Universidad
Carlos III de Madrid

Departamento de Bioingeniería e Ingeniería Aereospacial

TRABAJO FIN DE GRADO

DEVELOPMENT OF THE ACQUISITION SOFTWARE FOR A CONE BEAM IN-VITRO MICRO-CT

Autor: Rafael Moreta Martínez

Tutor: Asier Marcos Vidal

Leganés, Julio de 2015

Título: Development of the acquisition software for a cone beam in-vitro micro-CT
Autor: Rafael Moreta Martínez
Director: Asier Marcos Vidal

EL TRIBUNAL

Presidente: Sara Guerrero Aspizua

Vocal: Carlos León Cansenco

Secretario: Dhiraj Kumar

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 9 de Julio de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Abstract

During the 70's Computed Tomography (CT) opened the door to the possibility of obtaining anatomical information from living subjects non-invasively. Since then, this technology has become indispensable for clinical diagnostic in medicine as well as for sanitary research. Consequently, micro-CTs rose from the necessity in the preclinical environment of the advantages that CT offers to the clinicians, providing high-resolution images of small samples.

The Universidad Carlos III de Madrid has designed a new high-resolution in-vitro x-ray micro-CT that will serve as a test bench for wide number of applications in research and teaching. The goal of this project is to implement a control and data acquisition software for this device. LabVIEW has been used as the development environment due to the advantages that offers in the creation of graphical user interfaces, its commodious configuration to communicate with many types of hardware elements and its flexibility to expand the software in future works with few modifications.

This project comprises the development of a set of libraries to control the hardware elements: x-ray source, flat-panel detector and mechanical system of the UC3M test bench. In addition, the implementation of a "step and shoot" acquisition protocol is needed, which combines control libraries previously developed. The architecture of the implemented software leads to the possibility of expanding its functionalities into more advanced features such as advanced acquisition protocols or imaging techniques.

The work included in this project is framed in one of the lines of research carried out at the Biomedical Imaging and Instrumentation Group at the Departamento de Bioingeniería e Ingenería Aereospacial of the University Carlos III de Madrid.

Keywords: x-ray, in-vitro micro-CT, LabVIEW, acquisition protocol

Index

Image Index

Table Index

Chapter 1

Motivation and Objectives

Since the middle of the 20th century, medical imaging has become one of the most important tools in medicine, not only because it has helped for a better understanding of human anatomy, but also it has largely improved medical diagnostic. The new advances in technology have helped to introduce revolutionary changes to this field. Computed Tomography (CT) techniques have contributed to these improvements beyond from what could have ever been imagined, by obtaining images from the inside of the body with a non-invasive technique [1]. These systems are now indispensable tools for both, clinical diagnostic and in laboratories for research. Therefore, new devices have been developed, as micro-CTs, which provide high-resolution images from small animals or samples in laboratories.

Therefore, if the purpose is to improve the field of medical imaging it is comprehensible the necessity of this new devices, where this field still needs future contributions. For this reason, laboratories and research groups are interested in the purchase of these devices in order to make progress in the field, as it has many applications. Some of the new lines of research try to reduce doses for the patients, reduce costs and improve diagnoses, among others.

The Universidad Carlos III de Madrid (UC3M) has designed a new high-resolution in-vitro x-ray micro-CT that will serve as a test bench for wide number of applications in research and teaching. The new device has been built and assembled without a specific software. The goal of this project is to implement it to control every component of the UC3M test bench in order to be used in wide range of applications.

An in-vitro micro-CT as the UC3M test bench has many different applications, being teaching one of them. It could be used to instruct the students in x-ray radiography and CT techniques by looking directly into a real device and to have the possibility to handle it from the hardware to the end-user software layer. In addition, thanks to the easy setup of the device and the simple software configuration, this machine could be used by anyone interested in applications, not only the field of biomedical engineering, but also in others like industrial or aerospace engineering. Right now, the UC3M test bench could be a complement in any practical use, as it could be lab practices, teaching materials, projects, thesis, etc.

UC3M test bench has been designed with a modular architecture with the necessity of having an advanced simple software that provide to the researcher enough freedom to build more advanced functions without any kind system design restrictions. In this device, advanced acquisition protocol geometries would be easy to implement, such as helical acquisition. Furthermore, new imaging techniques that give different information in images than normal radiography can be tested. Examples of uses that could be implemented in to this in-vitro micro-CT are phase and dark field contrasts, dual energy protocols, among others. Moreover, due to its modular architecture, its components could be exchanged or replaced in order to test new elements, such x-ray sources or detectors.

In order to fulfill these necessities, a control and acquisition software is needed in the test bench, being this the main goal of this project. LabVIEW programming language has been chosen as the programming platform due to its simplicity of making user interfaces and its facilities for the communication with external devices. This fits with the need of a flexible software in order to give to the researcher as much facilities as possible to develop more advanced functions.

The fact of using LabVIEW as a programming tool for controlling a micro-CT is not new. It has been demonstrated before that LabVIEW is perfectly capable of managing complex devices, as shown in previous works [2]. LabVIEW is a programming tool that has infinite features that will help to this test bench to be capable of being used in tremendous applications [3]. It also has the characteristic of being much more simple than other programming languages, giving the opportunity to a not programmer expert to implement incredible advance functions.

It would not be surprising to highlight the importance of this project. From now on, the machine is available for many different applications, as the one described above. The software implemented will give a wide number of opportunities to make the UC3M test bench a determining tool for research and teaching.

1.1 Key milestones

Having in mind these previous considerations, the specific objectives of this project are the following:

- To develop a set of libraries to control the device hardware elements independently using LabVIEW programming language.
- To implement a "step and shoot" acquisition protocol using the base implemented control libraries.
- To design and implement an end-user control and acquisition console for acquiring data.
- Test and assessment of the software.

The work included in this project is framed on one of the lines of research carried out at the Biomedical Imaging and Instrumentation Group at the Departamento de Bioingeniería e Ingenería Aereospacial of the Universidad Carlos III de Madrid.

1.2 Outline of the document

The document is organized in six chapters. The first section defines the main motivation that has led to the development of this project, in addition of the key milestones of the project. The following section presents the state-of-the-art of the CT technique, including its main characteristics and its evolution along time, giving importance to the uses of in in-vitro and in-vivo systems.

Chapter 3 presents in details the principal components of the hardware and software used in the project.

In chapter 4 an overview of the implemented software in the device is given. Including how was manage every part individually and the final software implementation combining every part of the device.

The system evaluation of the final software is given in chapter 5, including results and examples of functioning.

The last chapter of the document includes a short analysis of the obtained results, the future work lines and an approximation of the budget of the project.

Chapter 2

Principles of CT imaging

2.1 Introduction

Extracted from the Merriam-Webster's dictionary, the word tomography comes from the Greek word "tomos" – section - standing for "a method of producing a three-dimensional image of the internal structures of a solid object (as the human body) by the observation and recording of the differences in the effects on the passage of waves of energy impinging on those structures" [4]. This technique is used in medical imaging to obtain images of the body by different methods, as for example, x-ray imaging, magnetic resonance, nuclear imaging, etc.

X-ray radiation was discovered in 1895 by Wilhelm Röntgen. Röntgen realized the importance of its properties and found that could be used to obtain images of the inside of the body using a non-invasive approach [5]. Since then, new x-ray techniques as Computed Tomography have been developed in medicine. CT has helped to diagnose many serious illness and complex post-traumatic complications by the generation of high-resolution 2D and 3D images in of the interior of patients.

At first, radiography was the only known technique to obtain images from the interior of the body of the patient without violating the outer surface of it. Radiography consists in irradiating a patient by an x-ray source and then measure that attenuated radiation in a special photographic film. The three-dimensional volume of the human body is projected along the direction of the x-rays into a planar (2D) image. The final 2D image shows the superposition of the different structures of the body, which it induces some limitations that results in significant reduction of the visibility of the elements of interest. These limitations led to the development of new techniques that would give much more information than just a single radiography. Figure 1 compares a conventional radiography of a human thorax (a) with an image of an axial section of a human thorax from a CT study (b). Notice that in image (b) structures are not overlapped, giving more precise information of the interior of the body than (a).

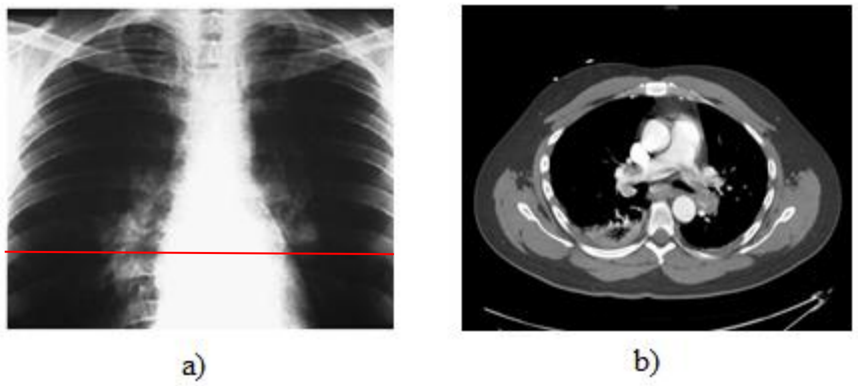


Figure 1 – a) Convenrtional radiography of a human thorax. b) Axial section of a human thorax from a CT study.

A. E. M. Bocage made the first approximation to a CT scanner in 1921. Bocage described an apparatus consisted of an x-ray source, an x-ray film, and a mechanical system. The motion system made synchronous linear movements in opposite directions of the x-ray source and the detector that allowed focusing on a specific plane while the uninterested planes kept blurry in the image. However, the quality of this technique varies depending on the thickness of the structure of interest [6].

Tomography from projections was attempted as early in 1940. Gabriel Frank described the basic idea of reconstructing the inside of a body taking in consideration different projections at different angles around the body. This idea characterized the today's reconstruction algorithms [7]. However, the application of this method was very limited due to the lack of computational tools in that period.

In the decade of 1960s, the first CT scanner was built simultaneously by two scientists, Allan M. Cormark and Godfrey N. Hounsfield. They built their own CT scanners independently. Cormark wanted to estimate the attenuation coefficient of tissues to improve the radiation treatment [8]. In the case of Godfrey, he built another device for tomographic image acquisition that became the first CT scanner for clinical application. It was known as EMI scanner and it was finished in 1971. The work made by Cromark and Godfrey was awarded with the Nobel Prize in 1979, shared by both of them [9].

In order to understand how a CT scanner works, the fundamentals of x-rays will be presented in the following sections.

2.2 Fundamentals of x-rays

X-ray radiation is an electromagnetic wave form, as are microwaves, infrared, visible light, ultraviolet or radio waves. The wavelength of the x-ray ranges from a few picometers to a few nanometers. X-rays are a form of ionizing radiation, which means that are energetic enough to interact with atoms, having the potential to liberate electrons from the atoms that bind them. When an electron is stripped off from an atom or a molecule, an ion pair is formed. This feature distinguishes these rays from the rest of the electromagnetic spectrum. *Figure 2 shows* the whole electromagnetic spectrum.

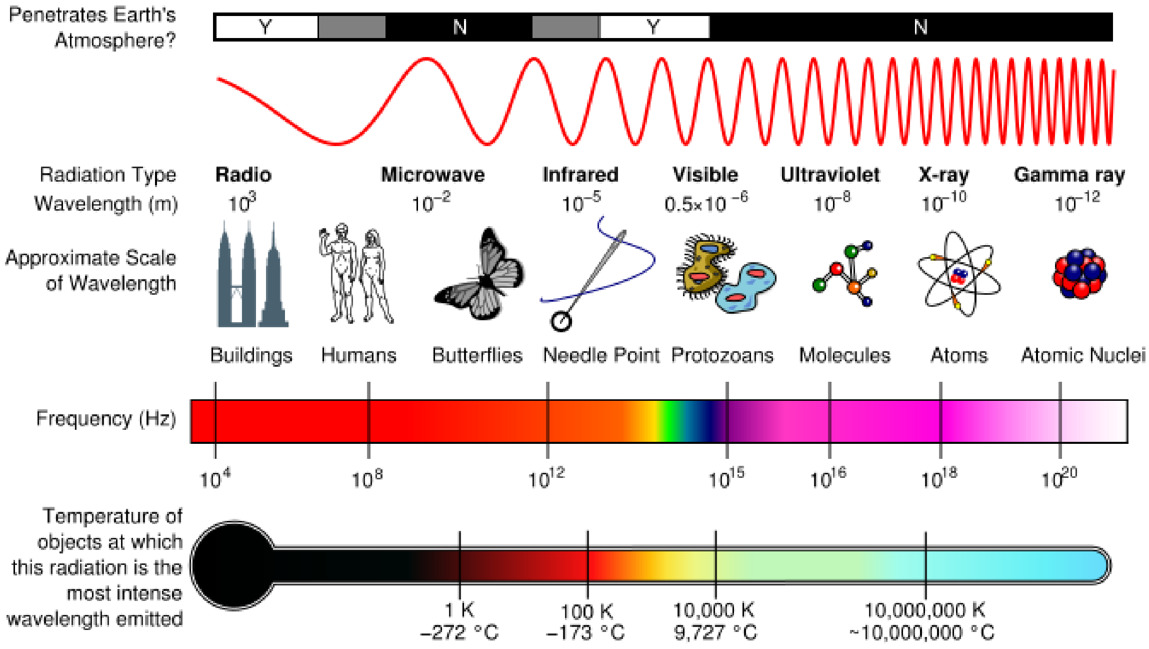


Figure 2 – Electromagnetic radiation spectrum

The energy E of an electromagnetic wave is proportional to its frequency v, and is defined by the following expression:

$$E = h * v = \frac{h * c}{\lambda}$$

Where h is the Planck's constant which equals to $6.63 * 10^{-34} J \cdot s$, and c is the speed of light and λ de wavelength of the x-rays [10].

For medical diagnosis in radiology, the wavelength of x-rays varies from 0.1 nm to 0.01 nm, corresponding to energy values from 12.4 keV to 124 keV. X-rays with much shorter wavelength are highly penetrating, providing very low contrast information.

2.2.1 X-ray production

The mechanism to generate x-rays consists in bombarding a high atom density element (usually tungsten and molybdenum) with high-speed electrons. These high-speed electrons collide with the target, leading to different interactions between the material and the particles that, due to several physical properties, will produce x-ray radiation.

X-ray photon production occurs in an x-ray tube. The x-ray source is composed by a cathode and an anode. In the cathode (negative pole) an accelerated electrons beam is produced through a high potential difference. The anode (positive pole) is the metallic substance that is going to be bombarded by the electrons. When the electrons impact the target, part of the energy is transformed in x-ray photons and the rest into heat (96-99%). *Figure 3* shows a basic scheme of an x-tube.

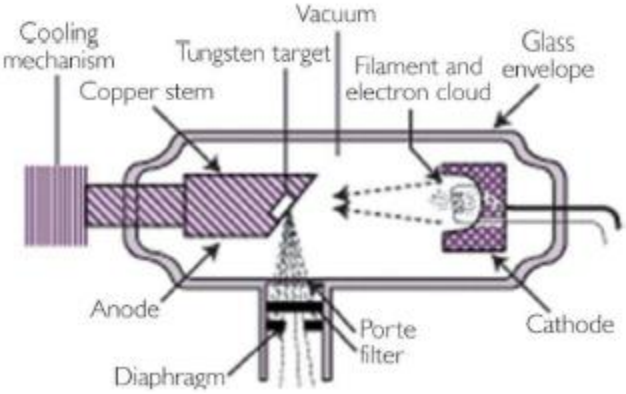


Figure 3 - Scheme of the components of an x-ray tube.

The interactions of the electrons with the high-density atoms of the target lead to the generations of x-rays in two forms:

1. **Bremsstrahlung radiation**: This radiation occurs when an electron approaches close to the nucleus of an atom of the target and is deviated from its path, or in the limit case, collide to the nucleus. In both cases, the electron decelerates, implying the emission of a photon. The energy of the created photon is proportional to the loss of kinetic energy, which depends on the distance between the nucleus of the atom and the path of the electron. *Figure 4* (a) and (c) represents this phenomenon.

2. **Characteristic radiation:** Produced when a high-speed electron collides with one of the inner shell electrons of the target atom, ejecting it from the layer (*Figure 4 (b)*). When an electron from the outer shell fills the hole, characteristic radiation is emitted. Therefore, the energy of the characteristic radiation is equal to the difference between the energies of both electrons' orbital.

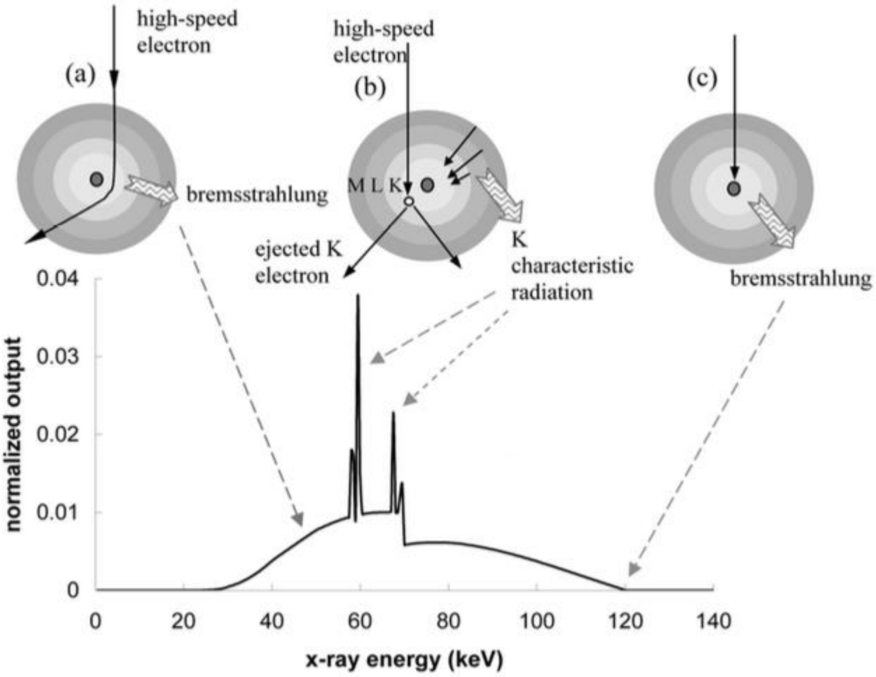


Figure 4 - Illustration of electron interaction with a target and its relationship to the x-ray tube energy spectrum. (a) Bremsstrahlung radiation. (b) Characteristic radiation. (c) Bremsstrahlung radiation when an electron hits the nucleus directly of an atom. The graph at the bottom represents the radiation spectrum of the x-ray source [11].

In *Figure 4*, besides the three diagrams of how both radiation energies are produced, there is a graph that represents the *energy spectrum* of an x-ray source. It shows the distribution of the energies of the photons that are produced during the interactions. The peaks on the graph represent the characteristic radiation, which correspond to the energy differences between the electrons' orbitals of the atoms. Their localization in the spectrum is unique for the composition of the anode. The Bremsstrahlung radiation represents the continuous part of the spectrum, which is distributed along the whole graph. The amplitude of the spectrum will be different depending on the energy used in the cathode [11].

2.2.2 Interactions of x-rays with matter

The basic principle of Computed Tomography is based on the difference between the amount photons emitted by the source and detected after passing through the sample. These photons interact with the matter, which means that the x-ray beam is attenuated after going through the body. Thanks to these phenomena, it is possible to acquire tomographic images. The different types of interactions that occur between x-rays and atoms are the followings:

1. **Photoelectric Effect:** the incident x-ray interacts with an electron in the medium by giving all of its energy and ejecting the electron from the atom. The energy of the incident x-ray photon is greater that the binding energy of the electron, therefore it is possible to liberate the electron from the shell of the atom. The vacancy created is almost immediately filled up by another electron from a higher energy level and at this moment, a secondary photon results from the energy difference. The energy of the new photon belongs to the characteristic radiation part of the spectrum. However, these secondary photons are neglected because of their low interaction with matter (Figure 5).

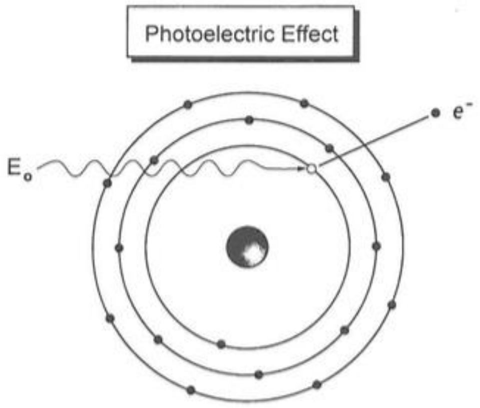


Figure 5 – Photoelectric effect [10].

2. **Rayleigh scattering:** the incident x-ray photon interacts with the electric field of an atom’s electron suffering a deviation of its path without changing its initial energy (Figure 6).

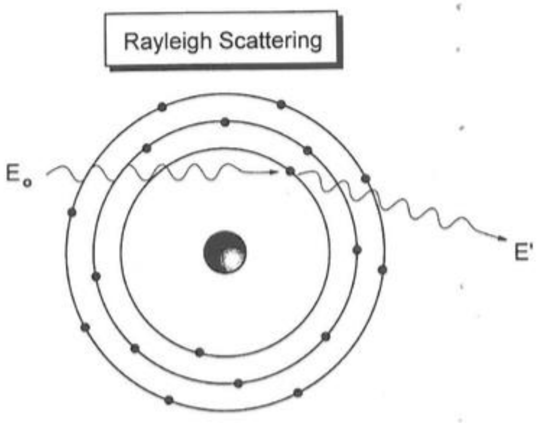


Figure 6 – Rayleigh scattering [10].

3. **Compton scattering:** In this interaction, the energy of the incident x-ray photon is considerably higher than the binding energy of the electron of the atom. An incident x-ray photon strikes an electron and frees it from the atom. The incident x-ray photon is deflected or scattered with partial loss of its initial energy. This interaction produces a positive ion, a recoil electron, and a scatter photon. The photon is deflected with an angle from 0º to 180º (*Figure 7).*

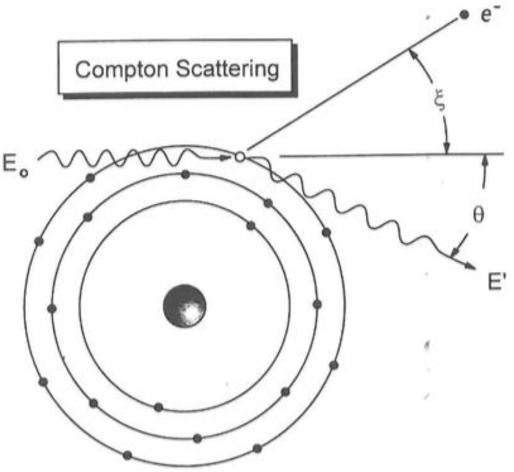


Figure 7 – Compton scattering [10].

Attenuation is defined as the reduction of the intensity of an x-ray photon when it passes though the sample due to absorption or scattering interactions that take place between the x-ray beam and the atoms. The attenuation varies depending on the element that interacts with the x-ray beam. Therefore, every material has a specific attenuation coefficient [11]. In the *Figure 8,* different attenuation coefficients are presented for different types of tissues present in the human body.

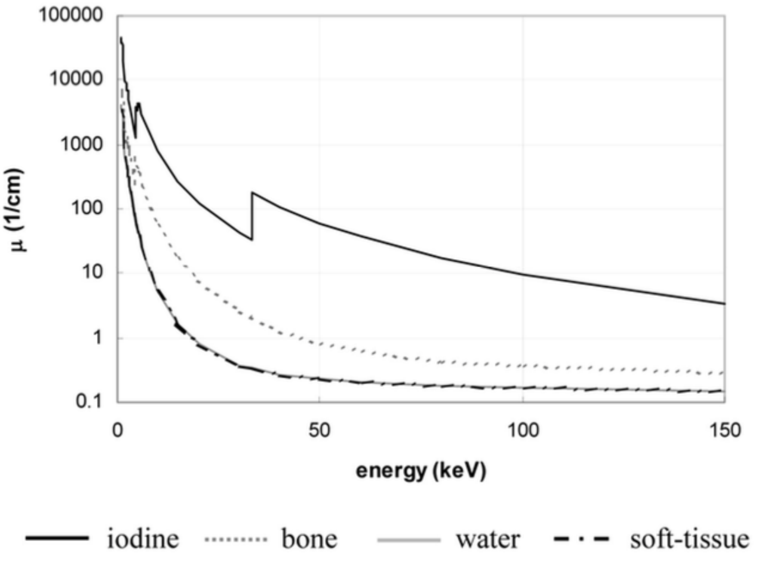


Figure 8 – Linear attenuation coefficients for different materials [11].

2.2.3 X-ray detection

The first x-ray detectors used in medicine were photographic films covered with silver crystals. However, around the 80s digital detectors appeared, and they were immediately installed in clinical equipment as they were much cheaper to maintain and offered better image quality. The technology was known as Computer Radiography (*CR*). The detector process of generating a digital image took place in two stages. First, the x-ray photons reach the first screen exciting the electrons of a phosphor crystal layer, storing a temporal form of energy. In the second stage, the crystal layer is excited by a laser leading to a form of energy converted into visible light. After that, this light is captured with an array of photodiodes that generates an analogue electrical signal for a further storage as a digital signal [12]. This process is represented on *Figure 9*:

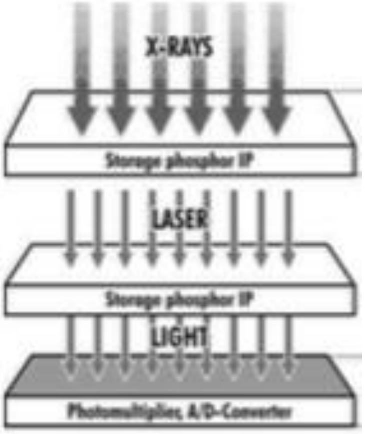


Figure 9 – Scheme of a CR digital detector.

Later on, around the 90s another evolution came up on detection systems with the existence of the first Direct Radiography (*DR*) systems, where x-rays were converted into electrical signals avoiding the necessity of having a read stage as in CR systems [10].

There are two types of DR systems:

1. **Direct detectors:** made of a solid material (a photoconductor) placed between two electrodes. In the absence of x-rays, the photoconductor acts as an insulator, but when x-rays strike the photodetector bounds of electrons are promoted to the conduction band and become conductive electrons. These charges then migrate to the upper and lower electrodes, and the accumulated charge is measured electronically. This will lead to generate an image. Figure 10 *(a)* represents a direct DR detector.

2. **Indirect detectors:** These systems are based on scintillators. X-rays interact with the phosphor, causing it to emit light in the visible range (Figure 10 *(b)*). After that, the photodetector records the pattern of visible light giving the final image.

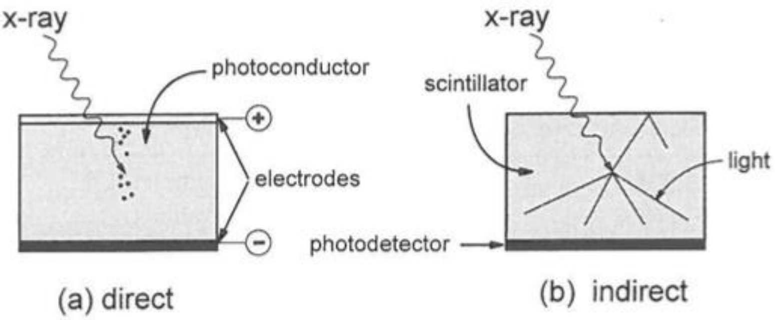


Figure 10 – (a) Direct DR detector. (b) Indirect DR detector [10].

Flat-panel detectors are indirect detectors of reduced size, which offer better image quality and efficiency in low contrast than all every other digital detector. These kinds of detectors are one of the most common used nowadays for in-vitro and in vivo medical imaging [13].

2.3 Introduction to CT Scanners

X-ray *Computer Tomography scanner* (CT scanner) is the structural medical imaging modality that provides a volumetric representation of the attenuation coefficients shown by the different body tissues to the incoming x-ray radiation [14].

This technique provides a 3D image through a mathematical algorithm which solution is the volumetric distribution of the attenuation coefficients along the volume of the sample. The implementation of the reconstruction method is done using different images that are obtained of the patient from multiple angles, which are referred to projections.

The most important elements that a CT scanner has are an x-ray source and a detector positioned in front of it. Then these elements will rotate around the sample of interest and different projections will be acquired.

Since this technique was initiated, it has experimented a considerable development at both software and hardware level. This has led to the evolution of different generations of CT, improving their initial design and performance.

The first generation of CT is represented in *Figure 11*. This design uses a pair of x-ray source-detector. The emitted beam is narrow and cylindrical and covers a really small area. Then, it combines translational and rotational movements in order to obtain a complete set of projections for obtaining a complete final image. This approach has the drawback of taking excessive time to make a complete acquisition. Long times would imply a high degradation process in the image quality due to the possible movements of the patient.

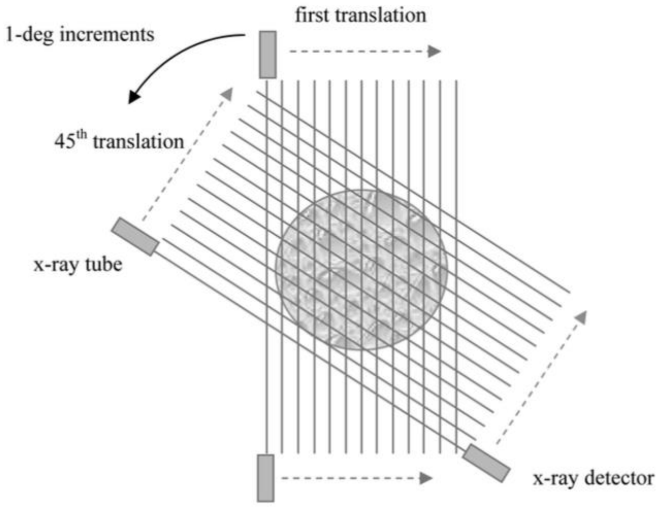


Figure 11 – First generation CT [11].

These limitations led to the development of the second generation of scanners, represented in *Figure 12*. However, the geometry is similar to the first generation, the main difference is that although this is still a translational-rotation scanner, the number of rotation steps is reduced by the use of a set of multiple detectors for and sources. This way helps to reduce the acquisition time considerably.

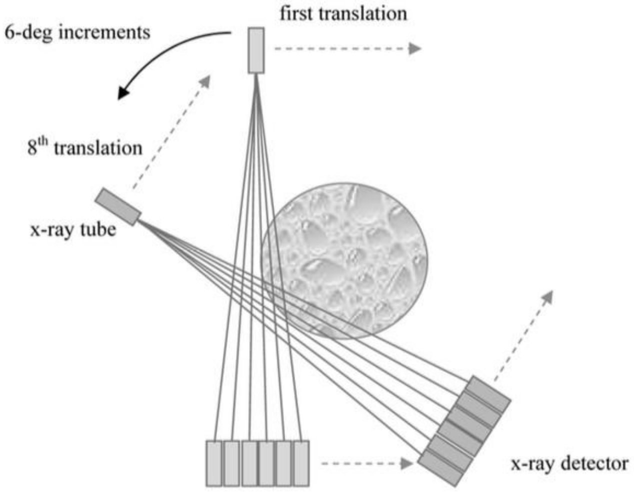


Figure 12 – Second generation CT [11].

However, it was not until the next generation when a significant change was experimented, as it can be shown in *Figure 13*. In this configuration many detectors cells are located on an arc concentric to the x-ray source. The size of each detector is sufficiently large so that the entire object is within each detector *field of view* (FOV) at all times. The x-ray source and the detector remain stationary to each other while the entire apparatus rotates about the sample. Translational motion is eliminated to reduce the data acquisition time. The focal point of the x-ray source opens to 60º allowing it to illuminate the whole detector. This beam shape is known as *fan-beam*.

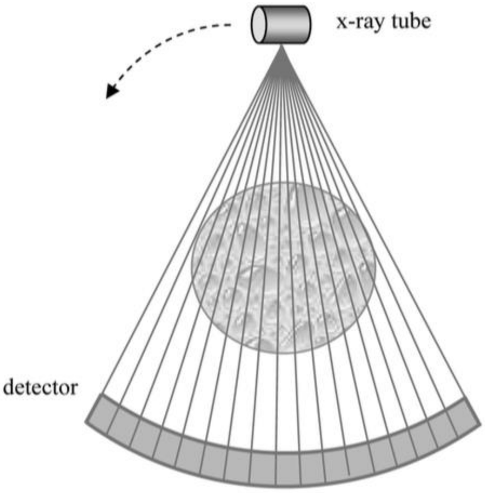


Figure 13 – Third generation CT [11].

Fourth and fifth generations are similar to those of third generation in terms of acquiring and reconstructing the image (*figure15*). The differences are that the detectors cells are stationary all around the sample and only the x-ray rotates, avoiding the necessity of sending the acquired information through complicated slip-rings [11].

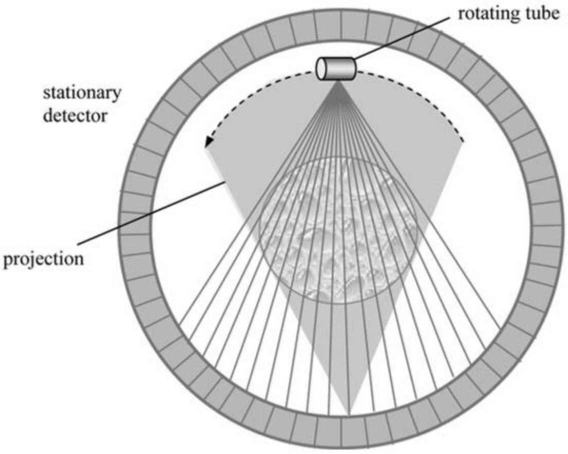


Figure 14 – Fourth and fifth generation [11].

Recently, with the use of bi-dimensional *flat-panel* detectors, the acquisition obtains data of a whole volume and not just a single slice of the sample in a single rotation. The x-ray source opens divergently with a cone-beam shape of radiation. Cone-beam CT (CBCT) scanners are nowadays largely used, thanks to the new applications that can fit such as orthodontics, image-guided surgery or small animal imaging [15].

2.4 Micro-CT: in-vivo and in-vitro

There are two distinguishable types of CTs that from this project scope deserve to be mentioned. These are *in-vivo* and *in-vitro* CT devices.

In-vivo CTs systems are those used for living samples. Therefore, these systems have a special architecture where the x-ray source and the detector are within the same mechanical structure rotating around the sample, being this still. This approach avoids unwanted movements of the sample alive.

On the other hand, in-vitro CTs are systems used for ex-vivo samples. Due to this fact, the architecture changes with respect to the in-vivo systems. This time the x-ray source and the detector remain stationary while the sample platform rotates in order to get the projections. This architecture provides higher mechanical stability. In addition, it is possible to acquire images from objects of sizes and shapes that with the in-vivo CT would not be possible. However, these kinds of devices require a higher acquisition time, which implies higher doses to the sample of study. Furthermore, due to the architecture of in-vitro CTs, these systems cover wider number of applications, not only in the biomedical field, but also in industrial and aerospace engineering, for example. This kind of CT will give different uses for different materials or structural elements, as the size and the magnification can be easily changed.

Additionally, there is also a classification of CTs not only based its architecture but also based on the sample size. For this project, the work is focused in micro-CTs.

Micro-CT is a CT technique developed to study small samples in the sub-millimeter range. These devices have significant differences in the structure and the elements in comparison to a human CT. Firstly, micro-CTs usually use fan-beam microfocus x-ray sources with high resolution flat-panel detectors. Secondly, due to the small size and mass of the sample, much higher image resolution is needed in order to get the equivalent level detail than in the clinical environment, allowing the differentiation of small structures. This implies smaller pixel sizes in the detector and smaller focal spot in the source. Therefore, to keep the *signal to noise ratio* (SNR) in a reasonable value, higher doses are needed; hence, a balance between resolution, noise, maximum x-ray source power, acquisition time and doses received must be achieved.

Chapter 3

Materials: Hardware and Software

The Universidad Carlos III de Madrid has purchased an in-vitro micro-CT to be used as a test bench for research and as a teaching platform (*Figure 15* shows the UC3M test bench). The main task in this project is to design an acquisition *software* for this device, including from the hardware control libraries to the end-user acquisition console as well as well as performing the assessment and testing of the system. For a better understanding of its development, a specific description of the *hardware* and *software* that has been used will be presented in the following sections.

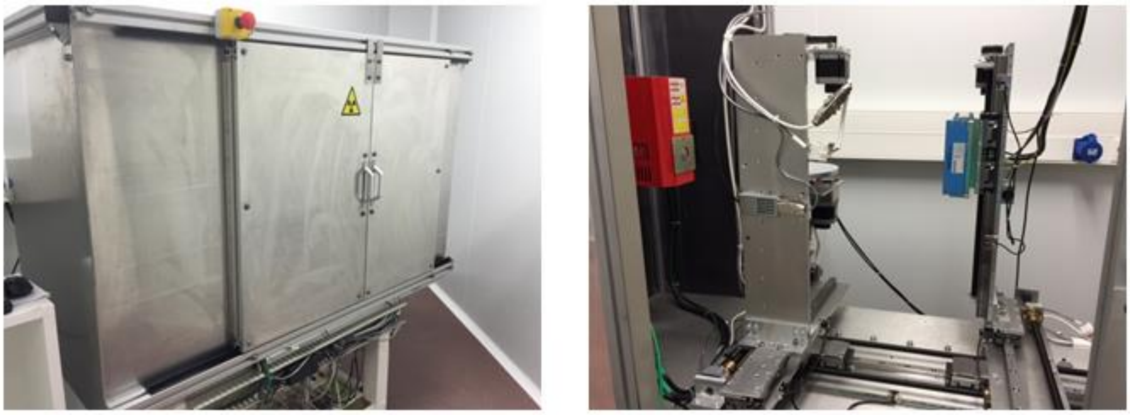


Figure 15 – UC3M test bench.

3.1 UC3M test bench overview

The UC3M test bench is an in-vitro cone-beam micro-CT located at a special lead shielded room in the Bioengineering laboratories of the Leganes Campus at the Universidad Carlos III de Madrid. This area has a special permission from the Consejo de Seguridad Nuclear (CSN) to accommodate radiation sources in safety conditions. The shielded room is shown in *Figure 16*. The device was designed at the Universidad Carlos III de Madrid an assembled by SMI (Sedecal Molecular Imaging, Madrid, Spain)

Figure 16 – Lead shielded room in the Bioengineering laboratories of Leganes campus.

The UC3M test bench setup has the following hardware configuration (Figure 17 shows an image of the components of the UC3M test bench):

- X-ray source: fixed in one side of the bench.
- Digital flat panel detector: placed face to face with the x-ray source, it is installed in a three directions motion stage that allows it to perform displacements along the axis x-y-z.
- Sample tray: A circular sample plate is placed in between the x-ray source and the detector. It has attached a rotating motion stage to perform angular movements during the acquisition. Apart from the circular motion, it is mounted over a three axis mechanical system to perform movements along the x-y-z directions.
- Chassis: All the described hardware elements are placed in an aluminum structure.

- Working station: Computer that connects with all the elements of the test bench, controlling the hardware and executing the acquisition protocols.

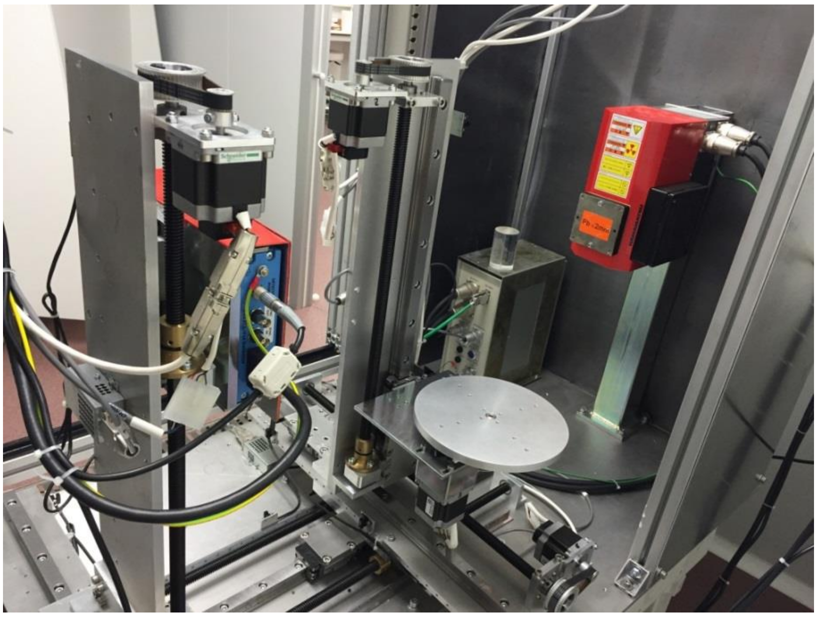


Figure 17 – UC3M test bench.

The scheme on *Figure 18* shows a sketched diagram of the basic elements of the UC3M test bench:

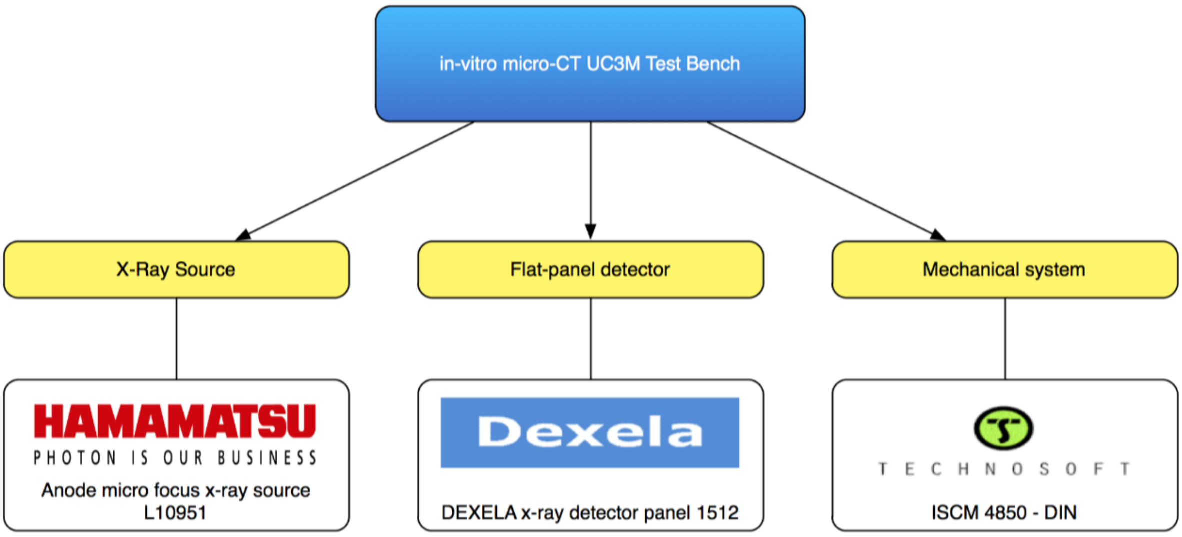


Figure 18 – Individual elements of the UC3M test bench.

In the following sections, the components of the test bench will be exposed in detail, giving detailed characteristics of them.

3.2 Hardware

3.2.1 X-ray Source

The system makes use of a tungsten anode *micro focus x-ray source (L10951, Hamamatsu* Photonics K.K., Japan). The unit is composed of two elements, the x-ray tube and a control unit (*Figure 19*).

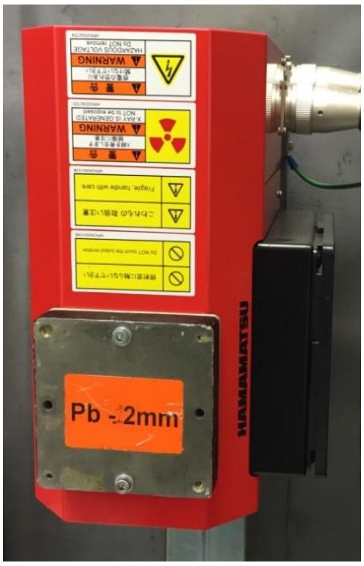


Figure 19 – X-ray source (L10951, Hamamatsu Photonics K.K., Japan).

The source delivers a maximum power of 50 W, having an energy range of 40 kV and 110 kVp and an anode current in the range of 10 and 800 μA. The focal spot size varies between 15 and 80 μm, depending on the power delivered, with a maximum cone aperture angle of 62°.

The control unit of the system is separated from the x-ray source and has a high-voltage power unit attached. In this system, the safety is carried by two interlock security inputs, which avoid the emission of x-rays in case any of any shield protection systems are opened. The control unit works as a hardware controller. It is connected to the computer through a serial connection RS-232 and sends and receives commands in order to know different specifications or control the emission of the source. These functions are explained in detail in section 4.2.

The complete specifications of the x-ray source are exposed in Annex A.

3.2.2 Detector

The x-ray *flat-panel* detector is the *Dexela 1512 model* (Perkin Elmer, Inc., USA) (*Figure 20*). It is an indirect detector based on a CsI:Tl scintillator screen over a CMOS image sensor of 1944x1536 pixels matrix to detect the visible light coming from the scintillator crystal. The pixel size is of 75 µm. The photodiode matrix has different binning capabilities up to 4x4 pixel area. The frame rate allows reading up to 86 frames per second at binning 4x4. The data transfer between the PC and the detector is carried by a Pixci EB1 (EPIX, Inc. USA) and a camera link connection.

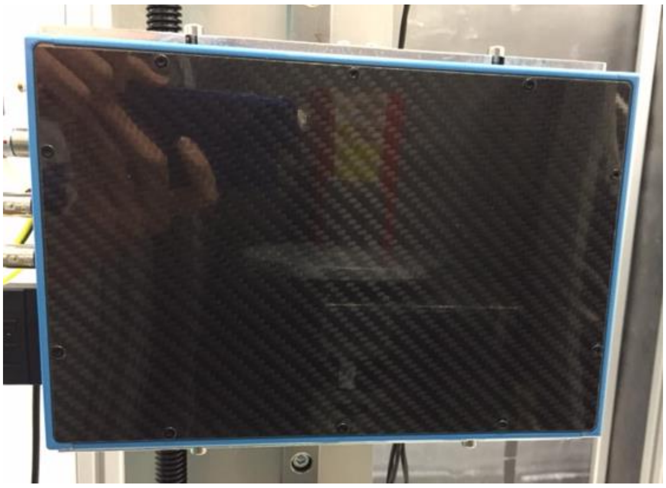


Figure 20 - X-ray flat-panel detector (Dexela 1512, Perkin Elmer, Inc., USA)

The detailed data-sheet is available in Annex B.

3.2.3 Mechanical System

The mechanical system is composed by seven motion stages controlled by an ISCM 4805 servo control unit (Technosoft SA, Switzerland). Six of the units perform linear motions and one is a rotational stage (*Figure 21*). The connection with the working station is through a serial port RS-232 connection.

The detector and the sample plate structures can be moved in the x-y-z axis. The sample plate has an extra stage to perform rotation, with the purpose of placing the sample at different angular positions. The length of the displacement of the elements is limited in order to avoid collisions.

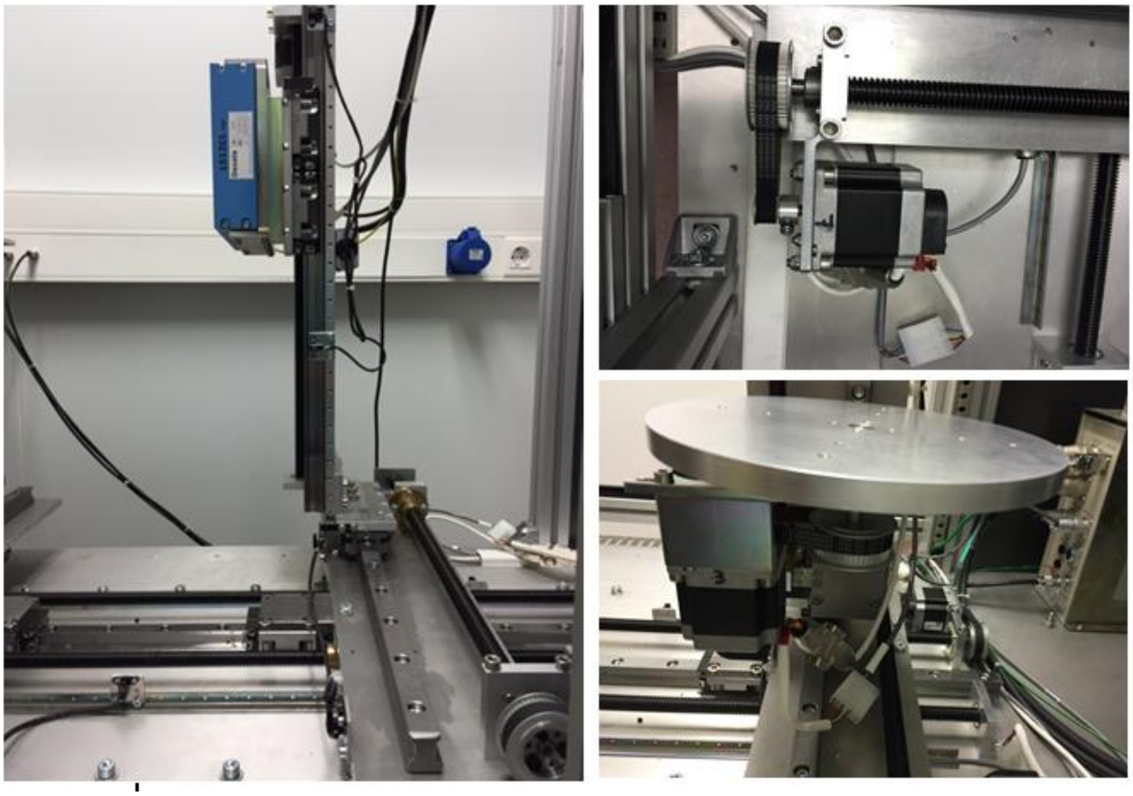


Figure 21 – Different motion stages of the UC3M test bench.

Except for the x-ray source, every other part can be moved along the three axes. This feature gives flexibility to the system allowing modifying the acquisition geometry arbitrarily in terms of magnification, resolution and trajectories of the sample. In addition, a wide range of samples sizes can be scanned as they can be placed easily in the device without any structural limitation.

3.2.4 Working Station

The system is managed by a PC that runs the programs that control every part of the system through serial connections, PCI and Ethernet. It uses Windows 7 x64 as operative system, with a processor INTEL® Core™ i5-3330 CPU 3.00 GHz – 3.2 GHz, with RAM 4 GB and LabVIEW 2012 x64 as the development platform.

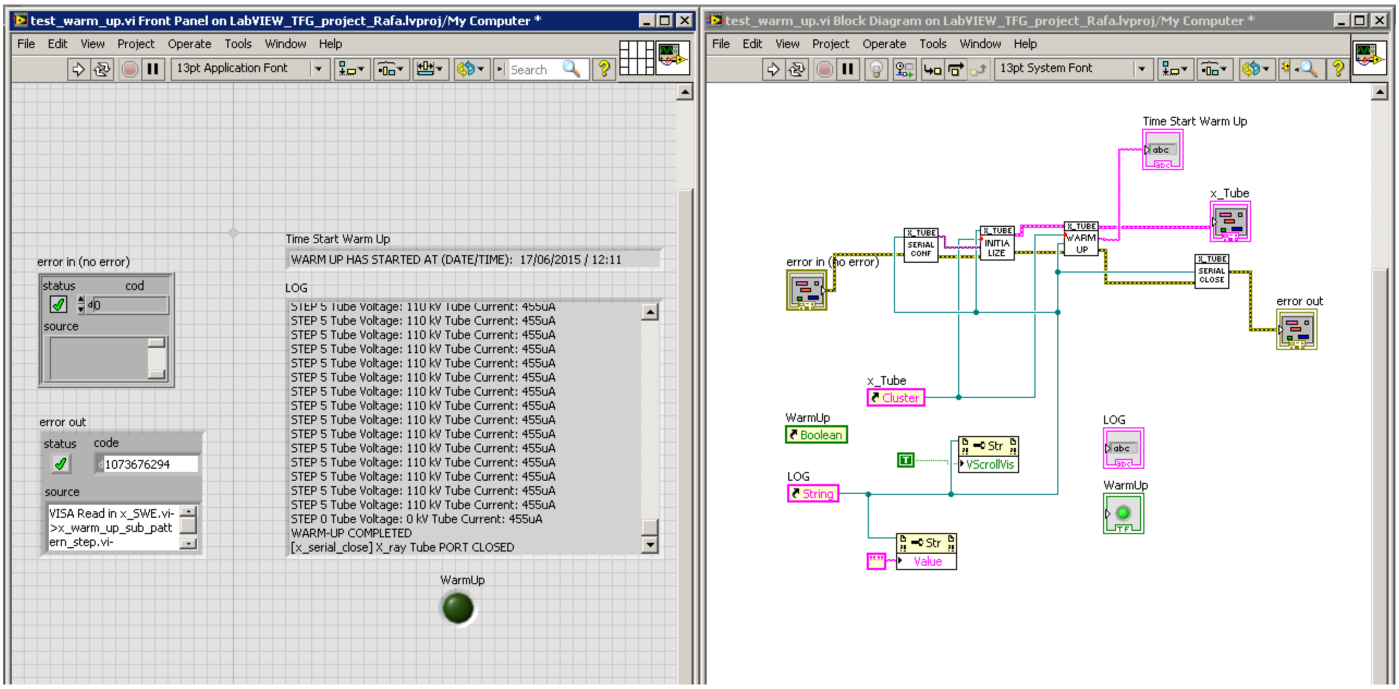
3.3 Software

3.3.1 LabVIEW Language

The acquisition *software* has been developed with LabVIEW (National Instruments, USA) programming language. This dataflow graphical high-level programming language uses block diagrams instead of introducing sentences through code lines. Some of the advantages that LabVIEW offers are the simplicity of creating graphic user interfaces and its commodious configuration to connect with many types of hardware, great features that satisfy perfectly to the requisites for this project.

LabVIEW has a special code structure denominated *virtual instruments* (VI), which are equivalent to traditional programming functions. These VIs interconnect between each other in order to create complex structures and. In addition, it allows the integration of code from other programming languages as are C/C++, using calls to *dynamic-link libraries* (commonly known as DLL).

VIs have two different components. One is the *front-panel*, where all the graphical elements are placed such as buttons, switches and indicator panels, among others. The other is the *block-diagram*, where all the functions and instructions of the program remain. The graphical user interface is created before writing code. In Figure 22 an example of a LabVIEW code structure is shown.



Time Start Warm Up: WARM UP HAS STARTED AT (DATE/TIME): 17/06/2015 / 12:11

error in (no error): status ✓ cod 0

error out: status ✓ code 1073676294, source: VISA Read in x_SWE.vi->x_warm_up_sub_pattern_step.vi-

LOG

```
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 5 Tube Voltage: 110 kV Tube Current: 455uA
STEP 0 Tube Voltage: 0 kV Tube Current: 455uA
WARM-UP COMPLETED
[x_serial_close] X_ray Tube PORT CLOSED
```

Figure 22 – Example of a virtual instrument (VI) of LabVIEW code structure

If the task is to develop advanced complex programs, LabVIEW has the possibility of creating projects that allows building a large number of VIs and organize them in libraries. One of the most important advantages that will contribute to this project is that it is specially designed to program data acquisition protocols. The execution of code in parallel synchronously is very simple and direct.

To conclude, LabVIEW is a modular environment, where it is easy to exchange big sections of code with others in a quick way, contributing to this its high level of abstraction in the hardware layer.

Every single fact commented previously make LabVIEW a magnificent solution for this project, since it permits a single implementation and to be executable in different devices with just small changes. Moreover, it allows implementing new code in the programs and adding new functionalities in a clean, fast and comfortable way.

3.3.2 UC3M Libraries

In order to design the *software* to control the flat-panel detector and read the images, there have been used libraries in C++ code developed at the Universidad Carlos III de Madrid. These libraries provide functions to manage the detector using the manufacturer low-level libraries. They are used under the LabVIEW software layer, as the read-out performance of low-level languages is much better. These libraries will be integrated into the LabVIEW execution workflow.

3.3.3 3D reconstruction software

In order to validate the developed software, the data acquired by the CT must be reconstructed. For this, Mangoose (UC3M, Madrid, Spain) [16] reconstruction software is available . This software implements the *FDK* (Feldkamp David and Kress) [17] approximation for cone beam sources of the analytical *filtered back projection* (FBP) algorithm.

3.4 Connection diagram

Figure 23 represents and scheme of all the connections between the computer and all the elements of the UC3M test bench.

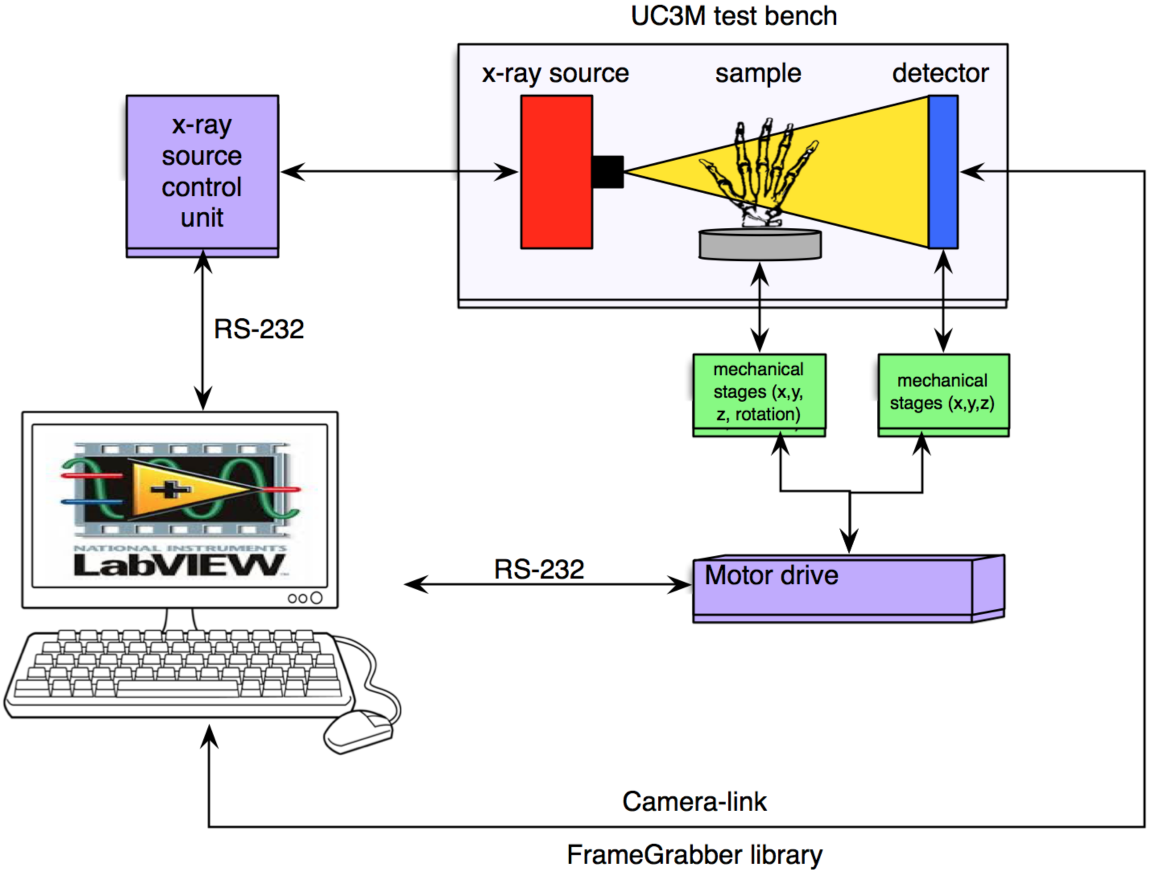


Figure 23 – Scheme of the connections between the working station and the elements of the UC3M test bench.

Chapter 4

Software Architecture

This chapter presents the architecture of the developed *software* that was implemented to make operable the in-vitro micro-CT UC3M test bench. It will first start with a section explaining the basic characteristics of how the software has been designed. In the following parts, each individual module of the device will be explained in detail, including the individual connections with the PC working system and the software created to control them individually.

The last section details the software library that interconnects all the different modules in order to create a graphical user interface to control the micro-CT and perform acquisitions using a *step and shoot* protocol.

4.1 Software characteristics

4.1.1 Application Requirement

The software has been designed in order to fulfill the specifications needed for making the UC3M test bench system operable. This would mean to be able to control every part of the micro-CT and to combine all the libraries in order to make more advanced functions, with the final goal of increasing the number of applications of the device. Consequently, the software has been created with the specification of being flexible and having a modular structure.

The flexibility that the software offers to the user is quite open. It has been developed in a manner that when creating a graphic user interface for acquisition protocols can act as the intermediary between the user and the main system. Therefore, this is a feature that allows adding, modifying or deleting new acquisition protocols or functionalities without having to change the base of the program.

The modular structure gives to the user the possibility of modifying the process of any of the different modules without changing the whole code. This is possible due to how the different modules are based on the different libraries with specific characteristics of each of the modules.

4.1.2 Structure Software LabVIEW

The acquisition software has been structured in a LabVIEW project named "uc3m_testbench_Rafa". Every file is organized in different folders based on the specific library that they belong to. There are four principal directories, which are subdivided into directories that are more specific. The main directories are x-ray tube, detector, motors and ACQ. Every directory contains each library of each module of the micro-CT, except for the ACQ directory that stores the acquisition protocols, besides the graphical user interface of *step and shoot*.

In Figure 24, a diagram of the structure of the project is illustrated:

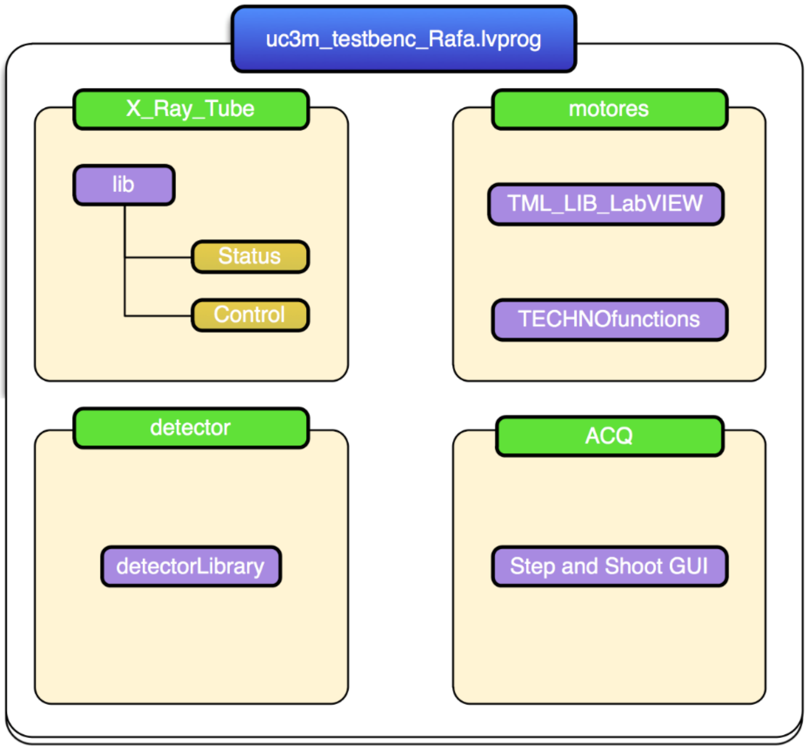


Figure 24 - Project structure diagram.

4.2 X-ray source module

The x-ray source control unit controls the x-ray tube, which is connected to the PC working station via a serial connection RS-232. The communication is carried out by sending commands in ASCII code from the PC to the x-ray tube control unit followed by the response of the control unit to the PC. The manufacturer provides a document with all the commands that can be sent to the tube.

To have a full control of the x-ray tube a wide number of functions have been implemented and packed into a library. In the following subsections, the most important functions will be explained in detail. First, it will be specified how the communication works, as well as the meaning of some of the commands. Then, it will be described how it was implemented to process the response of the tube to all the possible commands. To conclude, in the last subsection the most advanced functions designed will be presented in an understandable way.

4.2.1 Communication process and "x_tube"

COMUNICATION PROCEDURE

Communication with the x-ray source is done by sending specific commands to the control unit through the serial port. When the control unit receives the command, it processes it and sends a response back to the computer with the requested information. The sending and receiving process is detailed below:

Commands:
The commands are sent as ASCII format messages ended with a carriage return character (\r).

Example: "STS\r", "HIV\s30\r".

Responses:
Responses received from the x-ray source are also finished with a carrier return character (\r). There are two types of responses that the tube can send to the user:

- **Acknowledge:**
  The control unit sends the same command that receives, sometimes followed by the requested data. It indicates that the command has been received successfully.

  Example: "STS\s2\r", "HIV\r".
- **Error:**
  When an error occurs, the tube sends the command "ERR" followed by a specific error identifier code. This number helps to identify what kind of error has occurred. Table 1 explains the different codes that can be received.

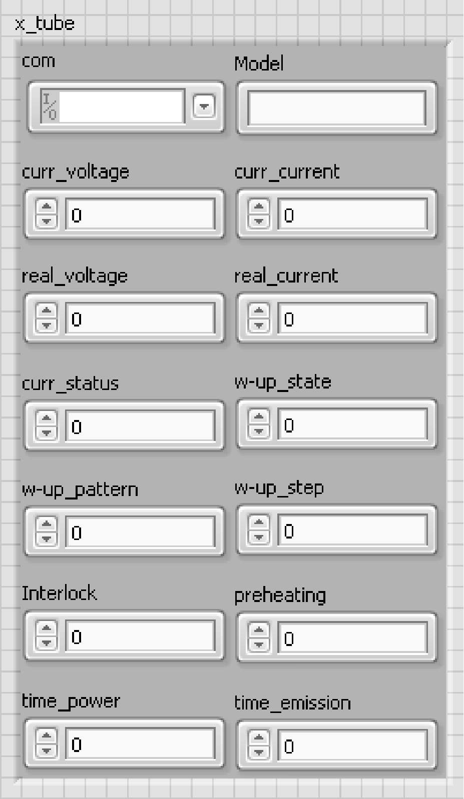
  Example: "ERR\s10\sXON\r".

Code	Description
ERR 0	Unknown command
ERR 10	System not ready to run command
ERR 20	Parameter out of range
ERR 40	Power out of range

Table 1 - Error codes description from the tube.

X_TUBE

In order to control at any moment the status of the x-ray source there has been created the data cluster "x_tube", a structure in LabVIEW. The cluster contains the most relevant status parameters of the tube, which are useful for the user in order to know the status of the tube while the user is using the micro-CT. The values of these parameters will be updated dynamically during the execution of the program. This structure provides a safer and more controlled use of the device. Figure 25 illustrates the "x_tube" cluster:



x_tube

com

Model

curr_voltage 0

curr_current 0

real_voltage 0

real_current 0

curr_status 0

w-up_state 0

w-up_pattern 0

w-up_step 0

Interlock 0

preheating 0

time_power 0

time_emission 0

Figure 25 - x_tube cluster from the front-panel screen shot.

In Table 2 every one of these parameter are briefly explained:

Parameter	Command	Data Type	Description
Com		Instrument	Serial port
Model	TYP	String	X-ray tube model
Cur_voltage	SPV	Int	Selected voltage
Cur_current	SPC	Int	Selected Current
Real_voltage	SHV	Int	Actual real voltage value
Real_current	SCU	Int	Actual real current value
Curr_status	STS	Int	Tube state
W-up_state	SWE	Int	If warm up is needed
W-up_patter	SWS	Int	Warm up pattern
W-up_step	SWE	Int	Warm up step
Interlock	SIN	Int	Interlock
preheating	SPH	Int	Preheating status
Time_power	STM	Int	Turn on time
Time_emission	SXT	Int	Emission time

Table 2 - x_tube parameters description.

At the beginning of the execution of any application that makes use of the tube, the cluster “x_tube” is initialized with the current values of all these parameters by running a specific VI named “x_initialize_XTUBE”.

From all these parameters there are some that should be mentioned and explained more extensively:

- **Curr_status:**
  It informs of the current status of the x-ray tube by sending the command “STS\r”. The Table 3 explains the possible different values of the response:

Status (STS)	Description
0	Waiting for warm-up
1	Warm-up in progress
2	Ready to emit x-rays
3	X-rays are being emitted
4	Overload protection is activated
5	X-rays cannot be emitted

Table 3 - curr_status status description.

- **Real_volt and real_curr:**
  It reports the value of the voltage and the current that the x-ray tube is emitting in real time. Voltage values are between 10 and 110 kV and the current value are between and 0 and 800 µA.

- **Cur_voltage and cur_current:**
  It describes user preset values of voltage and the current. It means that if the tube is turned on then it will emit at those values of current and voltage.

- **W-up_state:**
  The x-ray tube needs a warm-up process before the emission of x-rays if long times have passed without using it. This parameter allows the user to know if the warm up process is needed or not. The duration of the warm-up protocol depends on how much time has passed since the last emission of x-rays. Table 4 shows the possible values:

Warm up state (SWE)	Description
0	Warm-up completed
1	Warm-up in progress
2	Waiting for warm-up

Table 4 - W-up_state status description.

4.2.2 Control and state library of the x-ray tube

In the previous subsection it was explained the manner of communication of the x-ray source with the user. Therefore, there has been created a library with VIs for each command that process specifically the response of the tube, which updates the cluster and gives information to the user.

This control library is composed of different Vis organized in two types: control and state. The control VIs send commands that perform an action, changing the status of the source, and the status VIs get information about the state of the x-ray tube. All these Vis have the same function prototype as it is shown in Figure 26 and include the following parameters:

- **Inputs:**
  - **x_tube**: Structure with the parameters of the tube.
  - **Error in**: cluster of error.
- **Output:**
  - **x_tube**: updated structure with the parameters of the tube.
  - **Output X_Tube**: response command of the tube.
  - **Message Response**: processed answer.
  - **Error out**: cluster of error.

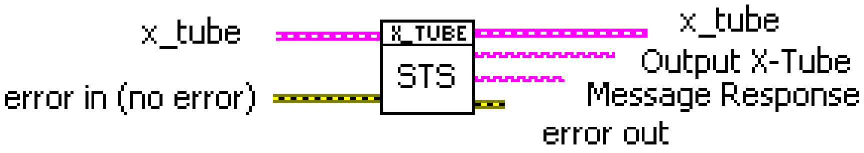


Figure 26 - VI prototype from the tube control library.

The workflow of these VIs follow three steps: send the command, read the response and process it in order to update the cluster and give a message to the user if necessary.

Response processing:
The response processing has two possible steps depending on the type of response:

- **Error:**
  If the command received is an error code, then it will be processed and the VI will return a description of the error. This section is performed by subVI called "x_ERR", which checks which error has occurred and searches for the associated message.

- **No error:**

If there is no error, the requested information sent by the x-ray tube in the response is extracted and the cluster “x_tube” is updated. After that, a specific message to the user is generated.

There are about 20 different functions that have been implemented with this structure, each one processing a different command of the tube.

In Figure 27 the block panel from one of these functions is shown:

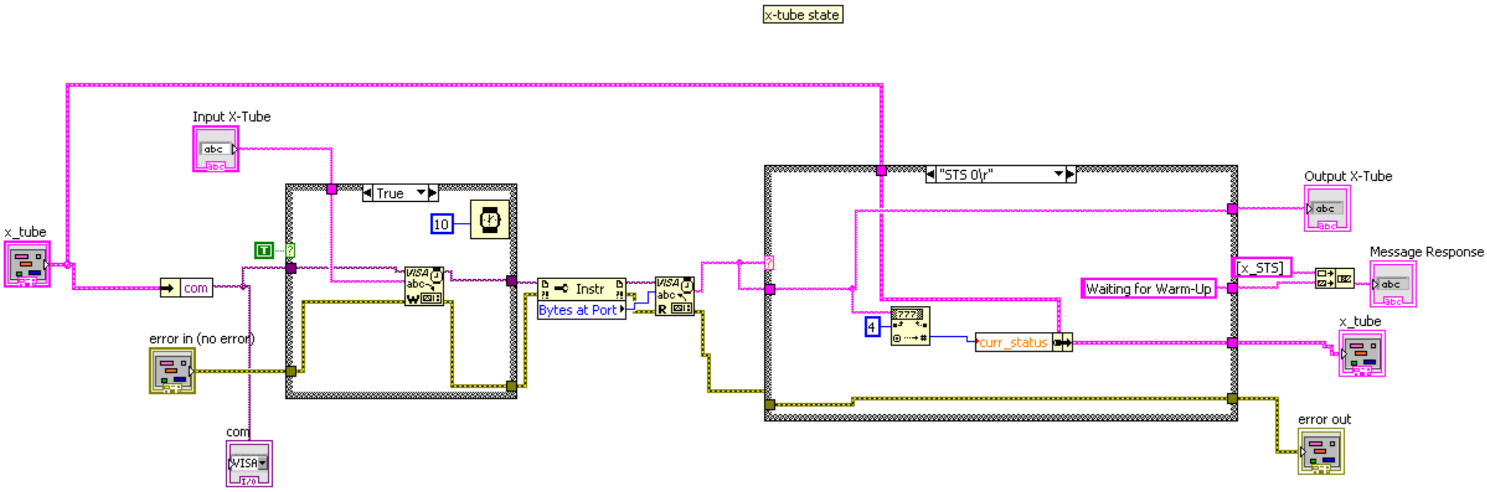


Figure 27 - VI block-panel example from the tube control library.

Figure 28 represents a flow chart of the process of this library functions.

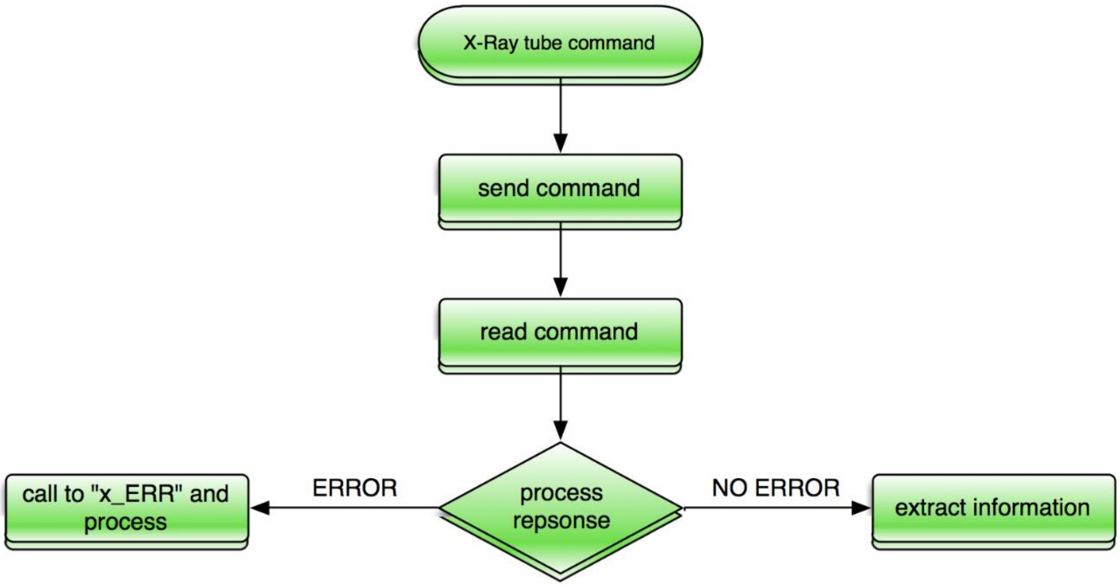


Figure 28 - Flow chart of the tube control library process.

4.2.3 Advanced Functions

In chapter 3 there has been discussed the major library to control the x-ray tube. This library is used as a base to build more advanced functions. For this project, there are three advanced function that have been created. These functions are a warm-up protocol, “x_XON_Ramp” and “x_setVC” and they are explained in the following subsections.

4.2.3.1 Warm-up protocol

It was said before that the x-ray tube has a warm-up process that must be done before the emission of x-rays. The warm up process is a way of making sure that the x-ray tube can reach to the desired voltage and current of the x-ray beam. Therefore, there has been created a user program that manages the warm up process. This program will tell the user in real-time the state of the warm-up process. It controls the warm-up procedure, checking up that in every step is performed properly. If something goes wrong and the process stops the user will know what it has occurred.

The warm up function process has the following steps:

Step 1 – Checking state:
First, it is verified if the warm up process is needed by sending the command “STS”. If so, the program will continue, if not the program will finish. If there is any other problem then it will return to the user the specific problem that has occurred.

Step 2 – Starting warm up:
The control command to launch the warm up process is sent. If there is an error when sending the command the programs stops. If not, it waits until the warm up process starts by sending another specific command.

Step 3 – Pattern check:
In this step, the program verifies the warm up pattern that is being followed. There are three different patterns depending on how much time has passed since the last time that the x-ray tube has emitted. If it has passed long times (one month, for example) the warm up process will be longer. Therefore, the response is processed and the program will inform the user about how long it will take to finish the warm-up process.

Step 4 – Check step:
In the last step, it is checked whether the warm-up process has stopped for any reason or still running. If it has not stopped, then the program will display to the user in which step is the process, and the values of voltage and current. When the process is completed the program will finish.

In Figure 29 the flow diagram is illustrated of the warm up function:

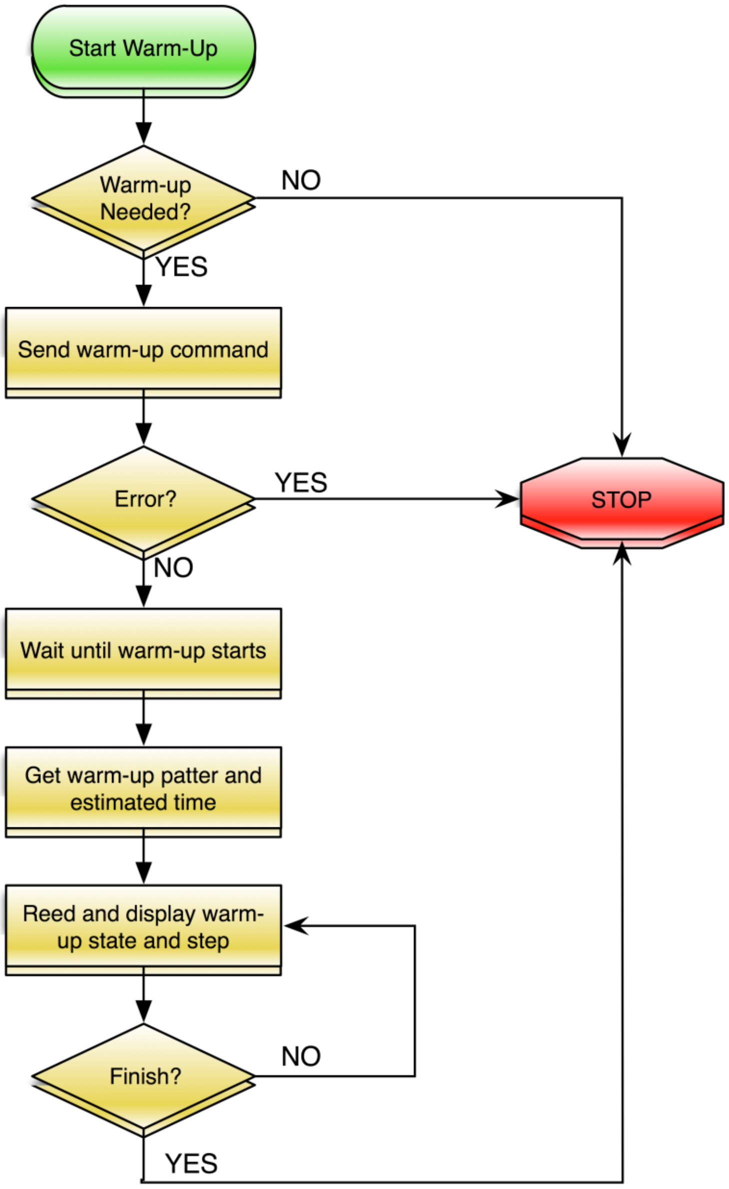

Start Warm-Up
Warm-up Needed?
NO
YES
Send warm-up command
Error?
YES
STOP
NO
Wait until warm-up starts
Get warm-up patter and estimated time
Reed and display warm-up state and step
Finish?
NO
YES

Figure 29 - Flow chart of the warm-up protocol.

4.2.3.2 x_XON_Ramp

The advanced function "x_XON_Ramp" has been developed due to the following reason. The x-tube cannot emit the desired voltage and current immediately right after the tube is turned on. It takes a while until it reaches to the desire levels of voltage and current. Therefore, this function informs the user when the x-ray tube is emitting at the preset voltage and current values.

The workflow of this function is divided in several steps:

Step 1 – Turning x-rays on:

It checks the preset values of voltage and current and the X-ray tube is turned on.

Step 2 – Current voltage and current check up:
The function enters into a while loop that is executed every second and a variable that counters the number of iterations is set to zero.

Step 2.1:
In every iteration of the while loop, a status command is sent to the tube in order to check if x-rays are being emitted or something has gone wrong.

- If the source is not emitting, the program stops and informs to the user that the emission has stopped.

- If the tube is emitting x-rays then the function checks if the x-rays voltage and current values have reached to the desired value.
    - When the desired value is reached, the counter variable starts counting iterations. When there has passed three iterations (three seconds) without changing, then the program stops and informs the user that the x-ray emission is ready. However, if the voltage or the current changes before reaching the variable to value three, this counter is restarted to 0 and the process keeps running.

Figure 30 illustrates the flow diagram of the function:

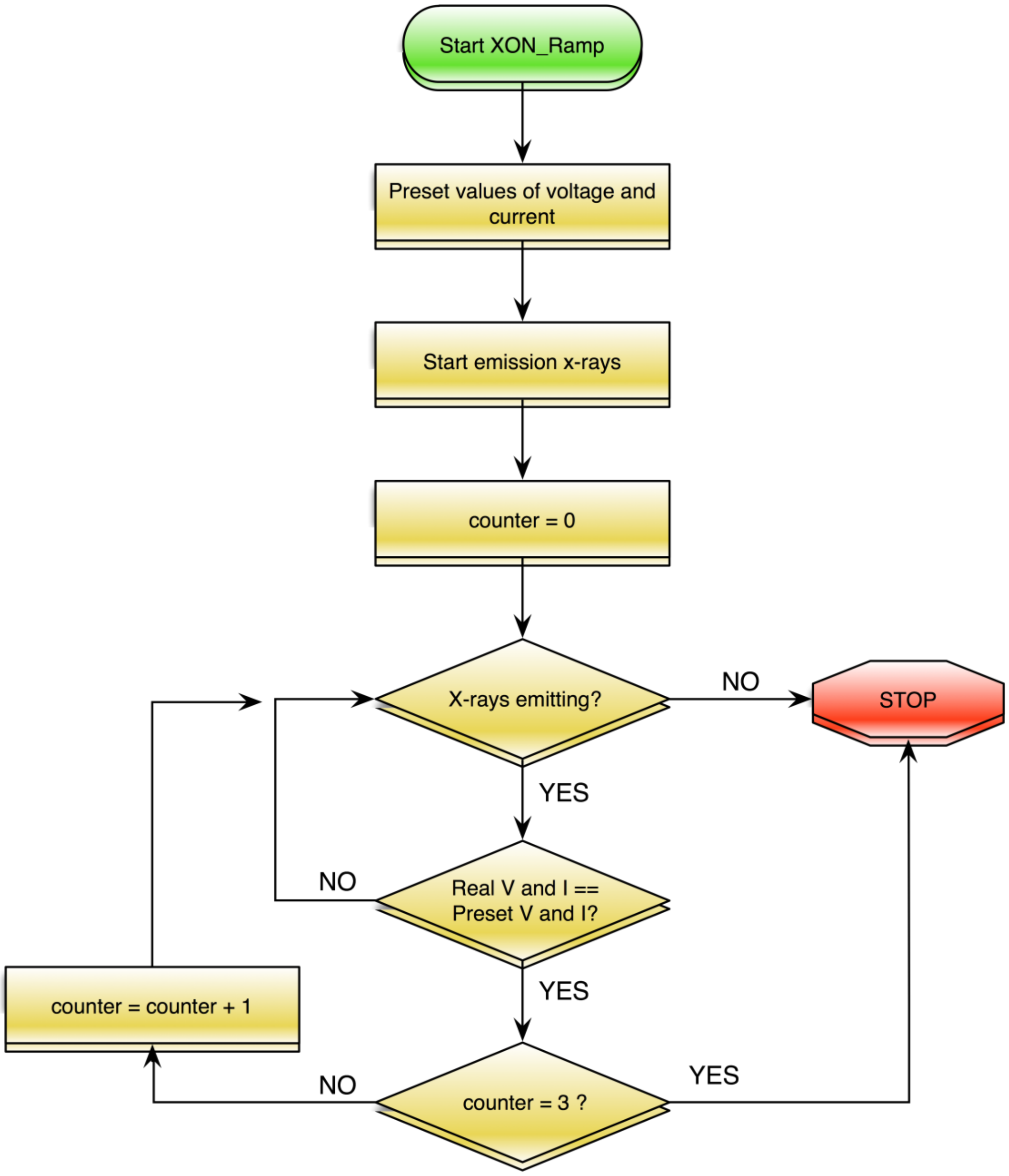


Figure 30 - Flow chart of the function x_XON_Ramp.

4.2.3.3 x_SetVC

The function “x_setVC” is a simple VI that combines the Vis of setting a desired voltage and current in one VI. This function has been made in order to be faster in the process of setting the current and the voltage.

This function has two sections. The first step, verifies that the desired voltage and current values set does not exceed the specifications power limit of 50 W. If this is exceeded then an error message to the user is retuned. Otherwise the voltage and the current will be set.

4.3 Detector module

The implementation of the control library of the detector has been the most complex task in the development of the acquisition software. The code has been developed taking advantage of a C++ library, named “FrameGrabber”. This library was built in a previous project at the Universidad Carlos III de Madrid to control another model of Dexela detector.

LabVIEW does not support the execution of C++ programming language because it is unable to create and manage instances from a class and to perform callings to its class methods. Consequently, to solve this problem, there has been implemented a code in C that works as a “wrapper”, abstracting LabVIEW from the object oriented in a C++ architecture. Then the whole code is packed into a dynamic-link library (DLL) and executed from LabVIEW.

In the following subsections, it is explained in detail the detector library functioning by first explaining how the C wrapper has been implemented, followed by the image acquiring process and ending with the explanation of some functions.

4.3.1 Wrapper from the C++ library

At the beginning of this section it was said that the integration of the C++ library in LabVIEW has been done through a C “wrapper” code that encapsulates and manages the C++ library, making it opaque to LabVIEW due to its limitation in the class objects management.

LabVIEW neither can create objects from a class nor perform calls to class methods from C++ code. However, it is possible for LabVIEW to interact with code in C programming language.

C++, through the “extern C” keyword, informs to the C++ compiler to use the C style name mangling, so the C compiler is able to find the correct symbols in the object file. In the C code, the “#ifdef __cplusplus” condition is also used because the C compiler does not know the keyword extern.

The wrapper is compiled with the C++ compiler, despite some implemented functions use C syntaxes in order to integrate the library with LabVIEW. The implemented code in C, creates and destroys objects with all its properties using the C++ “FrameGrabber” library and performs calls to the class functions.

Figure 31 shows this process more clearly with some function examples.

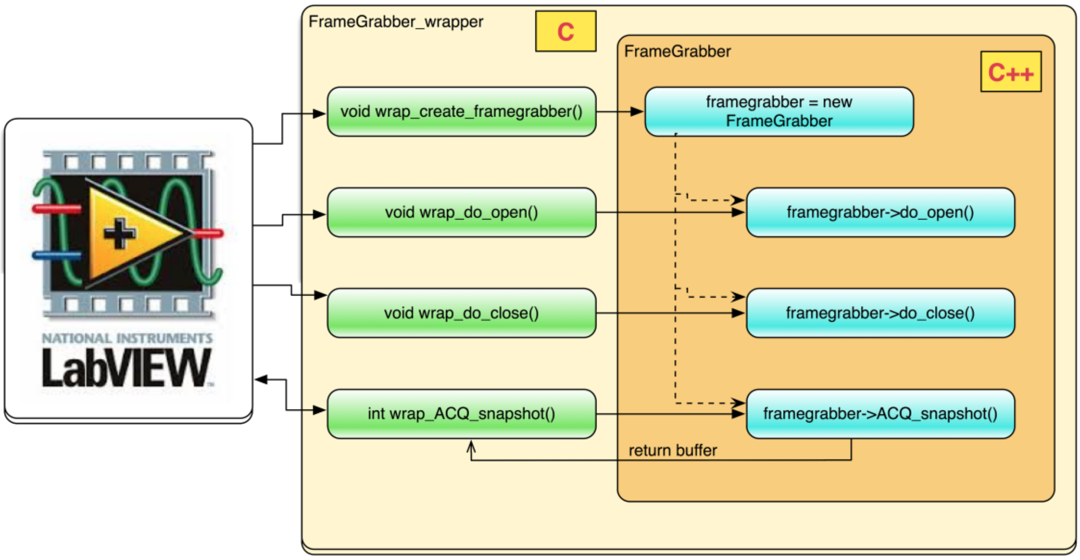
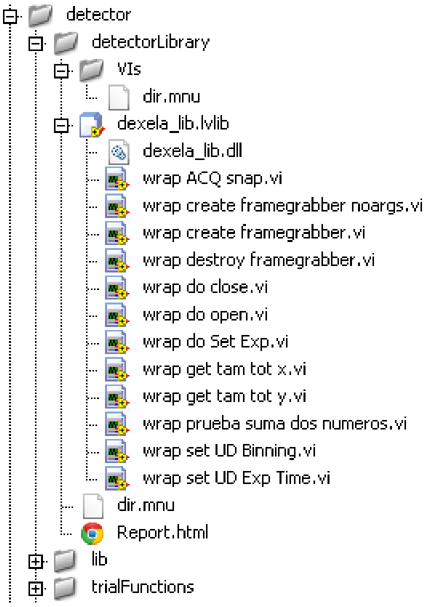


Figure 31 - Structure scheme of the integration of FrameGrabber library into LabVIEW.

Every function from the "wrapper" code written in C language will include one call to the corresponding function (or group of functions in the case of more complex tasks) from the "Framegrabber" class, using a reference to an instance of Framegrabber defined as a global variable. This avoids including this object as an input parameter in the methods that would interact with LabVIEW.

The resultant code from the C and C++ library is compiled with the C++ compiler and packed in a dynamic-link library (DLL) that is imported to LabVIEW with the "import/shared_library" option. The result in LabVIEW is a library of Vis named "detectorLibrary" with all the functions that have been exported from the DLL through the wrapper in C, as it is illustrated in Figure 32.



```
detector
  detectorLibrary
    VIs
      dir.mnu
    dexela_lib.lvlib
      dexela_lib.dll
      wrap ACQ snap.vi
      wrap create framegrabber noargs.vi
      wrap create framegrabber.vi
      wrap destroy framegrabber.vi
      wrap do close.vi
      wrap do open.vi
      wrap do Set Exp.vi
      wrap get tam tot x.vi
      wrap get tam tot y.vi
      wrap prueba suma dos numeros.vi
      wrap set UD Binning.vi
      wrap set UD Exp Time.vi
    dir.mnu
    Report.html
  lib
  trialFunctions
```

Figure 32 - Screen-shoot of the functions from the detectorLibrary library.

4.3.2 Image acquisition

For this project, it has been implemented one acquisition mode of the detector. This mode is called “ACQ_snap” and it just takes one image of the corresponding actual pixel value the detector has at that moment.

To obtain images with this mode a specific process must be followed. This process makes use of the functions from the “detectorLibrary” library explained in the previous section. Every VI used in labVIEW will call to its respective function in the C code, which correspond to the “FrameGrabber” C++ library. The steps of the process are the following:

1. **Detector initialization**
   First, an instance of the class “FrameGrabber” class must be created by calling the function “wrap_create_frameGrabber()”, which internally calls the constructor of the C++ library. Secondly, after the creation of the object, a call to the function “wrap_do_open()” must be done. This function initializes the detector configuration.

2. **Memory allocation**
   The program must reserve memory to work as a buffer and to store the acquired image until saving it into the disk. Therefore, the memory will save as much memory as the number of pixels the image has.

3. **Image acquisition**
   Once the memory space has been allocated, the function “wrap_ACQ_snap” is called and the pixels values of the detector at that moment will be copied into the buffer.

4. **Data storage and resources release**
   To conclude the process, the acquired image is storage into the disk and the memory saved for the buffer is freed. The “wrap_do_close()” function will be called to end the connection with the detector. After that, the class destructor is called using the function “wrap_destroy_frameGrabber()”.

The library saves the data in a one-dimensional vector of type *unsigned int 16 bit*. Later on, this format is resized to make a two-dimensional matrix in order to display the image.

This process is represented as a flow chart in Figure 33.

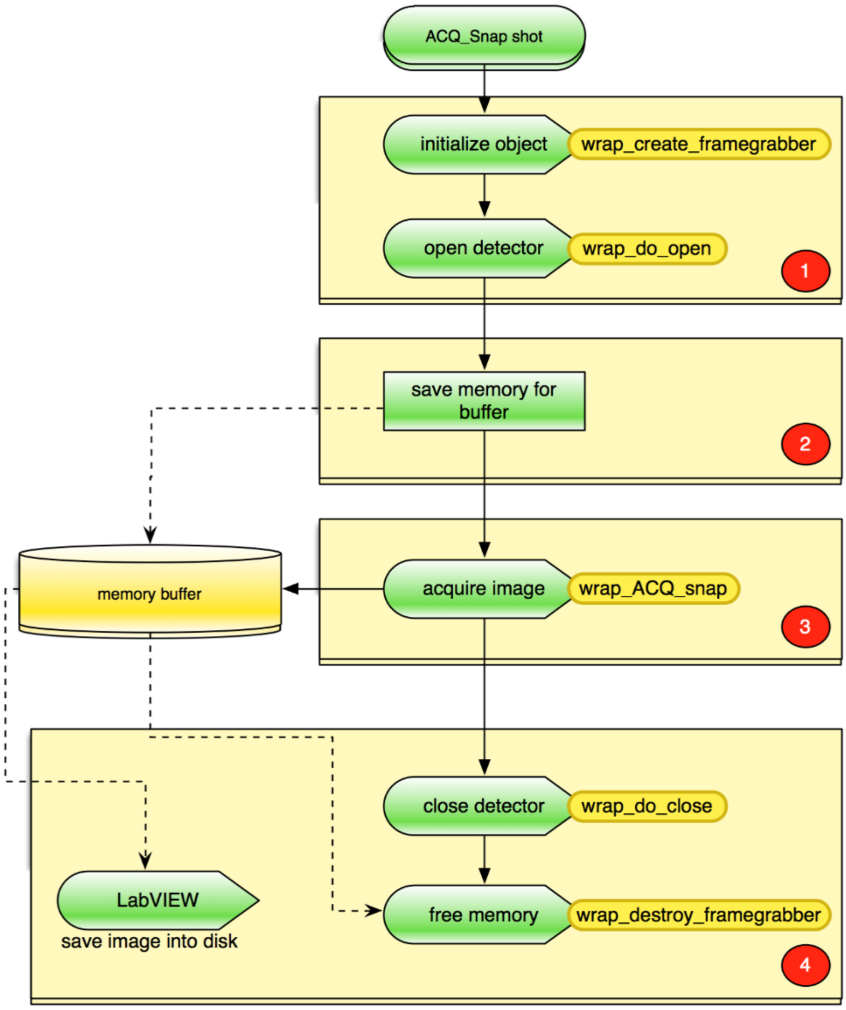


Figure 33 - Flow chart of the image acquisition process of the detector.

DESCRUMBLE

How the data of the detector is read is an important fact that must be explained.

Some detectors, in order to improve the performance of reading data, image from the detector comes out disorganized. The readout process is done dividing the surface in blocks and reading them concurrently. This procedure is common in flat-panel detectors. However, it is different depending on the model.

Therefore, for this project it has been implemented a particular descrumble process for the model used in the test bench. The process has been implemented in the "FrameGrabber" C++ library due to obtain a better performance of the final code.

4.3.3 More functions

This detector has some acquisition parameters that can be changed its configuration in order to modify the acquired image properties.

EXP_TIME

The exposure time is the integration period of the detector, which refers to the time that the light is interacting with the photodiode region of the pixel. This will produce electrons that are stored in the silicon potential well lying beneath the surface. When it expires, the voltage value of the pixel is read after the amplification stage. It is directly related with the signal level of the image as it lets to read more x-rays that reach to the flat panel detector. The default value of the exposure time is 125 ms.

BINNING

Binning allows the detector to combine charges from a group of adjacent pixels. This feature increases the pixel size of the detector as well as reduces the size of the data matrix. This is directly related with the signal level in the pixels as it increases de sensitivity and lowers the noise in the images. Binning can be set both symmetrically and asymmetrically as it details the panel manufacturer in the parameters datasheet.

4.4 Mechanical system module

The mechanical system control software of the UC3M test bench is based on a specific library called “TML_LIB_LabVIEW” which is sold by the manufacturer, in this case *Technosoft Motion S.A.* This library provides a collection of VIs with all the control commands that allow managing and controlling the different position stages of the motors.

Concerning to manage the motion system of the micro-CT, there has been developed functions that make it possible. The process will be explained in the following subsections. First, it will be described how the communication with the motor drives and the steps must be followed before performing a movement of any motor, besides the different structure variables made for the different motor units. In the following section, the “TML_LIB_LabVIEW” library structure will be exposed. Then, how the “TECHNOfunctions” library is structured, in addition of more advanced functions that have been implemented.

4.4.1 Communication process and “t_TECHNO_system”

COMMUNICATION PROCESS WITH THE MOTOR UNITS

It would be appropriate to explain in detail how does the motor drive works before explaining the different cluster structures and libraries implemented.

The control unit of the motor used in this micro-CT offers high precision of movement that permit making really short displacements in a system with its own units. When it is desired to move an axis to a specific distance, there must be done a conversion into real units (mm and degrees) into the internal units through the corresponding conversion factors. The same process is applied to the velocity and acceleration values. In addition, six of the seven axis have digital inputs connected to opto couplers that are activated when the axis are in a specific position, allowing absolute positioning without knowing the initial position of the motor.

The control units are connected between each other in a CAN bus with one of the control unit working as a master (it manages the bus in charge of sending the command information to the others) and the rest as slaves. The master drive unit establishes the connection with the working station. The connection is made through serial port RS-232.

Before sending any command to move the motors, it is needed to configure the motor unit. The following steps compose this procedure:

1. **Open serial port communication**
   The serial port indicated in the configuration file ("motor_config_testBench.txt") is opened in order to stablish the communication.

2. **Load of the motor configuration file**
   In order to configure each axis correctly, is needed to have a configuration file that is generated through the manufacturer application *EasySetup* (Technosoft Motion S.A.).

3. **Assignation of the configuration file to a specific axis**
   The file loaded in previous step is assigned to one of the axis.

4. **Selecting the axis**
   The appropriate axis selection command is sent.

5. **Axis initialization**
   A command to initialize the axis is sent, and here the configuration process ends.

6. **Power On axis**
   Turn on the drive by sending the specific command.

7. **Activate the configuration indicator**

The corresponding variable has to be set to 1 if everything has been successfully done.

Once the motors have been configured, every time that it is desired to perform a motion in any of the axis, it is necessary to select the axis before sending the command. Otherwise it can lead to select the wrong axis. In addition, it is required to check every time that the axis is configured by checking the parameter “configured” from the cluster.

“T_TECHNO_SYSTEM” & “T_TECHNO_UNIT”

As it was discussed in Chapter 3, the test bench has seven different motor drives. Each one of them controls one axis stage with its own configuration parameters. Because of this, there has been designed another structure in LabVIEW denominated “t_TECHNO_system”.

The cluster “t_TECHNO_system” stores every parameter needed at system level. Those are the values that permit configuring globally the system (serial port, total number of axis of the whole system…).

To store the parameters individually for each axis another structured has been created, because each axis may have different parameters in terms of velocity, position acceleration and movement type. The structure is denominated as “t_TECHNO_unit”. This structure is included in an array inside “t_TECHNO_system”. This allows storing one cluster for each motor unit.

The parameters are filled out with a specific VI that reads the configuration file “motor_config_testBench.txt” that must be executed before the motors are going to be used. This file has all the information needed to complete the structures. Thanks to the way it is organized, the values can be known at any time during execution of the software. The configuration file can be found on Annex C.

Figure 34 shows the structure of the cluster “t_TECHNO_system”:

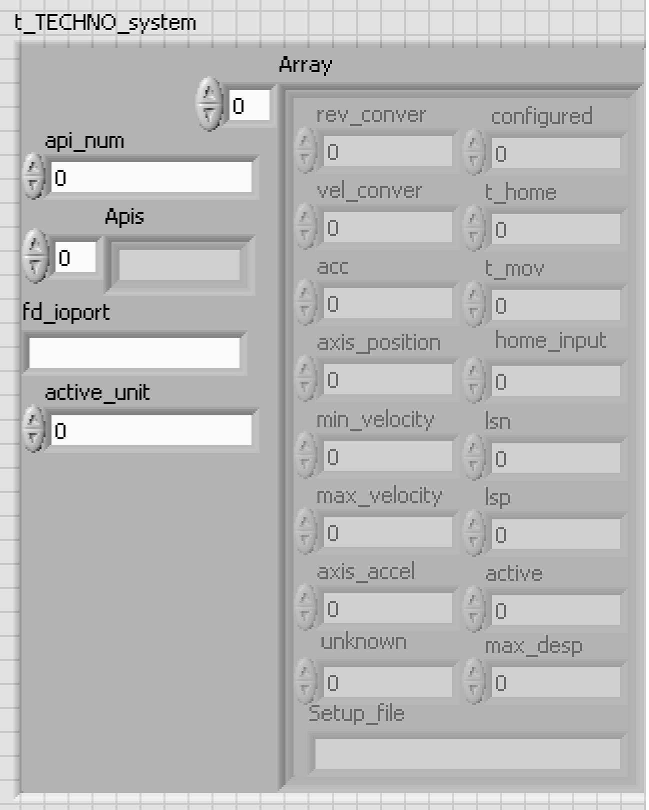


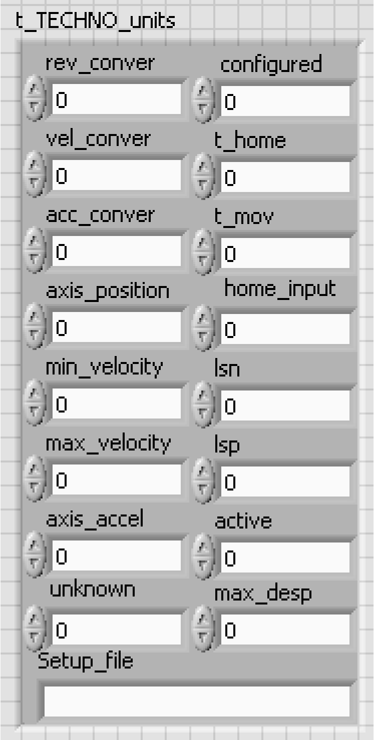
Figure 34 - Screen-shot of front-panel from the t_TECHNO_system cluster.

The Table 5 shows in detail the elements that have the cluster “t_TECHNO_system”:

Parameter	Type	Description
Api_num	Int	Number of axis
Apis	String array	axis identifier
Active_unit	Int	Active units
Fd_ioport	String	Serial port
Array	Array t_TECHNOunit	Parametric Cluster for each axis

Table 5 - t_TECHNO_system parameters.

The Figure 35 shows the cluster structure of “t_TECHNO_unit”:



t_TECHNO_units

rev_conver 0

configured 0

vel_conver 0

t_home 0

acc_conver 0

t_mov 0

axis_position 0

home_input 0

min_velocity 0

lsn 0

max_velocity 0

lsp 0

axis_accel 0

active 0

unknown 0

max_desp 0

Setup_file

Figure 35 - Screen-shot of the front-panel from the t_TECHNO_units cluster.

Table 6 explains the parameters that the "t_TECHNO_units" contains associated to each axis:

Parameter	Type	Description
Rev_conver	Int	Distance conversion factor
Vel_conver	Double	Velocity conversion factor
Acc_conver	Double	Acceleration conversion factor
Axis_position	Int	Axis position
Min_velocity	Double	Minimum velocity of the axis
Max_velocity	Double	Maximum velocity of the axis
Axis_accel	Double	Maximum acceleration of the axis
Setup_File	String	Configuration file path
Configured	Int	Flag that determines if the axis is ready to be moved
T_home	Int	Home type movement
T_move	Int	Type of movement of the axis
Home_input	Int	Digital input to detect position home in the axis
lsn	Int	Negative movement digital input
Lsp	Int	Positive movement digital input
Active	Int	If axis is active
Max_desp	Int	Maximum displacement that the axis can
unknown	Anything	Free space to be used for what could be needed

Table 6 - Parameters description of the cluster t_TECHNO_units

4.4.2 TML_LIB_LabVIEW library

The “TML_LIB_LabVIEW” library is composed of a set of VIs that permits the user to control and to configure the motor drives of the mechanical system. The motor drive manufacturer provides this library and it works over a dynamic-link library (DLL).

The VIs of the library are classified according to the functionality that they provide. The most used categories for this project are the following:

- **Motion programming:**
  They allow the user to program specific motion commands to the selected axis. These commands can be executed right immediately after they are called or can be programmed to be executed after sending a motion update command.

- **Motor commands:**
  It includes functions to upload programmed commands, turn on and off a drive, stop motion, etc.

- **Events:**
  These sets of VIs allow setting events in order to execute different actions when certain situations occur. For example, an event can be set when a movement has finished, when the motor has reached to a certain position, etc.

- **IO handling:**
  These commands control the digital inputs of the drives. Also, they allow making positioning sequences where the initial position of the stage is unknown, such as “homing”.

- **Data Transfer:**
  The included functions permit knowing internal parameters of the axis.

- **Drive administration:**
  These commands let initialize and setup the drives of the motor units.

- **Communication setup:**
  It includes the commands to open and close the serial port.

4.4.3 TECHNOfunctions library

The “TECHNOfunctions” library consists in a set of VIs that combine the library “TML_LIB_LabVIEW” and the LabVIEW structure “t_TECHNO_system” in order to build more complex command sequences. These sequences perform specific positioning and configuration actions of the micro-CT system.

Almost all the VIs form the library have the same prototype as it shown in Figure 36. The inputs and outputs are described in detail below:

- **Inputs:**
  - LOG: reference to the panel where the messages from this VI are exposed.
  - Motor: motor axis that is going to be moved.
  - t_TECHNO_system: cluster with parameters of the mechanical system.
  - Error in: cluster of error.
  - Others: depending on the function it could be included more inputs, as for example distance to travel.
- **Outputs**
  - t_TECHNO_units: structure with the specific data from the selected motor axis.
  - Error out: cluster of error.

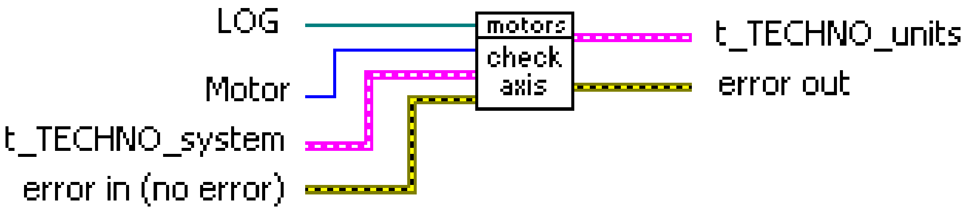


Figure 36 - VI prototype from TECHNOfunctions library.

The most important VIs are described in the following section:

- **configureTECHNO:**
  The configuration process of the drives is done with this VI. It reads the configuration file “motor_config_testBench.txt”, and then it fills out the “t_TECHNO_system” cluster. It runs automatically the communication process that has been described previously.

- **t_check_axis:**
  This VI selects the desired axis in order to be moved. It has a special structure that makes sure that the axis selected has been selected by processing every possible error that could happen during the process.

- **t_moveABS:**
  This function uses the combination of different VIs to perform a motion to an absolute position from the home position of the axis.

- **t_home_pos_individual:**
  It performs the home sequence of the axis, which is a sequence of movement that brings the stage unit to the user defined initial position. The sequence starts by moving the axis to the initial position at a constant velocity. When the axis reaches the correspondent digital input the movement of the motor is stopped and a hysteresis movement starts. The hysteresis movement consists in moving the motor to the opposite direction until the digital input is deactivated, and when this happens it goes back at a really low pace until the digital input is activated again. This sequence of movements is necessary because the motor does not stop instantaneously due to the inertia forces that make it to go a little bit farther. Therefore, by going very slow this error is reduced considerably. This is also a good method in order to make sure that the motor is at the desired position precisely. In Figure 37 can be seen a diagram that explains in detail this process.

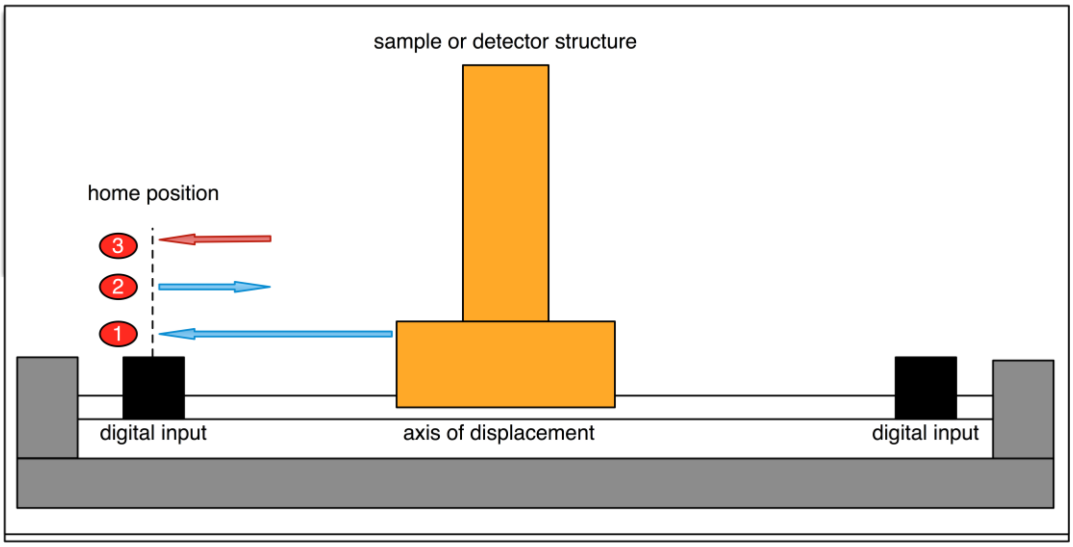


Figure 37 - Diagram of the sequence of t_home_pos_individual.

4.4.4 Motion Controller

As it was said before, the UC3M test bench has seven different motor drives that move seven different axes. Six of these axes perform linear motions. The linear movements are used to move the sample plate and the detector mechanical set freely in the x-y-z. As there are many motor units, a simple motion controller console has been implemented to move all the stages.

The motion controller manages and controls the motors with the end purpose of moving around the different axis in an uncomplicated, precise, comfortable, not dangerous manner. It lets the user to control the motor position from a single VI.

In order to make the motion controller simple and useful, the graphical user interface has been designed with the essential parameters to be changed by the user. This interface is shown in Figure 38. Table 7 shows the inputs that the user can change or select in the motion controller.

Input	Type	Description
Motor Selection	Int	Selection of axes from 1 to 7.
Position	Float	Absolute position of the motor in mm.
Velocity	Float	Velocity of the movement in mm/s
Acceleration	Float	Acceleration of the movement in mm/s^2
Waiting time until move	Integer	Time to wait for the axis to move in seconds
Home Position	Bottom	Send motor axis to home position
Play	Bottom	Start the desired movement
Stop	Bottom	Stop movement
STOP de program	Bottom	Stop the CONTROLADROA
Set position to 0	Bottom	Set absolute position to 0

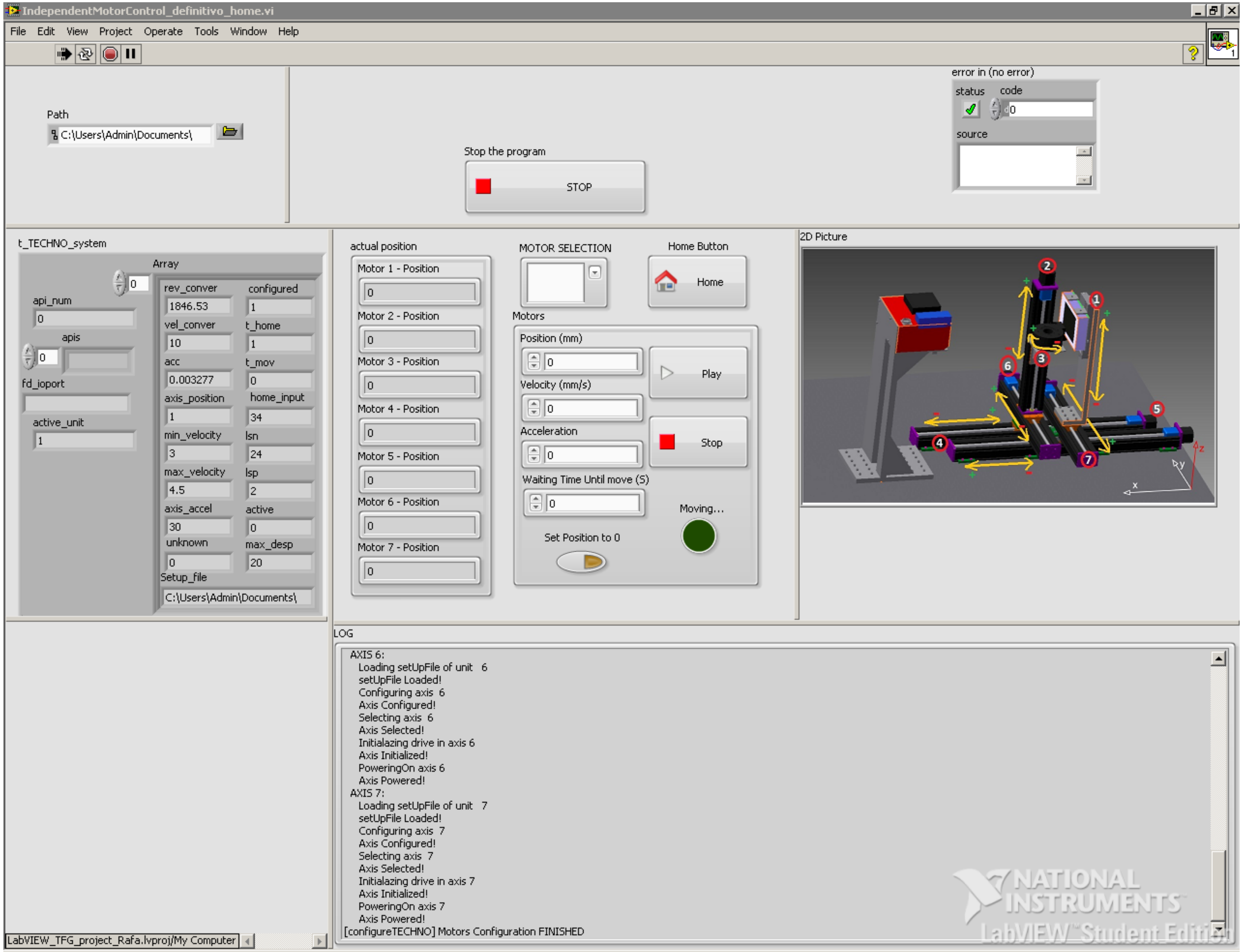
Table 7 - Motion Controller inputs description.

The motion controller also has some parameters that give information to the user. These parameters are described in Table 8.

Parameter	Type	Description
Actual position	Float	Gives the absolute position of the motor
Moving…	Light indicator	Lights up when the motor is moving
t_TECHNO_system	Cluster	Mechanical system structure with its parameters
LOG	String	Messages of the VI to the user
Micro-CT diagram	2D image	Diagram of the micro-CT
Error out	Cluster	Cluster of error

Table 8 - Motion Controller outputs description.

The motion controller moves the motor axes to absolute positions. This means, for example, that if the position is 10 mm, the motor is at 10 mm with respect to home position. As a safety limitation, it can only perform motions of one motor at a time.



Path: C:\Users\Admin\Documents\

error in (no error)
status ✓ code 0
source

api_num: 0
apis: 0
fd_ioport:
active_unit: 1

rev_conver: 1846.53	configured: 1
vel_conver: 10	t_home: 1
acc: 0.003277	t_mov: 0
axis_position: 1	home_input: 34
min_velocity: 3	lsn: 24
max_velocity: 4.5	lsp: 2
axis_accel: 30	active: 0
unknown: 0	max_desp: 20

Setup_file: C:\Users\Admin\Documents\

Motor 1 - Position: 0
Motor 2 - Position: 0
Motor 3 - Position: 0
Motor 4 - Position: 0
Motor 5 - Position: 0
Motor 6 - Position: 0
Motor 7 - Position: 0

Position (mm): 0
Velocity (mm/s): 0
Acceleration: 0
Waiting Time Until move (S): 0

```
AXIS 6:
  Loading setUpFile of unit  6
  setUpFile Loaded!
  Configuring axis  6
  Axis Configured!
  Selecting axis  6
  Axis Selected!
  Initialazing drive in axis 6
  Axis Initialized!
  PoweringOn axis 6
  Axis Powered!
AXIS 7:
  Loading setUpFile of unit  7
  setUpFile Loaded!
  Configuring axis  7
  Axis Configured!
  Selecting axis  7
  Axis Selected!
  Initialazing drive in axis 7
  Axis Initialized!
  PoweringOn axis 7
  Axis Powered!
[configureTECHNO] Motors Configuration FINISHED
```

LabVIEW_TFG_project_Rafa.lvproj/My Computer

Figure 38 - Graphical user interface of the Motion Controller.

4.5 Step and Shoot acquisition protocol

After implementation of the set of libraries to control each hardware module of the UC3M test bench individually, it is time to combine all of them in order to start designing an acquisition protocol. For this project, the *step and shoot* acquisition protocol has been implemented with the final goal of performing micro-CT data acquisitions and to evaluate the system.

For the implementation of this acquisition protocol there has been designed a simple graphical user interface that allows acquiring projections of the sample of study. The goal is to use these set of projections to create a 3D view of the sample using the reconstruction software.
()

GRAPHICAL USER INTERFACE

The graphical user interface is shown in Figure 39. It has been designed in order to be as simplest as possible providing to the user the possibility of changing the basic parameters of an acquisition. The graphical user interface has two separate sections, the status panel and the acquisition panel.

- **Status panel**
  The status panel is divided in two parts:
  - **Image panel:**
    This panel displays the acquired projection images in real time while an acquisition is performed.
  - **Text window:**
    This window informs the user about the execution of the program. It works as a log, showing informative messages of the system status and the evolution of the running processes.
- **Acquisition Panel**
  This panel is divided in three parts:
  - **Numerical inputs:**
    - Number of projections in one revolution.
    - Exposure time of the acquisition of the detector, in ms.
    - Voltage of the x-ray beam, in kV.
    - Current of the x-ray beam, in µA.
  - **Buttons:**
    - Acquire: it starts the acquisition.

        - Cancel: it cancels the process when the program is acquiring.
        - Stop: stops the full program.
    - **Beacon:**
      A green light lights up when the program is acquiring in the green symbol "AQUIRING". It turns off when the acquisition has finished.

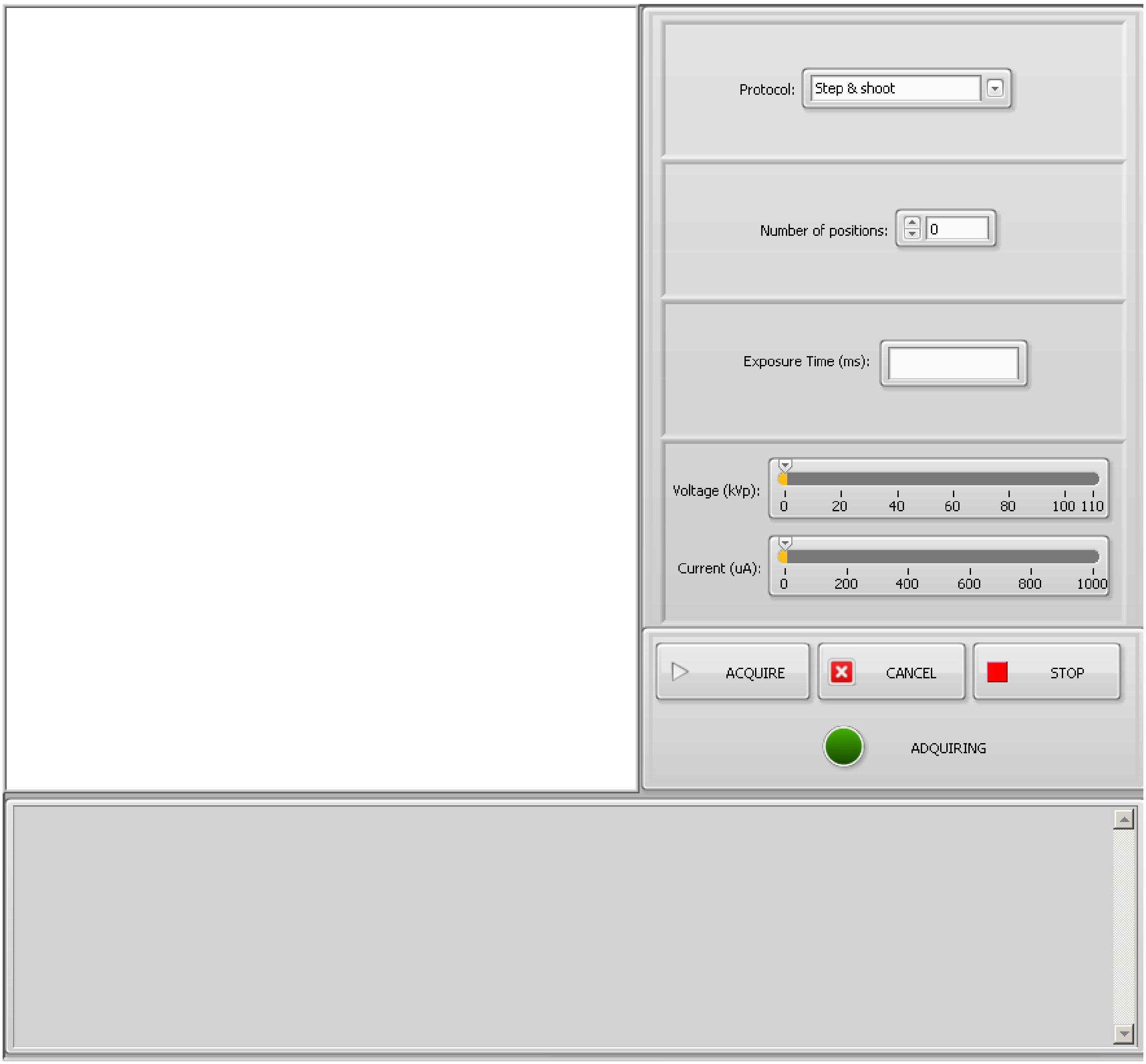


Figure 39 - Graphical user interface of the Step and Shoot acquisition protocol.

STEP AND SHOOT ACQUISITION PROCESS

Step and shoot acquisition protocol consists of alternating the acquisition of one image with a rotational movement of the sample. Below the sequence of steps is exposed commenting the actions of the user in the interface and Figure 40 represents the process in a diagram.

Step 1 - Initialization of the system and introduction of parameters

First, when the VI is executed the configuration parameters of the motors and the x-ray source are loaded and the system is initialized. Now the user is able to set all the desired parameters of the acquisition. These parameters are the number of projections in one turn, the exposure time of the acquisition detector, the voltage and the current of the x-ray beam.

Step 2 - Acquisition process

When the acquire button is clicked, the acquisition process begins.

Step 2.1 – Step calculation

Here the number of projections the user introduces will be divided by 360º, one complete turn. This will give the step that the sample must rotate for each projection.

Step 2.2 – X-ray source turned on

The green light starts lighting and the x-ray source is turned on using the ramp function in order to assure the desired energy.

Step 2.3 – Step and Shoot

In this part the different projections are obtained, repeating the following steps until all the projections have been performed:

1. **Acquire and display image**
   The detector acquires the image, and then is displayed in the user interface.
2. **Ask state to the x-ray source**
   It checks that the x-ray source is still emitting.
3. **Save image**
   The program saves the image in a binary file.
4. **Ask the state to the x-ray source**
   It checks that the x-ray source is still emitting.
5. **Rotate sample**

The sample rotates the step distance that was calculated before. When the movement finished the angle of the position of the sample is stored in an array.

Steo 2.4 – X-ray turned off
When all the projections have been acquired, the x-ray tube stops emitting radiation. The green light is turned off.

Step 2.5 – Save angles
The array with the angle positions of the sample rotation is stored in a file named “angles.txt”. These angles will be useful for the 3D reconstruction of the sample.

This step process could be stopped at any time if the user clicks the Cancel button.

Step 3 – Stop the program
When any acquisition has finished, if the STOP button is clicked, the program stops running.

To clarify, the detector saves the images into a binary file in a specific folder already specified by the program. All projection files are numbered with the extension “.ct”, being the first projection named by “0”. The files are written in *unsigned int 16-bit* as it was mentioned in section 4.3.2.

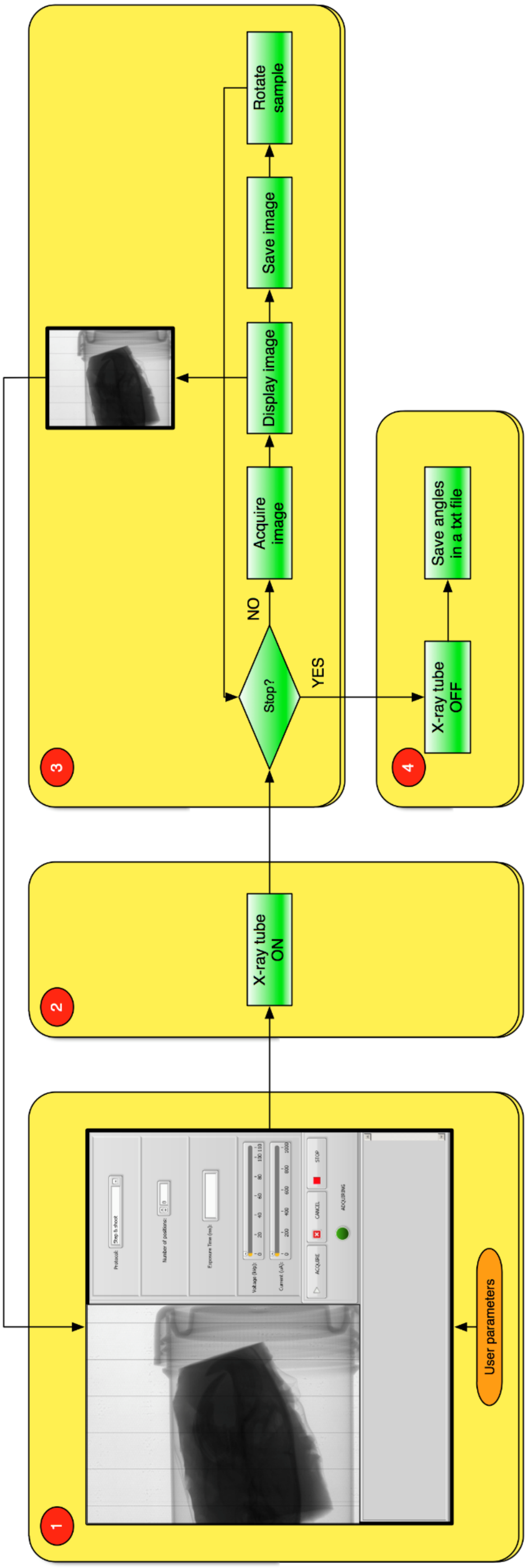


Figure 40 - Diagram of process of the Step and Shoot acquisition protocol

Chapter 5

System evaluation

After completing the implementation of the acquisition software in the UC3M test bench, an acquisition of real data is performed. It was done in order to validate the application and the hardware control libraries and prove that the machine is ready to acquire tomographic data that can be reconstructed with an appropriate software. For this task, several acquisitions of test phantoms will be performed using the “step and shoot” protocol. A set of projections from each sample will be obtained during this process. Then, the acquired raw projections will be corrected from system non-idealities in order to avoid the appearance of artifacts during the reconstruction procedure. Finally, the reconstructed volume will be inspected in order to validate and evaluate the results.

5.1 Image Acquisition

As mentioned before, several “step and shoot” acquisitions have been performed. The scanned samples were a cylindrical phantom of Polymethyl methacrylate (PMMA) with four metallic wires inserted longitudinally on its surface and a rat skull.

The details of the configuration of the system parameters in terms of source energy, acquisition protocol and geometry are presented in Table 9. The source parameters and the exposure time were chosen to maximize the contrast to noise ratio in the images and ensure a proper level of signal in the detector and avoid saturation.

Parameter	ACQ Phantom	ACQ Rat Skull
Number of projections	360	360
Step rotation	1º	1º
Voltage	68 kV	68 kV
Current	600 µA	600 µA
Power	40.8 W	40.8 W
Binning	1x1	1x1
Exposure Time	125 ms	125 ms
DSO (distance source object)	470 mm	270 mm
DDO (distance detector object)	145 mm	345 mm
Magnification Factor	1.3	2.28
Image Size	1944x1536 pixels	1944x1536 pixels

Table 9 - Acquisition parameters

Each of the acquired set of data is composed by 360 projections with an image size of 1944x1536 pixels, stored separately in binary files with *unsigned int 16-bit* data format. Figure 41 shows one projection of each of the samples.

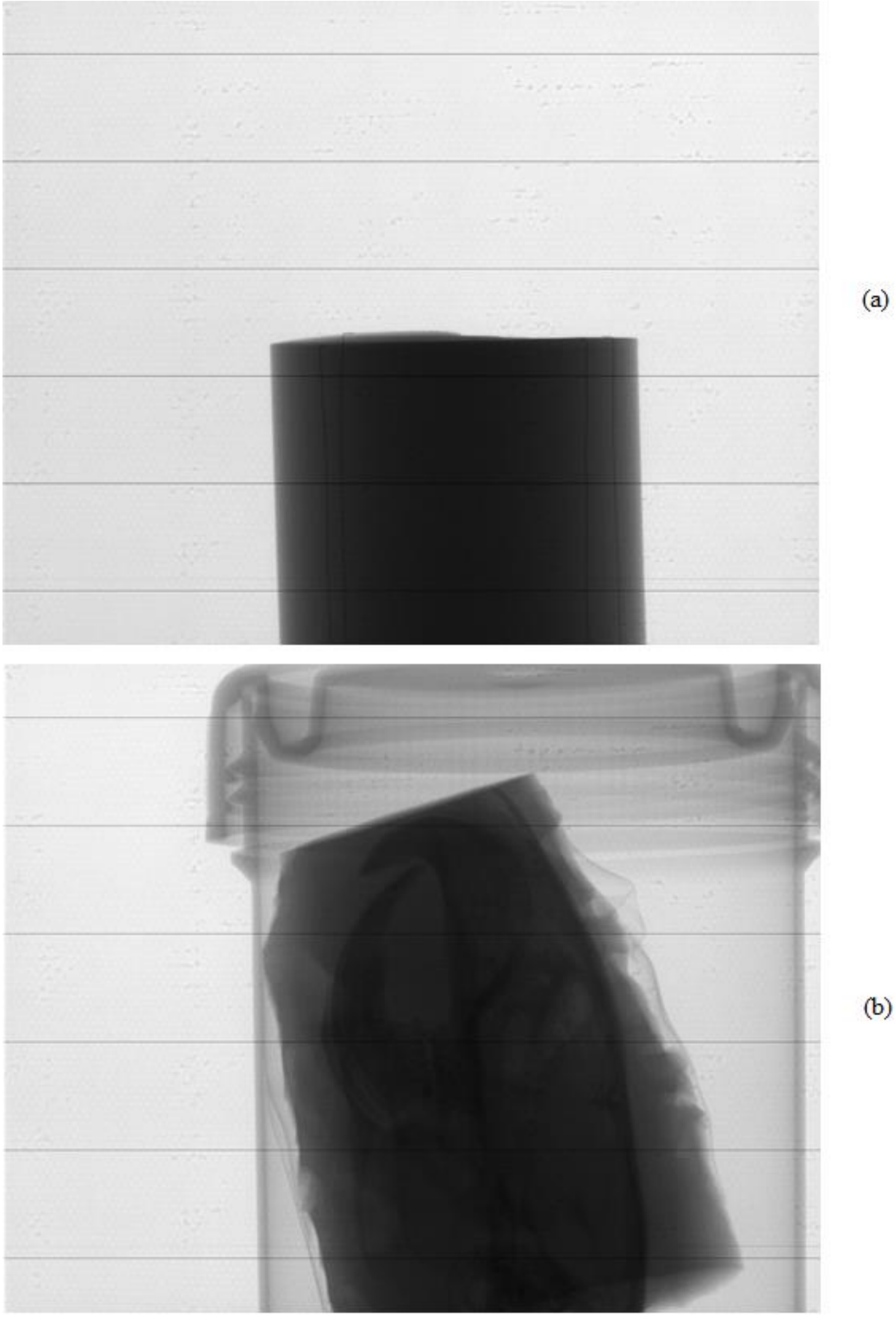


Figure 41 - Raw projections of (a) a PMMA phantom and (b) a rat skull.

Non-idealities of the system induce the appearance of defects in the images that if not corrected would produce artifacts in the reconstructed images. This step is common in every cone beam flat-panel based CT system and is not related with the implementation of the acquisition software. Below, these defects and their cause are explained in detail:

- **Image inhomogeneities:**
  Images are affected by two types of effects due to the properties of a cone x-ray beam and the electronic components of the detector:

  - Dark currents: The thermal noise of the flat panel electronics induce to the appearance of small currents in the photodetectors even during the absence of radiation. This fact causes the panel to produce an image with positive values, instead of having zero values when the source is not emitting. The result in the projections is a positive offset in the pixel values of the images that is corrected by subtracting a dark image. A dark image consists in an average of a certain number of images, in order to low the noise and improve the image statistics, acquired without radiation. Dark current intensity is dependent from the temperature of the panel; therefore, dark image must be acquired before performing the acquisition as the panel varies its temperature with the radiation.

  - Inhomogeneous illumination: Cone beam x-ray sources produce inhomogeneous intensity of illumination in the panel. The cone shape of the beam tends to deliver more photons near to the central ray than in the surroundings, therefore, the level of signal in the pixels of the acquired images is spatially dependent. This effect is fixed dividing each projection by an illumination map to get a homogeneous background in the images. This map is an average of an arbitrary number of flood images, which are projections with the source configured with the same energy as in the acquisition and no sample.

- **Flat-panel detector defective elements:**
  Flat-panel detectors have imperfections in the functioning of the pixels due to the manufacturing process. The sensor has defective elements whose response to radiation is abnormal or even absent. They appear in the images as dark dots or lines with pixel values much lower than the adjacent. There are defective pixels, rows and columns. This defect is corrected interpolating with the values of the adjacent pixels, rows or columns.

After the acquisition process, a set of 60 dark and flood images were averaged in order to perform the corrections of the projections.

5.2 Image preprocessing

The correction of the defects previously mentioned is performed as a post-processing step prior to the reconstruction. This process has been done using MATLAB 2013a (Mathworks, USA).

The corrections made are the following:

- **Flat-field correction:**
  The flat-field correction is applied using the following formula for each pixel of a projection [10]:

$$g_{corrected}(u,v) = \frac{g_{raw}(u,v) - g_{dark}(u,v)}{g_{flood}(u,v) - g_{dark}(u,v)}$$

  Being:
  $g_{corrected}$: Corrected image.
  g_{raw}: Raw image.
  g_{dark}: Dark image.
  g_{flood}: Flood Image.

- **Defective elements correction:**
  The defective pixels are corrected interpolating from adjacent elements. Elements were identified previously.

The resultant projections after the corrections are shown in Figure 42 and Figure 43. Now, the defective elements of the detector are not noticeable in the images and the background is completely homogeneous. Data is ready to be reconstructed.

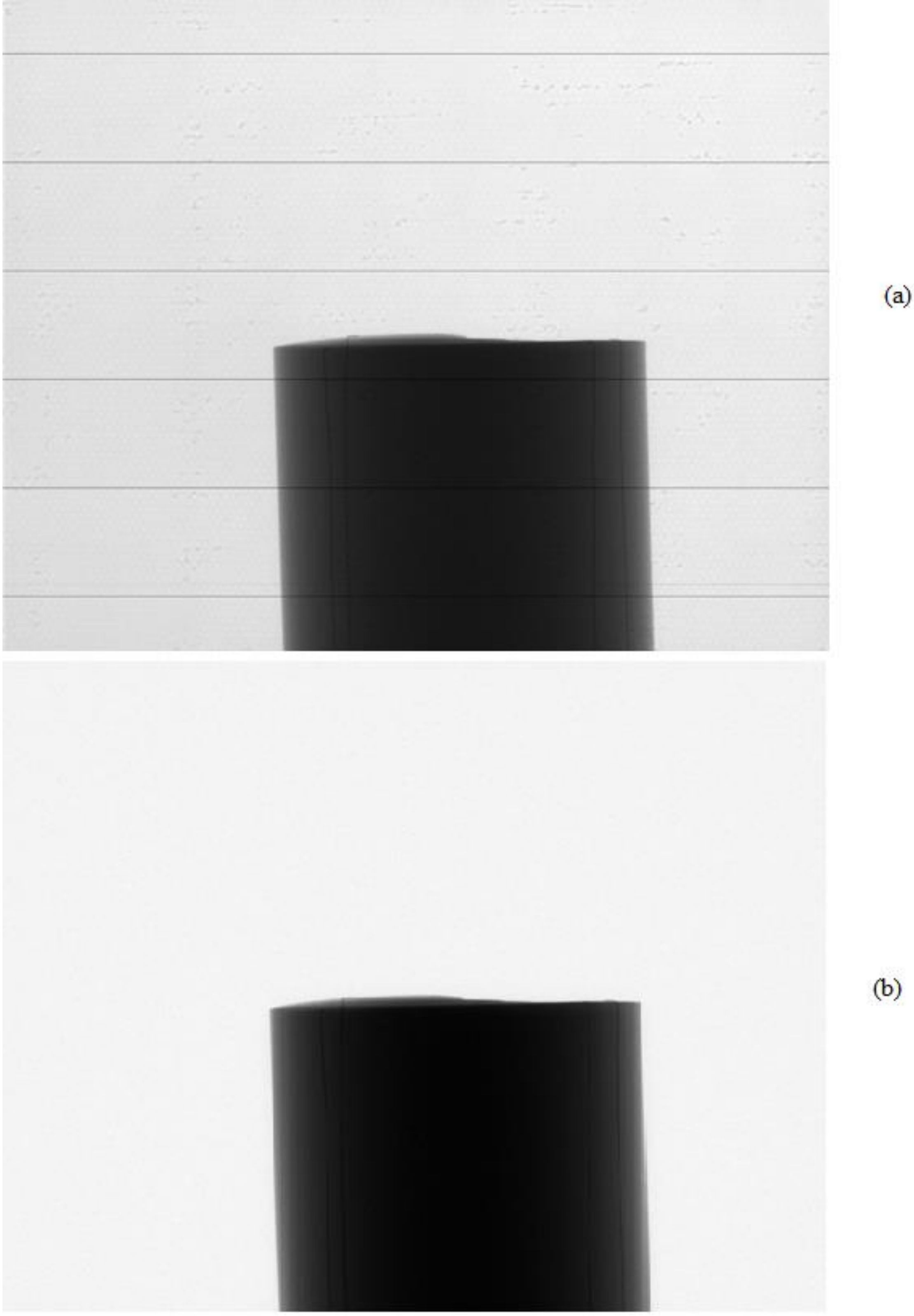


Figure 42 - PMMA phantom projection: (a) Raw projection and (b) Corrected projection.

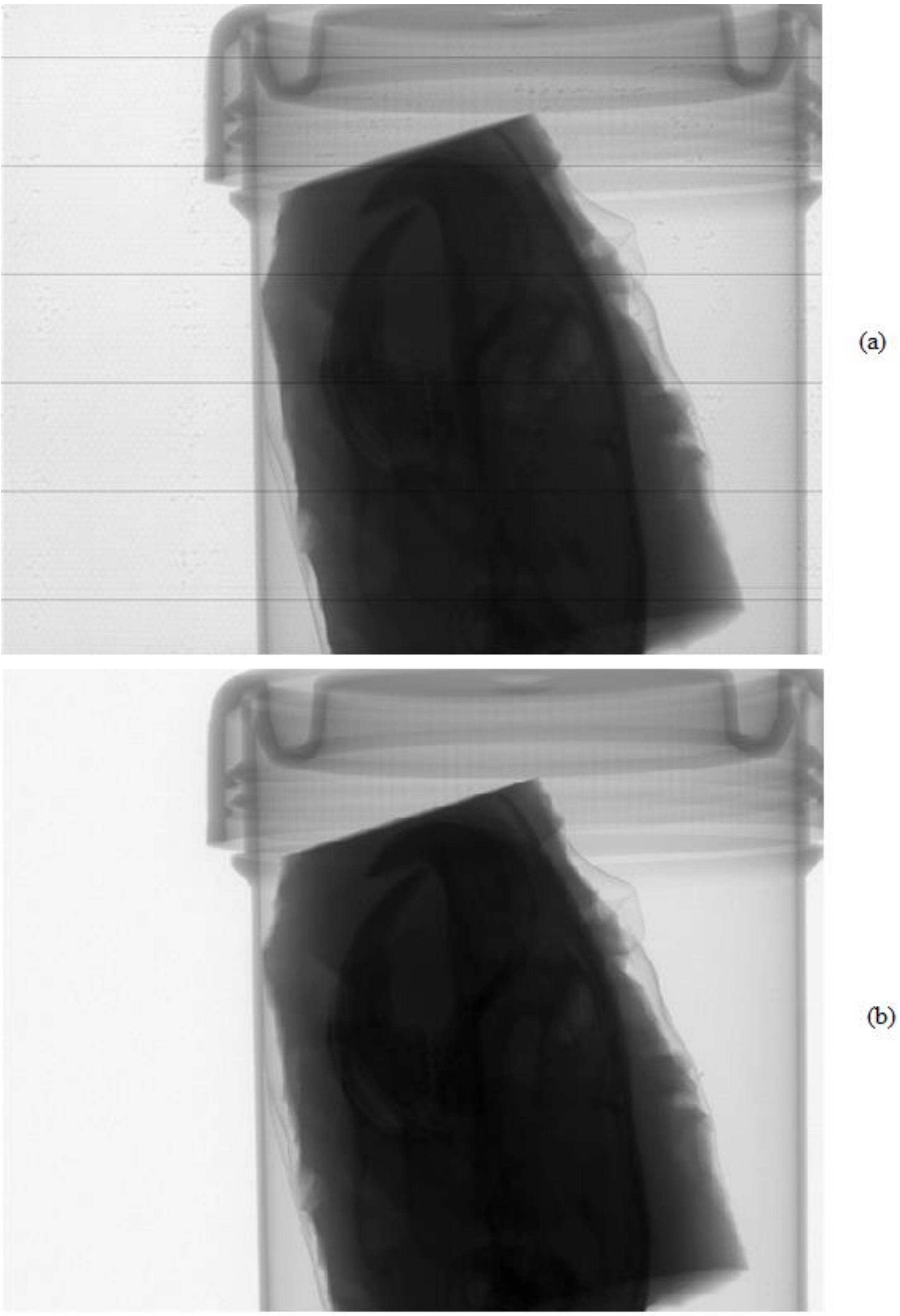


Figure 43 - Rat skull projection: (a) Raw projection and (b) Corrected projection.

5.3 Image reconstruction

From the 360 images of 1944x1536 pixels it has been obtained a reconstruction with the following parameters:

PMMA PHANTOM

- Binning: 1x1
- Number of axial sections: 1536
- Image resolution: 1944x1944 pixels
- Voxel size: 0.057x0.057x0.057 mm.

Result:

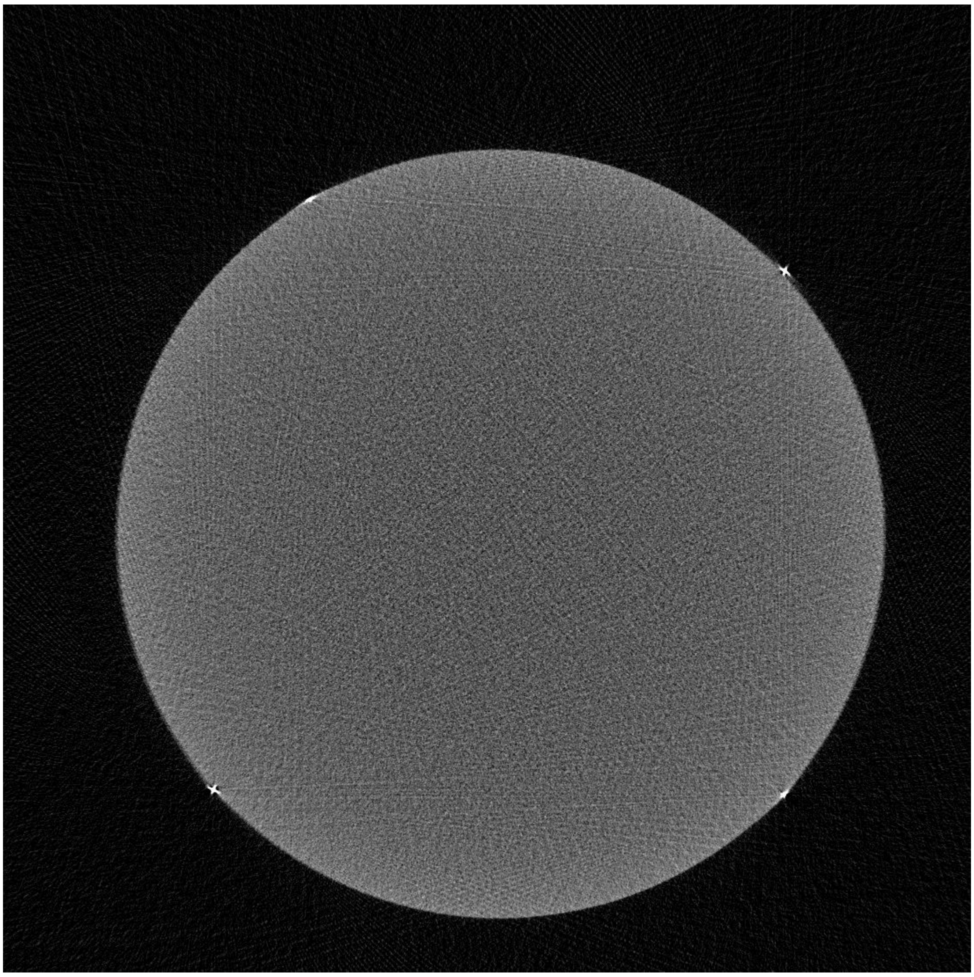


Figure 44 - PMMA reconstruction of one axial section.

Figure 44 shows one axial section from the final reconstructed volume. It can be seen a perfect circle, which represents the axial section of the cylindrical phantom. Moreover, the four metallic wires that it has on the surface are represented as four brightest points in the circumference perimeter of the circle.

However, if we zoom in the axial section of Figure 44, represented in Figure 45, it can be appreciated some straight lines that should not be there in the whole image. These lines are called streaks artifacts. This occurs due to the insufficient number of projections that have been acquired to the sample. This could be corrected by acquiring more projection around the sample.

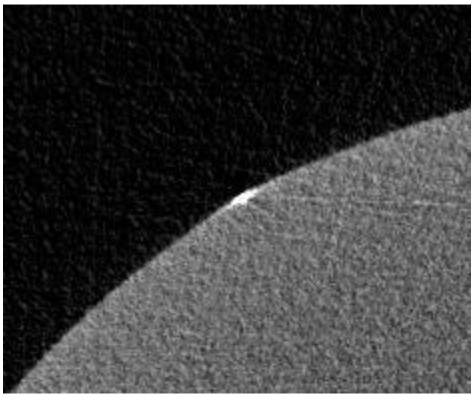


Figure 45 - PMMA reconstruction: section closed- up.

In addition, it can also be noticed from the projections some kind of noise along the whole image. This is due to the binning parameters used for the acquisition. It has been used binning 1x1, which means that no average of the pixels values has been performed. Therefore, it implies more noticeable random noise affecting to the final reconstruction image.

RAT SKULL

- Binning: 4x4
- Number of axial sections: 384
- Image Resolution: 486x486 pixels
- Voxel size: 0.1x0.1x0.1 mm

Result:

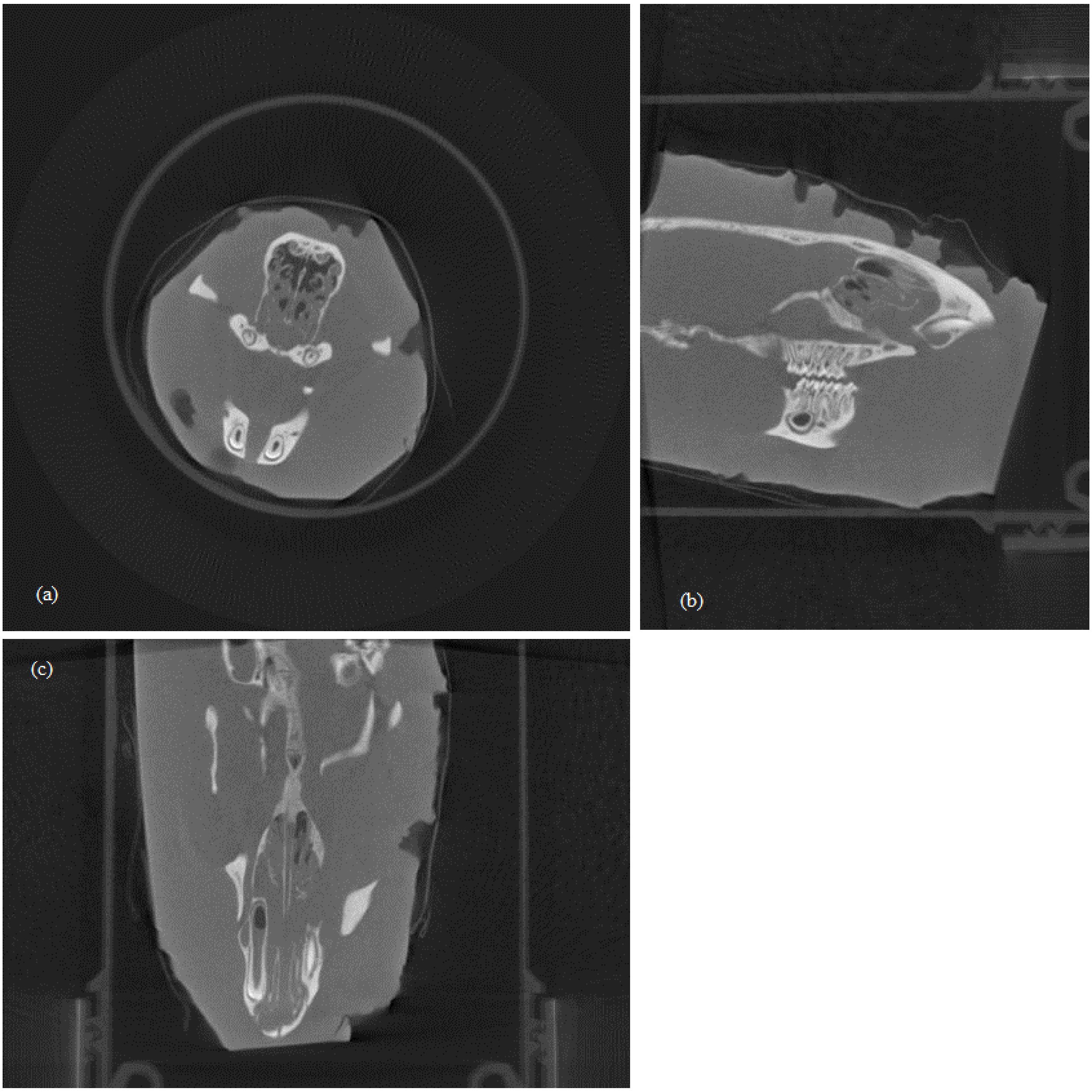


Figure 46 - Rat skull reconstruction. (a) Axial section. (b) Sagittal section. (c) Coronal section.

The rat skull reconstruction can be seen in **¡Error! No se encuentra el origen de la referencia.**. From the image, the bone structures of the skull are represented as the whitest sections of the image. These structures can be easily distinguished from the medium, noticing the different bones of the rat skull. Moreover, as the sample was placed inside a plastic recipient, it can be seen in image (a) a circumference representing it.

In addition, the rat skull has been reconstructed using binning 4, instead of binning 1x1 as with the PMMA phantom. The result can be appreciated with much less noise than the reconstruction of the PMMA (Figure 44).

Moreover, from the final reconstructed volume of the samples used it is not observed any artifact that could aggravate the quality of the data obtained being possible to differentiate the different axial sections of the samples. Therefore, the reconstruction does not have any apparent artifacts caused by the software developed for the UC3M test bench.

Chapter 6

Conclusion

6.1 Conclusion

The main goal of this project is to create a set of software libraries in order to control the new high-resolution in-vitro micro-CT designed by the Universidad Carlos III de Madrid (UC3M). LabVIEW has been used as the development platform and programming language as its particular features fulfill the requisites and specifications of the project.

The work started with the development of control tools for the hardware elements of the test bench. For this, software libraries for each individual module of the micro-CT have been designed and implemented. This allows taking advantage of all features that the elements of the system offer and allows building advanced functions by combining them.

Then, a “step and shoot” acquisition protocol has been developed and implemented by combining the control libraries. This acquisition protocol makes possible to acquire tomographic data, which consist in a set of images of the sample from different angles by rotating it around its longitudinal axis.

After that, a graphical user interface for the acquisition protocol has been implemented in order to allow the user to carry out acquisitions in and simple and easy manner.

Finally, the acquisition program has been tested in the device in order to verify that the implemented software, libraries and acquisition protocol, works as expected. To do so, a set of acquisitions of different samples have been performed. The obtained projections data have been processed and then introduced into the reconstruction software. The reconstructed volumes of the samples have no apparent artifacts that could be produced by the malfunctioning of the system. Hence, we can conclude that the system is able to acquire valid tomographic data, controlled by the developed software.

To conclude, the software implemented for the UC3M test bench has provided data suitable for tomographic reconstruction, proving that the software performs its task properly. Here, it can be said that the micro-CT is now ready to be used for its applications with high future expectations.

6.2 Future work

For future work lines, this project leads to the following:

- Implementation of advanced features in the detector. The implementation of the continuous acquisition mode of the detector would allow for the implementation of advanced acquisition protocols in the CT, that would improve the performance of it. Also, a sequential acquisition mode would be very useful as it reduces the noise of the “step and shoot” projections by averaging multiple frames for a single projection.

- Implementation advanced acquisition geometries: helical or continuous rotation acquisitions would expand the possibilities of the micro-CT allowing acquiring bigger samples with high resolution in less time.

- Development of an advanced graphical user interface for the acquisition: Increasing the number of parameters, as for example magnification factor, desired directory to save the images, etc. would be an useful feature for the user

- Mechanical presets: Let the user to save certain positions of the elements of the micro-CT that are more practical and ideal for each acquisition protocol. This would simplify the control of the geometry of the system and the control of the motion stages.

Chapter 7

Project Budget

HUMAN RESOURCES

Table 10 represents total cost for the salary of the personnel that has worked in the project.

Human resources			
Category	**Hours**	**Cost per hour**	**Cost (€)**
Asier Marocs Vidal (Senior Engineer)	50	€ 30,00	€ 1.500,00
Rafael Moreta Martinez (Engineer)	600	€ 20,00	€ 12.000,00
		TOTAL:	**€ 13.500,00**

Table 10 - Budget: human resources costs.

MATERIALS

In Table 11 the material costs are presented. For the characterization equipment calculation cost it has been followed the following procedure:

Formula:

$$\frac{A}{B}\cdot C\cdot D$$

A: months of use of the equipment.
B: depreciation method (60 months).
C: equipment cost (no IVA)
D: % use

Materials

Materials	**Cost (€)**
Working Station	€ 500,00
Microsoft Office 2007	€ 130,00
MATLAB License	€ 6.000,00

Characterizaqtion Equipment	**Investment**	**% Use**	**Dedication**	**Depreciation period**	**Cost (€)**
UC3M test bench	€ 100.000,00	100%	6	60	€ 10.000,00
LabVIEW License	€ 1.580,00	100%	6	60	€ 158,00
TECHNOSOFT LabVIEW library License	€ 914,00	100%	6	60	€ 91,40

TOTAL: **€ 23.509,40**

Table 11 - Budget: material costs.

INDIRECT COSTS

Table 12 represents the total cost of the indirect costs, which represent the 20% of the material and the human costs.

Indirect costs			
20% of material and human cost	TOTAL:	**€**	**7.401,88**

Table 12 - Budget: indirect costs.

SUMMARY OF COSTS

The summary of the estimated total cost is represented in Table 13- It is included the personnel, the materials, the indirect costs and the 21% of IVA of the materials and indirect costs. The estimated cost for this project is of € 50,902.65.

SUMMARY OF COSTS		
Description	**Budget on total costs (€)**	
Personnel	€	13.500,00
Materials	€	23.509,40
Indirect costs	€	7.401,88
Total without IVA	€	44.411,28
IVA (21%) (for materials and indirect costs)	€	6.491,37
Estimated total	**€**	**50.902,65**

Table 13 - Budget: summary of costs.

Annex A – X-Ray Source

SPECIFICATIONS

GENERAL

Parameter	Description / Value	Unit
Tube Voltage Operational Range	40 to 110	kV
Tube Current Operational Range ①	10 to 800 (50 W Max.)	µA
Maximum Output	50	W
X-ray Focal Spot Size	15 to 80	µm
X-ray Beam Angle (Max.)	62	°
Focus to Object Distance (FOD)	16.8	mm
Operation	Continuous	—
Conformance Standards	CE (EMC: IEC 61326-01, Group1, Class A)	—

X-RAY TUBE UNIT

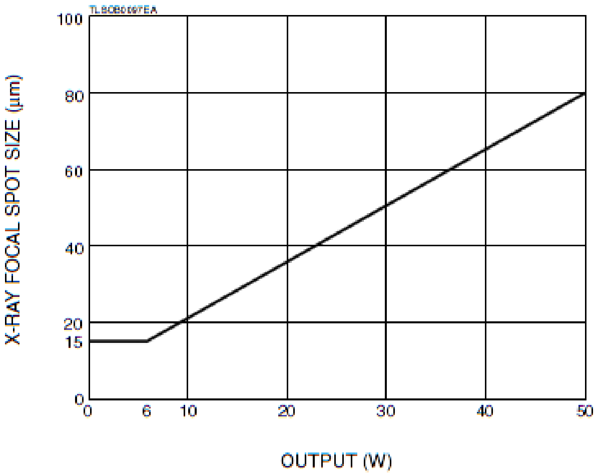
Parameter	Description / Value	Unit
X-ray Tube	Sealed Type	—
X-ray Tube Cooling Method	Convection Cooling	—
X-ray Tube Window Material / Thickness	Beryllium / 200	µm
Target Material	Tungsten	—
High Voltage Power Supply	Built-in	—
Operating Ambient Temperature	+10 to +40	°C
Storage Temperature	0 to +50	°C
Operating and Storage Humidity	Below 85 (No Condensation)	%
Weight	8.2	kg

X-RAY CONTROL UNIT

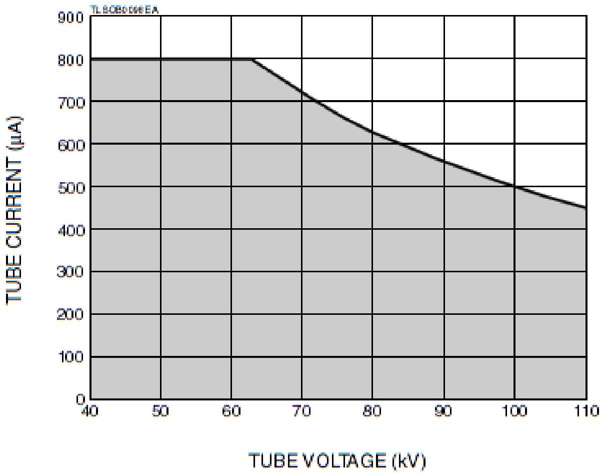
Parameter	Description	Unit
Function	Tube Voltage and Tube Current Preset / Auto Warm-up	—
Protection	Interlock	—
External Control	RS-232C	—
Operating Ambient Temperature	+10 to +40	°C
Storage Temperature	0 to +50	°C
Operating and Storage Humidity	Below 85 (No Condensation)	%
Power Consumption (Max.)	120	W
Input Voltage (DC)	24	V
Applicable OS	Windows® 2000 Professional, XP Professional	—
Computer Operating Conditions	CPU: Intel Pentium or Higher, Memory: 64 MB or More	—
Weight	3.3	kg

NOTE: ①See the graph of the tube current operational range.

X-RAY FOCAL SPOT SIZE vs. OUTPUT



TUBE CURRENT OPERATIONAL RANGE



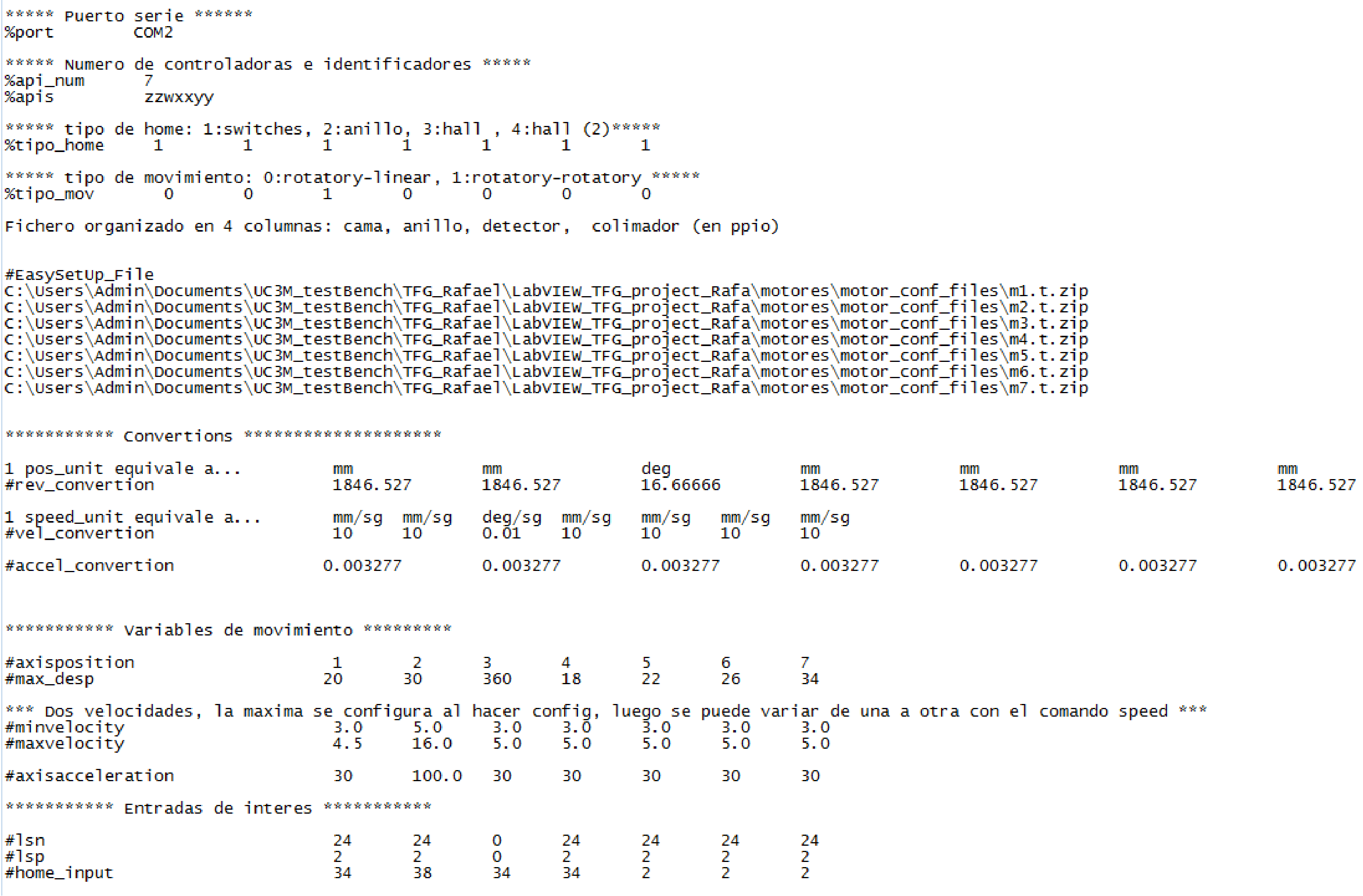
Annex B – Detector

DEXELA 1207, 1512, 2307, 2315 and 2923

	1207		1512		2307	2315	2923	
Sensor								
Pixel Size (µm)	74.8		74.8		74.8	74.8	74.8	
Sensitive Area (mm^2)	114.9 × 64.6		145.4 × 114.9		229.8 × 64.6	229.8 × 145.4	290.8 × 229.8	
Pixel Matrix (px)	1536 × 864		1944 × 1536		3072 × 864	3072 × 1944	3888 × 3072	
Max Frame Rate (fps)								
Pixel Binning	Camera Link	GigE Vision	Camera link	GigE Vision	Camera Link	Camera Link	Camera Link	GigE Vision
1×1	60	32	26	14	60	26	26	3.6
2×2	156	127	70	56	156	70	70	NA
4×4	191	191	86	86	191	86	86	NA
Image Performance								
Dynamic Range (dB)								
High Dynamic Range Mode	typ. 66		typ. 66		typ. 66	typ. 66	typ. 66	
High Sensitivity Mode	typ. 65		typ. 65		typ. 65	typ. 65	typ. 65	
X-ray Energy Range (kV)	12–130		12–130		12–130	12–130	12–130	
Mechanical								
Weight (kg)	1.9		2.2		2.5	3.7	6.1 (CL)/6.3 (GigE)	
Dimensions l × w × h (mm^3)	223.5 × 150 × 42		223.5 × 150 × 42		278.7 × 205 × 43	267 × 257 × 43	352 × 272.5 × 43	
Communications								
Camera Link	Base 80 MHz (1 cable)		Base 80 MHz (1 cable)		Medium 80 MHz (2 cables)	Medium 80 MHz (2 cables)	Full 80 MHz (2 cables)	
GigE Vision	1000BaseT		1000BaseT		NA	NA	1000BaseT	
Control Channel	115 kBaud serial link Camera Link / GigE Vision		115 kBaud serial link Camera Link / GigE Vision		115 kBaud serial link Camera Link	115 kBaud serial link Camera Link	115 kBaud serial link Camera Link / GigEVision	
Sync Ports	BNC		BNC		BNC	BNC	BNC(CL)/SMB(GigE)	
Sync In Port	3 – 15 V edge or level trigger		3 – 15 V edge or level trigger		3 – 15 V edge or level trigger	3 – 15 V edge or level trigger	3 – 15 V edge or level trigger	
Sync Out Port 1	TTL (0 – 5 V)		TTL (0 – 5 V)		TTL (0 – 3.3 V)	TTL (0 – 3.3 V)	TTL (0 – 3.3 V-CL/5 V-GigE)	
Sync Out Port 2	NA		NA		TTL (0 – 3.3 V)	TTL (0 – 3.3 V)	TTL (0 – 3.3 V -CL/5 V-GigE)	
Software	Support for 32 and 64 bit Windows® OS							
Power								
	Camera Link/GigE		Camera Link/Gig E		Camera Link	Camera Link	Camera Link/GigE	
Dissipation	11 / 13 W		9.6 / 13 W		17 W	17 W	36 / 37 W	
Scintillator Options								
	High Resolution CsI							
	High Efficiency CsI							
	Various Gd_2O_2S:Tb (GOS) fluorescent screens							
Temperature Range								
Operating Temperature	+10 °C to +40 °C							
Storage Temperature	-10 °C to +50 °C							
Accessories								
Power Supply	Camera Link/GigE 50-60 Hz 110 V – 240 VAC		Camera Link/GigE 50-60 Hz 110 V – 240 VAC		Camera Link 50-60 Hz 110 V – 240 VAC	Camera Link 50-60 Hz 110 V – 240 VAC	Camera Link/GigE 50-60 Hz 110 V – 240 VAC	
Power Cable	3m Lemo Low Voltage		3m Lemo Low Voltage		D-sub Power	D-sub Power	3 m Lemo D-sub Power Low Voltage	
Interface Card	EPIX EB1	Intel PRO1000/PT	EPIX EB1	Intel PRO1000/PT	EPIX E4	EPIX E4	EPIX E4	Intel PRO1000/PT

Annex C – Motor configuration file

8



```
***** Puerto serie ******
%port         COM2

***** Numero de controladoras e identificadores *****
%api_num       7
%apis          zzwxxyy

***** tipo de home: 1:switches, 2:anillo, 3:hall , 4:hall (2)*****
%tipo_home     1        1       1       1       1       1       1

***** tipo de movimiento: 0:rotatory-linear, 1:rotatory-rotatory *****
%tipo_mov       0       0       1       0       0       0       0

Fichero organizado en 4 columnas: cama, anillo, detector,  colimador (en ppio)


#EasySetUp_File
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m1.t.zip
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m2.t.zip
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m3.t.zip
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m4.t.zip
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m5.t.zip
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m6.t.zip
C:\Users\Admin\Documents\UC3M_testBench\TFG_Rafael\LabVIEW_TFG_project_Rafa\motores\motor_conf_files\m7.t.zip


*********** Convertions ********************

1 pos_unit equivale a...        mm              mm              deg             mm              mm              mm              mm
#rev_convertion                 1846.527        1846.527        16.66666        1846.527        1846.527        1846.527        1846.527

1 speed_unit equivale a...      mm/sg   mm/sg   deg/sg  mm/sg   mm/sg   mm/sg   mm/sg
#vel_convertion                 10      10      0.01    10      10      10      10

#accel_convertion              0.003277        0.003277        0.003277        0.003277        0.003277        0.003277        0.003277


*********** Variables de movimiento *********

#axisposition                  1       2       3       4       5       6       7
#max_desp                      20      30      360     18      22      26      34

*** Dos velocidades, la maxima se configura al hacer config, luego se puede variar de una a otra con el comando speed ***
#minvelocity                    3.0     5.0     3.0     3.0     3.0     3.0     3.0
#maxvelocity                    4.5     16.0    5.0     5.0     5.0     5.0     5.0

#axisacceleration               30      100.0   30      30      30      30      30

*********** Entradas de interes ***********

#lsn                            24      24      0       24      24      24      24
#lsp                            2       2       0       2       2       2       2
#home_input                     34      38      34      34      2       2       2
```

References

[1] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics, 2001.

[2] C. N. Ionita, K. R. Hoffmann, D. R. Bednarek, R. Chityala, and S. Rudin, “Cone-beam micro-CT system based on LabVIEW software.,” *J. Digit. Imaging*, vol. 21, no. 3, pp. 296–305, Sep. 2008.

[3] S. Wang, H. Han, K. Gao, Z. Wang, C. Zhang, M. Yang, Z. Wu, and A. Marcelli, “A LabVIEW based user-friendly X-ray phase-contrast imaging system software platform,” *J. X-Ray Sci. Technolgy*, pp. 1–11, 2014.

[4] *D. B. Guralnik, Ed., Webster’s New World Dictionary of the American Language, 2nd college edition, William Collins + World Publishing Co., Cleveland (1974)*. .

[5] “Röntgen CW (1895) Uber eine neue Art von Strahlen. Vorläufige Mitteilung.”

[6] “A. Bocage, ‘Procede et dispositifs de radiographie sur plaque en mouvement,’ France Patent, vol. 536464, 1921.”

[7] “G. Frank, ‘Verfahren zur Herstellung von Körperschnittbildern mittels Röntgenstrahlen,’ Germany Patent, vol. 693374, 1940.”

[8] “Cormack AM (1973) Reconstruction of densities from their projections, with applications in radiological physics. Phys Med Biol 18(2):195 207.”

[9] R. Cierniak, *X-Ray computed tomography in biomedical engineering*. Springer Science & Business Media, 2011.

[10] J. Beutel, H. L. Kundel, and R. L. Van Metter, “Handbook of Medical Imaging, volume 1: Physics and Psychophysics,” 2000.

[11] J. Hsieh, “Computed tomography: principles, design, artifacts, and recent advances,” 2009.

[12] M. Korner, C. H. Weber, S. Wirth, K.-J. Pfeifer, M. F. Reiser, and M. Treitl, “Advances in Digital Radiography: Physical Principles and System Overview 1,” *Radiographics*, vol. 27, no. 3, pp. 675–686, 2007.

[13] J. H. Siewerdsen and D. A. Jaffray, “Cone-beam computed tomography with a flat-panel imager: magnitude and effects of x-ray scatter,” *Med. Phys.*, vol. 28, no. 2, pp. 220–231, 2001.

[14] P. Suetens, *Fundamentals of medical imaging*. Cambridge university press, 2009.

[15] W. C. Scarfe, A. G. Farman, and P. Sukovic, “Clinical applications of cone-beam computed tomography in dental practice,” *Journal-Canadian Dent. Assoc.*, vol. 72, no. 1, p. 75, 2006.

[16] M. Abella, J. J. Vaquero, A. Sisniega, J. Pascau, A. Udías, V. García, I. Vidal, and M. Desco, “Software architecture for multi-bed FDK-based reconstruction in X-ray CT scanners,” *Comput. Methods Programs Biomed.*, vol. 107, no. 2, pp. 218–232, 2012.

[17] L. A. Feldkamp, L. C. Davis, and J. W. Kress, “Practical cone-beam algorithm,” *JOSA A*, vol. 1, no. 6, pp. 612–619, 1984.