INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Carlos Alexandre Novak Madureira

DEVELOPMENT OF A TOOL TO SUPPORT THE RESEARCH ON NOVEL ACQUISITION MODES OF SYNTHETIC APERTURE RADAR (SAR)

Final Paper
2020

Course of Aeronautical Engineering

CDU 621.396.96:681.3

Carlos Alexandre Novak Madureira

DEVELOPMENT OF A TOOL TO SUPPORT THE RESEARCH ON NOVEL ACQUISITION MODES OF SYNTHETIC APERTURE RADAR (SAR)

Advisor

Prof. Dr. Renato Machado (ITA)

Co-advisor

Dr. Michelangelo Villano (DLR)

AERONAUTICAL ENGINEERING

Gilching
Instituto Tecnológico de Aeronáutica

2020

Cataloging-in Publication Data
Documentation and Information Division

Novak Madureira, Carlos Alexandre
Development of a tool to support the research on novel acquisition modes of Synthetic Aperture Radar (SAR) / Carlos Alexandre Novak Madureira.
Gilching, 2020.
62f.

Final paper (Undergraduation study) – Course of Aeronautical Engineering– Instituto Tecnológico de Aeronáutica, 2020. Advisor: Prof. Dr. Renato Machado. Co-advisor: Dr. Michelangelo Villano.

I. Instituto Tecnológico de Aeronáutica. II. Title.

BIBLIOGRAPHIC REFERENCE

NOVAK MADUREIRA, Carlos Alexandre. **Development of a tool to support the research on novel acquisition modes of Synthetic Aperture Radar (SAR)**. 2020. 62f. Final paper (Undergraduation study) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

AUTHOR'S NAME: Carlos Alexandre Novak Madureira
PUBLICATION TITLE: Development of a tool to support the research on novel acquisition modes of Synthetic Aperture Radar (SAR).
PUBLICATION KIND/YEAR: Final paper (Undergraduation study) / 2020

Carlos Alexandre Novak Madureira
Rua H8B, 204
12.228-461 – São José dos Campos–SP

DEVELOPMENT OF A TOOL TO SUPPORT THE RESEARCH ON NOVEL ACQUISITION MODES OF SYNTHETIC APERTURE RADAR (SAR)

This publication was accepted like Final Work of Undergraduation Study

Carlos Alexandre Novak Madureira

Author

Renato Machado (ITA)

Advisor

Michelangelo Villano (DLR)

Co-advisor

Prof. Dr. Maurício Andrés Varela Morales

Course Coordinator of Aeronautical Engineering

Gilching: September 25th, 2020.

Acknowledgments

Agradeço ao meu pai, mãe e madrasta pelo apoio incondicional para que eu pudesse realizar meu primeiro grande sonho que foi ter estudado nesta instituição. Também pela educação e pelos ensinamentos que me deram para a formação do meu caráter. Obrigado também à minha companheira pelo apoio, amor e compreensão durante todos esses anos de graduação.

Aos meus amigos pelo apoio e companheirismo antes e durante a graduação. Sem essas pessoas com certeza a vida teria sido muito menos divertida e imprevisível. Agradeço também a eles pelos debates e pelo convívio. Sem dúvida, hoje trago comigo muito do que aprendi com vocês.

Ao professor Renato Machado pela orientação e auxílio na condução deste trabalho. À Nertjana Ustalli, Luca Delamore e Michellangelo Villano por me ensinar tanto e me desafiar mais ainda neste trabalho. Ao DLR e ao Alberto Moreira pela oportunidade de ter feito parte de um centro de pesquisa tão renomado e importante para o desenvolvimento da tecnologia.

Ao ITA pelo ensino de qualidade e gratuito. Por tantas oportunidades de conhecer novos países e de ampliar minha bagagem cultural. Por me desafiar a ser um profissional mais resiliente e qualificado. Aos meus professores de todas as fases da minha educação. Agradeço por me despertarem o gosto por exatas e olimpíadas científicas. Por me ensinarem a pensar de formas criativas sobre os problemas e a estudar pela paixão em aprender mais sobre esse incrível universo.

A todos com quem pude aprender a ser profissional melhor. Aos colegas e amigos do banking, pelos quais nutro grande admiração e respeito. Foi um privilégio poder fazer parte de tantas etapas desse projeto e ter colega e algumas vezes também liderado por pessoas tão incríveis.

Por último, agradeço à Deus pela vida e por tantas oportunidades de me tornar um ser melhor.

"The most important thing, the thing that unites all of us, is that we can inspire and challenge one another to be better."

— Kobe Bryant

Resumo

Neste trabalho foi desenvolvida uma ferramenta para estudo de modos inovadores de aquisição em radares de abertura sintética. O objetivo foi implementar os códigos propostos pelo grupo de pesquisa NewSpaceSAR de maneira a se ter uma ferramenta em Python cujo uso permite a simulação do sinal usual e do sinal com ambiguidade na direção de range, para quatro modos de aquisição inovadores para radar de abertura sintética e para o modo convencional de operação. Foi feita também a otimização de cada uma das rotinas em termos do uso de memória para que se mitigue o problema do uso excessivo de RAM. O código final foi desenvolvido com uma estrutura modular e seguindo os princípios de boas práticas de programação. Foram também desenvolvidas rotinas para que a análise dos resultados das simulações seja facilitada.

Abstract

In this work a tool was developed to study innovative modes of acquisition in synthetic aperture radars. The objective was to implement the codes proposed by the research group NewSpaceSAR to have a tool in Python whose use allows the simulation of the usual signal and the signal with ambiguity in the range direction, for four innovative acquisition modes for synthetic aperture radar and the conventional mode of operation. The optimization of each of the routines in terms of memory use has also been done to mitigate the problem of excessive use of RAM. The final code was developed with a modular structure and following the principles of good programming practices. Routines were also developed so that the analysis of the simulation results is facilitated.

Contents

1 Introduction

1.1 Motivation

Although the use of synthetic aperture radars has brought a significant advance in data collection from Earth to support scientific research, it is also true that the technology still has open problems. Among these is the lack of data for climate and geological research. In this sense, it is necessary to make an effort to have more technologies operating in the future to produce this data. In line with this demand, the synthetic aperture radar has proved to be a fundamental system for ground monitoring with a high degree of accuracy and independence from climatic conditions, being therefore an important technology in remote sensing.

However, in order to have a high level of precision with this technology, one must use hardware whose cost is quite high, which makes it impossible to do these missions more often. The research group in NewSpace SAR aims to develop innovative ways for the operation of these radars in order to reduce costs and increase the performance of this synthetic aperture radar technology.

This work was developed during my experience as an intern in the NewSpace SAR group at the German Aerospace Center (DLR). My contribution to the group was the development of a tool to support research on new acquisition modes for synthetic aperture radars (SAR). The research of these new acquisition modes aims to improve the technology of this type of radar, through the variation in the waveform of the waves emitted by the sensor. This change implies an improvement in the level of ambiguity suppression in the image, since it forces a mismatch of the filter with the ambiguous signal. As a final consequence, there is an image where the ambiguities are smeared. This improvement has the impact of a greater ability to focus on the local level, and is therefore a solution with the objective of local optimization, but not global. The application of innovative acquisition modes for SAR, precisely due to this capacity of local improvement, has as another benefit the cheapening of this type of mission. Therefore, the development of a tool to support the research on these novel acquisition modes can accelerate the advances on searching for modes to developed system's cheaper and with higher performance.

1.2 General objectives of this work

This work aims to describe the rationale, steps, and results achieved in the development of a new SAR research tool for the study of innovative SAR acquisition modes. This document aims to serve as a record of the main aspects considered and also as a first workbook on how to use the code developed. It also seeks to clarify what the next steps are and to delimit what are the advances in terms of code efficiency due to the application of good programming practices.

1.3 Outline

Chapter 2 explains the main concepts for understanding the SAR context and the variables considered for the characterization and study of images produced in this research field. The signal characteristics relevant to understanding the innovation proposed by the NewSpace SAR research group are explained. In Chapter 3, the innovations proposed by the group and their characteristics are detailed. Chapter 4 describes the methodology used in the development of the tool, delimiting what was the rationale for deriving the requirements and the process of solving the problems linked to each of these requirements. It is also explained the main concepts that guided the development of a tool with the level of availability and maintainability required for this application. Chapter 5 finally explains the result developed, clarifying each of the regions of this project and how the separation of contexts was made. In Chapter 6 the results of the execution times, for the single target simulation use case, for each of the required data acquisition and processing modes are explained and its efficiency discussed. In Chapter 7 the results of the Real SAR Scene use case are described for each type of signal and acquisition mode, then the performance enhancement achieved is discussed. Finally, in Chapter 8 the main conclusions and the next steps suggested for the further development of this tool are outlined.

2 Introduction to the Synthetic Aperture Radar

SAR is a particular type of radar. This Chapter explains the concept of radar and its application. Then the SAR type will be characterized and the main concepts required to understand how this technology works will be explained. Among these aspects, there is the way the signals are emitted by this equipment and the characteristics of the signals in the fundamental directions of the SAR image. After that, the algorithms for the treatment of SAR signals in terms of their fundamental steps will be exposed.

2.1 The RADAR technology

The radar was initially developed for military purposes during World War II. Its initial proposal was to identify aircraft and ships during conditions of darkness or adverse weather. Advances in radio frequency studies brought improvements in digital technology and antennas to this area. The radar systems initially lent themselves to measuring the distance to the target by measuring the time difference between the emission and the perceived echo. Today, radars are used to measure the distance, angle, and speed of targets such as ships, aircraft, missiles, and even weather formations such as storms and clouds (Cumming; Wong, 2002).

2.2 Brief contextualization about SAR

Synthetic aperture radar technology has been widely used to generate images of the Earth. This remote sensing mode has its creation credited to the mathematician Carl A. Wiley, in 1951. The invention of the then employee of Goodyear Aircraft Company operates mainly through satellites and is today present in more than 10 space missions. These satellites produce daily images that allow us to support research in climatology, geology, and geography. SAR stands out in the area of remote sensing because it allows ground sensing regardless of weather and lighting conditions in the area under study.

Added to this is the advantage of the resolution of this type of radar being independent on the distance of the equipment to the study point, which allows us to have a constant resolution along the same line of sight. One last point of emphasis and which is still related to this same performance parameter is that the SAR has an azimuth resolution that depends on the antenna length, which allows gains in the order of dozens in relation to the resolution that can be achieved in a traditional radar situation. The resolution in range depends on the bandwidth of the transmitted pulse.

2.3 SAR fundamentals

In this section, it will be detailed better how the characterization of the SAR scene geometry is done. The main aspects of the SAR signal will be explained for the case of a single target being illuminated by the radar beam. The backscattered signal will be characterized in both range and azimuth directions. Then the algorithm applied to simulate the raw data for a real SAR scene will be shown. Finally, the steps of the data processing algorithms necessary to build the focused image through this technology will be exposed.

2.3.1 SAR Geometry characterization

SAR is a type of radar that works onboard a vehicle, such as an aircraft or a satellite. The acquisition mode of this equipment is in a lateral view to avoid ambiguities in the captured signal. This type of system has several variations in its mode of operation and type of acquisition, but in this study, we will focus on the explanation of a conventional SAR system embarked on a satellite and operating in the stripmap mode.

In this system, there is a satellite traveling at a speed of V_s at a height of the Earth's surface of H. Simplifying the geometry of this system, we have that the direction of motion of the satellite is called azimuth direction and the direction orthogonal to it is called range direction. These directions constitute an inclined plane in relation to the Earth plane. The distance from a target on the ground to the system is called slant distance and is represented by R. The smallest distance from this same target to the azimuth direction is represented by R_0. In the scene illumination process, the platform then moves along the azimuth direction and illuminates a target with a conical emission beam whose azimuth extension is called the azimuth opening and in the range direction is the swath width.

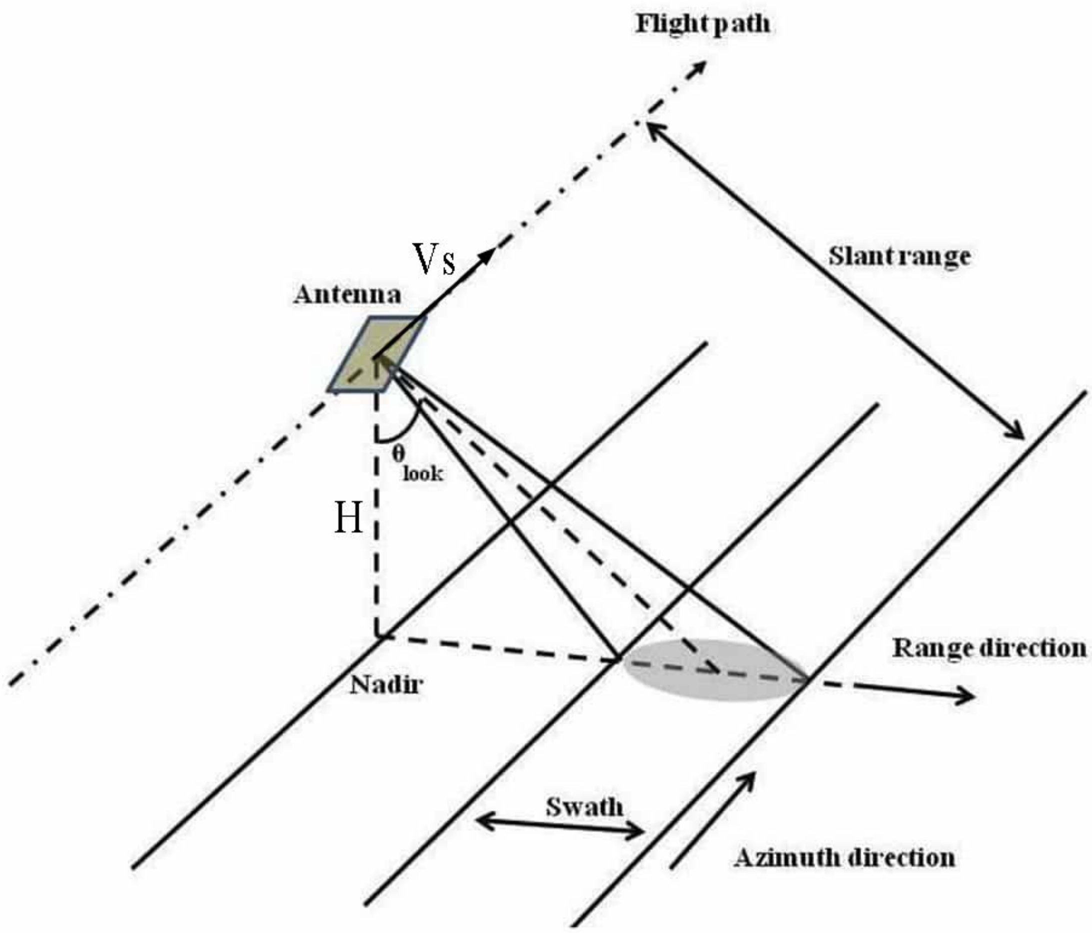


FIGURE 2.1 – Basic scheme of the geometry of SAR observation. In this figure, H stands for the satellite altitude related to the ground and V_s is the platform's speed

The range and azimuth directions form the basis of the coordinate system of a SAR image in two dimensions. The temporal coordinates of these axes are named specifically, the time in the range direction being called *fast time* and represented by the letter τ. The time coordinate on the azimuth axis is called *slow time* and is represented by the letter *eta*.

2.3.2 Transmitted Pulses characterization

In the conventional SAR acquisition mode, the waveform of a standard radar pulse wave is a chirp wave. This wave is a frequency modulated signal that is ruled by two variables, the carrier frequency (f_0) and the chirp rate (K_r) The main characteristic about this wave is that its frequency has a linear variation with the time, the general format of the frequency of this wave is:

$$f(t) = f_0 + K_r.t \tag{2.1}$$

where the K_r is the chirp rate and gives the rate of change of the chirp frequency with the time.

To derive the shape of the chirp wave, one starts from the general equation of a one-dimensional wave as a function of time:

$$s(t) = A_0 cos(\omega.t) \tag{2.2}$$

Where A_0 is the amplitude of the wave and ω is its angular frequency.

Replacing the 2.1 in 2.2 we have:

$$s(t) = A_0 cos(2\pi(f_0.t + K_r.t^2)) \tag{2.3}$$

The last point to consider when characterizing the signal emitted by SAR is that this transmission does not take place continuously and interrupts, but periodically. The periodic signal transmitted consists of a chirp wave whose duration is equal to T units of time. After this time, the transmission ceases for T_i time units and this set constitutes the transmission pattern of a SAR radar in conventional acquisition operation mode. The figure 2.2 illustrates this process.

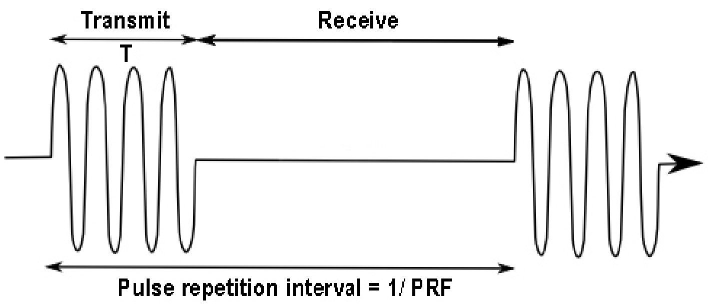


FIGURE 2.2 – Schematic to illustrate what a conventional SAR acquisition system issuing pattern looks like. The pulses are periodically emitted with a PRI interval between them, while the chirp wave has a total duration of T. The time in which there is no emission is used to make the acquisition of the signal reflected by the illuminated scene.

For the chirp wave to be emitted in this rectangular shape, its analytical form must be multiplied by a function that has this effect of limiting the emission range. This function is a pulse envelope and is given by:

$$\omega_r = rect(\frac{\tau}{T}) \tag{2.4}$$

where $rect(\frac{x}{T})$ denote the rectangular function with duration T.

However, as mentioned, the SAR image depends not only on the emission of waves but also on the backscattering acquisition. To this process of emission and acquisition, the SAR usually has a fixed time interval between its pulses and the radar wave collection. This acquisition step happens between the emission of the consecutive pulses and is illustrated 2.2.

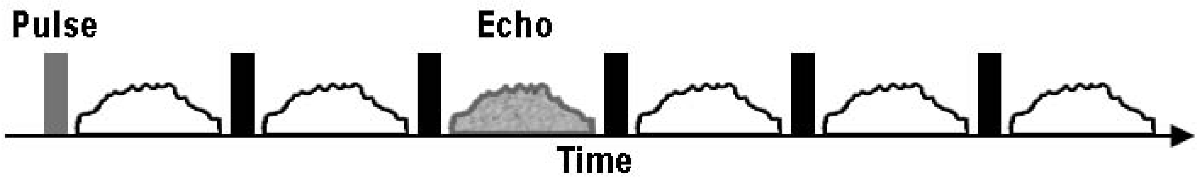


FIGURE 2.3 – Schematic illustration of the characterization of the emitted and received signal in time for a conventional SAR operation mode. The figure shows the magnitude of the signal as a function of time during the pulse emission and echo receiving process

The wave transmitted has an echo that depends on variables such as the terrain geometry, the distance to the radar, and so on. Thus the echoes acquired by the antenna may have any shape, like the ones represented in figure 2.3.

The interval between two pulses is called Pulse Repetition Interval (PRI) and is an important configuration parameter on SAR. With respect to the signal processing constraints, its inverse value, the Pulse Repetition Frequency (PRF) is also important to define before the acquisition is made.

They are related by:

$$PRI = \frac{1}{PRF} \tag{2.5}$$

2.3.3 Backscattered signal characterization

Once the format of the signal emitted by the radar is characterized, it is now necessary to understand how the echo of the emitted signals is received and the main aspects of this process.

According to (Cumming; Wong, 2002), let's first characterize the signal in the range direction and then in the azimuth direction.

As discussed in the previous section, the signal reflected by the terrain influences the shape of the wave captured by the radar. Considering this effect, the final equation of the reflected signal in the range direction is given by:

$$s_0(\tau) = A_0'.\omega_r(\tau - \frac{2R_0}{c}).cos(2\pi f_0(\tau - \frac{2R_0}{c} + \pi K_r(\tau - \frac{2R_0}{c}^2)) \quad (2.6)$$

where c is the speed of light, R_0 is the minimum slant range for this point target, and A_0' is an arbitrary complex constant.

To account now the azimuth dependence in the characterization of the received signal, as explained in (Cumming; Wong, 2002), the modulation effects of the signal in this direction due to the Doppler effect must be considered, as well as the effect of the acquisition pattern coming from the radiation diagram of the antenna used in the satellite. Once these two factors are accounted for, the equation that characterizes the signal received by the system as a function of the range and azimuth time coordinates, after demodulation, is:

$$s_0(\tau, \eta) = A_0.\omega_r(\tau - \frac{2R(\eta)}{c}).\omega_a(\eta - \eta_c).exp(-j4\pi f_0\frac{R(\eta)}{c}).exp(j\pi K_r(\tau - \frac{2R(\eta)}{c})^2) \quad (2.7)$$

In which ω_a is the term corresponding to the satellite antenna acquisition pattern. As exposed in (Cumming; Wong, 2002) for a rectangular-shape antenna this term corresponds to a sinc-like-squared function.

$$\omega_a \sim sinc^2(\frac{0.886.\theta(\eta)}{\beta_{bw}}) \quad (2.8)$$

where β_{bw} is the azimuth beamwidth and $\theta(\eta)$ is the angle measured from boresight in the slant range plane and it is also a function of the slow time (η).

Furthermore, due to the movement of the satellite, the azimuth signal received by the antenna is also modulated. This modulation is originated due to the Doppler effect. The movement of the platform also causes a second important effect on the data structure which is the range cell migration (RCM). In this effect, the data corresponding to the same target, whose distance from the movement axis of the system is fixed, has a symmetrical arc pattern in relation to the zero doppler plane. This effect happens because the distance from the target to the satellite increases as the data is acquired and as a consequence, in order to process the data later, it will be necessary to correct this migration first so that the image can be properly focused.

Finally, still concerning to this platform movement, the $R(\eta)$ in the equation 2.7, represents the range of the target as a function of the slow time (η). To derive this term, it is used the hyperbolic form of the range equation, which is defined as:

$$R^2(\eta) = R_0^2 + (V_r.\eta)^2 \tag{2.9}$$

where V_r is the approximated radar velocity. This approximation is discussed in (Cumming; Wong, 2002) and it is calculated for the zero Doppler point according to the relation 2.10.

$$V_r \approx \sqrt{V_s.V_g} \tag{2.10}$$

where V_s and V_g are the instantaneous velocities of the satellite and the radar beam footprint respectively.

2.4 SAR image generation

SAR raw data generation is an important step for testing different SAR image formation algorithms as well as studying the interaction of electromagnetic waves with a scene. The SAR raw data can be generated by exploiting eq. (2.7) introduced in the previous section. This temporal approach is computationally very intensive, especially as the scene gets larger, and the number of points scatterers increases. An alternative is to make use of SAR image formation algorithms in their inverse forms. The advantages gained by this approach are computational time efficiency and its independence with respect to the number of scatterers in a given scene. In addition, we are also interested to simulate, starting from real SAR images (i.e., TerraSAR-X images), the SAR images received from novel SAR operation.

2.4.1 Raw data

The raw data matrix contains the data as a function of the range and azimuth time coordinates. This matrix is generated, in the case of point target simulation, using the equation 2.7. Another type of raw data simulation is made from a real SAR scene already focused. In this case, the raw data corresponding to that scene can be obtained using the inverse omega-k algorithm, which will be explained in more detail below.

2.4.1.1 Single target

It is common in the SAR study that the first exercise done to understand this technology is the simulation of the signal received by the system in a situation of emission against a point target. In this work, this study mode was also implemented, and according to the theory discussed, the echo signal received by the system as a function of time, as described in 2.7.

2.4.1.2 The Inverse $Omega - K$ algorithm

The inverse Omega-K algorithm is used to generate raw data from real SAR images. It produces the raw data based on a set of parameters provided by the User and the System's configuration, for instance: all the satellite's configuration and the required processing parameters.

The algorithm has an input a real Scene data image and produces the simulated raw data performing a set of steps sequentially. The following block diagram 2.4 shows the steps of the inverse $Omega - K$.

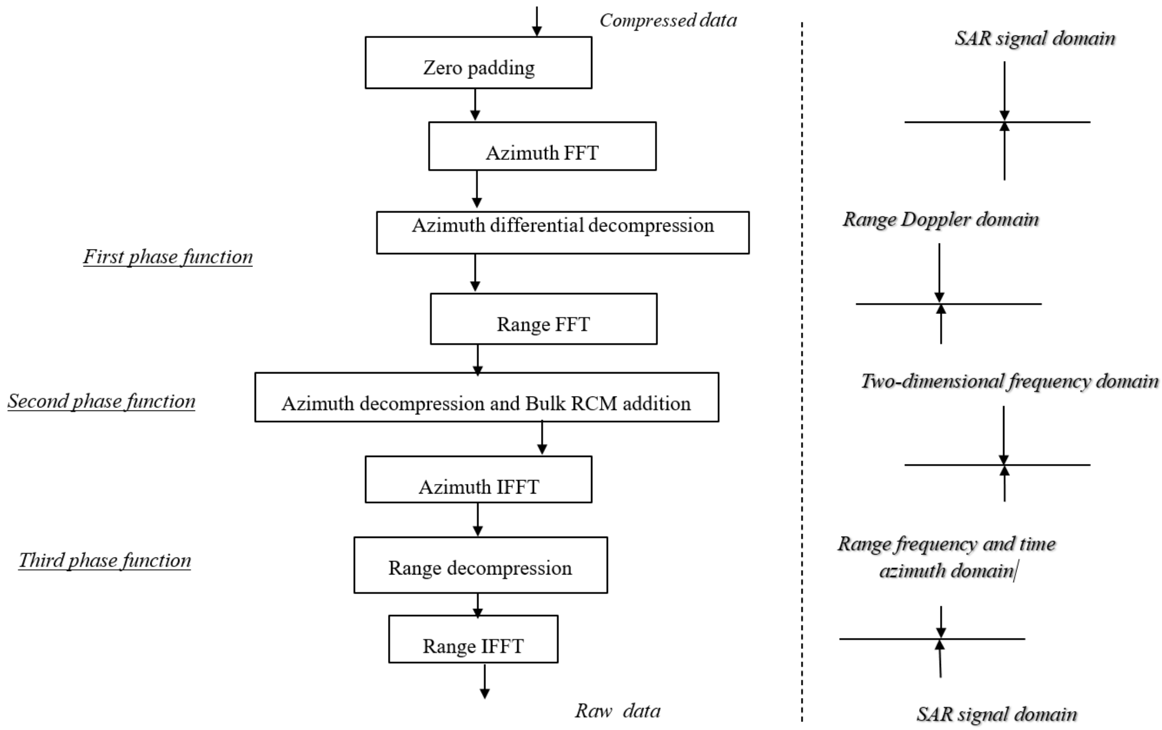


FIGURE 2.4 – Block diagram of the inverse $Omega - K$ algorithm.

This algorithm was used in the project for the simulation of raw data from a real scene image obtained with SAR. For the exploration of innovative acquisition modes, this algorithm was adapted to correspond to the acquisition mode in study.

2.4.2 Focused image

Once the echo of the signal emitted by the radar has been received, in order to build the image from it, signal processing techniques must be applied so that this signal can be transformed into data in which the scene that had been illuminated can be visualized.

To do this process of focusing the raw data, a matched filter is applied in the range and azimuth direction. This filter is applied using the known pulse response of the signal in each direction of the image. This is possible because the SAR system, is a linear system and, therefore, the knowledge of the impulse response of the pulses emitted allows the construction of the image by applying the superposition principle.

To do this focusing process, according to (Cumming; Wong, 2002), you can choose one of several known and already tested algorithms. The differentiation in the choice of code is due to the comparison in terms of precision, the difficulty of implementation, and its adequacy to the geometric conditions of the scene under study.

The most primitive algorithm for the processing of raw SAR data is Range Doppler Algorithm (RDA). This algorithm has as one of its advantages the ease of implementation. However, to have more quality in the focus, it is possible to use more elaborated algorithms, being that in this work it was opted to use the adapted omega-K algorithm.

2.4.2.1 The Omega-K algorithm

As described in (Cumming; Wong, 2002) the $omega - k$ algorithm for SAR raw data image focusing is better than the traditional RDA due to its capacity of data processing in conditions of wider azimuth apertures or high squint angles.

The Omega-K, in the accurate implementation, is built using the following steps:

1. A two-dimensional FFT is performed to transform the SAR signal data into the two-dimensional frequency domain
2. Reference function multiply
3. Stolt mapping along range frequency axis
4. two-dimensional IFFT

The approximate algorithm differentiates to the full omega-K on the aspect of the Stolt interpolation what is done in favor of a simpler phase multiply. Hence, the steps on the approximate case are:

1. A two-dimensional FFT then a multiplication by the reference function

2. A range IFFT to change the domain to the range Doppler domain
3. A matched filtering with the range dependency to remove the residual azimuth modulation
4. An IFFT to transform the compressed data back into the time domain in both directions

The block diagram 2.5 shows its main steps.

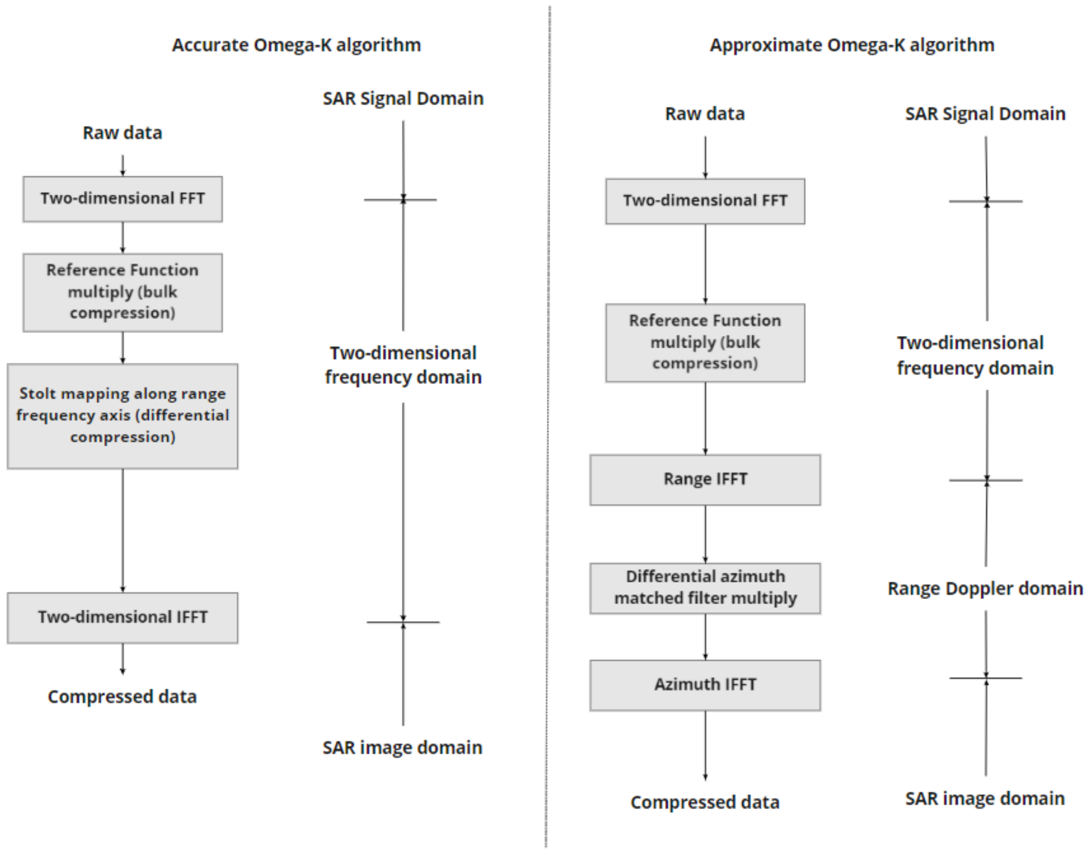


FIGURE 2.5 – Block diagram of the $Omega-K$ algorithm and of its approximate version. (Cumming; Wong, 2002)

In this work, the $omega - k$ algorithm was adapted so that it could fit all the novel SAR acquisition modes. This way, the final algorithm implemented was the represented in the diagram 2.6.

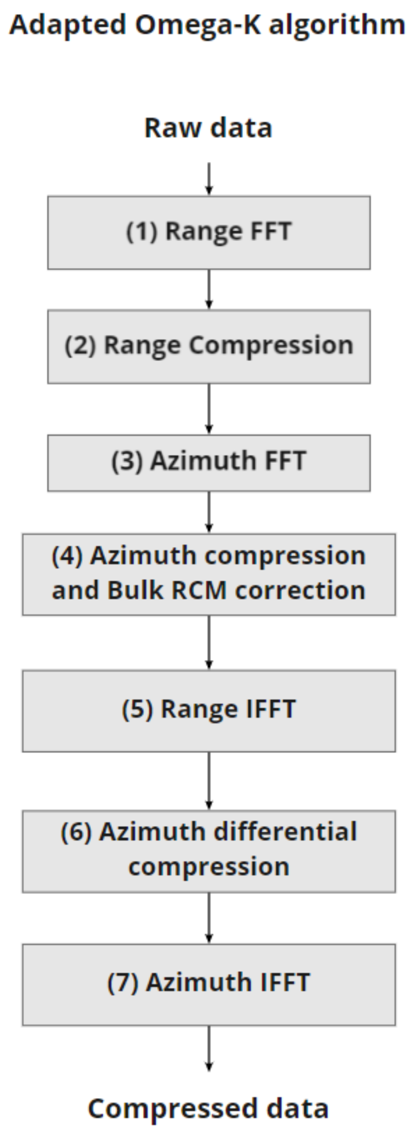


FIGURE 2.6 – Block diagram of the adapted $Omega - K$ algorithm used in this work. The numbers on the left of each block in this diagram refers to the step number of the implemented algorithm. It is important to notice that these are the same steps of the figure 2.4, but in the opposite order.

3 Novel SAR acquisition modes

Among the innovative acquisition modes developed in the NewSpace SAR group, those studied in this work were four: waveform encoded SAR with up-and-down alternation (1), waveform encoded SAR with cyclically-shifted chirps (2), Doppler-matched azimuth phase code (3) and the waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code (4). These modes were compared with the conventional acquisition mode in stripmap operation mode. The difference between the innovative modes is the variation in the waveforms of the emitted pulses. At each pulse, a change is imposed that allows suppression of the ambiguities in the image and a decrease in the ASR level at the end of the processing.

3.1 About the importance of the new SAR acquisition modes

As mentioned (Moreira *et al.*, 2013) the problem of the Earth's lack of data is often the reason why geological, geographic, and climatic research has its advance delayed. According to the same source, the number of searches in these areas that require accurate data and increasingly shorter time intervals has grown, as explained in 3.1 and one way to meet this demand is through the use of SAR.

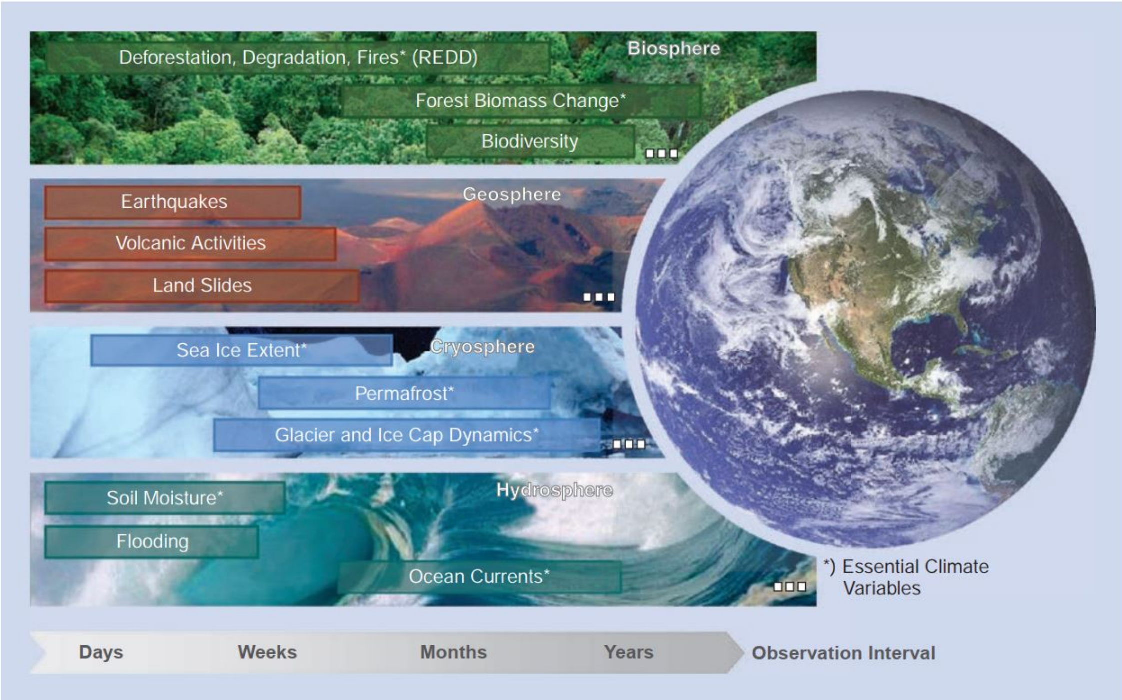


FIGURE 3.1 – Earth study areas and their physical data time intervals required to support research. Figure from (Villano *et al.*, 2020)

The use of synthetic aperture radars may be a key way in the future to prove climate and terrestrial data with the precision of the time interval required for the advancement of scientific research on Earth's physical phenomena. With this mission in mind, the NewSpace SAR group proposes to study variations in conventional SAR modes of operation to innovate in this field and provide greater resolution, accuracy, and optimization in the use of this technology.

3.2 Innovative SAR acquisition modes

Among the innovative acquisition modes developed in the NewSpace SAR group, those studied in this work were four: waveform encoded SAR with up-and-down alternation, waveform encoded SAR with cyclically-shifted chirps, Doppler-matched azimuth phase code and the waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code. These modes were compared with the conventional acquisition mode in stripmap operation mode. The difference between the innovative modes is the pulse to pulse variation in the waveform. As a consequence of this waveform change pattern, it is possible to suppress ambiguities after the image focusing.

The figure 3.2, taken from (Villano *et al.*, 2020) illustrates the quality of results that can be achieved by applying this technique. The image used for the simulation was a real image collected by TerraSAR-X.

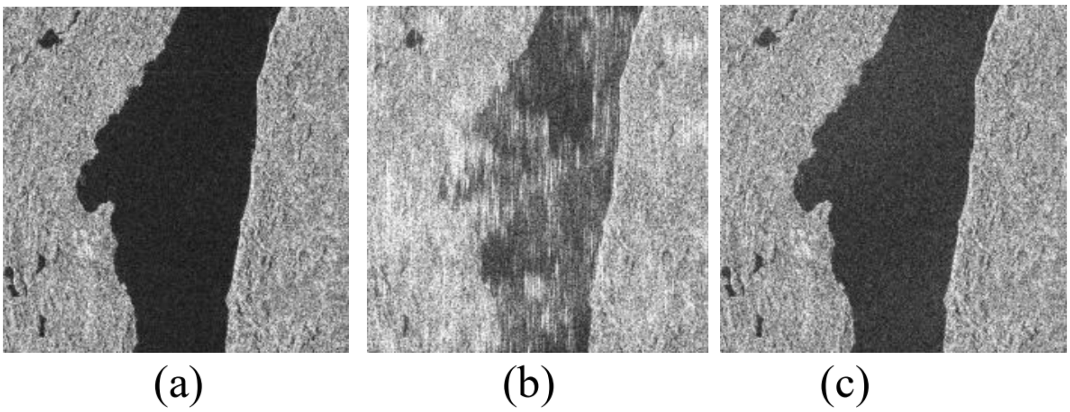


FIGURE 3.2 – Range ambiguity smearing as result of the waveform variation. (a) Reference ambiguity-free image. (b) Image corrupted by strong image ambiguities for a conventional SAR.(c) Image corrupted by the same ambiguities as (b), but for a SAR with waveform variation. This figure was got from (Villano *et al.*, 2020)

One of the final's motivations of the study of these novel SAR acquisition modes is the ability to relax the system's constraints concerning the PRF. This variable has an upper and lower design boundary defined by the limit of the ASR (Ambiguity to Signal ratio) level. With these novel modes, it is desired to be able to work with relaxed constraints on the system's PRF once the level of the ambiguities would be lower due to the imposed mismatching on the useful filter and the ambiguous signal. In this work, it was focused on the study of the range ambiguous signal and useful signal only.

3.2.1 Mode 1: Waveform encoded SAR with cyclically-shifted chirps

In this acquisition mode, the chirps signals are placed in phase difference at each PRI. The article (Villano *et al.*, 2018) details this acquisition mode. The figure 3.3 represents this method.

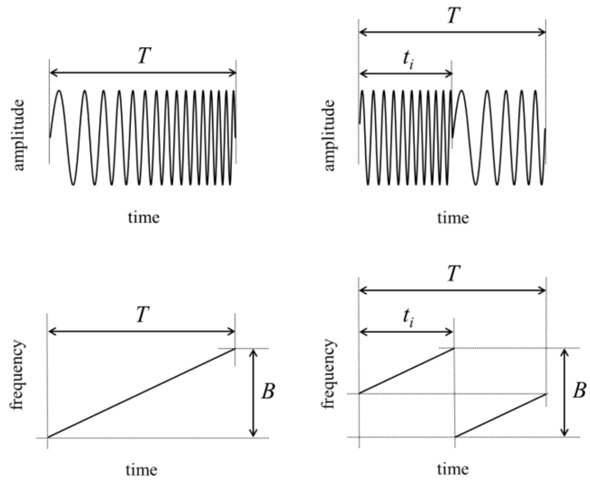


FIGURE 3.3 – Schematic representation of the (Top) radio frequency signals and the (Bottom) time–frequency diagrams of a conventional, (Left) nonshifted chirp, and (Right) cyclically shifted chirp. Image from (Villano *et al.*, 2018)

3.2.2 Mode 2: Waveform encoded SAR with up-and-down alternation

In this acquisition mode, the signal of the value of the wave modulation constant K_r is alternated with each PRI. The figure 3.5 represents the way the K_r constant of the emitted signal varies. The article (Martínez, 2003) discusses this mode in detail.

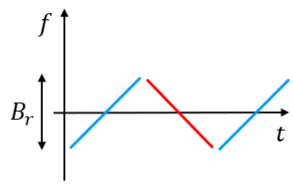


FIGURE 3.4 – Up and Down range constant alternation. (Ustalli; Villano, 2020)

3.2.3 Mode 3: Doppler-matched azimuth phase code

In this mode, a phase difference is added to every other pulse. This phase difference is not constant and depends on the wavelength and on a function of the azimuth time (η). This acquisition mode is detailed in (Villano *et al.*, 2020).

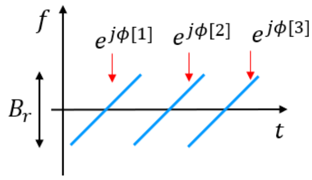


FIGURE 3.5 – Waveform for the Doppler-matched phase code acquisiton mode. This figure was got from (Ustalli; Villano, 2020)

3.2.4 Mode 4: Doppler-matched azimuth phase code & waveform encoded with Up-and-down chirp alternation

In this acquisition mode, the acquisition modes 2 and 3 are merged. The figure 3.6 shows the waveform in this case.

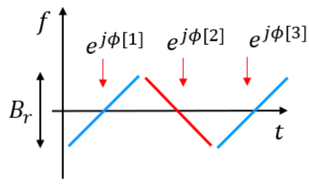


FIGURE 3.6 – Doppler-Matched azimuth phase code combined with the waveform encoded with cyclically shifted chirp mode. This figure was got from (Ustalli; Villano, 2020)

4 Methodology

In this Chapter, it will be explained how the system's desired features for the development of the proposed tool were derived. It will also be presented how new features were identified as the tool was developed, and how these new features were inserted in the project. Finally, it will be discussed how each feature was developed, and what was the rationale and basis for meeting this particular demand.

4.1 System's desired features derivation method

The software's capabilities were derived primarily based on the demands pointed out by the group, within a structure of prioritization and partial delivery of results for evaluation throughout the stage. Also, new capabilities were derived as the proposed tool was being used. Finally, it can be stated that the tool was developed based on an initial direction, but continuously redirected, when necessary, by the feedbacks of the group.

4.2 Systems's capabilities derived by the NewSpace SAR research group

As discussed earlier, the initial direction of the program was done in a meeting with the group in which was outlined the functions that the system should fulfill, what processing algorithms it should have, and the required system capabilities. The capabilities listed by the group were:

1. **The tool shall be able to simulate the two types of SAR scenes:**
   (a) Single Target
   (b) Real SAR scene
2. **For each type of scene above, i.e., Single Target Simulation and Real SAR scene, it shall be able to generate the following data related to it:**

(a) Generation of the SAR raw data

(b) Generation of the focused image from SAR raw data

4.2.1 The options for each acquisition mode: raw data generation and focusing

For each type of signal processing operation listed above, five types of acquisition modes should be simulated and for each one of these, there should be also the option to simulate both the useful or range ambiguous signal. The list below summarizes the system's capabilities for each data processing step,

1. **Single Target raw data simulation and data focusing**
   (a) Useful Signal:
      i. Conventional acquisition mode
      ii. Phase shifted novel acquisition mode
      iii. Doppler-matched novel acquisition mode
      iv. Up and Down chirp novel acquisition mode
      v. Doppler Matched & Up and Down novel acquisition mode

   (b) Range Ambiguous Signal:
      i. Conventional acquisition mode
      ii. Phase shifted novel acquisition mode
      iii. Doppler-matched novel acquisition mode
      iv. Up and Down chirp novel acquisition mode
      v. Doppler Matched & Up and Down novel acquisition mode

2. **Real Scene raw data creation and data focusing**
   (a) Useful Signal:
      i. Conventional acquisition mode
      ii. Phase shifted novel acquisition mode
      iii. Doppler-matched novel acquisition mode
      iv. Up and Down chirp novel acquisition mode
      v. Doppler Matched & Up and Down novel acquisition mode

   (b) Range Ambiguous Signal:
      i. Conventional acquisition mode
      ii. Phase shifted novel acquisition mode

iii. Doppler-matched novel acquisition mode
iv. Up and Down chirp novel acquisition mode
v. Doppler Matched & Up and Down novel acquisition mode

4.3 Additional system's capabilities derived during the code development

Throughout the program's development, some problems were listed and based on these intermediate setbacks, new features for the program were derived within the capacity of not obstructing the main objectives. These new capabilities were also derived during intermediate delivery meetings. The problems observed and that led to the derivation of secondary objectives were:

1. **Codes were not clear:** the codes until then contained very distinct steps together in the same file. Accordingly to (MARTIN, 2008), this made debugging more difficult, and the understanding of the code slowed down and generated more doubts.

2. **Low scalability:** in research, it is often necessary to make successive simulations in large numbers to validate the influence of parameters in particular on the final result. Until then, the code did not allow this use in batch and required the researcher to do code repetition or extensive manual changes to simulate new conditions.

3. **Repeated code:** causes the problem of having to change many parts of the code in case of refactoring and increases the chance of bugs.

4. **There were very often problems with RAM over-usage:** Due to many auxiliary variables being kept during the code's execution, the user often had had problems with the over-usage of the RAM memory. This has led to frequent system crashes.

4.4 How each problem was addressed

As mentioned before, to develop the software, there are the system's desired: capabilities list, performance, and integration. For each of these needs, there was a rational and methodology to be met. In the case of the needs related to data processing, the algorithms in Chapters 2 and 3 were implemented.

For the raw data creation, the equation 2.7 was used, while for the real scene case, an algorithm was used 2.4. For both cases, the algorithm was adapted as necessary due to the acquisition mode being either conventional or not. In data processing, the algorithm 2.6 was also used to adapt the acquisition mode in study.

In the case of problems related to the structure of the code and its performance, the discussion is more extensive and methods based on several principles of good programming, and optimization practices were applied. However, it is worth pointing out that this rationale is not intended to be the best possible solution, but only to meet the needs outlined above in a fast and satisfactory manner. There is, therefore, plenty of opportunities for incremental improvements in this tool.

4.4.1 Issues related to the code structure

According to (MARTIN, 2008), in the development of a code where there is the possibility of multiple flows, which is not precisely known the possibilities of inputs and outputs and which should still have easy maintenance, one should first seek to separate the domains in order to make them independent and easy to understand. Following this logic, it was sought to produce code in a modular way, where there was a separation between the input and output of data from the part where the data are processed and analyzed. It was also decided to prioritize a structure in which the parts of the codes could be changed without damaging the entire tool. For this purpose, classes, methods, and attributes were defined so that the integration between the different domains within the tool could have predictability in its integration. Thus, a structure was prioritized in which, based on a small number of classes, it was possible to process data in any of its phases: input, processing, analysis, and output. In addition, aiming at context separation, the processing, and data treatment codes were separated in a different module from the one where the user needs to interact to use the tool. This brings a direct benefit to a simplified use for the researcher and an interface that makes inappropriate use difficult. Finally, thinking about the process of use by the researcher, a facilitated way of data input and output was suggested and implemented. This structure also favors the use of code in a scalable way, as in multiple simulations in a concatenated way.

4.4.2 Issues related to code performance

Many performance problems were observed as the codes were being executed. The main problems observed were excessive use of RAM and high processing time. Thus, it was aimed to optimize each processing step by applying functions using the NumPy library and the operation with vectors.

On the other hand, to solve the problem of excessive memory use, the processing functions were separated in their respective steps, and auxiliary variables were eliminated when possible. In this way, it was sought to obtain an optimized memory use that would still allow programming languages with greater performance power in the future for the most critical steps. This methodology also aimed at indirect improvement in algorithm processing time, since it sought to eliminate the use of swap.

5 SAR Tool: an instrument to support the research on the novel SAR acquisition modes

As discussed and explained in previous Chapters, the main objective of the work was to develop a tool to assist the NewSpace SAR research group. This program has all the primary and secondary system's capabilities discussed in the previous Chapter. Therefore, this Chapter will explain the final result of the work. First, the tool's folder structure will be exposed, where it will be possible to give an idea about how the separation of contexts and the delegation of responsibilities were made. Then it will be shown how it is a usual workflow with the tool, which is detailed step by step. Once it has been clarified how the flow of information takes place and which are the tool's main areas, the input and output modes and their particular details will be discussed. Finally, in accordance with Chapter 4, it will be discussed how the separation of contexts was made and which classes were used in the final program.

5.1 SAR tool structure

The final program developed presents two major contexts: the user and the core. In the first context, the codes are related only to the configuration and use of the tool; therefore, the only layer in which the user should interact. In this layer, the simulations are configured, and the default model for simulation is changed. Here we can also find the codes related to the execution of the tool itself. In another context, in the core layer, according to the principles of clean code (MARTIN, 2008), there is all the system's business' rules. Thus, in this layer are all the processing and image generation codes, and the main entities of the system are defined.

5.1.1 Folders structure

```
Core
    Backend
    Models
    Simulations
    RawDataFromSceneAlgorithms
    FocusingAlgorithms
    PerformanceAnalysis
    Tests
User
    Requests
    setup.xlsx
    config_model.xlsx
    path_setter.py
    main_create_requests.py
    main_run_requests.py
```

The system is composed of two main contexts. In the user context, there are functions related to the use and configuration of the tool at the searcher level only. On the other hand, in the core layer, there is already a second division into contexts, so the codes are separated into seven different packages, they are: Backend, Models, Simulations, RawDataFromSceneAlgorithms, FocusingAlgorithms, PerformanceAnalysis, Tests.

In a simplified detailing of each of the packages, we have:

1. **Backend:** Here are all the modules related to the treatment and direction of the data so that the desired processing is carried out. Here are also the codes required for loading and saving data.

2. **Models:** it is the module where base classes of the system are defined
3. **Simulations:** is where the codes for single target simulation are for both signal cases
4. **RawDataFromSceneAlgorithms:** where the raw data generation from scene, using inverse omega-K, are located
5. **FocusingAlgorithms:** module where the omega-K algorithms are defined
6. **PerformanceAnalysis:** module that contains all the codes related to the data display and analysis.
7. **Tests:** all the codes related to testing each of the implemented functions and modules.

5.1.2 Tool's usage workflow

Once the components and layers of the code are understood, it becomes possible to explain the steps and how the tool is used. The tool requires two steps for a simulation to be made from the user's perspective, and its results duly persisted. A third optional step is still possible, that is the analysis of the results produced. Therefore, broadly, the roadmap for using the tool is:

1. **Setup arrangement**
   (a) Requests creation
      i. set the operational mode: batch or other
      ii. set the number of requests to create
      iii. configure the standard config file
   (b) Run the *main_create_requests.py* file
   (c) Configure each request editing the *config.xlsx* file inside of the *Input* folder
2. **Setup running**
   (a) run the *main_run_requests.py* file
3. **Result analysis**
   (a) PerformanceAnalysis

About the first part, the first step is the creation of requests. This step is done by editing the *setup.xlsx* file where the type of simulation desired and the number of simulations to be performed is selected. After that, the user must still modify the default configuration model file of a request, the file *model_config.xlsx*. Once these two files have been configured, the searcher must execute the *main_create_requests.py* function, which will create the Requests repository within the User context, and within this new directory, *request_k* folders will be inserted sequentially up to a k number equal to the number of requests requested in the *setup.xlsx* file. The figure 5.1 represents it schematically.

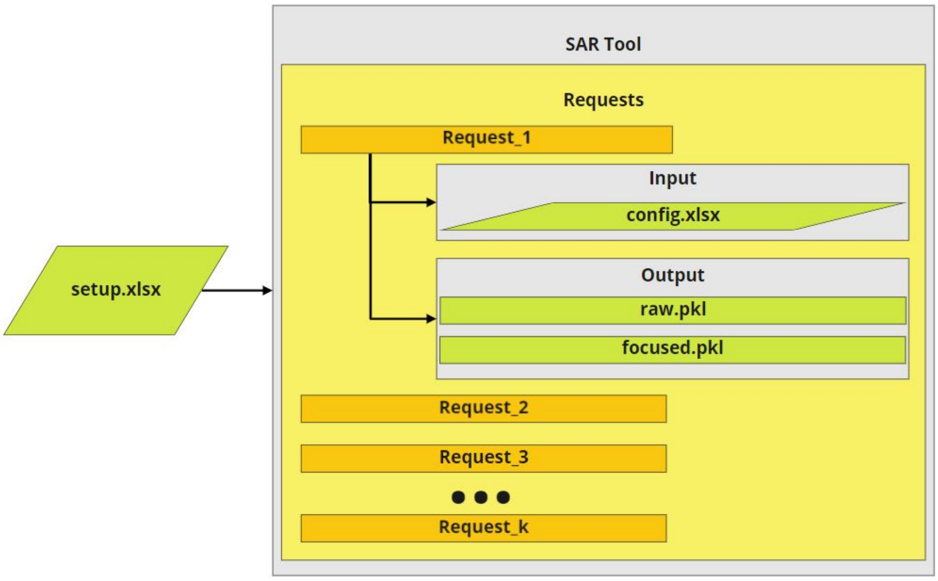
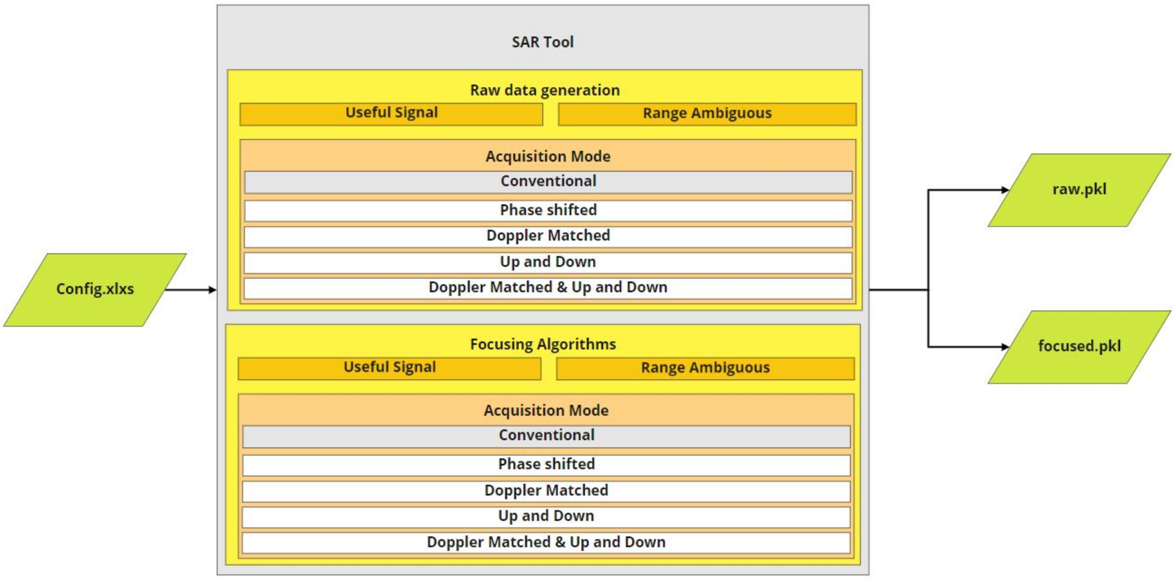


FIGURE 5.1 – Scheme of the requests folder creation

It is important to point out that currently, the tool only makes batch simulations (batches), but that in the future new simulation modes can be inserted and that these can have a different flow of use than explained above.



Following the same usage flow, once the request files have been created, the user must manually change each standard config file ($config.xlsx$) within each of the *request_k* repositories since these contain, right after their creation, default configuration files identical to the $model\_config.xlsx$ file. Once the files have been configured one by one manually, the user can proceed to the next step of using the tool.

In this new Setup Running step, the user must execute the $main_r un_r equests.py$ file and wait until all the requests have been executed.

The third and last step in using the tool is optional and consists of using the codes in Performance Analysis so that the user can plot and load the results files generated in the simulations and perform the proper analysis of them.

5.2 The system's input and output files

Here the input and output files are discussed. They have the main goal to be a structured and easy way to input the data and configure the required parameters to perform the simulations. The parameters input interface selected on this project was excel sheets due to easiness to change and configure. This project uses two types of sheets.

5.2.1 The first input file: Setup Sheet

This file *setup.xlsx* is used as the first input form for the SAR tool and it has only two fields: the $UseMode$ and the $NumberOfRequests$. The first field is related to the manner that the researcher wants to use the code. It has nowadays only two options, the *Batch* and *Other*. The objective here to choose the number of simulations that the user wants to perform. If the UseMode is set as *Batch* the tool will create as many as requests as on the field $NumberOfRequests$. Each simulation will be created on the *Requests* folder and it would be enumerated sequentially starting from 1 up to the value on $NumberOfRequests$.

However, if the researcher intends to perform a specific algorithm to create the $Requests$, it can be created using the $UseMode$ as Other, but it has not been implemented yet. However, it was put in case of new $UseCases$, such as Monte Carlo simulation.

After the researcher selected the $UseMode$ value, the *main_create_requests.py* should be executed and then the *Requests* folder would be created.

5.2.2 The second input file: Config Sheet

The second and last file to the researcher set all the desired parameters before running the raw data creation, and focusing algorithms is the *model_config.xlsx* file. This file has not only one tab, however despite it having many tabs, but all of them also have the values and their descriptions, which gives the user a first glance about what is each sheet field. To clarify a little bit more about what are the variables and configurations that can be set on this sheet, there are the following tabs: Config, TargetParameters, Constants, SystemParameters, ProcessingParameters.

A brief description of each part of this sheet is given as follows:

1. **Config**: Allows the user to select which Acquisition Mode, Signal Mode, Scene Mode, Use Case, InputPath, and the flags: save raw data, save focused data.
2. **TargetParameters**: Allows the user to set the number and characteristics of the targets in case of target simulation.
3. **Constants**: Contains all the relevant physical constants.
4. **SystemParameters**: Contains all the variables related to SAR vehicle, such as the main chirp wave characteristics and the satellite height.
5. **ProcessingParameters**: Here all the parameters related to the hamming window application, such as flag, decide whether or not to use it and the respectives α. Also, the user can set the Pulse-Doppler Bandwidth.

5.2.2.1 The field validation constraints

As requested by the group on meetings, to agree with a set of SAR variables preconditions, some previous field validations were configured. The main ideas that were discussed were about the number of samples either in range and azimuth directions. Other validations inserted on this sheet are related to the type of each variable, such as the number of targets, that must be a natural number or the sign of the range constant ($sign\_Kr$) that can assume only -1, +1, or *alternated* value. All those beforehand validations mean to the system a good manner to prevent the researcher from starting simulations that can lead to results that do not meet the minimum SAR conditions to create reliable images with sufficient resolution and other performance parameters.

5.2.3 The output files

The output files produced in the simulations must be located at the end of the simulation in the Output folder within each request are raw data or focused data. The format of these files is *.pkl*, which is why this format allows us to dispose of more complex data structures without losing the structure, thus being a suitable way to save objects of the *SARData* class and preserve their attributes.

5.3 The Core Layer

After all the relevant chunks of code being briefly discussed, it can now be detailed how the data and parameters are organized.

This layer is responsible for all the data focusing and creation algorithms, as well as for the data handlers. The main idea here is to separate the different acquisition modes into its raw data creation and focus parts from the other dependencies. This layer also holds all the required models, such as the SARData and Requests models.

5.3.1 Models

Here, the main classes are presented and discussed. They hold the main idea of context segregation. Its use provides to the system flexibility when some new parameters need to be inserted and make it easier to add new processing algorithms on the tool in the future. Hence, the classes used in this tool are the following:

1. **SARData**

2. **Requests**

3. **Parameters**:

   (a) Constants

   (b) SystemParameters

   (c) ProcessingParameters

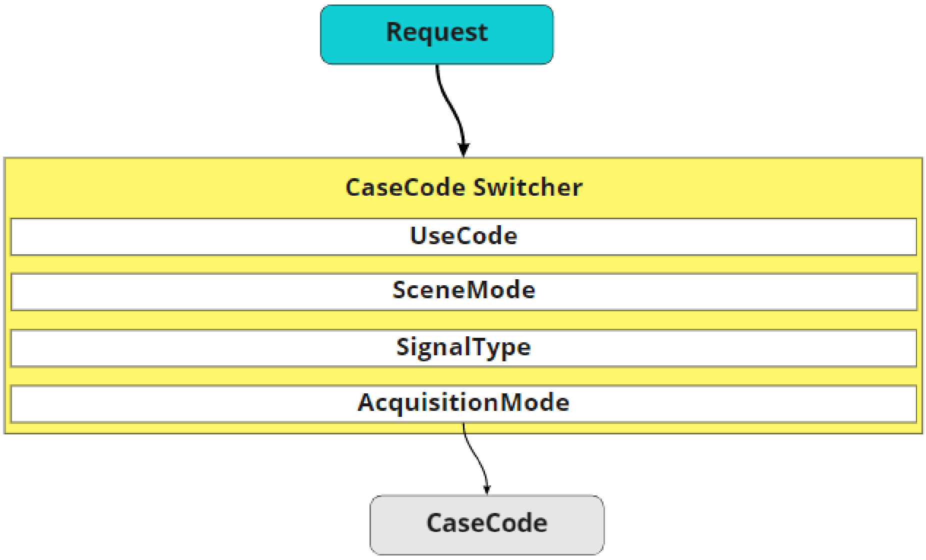
   (d) SceneParameters

5.3.2 Data Handlers

After a Request object is created, the tool should be handled to perform all the required processing steps. This object handling is done by several handlers scripts that read the object and extracts from it the required values to, in the end, address this object to the correct data processing scripts.

5.3.2.1 The CaseCode switcher

One of theses handlers scripts is the CaseCode switcher. It gets the Request object attributes and creates a CaseCode identifier, that would be used to identify what are the scripts required on a specific step. This code gets the UseCase, SceneMode, SignalType and the Acquisition mode to create the identifier to be used on the handling process.



Once the CaseCode identifier was created, the Core layer would run the corresponding algorithms to process the data until it reaches the pre-set final state. This flow would be controlled by a simple finite state machine.

5.3.2.2 The data state control

After the CaseCode being set, the tool starts to process this Request. A simple finite state machine is used by reading the UseCase it gets the initial, current, and final states for an object. With this information, it is possible to run all the required processing steps sequentially and saving the raw data in the middle of the data processing when required. Figure 5.2 represents this finite state machine.

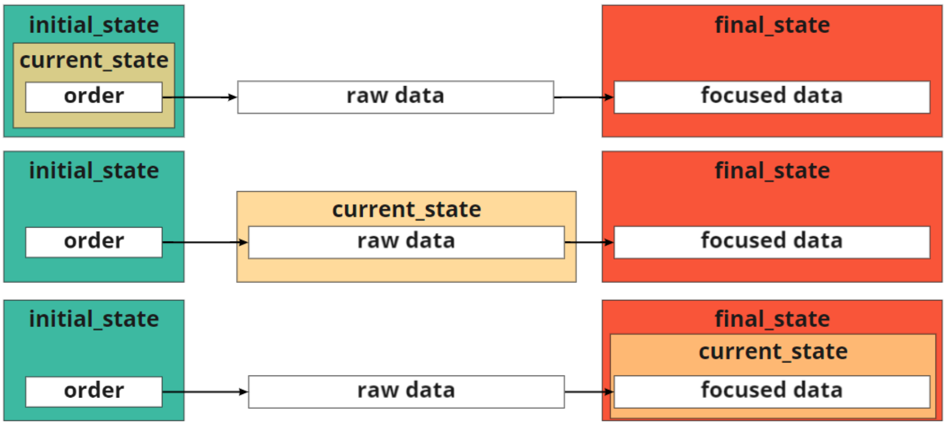


FIGURE 5.2 – States diagram of the Request object in the Requests queue

This state machine allows the use of the tool only for a specific step, such as only focusing or only creating the raw data.

5.3.3 The Performance Analysis block

After the request processing, the researcher can analyze the produced data using the performance Analysis script. It allows data loading to perform the raw data or focused data plotting. It also has prebuilt functions to support the plotting for the most common use cases.

6 Results for the Single Target Simulation use case

This Chapter and the next contains the execution times results for each algorithm implemented. The values reported here are not necessarily fixed, i.e the real execution time will depend on the researcher's hardware. Hence, those values reported are only for comparison purposes. Nevertheless, they still can show the most relevant results and drive to the main conclusions.

For the Single Target Simulation scene use case, the results were produced using the equation 2.7. the five proposed acquisition modes were simulated both for the useful signal and for the range ambiguous signal. the dimensions of the simulated scenes were

- number of samples in range $(n_{rg}) = 4096 = 2^{12}$
- number of samples in azimuth $(n_{az}) = 16384 = 2^{14}$

After the raw data creation, the scene was focused using the omega-k algorithm, as described in 2.5.

6.1 Raw data generation

The results in this section are the execution times related to the generation of the SAR raw data. They were separated on the two signal types: useful and range ambiguous signal and for each type of signal the five SAR acquisition modes processing times were reported. The times were gathered using the *time* python library.

To reduce the size of the tables in the presentation of results, the names of the innovative SAR acquisition modes under study were simplified according to the table 6.1

Shortened name	Novel SAR acquisition mode
Phase Shifted	waveform encoded SAR with cyclically-shifted chirps
Doppler Matched	Doppler-matched azimuth phase code
Up and Down	waveform encoded SAR with up-and-down alternation
Doppler Matched & Up and Down	waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

TABLE 6.1 – Shortened names for each of the novel SAR acquisition modes analyzed in this work

6.1.1 Useful Signal use case

For this type of signal, the execution times for each acquisition mode were reported on the table 6.2

TABLE 6.2 – Execution times for the original and optimized codes in the useful signal for the single target simulation use case

Acquisition Mode	Original Code [s]	Optimized Code [s]	Relative difference (%)
Conventional	13.19	13.33	-1.06
Phase Shifted	15	15.57	-3.80
Doppler Matched	20.34	20.09	1.23
Up Down	17.02	15.38	9.64
Doppler Matched & Up Down	24.23	29.48	-21.67

6.1.2 Range Ambiguous Signal use case

On the Range ambiguous signal case, for the *range_ambiguity_order* $= 1$, the results were the ones described in the table 6.3.

TABLE 6.3 – Execution times for the original and optimized codes in the range ambiguous signal for the single target simulation use case

Acquisition Mode	Original Code [s]	Optimized Code [s]	Relative difference (%)
Conventional	13.77	13.45	2.32
Phase Shifted	16.58	15.57	6.09
Doppler Matched	53.64	18.00	66.44
Up Down	15.69	15.9	-1.34
Doppler Matched & Up Down	23.03	15.9	30.96

6.2 Raw data focusing

After generating the raw data in the previous step, the Omega-K algorithm was applied to focus the scene. The steps reported are the one represented in 2.5.

6.2.1 Useful Signal

In this subsection, the execution times, of the focusing data processing step, were reported for the useful signal use case for each one of the acquisition modes.

6.2.1.1 Conventional Acquisition mode

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**39.33**	**35.83**	**8.89%**
Step 1	3.88	-	-
Step 2	8.09	9.05	-11.85%
Step 3	6.55	6.78	-3.43%
Step 4	16.88	14.7	12.94%
Step 5	3.93	5.31	-35.27%

6.2.1.2 waveform encoded SAR with cyclically-shifted chirps

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**74.98**	**46.34**	**38.21%**
Step 1	3.89	-	-
Step 2	20.84	20.86	-0.08%
Step 3	7.48	6.49	13.20%
Step 4	36.37	15.57	57.19%
Step 5	6.40	3.42	46.64%

6.2.1.3 Doppler-matched azimuth phase code

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**45.40**	**48.77**	**-7.42%**
Step 1	3.76	-	-
Step 2	13.17	12.67	3.81%
Step 3	5.10	6.78	-32.94%
Step 4	14.49	14.29	1.38%
Step 5	8.88	15.03	-69.26%

6.2.1.4 waveform encoded SAR with up-and-down alternation

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**40.82**	**38.3**	**6.17%**
Step 1	2.92	-	-
Step 2	14.89	14.19	4.70%
Step 3	5.54	6.82	-23.10%
Step 4	15.51	14.04	9.48%
Step 5	1.96	3.25	-65.82%

6.2.1.5 waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**189.55**	**50.79**	**73.20%**
Step 1	3.74	-	-
Step 2	16.20	19.3	-19.14%
Step 3	7.09	6.47	8.80%
Step 4	52.54	6.47	87.69%
Step 5	109.98	18.55	83.13%

6.2.2 Range Ambiguous Signal

In this subsection, were reported the execution times, for the data focusing processing step, for the range ambiguous signal use case, for each one of the acquisition modes.

6.2.2.1 Conventional Acquisition mode

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**35.33**	**34.05**	**3.62%**
Step 1	2.75	-	-
Step 2	6.25	7.45	-19.20%
Step 3	5.60	6.89	-23.04%
Step 4	15.61	14.35	8.07%
Step 5	5.12	5.36	-4.69%

6.2.2.2 waveform encoded SAR with cyclically-shifted chirps

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**57.05**	**44.26**	**22.42%**
Step 1	3.67	-	-
Step 2	21.95	20.13	8.29%
Step 3	7.05	6.39	9.36%
Step 4	19.08	14.25	25.31%
Step 5	5.30	3.49	34.15%

6.2.2.3 Doppler-matched azimuth phase code

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**1,264.80**	**54.78**	**95.67%**
Step 1	3.14	-	-
Step 2	216.06	14.92	93.09%
Step 3	9.83	7.1	27.77%
Step 4	427.45	15.37	96.40%
Step 5	608.32	17.39	97.14%

6.2.2.4 waveform encoded SAR with up-and-down alternation

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**45.06**	**39.93**	**11.38%**
Step 1	3.90	-	-
Step 2	14.46	14.95	-3.39%
Step 3	6.15	6.81	-10.73%
Step 4	15.07	14.86	1.39%
Step 5	5.48	3.31	39.60%

6.2.2.5 waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**496.48**	**44.48**	**91.04%**
Step 1	3.26	2.34	-480.67%
Step 2	18.78	18.93	71.94%
Step 3	8.38	5.27	-31.98%
Step 4	463.80	11.06	97.62%
Step 5	2.26	9.22	-307.96%

6.3 Discussion

As expected, in the case of raw data generation, the optimization in the organization of the code and its data structures did not cause an increase in the execution time efficiency. This happened because the routine applied in this processing step is only the 2.7, which does not demand excessive memory for storage of auxiliary variables.

On the other hand, the optimization in the focusing stage using the adapted omega-k algorithm proved effective for three of the acquisition modes in the case of ambiguous range. The optimization of memory usage in this stage of data processing makes the limitation in processing time no longer of the lack of memory, but of the processing power of the hardware in question. The optimization in this case of ambiguous range had one of the greatest highlights in the case of the Doppler-Matched azimuth phase code acquisition mode, in which the original execution time was more than 20 times longer than the time required in optimized code condition.

In terms of the algorithm steps, in the case of data focusing, steps 4 and 5 were, in relative terms, the most demanding to the code and that generated the most distortions in the execution time. The optimization allowed no more abrupt peaks in memory use in these steps which eliminated the bottleneck in this case.

7 Results for the Real Scene use case

In this Chapter, the results are exposed and discussed as to the processing time for the case of simulation from the real scene. The simulated scene was generated by real images gathered with TerraSAR-X.

7.1 Raw data creation

The results in this section are the execution times related to the generation of the SAR focused data. They were separated on the two signal types: useful and range ambiguous signal and for each type of signal the five SAR acquisition modes processing time were reported.

7.1.1 Useful Signal

In this subsection, the execution times, of the raw data creation processing step, were reported for the useful signal use case for each one of the acquisition modes.

7.1.1.1 Conventional Acquisition mode

TABLE 7.1 – Processing times for the Raw data creation from real SAR scene use case, for the useful signal transmitted in the conventional SAR acquisition mode

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**75.93**	**54.99**	**27.58%**
Step 1	8.82	7.72	12.47%
Step 2	3.82	8.73	-128.53%
Step 3	29.25	3.79	87.04%
Step 4	23.15	26.92	-16.29%
Step 5	4.18	3.83	8.42%
Step 6	2.75	1.01	63.34%
Step 7	3.96	2.99	24.45%

7.1.1.2 waveform encoded SAR with cyclically-shifted chirps

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**96.08**	**69.25**	**27.92%**
Step 1	9.108	7.58	16.73%
Step 2	7.76	10.42	-34.31%
Step 3	3.39	3.72	-9.84%
Step 4	25.61	27.5	-7.40%
Step 5	3.29	3.76	-14.15%
Step 6	44.15	13.21	70.07%
Step 7	2.77	3.05	-10.11%

7.1.1.3 Doppler-matched azimuth phase code

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**65.68**	**67.88**	**-3.35%**
Step 1	8.85	7.98	9.85%
Step 2	7.40	9.21	-24.53%
Step 3	3.14	3.83	-21.94%
Step 4	25.36	27.92	-10.08%
Step 5	3.17	3.78	-19.24%
Step 6	13.76	12.14	11.77%
Step 7	4	3.02	24.49%

7.1.1.4 waveform encoded SAR with up-and-down alternation

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**73.72**	**67.97**	**7.80%**
Step 1	7.66	8	-4.44%
Step 2	7.25	8.89	-22.62%
Step 3	3.27	3.85	-17.65%
Step 4	27.16	27.13	0.11%
Step 5	3.27	3.78	-15.60%
Step 6	22.32	13.28	40.51%
Step 7	2.79	3.04	-8.96%

7.1.1.5 waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**66.35**	**75.52**	**-13.81%**
Step 1	8.15	7.57	7.14%
Step 2	7.95	9.41	-18.40%
Step 3	3.255	3.79	-16.53%
Step 4	23.08	27.57	-19.45%
Step 5	3.21	3.77	-17.56%
Step 6	18.10	20.4	-12.74%
Step 7	2.62	3.01	-14.85%

7.1.2 Range Ambiguous Signal

below are the results for the range ambiguous case on the raw data generation for the real scene

7.1.2.1 Conventional Acquisition mode

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**63.16**	**59.11**	**6.40%**
Step 1	7.26	6.56	9.64%
Step 2	7.48	7.99	-6.83%
Step 3	3.48	3.34	4.05%
Step 4	21.91	23.40	-6.80%
Step 5	18.64	4.01	78.49%
Step 6	3.43	11.21	-226.82%
Step 7	0.955	2.60	-172.25%

7.1.2.2 waveform encoded SAR with cyclically-shifted chirps

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**734.96**	**59.95**	**91.84%**
Step 1	4.07	6.96	-71.07%
Step 2	22.95	7.78	66.12%
Step 3	18.25	3.20	82.50%
Step 4	210.22	23.84	88.66%
Step 5	29.40	4.31	85.34%
Step 6	277.10	11.23	95.95%
Step 7	172.97	2.64	98.47%

7.1.2.3 Doppler-matched azimuth phase code

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**235.30**	**61.78**	**73.75%**
Step 1	7.0562	6.91	2.02%
Step 2	7.89	7.33	7.10%
Step 3	3.61	3.15	12.61%
Step 4	32.76	23.46	28.39%
Step 5	3.34	4.14	-23.95%
Step 6	180.65	14.26	92.11%
Step 7	32.93	2.52	92.35%

7.1.2.4 waveform encoded SAR with up-and-down alternation

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**239.37**	**58.67**	**75.49%**
Step 1	7.83	7.12	9.06%
Step 2	7.39	7.46	-0.95%
Step 3	3.49	3.32	4.95%
Step 4	47.22	23.22	50.82%
Step 5	3.25	3.77	-15.89%
Step 6	164.62	11.18	93.21%
Step 7	5.56	2.61	53.09%

7.1.2.5 waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**756.46**	**62.57**	**91.73%**
Step 1	7.63	7.23	5.24%
Step 2	7.77	7.42	4.53%
Step 3	3.26	3.21	1.53%
Step 4	258.55	24.06	90.70%
Step 5	3.54	4.35	-23.02%
Step 6	472.88	13.79	97.08%
Step 7	2.83	2.51	11.31%

7.2 Raw data focusing

In this step, the raw data obtained with the inverse algorithm omega-K was focused on the previous step. The focus was done using the adapted omega-K algorithm, as 2.5. The number of steps for this step is different from the one represented in the block diagram because of the way the code was implemented.

7.2.1 Useful signal

In this subsection, the execution times, of the data focusing processing step, were reported for the useful signal use case for each one of the acquisition modes.

7.2.1.1 Conventional Acquisition mode

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**589.54**	**45.15**	**92.34%**
Step 1	3.24	2.71	16.20%
Step 2	4.22	3.7	12.32%
Step 3	11.17	10.04	10.11%
Step 4	30.17	10.65	64.71%
Step 5	4.51	3.61	19.96%
Step 6	118.63	7.82	93.40%
Step 7	417.6	6.62	98.41%

7.2.1.2 waveform encoded SAR with cyclically-shifted chirps

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**289.46**	**69.25**	**76.08%**
Step 1	2.91	7.58	-160.36%
Step 2	18.15	10.42	42.56%
Step 3	9.94	3.72	62.58%
Step 4	54.57	27.5	49.61%
Step 5	16.02	3.76	76.53%
Step 6	168.14	13.21	92.14%
Step 7	19.73	3.05	84.54%

7.2.1.3 Doppler-matched azimuth phase code

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**481.41**	**67.88**	**85.90%**
Step 1	2.63	7.98	-203.42%
Step 2	13.53	9.21	31.91%
Step 3	10.12	3.83	62.15%
Step 4	144.72	27.92	80.71%
Step 5	5.44	3.78	30.58%
Step 6	123.89	12.14	90.20%
Step 7	181.08	3.02	98.33%

7.2.1.4 waveform encoded SAR with up-and-down alternation

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**593.10**	**67.97**	**88.54%**
Step 1	2.92	8	-173.97%
Step 2	15.42	8.89	42.35%
Step 3	8.94	3.85	56.98%
Step 4	41.47	27.13	34.58%
Step 5	5.491	3.78	31.16%
Step 6	87.11	13.28	84.76%
Step 7	431.75	3.04	99.30%

7.2.1.5 waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

	Original [s]	**Optimized [s]**	**Relative dif.**
Full time processing	**96.53**	**75.52**	**21.76%**
Step 1	2.42	7.57	-212.81%
Step 2	25.64	9.41	63.30%
Step 3	9.8	3.79	61.30%
Step 4	21.11	27.57	-30.58%
Step 5	2.65	3.77	-42.43%
Step 6	8.38	20.4	-143.44%
Step 7	26.53	3.01	88.66%

7.2.2 Range Ambiguous Signal

In this subsection, the execution times, of the data focusing processing step, were reported for the range ambiguous signal use case for each one of the acquisition modes.

7.2.2.1 Conventional SAR acquisition mode

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**564.68**	**56.52**	**89.99%**
Step 1	2.69	7.44	-176.58%
Step 2	3.43	8.11	-136.44%
Step 3	8.74	3.49	60.10%
Step 4	31.41	24.83	20.95%
Step 5	4.6	3.47	24.63%
Step 6	79.77	12.20	84.71%
Step 7	434.04	2.7	99.38%

7.2.2.2 waveform encoded SAR with cyclically-shifted chirps

	Original [s]	Optimized [s]	Relative dif.
Full time processing	**251.42**	**56.52**	**77.52%**
Step 1	4.00	2.74	31.58%
Step 2	23.81	15.87	33.35%
Step 3	12.06	10.46	13.26%
Step 4	41.51	9.87	76.22%
Step 5	9.48	3.49	63.15%
Step 6	122.57	7.15	94.17%
Step 7	37.99	6.94	81.73%

7.2.2.3 Doppler-matched azimuth phase code

	Original [s]	Optimized [s]	Optimization level
Full time processing	**280.84**	**66.73**	**76.24%**
Step 1	2.88	2.92	-1.34%
Step 2	14.37	22.11	-53.91%
Step 3	10.31	11.56	-12.12%
Step 4	121.50	11.04	90.92%
Step 5	8.33	3.89	53.30%
Step 6	46.13	7.65	83.42%
Step 7	77.33	7.57	90.21%

7.2.2.4 waveform encoded SAR with up-and-down alternation

Full time processing	**Original [s]** **454.03**	**Optimized [s]** **55.10**	**Optimization level** **87.86%**
Step 1	4.02	2.65	34.17%
Step 2	15.59	14.29	8.36%
Step 3	9.57	10.53	-10.09%
Step 4	275.5	10.05	96.35%
Step 5	130.82	3.45	97.36%
Step 6	11.69	7.13	39.01%
Step 7	6.84	7	-2.34%

7.2.2.5 waveform encoded SAR with up-and-down alternation with Doppler-matched azimuth phase code

Full time processing	**Original [s]** **508.43**	**Optimized [s]** **65.51**	**Optimization level** **87.12%**
Step 1	15.48	2.82	81.78%
Step 2	22.5	21.68	3.66%
Step 3	10.96	11.68	-6.54%
Step 4	316.9	10.93	96.55%
Step 5	119.35	3.67	96.92%
Step 6	10.89	7.60	30.26%
Step 7	12.35	7.14	42.19%

7.3 Discussion

In the case of simulation from a real scene, optimization was effective both for raw data generation and for the stage of focusing to create the scene.

In the case of the first step, the use of the algorithm inverse omega-k, as represented in 2.4, in relative terms, had an improved performance in steps 4 and 6, only in the case of the ambiguous signal. Before optimization, these steps had peaked in execution time, precisely because of the excessive use of memory to store auxiliary variables. After the optimization, there was a constancy in the execution time for a value of around 70 seconds. The bottleneck in the processing times observed in the non-optimized cases was eliminated, which is evidenced by the maximum execution time for anyone of the five modes in the optimized case being in the order of one-tenth of the maximum time in the non-optimized cases.

In the stage of focusing the raw data for the generation of the SAR scene, the same order of magnitude in the achieved results is given. The optimized case also made that the execution time of the bottleneck stages was now limited due to the improved use of RAM. The order of magnitude in the algorithms performance increase in terms of processing was also in the order of ten times faster for the optimized case.

It is interesting to note that in this case the same algorithm is applied for raw data generation and for focusing, with only the steps in reverse order. An expected consequence is that the data processing time for focusing is similar to the time for generating this raw data. This result is observed in the case of the optimized code, where both steps have an execution time of about 65s.

Finally, a consequence of the optimization of the data structure and the way of allocating memory to auxiliary variables in an optimized way is that the execution time of the algorithm becomes much more limited by the processing power of the hardware in use than the available memory.

8 Conclusions

As the main conclusions about the final results gathered from the tool and the program's usage we got that it was successfully developed a tool to support the NewSpace SAR group researchers. All the listed desired system's capabilities were implemented. The final code allows the use using an easy sheet to configure the simulation and each user's request. The use of sheets on opposite to the standard main file parameters configure, gives the flexibility to the researcher to set all the parameters in a clearer way and provide an interface that can be modified easily in the future. Furthermore, all the desired SAR acquisition modes were either implemented as well as optimized.

On the aspect of the algorithm's optimization, the tool allows the researcher to use much less memory and also take less time to run the simulations due to the mitigation of the RAM over-usage problem. The main driver of this improvement was the OOP that brought to the algorithms and data structures a better implementation, especially when it comes to the focusing algorithm, where the execution times cutting observed were at least 70% of the original required execution time when in a condition of RAM over-usage.

Another benefit was the code reliability improvement. Due to the better memory allocation, now the simulations can be made in a manner that prevents the memory over-usage problem. This also makes the program more stable and less prone to crash.

The modular structure supports the further improvement of the code and the implementation of new features without a heavy necessity of code refactoring. It also enables the integration of the tool with other codes and studies, widening the ability of integration of the group with other researchers. Another benefit of this type of structure is that it makes it easier to give maintenance.

Finally, the queue structure to handle the requests provides the feature of multiple simulations, which accelerates the research and exploitation of a large set of systems parameters ad configurations.

8.1 Further work

Focused on the Python language issue concerning the processing times, the tool could be improved using other programming languages more focused on performance, such as the C. By doing this, the heavy steps of auxiliary variables creation can be made faster and using memory wisely.

Related to the actual code, the simulation's steps by itself could also be improved. For the use case where the user needs to simulate more than one case at once, the auxiliary variables could be calculated upfront and it would also lead to a big optimization in terms of the execution memory load. Concerning this same use case, some specific libraries for mathematical optimizations in python, such as $Numba$, can be implemented to pre-compile the common steps for the different simulations options.

Related to the new features to be implemented on the SAR tool, one improvement can come from the multiple target simulation capacity expansion on the single target simulation feature. The system already supports the creation of multiple targets, hence a few changes in the target simulation could be sufficient to add this feature.

In a usability improvement level, instead of two excel sheets being used as input standards it can be substituted by a Jupyter notebook.

Another improvement suggested by the author is the insertion of an input validation layer in between the input sheets and the requests creation routine, such as validating the number of samples in both directions to avoid problems with the sampling rate that does not meet the Nyquist condition.

Bibliography

Cumming, I. G.; Wong, F. H. **Digital processing of synthetic aperture radar data: algorithms and implementation**. [S.l.]: Artech House, 2002.

MARTIN, R. C. **Clean Code: A Handbook of Agile Software Craftsmanship**. 1st. ed. [S.l.]: Pearson, 2008.

Martínez, J. M. J. M. Analysis of range ambiguity suppression in sar by up and down chirp modulation for point and distributed targets. **IEEE**, v. 6, p. 3, 2003.

Moreira, A.; Prats-Iraola, P.; Younis, M.; Krieger, G.; Hajnsek, I.; Papathanassiou, K. P. A tutorial on synthetic aperture radar. **IEEE Geoscience and Remote Sensing Magazine**, v. 1, n. 1, p. 6–43, 2013.

Ustalli, N.; Villano, M. **Impact of Ambiguity Statistics on Information Retrieval for Conventional and Novel SAR Modes**. 2020. Conference Presentation.

Villano, M.; Krieger, G.; Moreira, A. Nadir echo removal in synthetic aperture radar via waveform diversity and dual-focus postprocessing. **IEEE GEOSCIENCE AND REMOTE SENSING LETTERS**, v. 15, n. 5, 2018.

Villano, M.; Krieger, G.; Moreira, A. Newspace sar: Disruptive concepts for cost-effective sar system design. 01 2020.

FOLHA DE REGISTRO DO DOCUMENTO			
1. CLASSIFICAÇÃO/TIPO TC	2. DATA 25 de setembro de 2020	3. DOCUMENTO Nº	4. Nº DE PÁGINAS 62
5. TÍTULO E SUBTÍTULO: Development of a tool to support the research on novel acquisition modes of Synthetic Aperture Radar (SAR)			
6. AUTOR(ES): **Carlos Alexandre Novak Madureira**			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: SAR; Omega-K; Memory Optimization; Inverse Omega-K			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Radar de abertura sintética; Memória (computadores); Otimização; Linguagem de programação; Engenharia eletrônica			
10. APRESENTAÇÃO: ( ) **Nacional** (**X**) **Internacional** Trabalho de Graduação, ITA, Oberpfaffenhofen, Germany, 2020. 62 páginas.			
11. RESUMO: In this work a tool was developed to study innovative modes of acquisition in synthetic aperture radars. The objective was to implement the codes proposed by the research group NewSpaceSAR to have a tool in Python whose use allows the simulation of the usual signal and the signal with ambiguity in the range direction, for four innovative acquisition modes for synthetic aperture radar and the conventional mode of operation. The optimization of each of the routines in terms of memory use has also been done to mitigate the problem of excessive use of RAM. The final code was developed with a modular structure and following the principles of good programming practices. Routines were also developed so that the analysis of the simulation results is facilitated.			
12. GRAU DE SIGILO: (**X**) **OSTENSIVO** ( ) **RESERVADO** ( ) **SECRETO**			