels as a function of acquisition time for K-alpha radiation. Optimal counting occurs at the K-alpha peak.

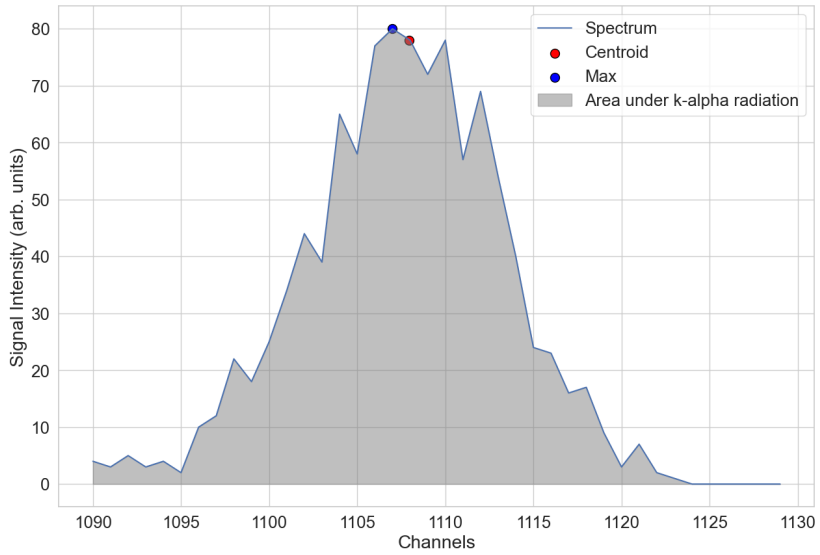


Figure 3.50: K-alpha radiation spectrum obtained with an accumulation time of 150 seconds from a silver (Ag) anode source. The red marker indicates the centroid of the bell-shaped curve, the blue marker shows the maximum of the K-alpha radiation, and the highlighted gray area represents the integral of the radiation.

percentage standard deviation is significantly lower for the integral of the K-alpha radiation compared to the centroid and peak counts. This suggests that using the integral of the K-alpha radiation offers a more stable measurement, making it an optimal choice for accurately computing the desired radiation.

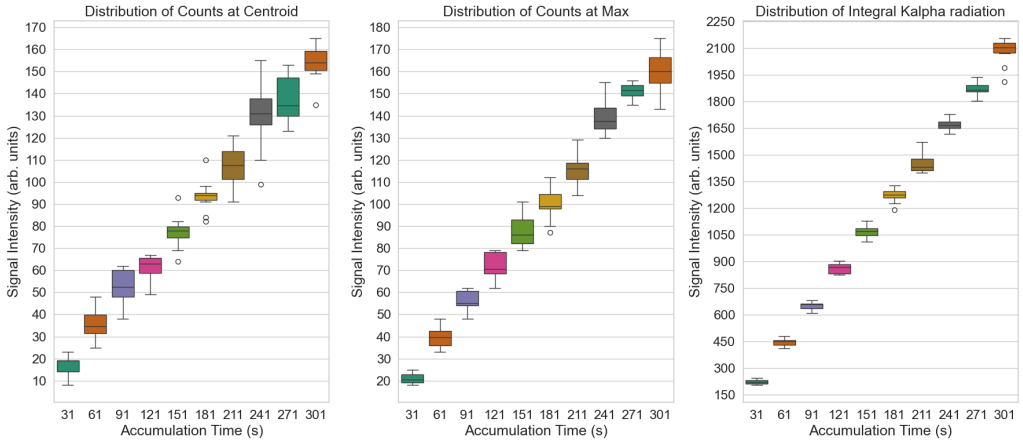


Figure 3.51: Distribution of X-ray counts measured at the centroid (left), maximum (center) and computing the integral (right) of the K-alpha radiation spectra.

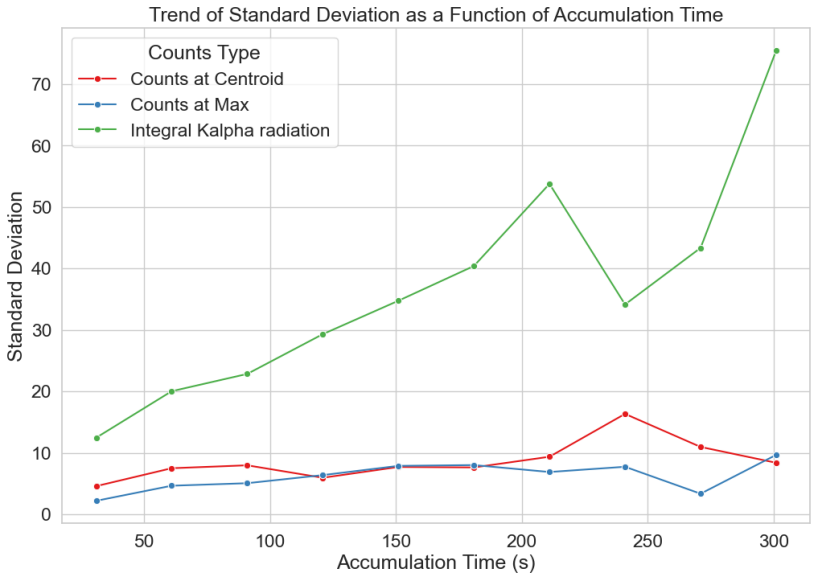


Figure 3.52: Behavior of the standard deviation of k-alpha counts detected at the centroid, peak, and by integrating the K-alpha radiation, plotted as a function of accumulation time. The figure illustrates how the variability in counts increases with increased accumulation time.

$$\text{Percentage Std. Deviation (X)} = \frac{\text{Std. Deviation (X)}}{\text{Mean (X)}} * 100\% \qquad (3.16)$$

Finally, to further reduce any residual noise in the measurements, a Savitzky-Golay filter [70] can be applied to the counted radiation.

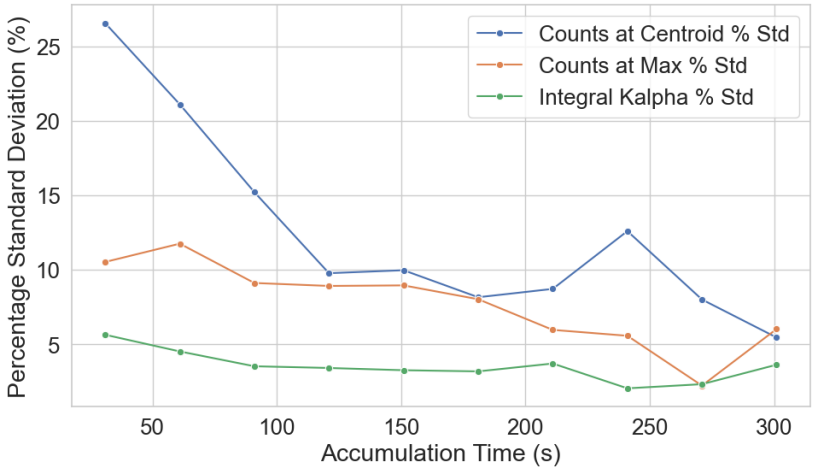


Figure 3.53: Percentage standard deviation of X-ray count measurements as a function of the accumulation time. At shorter accumulation times, the percentage standard deviation is significantly lower for the integral of K-alpha radiation compared to the centroid and peak counts. This indicates that the integral provides a more stable measurement.

3.4.2 Beam alignment

The second step in the validation of the X-Ray diffractometer has been the beam alignment, which involves two key tasks: aligning the X-ray source and aligning the monochromator. This alignment is essential to ensure the beam is properly aligned with the x-ray detector before starting the measurements. All alignments have been performed with the X-ray source set at 50 kV, 200 uA.

X-Ray source alignment

The X-ray source alignment is carried out by incrementally adjusting the position of the device while keeping all other components out of the beam path, except for the X-ray detector. The objective of this alignment is to position the X-ray source so that it is perfectly aligned with the X-ray detector. As illustrated in Figure 3.54, the source is considered aligned with the beam at the center of the bell-shaped curve. Initially, at the beginning of the scan on the left-hand side of the plot, the count rate is low because the beam is not fully intercepted by the X-ray detector. As the X-ray source is moved along the positive x-axis, the count rate increases, peaking when the alignment is optimal, and then decreases again as the source moves out of alignment.

The central position in the alignment is determined using the Full Width at Half Maximum (FWHM) method [71]. This involves smoothing the sen-

sor data and identifying the x-values where the smoothed signal crosses the half maximum value, which is the point halfway between the peak and the baseline. These crossing points define the FWHM, and the central position is the midpoint between them, as indicated by the green vertical line in Figure 3.54. In this case, the alignment position is 1.963 degrees. This method is robust and provides a clear, quantitative measure of the central alignment position based on the characteristics of the signal itself.

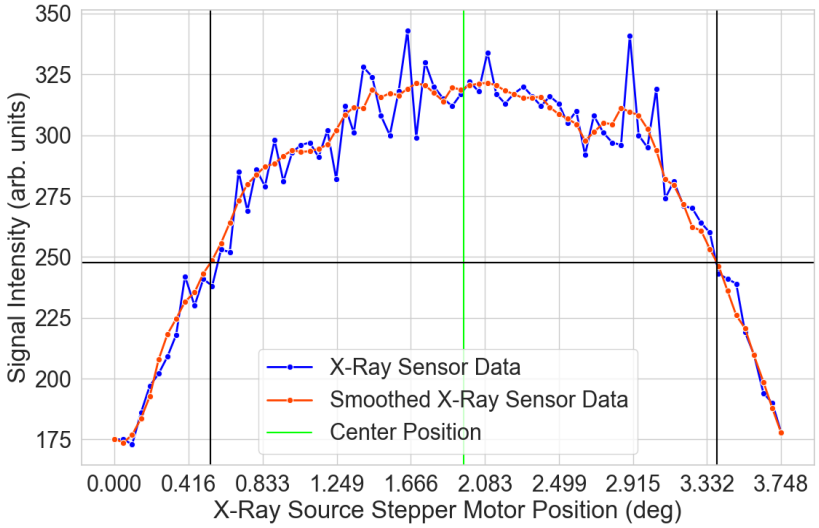


Figure 3.54: Count rate of the X-ray beam as a function of the X-ray source position. The alignment position, highlighted by the green vertical line, is calculated using the Full Width at Half Maximum (FWHM) method.

Monochromator alignment

The alignment of the monochromator consists of determining the point at which the device intercepts the X-ray beam, ensuring accurate positioning along the beam path.

The linear alignment involves incrementally adjusting the linear axis of the monochromator. As depicted in Figure 3.55, the count rate is initially low at the start of the scan because the device intercepts the full width of the X-ray beam. As the device moves, the count rate increases, peaking when the device is completely out of the beam path. The alignment position is determined at the midpoint of the slope between the maximum and minimum count rates, which in this case is at 83.872 mm.

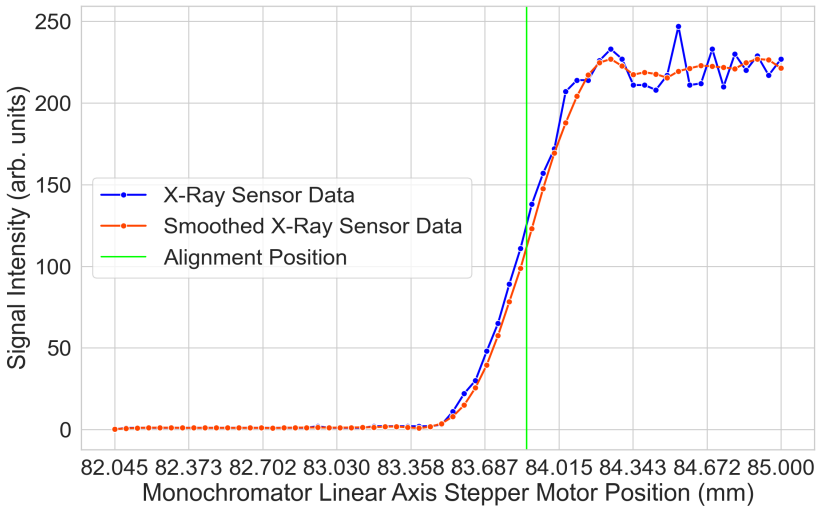


Figure 3.55: Count rate of the X-ray beam as a function of the linear position of the monochromator. The alignment position is highlighted by the green vertical line.

3.4.3 Crystal alignment

Proper crystal alignment involves several steps to ensure the device is correctly positioned with the collimated beam before measurements can begin. These steps include:

1. Aligning the X-axis of the crystal;
2. Aligning the W-axis of the crystal;
3. Searching for the Bragg peak.

X-Axis alignment

This alignment involves moving the crystal incrementally along the X-axis while recording X-ray sensor data as depicted in the diagram in Figure 3.56. Initially, the crystal should be positioned outside the beam path and then gradually inserted into it. As the crystal enters the beam, the signal intensity should decrease because the crystal begins to obstruct the beam. The intensity will then increase again as the crystal moves out of the beam path. In Figure 3.57 is displayed the behaviour aforementioned.

By calculating the average counts before the crystal intersects the beam and the average counts when the crystal is obstructing the beam, the midpoint that represents the alignment position on the X-axis between these two values can be determined (Figure 3.58).

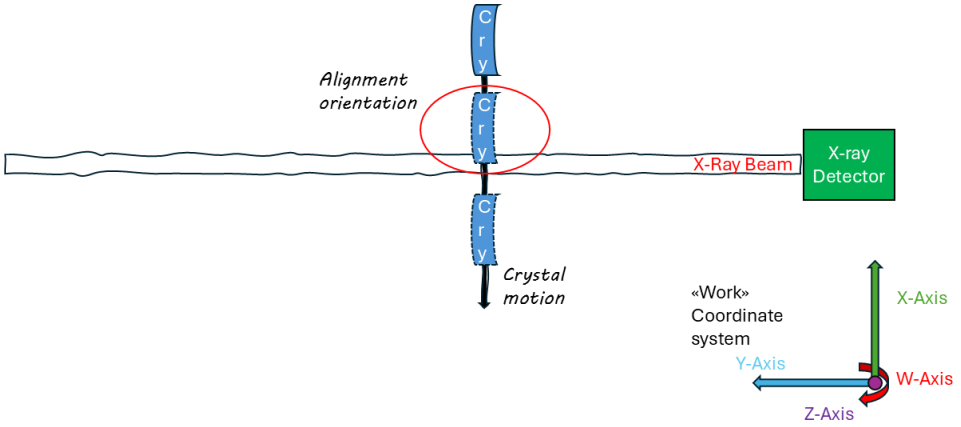


Figure 3.56: Illustration of the crystal being positioned within the X-ray beam. The crystal is considered properly aligned when it intersects 50% of the beam.

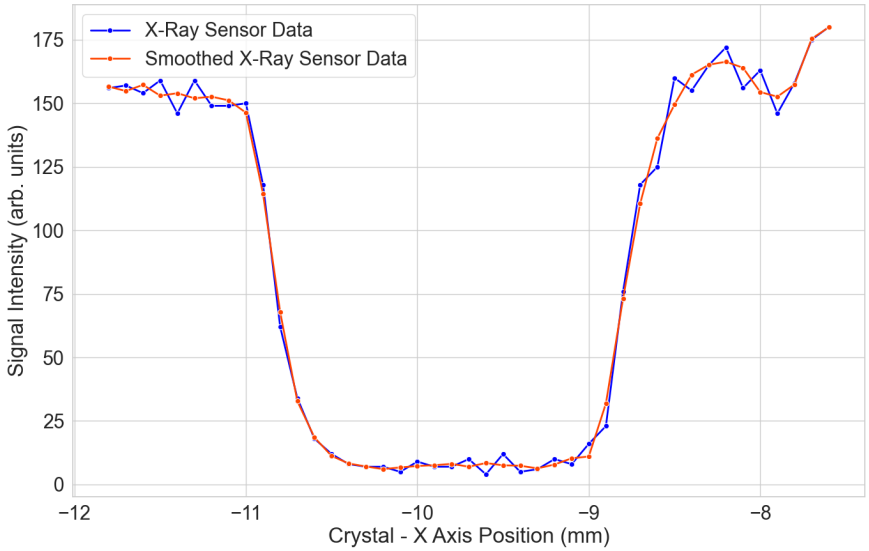


Figure 3.57: Signal intensity as a function of the crystal x-axis position illustrating the alignment process where the crystal is moved incrementally along the x-axis while recording X-ray sensor data. Initially, the crystal is positioned outside the beam path and then gradually inserted. As the crystal enters the beam, signal intensity decreases due to beam obstruction, and increases again as the crystal exits the beam path.

W-Axis alignement

The goal of this alignment is to position the front face of the crystal relative to the X-ray beam. This alignment is essential for determining the

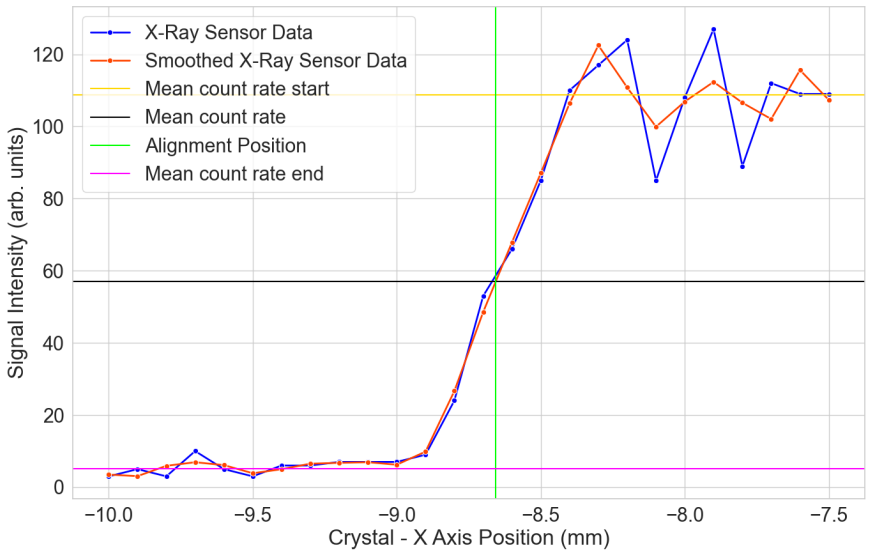


Figure 3.58: Signal intensity as a function of the crystal x-axis position. By averaging the counts before and during the crystal's obstruction of the beam, the midpoint between these values is determined, representing the optimal alignment position on the X-axis.

offset needed for the rotational W-axis position of the crystal when locating Bragg's peak. To achieve this, an X-axis alignment needs to be carried out for various W-axis orientations. Afterward, as shown in Figure 3.60, a Y-axis scan within the "Tool" coordinate system is performed, spanning from -1.5 to +1.5 mm.

For each X-ray sensor set of data points collected during the Y-axis scans, a linear fit was calculated (Figure 3.60). The goal of this alignment is to minimize the slope of the linear fit. When this is accomplished, the crystal is intercepting 50% of the X-ray beam throughout the entire scanned range of the Y axis.

Figure 3.61 demonstrates how the slopes of the linear fits decrease as the crystal's position is adjusted from -0.4 to +0.2 degrees. Additionally, it shows that at a W-axis position of 0 degrees, the slope of the linear fit approaches zero, indicating that this position is close to the optimal alignment therefore no offset in expected on this axis when detecting the diffraction peak.

Bragg's angle

After aligning the crystal with the X-ray beam, the next step is to identify the first Bragg's Angle—the angle at which the diffracted X-ray beam

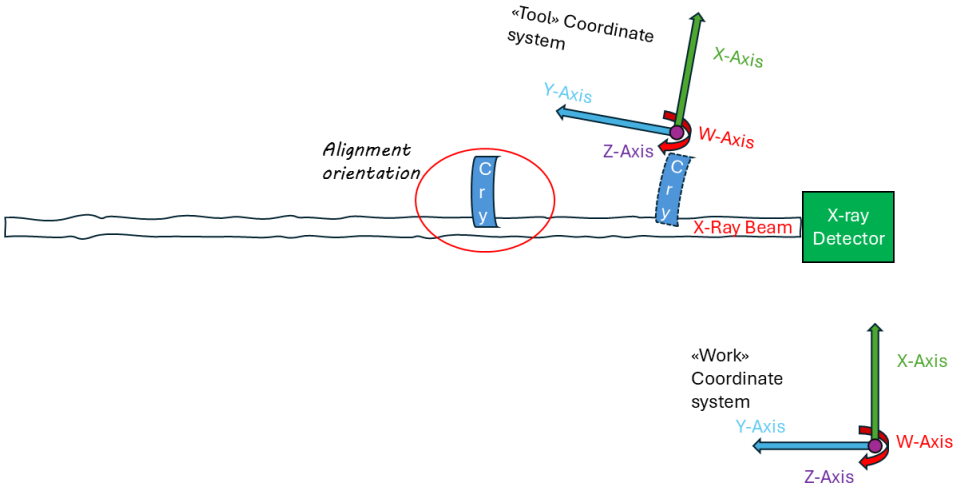


Figure 3.59: Illustration of the W-axis alignment of the crystal. The device is considered correctly aligned when its front face is oriented towards the X-ray beam, as indicated by the red oval. The tool coordinate system is is a reference frame that moves with the Hexapod's carriage (the crystal).

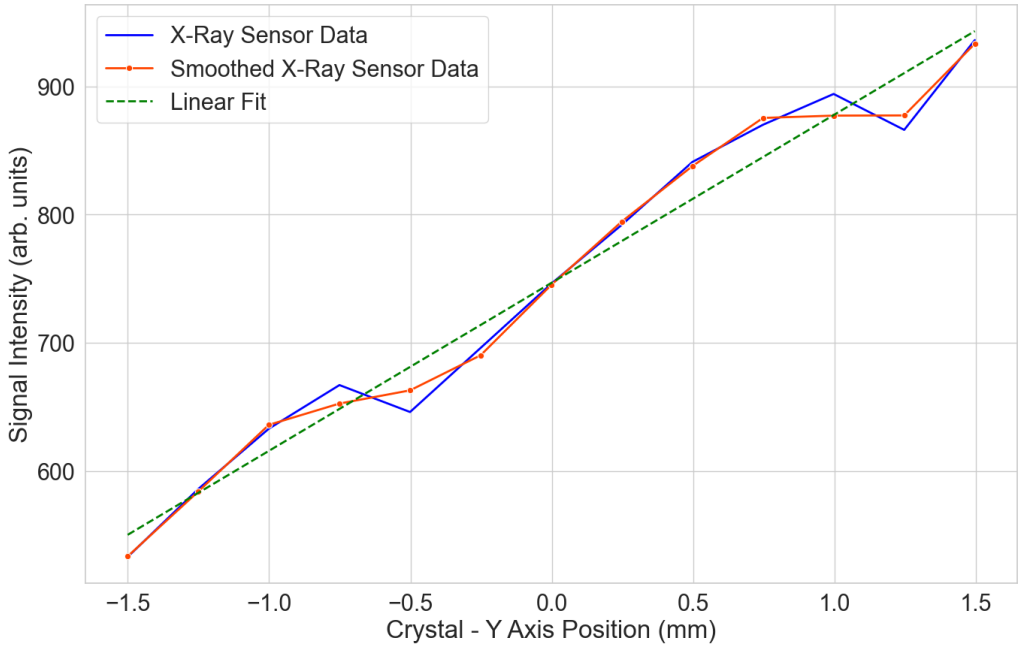


Figure 3.60: X-Ray signal intensity as a function of the Y-axis position. When the front face of the crystal is misaligned the device. At the beginning of the scan intercepts 50% of the beam, as it is moved on the Y axis of the tool coordinate system it starts to go out of the beam path and the count rate increases.

reaches its maximum intensity (Figure 3.62). This is done by rotating the crystal's yaw angle in small increments while collecting data from the X-ray sensor. Theoretically, with the used crystal collimator (strip), diffraction

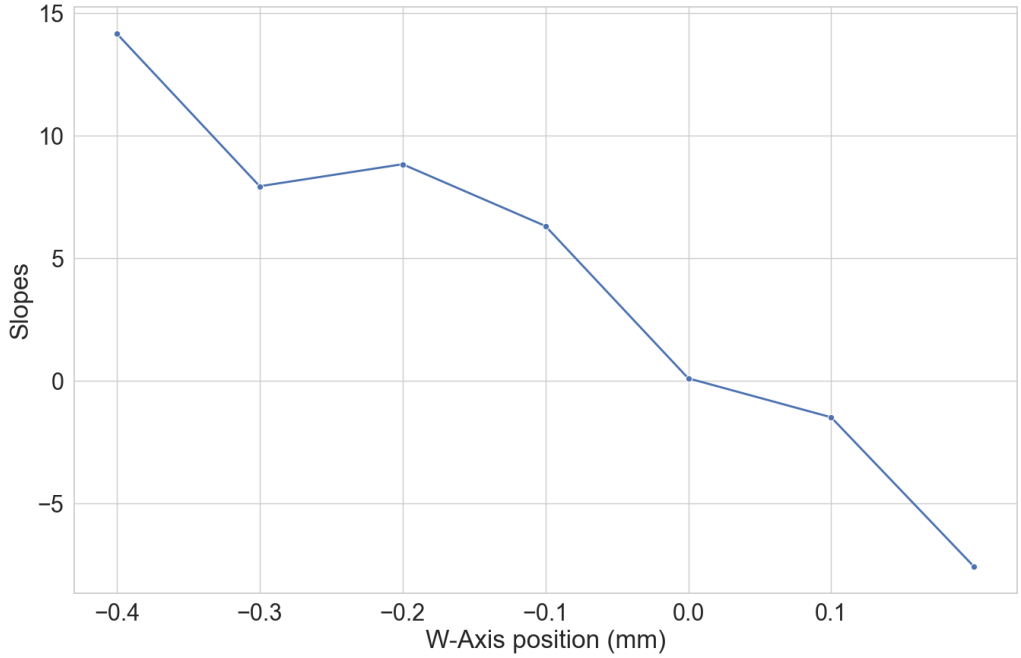


Figure 3.61: Slopes of the linear fit applied to the Y-axis scans as a function of the W-axis orientation. The slopes decrease as the crystal's position is adjusted from -0.4 to 0.2 degrees.

is expected to occur on the (110) plane. This corresponds to a yaw angle of approximately 4.17° as demonstrated in Equation 3.13. Thus, the Bragg angle θ, which represents the angle between the incident X-ray beam and the crystal surface, is expected to be $\theta = 4.17°$. Consequently, the angle between the incident X-ray beam and the detector is $2\theta = 8.34°$.

Since there was no reference on this machine for where the diffracted beam would appear, even though it was expected at an angle of $2\theta = 8.34°$ between the incident X-ray beam and the detector, a dual-axis scan involving the detector and the crystal yaw axis was performed. For each 1° step of the detector, a corresponding yaw scan was conducted, with the crystal yaw angle (W-axis) ranging from -3.5° to -5°, using a step size of 0.001° and acquisition times of 30, 45, or 60 seconds per step, resulting in each scan taking approximately 16 hours. Figure 3.63 shows an example of a W-axis scan with the detector positioned at 8.34°, where no diffraction was observed. In total, 15 yaw axis scans were performed at different positions of the X-ray detector. The summary of X-ray detector positions and corresponding acquisition times is presented in Table 3.1. Despite the efforts, no distinctive diffraction pattern was observed at any of the angles tested, as shown in the table, indicating no detection of the expected radiation diffracted around the Bragg angle depicted in Figure 3.64.

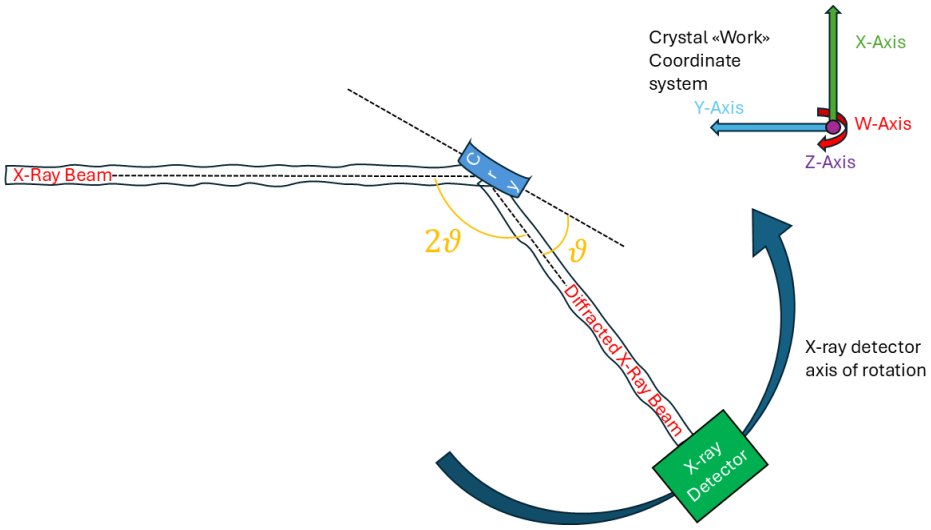


Figure 3.62: A diagram illustrating the crystal diffracting a monochromatic X-ray beam, with the resulting diffracted beam being intercepted by the X-ray sensor. The diagram also highlights the angle between the X-ray beam and the crystal (θ) and the angle between the X-ray beam and the detector (2θ).

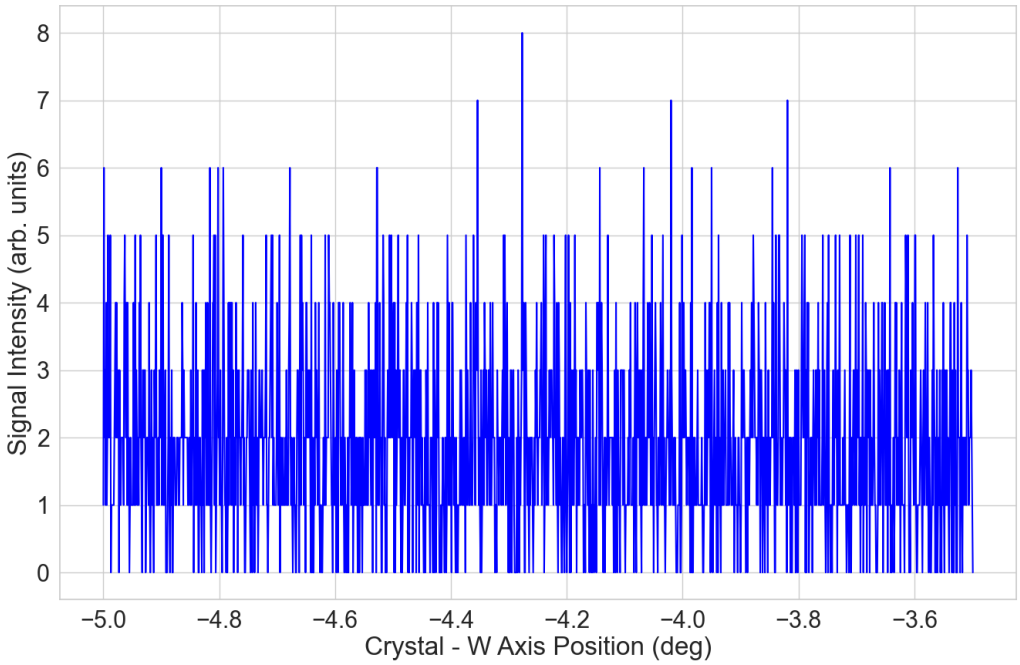


Figure 3.63: Signal intensity over time during crystal rotation along the yaw axis. No indication of a diffracted beam being detected.

The absence of a clear diffraction pattern may be influenced by the crystal's pitch angle. A misalignment in this angle could cause the diffracted beam to shift vertically, potentially missing the detector aperture. This issue could be resolved by having an X-ray detector with increased aperture size.

Table 3.1: X-ray detector positions and corresponding acquisition times.

X-ray Detector Position (deg)	Time of Acquisition (s)	Diffraction
6.34	30	No
7.34	30	No
8.34	60	No
9.34	60	No
10.34	30	No
11.34	30	No
12.34	30	No
13.34	30	No
14.34	30	No
15.34	45	No
16.34	45	No
17.34	45	No
18.34	45	No
19.34	45	No

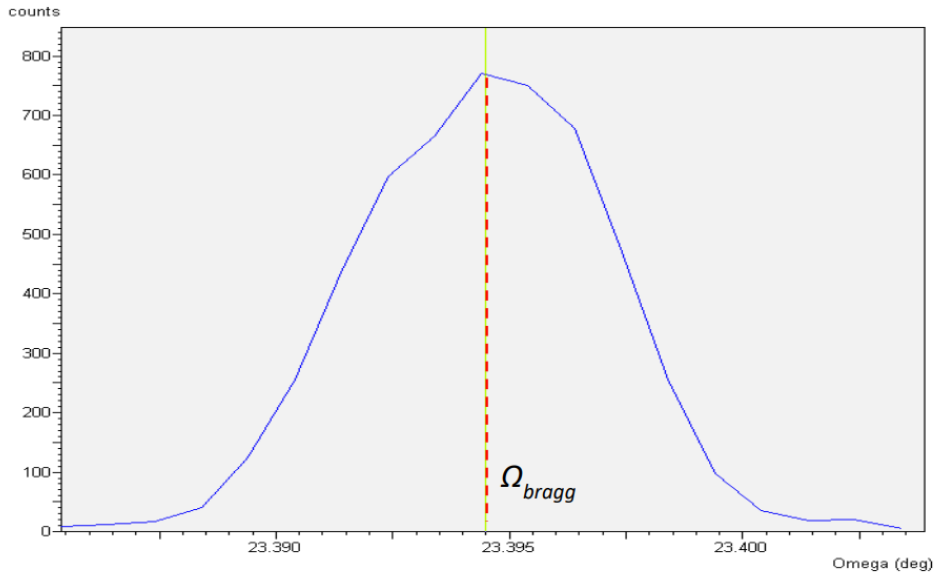


Figure 3.64: Rocking curve showing crystal diffraction. The curve illustrates the diffraction pattern of the crystal as the yaw angle is varied. The angle at which the intensity of diffracted X-rays reaches its peak corresponds to the Bragg angle, from [29].

If the aperture is not wide enough, it might fail to capture the diffracted beam. This limitation can result in missing the scattered radiation.

3.4.4 Crystal measurements

Bending angle

This measurement involves acquiring a series of rocking curves around the previously determined Bragg angle at various positions along the Y-axis of the hexapod as depicted in Figure 3.65. For each new Y-axis position, a corresponding rocking curve must be obtained. Upon completion of the measurement, a collection of rocking curves will be available. From this dataset, the yaw angle position corresponding to the maximum intensity of diffracted X-rays for each Y-axis position can be extracted.

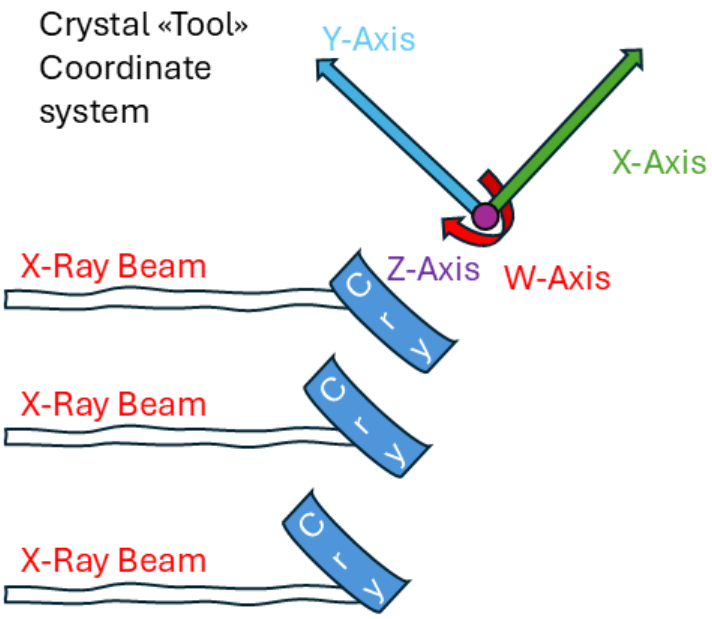


Figure 3.65: Illustrative diagram of the bending angle measurement. For each step along the crystal's Y-axis, a corresponding rocking curve is obtained by scanning the crystal's W axis.

To determine the crystal's bending angle, a linear fit is applied to the yaw angle positions corresponding to the peaks of the rocking curves where diffraction is maximized. The bending angle is then calculated by multiplying the slope of this linear fit by the known thickness of the crystal (Equation 3.17). The process for analyzing these measurements is summarized in Algorithm 1.

$$\text{Bending angle} = \frac{\text{Slope linear fit}}{\text{Crystal's thickness}} \tag{3.17}$$

Algorithm 1: Measurement of the Crystal's Bending Angle

Data: Rocking curves and Y-axis positions
Result: Crystal's bending angle

```
begin
    Acquire rocking curves around the Bragg angle at different
     Y-axis positions;
    foreach Y-axis position do
        Measure the corresponding rocking curve;
        Identify the yaw angle position with maximum diffraction
         intensity;
    Plot yaw angle positions versus Y-axis positions;
    Apply linear fit to the curve;
    Calculate the slope of the linear fit;
    Compute the bending angle using Equation 3.17.
```

Miscut angle

To measure the crystal's miscut angle, the procedure begins with rotating the crystal 180°around the rotational axis to analyze the back face. Following this rotation, a new set of rocking curves is acquired at the same Y-axis positions where rocking curves were previously measured for the bending angle. From this new dataset, obtained in the flipped orientation, the yaw angle positions corresponding to the maxima of diffraction are recorded.

For each Y-axis position, the yaw angle corresponding to the diffraction maximum in the flipped orientation is subtracted from the yaw angle observed in the original orientation. The resulting difference is then divided by 2 to calculate the miscut angle. This process is summarized in Equation 3.18.

$$\text{Miscut angle}[i] = \frac{\text{Bragg angle before flip}_{y[i]} - \text{Bragg angle after flip}_{y[i]}}{2}. \tag{3.18}$$

Upon completing this procedure a set of miscut angles is determined for each Y-axis position based on the rocking curves acquired before and after the crystal flip. The final miscut angle of the crystal is then obtained by averaging these calculated miscut angles. The process for the computation of the miscut angle is summarized in Algorithm 2.

Algorithm 2: Measure the Crystal's Miscut Angle

Data: Initial rocking curves (before flip), rocking curves (after flip), Y-axis positions

Result: Crystal's miscut angle

```
begin
    Rotate the crystal by 180°around the rotational axis to analyze the back face;
    foreach Y-axis position do
        Acquire a new set of rocking curves in the flipped orientation;
        Record the yaw angle positions corresponding to the maxima of X-ray diffraction in the flipped orientation;
        Compute the miscut angle using Equation 3.18;
    Calculate the average of the computed miscut angles.
```

Torsion angle

This measurement entails collecting a series of rocking curves around the previously determined Bragg angle at different positions along the Z-axis (vertical orientation) of the hexapod, as illustrated in Figure 3.66. For each new position along the Y-axis, a corresponding rocking curve is recorded. From this dataset, the yaw angle position that corresponds to the peak intensity of the diffracted X-rays for each Z-axis position can be determined.

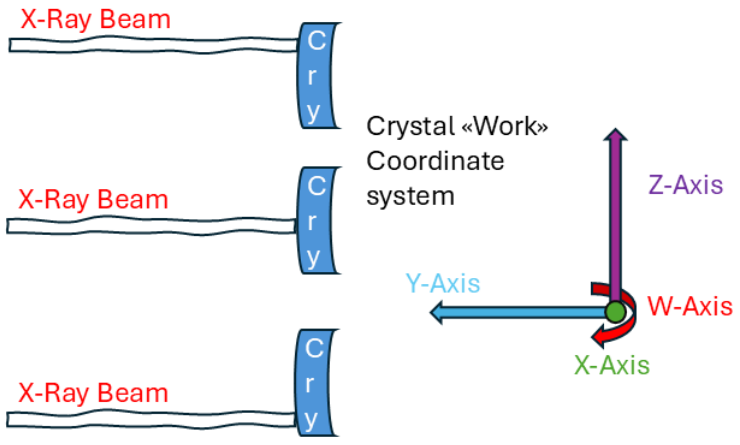


Figure 3.66: Illustrative diagram of the torsion angle measurement. For each step along the crystal's Z-axis, a corresponding rocking curve is obtained by scanning the crystal's W axis.

To determine the crystal's torsion angle, a linear fit is applied to the yaw angle positions corresponding to the peaks of the rocking curves where diffraction is maximized. The torsion angle is then equal to the slope of this linear fit. The process for analyzing these measurements is summarized in Algorithm 3.

Algorithm 3: Determine the Crystal's Torsion Angle

Data: Rocking curves measured at different Z-axis positions
Result: Crystal's torsion angle

```
1 begin
2     foreach position along the Z-axis do
3         Acquire a rocking curve around the previously determined
            Bragg angle;
4     foreach rocking curve do
5         Determine the yaw angle position corresponding to the peak
            intensity of the diffracted X-rays;
6     Plot the yaw angle positions as a function of Z-axis positions;
7     Apply a linear fit to the plot;
8     Extract the slope of the linear fit;
```

Chapter 4

Implementation of crystal piezo-goniometer controller

This section details the development of a novel testbench designed for assessing the control of piezo goniometers within SAMbuCa [55]. The development process encompasses the selection of the hardware, the creation of Very High-Speed Integrated circuit (VHSIC) Hardware Description Language (VHDL) code to regulate the devices, and concludes with the development of a real-time GUI for controlling the angular position of the crystal using interferometers.

4.1 The testbench hardware setup

The testbench comprises the design of both an angular movement controller and an angular reader, which will be described in this section. The schematic of the project is illustrated in Figure 4.1. It features an FPGA (Xilinx Zynq Ultrascale+ ZCU102 [72]) that interfaces with a digital-to-analog converter (AD5781 from Analog Devices [73]) via the Serial Peripheral Interface (SPI) [74] protocol. The converter's output is then sent to a voltage amplifier with a gain of 20, which provides a voltage range of [-20, 150] V. This amplified voltage drives a piezoelectric actuator which produces the rotation for the crystal. Additionally, two interferometric heads in combination with the Attocube IDS3010 interferometer [31] measure the distances to the left and right edges of the crystal. The measured distances are then transmitted to the FPGA using the Attocube proprietary HSSL

[31] protocol.

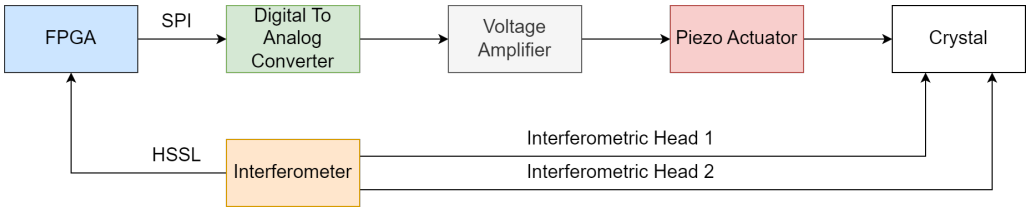


Figure 4.1: Schematic of the testbench setup. An FPGA interfaces with a digital-to-analog converter via the SPI protocol. The converter's output is amplified by a voltage amplifier which drives a piezoelectric device to control the rotation of the crystal. Two interferometric heads, in conjunction with the interferometer, measure the distances to the left and right edges of the crystal. These measurements are transmitted to the FPGA using the proprietary HSSL protocol.

Figure 4.2 illustrates the complete hardware setup employed in this study. The setup includes the Xilinx Zynq Ultrascale+ ZCU102 FPGA evaluation board, which is connected to the AD5781 Digital to Analog Converter (DAC) and the interferometer via two custom-made cables. The DAC is linked to a voltage amplifier, whose output is then fed into the piezo actuator.

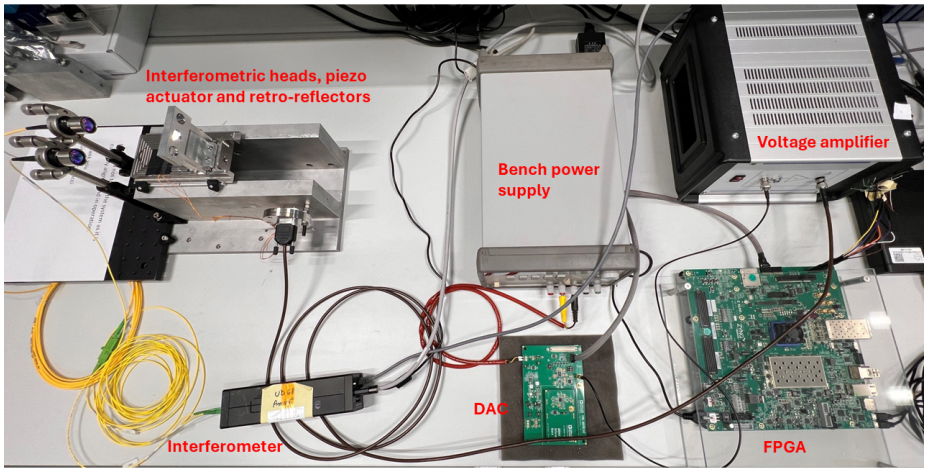


Figure 4.2: Full hardware configuration used in this study. The setup includes the ZCU102 FPGA evaluation board, connected to the AD5781 DAC and the interferometer. The DAC output is routed through a voltage amplifier before being delivered to the piezo actuator.

Two optical interferometric heads precisely measure the distance to corresponding retro-reflectors firmly connected to the crystal assembly (Figures 4.3 and 4.4), offering a measurement feedback with a resolution of 1 pm.

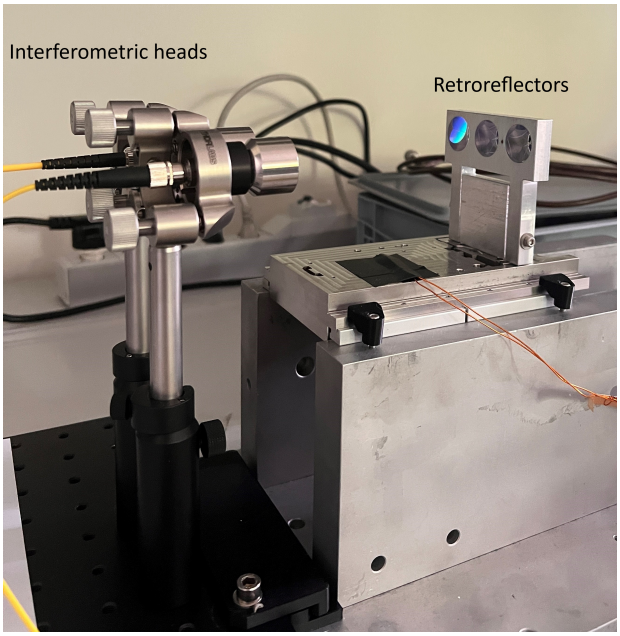


Figure 4.3: The two interferometric heads measure the distance to corresponding retro-reflectors, which are securely attached to the crystal assembly. This setup provides a high-resolution measurement feedback with accuracy up to 1 picometer (pm).

4.1.1 Vivado block design

The DAC controller and interferometer encoder are designed using a Xilinx Zynq Ultrascale+ ZCU102 and developed within the Vivado [75] IDE. The hardware's block diagram is depicted in Figure 4.5. It features the Zynq Processing System (PS), which provides the clock and reset signals to the 'AXI Interconnect' Intellectual Property (IP) block. This IP block connects the PS to the custom IP block 'spi_dac_init,' which contains the finite state machine (FSM) responsible for managing DAC write operations. The 'spi_dac_init' block features two input lines: 'clk_i' for the clock and 'resetn_i' for the reset. The clock is generated by a 'Clocking Wizard,' producing a 6.25 MHz signal. The output lines of the custom IP block include 'sck_o' for the clock output, 'ldacn_o' and 'syncn_o' for load DAC and sync signals governing the writing operation, 'clrn_o' and 'dac_resetn_o' for clear and reset signals, and 'sdi_o' for serially inputting the data to be written to the DAC. The IP blocks 'HSSL_reader_mealyFSM_1' and 'HSSL_reader_mealyFSM_2' include FSMs that decode the data received from the two interferometric heads. Each of these blocks has an input clock ('clk_i') from the PS running at 50 MHz, a reset signal ('resetn_i'), and external ports for data input, error signals, and clock inputs, which are directly interfaced with the interferometer.

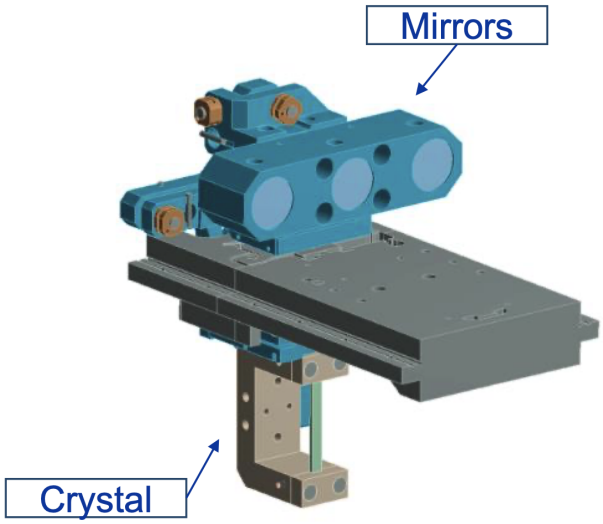


Figure 4.4: CAD rendering of the crystal assembly, showing how the retro-reflectors (mirrors) are attached to the C-shaped crystal holder, from [17].

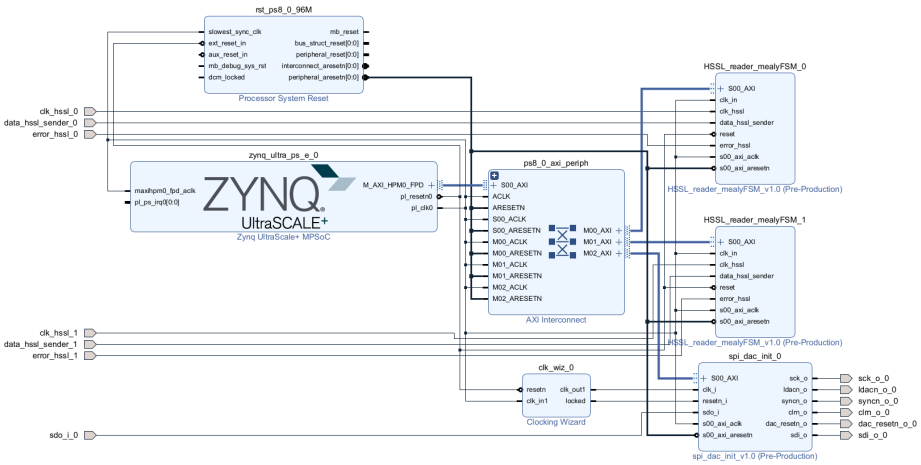


Figure 4.5: Vivado block design featuring the Zynq Processing System (PS), which provides clock and reset signals to the 'AXI Interconnect' Intellectual Property (IP) block. The AXI Interconnect links the PS to the custom 'spi_dac_init' IP block, which contains the finite state machine (FSM) responsible for managing DAC write operations. Additionally, the design includes the 'HSSL_reader_mealyFSM_1' and 'HSSL_reader_mealyFSM_2' IP blocks, which decode the data received from the two interferometric heads.

4.1.2 Digital to analog converter firmware

The firmware to control the writing operation to the DAC has been custom developed for this work and implemented in the IP block 'spi_dac_init'. This allows communication with the target device using the serial peripheral interface (SPI). The IP block incorporates the Finite State Machine (FSM) and represents the control logic for SPI communication with an AD5362

DAC. This FSM is implemented as a process sensitive to the input clock (‘clk_i‘) and reset signal (‘resetn_i‘). The main purpose of this FSM is to transmit data to a slave device (the DAC) using the SPI protocol, and it incorporates a synchronous DAC update.

As formulated in [30], the AD5781's mode of operation is governed by the 'control register,' a 24-bit wide register (refer to Figure 4.6). To initiate a write operation, the device undergoes an initialization process involving the writing of values '001' to bits 23 to 20 and '00 0001 001' bits 9 to 1 within the 'control register.' Subsequently, the actual write operation to the DAC register (refer to Figure 4.7) takes place. In this operation, bit 23 is set to 0 to indicate the write option. Bits 22 to 20 are configured as '001' to denote the specific register address, while bits 19 to 2 represent the data bits. Bits 1 and 0 are disregarded and can take any value. The transfer function linking the data word written to the 'dac register' and the resulting output voltage is expressed as follows:

$$V_{out} = \frac{V_{ref,p} - V_{ref,n} * D}{2^{18} - 1} + V_{ref,n} \tag{4.1}$$

where:

- $V_{ref,p}$ is the positive voltage applied at the VREFPx input pin;
- $V_{ref,n}$ is the positive voltage applied at the VREFNx input pin.
- D is the 18-bit data word programmed to the DAC.

MSB															LSB
DB23	DB22	DB21	DB20	DB19...DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
R/$\overline{W}$	Register address			Control register data											
R/$\overline{W}$	0	1	0	Reserved	Reserved	LIN COMP				SDODIS	BIN/2sC	DACTRI	OPGND	RBUF	Reserved

Figure 4.6: Control Register, from [30]

MSB							LSB
DB23	DB22	DB21	DB20	DB19	DB2	DB1	DB0
R/$\overline{W}$	Register address			DAC register data			
R/$\overline{W}$	0	0	1	18-bits of data		X[1]	X[1]

Figure 4.7: DAC Register, from [30]

The simplified version of the Finite State Machine (FSM) that controls the DAC is depicted in Figure 4.8. The FSM begins in the 'Idle' state. From Idle, it transitions to the 'Init' state if the current input data is not equal to the previous data. In the 'Init' state, it checks if the initialization word has been sent, and if so, it moves to the 'WriteDacReg' state. In the

'WriteDacReg' state, the FSM writes data to the DAC register. It stays in this state until the data word has been sent, at which point it transitions back to the 'Idle' state.

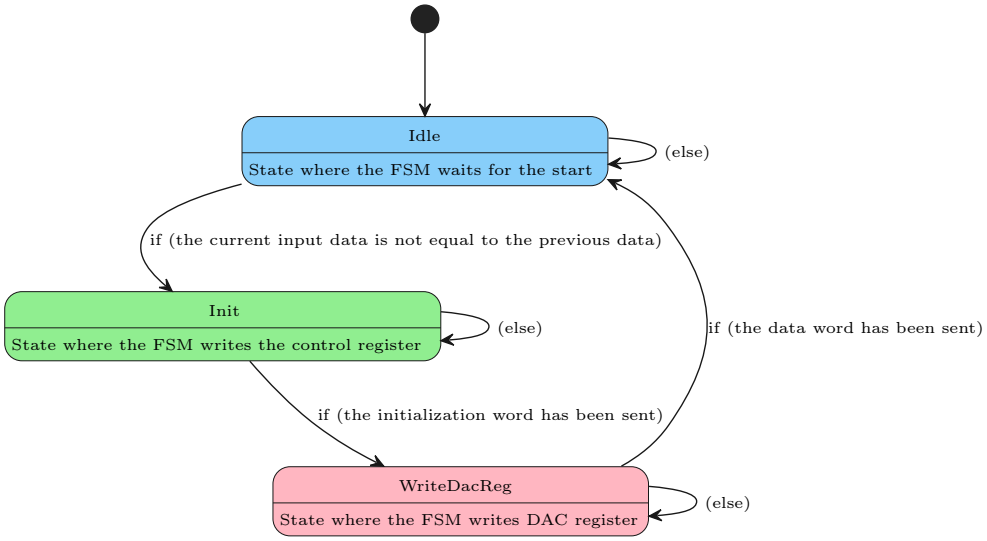


Figure 4.8: Diagram representing a simplified version of the engineered Finite State Machine for controlling the digital to analog converter.

The detailed description of the FSM states is as follows: in the Initialization State, the FSM is set to its initial configuration when a reset occurs. During this state, all relevant signals and counters are reset. The shift register, data counters, and various control signals are cleared, and the initialization signal is set high. Once the FSM moves out of the Initialization State, it enters the Data Loading State. In this state, if there has been a change in the input data, the shift register is loaded with the new initialization data. The FSM then progresses to the Data Transmission State. In the Data Transmission State, the FSM begins sending out the Most Significant Bit (MSB) of the data. The data counter is incremented with each clock cycle. If all data bits have been transmitted, the synchronization signal is activated, and the FSM transitions to the Data Finalization State. If not all bits have been sent, the FSM continues shifting the data and remains in the Data Transmission State.

In the Data Finalization State, the final data shifting is completed, and the FSM moves to the Data Preparation State. In the Data Preparation State, the shift register is loaded with the data to be transmitted, including any necessary mode bits, addresses, and additional data. The preparation signal is set high, and the FSM transitions to the Synchronization State. In the Synchronization State, the FSM synchronizes with the clock's rising edge and then moves to the Final Data Transmission State. In the Final

Data Transmission State, the FSM shifts out the MSB of the data, similar to the Data Transmission State, and then transitions to the Completion State. In the Completion State, data transmission is finalized in a manner similar to the Data Finalization State, and the FSM returns to the Idle State. For error handling, the FSM includes a default case to manage unexpected errors or additional states if needed. In case of an error or an unknown condition, the FSM will revert to the Idle State.

The Finite State Machine oversees the 'standalone operation' of the device as detailed in [30]. In this context, the initiation of the write cycle occurs with the first falling edge of SYNC. It is crucial that exactly 24 falling clock edges are applied to SCLK before bringing SYNC high again. Bringing SYNC high before the 24th falling SCLK edge renders the written data invalid. Similarly, if more than 24 falling SCLK edges are applied before bringing SYNC high, the input data becomes invalid. The input shift register undergoes an update on the rising edge of SYNC. To initiate another serial transfer, SYNC must be brought low again. After the serial data transfer, the data is automatically transferred from the input shift register to the designated register. Upon completion of the write cycle, the output can be updated by lowering LDAC while SYNC is high. The FSM initially executes an initialization stage, during which the initialization data word is transmitted on the serial line. Subsequently, the transmission of the data word commences. The testbench simulation for the described VHDL code is depicted in Figure 4.9. The LDAC signal remains in the high state during the clocking of data into the DAC input register. The DAC output undergoes asynchronous updating achieved by lowering LDAC after SYNC has been raised. The update takes place on the falling edge of LDAC. The initial word transmitted on the 'sdi_o' signal corresponds to the initialization phase, while the subsequent word corresponds to the data word.

4.1.3 Angular reading

The rotational angle of the crystal is estimated with a dual interferometer-based angular measurement system first designed in [18]. The dual-interferometer-based angular measurement system is mounted on top of the rotational stage. The reflecting surface is made of two retro-reflectors that are attached to the rotational head with the use of a support (Figure 2.17). The interferometer used in this work is an Attocube IDS3010. The communication between the reading device and the FPGA Zynq ZCU102 is managed

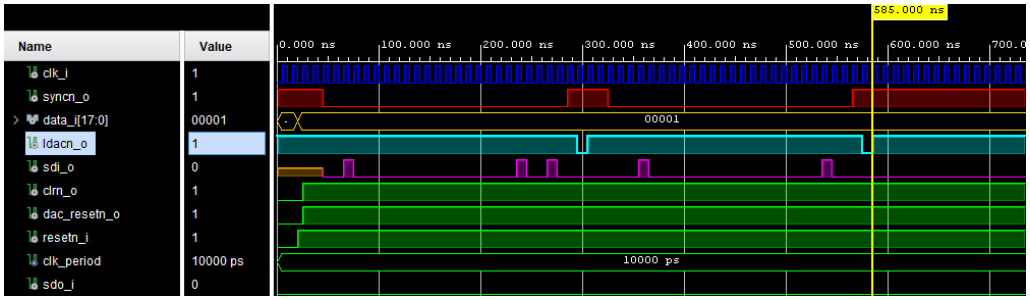


Figure 4.9: Testbench Simulation of the Digital-to-Analog Converter (DAC). This testbench demonstrates the initialization phase of the DAC followed by the data writing phase to the DAC register. The 18-bit data word to be sent to the DAC is shown as the gold signal 'data_i'. Initially, the FSM completes the initialization phase, and then the data word is transmitted serially to the DAC using the 'sdi_o' signal. The DAC output voltage becomes available after the second falling edge of the 'ldac_o' signal (represented by the sky blue line).

with the use of the HSSL (High Speed Serial Link) protocol. As formulated in [31], the HSSL protocol periodically transfers the displacement value read by the interferometric head using a binary serial format based on 1 pm resolution. The HSSL protocol is defined by its resolution, clock time, and gap. The clock time represents the bit output rate, and the resolution defines the binary bit configuration, which provides the position information. The data are synchronized with the rising edge of the CLK signal. A third sig-

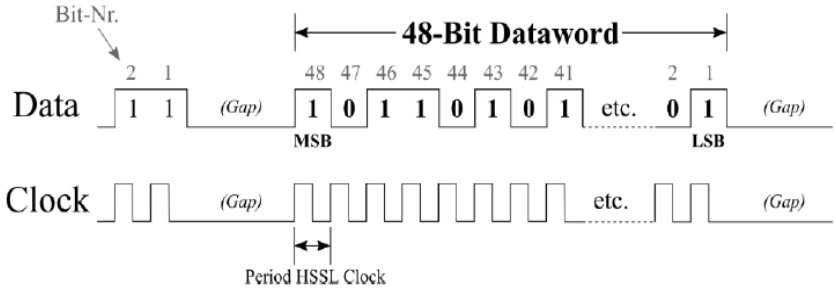


Figure 4.10: Signal pattern of a HSSL transfer, from [31]

nal indicates an error condition when the position is lost due to laser beam interruption. The HSSL word size is given by:

$$\text{Word Size} = \text{Resolution HSSL High} - \text{Resolution HSSL Low} + 1. \quad (4.2)$$

Where "Resolution HSSL High" designates the uppermost bit to be utilized in the 48-bit distance value, with the maximum value being 47. While "Resolution HSSL Low" is a parameter that defines the position resolution of the HSSL interface. The HSSL interface deals with a 48-bit distance value, and

the purpose of "Resolution HSSL Low" is to specify the position resolution by determining the lowest bit of this 48-bit distance value. The formula that expresses the relationship between the resolution and the "Resolution HSSL Low" parameter:

$$\text{Resolution} = 2^{\text{Resolution HSSL Low}} \times 1 \text{ pm} \tag{4.3}$$

Here:

- Resolution represents the position resolution.
- Resolution HSSL Low is the specified parameter defining the lowest bit of the 48-bit distance value.

The result of this calculation gives the position resolution in picometers. The lowest possible value for "Resolution HSSL Low" is 0, meaning that the resolution factor is $2^0 = 1$, and the position resolution is 1 pm. As "Resolution HSSL Low" increases, the resolution factor increases accordingly, providing finer granularity in the measurement [31].

4.1.4 Interferometer reading firmware

A tailored firmware has been specifically designed for managing the read operation from the interferometer and has been integrated into an IP block. This facilitates communication with the target device (IDS3010) via the HSSL protocol. Within the IP block, there is an integrated Finite State Machine (FSM) that serves as the control logic governing HSSL communication with the interferometer device (Figure 4.11). To effectively synchronize the clock and data signals from the interferometer with the internal clock of the board, an ‘xpm_cdc_single‘ [76] module has been used. The primary purpose of this module instantiation is to safely transfer the clock signal and data signals sent from the interferometer into the board’s clock domain, which operates at four times the frequency of the interferometer clock. Proper synchronization is essential because, without it, signals crossing clock domains can result in unreliable behavior and potential data corruption due to timing mismatches and metastability.

The detailed description of the FSM states and transitions is as follows: in the Idle State, the FSM waits for the signal indicating the start of gap counting. It transitions to the Gap Counting State when the gap-counting signal becomes active. If the signal is not active, the FSM remains in the

Idle State. In the Gap Counting State, the FSM increments the gap counter with each clock cycle. Once the gap counter has counted all the gap bits, the FSM transitions to the Data Reading State. In the Data Reading State, the FSM reads data bits from the high-speed serial link, incrementing the data bit counter with each bit read. After the required number of data bits have been read, the FSM transitions to the Output Data State. In the Output Data State, the FSM outputs the data that has been read and then transitions back to the Idle State to prepare for the next data word.

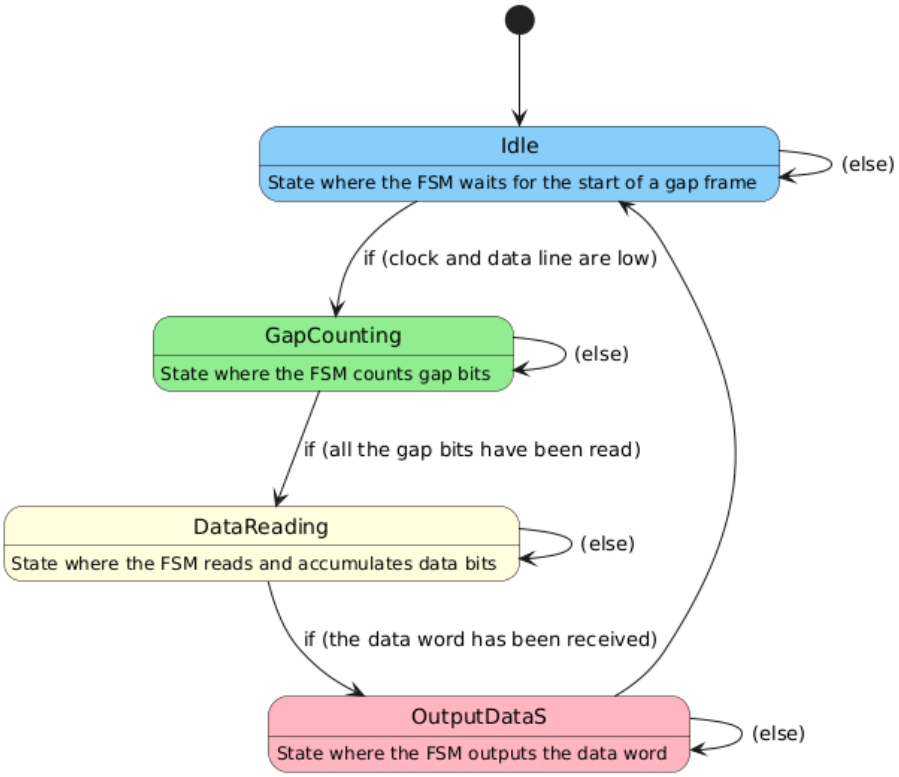


Figure 4.11: Diagram representing a simplified version of the HSSL reader Finite State Machine

The testbench simulation for the described VHDL code is depicted in Figure 4.12. As illustrated, the ‘synced_data_hssl‘ and ‘clk_hssl_sync‘ signals (represented by red lines) are interspersed with gap bits, during which both signals are held low. The ‘synced_data_hssl‘ and ‘clk_hssl_sync‘ are signals received from the interferometer and synchronized with the internal clock, ‘clk_in_tb‘, which operates at four times the frequency of the received clock. The FSM begins reading the word sent by the interferometer upon detecting the first rising edge of ’clk_hssl_sync’, transitioning from the ’gap counting’ state to the ’data reading’ state. Once all bits have been read, the data is output (magenta signal). Figure 4.13 provides a close-up view of the end of the data word transmission, highlighting the last rising

Figure 4.12: Testbench simulation illustrating the FSM reading a data word from 'synced_data_hssl' upon detecting its first rising edge, transitioning through states and ultimately outputting the data (magenta signal).

edge of the 'synced_data_hssl' signal. Following this, the FSM transitions from the 'data reading' state to the 'output data' state before returning to the 'idle' state.

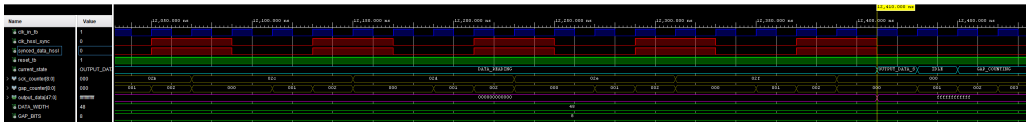


Figure 4.13: Close-up view of the end of the data word transmission. The last rising edge of the 'synced_data_hssl' signal is highlighted, showing the FSM transitioning from the 'data reading' state to the 'output data' state before returning to the 'idle' state.

To validate the developed firmware, the distances from the interferometric heads were sampled at a rate of 2 kHz over 60 seconds. Figure 4.14 presents the interferometric readings from both heads, with each subplot representing the signal from one head. The x-axis shows time in milliseconds, while the y-axis represents the corresponding interferometric readings in picometers (pm). Small fluctuations in the readings, attributed to environmental effects on the system, are observed. This confirms that the firmware accurately captures the data transmitted by the interferometer at the specified frequency.

4.2 Software application for interferometry reading, piezo stack actuation, and graphical user interface communication

Vitis [77] is a unified software platform by AMD that allows developers to integrate and interface firmware designed in Vivado with higher-level C and C++ applications, enabling accelerated development on adaptive SoCs and FPGAs. The Vitis application developed in this work has been engineered to streamline interaction with the VHDL firmware described earlier.

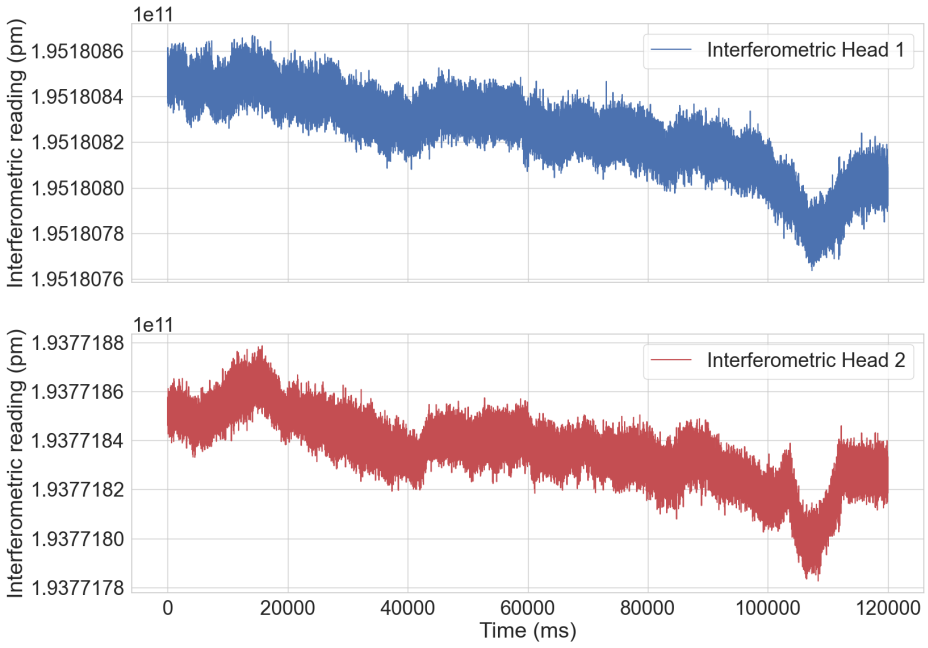


Figure 4.14: Interferometric readings from two heads sampled at 2 kHz as a function of time. The top plot corresponds to Interferometric Head 1, while the bottom plot shows the readings from Interferometric Head 2.

This involves implementing a set of functions in C++ to interface with FPGA registers, enabling the acquisition of interferometry data, generation of voltages for controlling piezo stack movements, and communication with a graphical user interface. These functions play a crucial role in coordinating system operations, ensuring precise execution of tasks critical for system control and data processing. The software structure is shown in Figure 4.15, featuring a block representing the graphical user interface, which communicates with the Vitis application through a Universal Asynchronous Receiver-Transmitter (UART), and another block illustrating communication with the FPGA.

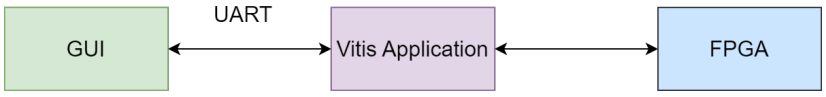


Figure 4.15: Diagram illustrating the software structure responsible for interferometry data acquisition, piezo stack control, and communication with the graphical user interface (GUI). The Vitis application, developed in C++, communicates with the GUI via Universal Asynchronous Receiver-Transmitter (UART), enabling data exchange and interpretation of commands. Additionally, the C++ application handles read and write operations with FPGA registers and manages interrupts for precise system control.

A key aspect of this coordination is a callback function, triggered by timer interrupts at 2 kHz, which ensures the timely execution of critical tasks such as interferometer head readings and DAC voltage updates. Based on displacement measurements, the current angle of the crystal is computed using the algorithm developed in [56], and the required DAC voltage is calculated. This voltage is then written in an FPGA register, which the FSM described in Section 4.1.2, transfers it to the DAC, enabling analog output for the control system.

During the interrupt service routine, data processing and communication tasks are also executed. Specifically, the function updates a JSON object with relevant data for transmission to the GUI. Additionally, data from the GUI is received via the UART interface, and upon reception, the application interprets the commands for further processing.

In conclusion the designed Vitis application orchestrates various tasks including data acquisition, control system computation, DAC update, data processing, and communication with the GUI via UART, ensuring smooth operation of the system in response to timer events.

4.2.1 Vitis application code structure

The Vitis application developed in this work (Figure 4.16) is structured within a header file named ‘utils.hpp‘. This header file serves as a central repository for utility functions, global variables, and configurations essential for various aspects of the application, including UART communication, interferometer data processing, and control logic implementation.

Within ‘utils.hpp‘, two important shared objects of different classes are declared: the logging object is a pointer to an instance of the ‘Logging‘ class. Its main role is to handle logging operations related to interferometer readings. The GUIDataHandler Object is a shared pointer to an instance of the ‘GUIDataHandler‘ class. This object is designed to manage GUI-related data operations, handle messages between the application and the GUI, and facilitate data communication with the GUI. Notably, during its initialization, it receives a shared pointer to the ‘Logging‘ class as a dependency, allowing the ‘GUIDataHandler‘ object to seamlessly use the logging capabilities provided by the ‘Logging‘ class. The ‘utils.hpp‘ header file plays a pivotal role in organizing and centralizing utility functions and configurations essential for the Vitis application. By declaring instances of ‘Logging‘ and ‘GUIDataHandler‘ classes as shared objects within this file, it ensures efficient and modular handling of logging and GUI-related

functionalities throughout the application.

Figure 4.16: Diagram illustrating the interdependencies among the header file 'utils', Logging, and GUIDataHandler classes in the vitis application system. 'utils' integrates with both Logging and GUIDataHandler, where Logging manages logging functionalities and GUIDataHandler handles data interpretation and control modes.

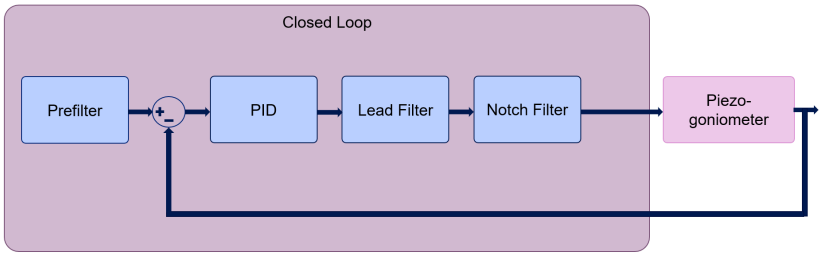


Figure 4.17: Piezo-goniometer control diagram, from [32].

4.2.2 Piezo goniometer controller

The piezo goniometer controller was developed using MATLAB design tools, as described in [56]. At the forefront of the control loop architecture shown in Figure 4.17 there is a prefilter—a second-order low-pass filter designed to attenuate high-frequency components in the system. This prefilter also stabilizes the step response in closed-loop[1] configurations by reducing oscillations and improving system dynamics. Central to the control loop is a Proportional-Integral-Derivative (PID) [78] controller, which regulates the system's output angle. A notch filter is incorporated to eliminate resonant frequencies, while a lead filter is added to enhance controller stability by increasing the phase margin.

To integrate this control system, the firmware was developed in Simulink [79] and then translated into C code using the Embedded Coder [80], which was deployed in the Vitis application presented in this work in Section 4.2 and implemented on the final target hardware.

4.2.3 Graphical user interface

To enhance the user interaction with the Vitis application and facilitate precise control of the piezo goniometer actuation, a Graphical User Interface (GUI) has been developed using QT [81] with PySide6 [82]. The GUI communicates with the Vitis application which integrates in turn the controller developed in [56] via serial communication, exchanging JSON files. Each button in the GUI triggers an action that sends a JSON file to the Vitis application, where it is processed by the GUIDataHandler class (as detailed in Section 4.2.1).

The GUI is structured into four tabs: Controls, General Settings, Control Settings, and Expert Procedures.

[1] A closed-loop system, uses feedback to continuously monitor and adjust its performance. It measures the output and compares it to the desired input or setpoint.

Controls

The "Controls" tab (Figure 4.18) of the graphical user interface (GUI) facilitates interaction with the Vitis application. Organized into distinct sections, each corresponding to specific functionalities, the tab is structured as follows:

- Control Mode Section: This section allows selection between different operational modes of the crystal controller ("Standby", "Open Loop", "Closed Loop") using a dropdown menu. Changes in mode are dynamically handled by the application;
- Motion Control Section: Users can set and adjust the goniometer's setpoint[2] through a precision input box, choosing between "Absolute" and "Relative" motion modes. Buttons for starting and stopping motion ensure precise control over actuation.
- Acquisition Section: Real-time data acquisition is displayed here, including the current controller state, yaw angle in microradians (urad), and yaw angle standard deviation in urad. These values are continuously updated as the goniometer operates.
- Errors Section: Indicators for various error conditions are visually represented using LED panels. This includes alerts for "Head Error 1", "Head Error 2", and an indicator for an unstable interferometer condition.
- Error Reset Section: To manage errors effectively, a button is provided to reset control and interferometer protections, ensuring operational safety and reliability during use.
- Serial Connection Section: This section offers functionality to establish and terminate serial communication with the Vitis application. Buttons labeled "Connect" and "Disconnect" facilitate the initiation and termination of serial connections, enhancing the interface's responsiveness and usability.

Through its structured layout and interactive elements, the "Controls" tab streamlines user interaction with the Vitis application, to provide comprehensive control and monitoring capabilities over the piezo goniometer system.

[2]The setpoint is the reference point that the controller uses to compare with the current state or output of the system to determine if corrective actions are needed.

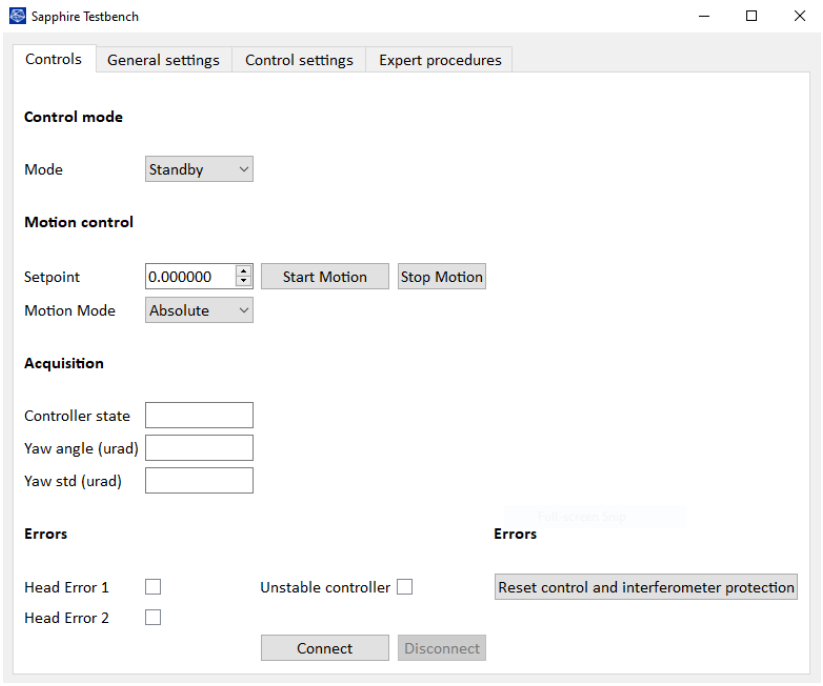


Figure 4.18: The GUI 'Controls' tab offers functionality, from selecting operational modes to precise motion control and real-time data acquisition.

General settings

The "General Settings" tab (Figure 4.19) of the graphical user interface (GUI) provides a comprehensive set of controls and displays for configuring and monitoring various operational parameters of the system. It is structured into several key sections:

- Offset section: This section allows the user to set the yaw offset and the Angular Alignment Reference (AAR) offset in picometers.
- Protection Layers Section: This section includes a checkbox that enables or disables the control instability protection feature, ensuring system stability.
- Software Limits Section: section includes fields for setting the minimum and maximum operational voltage, as well as controlling the maximum speed in both open loop mode (measured in volts per second) and closed loop mode (measured in microradians per second). Additionally, there are inputs for configuring the minimum and maximum PID limits.
- For settings management, there are buttons to read the current settings from the Goniometer controller code and to write the adjusted settings back to the system.

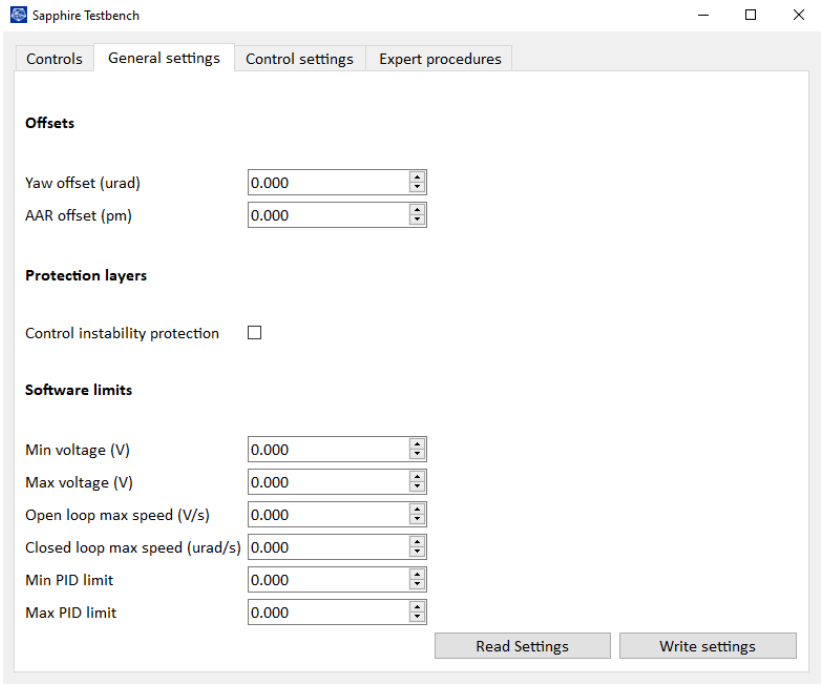


Figure 4.19: The "General Settings" tab in the graphical user interface (GUI), offering a range of controls and displays for configuring and monitoring system parameters.

Control settings

The "Control settings" tab (Figure 4.20) of the GUI includes various elements for configuring prefilters, filters, and hysteresis compensation. Filter settings are divided into three sections, each allowing users to specify numerator and denominator values[3], similar to the prefilter settings. The hysteresis compensator section includes a checkbox for enabling or disabling hysteresis compensation and fields for setting the compensation offset and various quadratic, f, and k parameters. As outlined in [56] at the moment this last functionality has not been implemented in the control algorithms. Finally, at the bottom, there are buttons for reading the current settings from the system and writing the adjusted settings back to the system.

Expert procedures

The "Expert procedures" tab (Figure 4.21) of the GUI is divided in sections:

- Profile Motion Section: Allows the user to select a waveform ID. Each waveform ID corresponds to a specific predefined motion profile or

[3]Numerator and denominator values refer to coefficients that define the mathematical transfer function of the filter.

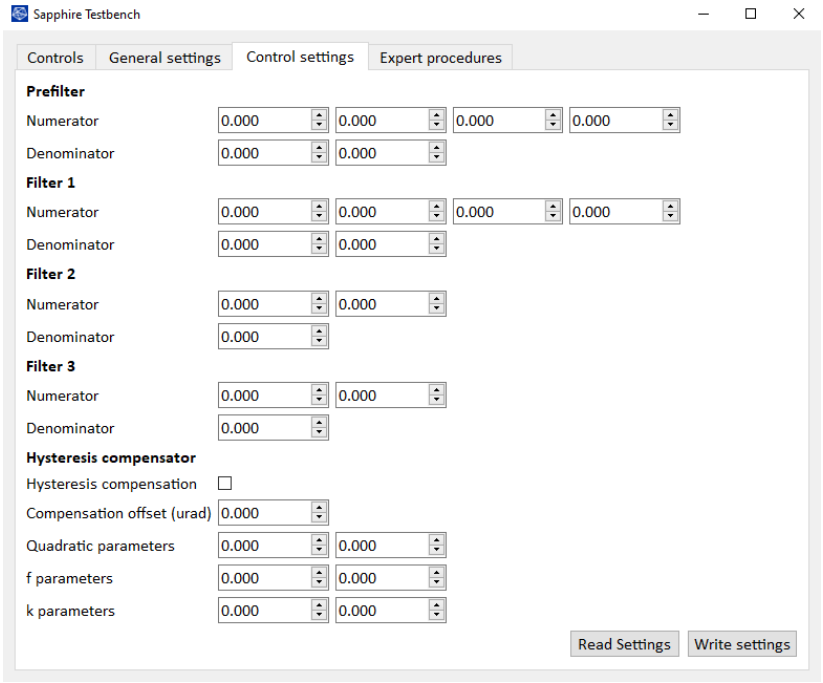


Figure 4.20: The "Control Settings" tab in the GUI, featuring various configuration options for prefilters, filters, and hysteresis compensation.

pattern of movements that the system will execute. To start and stop the motion based on the selected waveform ID.

- Ramp Cycles Section: This section allows the user to configure the ramp waveform, which is a type of signal that linearly increases or decreases over time. It gives the user precise control over the behavior of the ramp waveform, allowing for tailored system warm-up, testing, or characterization by adjusting how quickly and how many times the system's input is ramped up or down. Specifically, the user can:
  - Input the Number of Cycles: This determines how many times the ramp waveform will repeat. A ramp cycle refers to a single complete execution of the ramp, where the signal smoothly increases (or decreases) according to the specified ramp rate. Multiple cycles mean the ramp will repeat this process multiple times, which can be useful for extended warm-up periods or repeated testing.
  - Input the Ramp Rate: The ramp rate, or slope, controls how quickly the signal changes during each cycle. A higher ramp rate results in a steeper slope, meaning the signal will increase or decrease more rapidly. Conversely, a lower ramp rate creates a gentler slope, with the signal changing more slowly over time.

- Start/Stop Motion: Includes buttons to initiate the ramp cycles motion and profile motion with the specified waveform ID number of cycles and ramp rate.

- Logging: It also includes buttons to starts the logging process of the interferometric heads for 60s and save the log in a csv file.

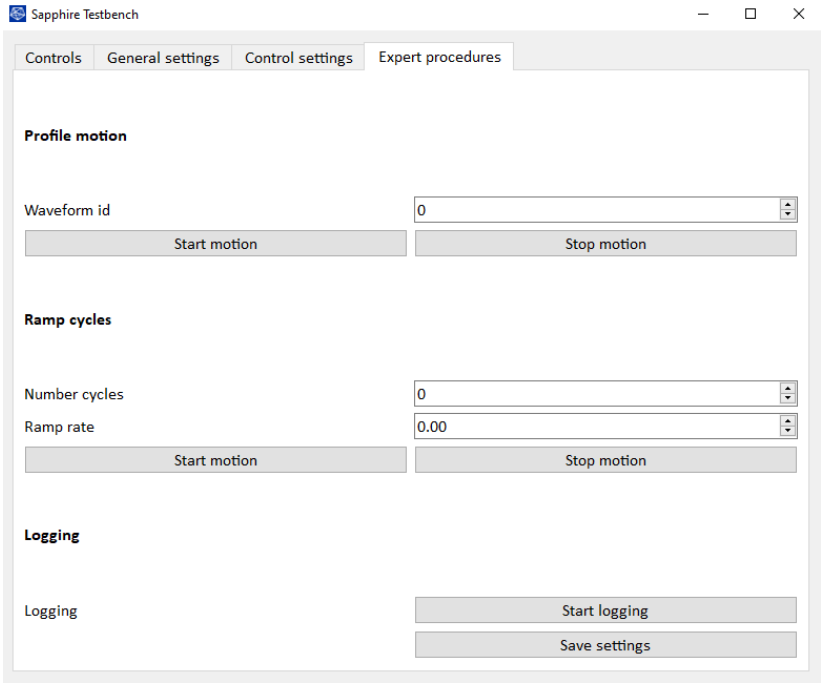


Figure 4.21: The "Expert Procedures" tab in the GUI, showcasing various sections for advanced system control. The Profile Motion Section allows users to select and control predefined motion profiles via waveform IDs. The Ramp Cycles Section enables configuration of ramp waveforms, where users can specify the number of cycles and ramp rate for tailored system warm-up, testing, or characterization. The Start/Stop Motion section provides controls to initiate or halt the configured motions, while the Logging Section allows users to start logging data from interferometric heads and save the results in a CSV file for further analysis.

Chapter 5

Enhancing crystal alignment and monitoring through machine learning-based optimization

This chapter presents the development of machine learning models aimed at improving the setup and supervision of crystal collimators in the LHC operation. Specifically, the work also published in [4] introduces a deep learning framework based on 1D Convolutional Neural Networks (CNNs) designed to optimize the search for crystalline planes, a task known as the "angular scan," described in Section 2.2.5. This problem is framed as a time-series classification task, falling within the domain of supervised learning [83]. Additionally, an updated study of [84] is presented, focusing on enhancing the monitoring of crystal collimation by developing a tool to detect deviations from the ideal channeling orientation during beam operations. Such deviations may occur due to undesired crystal movements or fluctuations in beam dynamics, and the ability to adapt to and correct these changes is critical for maintaining consistent and stable crystal collimation during LHC operations.

In summary, this chapter explores the development of machine learning tools to assist in the setup and monitoring of beam-intercepting devices, paving the way for a more automated and efficient generation of particle

accelerators.

5.1 Crystal alignment optimization

The manual process of aligning the crystal with the circulating beam envelope is labor-intensive and consumes valuable machine time, often requiring hours that could be spent on physics experiments. To address this challenge, this work aims to enhance the alignment process by automating the classification of channeling signatures using deep learning. This approach promises to significantly improve both the efficiency and accuracy of identifying the optimal channeling orientation compared to traditional manual methods [4].

5.1.1 Dataset overview

In this study, the dataset used for training and evaluating the machine learning model consists of 1689 sets labeled from hundreds of angular scans conducted by collimation experts. To increase the dataset's variability and enhance its generalization across various scenarios, a data augmentation technique was applied. This technique involved incorporating signals from multiple beam loss monitors (BLMs) positioned around the ring during angular scans. These signals helped in creating detection patterns relevant to the problem under study. The scans covered both beams in horizontal and vertical planes at different beam energies and with different settings of others collimators. The dataset comprises 1 Hz BLM signals collected during machine development studies with proton and Pb ion beams from 2015 to 2022. Each set includes two time-series signals: one recorded near the crystal and the other near the secondary collimator. The signals includes scans with a variety of rotation speeds, ranging from 0.5 μrad/s to 2 μrad/s, to ensure a more comprehensive dataset.

The BLM signals were divided into a main dataset for training and a validation set (approximately 20% of the main dataset) for testing the model on randomly selected, unseen data. Both datasets were categorized into three classes. Table 5.1 provides a detailed breakdown of the dataset composition used in the study, showing the distribution of signal counts across different classes within both the main and validation sets [4].

A signal belonging to the "Channeling Well" class (Figure 5.1) is a BLM signal that exhibits the channeling signature described in Section 2.2.5. In

Table 5.1: Composition of Datasets, from [4]

Class	Dataset	Signal Counts
"Channeling Well"	Main Dataset	530
	Validation Set	125
"No Channeling"	Main Dataset	354
	Validation Set	61
"Partial Well"	Main Dataset	509
	Validation Set	110

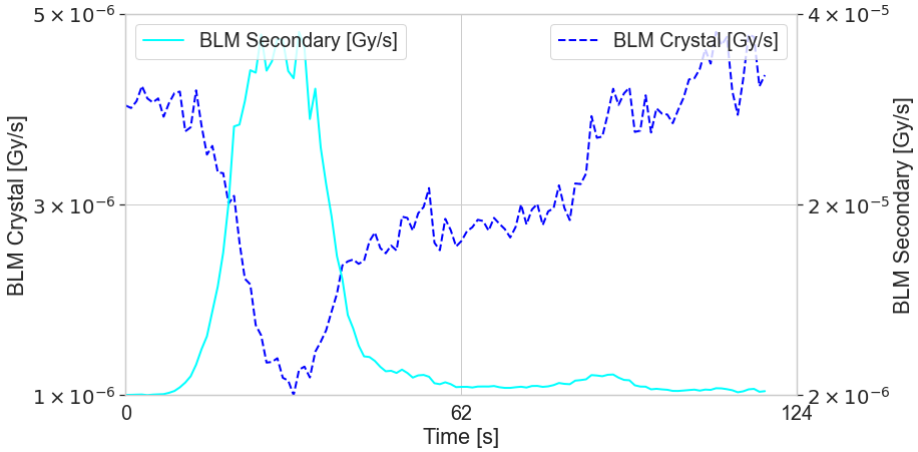


Figure 5.1: BLM signals registered in correspondence of crystal and secondary collimators during angular scan identified as "*Channeling Well*".

contrast, a signal categorized as "No Channeling" (Figure 5.2) lacks a discernible channeling pattern, while a signal classified as "Partial Well" (or "anomalous channeling") can be seen in Figure 5.3 and it displays the channeling pattern but not the volume reflection pattern. One of the objectives of this work was to avoid the incorrect identification of a "partial well," which refers to additional symmetries known as Skew Planes (SK). These planes appear diagonally relative to the primary planes used for channeling. While SK can trap charged particles, they do so with lower efficiency due to the reduced potential well they generate and the smaller deflection angle compared to channeling. This reduces the overall channeling efficiency of the crystal collimation concept [4, 11].

5.1.2 Convolutional Neural Network model design

According to [85], 1D-Convolutional Neural Networks (1D-CNNs) have recently been introduced and have rapidly achieved state-of-the-art performance across various applications, including personalized biomedical data classification, early diagnosis, structural health monitoring, anomaly de-

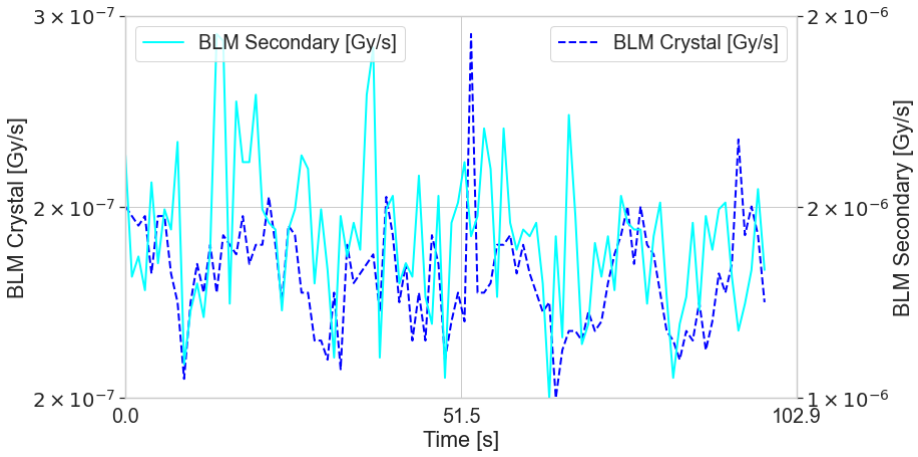


Figure 5.2: BLM signals registered in correspondence of crystal and secondary collimators during angular scan identified as "*No Channeling*", from [4].

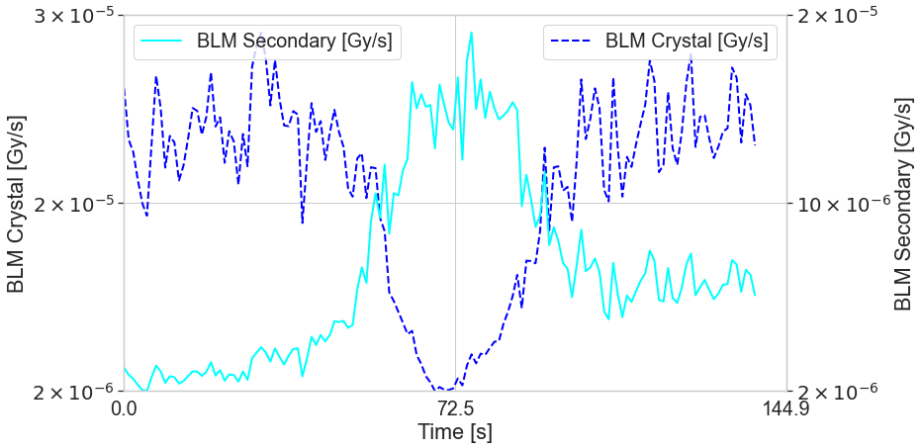


Figure 5.3: BLM signals registered in correspondence of crystal and secondary collimators during angular scan identified as "*Partial well*", from [4].

tection, and fault identification in power electronics and electrical motors. Other studies, employing diverse approaches and focusing on distinct machine subsystems, also highlight the promising potential of machine learning [86, 87]. In addition, in [3, 88], a new generation of machine learning models based on Long Short-Term Memory (LSTM) - Recurrent Neural Networks (RNNs) [89] is introduced. These models aim to improve beam loss monitor (BLM) classification to accelerate the alignment process of standard collimators, reducing the alignment time by approximately 50%. This reduction significantly decreases the time required to set up the 200 collimator jaws of the LHC. These models validate the reliability and effectiveness of deep learning for classifying BLM time series signals in complex particle accelerators like the LHC, where precision and efficiency in beam intercepting devices are critical.

As detailed in [4], the model designed for classifying BLM signals to automate the identification of the main planar channeling is a 1D Convolutional Neural Network (CNN). 1D CNNs are widely used in signal classification to analyze time-series data by learning the underlying patterns and relationships present in the data. This approach proves particularly effective with raw BLM data, as it allows the model to identify significant features without requiring pre-processing or feature extraction. Additionally, 1D CNNs are well-suited for real-time applications and are cost-effective due to their straightforward and compact architecture. The model described in this study was developed using the Keras deep learning library [90] with a TensorFlow [91] backend. Prior to inputting the data into the first CNN layer, Z-Score normalization was applied to each signal to ensure they follow a standard normal distribution with a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$ [92]. The Z-Score for each sample x is computed as follows:

$$z = \frac{x - \mu}{\sigma}. \tag{5.1}$$

Table 5.2 in [4] offers a comprehensive overview of the 1D CNN's network design, showcasing the architecture layers and their respective output shapes. The first column outlines the sequence of layers in the network, while the "Output Shape" column provides details on the batch size (the number of training examples processed per iteration), the signal length being analyzed, and the channel dimensions of the output from each layer. These dimensions are displayed in a tuple format, with each element representing a specific dimension. The terms "Batch Size" and "Signal Length" denote that these dimensions can vary during training based on the dataset's characteristics.

The model presented in in [4] consists of two 1D convolutional layers followed by batch normalization layers, rectified linear unit activation functions, and dropout layers, with respective frequency rates of 0.75 and 0.6, adopted to reduce overfitting. In a classification problem, the output of a neural network commonly comprises a vector of scores. Each element within this vector indicates the level of confidence exhibited by the network regarding the input's association with a specific class. To achieve this, the CNN architecture aforementioned is closed by a 1D global average pooling layer and a dense layer with three output nodes, accompanied by a softmax activation function. The softmax function normalizes the vector of scores ensuring that their summation results in unity. Consequently, the network's output can be interpreted as a probability distributions. In other words, the

choice of the activation function allows the output of three different probabilities, precisely the probabilities that the time series analyzed shows a pattern compatible with Figure 5.1, Figure 5.2 or Figure 5.3.

To achieve optimal performance on the given classification task, two key techniques have been used: early stopping [93] and random search for hyperparameter tuning [94]. The early stopping technique was implemented to monitor the validation performance of the model during the training process and halt the training when the validation performance plateaued, thereby preventing overfitting. Additionally, the model underwent fine tuning. In particular, 50 sets of hyperparameters from a predefined search space have been sampled, and the CNN has been trained for each set of hyperparameters. The performances have been evaluated on a held-out validation set, and the hyperparameters that resulted in the best performance have been selected [4].

Table 5.2: Network architecture layers and output shapes, from [4]

Layer(s)	**Output Shape**
Conv1D	(Batch Size, Signal Length, 256)
Batch Normalization	(Batch Size, Signal Length, 256)
ReLu	(Batch Size, Signal Length, 256)
Dropout	(Batch Size, Signal Length, 256)
Conv1D	(Batch Size, Signal Length, 160)
Batch Normalization	(Batch Size, Signal Length, 160)
ReLu	(Batch Size, Signal Length, 160)
Dropout	(Batch Size, Signal Length, 160)
Global Avg. Pooling	(Batch Size, 160)
Dense	(Batch Size, 3)

5.1.3 Model evaluation

The performance of the model has been evaluated with the use of the precision metric:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5.2}$$

where true positives (TP) indicate the number of signals of the class that are correctly predicted by the algorithm and false positives (FP) indicate the number of signals not belonging to the class that are mistakenly classified [4].

The 1-D CNN model achieved an average precision of 93% on the unseen validation set BLM signals with proton and ion beams, which served as a

benchmark for the conducted experiment. The confusion matrix depicted in Figure 5.4 offers a comprehensive breakdown of the model's performance, facilitating precision assessment. In instances where a CNN incorrectly labels a "Channeling Well" as a "Partial Well", it may necessitate re-scanning the crystal within the previously examined range, at a reduced rotational speed to enhance the patterns detection in the BLMs signals. Despite this, such results are promising, indicating the reliability of convolutional neural networks in identifying main planar channeling, showcasing their potential for practical applications [4].

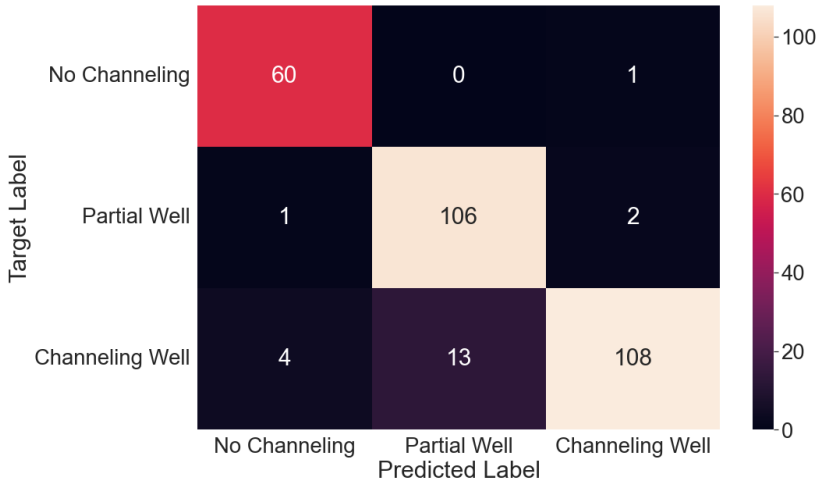


Figure 5.4: Confusion matrix obtained by classifying validation set signals, from [4].

Figure 5.5 depicts the predicted probability distributions of the CNN applied to the validation set signals, regarding the three targeted classes. These outcomes reveal that the model presents high levels of confidence in its classification performance on previously unseen validation data.

5.1.4 Proposed implementation approach

In this section, the framework to deploy the developed machine learning model into operational settings developed in [4] is presented. The machine learning model has been implemented into a software application [95] designed to streamline and consolidate all major operational requirements for crystal collimation, one of which is the identification of the correct channeling orientation. The application enables the control of the rotational stage of the goniometer and the online classification of beam loss monitors signals. In particular, the tool allows the definitions of the settings of the angular scan, such as start scan position, stop scan position and scan speed. During

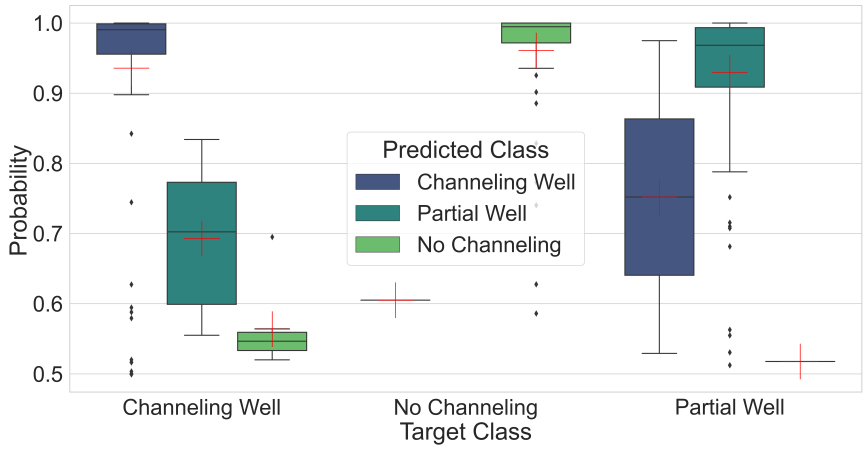


Figure 5.5: Classification probabilities distributions of validation set signals with respect to the target classes. The average classification probability is indicated by a red cross within the boxplot, from [4].

the rotation of the crystal the BLM signals of fixed length are fed into the machine learning model. The classification window (W) is determined with the following formula:

$$W\ [\mathrm{s}] = 4 \cdot \frac{\theta\ [\mu\mathrm{rad}]}{\dot{\theta}\ [\mu\mathrm{rad/s}]}, \tag{5.3}$$

where θ is the crystal bending angle (defined as $\theta = \frac{l}{R}$ where l is the crystal length in the beam direction and R its bending radius), and $\dot{\theta}$ is the speed with which the angular motion is performed.

As detailed in [4], continuous time-series data is divided into fixed-size windows by updating the classification window as the crystal rotates and BLM signals are collected. These segmented time-series are normalized and input into the trained 1D CNN model, which predicts class probabilities in real-time. After the scan concludes, a feedback mechanism uses the predicted class probabilities to guide the crystal to its channeling orientation (the position with minimal losses at the crystal location during the scan), subject to user confirmation. This confirmation acts as a crucial secondary classification step performed by the human operator, given the importance of precisely identifying the crystal's channeling orientation. Thus, the machine learning algorithm assists the operator by automating the detection of "channeling well" patterns, while the operator performs a final verification. Figure 5.6 presents a flowchart that visually represents this process. The embedded deep learning model was first successfully deployed in operations with Pb ion beams in 2022.

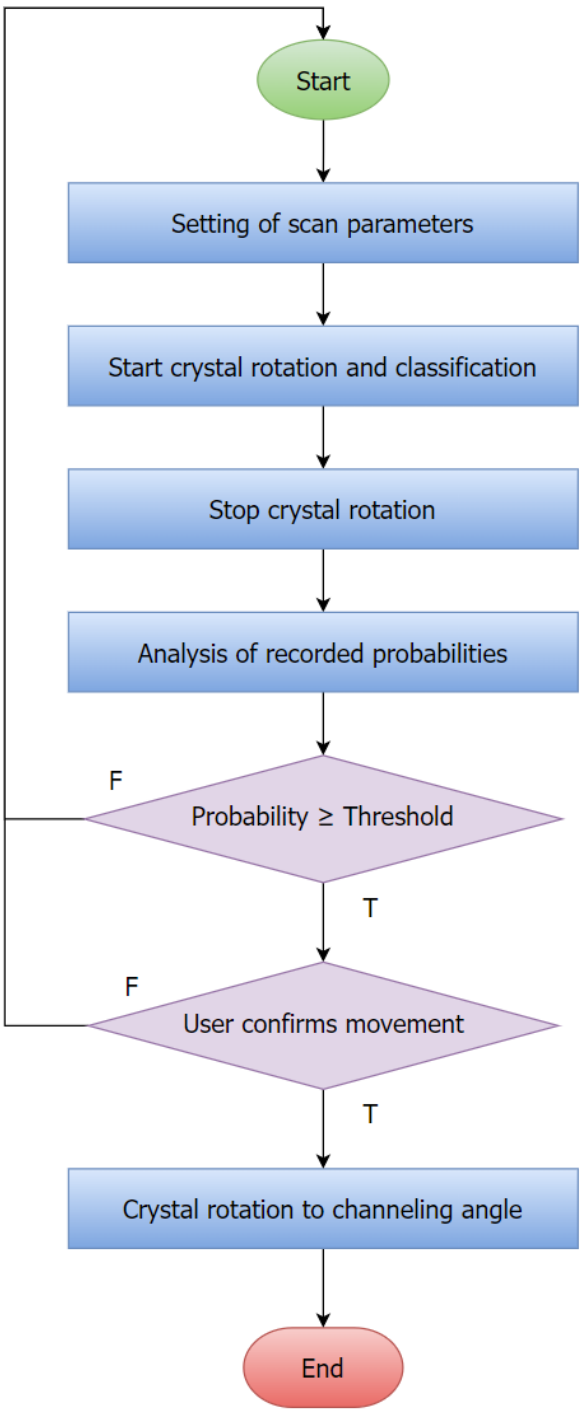


Figure 5.6: Flowchart that represents the CNN in use, from [4].

Figure 5.7 illustrates the CNN signals during an angular scan. The two distinct BLM signals, collected from sensors near the crystal and the secondary collimator, are processed by the CNN using a translating classification window (represented by vertical solid lines). The plot shows the CNN's output probabilities over time throughout the scan. Initially, the probability

that the signals correspond to a channeling well (indicated by the green line) is below 10% during the first half of the scan. As the model processes the second amorphous plateau and the classification window shifts, the probability of channeling well increases, reaching its peak. Once the "Channeling Well" pattern moves out of the classification window, the probability for "No Channeling" increases again.

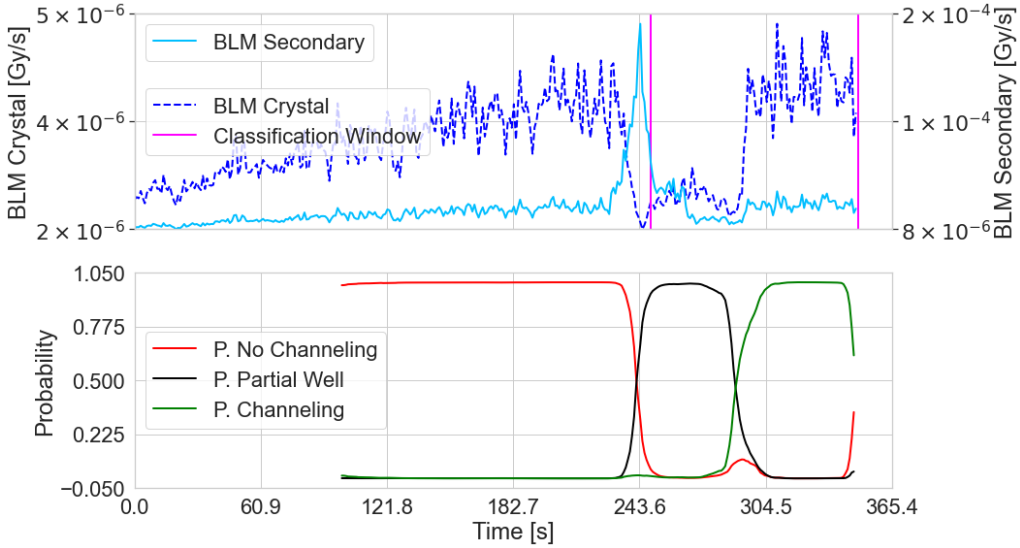


Figure 5.7: (Top) Time-series of BLM signals recorded at the crystal and secondary collimator positions, indicating a "Channeling Well," with vertical solid lines marking the classification window. The input signals for the machine learning model are continually updated as the crystal rotates, causing the classification window to shift over time. (Bottom) Output probabilities generated by the CNN as a function of time. The CNN output follows the conclusion of the classification window (marked by the right vertical solid line), from [4].

Figure 5.8 illustrates the use of the CNN in a different angular scan. At the beginning of the scan, the probability of detecting a partial well (shown by the black line) is below 10%. As the pattern moves into the classification window, the probability of a partial well gradually increases, nearing its maximum value. Once the "Partial Well" pattern exits the classification window as the scan continues, the probability of "No Channeling" rises once more.

In Figure 5.9, results obtained during ion beam operations are shown, showcasing the machine learning functionality within the custom application used for crystal operations [95]. The bottom-left plot shows the losses measured on the crystal and absorber collimators over time, with the shaded blue region indicating the data processed by the machine learning algorithm. The upper-left plot displays the model's probability estimates as a function of time. Notably, when the second amorphous shoulder pattern enters the

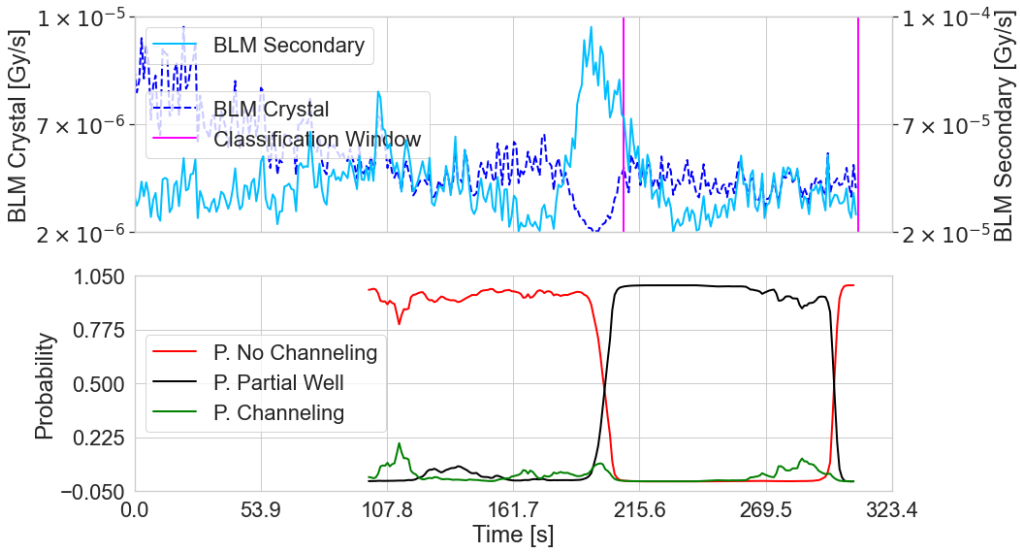


Figure 5.8: (Top) Time-series of BLM signals recorded at the crystal and secondary collimator positions, indicating a "Channeling Well," with vertical solid lines marking the classification window. The input signals for the machine learning model are continually updated as the crystal rotates, causing the classification window to shift over time. (Bottom) Output probabilities generated by the CNN as a function of time. The CNN output follows the conclusion of the classification window (marked by the right vertical solid line), from [4].

highlighted region in the bottom-left plot, the probability of the signal corresponding to channeling increases. By the end of the crystal rotation, the application has successfully identified the main planar channeling, enabling the user to proceed directly to the identified position.

5.1.5 Classification of high frequency beam loss monitor signals

Classifying high-frequency BLM signals (25 Hz) in comparison to 1 Hz BLM signals enables faster crystal rotations, significantly reducing the time required for an angular scan. For instance, Figure 5.10 shows an angular scan performed at a rate of 50 μrad/s, which is 50 times faster than the average scan speed. Even for a highly experienced operator, identifying the channeling signature would be challenging, making the use of machine learning tools essential. However, when the BLM signals are classified using a classification window of 0.9 second, the model detects a channeling-like pattern with over 80% probability (Figure 5.11). This demonstrates that the model, trained on 1 Hz BLM signals during slow crystal rotations, can successfully classify signals at higher rotational speeds and with higher-frequency BLM

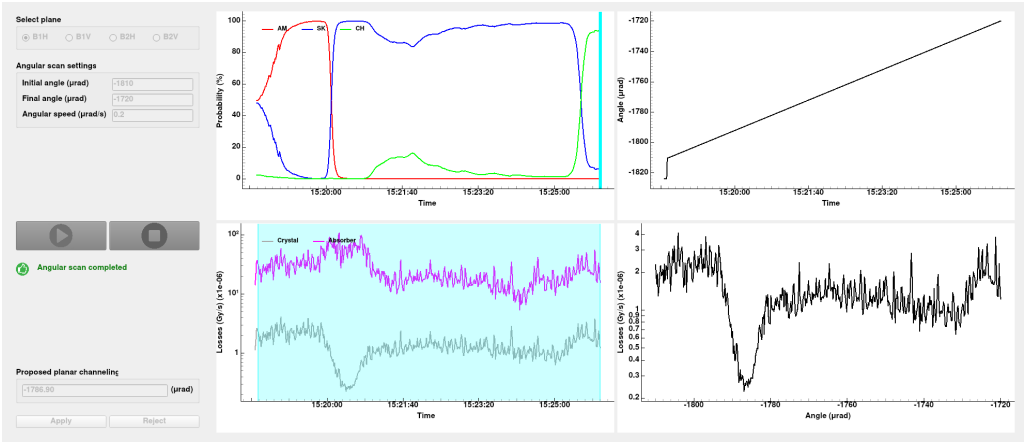


Figure 5.9: Screenshot of the machine learning application in use during ion beam operations for crystal collimation. The bottom-left plot shows the losses recorded on the crystal and absorber collimators over time, with the shaded blue region indicating the period of data processed by the machine learning algorithm. The upper-left plot displays the model's probability estimates over time, highlighting an increase in the probability of channeling as the second amorphous shoulder pattern enters the shaded region. By the end of the crystal rotation, the main planar channeling is successfully identified, enabling precise positioning. The upper-right graph depicts the crystal's rotational position over time, while the bottom-right graph shows beam loss monitor data as a function of the crystal's angle.

data.

To illustrate the time savings achieved through the classification of high-frequency BLMs during rapid crystal rotations, consider the worst-case scenario where the full angular range of the goniometer (20 mrad) must be scanned at a slow rotation rate of 1 μrad/s. In this case, the task would require 5.5 hours to complete. However, with a faster crystal rotation rate of 50 μrad/s, the same task could be completed in just 6.65 minutes. Given that a new channeling orientation must be determined each time ion beams are commissioned—currently taking approximately two operational shifts (16 hours)—the rapid setup of crystal collimators could provide significant operational benefits. Due to the limited availability of data and the high cost associated with acquiring additional angular scans at faster rotation speeds (considering that LHC operational costs exceed 100'000 CHF per hour), it is currently not feasible to verify these findings with a larger, static validation set. As a result, while this approach has the potential to reduce the time required for angular scans from hours to minutes—thereby increasing the LHC's integrated luminosity—this remains an open area for future investigation and validation.

In conclusion, this study investigates the use of machine learning to au-

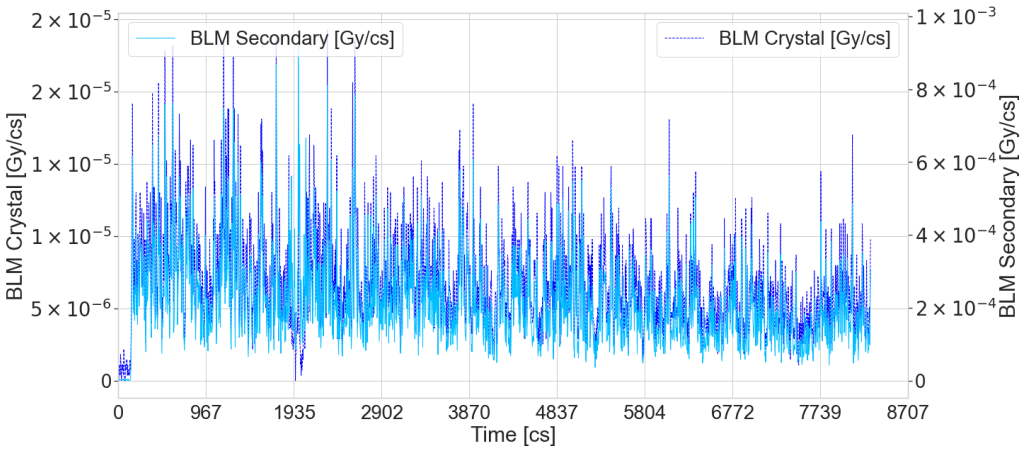


Figure 5.10: Time-series data of 25 Hz BLM signals recorded at the crystal and secondary collimator positions during rapid crystal rotations. The initial sharp increase of losses is due to the use of the ADT to increase the dispersion of particles. The high noise level in the signals makes it difficult for the human eye to determine whether the crystal is properly aligned with the beam envelope.

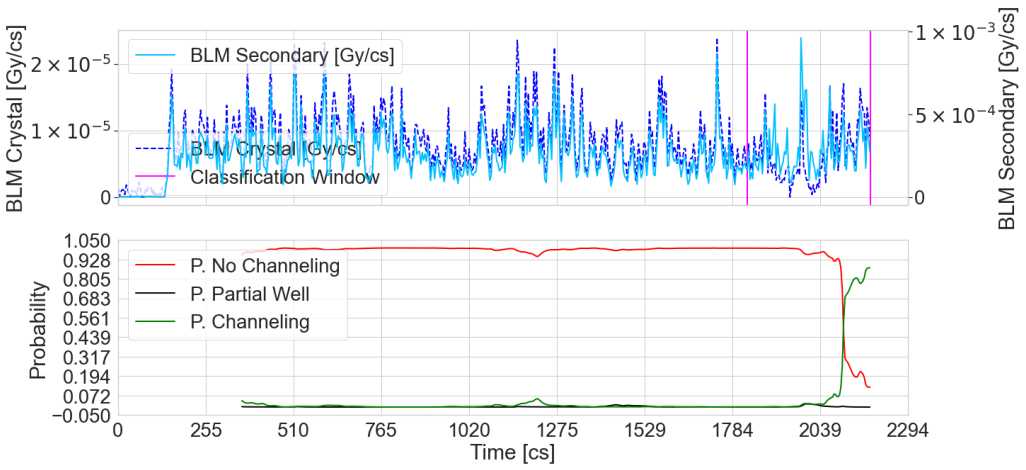


Figure 5.11: (Top) Time-series data of 25 Hz BLM signals recorded at both the crystal and secondary collimator positions, highlighting a "Channeling Well." Vertical solid lines indicate the classification window. As the crystal rotates, the input signals for the machine learning model are continuously updated, resulting in a shifting classification window. (Bottom) Output probabilities produced by the CNN over time. The CNN output aligns with the conclusion of the classification window, as marked by the right vertical solid line.

tomate the identification of main planar channeling conditions in crystal collimators during angular scans. Given the importance of operational efficiency for the HL-LHC, this research represents a significant advancement toward automating crystal angular scans, which could greatly enhance the setup efficiency of future particle accelerators. By incorporating machine learning tools into machine operations, the goal is to advance particle accelerator technology, specifically by streamlining the collimator setup process at the LHC. Furthermore, without the automation tool developed in [4], expert operators are required to manually assess crystal alignment by analyzing the BLM response. The newly developed model, however, enables less experienced operators to configure the crystal collimators by utilizing the probability outputs of the machine learning model, which reflect its confidence in classifying BLM data. This approach aligns with CERN's broader initiative to automate and streamline the use of advanced technologies.

The presented work also lays a strong foundation for future enhancements, specifically targeting the classification of higher-frequency BLM signals. Implementing this capability could enable a 25 Hz online feedback rate, compared to the current 1 Hz, thereby facilitating the detection of the channeling signature during rapid crystal rotations and reducing the time required for angular scans.

5.2 Real time crystal monitoring

Maintaining the optimal angular alignment of the crystals with the circulating beam is essential for the effectiveness of crystal collimation concept. This section addresses the problem under study with a a feed-forward neural network (FNN) has been trained using data simulated with the SixTrack-FLUKA Coupling, replicating the loss patterns observed during the 2023 ion run [84].

5.2.1 Problem formulation

As discussed in Chapter 5.1, achieving optimal performance with this innovative collimation technique requires precise angular alignment of the crystals with the beam envelope. Determining and maintaining the correct channeling orientation are highly challenging tasks.

Extensive tests were conducted at the LHC to thoroughly characterize crystal-assisted collimation before its operational deployment. However,

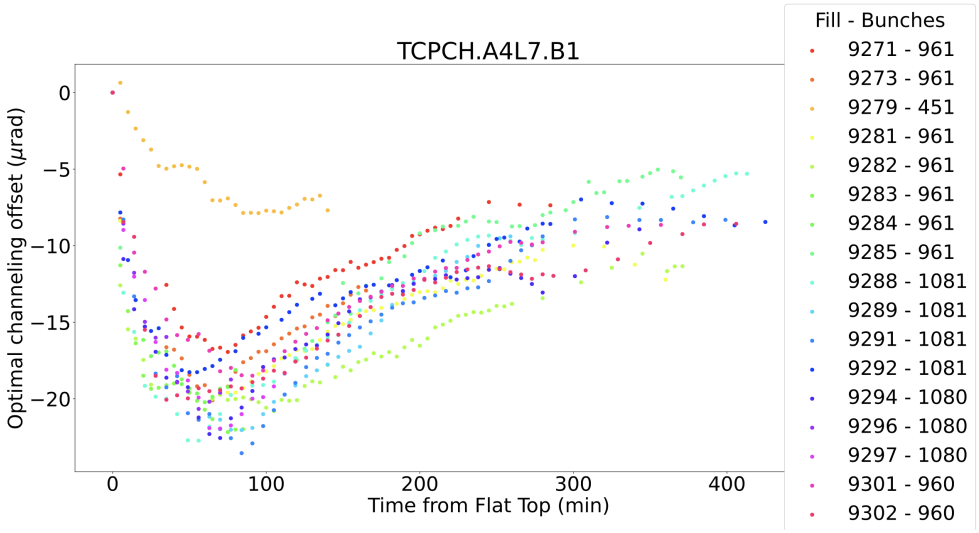


Figure 5.12: Offset from the initial optimal channeling orientation as a function of time at flat top measured in different ion physics fills at 6.8 Z TeV, from [33]

these tests were limited to Machine Development studies, carried out over short periods and with low beam intensities. During the first operational use of crystal-assisted collimation with high-intensity beams in the 2023 Pb run [33], significant stability issues were observed with the devices, making it difficult to maintain proper channeling throughout the run. All four devices (one per beam per transverse plane) were affected, but this study focuses on the horizontal device of Beam 1, as it exhibited sufficient persistent losses to allow for continuous monitoring of the crystal orientation. Figure 5.12 shows the angle adjustments needed to maintain the optimal channeling orientation over time. The crystal positioned on Beam 1 in the horizontal plane during the flat top phase (when the LHC operates at its maximum energy) exhibited significant deviations from the initial optimal alignment [84]. The initial optimal channeling orientation was established during commissioning with low-intensity beams, and deviations from this reference value were calculated using a tool [95] specifically designed to optimize angular orientation [84].

These observations led to the conclusion that real-time monitoring and classification of the crystal's state during operation are essential [84].

5.2.2 Dataset overview

The data utilized in this study are consistent with those presented in [84]. Specifically, they are based on simulated losses at various locations within

the LHC particle accelerator, which were generated through the coupling of the SixTrack and FLUKA codes [96–99]. FLUKA [100, 101], a widely utilized Monte Carlo simulation software for particle transport and interactions with matter, accurately models particle behavior as they interact with different materials. SixTrack [102, 103] enables 6D symplectic multi-turn magnetic tracking of particles. By leveraging these tools, we successfully simulated the beam's interaction with the crystal across varying energy levels. The simulations were based on the LHC conditions during the 2023 ion run. An initial distribution of 1×10^6 Pb ions in a pencil beam was configured to impact the crystal at an angle of 1rad [104]. The trajectories of the resulting particles were then tracked until they either struck the aperture or were absorbed by a collimator.

As outlined in [84], the simulation examines the crystal's interaction with the beam under three distinct conditions: channeling, volume reflection, and amorphous configurations. As shown by the green bars in Figure 5.13, when the crystal (TCPCH.A4L7.B1) is in channeling mode, it absorbs minimal energy. This is because particles that are channeled through the lattice planes experience fewer interactions with atomic nuclei within the crystal. In contrast, the secondary collimator (TCSPM.B4L7.B1) absorbs significantly more energy since it intercepts particles channeled by the crystal.

For amorphous conditions (indicated by red bars in Fig. 5.13), the energy absorbed by the crystal is higher than that observed by the secondary collimator. This is because, in amorphous mode, the crystal acts as a standard scatterer, causing more interactions between the beam and the crystal material. As a result, the energy absorbed by the secondary collimator is reduced.

In the volume reflection scenario (depicted by blue bars), the loss pattern is similar to that of the amorphous case. However, there is an additional concentration of energy deposition near the crystal and the TCSPM.6R7.B1 collimator.

This simulation-based approach for training was chosen to establish a retrainable framework for each new ion run. This strategy eliminates the need for time-consuming and resource-intensive data collection sessions at the LHC, thereby conserving time and resources. Moreover, the ability to train a machine learning model using simulations enables the development of updated models for classifying crystal states at the start of each ion run.

The features illustrated in Figure 5.13 were selected based on the following criteria:

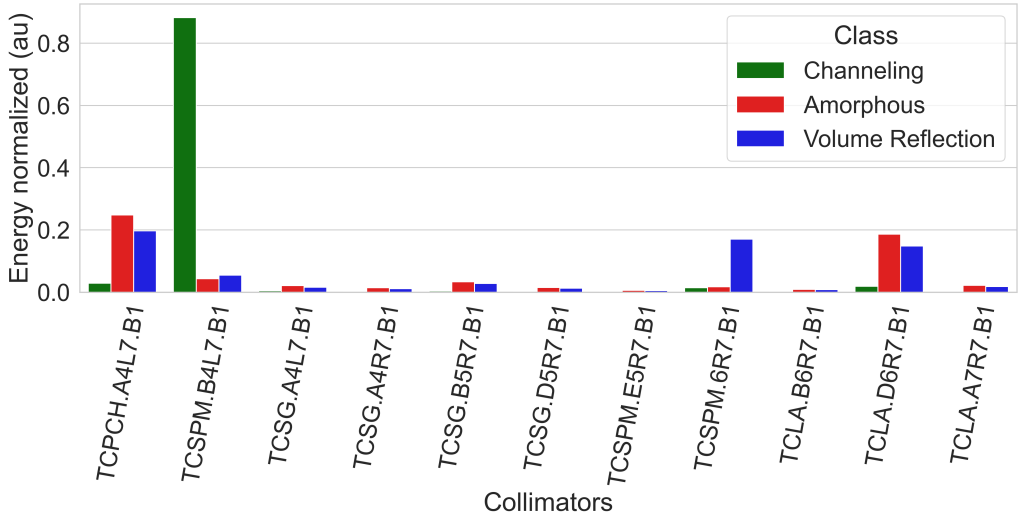


Figure 5.13: Absorbed energy distribution of impacting particles normalized to the total energy loss observed on various collimators when the horizontal crystal experiences channeling, amorphous, or volume reflection conditions on Beam 1.

- Clear Dependence on Crystal Orientation: The chosen feature must show a distinct relationship with the orientation of one of the crystals in the beam. This criterion ensures that the signal variations are strongly correlated with the specific orientation of the target crystal.
- Minimal Cross-Talk Sensitivity: The selected features must exhibit minimal interference from the other crystal in the beam. Specifically, the signal from the target crystal should be substantially greater than the signal from the other crystal, ensuring that the measurement accurately reflects the orientation of the target crystal with minimal cross-talk.

Dataset enlargement

As detailed in [84], Neural Networks (NNs) necessitate a substantial volume of diverse data to accurately identify complex patterns. To address the computational demands associated with large-scale simulations, the dataset was expanded, thereby increasing the number of examples available for training. This enlargement enhances the network's generalization and robustness. Additionally, data augmentation was employed to avoid overfitting, by providing regularization through exposure to a wider array of dataset instances. Thus, enlarging the dataset improves the performance and reliability of Deep Learning models by introducing a broader spectrum of training samples.

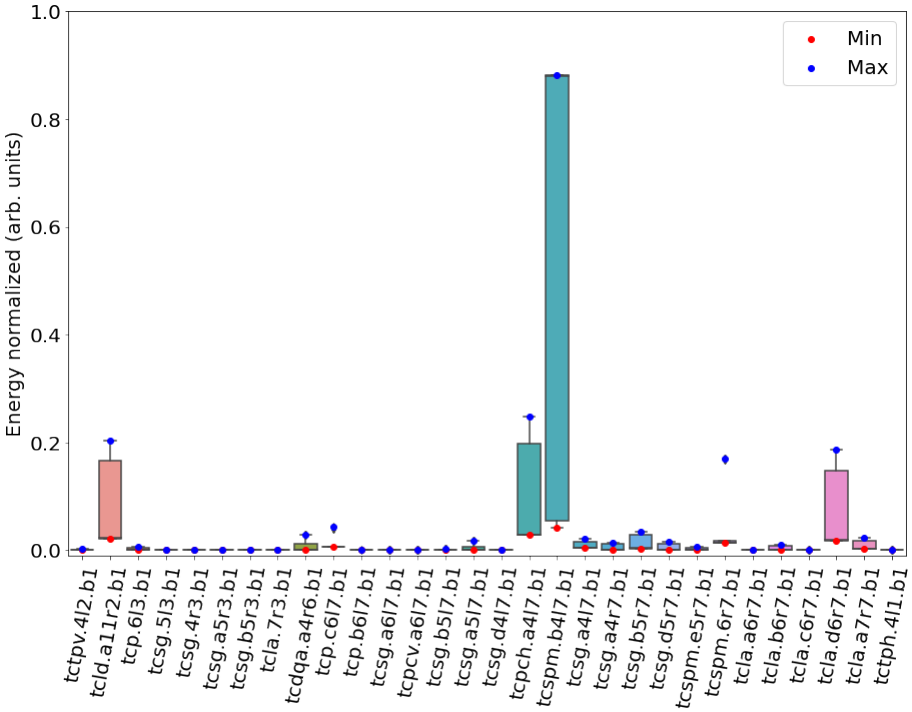


Figure 5.14: Boxplot showing the distribution of values for each simulated dataset feature. The minimum and maximum values are highlighted with red and blue markers, respectively.

To expand the dataset, random noise was introduced based on a Gaussian distribution. This noise was generated using a normal distribution defined by each feature's mean and standard deviation, with values constrained within the feature's minimum and maximum ranges. Figure 5.14 displays the value distributions of each dataset feature in a boxplot format, with the minimum and maximum values highlighted by red and blue markers, respectively.

This augmentation increased the dataset size while preserving the underlying class patterns and core relationships within the data. The dataset was then partitioned into a test set, comprising 20% of the total data (300 samples), and a training set, consisting of the remaining 80% (1200 samples).

5.2.3 The neural network design and validation on real data

The model employed in this study mirrors the one described in [84]. It is a Feedforward Neural Network (FNN) that receives as input the normalized energy measurements from key collimators situated in IR7. The FNN is

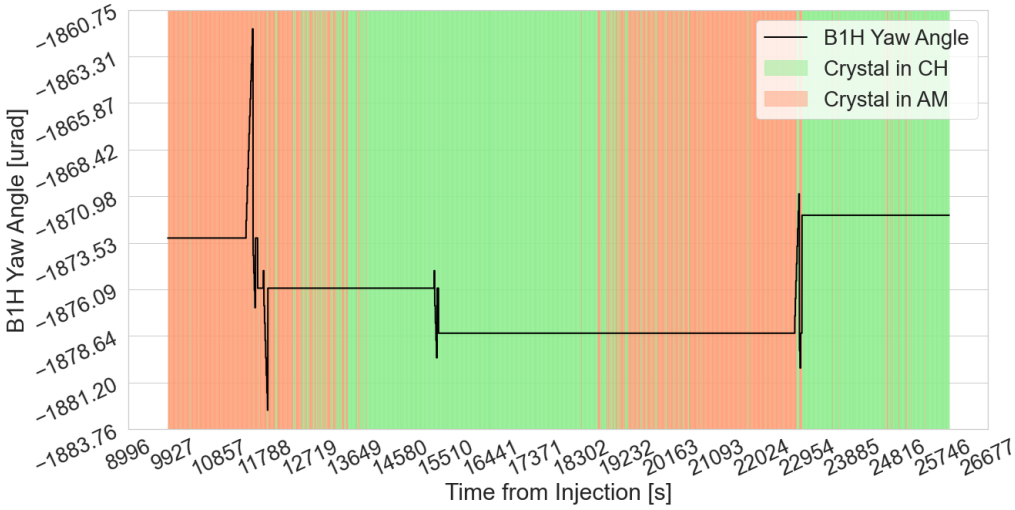


Figure 5.15: Classification of ion Physics fill number 9250 during the collision beam state. The plot shows the yaw angle of the horizontally positioned crystal on beam line 1 as a function of time. Red sections correspond to periods where the model classified the crystal as being in the amorphous state, while green sections indicate periods of channeling, adapted from [34].

designed to identify complex patterns in the dataset that correspond to the three states of the crystal. The network utilizes the Adam optimizer for training and features an architecture with three dense layers. The first layer comprises 16 units and includes a dropout rate of 0.1 to mitigate overfitting. The second layer contains 80 units and has a dropout rate of 0.5, while the third layer consists of 176 units with a dropout rate of 0.5. The architecture was optimized using a systematic random search process, consistent with the technique described in Section 5.1.2, to ensure both effective performance and generalization.

The developed model has been visually validated with real BLM data from the 2023 Ion Run. For example, Figure 5.15 shows the classification of ion Physics fill number 9250 in the collision beam state[1]. The figure displays the yaw angle of the horizontally positioned crystal on beam line 1 as a function of time. Red sections indicate periods when the model classified the crystal as being in the amorphous state, while green sections represent periods when the crystal was classified as channeling. The adjustments in the yaw angle, as shown in the plot, are due to realignment performed by an algorithm [105] designed to optimize the device's angular orientation.

As shown in Figure 5.15, two large sections are classified as the crystal being in the amorphous state. Following these, after a middle section where

[1]The collision beam state refers to the phase in which the two high-energy LHC beams collide, following the ramp, injection and flat top states.

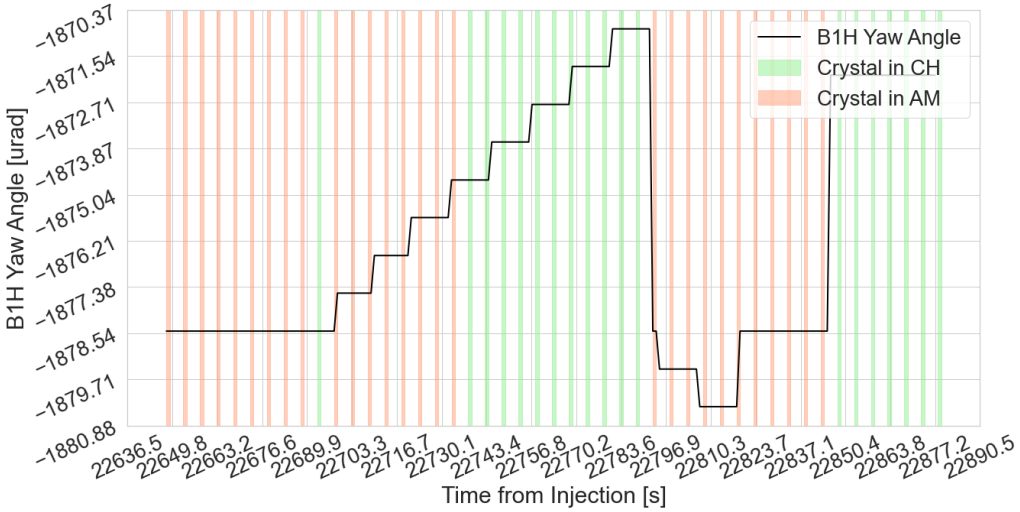


Figure 5.16: Classification of a zoomed section of the ion Physics fill number 9250 during the collision beam state. The plot shows the yaw angle of the horizontally positioned crystal on beam line 1 as a function of time. Red sections correspond to periods where the model classified the crystal as being in the amorphous state, while green sections indicate periods of channeling, adapted from [34].

the crystal was classified as channeling, there is a gradual transition back to the amorphous state. This is followed by a yaw angle adjustment and a correct classification of channeling. On a positive note, the model demonstrates good sensitivity in classifying signals during yaw angle adjustments, as seen in Figure 5.16. This zoomed-in view shows that, at the start of the third yaw adjustment, the crystal is correctly classified as not channeling. It then accurately identifies channeling when the yaw angle reaches the channeling region, followed by a classification of amorphous and then channeling once more as the yaw angle is set to the new channeling position.

In conclusion the designed model is able to classify with good sensitivity the state of the crystal during the presented physics fill. Further studies are needed to verify the model's precision with data from loss maps in the collision beam, which will serve as a ground truth validation set. However, this data has not yet been collected with the 2023 collimator settings. Additionally, model training should be expanded to include other beam states and planes to enable crystal monitoring during injection and ramp phases as well.

Chapter 6

Conclusions

The High-Luminosity Large Hadron Collider (HL-LHC) upgrade introduces significant challenges to current particle accelerator technology, particularly in the collimation system. This dissertation addressed these challenges by developing advanced crystal collimation technologies for the HL-LHC, focusing on alignment, characterization, control, and operational supervision. With operational efficiency being crucial to the HL-LHC, this work represents a major advancement in this domain.

A key contribution of this work was the design of an innovative X-ray diffractometer software, which simplifies the complex process of crystal characterization. The development of a comprehensive C++ framework and graphical user interface (GUI) enables intuitive interaction with the mechatronic system. The post-processing unit aids in data analysis and visualization during the X-ray measurements of crystal collimators. This semi-automated system significantly improves the flexibility of crystal measurements and streamlines the overall characterization process, ensuring that crystals meet the stringent standards required for effective collimation of high energy beams. Future work on this project will involve incorporating an autocollimator into the hardware setup and software, which will improve the accuracy of crystal's yaw angle readings and further increase the precision of the measurements. Furthermore, a thorough characterization of what might be sources of uncertainty in measurements is needed, including factors such as temperature, humidity, and acquisition time during scans. Understanding the influence of these parameters will further enhance measurement precision. Future development will also include meticulous scans of the crystal's pitch, roll, and yaw angles in relation to scans of the detector position to accurately identify the Bragg peak of the crystal. The software

system and algorithms are already in place to support this advanced work.

Additionally, a testbench for a new piezo goniometer controller based on field-programmable gate array (FPGA) was developed. This in-house solution offers a modular and flexible hardware and software framework tailored to the demanding motion control applications at CERN. The design of the Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) firmware was implemented and tested through simulations and real hardware integration. Furthermore, the new piezo goniometer controller was integrated into a C++ application that interfaced with the VHDL firmware, and a dedicated GUI was created to facilitate its operation and testing.

A significant aspect of this work involved the application of machine learning, particularly convolutional neural networks (CNNs), to automate and enhance crystal alignment, a crucial factor for efficient particle collimation in high-energy accelerators when these devices are used operationally in the LHC. A deep learning framework was introduced for automating crystal alignment using beam loss monitor (BLM) signals. This approach allows for the automatic identification of optimal channeling orientations, addressing limitations in manual alignment procedures. This solution has been tested in machine operations and is now part of the standard tool-set used by collimation operators. Future enhancements to the machine learning model, such as incorporating higher-frequency BLM signals, could further improve precision in classification and enable faster crystal rotations during alignments. This advancement would reduce the setup time for crystal collimators from hours to minutes and contribute to increased operational efficiency, reducing time spent out of physics and improving the machine's integrated luminosity.

The implementation of a feed-forward neural network (FNN) for real-time crystal alignment monitoring was also introduced. By tracking the beam loss monitors losses around the accelerator complex during high-energy beam operations, this system can detect misalignments of the crystal and provide corrective feedback, improving the overall performance of the collimation system. While this work lays the foundation for integrating machine learning into crystal collimation, further research is necessary. Expanding the model's training to include additional operational states of the LHC, such as injection and ramp phases, would enhance its versatility. Additionally, although extensive simulations have been performed, further experimental validation during actual LHC operations, particularly in high-intensity beam runs, is crucial to fully verify the models' performance.

In conclusion, the advancements in crystal collimation technology presented in this thesis represent a significant leap in enhancing the efficiency of this innovative collimation technique. The integration of machine learning and advanced control systems optimizes crystal alignment and lays a solid foundation for future innovations in particle accelerator technology.

Appendix A

X-Ray Diffractometer Device Finite State Machine

This section outlines the fundamental structure of the Finite State Machine (FSM) implemented for each device within the X-Ray diffractometer. While the number of states remains uniform across all FSMs, variations arise in the specific events and actions that drive state transitions for each device. Consequently, this section will provide a detailed description of the core structure of the FSM, highlighting its general framework and essential components. A breakdown of transitions and associated actions is provided below:

1. **SystemNotInitialized**
    - State Description: Motors are not connected.
    - Transitions:
        (a) To **Error**
            – Condition: `start() == False`
            – Description: If `start` returns `False`, transition to **Error**.
        (b) To **SystemConnected**
            – Condition: `start() == True`
            – Description: If `start` returns `True`, transition to **SystemConnected**.
2. **SystemConnected**

- State Description: Motors and devices are connected.
- Transitions:
  (a) To **SystemInMotion**
    – Event: `eventSetupHome`
    – Description: Triggered by `eventSetupHome`.
  (b) To **SystemNotInitialized**
    – Condition: `disconnect() == True`
    – Description: If `disconnect` returns `True`, transition to **SystemNotInitialized**.
  (c) To **Error**
    – Condition: `disconnect() == False`
    – Description: If `disconnect` returns `False`, transition to **Error**.
  (d) To **SystemInMotion**
    – Event: `eventMoveStepper`
    – Description: Triggered by `eventMoveStepper`.
  (e) To **SystemInMotion**
    – Event: `eventStartScan`
    – Description: Triggered by `eventStartScan`.

3. **SystemInMotion**

- State Description: One of the devices is in motion.
- Transitions:
  (a) To **SystemHome**
    – Condition: `goHome() == True`
    – Description: If `goHome` returns `True`, transition to **SystemHome**.
  (b) To **Error**
    – Condition: `goHome() == False`
    – Description: If `goHome` returns `False`, transition to **Error**.
  (c) To **SystemConnected**
    – Condition: `moveStepperToRelativePosition() == True`
    – Description: If `moveStepperToRelativePosition` returns `True`, transition to **SystemConnected**.

(d) To **Error**
- Condition: `moveStepperToRelativePosition() == False`
- Description: If `moveStepperToRelativePosition` returns `False`, transition to **Error**.

(e) To **SystemConnected**
- Condition: `startScanStepper() == True`
- Description: If `startScanStepper` returns `True`, transition to **SystemConnected**.

(f) To **Error**
- Condition: `startScanStepper() == False`
- Description: If `startScanStepper` returns `False`, transition to **Error**.

(g) To **SystemNotInitialized**
- Event: `disconnectMotors()`
- Description: Triggered by `disconnectMotors()`.

(h) To **SystemConnected**
- Condition: `stop() == True`
- Description: If `stop` returns `True`, transition to **SystemConnected**.

(i) To **Error**
- Condition: `stop() == False`
- Description: If `stop` returns `False`, transition to **Error**.

4. **SystemHome**

- State Description: Motors are in home position.
- Transitions:

(a) To **SystemNotInitialized**
- Condition: `disconnect() == True`
- Description: If `disconnect` returns `True`, transition to **SystemNotInitialized**.

(b) To **Error**
- Condition: `disconnect() == False`
- Description: If `disconnect` returns `False`, transition to **Error**.

(c) To **SystemInMotion**

- Event: `eventMoveStepper`
- Description: Triggered by `eventMoveStepper`.

(d) To **SystemInMotion**

- Event: `eventStartScan`
- Description: Triggered by `eventStartScan`.

5. **Error**

- State Description: Error state.
- Transitions:

(a) To **SystemConnected**

- Condition: `start() == True`
- Description: If `start` returns `True`, transition to **SystemConnected**.

(b) To **Error**

- Condition: `start() == False`
- Description: If `start` returns `False`, remain in **Error**.

(c) To **SystemNotInitialized**

- Condition: `disconnect() == False`
- Description: If `disconnect` returns `False`, transition to **SystemNotInitialized**.

Bibliography

[1] S. Redaelli, "Beam cleaning and collimation systems," *CERN Yellow Reports*, p. Vol 2 (2016): Proceedings of the 2014 Joint International Accelerator School: Beam Loss and Accelerator Protection, 2016.

[2] G. Valentino, R. W. Assmann, R. Bruce, and N. Sammut, "Classification of lhc beam loss spikes using support vector machines," *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics*, 2012.

[3] G. Ricci, A. Mostacci, M. Balsi, G. Azzopardi, and S. Redaelli, *Long Short-Term Memory Recurrent Neural Network for the Fully-Automatic Collimator Beam-Based Alignment in the Large Hadron Collider (LHC).* PhD thesis, Universitá degli Studi di Roma "La Sapienza", Rome, 3 2021. Presented 2021.

[4] G. Ricci, M. D'Andrea, M. Di Castro, E. Matheson, D. Mirarchi, A. Mostacci, and S. Redaelli, "Machine learning based crystal collimator alignment optimization," *Phys. Rev. Accel. Beams*, vol. 27, p. 093001, Sep 2024.

[5] M. D'Andrea, *Applications of Crystal Collimation to the CERN Large Hadron Collider (LHC) and its High Luminosity Upgrade Project (HL-LHC).* PhD thesis, Padua U., 2021. Presented 23 Feb 2021.

[6] "Unit cells | chemistry for non-majors | | course hero." https://www.coursehero.com/study-guides/cheminter/unit-cells/. Accessed: 2021-11-05.

[7] D. G. Arnold, T. L. Beauchamp, and N. E. Bowie, *Elements of X-Ray Diffraction.* Pearson Education Limited.

[8] A. I. Sytov, V. Guidi, V. V. Tikhomirov, E. Bagli, L. Bandiera, G. Germogli, and A. Mazzolari, "Planar channeling and quasichan-

neling oscillations in a bent crystal," *Eur. Phys. J. C*, vol. 76, p. 77, 2016.

[9] A. Fomin, *Multiple Scattering Effects on the Dynamics and Radiation of Fast Charged Particles in Crystals. Transients in the Nuclear Burning Wave Reactor*. PhD thesis, National University V.N.Karazin de Kharkiv, 2017. Thèse de doctorat Physique des accélérateurs Université Nationale de V.N Karazin de Kharkiv (Ukraine) 2017.

[10] D. Mirarchi, *Vol. 31 - Crystal Collimation for LHC. Crystal collimation for LHC*. PhD thesis, Imperial Coll., London, 2015. Presented 18 Jun 2015.

[11] R. Rossi, S. Redaelli, and W. Scandale, *Experimental Assessment of Crystal Collimation at the Large Hadron Collider*. PhD thesis, Università "La Sapienza" di Roma, 2017.

[12] M. Romagnoni, V. Guidi, L. Bandiera, D. D. Salvador, A. Mazzolari, F. Sgarbossa, M. Soldani, A. Sytov, and M. Tamisari, "Bent crystal design and characterization for high-energy physics experiments," *Crystals 2022, Vol. 12, Page 1263*, vol. 12, p. 1263, 9 2022.

[13] A. Ivanov, "Quasimosaic crystals for focusing high energy x-rays in laue diffraction geometry," in *Proceedings of the 10th International Conference on Synchrotron Radiation Instrumentation (SRI 2006)*, 2006.

[14] S. Gargiulo, "Modelling for metrological analysis of an X-ray diffractometer for bent crystals characterization in crystal-assisted collimation," 2018. Presented 20 Feb 2018.

[15] R. Stefano, R. Bruce, and A. Mereghetti, "Upgrade of the crystal collimationtest stand in ir7," tech. rep., CERN, 4 2018.

[16] P. S. Galvez, *Modelling and Control of Piezoactuators for High Precision Positioning Systems Used in Radioactive Environments*. PhD thesis, Universidad Politécnica de Madrid, 7 2019.

[17] E. Matheson, "Precision actuation & measurement: Piezo-goniometer devices," in *Indico: Mechatronics, Robotics, and Operations Section Presentations*, (774/R-013, CERN), CERN, 2022. Presented on 8 Nov 2022.

[18] P. Serrano Gálvez, M. Butcher, and A. Masi, "A dual-interferometer-based angular measurement system with absolute angle recovery method," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 3, pp. 864–873, 2019.

[19] R. Schmidt, "Machine protection," tech. rep., CERN-European Organization for Nuclear Research, 2013.

[20] M. Armstrong, "The schema frequency." https://www.armstrongeconomics.com/armstrongeconomics101/understanding-cycles/the-schema-frequency/,. Accessed: 2021-11-03.

[21] Y. Waseda, E. Matsubara, and K. Shinoda, *X-Ray Diffraction Crystallography X-Ray Diffraction Crystallography Introduction, Examples and Solved Problems*. Spinger, 2011.

[22] C. Suryanarayana and M. G. Norton, *X-Ray Diffraction*. Springer US, 1998.

[23] Wikipedia contributors, "Characteristic X-ray." https://en.wikipedia.org/wiki/Characteristic_X-ray, 2024. Accessed: 3-September-2024.

[24] A. R. Protection and N. S. A. (ARPANSA), "X-rays." https://www.arpansa.gov.au/understanding-radiation/what-is-radiation/ionising-radiation/x-ray. [Accessed: 2 March 2022].

[25] "Mini-x2 x-ray tube." https://www.amptek.com/products/mini-x2-x-ray-tube. Accessed: 3-November-2023.

[26] Newport Corporation, "Hxp50-meca: High precision hexapod." https://www.newport.com/p/HXP50-MECA, 2024. Accessed: 3-September-2024.

[27] Queppelin, "What is six degree of freedom?." https://www.queppelin.com/what-is-six-degree-of-freedom/, 2024. Accessed: 2024-08-05.

[28] "Silicon drift detectors explained." https://engineering.temple.edu/sites/engineering/files/Oxford%20EDS-%20Silicon%20Drift%20Detectors%20Explained.pdf, 2012. Accessed: 3-November-2023.

[29] P. Serrano Galvez and Y. Gavrikov, "Instrumentation and Procedures to Test Bent Silicon Crystals with X-Rays at CERN," tech. rep., CERN, Geneva, 2020.

[30] Analog Devices, "Ad571: Complete 10-bit a/d converter." https://www.analog.com/media/en/technical-documentation/data-sheets/AD571.pdf, 2018. Accessed: 2024-07-08.

[31] "Displacement measuring interferometer." https://www.attocube.com/en/products/laser-displacement-sensor/displacement-measuring-interferometer. Accessed: 2023-08-01.

[32] M. R. J. Faure, "Study and redesign of a control solution for the piezo-goniometers," in *Indico: Mechatronics, Robotics, and Operations Section Presentations*, CERN, 2024. Presented at Universite de Technologie de Belfort Montbeliard (UTBM).

[33] S. Redaelli, "Crystal collimation of heavy-ion beams," in *15th International Particle Accelerator Conference (IPAC 2024)*, 2024.

[34] G. Ricci and et al., "Status of ml studies for loss pattern recognition.," in *178th Special Collimation Upgrade Specification Meeting*, (Geneva), CERN, July 26 2024.

[35] G. Brumfiel, "Lhc by the numbers," *Nature*, vol. 455, p. 172, September 2008.

[36] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock, *LHC Design Report*. No. Report No. 1 in CERN Yellow Reports: Monographs, Geneva: CERN, 2004.

[37] O. Brüning and L. Rossi, "The high-luminosity large hadron collider," *Nature Reviews Physics*, 2019.

[38] S. Redaelli, R. Bruce, A. Lechner, and A. Mereghetti, "Chapter 5: Collimation system," tech. rep., CERN, Geneva, 2020.

[39] S. Redaelli, "Beam Cleaning and Collimation Systems," tech. rep., Joint International Accelerator School, 2016. Comments: 35 pages, contribution to the 2014 Joint International Accelerator School: Beam Loss and Accelerator Protection, Newport Beach, CA, USA , 5-14 Nov 2014.

[40] N. Fuster-Martinez, R. Bruce, F. Cerutti, R. D. Maria, P. Hermes, A. Lechner, A. Mereghetti, J. Molson, S. Redaelli, E. Skordis, A. Abramov, and L. Nevay, "Simulations of heavy-ion halo collimation at the cern large hadron collider: Benchmark with measurements and cleaning performance evaluation," *Physical Review Accelerators and Beams*, vol. 23, nov 2020.

[41] N. Fuster Martinez *et al.*, "Run 2 collimation overview," in *9th LHC Operations Evian Workshop*, (Evian Les Bains, France), pp. 149–164, 2019.

[42] P. D. Hermes, *Heavy-Ion Collimation at the Large Hadron Collider: Simulations and Measurements*. PhD thesis, Munster U., 2016. Presented 19 Dec 2016.

[43] V. M. Biryukov, Y. A. Chesnokov, and V. I. Kotov, *Crystal Channeling and Its Application at High-Energy Accelerators*. Springer Berlin, Heidelberg, 1 ed., 3 2013.

[44] W. Scandale, G. Arduini, R. Assmann, C. Bracco, M. Butcher, F. Cerutti, M. D'Andrea, L. S. Esposito, M. Garattini, S. Gilardoni, E. Laface, L. Lari, R. Losito, A. Masi, E. Metral, D. Mirarchi, *et al.*, "Feasibility of crystal-assisted collimation in the cern accelerator complex," *International Journal of Modern Physics A*, vol. 37, 5 2022.

[45] J. Wagner, R. Bruce, H. Garcia Morales, W. Hofle, G. Kotzian, R. Kwee-Hinzmann, A. S. Langner, A. Mereghetti, E. Quaranta, S. Redaelli, A. Rossi, B. M. Salvachua Ferrando, R. Tomas Garcia, D. Valuch, G. Valentino, and G. Stancari, "Active halo control through narrow-band excitation with the adt," tech. rep., CERN, 2016.

[46] C. Suryanarayana and M. G. Norton, *X-Ray Diffraction*. Springer New York, NY, 1 ed., 6 1998.

[47] R. Camattari, V. Guidi, V. Bellucci, and A. Mazzolari, "The 'quasi-mosaic' effect in crystals and its applications in modern physics," *Journal of Applied Crystallography*, vol. 48, pp. 977–989, 8 2015.

[48] A. Mazzolari, A. Carnera, V. Guidi, D. D. Salvador, G. D. Mea, A. M. Taratin, D. Vincenzi, E. Bagli, P. F. Dalpiaz, and S. Baricordi, "Fabrication of silicon strip crystals for ua9 experiment," in *IPAC 2010: International Particle Accelerator Conference*, 2010.

[49] M. Panalytical, "X'pert^3 mrd - malvern panalytical." https://www.malvernpanalytical.com/en/products/product-range/xpert3-range/xpert3-mrd, 2024. Accessed: 2024-09-10.

[50] M. Butcher, A. Giustiniani, R. Losito, and A. Masi, "Controller design and verification for a rotational piezo-based actuator for accurate positioning applications in noisy environments," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, (Yokohama, Japan), pp. 003887–003892, IEEE, 2015.

[51] D. Quartullo, P. Arpaia, N. Biancacci, F. Giordano, I. L. Garcia, A. Mostacci, L. Palumbo, S. Redaelli, B. Salvant, L. Teofili, and M. Migliorati, "Electromagnetic characterization of the crystal primary collimators for the hl-lhc," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1010, p. 165465, 9 2021.

[52] M. Butcher, A. Giustiniani, R. Losito, and A. Masi, "Controller design and verification for a rotational piezo-based actuator for accurate positioning applications in noisy environments," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015.

[53] G. Hernandez, *Fabry-Perot interferometers*. Cambridge University Press, 1988.

[54] N. Instruments, "Pxi systems - national instruments." https://www.ni.com/en/shop/pxi.html, 2024. Accessed: 2024-09-10.

[55] A. Masi, O. O. Andreassen, M. Arruat, M. D. Castro, S. Danzeca, M. Donze, S. Fargier, M. Gulin, E. V. D. Bij, I. Kozsar, J. Palluel, E. Matheson, P. Peronnard, J. Serrano, E. Soria, J. Tagg, F. Vaga, and G. Switzerland, "Sambuca: Sensors acquisition and motion control framework at cern," in *19th Int. Conf. Accel. Large Exp. Phys. Control Syst.*, JACoW Publishing, Geneva, Switzerland, 2023.

[56] M. Rachel Julie Faure and et al., *Development of a micro-rad precise angular stage control of a piezo-actuator*. PhD thesis, Université de technologie de Belfort Montbéliard (UTBM), Sevenans, France, 8 2024. Presented 2024.

[57] B. Dehning, E. Effinger, J. Emery, G. Ferioli, G. Guaglio, E. B. Holzer, D. Kramer, L. Ponce, V. Prieto, M. Stockner, and C. Zamantzas, "The

lhc beam loss measurement system," in *2007 IEEE Particle Accelerator Conference (PAC)*, pp. 4192–4194, IEEE, 2007.

[58] B. Dehning, "Beam loss monitors at lhc," 11 2014.

[59] M. D'Andrea, G. Azzopardi, M. D. Castro, E. Matheson, D. Mirarchi, S. Redaelli, G. Ricci, and G. Valentino, "Prospects to apply machine learning to optimize the operation of the crystal collimation system at the lhc," in *IPAC 2022: the 13th International Particle Accelerator Conference*, pp. 1362–1365, JACOW Publishing, Geneva, Switzerland, 7 2022.

[60] A. J. C. Wilson and V. Geist, *International Tables for Crystallography. Volume C: Mathematical, Physical and Chemical Tables*. Dordrecht/-Boston/London: Kluwer Academic Publishers, 1992. Published for the International Union of Crystallography.

[61] Amptek Inc., "FAST SDD® Ultra High Performance Silicon Drift Detector." https://www.amptek.com/products/x-ray-detectors/. Accessed: 2024-06-25.

[62] Amptek Inc., "Digital Pulse Processor PX5." https://www.amptek.com/products/digital-pulse-processors/px5-digital-pulse-processor. Accessed: 2024-06-25.

[63] "Introduction to abstract classes - cppreference.com." https://en.cppreference.com/book/intro/abstract_classes, 2024. Accessed: 2024-09-12.

[64] Google, "Googletest - google testing and mocking framework." https://github.com/google/googletest, 2024. Accessed: 2024-09-12.

[65] "std::shared_ptr - cppreference.com." https://en.cppreference.com/w/cpp/memory/shared_ptr, 2024. Accessed: 2024-09-12.

[66] MetaYeti, "mini - minimalistic ini parser for c++." https://github.com/metayeti/mINI, 2024. Accessed: 2024-09-12.

[67] "Asio c++ library." https://think-async.com/Asio/. Accessed: 2024-09-12.

[68] "Boost sml (state machine library)." https://boost-ext.github.io/sml/index.html, 2024. Accessed: 2024-09-12.

[69] "Boost c++ libraries." https://www.boost.org/, 2024. Accessed: 2024-09-12.

[70] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[71] M. H. Weik, *full-width at half-maximum*, pp. 661–661. Boston, MA: Springer US, 2001.

[72] Xilinx, Inc., "Zcu102 evaluation kit." https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html, 2024. Accessed: 3-September-2024.

[73] Analog Devices, Inc., "Ad5781: 18-bit, 1 µs, unbuffered voltage output dac." https://www.analog.com/en/products/ad5781.html#part-details, 2024. [Online; accessed 3-September-2024].

[74] I. Motorola, "Spi (serial peripheral interface) bus," 1980. Available in various microcontroller datasheets and reference manuals.

[75] AMD, "Vivado design suite." https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado.html, 2024. Accessed: 2023-09-10.

[76] AMD, "XPM CDC SINGLE: Single-bit Clock Domain Crossing." https://docs.amd.com/r/en-US/ug953-vivado-7series-libraries/XPM_CDC_SINGLE, 2024. Accessed: 2024-07-09.

[77] AMD, "Vitis unified software platform." https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vitis.html, 2024. Accessed: 2023-09-10.

[78] K. Ogata, *Modern Control Engineering*. Upper Saddle River, NJ: Prentice Hall, 5th ed., 2010.

[79] MathWorks, "Simulink." https://it.mathworks.com/products/simulink.html, 2024. Accessed: 2024-09-04.

[80] MathWorks, "Embedded coder." https://www.mathworks.com/products/embedded-coder.html, 2024. Accessed: 2024-09-04.

[81] The Qt Company, "Qt Framework." https://www.qt.io, 2024. Accessed: 2024-07-10.

[82] Qt for Python Development Team, "PySide6 Documentation." https://doc.qt.io/qtforpython, 2024. Accessed: 2024-07-10.

[83] H. Belyadi and A. Haghighat, "Chapter 5 - supervised learning," in *Machine Learning Guide for Oil and Gas Using Python* (H. Belyadi and A. Haghighat, eds.), pp. 169–295, Gulf Professional Publishing, 2021.

[84] G. Ricci and et al., "Real time crystal collimation monitoring at the cern large hadron collider," in *Proc. IPAC'24*, no. 15 in IPAC'24 - 15th International Particle Accelerator Conference, pp. 1750–1753, JACoW Publishing, Geneva, Switzerland, 05 2024.

[85] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, 4 2021.

[86] C. Obermair, T. Cartier-Michaud, A. Apollonio, W. Millar, L. Felsberger, L. Fischl, H. S. Bovbjerg, D. Wollmann, W. Wuensch, N. Catalan-Lasheras, M. Boronat, F. Pernkopf, and G. Burt, "Explainable machine learning for breakdown prediction in high gradient rf cavities," *Physical Review Accelerators and Beams*, vol. 25, Oct. 2022.

[87] C. Tennant, A. Carpenter, T. Powers, A. Shabalina Solopova, L. Vidyaratne, and K. Iftekharuddin, "Superconducting radio-frequency cavity fault classification using machine learning at jefferson laboratory," *Physical Review Accelerators and Beams*, vol. 23, Nov. 2020.

[88] G. Azzopardi and G. Ricci, "New Machine Learning Model Application for the Automatic LHC Collimator Beam-Based Alignment," in *ICALEPCS2021*, pp. 953–958, 2022.

[89] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[90] F. Chollet, "Github - keras-team/keras: Deep learning for humans." https://github.com/keras-team/keras. Accessed: March 5, 2021.

[91] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker,

V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, (USA), p. 265–283, USENIX Association, 2016.

[92] Google, "Normalization | machine learning | google developers." https://developers.google.com/machine-learning/data-prep/transform/normalization?hl=en. Accessed: 20 March 2021.

[93] L. Prechelt, *Early Stopping — But When?*, pp. 53–67. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[94] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, p. 281–305, feb 2012.

[95] D. M. et al., "Operational handling of crystal collimation at the lhc," in *14th International Particle Accelerator Conference (IPAC 2023)*, 2023.

[96] K. Skordis and et al., "FLUKA coupling to Sixtrack," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, pp. 17–25, 2019.

[97] P. Hermes and et al., "Simulation of heavy-ion beam losses with the sixtrack-fluka active coupling," in *Proc. IPAC'16*, pp. 2490–2493, 2016.

[98] A. Mereghetti and et al., "Sixtrack-fluka active coupling for the upgrade of the sps scrapers," in *Proc. IPAC'13*, pp. 2657–2659, 2013.

[99] R. Cai and et al., "Simulation framework and measurements of crystal collimation of proton beams at the large hadron collider," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1060, p. 169038, 2023.

[100] C. Ahdida, D. Bozzato, D. Calzolari, F. Cerutti, N. Charitonidis, A. Cimmino, A. Coronetti, G. L. D'Alessandro, and et al., "New capabilities of the fluka multi-purpose code," *Frontiers in Physics*, vol. 9, 2022.

[101] G. Battistoni, T. Boehlen, F. Cerutti, P. W. Chin, L. S. Esposito, A. Fassò, A. Ferrari, and et al., "Annals of nuclear energy," *Frontiers in Physics*, vol. 82, pp. 10–18, 2015.

[102] R. D. Maria and et al., "SixTrack status," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, pp. 11–15, 2019.

[103] R. D. Maria and et al., "Sixtrack version 5: Status and new developments," in *Proc. IPAC'19*, pp. 3200–3203, 2019.

[104] R. Bruce and et al., "Simulations and measurements of beam loss patterns at the cern large hadron collider," *Phys. Rev. Spec. Top. Accel. Beams*, vol. 17, p. 081004, 2014.

[105] D. Mirarchi, O. Andreassen, R. Bruce, R. Cai, M. Di Castro, M. D'Andrea, M. Hostettler, D. Jacquet, A. Masi, E. Matheson, S. Redaelli, G. Ricci, M. Solfaroli, S. Solis, J. Tagg, and J. Wenninger, "Operational handling of Crystal collimation at the LHC," in *IPAC23*, vol. 2023, p. MOPL022, 2023.

[106] M. D'Andrea, *Applications of Crystal Collimation to the CERN Large Hadron Collider (LHC) and its High Luminosity Upgrade Project (HL-LHC)*. PhD thesis, Padua U., 2021. Presented 23 Feb 2021.

[107] A. A. Garcia, B. Dehning, G. Ferioli, and E. Gschwendtner, "Lhc beam loss monitors," in *5th European Workshop on Diagnostics and Beam Instrumentation*, (ESRF, Grenoble, France), pp. 198–200, 2001.

[108] F. Zimmermann and M. G. Minty, *Measurement and Control of Charged Particle Beams*. Springer, Berlin, Heidelberg, 8 2015.