# IPBMA. Exercice 7.

## Building a basic CT system. Sinogram detection.

Built an industrial CT scanner using Python functions. Unlike the CT scanner for medical purposes, in the industrial CT scanner, the X-ray tube and detector remain fixed and is the object that rotates. There are two basic models. One is based on fan-beam X-rays and linear detectors, where the X-ray tube and detector move up and down to acquire the sinograms of the different slices needed to cover the entire object. The second is based on cone-beam X-rays and full-field detectors. Moving the source and detector up and down in this second case is not needed. This exercise aims to simulate a CT scanner based on parallel-beam X-rays and linear detectors (see Figure 1) and obtain a sinogram from the cubic phantom using such a device.

The Python functions to be implemented will be called from the main program and will be: ***source()*** (already implemented)***, cube_phantom_h()*** (already implemented)***, interactor_CT(), detectSinogram(), detector_1D(), and process_CT*()** and will include the following parameters:

***interactor_CT(N0, Object, zPos, nProjections).-*** function that simulates the interaction between the x-ray beam and the object. This interaction depends on the linear attenuation coefficient and the object's shape. Both quantities are included in the object itself. Also, for CT, the interaction depends on the height at which we shoot (zPos) and the number of angles we use (nProjections) for those shots. These angles are assumed to be equispaced for a total distance of 360 degrees.

i) N0 → Number of photons generated per unit area.
ii) Object → Subject to X-ray.
iii) zPos → The height of the object where the slice is acquired.
iv) nProjections → The number of angels (projections) used at the acquisition.

***Output***→ Numpy array (2D), whose values represent the sinogram of the CT in terms of quantum image.

***detectorSinogram(qImage, nProjections, nDetectors).-*** function that simulates the process of detecting the sinogram (as a quantum image) using a linear detector. Inside this function, you have to call the function detector_1D(), which you must also implement.

i) Image → Quantum image captured.
ii) nProjections → number of angles (projections) used at the acquisition.
iii) nDetectors → number of detector cells of the linear detector used.

***Output***→ Numpy array (2D), whose values represent the pixel values of the sinogram acquired.

***detector_1D(qImage, angle, nDetectors).-*** function that simulates the 1D detector used in the acquisition process. Suppose the detector is ideal. No noise considerations are made, and each cell's size equals the size of each pixel of the quantum image that represents the sinogram. Thus, nDetectors take control of the FOV of the system.

i) Image → Quantum image captured.
ii) angle → Acquired angle. In terms of quantum imaging, the row's height to be detected.
iii) nDetectors → Number of detector cells of the linear detector used.

***Output***→ Numpy array (1D), whose values represent the pixel values of the sinogram detected.

***process_CT(image, n0)***.- which represents the image transformation that the detector electronics make to compensate for the exponential distribution of the components of the captured quantum image. Therefore, this function calculates the *neperian* logarithm of each element of the incident radiation. Use N0 to normalize the number of detected photons, represented by the sinogram.

i) Image → detected image of the sinogram.
ii) N0 → Number of photons generated by the source per unit area.

***Output***→ Numpy array (2D), whose values represent the pixel values of the sinogram, compensated.

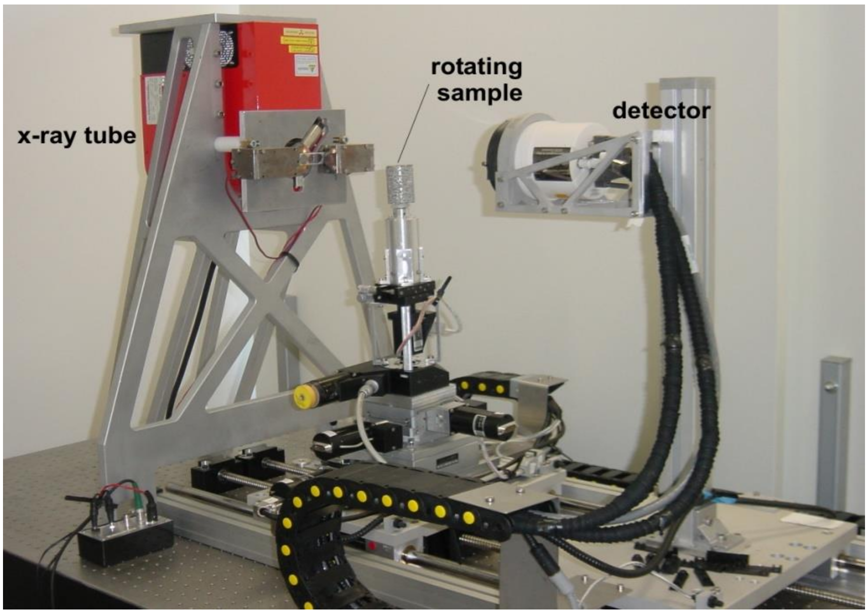


**Figure 1.** Industrial CT scanner based on fan-beam x-rays and linear detectors. The X-ray tube and detector must move up and down to cover the entire object.

## <u>Advises</u>

To implement this CT:

- As always, the fan-beam X-ray will be substituted by a parallel beam X-ray.
- Implement first the quantum image, but in this case, rotate the object before each acquisition.
- Suppose the size of the quantum image is double the size of the object, and the object is located in the middle of the parallel beam and surrounding of air. Suppose also that the linear attenuation coefficient of the air, in this case, is 0.
- Implement the detector as usual but consider now that the detector has just a row of cells.

**Note**.- each student has to bring a zip file called *lastName_Name_P7.zip*, to the following address: *pablogtahoces@gmail.com*. The subject of the e-mail should be: IPBMA_P7. Inside the zip should be included:

- A jupyter notebook showing how the software works (see the example).
- An html file of the notebook.
- All the necessary files to verify the correct operation of the application.
- The deadline will be: Wednesday, November 13, 10:00.