ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Máster

DISEÑO Y VERIFICACIÓN DE PROCESADOS DE DATOS DE MEDIDAS DE CAMPO CERCANO PLANO EN ENTORNO INDUSTRIAL

(DESIGN AND VERIFICATION OF DATA PROCESSES OF PLANAR NEAR FIELD MEASUREMENTS IN INDUSTRIAL ENVIRONMENT)

Para acceder al Título de

Máster Universitario en
Ingeniería de Telecomunicación

Autor: JOAQUÍN MARTÍNEZ CRESPO

OCTUBRE - 2017

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE MASTER

Realizado por: JOAQUÍN MARTÍNEZ CRESPO
Director del TFM: ENRIQUE VENERO GÓMEZ

Título: “ DISEÑO Y VERIFICACIÓN DE PROCESADOS DE DATOS DE MEDIDAS DE CAMPO CERCANO PLANO EN ENTORNO INDUSTRIAL ”
Title: “ DESIGN AND VERIFICATION OF DATA PROCESSES OF PLANAR NEAR FIELD MEASUREMENTS IN INDUSTRIAL ENVIRONMENT ”

Presentado a examen el 13 de OCTUBRE del 2017

para acceder al Título de

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

<u>Composición del Tribunal:</u>
Presidente (Apellidos, Nombre): Torres Jiménez, Rafael Pedro
Secretario (Apellidos, Nombre): Basterrechea Verdeja, José
Vocal (Apellidos, Nombre): Mirapeix Serrano, Jesús María

Este Tribunal ha resuelto otorgar la calificación de: ....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFM
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Máster Nº
(a asignar por Secretaría)

Este TFM ha salido adelante después de 6 meses en Asysol, así que todo mi agradecimiento a todos sus empleados (y los de TTI), en especial a Enrique y Ramón, por hacerme sentir como una rata más desde el primer día. Tenéis la culpa también de que me hayan contratado, así que gracias de nuevo.

También quería agradecer a todo el departamento de GISAR las facilidades que me han dado para poder realizar este TFM en una empresa.

A mis colegas. Los de clase, por cada hora de prácticas, cada clase... y cada aprobado, que nos hemos ganado a pulso. Los que no, por ayudarme a desconectar, servir de desahogo, ser consejeros, expertos... En Santander, en Madrid, en Barcelona, en donde sea... A todos por estar ahí.

A mi familia, por esa palmada en la espalda cada vez que lo necesitaba. Desde cada vez que mis padres me preguntaban por los estudios (sin muchas veces entender la respuesta), hasta la última ecuación de planos de Víctor, pasando por cada cena en familia.

Dedicatoria especial, como siempre, a todo el que dudara o no quisiera que me conviertiera en ingeniero. No pudo ser.

TOP, TOP, TOP, ******

Soy Iron-Man.
Tony Stark

TABLE OF CONTENTS

List of Figures

Chapter 1

STATE OF THE ART

1.1 INTRODUCTION

An antenna is, in most cases, a metallic structure which is used in a communication system for radiating or receiving electromagnetic waves. They act always as a transducer between the signal in a transmission line or waveguide and the free-space [1].

From a circuital point of view, the antenna behaves as an impedance connected to the guiding structure. Hence, typical equivalent circuit parameters can be used to account for antenna parameters like losses, power supplied, maximum transferred energy and so on. In a non-exhaustive way, the most common antenna types can be grouped into the following categories:

- Wire antennas: Their main characteristic is that they are constructed with conducting wires that support current densities which give origin to the radiated fields. They can be formed by straight wires (like a dipole), loops (circular, square or of any arbitrary form) or even helixes. These wires support currents and charge densities with harmonic behavior along them.
- Aperture antennas and reflectors: In these kind of antennas, the radiated fields are mainly generated by the distribution of fields generated by the feeding system over a real or equivalent area instead of the current and charge densities over their metallizations. They are usually excited with waveguides. Some examples are horn-type antennas (pyramidal and conical) or the apertures or slots on conducting planes. In this case, the electric and magnetic fields on the aperture have a harmonic behavior. The use of reflectors associated to a primary feeder, allows to have antennas with the quality for communications services at great distances (including space communications). The most common reflector is the parabolic antenna.
- Antenna clusters: In certain applications, radiation characteristics that are not achievable with a single element are required; However, with the combination of several of them a great flexibility is obtained that allows obtaining them. These clusters can be made by combining, in principle, any type of antenna.

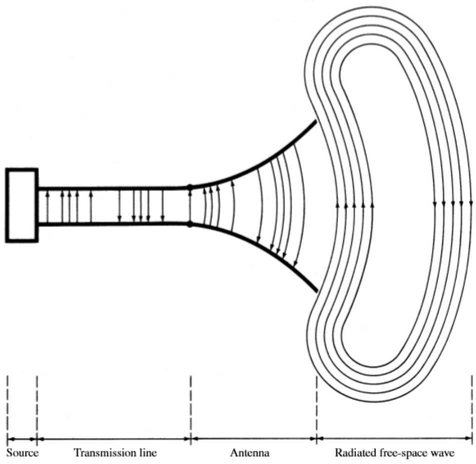

Figure 1.1: Antenna as a transition element between a transmission line and the free space [1].

Like each component in a communication system, antenna design involves a pure design stage, were the conceptual design is developed and, usually with the help of electromagnetic simulators, the conceptual design is translated into a physical structure. A validation stage where measurements are carried out to verify the physical design behavior is required as final step.

Apart from the specifications concerning equivalent circuit parameters, typically stated in terms of return losses (matching) and antenna efficiency (losses), both the directional characteristics of the antenna and the polarization of radiated field are typical specific requirements for each communication systems. As a consequence, during the design process antenna designers are required to include restrictions on the radiation pattern and the feeding system in order to control directional pattern characteristics as well as the resulting field polarization.

Antenna measurement systems are fundamental tools during the design process of any antenna in order to assure its proper operation under the restrictions imposed by the communication system.

Those measurement systems are used to check the degree of compliance with the specifications of the different specification parameters. Next section summarizes the most typical parameters used to specify an antenna system.

1.2 MAIN PARAMETERS OF AN ANTENNA

Apart from the typically used concerning equivalent circuit, most important specification parameters for an antenna are the following: radiation pattern (amplitude and phase), directivity, gain, efficiency and polarization. They are briefly described in the following: [2] [3].

1.2.1 RADIATION PATTERNS

The radiation pattern is the graphical representation of power radiated by an antenna in the different space directions. Radiation pattern is a far field (FF) parameter that shows the magnitude or phase for any (θ,ϕ) combination for an sphere located in the FF region.

Radiation patterns are often shown as field pattern (normalized and on a lineal scale) or power pattern (on a logarithm scale). The power density is proportional to the square of the electric field module, so the graphical representation of a power diagram contains the same information as a field radiation diagram.

A typical radiation pattern is constituted by lobes around local maxima with local minimum around it. These lobes show the power or field levels in the space directions around the local maximum, giving a representation of the distribution of power on different space directions. The lobe containing the global maximum is called the main lobe and determines the set of directions on which the antenna concentrates or directs most part of its radiated power.

While radiation information is three-dimensional, it may be of interest to represent just a simple cut of the diagram. The most common are those that follow the meridians in a hypothetical sphere (cuts for constant ϕ) or parallels (cuts with constant θ). The information of all the cuts of the diagram may be excessive, reason why this information is provided only for the main planes.

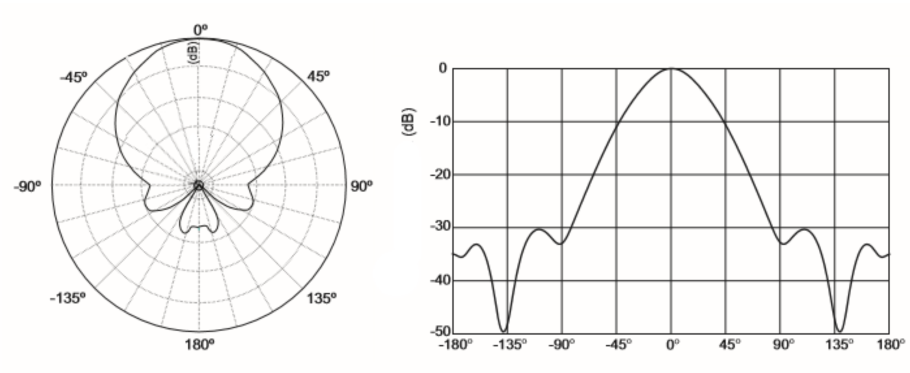

Figure 1.2: Polar diagram (left) and rectangular diagram (right). Radial/vertical axis shows the amplitude pattern (dB) and circular/horizontal axis shows the angle[3].

A parameter derived from the radiation pattern that is typically specified is the side lobe level (SLL). It is defined as the ratio, expressed in dB, between the pattern maximum and the pattern level at the maximum of the rest of the lobes, which are called secondary lobes. Normally, this relationship refers to the lobes adjacent to the main lobe which are the lobes of greater amplitude.

1.2.2 DIRECTIVITY

Considering rms values for the fields, the radiation power density in each direction of the space (θ, ϕ, r) is given in terms of the radiated fields by:

$$\vec{\mathcal{P}} = Re(\vec{E} X \vec{H}^*) \quad W/m^2$$

Total radiated power can be obtained by integration along the whole surface of the sphere of the radiated power density along the different directions according to:

$$P_{rad} = \int\int_S \vec{\mathcal{P}}(\phi,\theta) * d\vec{s}$$

From this total radiated power, according to the formal definition, directivity function can be obtained as the ratio between radiation power density in each direction at a given distance, and the density of an isotropic antenna with the same total radiated power, using the following expression:

$$D(\phi,\theta) = \frac{\mathcal{P}(\phi,\theta)4\pi r^2}{P_{rad}}$$

The directivity can be obtained, in general, from the knowledge of the radiation pattern of the antenna: A clearly predominant main lobe is expected working with high-directivity antennas, and the lowest directivity antenna is the theoretical isotropic antenna. When no direction is specified, directivity refers to the direction of maximum radiation.

1.2.3 GAIN

A second parameter directly related to the directivity is the gain of antenna. Gain definition is similar, but taking as reference the power delivered to the input terminals instead of to the total radiated power. This parameter takes into account possible losses in the antenna, since not all the power delivered is radiated to space. Gain and directivity are therefore related to the efficiency of the antenna.

Gain is the most important figure of merit that describes the performance of an antenna. It's closely related to the directivity and efficiency, according to the following expression:

$$G(\phi,\theta) = \frac{\mathcal{P}(\phi,\theta)4\pi r^2}{P_{del}} = \frac{P_{rad}}{P_{del}}\frac{\mathcal{P}(\phi,\theta)4\pi r^2}{P_{rad}} = \eta D(\phi,\theta)$$

From a practical point of view there are two methods to measure the gain: absolute-gain and gain-transfer measurements. Absolute gain takes into account all reflection or mismatch losses.

A really effective way to know the gain of an antenna, is to compare its gain with that of a well-known one. Standard gain horns (SGH) are well-known antennas, that can be measured in an antenna measurement system in order to determine its gain with a huge detailed level: They are typically used as reference antenna in the gain comparison method. In this case the gain of the antenna under test in obtained by comparing output signal levels of the AUT with the signal levels obtained with the SGH. For better accuracy, the comparison has to be corrected with return a cable loss and any other parameter that is relevant for the gain measurement.

1.2.4 EFFICIENCY

This parameter accounts for all power losses at the structure of the antenna caused, for example, by structure problems, mismatch, conduction losses, etc. Its defined as the ratio between the total power radiated by the antenna and the power supplied to its input. It's indicated by η.

1.2.5 POLARIZATION

The polarization of an antenna is the polarization state of the waves it radiates in the direction of its pattern maximum.

In general, all antennas can be considered as having elliptical polarization, being the most usual lineal and circular polarization cases two extreme cases of the elliptic one. For the linear case, the ellipse degenerates into a line (one of the ellipse axes becomes zero). For the circular case, the ellipse degenerates into a circle (both ellipse axes becomes equal).

Considering two orthogonal linear components of the field in the plane transversal to the direction of propagation:

$$\mathcal{E}_x(z,t) = Re[E_{x0}e^{\omega t+kz+\phi_x}] = E_{x0}cos(\omega t + kz + \phi_x)$$
$$\mathcal{E}_y(z,t) = Re[E_{y0}e^{\omega t+kz+\phi_y}] = E_{y0}cos(\omega t + kz + \phi_y)$$

- Linear polarization results when both linear components are in phase or presents a phase difference of 180º.
- Circular polarization results when both components have the same magnitude but presents a phase difference of 90º or 270º. Depending on both the component used as reference for the phase comparison and the phase difference, the circular polarization can be right-hand or clockwise circular or left-hand or counterclockwise polarization.
- Other combinations results in left- or right-hand elliptical polarization.

To maximize the link performance, both transmitter and receiver antennas must be polarization matched.

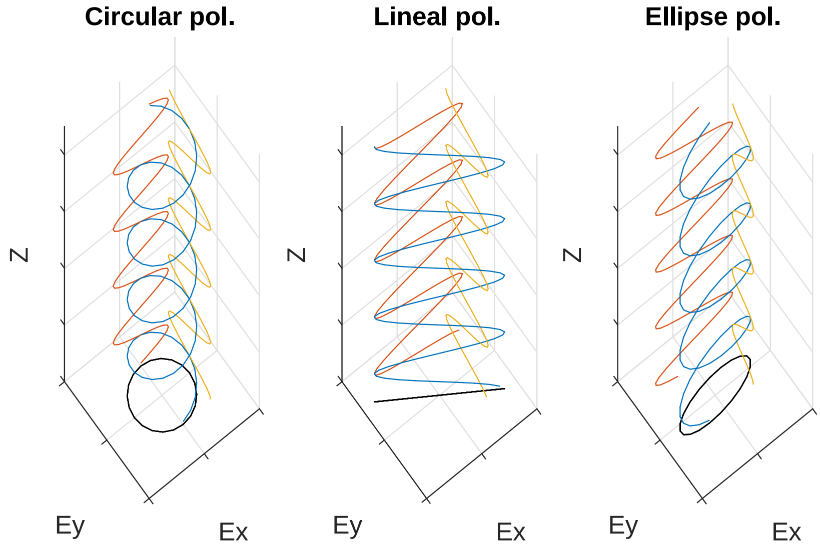

Figure 1.3: Different types of polarization: circular (left), lineal (center), and elliptical (right).

1.3 ANTENNA MEASUREMENTS

Nowadays, powerful commercial full wave simulators (like HFSS or CST) are used in the design process of antennas. These are able to provide really good approaches to the behavior of the structure. However, in order to test the real response of the device after construction, the measurement of the AUT and the checking of its FF behavior is mandatory. Since this work is oriented to Planar Near Field to Far Field processing implementation and the related parameters, the focus is put on this treatment.

FF region, also called Fraunhofer region, is the region where the radiated field behavior is completely satisfied, and it can be obtained from the Fourier Transform of the sources that produce it. It is typically considered to start at four times the Rayleigh distance ($2D^2/\lambda$, where D is the maximum dimension of the antenna). FF region is considered a phase-stable distance.

Near field (NF) or Fresnel region, cover from the FF region limit (Rayleigh distance) to the close or reactive field region limit (depending of the frequency, different criterial are considered, like L= λ or $0.62 * (D^3/\lambda)^{1/2}$), where evanescent fields are predominant.

In order to measure the radiation behavior of an antenna, the receiving antenna must be in the far field of the antenna being characterized. However, experimental considerations involve some problems:

- Distance can be really large for high frequencies.
- Big antennas can't always be moved to measurement facilities
- Some antennas have a measurement time enormous, such a phased array.

- There is no environment control (weather, temperature…)
- These systems are expensive.

Some solutions to those problems are:

- NF techniques.
- Anechoic chambers.
- Scale model measurements.
- Automated test systems.

Antenna measurement systems can be classified in two different categories: outdoor and indoor ranges.

1.3.1 OUTDOOR RANGES

Outdoor ranges allow an easy way to create FF conditions, but present two important disadvantages: the big size of the measurement field required and the uncontrollable environmental conditions.

In general terms, two kinds of outdoor antenna measurement ranges are used: the reflection and the free-space range.

- Reflection ranges take advantage of the constructive interference of direct signal and specular reflections from the ground to obtain a quiet zone in the region were the AUT is located during measurements.
- Free-space ranges designs avoid interference of reflected signals placing both AUT and receiving antenna physically located far away from ground (for example in the top of two mountains) in the case of elevated ranges, or with a receiving antenna designed with a null pointing to the specular reflection point in the case of slat ranges.

1.3.2 INDOOR RANGES

Indoor ranges have no problem with the environmental conditions, but, depending on the size of the antenna to be characterized, FF conditions can't be directly obtained. This implies post-processing of the measurement results with different alternative measurement setups, or require more acquisition tools (like compact antenna test ranges, CATRs).

If the size of the AUT satisfy the FF distance criteria, a direct FF measurement can be done in an anechoic chamber without any additional calculation.

1.3.2.1 ANECHOIC CHAMBERS

The indoor measurements are carried out into an anechoic chamber, that provide a controlled environment as alternative to outdoor testing. They are rooms fully covered with absorbing material that keeps reflected signal levels below a specified level. They are isolated from external environment, so they are not affected by weather conditions, temperature, etc. The biggest problems working in anechoic chambers are the room size (which is limited) and the working frequency, that has to be fixed before chamber's construction (a chamber works in a wide bandwidth) [9].

Figure 1.4: Photo of a radar cross section measurement in a aneechoic chamber. Source: *Eletrical-GReece*

A parameters definition is required to ensure a correct chamber design. Once the build is done, changing the operation specifications is not an option. So we need to define:

- Type of measurements to be test, in order to install the correct instrumentation (for example, a complete mechanical system to take planar and spherical scans).
- Frequency band of operation, that will affect the absorbent material design, system size... A high frequency, like V or W-band, could induce problems with secondary waves.
- Geometry of the chamber: the most extended configuration is the rectangular chamber, but also tapered chambers can be built. In these chapters, just rectangular chambers will be considered. A rectangular chamber needs some design considerations to be taken before starting to build. According to the Rayleigh distance ($2D^2/\lambda$), if the maximum antenna dimension (D) is defined, the chamber's minimum range length R is set. The chamber with (W) and quiet zone diameter (q) are directly related with R through $W>R/2$ and $q=W/3$.

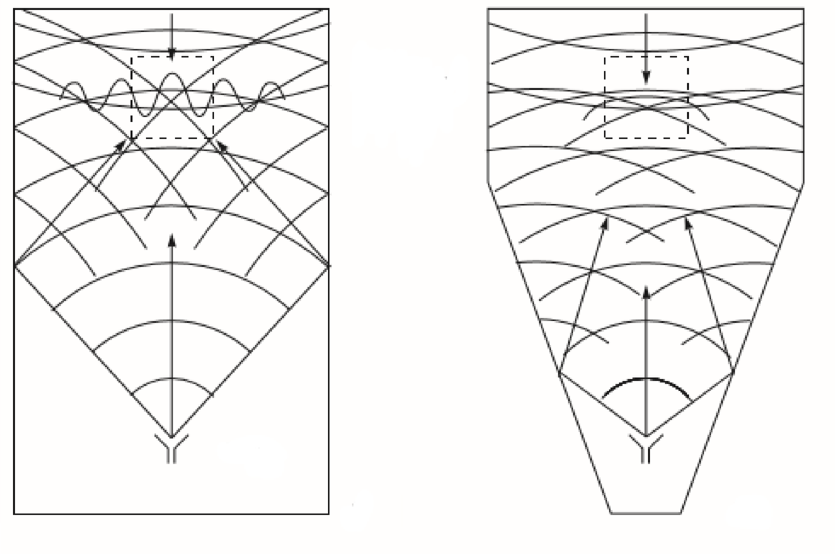

Figure 1.5: Rectangular (left) and tapered (right) anechoic chamber [4].

To design a measuring system, the instrumentation must be:

- Source antenna. Typical antennas would be gain horns, that operate over the common waveguide frequency bands, but the main parameter to set a source antenna is the frequency. Dipoles are used for low frequencies (f¡1GHz). The source antenna has a lineal polarization.

- Transmitting system, that controls the polarization, frequency (center, stability...), power level, modulation, and any other property of the signal. It can be built with a simple oscillator, or with a complete frequency synthesizer. The use of one or another depends on the requirements of the measure.

- Receiving and recording system, that is able to storage and export the data. A VNA (vector network analyzer) can be used to set the magnitude and the phase of the wave. A good receiving system is needed to get correct measures, because it's characteristics are the most influential in quality of the complete system. Recording system affects the speed of data acquisition, because the data must be processed in the right order.

- Positioning system, that must be able to rotate in different planes in order to get a complete radiation pattern in all directions. There is no difference between the AUT rotation or the source antenna rotation (as the same as TX/RX work). The mechanical part is critical in order to take good antenna's scans: the precision of the mobile parts becomes really important when the wavelength is just a few millimeters. Roll over azimuth is the most used configuration to scan the phase of the signal (with the magnitude, too).

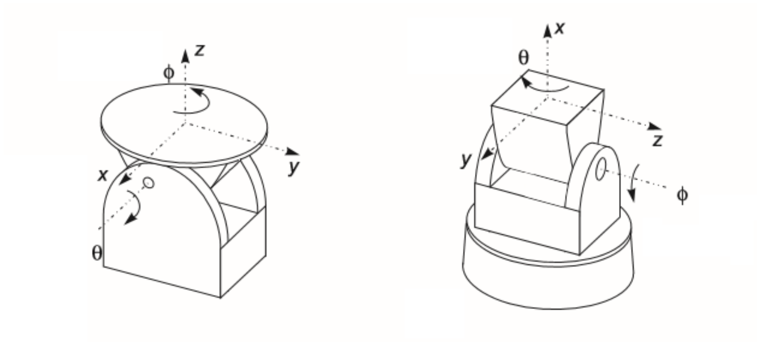

Figure 1.6: Azimuth over roll (left) and roll over azimuth (right) configuration [4]

- AUT/probe rotation is very important, too: a single polarized antenna can be used to get different polarization if the positioning system is able to rotate it (for example, to change from vertical-linear polarization to horizontal-linear polarization).
- The mechanical precision is crucial in order to take a good measurement. For a high frequency work, the wavelength can be just a few centimeters, so a bad regulated axis could compromise all the work.
- Data-processing system, that allows a post-processing after the data acquisition. The processing device (normally, a computer) takes the data from the VNA, and gives an output in a visual-friendly way. The most usual post-processing works are rotation (which makes possible a maximum/minimum signal position's correction) and interpolation (in order to increase the measurement precision, the system can calculate more field points to set the radiation value on a non-measured position).

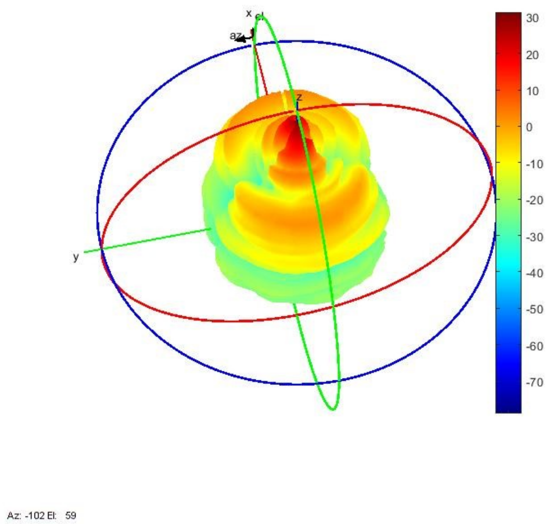

Figure 1.7: Example of radiation pattern representation colored by it magnitude.

1.3.3 NF MEASUREMENTS

The need for precise measurement techniques of the antenna radiation diagram has been linked to the development of high performance antennas and to the improvement of the methods of analysis. In many applications, like space communications, it is necessary to know the gain radiation pattern of the antennas with accuracies of the order of 1%. In other cases, the characteristics of the antenna (electrical size) impose, as the only viable alternative, the measurement in the near field. Furthermore, the criterion usually used for the antenna-probe Rayleigh distance may be insufficient if very precise measurements are required. NF measurements have become the usual measurement method when precise measurements are required, or when the dimensions of the antenna make far-field measurement prohibitive.

The process of obtaining the radiation pattern of the antenna is called NF to FF transformation, and requires later mathematical processing after measuring. Unlike in the FF measurements case, to obtain a single cut of the radiation pattern it is necessary to explore the whole measurement surface with two polarizations, taking information of both modulus and phase. Therefore, in general, the measurement time will be higher for the measurements in the near field, although this inconvenience is partially overcome with the use of rapid measurement instrumentation. On the other hand, the knowledge of the field components in module and phase on a surface near the antenna allows its use in the diagnosis of antennas using inverse scattering algorithms although this aspect is out of the scope of this TFM. [4].

As previously mentioned, to carry out the near-field to far-field transformation, it is necessary to acquire both tangential components of the electric field on the measuring surface. For this reason, the full exploration of the surface has to be carried out two times using in each case a probe sensitive to one of the orthogonal components, or only one time if using a double-polarization probe that allows to acquire the two orthogonal components at each point by electronic switching. Obviously, the latter case takes less time, but requires a high synchronization level.

It is necessary to compensate for the effects of the measurement probe. This process is called probe correction and aims to eliminate the effect of the radiation pattern of the probe on the measurement, and the presence of a certain probe response to the cross polarization. In FF measurements it is usual to use directional antennas as probes in order to improve the dynamic range and reduce the effect of the reflections. In NF measurements it is important that the probe does not present any null, since it could not be corrected. To perform probe correction, it is necessary to have previously characterized its radiation pattern. To do it correctly, it is necessary to locate the probe with the same reference to which it was measured, so that, in general, the whole system of alignment of the antennas is much more critical in the NF measurements than in the FF measurements.

NF provides some advantages:

- NF test allows to control all environment conditions (using an anechoic chamber), avoiding external interference.

- Big antennas can be measured in small environments.
- Data precision is the same or better than FF measurement.
- The AUT can be completely characterized.

Obviously, AUT's performance changes with frequency. All designs have a determined work band, where the antenna radiates with good conditions (gain, directivity…). For this reason, a measurement under working conditions is better than an outside-band test, but there isn't the only way. Low frequency antennas can have a huge size, which complicates their handling, and precision is critical for high frequency systems. Furthermore, FF conditions are harder to achieve with a frequency increase: Rayleigh distance is the minimum distance where the wave has a planar distribution, but it can include some NF components. This could affect the measurement with a factor about 0.05 dB (it depends on the distance, the antenna, and many other parameters). At those conditions, some frequency change techniques are developed, like scaling models and frequency conversion.

First works were designed at X-band (8-12 GHz), a frequency used for radar antennas and other military systems. Nowadays, mobile phone industry is one of the most interested in antenna's measurement, so frequencies have changed. 4G frequencies, like 850 MHz (UHF), is one of the most demanded, because phone operators are trying to cover all places with 4G coverage with their base stations.

The next step, 5G, will be prepared for higher frequencies (V and W-band). These frequency blocks (60, 100 and 110 GHz) imply other issues, like the small wavelength: any distance change could be critical for such a small value (5, 3 and 2'72 mm). Mechanical acquisition system has to handle a very high precision level, in order to avoid measurement errors. Measurement step or antenna pointing technologies are being developed to work almost without margin of error. Laser technology is often incorporated to mechanical system to solve this.

Different antenna measurement facilities are much more suitable for some different kinds of antenna under test. Take in count final use of this antenna will help into measurement processing. Some example of these antennas and its use are:[8]:

- X-band designs uses radar antennas, which means that big measurement systems are needed. Those antennas are typically parabolic reflectors and slotted waveguides, and they are able to rotate (physically or changing the phase of the beam) to cover all scanning area.
- Reflector antennas are used for Kurz bands (Ku, K and Ka). Dish antennas and flat systems are included, with one or more reflectors, like Cassegrain designs.
- Lower bands, like C or S-band, works with LPDA and Yagi. The LPDA consists of a number of driven elements of gradually increasing length. The dipoles are

mounted close together, connected in parallel to the feedline with alternating phase. It simulates a series of two or three-element connected together, each set tuned to a different frequency.

- Standard gain horns (SGH) are used for gain reference for high gain antennas and antenna measurements. They are good known antennas that work perfectly as a reference. In 1.8, we can see (in order, from left to right):a pyramidal horn (with a rectangular cross section) a sectoral E-plane and H-plane horn (often used for radar applications), a conical horn (better for spherical measurements) and a exponential horn (with curved sides).

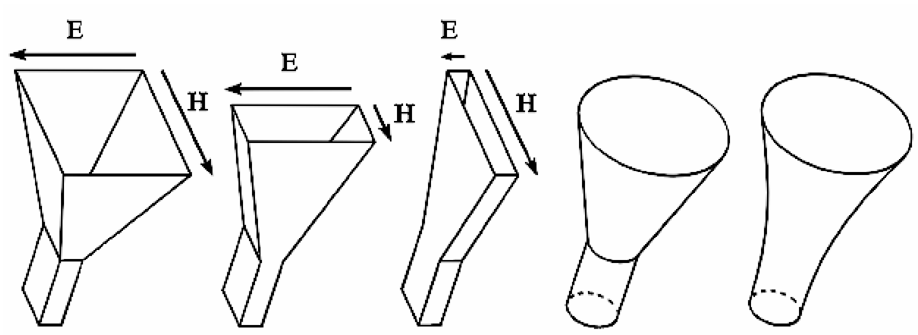

Figure 1.8: Different types of standard gain horns.

- Open-ended waveguide antennas (that works with the same principle that SGH antennas), has physical properties that make them recommended for some measurements. Circular waveguide antennas are often used for spherical NF to FF transformation, as the same that rectangular waveguide antennas for planar NF to FF transformation. These antennas can include filtering structures to avoid the propagation of annoying frequency modes.

Figure 1.9: Open-ended waveguide with a stepped filter.

1.3.3.1 MEASUREMENT TYPES

To emulate a FF response in NF conditions, we have two principal options: CATR measurement and NF to FF transformation.

The CATR (compact antenna test range) is a device that creates a nearly FF planar wave in a short distance. It uses collimated waves from a reflector close to the feeder, that achieve a planar pattern when arrives to the test zone. Different reflector configurations can be implemented in order to achieve different waves at the AUT's place.

The antenna to be measured is placed in front of the reflector, whose size must be large enough to ensure a flat wave over the entire antenna. Even so, the total dimensions of the set are much smaller than those required in a far-field direct measurement. Its practical realization is not simple, since certain important problems, such as the diffraction of fields at the edges of the reflector, or direct radiation from the feeder in the direction of the antenna to be measured, that must be solved. These problems basically give rise to curls in the measuring area. To avoid them, the feeder is usually hidden in conveniently insulated enclosures, which are hidden from the measurement area, and to reduce the diffraction of the edges of the reflector, it is given serrated or curved shapes.

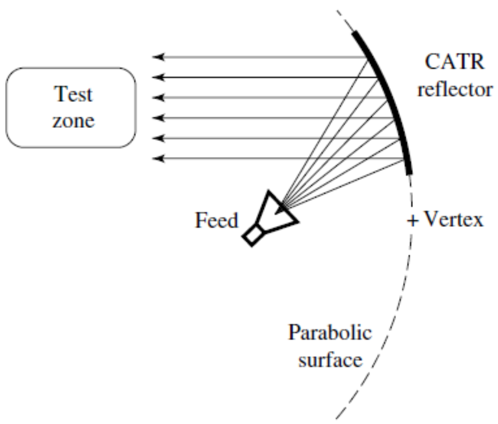

Figure 1.10: Example of how works a reflector: the nf signal rebounds and go to the test zone with a planar front [7].

As alternative to big-space-needed designs, near-field tests transform the magnitude and phase measure to a far-field waves. In order of computing cost, there are planar, cylindrical and spherical systems. All data acquisition begins with a system's sweep (according with the geometry). Planar sweep scans X and Y axes with a line move. Cylindrical sweep covers all ϕ range for each Z axis position. Spherical sweep combines θ and ϕ rotation (normally, an angle is covered between 0 and 180º and the other moves between 0 and 360º). The scanned surface could be anyone, but using one of the three commented geometries makes easily the data transform from NF to FF. Another point of view is to see the problem as a plane wave made with the sum of all close measures scanned.

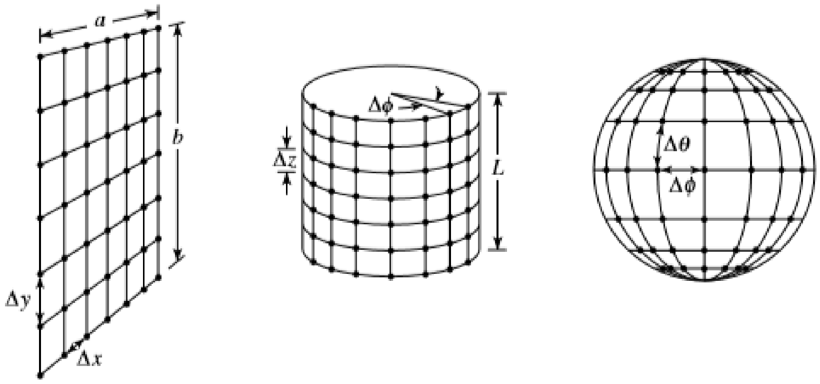

Figure 1.11: Different canonical measurements [8].

Fastest measurement is the continuous sweep measurement scheme. So, the AUT (or the probe, TX and RX behavior are reciprocal) has to be moved around the axes to emulate a planar, cylindrical or spherical distribution. In any interval o position, electrical field components will be measured in NF conditions. Modal-Expansion method can be used to translate NF to FF: waves can be studied as a plane, cylindrical or spherical waves composition, and with the determination of the magnitude and phase of these plane waves, FF wave can be calculated. According to the Equivalence Theorem, the field at any point can be determined from the tangential fields to a surface that enclose all sources. If we calculate a solution for $r=\infty$, we'll have the FF approach.

1.3.3.2 HISTORY OF NF TO FF MEASURES

The development of near-field scanning as a method for measuring antennas can be divided into four periods: experimental period without probe correction (1950-1961), first probe-corrected theories period (1961-1975), the period in which the first theories were put into practice (1965-1975), and the period of technology transfer (1975-1985), in which 50 or more near-field scanners were built throughout the world [8].

Firsts times of those periods, many of the basic techniques for measuring the characteristics of antennas were developed before and during World War II at Bell Telephone Laboratories, R.C.A. Laboratories, and M.I.T. Radiation Laboratory, among others. The introduction of commercial equipment specifically designed for antenna measurements was due, in part, to the large quantities of antenna patterns that were required by programs in the aerospace/defense industry. This period saw the introduction of antenna pattern recorders, a variety of positioners, receivers, pattern integrators, and signal sources. During the same period, the most difficult problems were associated with antenna design and not with elaborate measurements [10].

Beginning with the space program in the 1960's, system requirements, with smaller design margins, started the evolution of measurement techniques. The techniques previously employed became inadequate for the technical problems of the 1960's. This led to the search of new methods to be developed along with a requirement for more sophisticated instrumentation.

Probably, the first NF antenna scanner was the "automatic antenna wave front plotter" built around 1950 in the Air Force Cambridge Research Center, without any attempt to compute FF transformation. They obtained full-size maps of the phase and amplitude variations in front of microwave antennas. Richmond and Tice, in 1955 experimented with air and dielectric-filled, open-ended rectangular waveguide probes for NF measuring of microwave antennas, and compared calculated far fields with directly FF measured. In 1958, Kyle used an open-ended circular waveguide working at X-band to compare its FF measurement with its NF to FF transformation. Gamara (1960) compared computed FF with directly measured fields working at X-band[10].

All those investigations didn't apply any probe correction. In 1961 Brown and Jull gave a rigorous solution to the probe correction problem in two dimensions using cylindrical wave functions. However, it wasn't until 1963 when Kerns reported a rigorous and complete system's correction theory, in order to improve data acquisition for planar NF scanning. It was included three dimensions correction. Years later, in 1973, probe-compensated cylindrical near-field scanning was extended to three dimensions by Leach and Paris. The first probe-corrected NF measurements were in 1965, and it was developed for 10 years for planar and cylindrical case. During that period, acquisition system reached the 60 GHz. The electronic revolution in last years has allowed a huge computation capacity, so the NF measurements are quite developed, especially for antennas that are difficult to test in FF conditions (like satellites systems, array configurations, millimeter antennas...)[8][10].

Also, radiation patterns can't always be measured easily in FF region. To take some accuracy scanning lowest lobes, far distances are needed. Another NF application is to align the beamformers of an array antenna. We must remember that distance will be always electrical distance, so it could be a large physical distance for specific frequencies.

1.3.3.3 SPHERICAL AND CYLINDRICAL SURFACES

The solution of Maxwell's equations in cylindrical geometry includes Bessel functions and exponential functions, which complicates mathematical formulation with respect to planar geometry. On the other hand, the measurement surface involves much more to the AUT, so the errors due to the truncation are smaller.

These errors disappear in the spherical geometry, since it totally surrounds the antenna, and the fields found are completely accurate. In the mathematical formulation appear Hankel functions and Legendre polynomials, which make it the most complicated of the three formulations. However, with the great advance that has occurred in the capacity of memory and the speed of calculation of the computers, the mathematical problems are smaller.

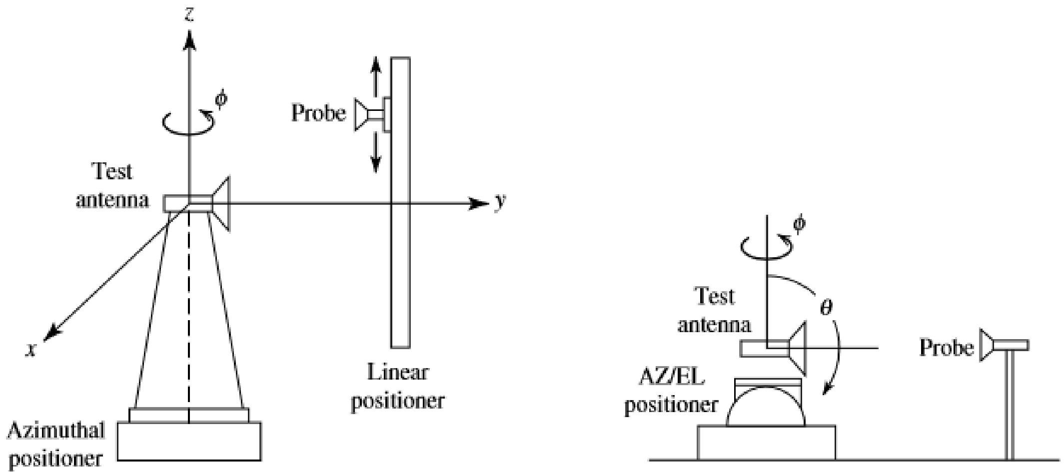

Figure 1.12: Cylindrical and spherical scheme of measurement.

1.3.3.4 PLANAR SURFACE

This is the easiest computational case. We assume the electromagnetic field for a rectangular grid close to the AUT as known, and then we will make the NF to FF transformation. This mathematical operation is based on the plane wave expansion using Fourier transform techniques. A wave can be formed as a superposition of plane waves in different directions, all of the same frequency. If we can determine the unknown directions and amplitudes of those waves, FF will be calculated [6] [7]. In the next chapters, the best performance conditions to measure AUT pattern will be discussed.

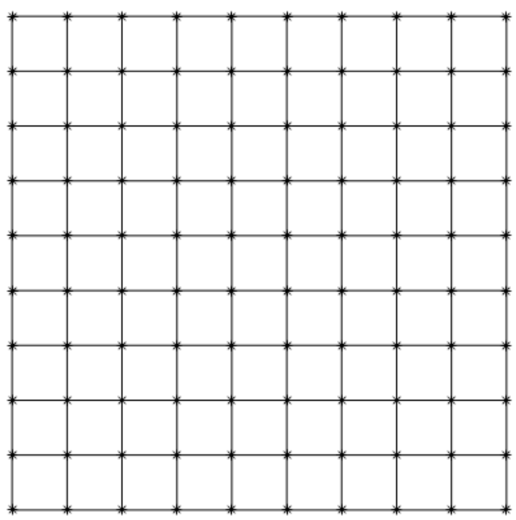

Figure 1.13: Planar grid. The measurement can be done by rows, columns, or meander movement. Each small square has a dimension of $\Delta x = \lambda/2$ and $\Delta y = \lambda/2$

In practice, the planar surface is not infinite, so some considerations have to be taken to assure a correct Fourier application. To avoid a pattern distortion, an angle limitation must be set:

$$tg\theta_m = \frac{L - D}{2z_0}$$

where L is the dimension of the measured zone. Of course, as any other Fourier transform, Nyquist theorem must be respected. The x, y and z dimension have to be transform in polar condition into

$$k_x = ksen\theta cos\phi, \quad k_y = ksen\theta sen\phi$$

Applying Nyquist:

$$\Delta x = \frac{2\pi}{2k_{x,max}} = \frac{\lambda}{2}, \quad \Delta y = \frac{2\pi}{2k_{y,max}} = \frac{\lambda}{2}$$

Calculating the power modes, the performance in any situation (distance, position...) can be set using the FFT. Once the FF is calculated, the FFT operation can be used to adjust the data with more precision (interpolate data, applying rotations...). Power modes offered the real interaction of the AUT's pattern, so if we have a high precision measurement (with the enough number of modes), the electromagnetic field form can be obtained with a high accuracy level.

1.3.4 OBJECTIVES AND DOCUMENT STRUCTURE

In this project, an tool for the near-field (NF) to far-field (FF) transform is going to be developed. Moreover, other useful processes are implemented, too, like field rotation and interpolation, to adjust the signal to desired conditions. Each calculation will be set in a independent function, to allow the program be used in a recursive way.

The object of the project is set a "simple" program able to calculate all the basic parameters of a NF to FF transformation, without a complex calculation environment installed. The goal is program an accurate and fast tool able to perform the correct calculation.

After comment about the antenna's measurement, Chapter 2 will explain how the transformation is done (equations and parameters) in a theoretical way. This approach is programmed using MATLAB. Chapter 3 shows the same algorithm in a compiled language (C#), which is used to create a executable able to perform all calculations. Results and comparison between MATLAB and C# programs are commented in Chapter 4 (according to time performance and data accuracy). Chapter 5 shows the future improvement we can develop for our tool.

Chapter 2

THEORETICAL IMPLEMENTATION

The mathematical process to transform a NF signal into a FF pattern needs to be tested before the final implementation step. Furthermore, most of the programming languages are not prepared for a complex mathematical operation, so a good mathematical reference is really helpful. To check the correctness of the transformation, MATLAB will be used. On the other hand, different corrections and options are needed (rotation, interpolation...) to properly process the data that also need to be included into final software. These will be also initially developed and debugged in MATLAB. In this chapter, the mathematical process will be discussed- with a theoretical explanation of the separated operations that the final program has to execute.

2.1 MATLAB

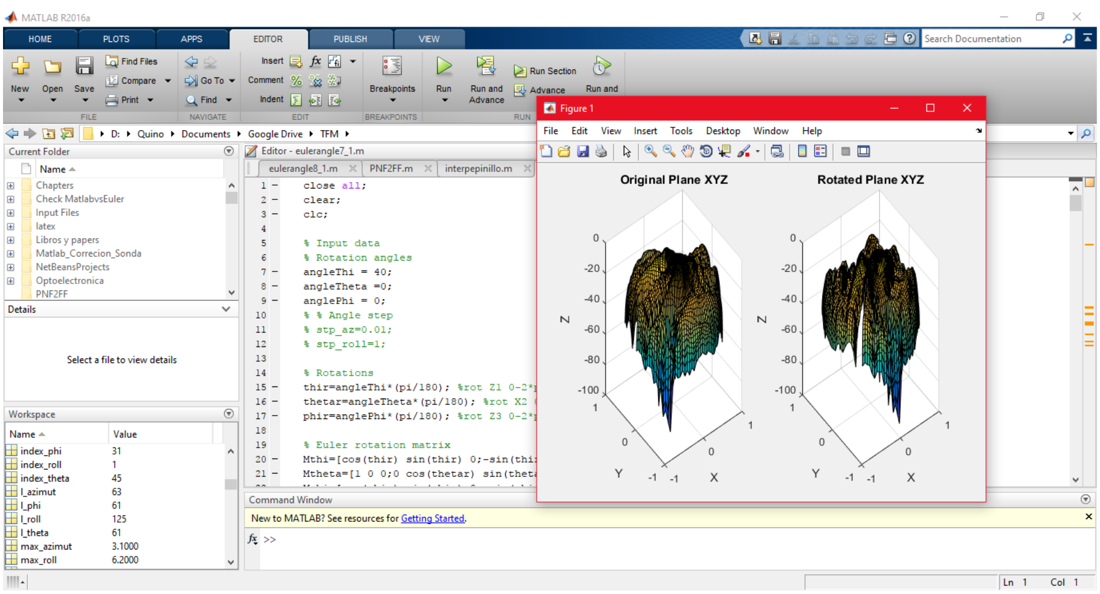

Figure 2.1: MATLAB window.

MATLAB is one of the most used software by engineers and scientist, basically because it includes many functions, toolboxes and libraries very helpful with complex calculations.

The easiest way to work with radiated electromagnetic field data is using matrix: ordering data with a 3-coordinate system (x-y-z) and computing data with MATLAB matrix operations. As is commented in Chapter 1, Fourier transforms and interpolation are needed, and MATLAB can easily work with them. Of course, matrix can be used in simple operations, but with different and useful options, too (for example: A*A or A.*A).

Matrix and vector operations can be set to avoid loop functions in MATLAB and that will decrease the computational cost. But these options are not available in all programming languages, so some functions are differently implemented, to compare results in a truthful way. For example, a double interpolation in MATLAB (interp2) will be set with two blocks of normal interpolation in a loop. That prevents algorithm errors too: some operations can be implemented with variants, so if the program is written in a simpler way, controlling all the steps will be easier.

The main use of MATLAB is based in mathematical block: linear algebra, interpolation and Fourier analysis (based on FFTW algorithm). In order to compute some specific calculations and functions, Communication Toolbox (with many signal processing tools) and Antenna Toolbox (with many graphical options) are implemented, too.

2.2 FAR-FIELD TRANSFORMATION

The main block developed in this TFM is the NF to FF transformation. As discussed in Chapter 1, antenna's measurement in FF conditions are not always available for many reasons (a difficult experimental setup needed, very long measurement distance, weather conditions…), so the NF measurement is a solution for the engineers. The complexity of the process is in the mathematical calculation of the FF: if the real behavior of an AUT under FF conditions is required, the transformation program must actually provide a very good approach to the FF of the antenna.

2.2.1 FORMULAS

The mathematics behind the transformation belongs to spectral processing techniques. We are going to consider a rectangular aperture of dimension a and b. The aperture is located in an infinite ground plane.

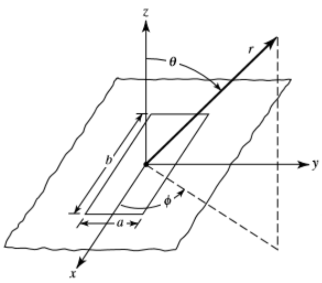

Figure 2.2: Rectangular aperture mounted on infinite ground planes [6].

The radiated wave can be calculated as a superposition of plane waves. These waves are all of the same frequency, traveling in different directions and with different amplitudes.

We can define the radiated electric field as:

$$E(x, y, z) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(k_x, k_y) e^{-jk*r} dk_x dk_y$$

where

- **f** is the plane wave spectrum function,
- kx and ky are the spectral frequencies,
- r is the direction of propagation of the plane wave,
- k is the propagation factor or vector wavenumber.

As we have said before, measurement plane is defined in X-Y coordinates, but there is a Z component that is directed related with the distance between the wave and the aperture. So, separating the coordinates, the field can be written as

$$E(x, y, z) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{E}(k_x, k_y, z) e^{-j(k_x x + k_y y)} dk_x dk_y$$

with

$$\mathcal{E}(k_x, k_y, z = 0) = \mathbf{f}(k_x, k_y)$$

.

Applying Fourier inverse transform we get

$$\mathcal{E}(k_x, k_y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x, y, z) e^{+j(k_x x + k_y y)} dx dy$$

That means that we can set $f(k_x, k_y)$ using the tangential components of the E-field at z=0

$$f_x(k_x, k_y) = \int_{-a/2}^{a/2} \int_{-b/2}^{b/2} E_{xa}(x, y, z = 0) e^{j(k_x x' + k_y y')} dx' dy'$$

$$f_y(k_x,k_y) = \int_{-a/2}^{a/2}\int_{-b/2}^{b/2} E_{ya}(x,y,z=0)e^{j(k_x x'+k_y y')}dx'dy'$$

with Exa and Eya as the tangential components of the electric field over the aperture.

The FF radiation is

$$E(\phi,\theta,r) \simeq j\frac{ke^{-jkr}}{2r\pi}[cos\theta\mathbf{f}(k_x,k_y)]$$
$$E_\theta(\phi,\theta,r) \simeq j\frac{ke^{-jkr}}{2r\pi}(f_x cos\phi + f_y sin\phi)$$
$$E_\phi(\phi,\theta,r) \simeq j\frac{ke^{-jkr}}{2r\pi}cos\theta(-f_x sin\phi + f_y cos\phi)$$

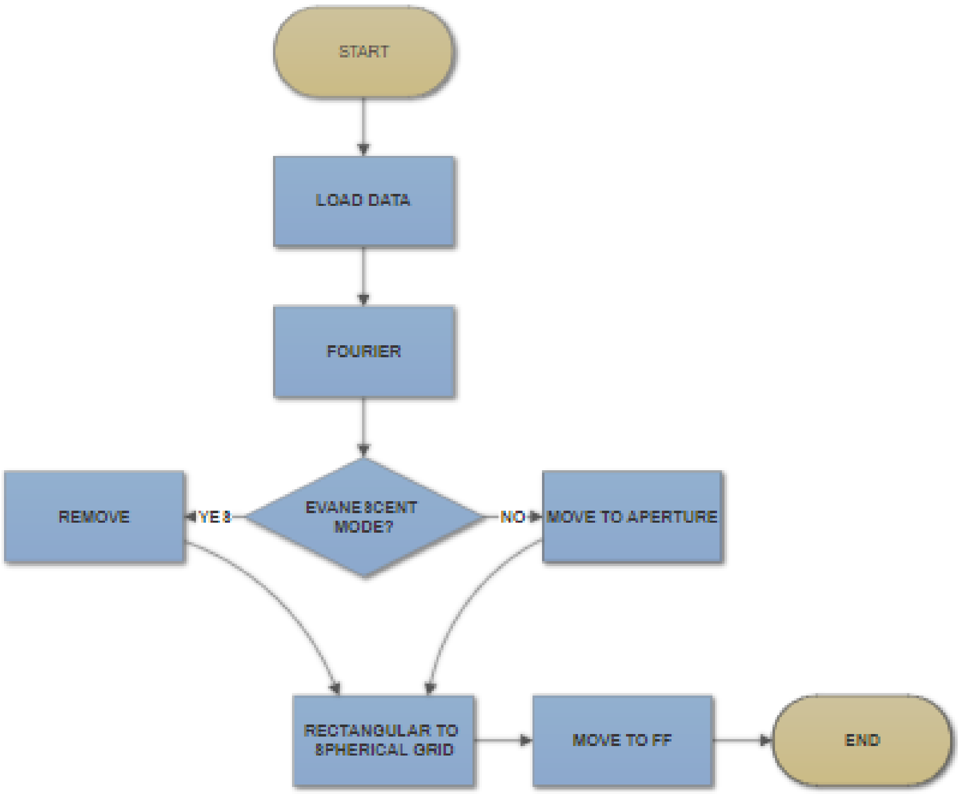

Figure 2.3: NF to FF algorithm flowchart.

2.2.2 IMPLEMENTATION

First of all, field data which comes from a real measurement system will be import to MATLAB (data could be in a .cvs file,. mat format, etc). Field measurement at each point will be stored in a matrix, as well other important parameters, like frequency, number of points on each axis and others need to be stored too. A relevant data is the output domain: NF measurement is test in X-Y plane, but FF output will be in phi-theta domain. All parameters must be imported from measurement conditions to ensure a correct calculation.

The first step is set the grid. All variables that contain information about one point, will be easily stored in a matrix. The coordinate (x=... or y=...) gives us no information to make the FF transformation, but the spectral frequency (k_x and k_y) at each point does. Furthermore, kz is easily calculated by

$$k_z^2 = k^2 - (k_x^2 + k_y^2)$$

Applying Fourier inverse transform (to get f_x and f_y, as we set before), and

$$f_z = -\frac{f_x k_x + f_y k_y}{k_z}$$

In this point we have now three matrix f_x, f_y and f_z, with the amplitude components of the wave. After that, we need to define the radiating field composition. A loop will check kz grid to set if the mode associated to that point is radiated or evanescent: if it is evanescent (k_z is not real), field will be set to 0. All the field is moved to the aperture to obtain the FF field: FF signal is proportional to NF field at the aperture. After the cleaning of the rectangular modes, wave will be interpolated into a semi-spherical grid. MATLAB allows to calculate a double interpolation for a matrix within the domain data. The input contains the k_x and k_yy grid for each field data, the field data (f_x, f_y and f_z), and the projection of the output phi-theta domain into a rectangular grid. Projection follows the spherical to cartesian coordinates transform:

$$x = rsin\theta cos\phi \quad y = rsin\theta sin\phi$$

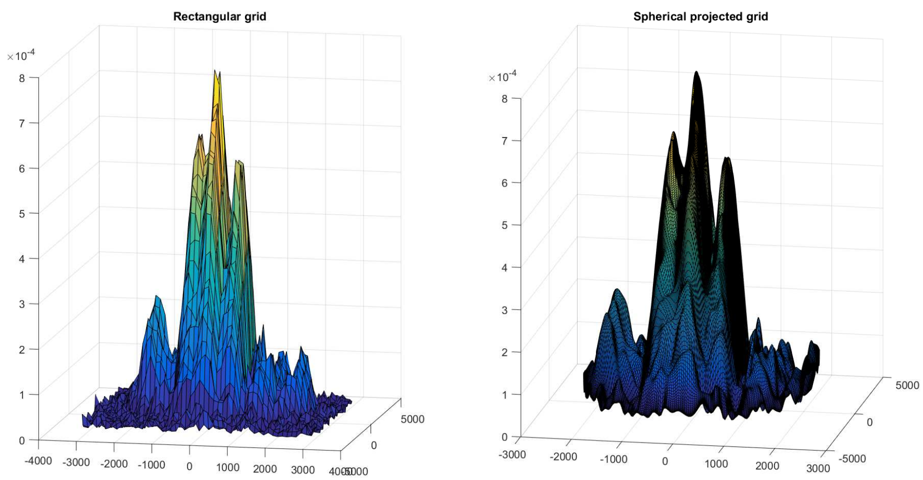

Figure 2.4: Radiated field on a rectangular grid and its interpolation over a spherical projected grid.

To finish the FF transformation, we will apply equation for both polarization: where r contains a long distance (r > Rayleigh distance = $2D^2/\lambda$) in the FF domain. Eϕ and Eθ are easily calculated with phi and theta grid calculated for the rectangular-into-semispherical interpolation. These last operations put the field into a FF distance, so the value of r will affect directly the magnitude and phase of the signal.Theoretically, this distance should be defined but we will use at least several times Far Field distance criteria defined previously.

$$E_\theta(\phi,\theta,r) \simeq j\frac{ke^{-jkr}}{2r\pi}(f_x cos\phi + f_y sin\phi)$$
$$E_\phi(\phi,\theta,r) \simeq j\frac{ke^{-jkr}}{2r\pi}cos\theta(-f_x sin\phi + f_y cos\phi)$$

2.3 FIELD ROTATION

We can assume that all details about FF signal are stable for any distance, so a field change applied under FF conditions, like a rotation, won't degrade the signal information. If we want to rotate the signal, we can use Euler angles processing. Rotation is useful to set a determined point into an angle position to offset a measurement error or prepare the signal for a specific process.

We consider two rotations:

- Euler rotation: A spatial rotation applying Euler angles. This calculation is done by the program to set the final position of the field.
- Shifting rotation: After the Euler rotation, the original position and final position are compared to express the movement by a ϕ and θ change (roll and azimuth). This rotation is the one that is applied to obtain the output.

2.3.1 FORMULAS

The Euler angles are three angles that describe the orientation of a body with respect to a fixed coordinate system. Any orientation can be achieved by composing three elemental rotations, typically denoted as ϕ, θ and ψ. Axes can be repeated, but not twice in a row (it would be just a rotation, not two). There are twelve different options to apply the rotation (xyx, xzx, yxy, yzy, etc). We will use the zxz rotation.

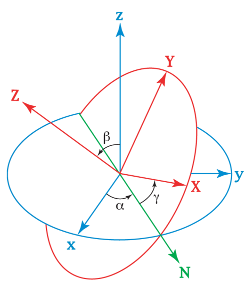

Figure 2.5: Euler rotation scheme. The xyz system is shown in blue and the rotated axes, XYZ, system is shown in red. Source: *Lionel Brits.*

The first field turn is done in z axis (0-360º). Then, we have a new rotated axis (x2, y2, z2=z), so the second turn will be applied to these axes. After the x2 rotation (0-180º), the last one will be on the z3 axis (0-360º) (Fig. 2.6).

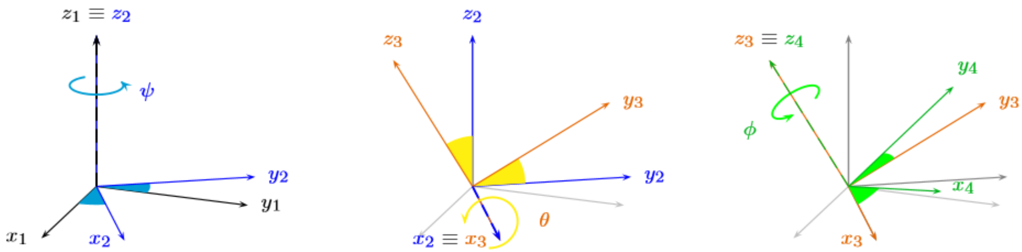

Figure 2.6: Euler rotation movements. By order (from left to right):Precession, nutation, and intrinsic rotation. Source: *UPM*.

Each rotation is represented by a rotation matrix (3x3), that can be assembled into a complete rotation matrix. If we call R to the three-dimensional rotation matrix representing the coordinate transformation from the fixed system to the mobile system, Euler's theorem on three-dimensional rotations states that there exists a unique decomposition in terms of Euler's three angles:

$$R = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} cos\phi & -sin\phi & 0 \\ -sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In order: z rotation (ψ), x2 rotation (θ) and z3 rotation (ϕ) (mathematical representation of Fig. 2.6).

2.3.2 IMPLEMENTATION

MATLAB's matrix management makes really easy the field rotation in XYZ domain: just multiply each point coordinates by R:

$$[px \quad py \quad pz]R = [p2x \quad p2y \quad p2z]$$

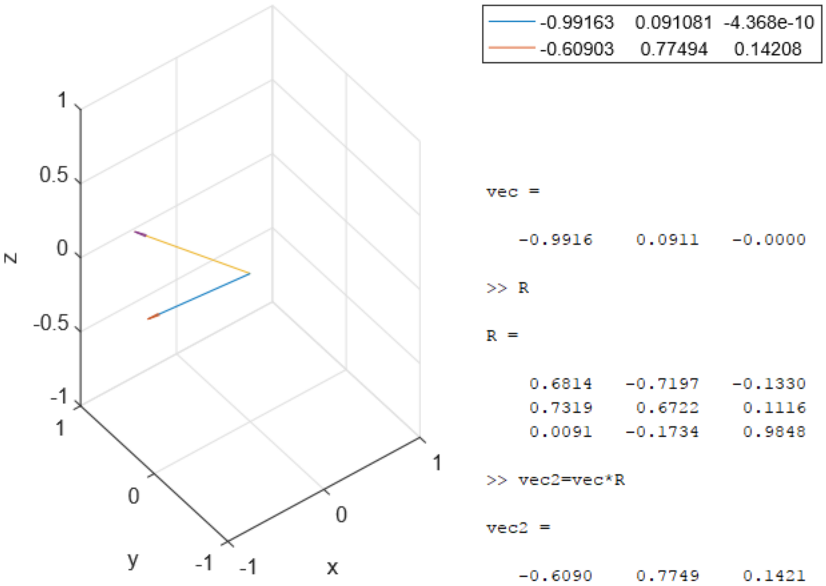

Figure 2.7: Example of vector rotation in MATLAB with ψ=-50º, θ=10 and ϕ=3º.

But, as we have discussed before, the rotation is set to change the field position into its angle distribution, not in xyz domain. That will be really helpful to correct the draws of azimuth and roll cuts. That will be really helpful to correct the picture of azimuth and roll cuts. If we think about the FF like a sphere, Euler rotation makes the sphere turns around (Fig. 2.8). If we measure its position change, we will be able to change the roll and azimuth field distribution.

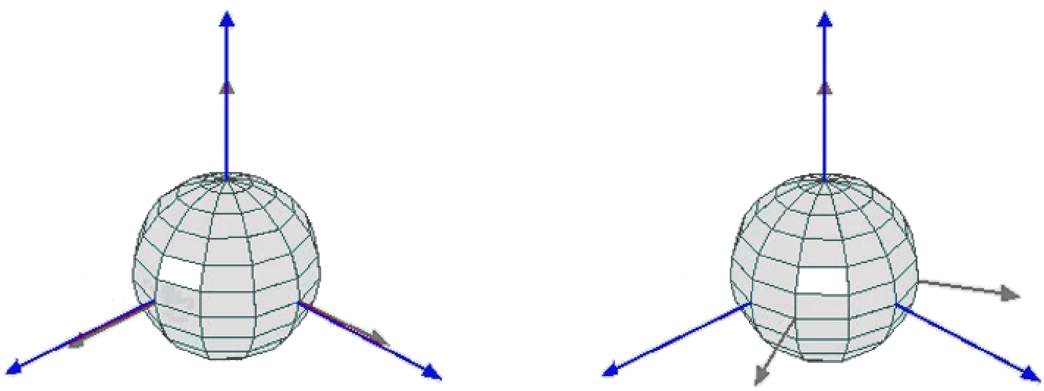

Figure 2.8: Rotation scheme of a sphere.

The FF is unrolled in roll and azimuth domain, so Euler rotation will turn into a shift. To measure this change, a xyz into roll and azimuth transformation is needed, so first of all we will use a reference to get the gap. The reference vector will be x=1, y=2, z=3 (each dimension must be different to avoid any swap between coordinates with the same value). Taking the azimuth and roll with no Euler rotation as a fixed value, we will compare these parameters with the reference R angles.

To assure a more detailed solution, it's recommended to work in a complete roll and azimuth domain. The signal could have been measured in a small range, but we will fill it with zero until the end. Main reason is the 2D-Fourier transform: as in the picture treatment, data can be easily shifted using an exponential factor with the desirable change.

$$F[k,l] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-j2\pi(\frac{k}{M}m + \frac{l}{N}n)}$$

Before this processing, Fourier transform, the input is interpolated into a complete roll (0:360º-step) and azimuth (0:180º) grid. Furthermore, this field is copied to add a small margin (the complete domain is cyclic, so signal repeats itself with a 360º-step roll period and 180º azimuth period).

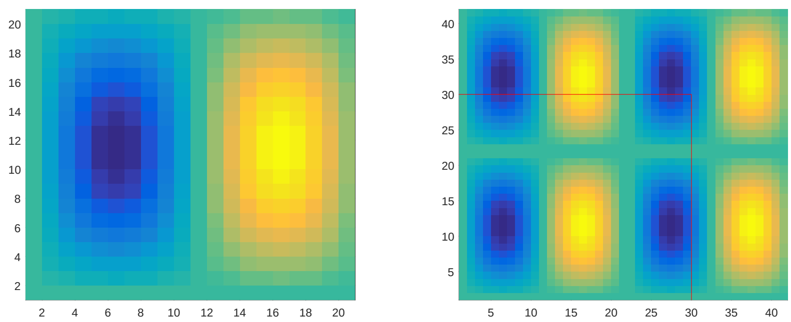

Figure 2.9: Example of the addition of a small margin (red square on the right picture) in a copied field. The left figure shows the original field.

Once the signal is shifted and inverse-transformed by Fourier, different values from zero can appear: the interaction of the modes can set different values after the shifting process. Finally, signal can be extracted into the input domain to compare the rotated values.

2.4 INTERPOLATION

The interpolation block is a processing based in linear 2D interpolation. In many cases, you need a more accurate analysis of the signal, but the measurement didn't have enough angular resolution due to different reasons (mechanical limits, measure time, for example). A linear interpolation allows to take a determined point at any wanted interval.

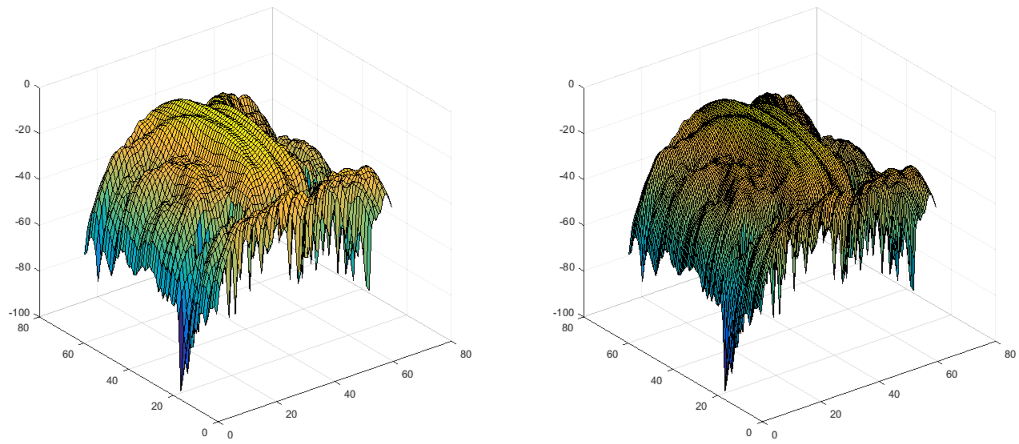

Figure 2.10: Original field (left) and interpolated field to increase the data resolution (right).

2.4.1 FORMULAS

The linear interpolation is a method used for curve fitting that obtains new data points within the range of a known data points. It approximates the value of an unknown point using a straight-line distribution between two known points. It's a particular case of Newtons interpolation, where the polynomial is n = 1.

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x_0) + y_1(x - x_0)}{x_1 - x_0}$$

2.4.2 IMPLEMENTATION

MATLAB 2D implementation needs to manage square matrix, so to avoid this limitation, the interpolation block is programmed as two-times 1D interpolation. The first loop will check the column values, and the second loop will do the same in the other direction (rows). The order between the direction doesn't matter.

If we start with a MxN matrix, after the first loop we will get a MxN2 (N2 > N) matrix. The final loop will get us a M2xN2 (M2 > M) matrix. That makes an easy implementation just with a normal interpolation algorithm.

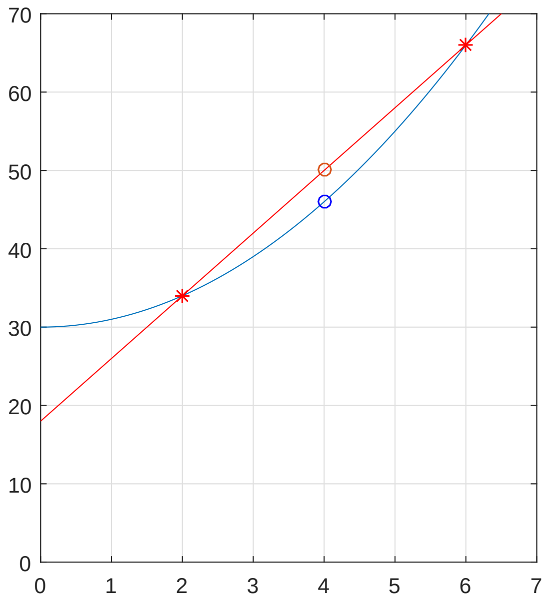

Figure 2.11: Linear interpolation. The original function (blue line) is approximated (red circle) using a straight line (red line) between two known points (red stars) to calculate an unknown point (blue circle).

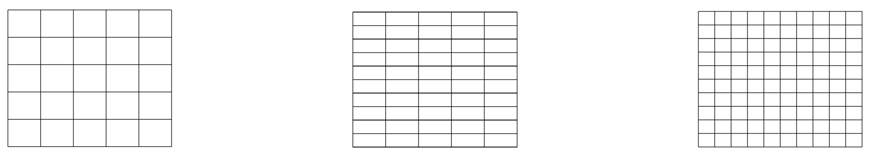

Figure 2.12: Scheme of how the interpolation increase the dimension of the field in two steps (first: rows, second: columns). By order (from left to right): Original dimension, dimension after the first interpolation and dimension after two interpolations.

Chapter 3

PRACTICAL IMPLEMENTATION

C# is an object-oriented programming language developed and standardized by Microsoft as part of its .NET platform. This language describes a virtual environment for application execution, whose main feature is to allow applications written in different high-level languages can then run on multiple hardware and software platforms without the need to rewrite or recompile their source code. Its basic syntax derives from C / C ++ and uses the object model of the .NET platform, similar to Java. The program has been developed using Microsoft Visual Studio.

C# is a compiled language, which means that its source code, written in a high-level language, is translated by a compiler into an executable file understandable for the machine on a certain platform, which reduce the execution time. This file can run be run as many times as necessary without having to repeat the process so the time between execution and execution is very small. That's the reason why the Planar FF transformation is programmed in C#: it's faster and reusable.

3.1 MICROSOFT VISUAL STUDIO

Microsoft Visual Studio (MSV) is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. It can produce both native code and managed code. Visual Studio supports 36 different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C#, C, C++ and C++/CLI, F# and TypeScript. Support for other languages such as Ruby, Python, Node.js, and M among others is available via language services installed separately. It also supports HTML/XHTML, XML/XSLT, JavaScript and CSS.

Using MVS allows not only to develop the final software, but also to debug and modify it. The modular structure designed also allows to call just a part of the code to probe it. It will be really helpful to develop properly the software. The different modules of the program will be explained in the next paragraphs.

3.2 MATH.NET

An advantage of MATLAB becomes a problem in C#: matrix management. While MATLAB has a full working capacity with matrix, C# has a basic matrix management, with limited options. Furthermore, mathematical complex process, like a Fourier transformation, are quiet simple working with MATLAB.

To get over the limitations that basic C# offers, Math.NET library and Math.NET Numerics module are used. Math.NET Numerics aims to provide methods and algorithms for complex numerical computations. Covered topics include special functions, linear algebra, probability models, random numbers, interpolation, integration, regression, optimization problems and more. It is license free, so is not limited for commercial purposes.

Math.NET allows to work with a like-MATLAB experience, so Fourier transform, interpolation, and matrix work are covered. Another small-functions are programmed to ease the work.

3.3 STRUCTURE

3.3.1 INPUT DATA

The main program has to read the input-field data and apply any of the operation showed at MATLAB's chapter. But, first of all, data must be correctly loaded. Two files will be read: a .DAT file, which contains field data, and a .PINP (planar-input) file, which contains some data for the calculation (angle domain, signal step, output paths. . .). Those files are in a known location, so the program can start without any other external data.

Knowing the file format, data can be loaded in a fixed way. All files (input and output) have a header, with some information about the file or the process: time of execution, name of the operator. . . . The format must be flexible enough to be used for all calculation, no matter what (format can't produce any error in the code).

The .DAT file (Fig.3.1) has four columns: two of them are for one polarization, and the other two are for the second polarization. Each row contains the real and the imaginary field component at each point. Both polarizations are loaded into two matrices, so data management can be quite similar to MATLAB. To assure that the recursion program works (1 call = 1 calculation), output data must have the same file format. The header contains some information about the antenna (that is not relevant to the calculation) and 6 vars: number of points of each polarization and the range of the measured axes.

```
Planar Near Field Data derived from file: prueba
File created by ASYSOFT on _01-09-2017_12-36-21_PM
NX =     1334, NY =     1334
X-range :     -1 to  1
Y-range :     -1 to  1

3.225729e-05      -1.421806e-04   6.206591e-04    1.093786e-05
-6.348517e-05     -2.739626e-04   5.771918e-04    4.318853e-04
4.155694e-04      6.389777e-04    2.583355e-04    2.648238e-05
```

Figure 3.1: Example of a .DAT file.

The .PINP file (Fig. 3.2) has many parameters (numbers and characters) followed by its name. To load all data, a function analyzes each line, look for the parameter name, and load the value into a specific variable. If the line has two or more parameters, they are ordered as VALUE1, VALUE 2, NAME1, NAME2. So, function look for the position, too.

```
AUT: t1_005000
PNF2FF                  COEF
90.0                DIST
5.0             FREC
21 61           NX, NY
1.0             MODEXP
1                       NYMAXI
0.0                     ROUT
-25.0   25.0   -75.0   75.0         XI, XE, YI, YE
-89.0   89.0   1.0   179.0          THEIOU, THEEOU, PHIIOU, PHIEOU
2.5   2.5           THETASTEP, PHISTEP
179.0   179.0           NTHETA, NPHI
1 2 2 X     BIAS_CORR, N_CORR, M_CORR, F_CORR
yes yes           PWNORM, CJGDAT
C:\TMPAsysoft\t1_005000.OUT PRINTER OUTPUT
C:\TMPAsysoft\t1_005000_NF.DAT  INPUT FIELD
C:\TMPAsysoft\HERTZ.IPR single  INPUT PROBE
C:\TMPAsysoft\t1_005000_FF.DAT  OUTPUT FIELD
```

Figure 3.2: Example of a .PINP file for NF to FF transformation.

Different calculations have different .PINP file, but the structure is the same for all types. In the picture (Fig. 3.2) a sample file is shown:

- COEF: a code that indicates what calculation has to be done.
- DIST: distance (in cm) between the probe and the AUT.
- FREC: frequency (in MHz).
- NX and NY: number of measured points in each dimension.
- MODEXP: indicates the frequency interpolation factor, that is applied to the FFT/IFFT.
- NYMAXI: a probe correction parameter.
- ROUT: distance of FF calculated

- XI, XE, YI and YE: axes limits.
- THEIOU, THEEOU, PHIIOU and PHIEOU: output ϕ and θ limits.
- THETASTEP and PHISTEP: output ϕ and θ steps.
- NTHETA and NPHI: number of calculated points in each dimension.
- BIASCORR, NCORR, MCORR and FCORR: noise correction parameters.
- PWNORM and CJGDAT: data management parameters .
- Directions: selected locations for input field, output data, output field and probe field.

All output variables from .DAT and .PINP are defined as public, to allow their access from another function. The calculation modules are just like the ones of MATLAB, but programmed with C# characteristics.

The calculation modules are just like the ones of MATLAB, but programmed with C# characteristics.

3.3.2 OUTPUT DATA

The program has three output files after each execution: ouput.DAT, output.OUT and output.LOG.

Output data file (ouput.DAT) has the same structure that input (Fig. 3.1). The only change we can find is the range of the measurement: FF data are calculated in a $\phi\theta$ area, while NF measures are obtained in a rectangular grid (XY).

Log file works like any other program log: it records all the important events to check (errors, most of the time). If there is a program exception, .LOG file gets a new line with the function that has the problem. This is possible using the try-catch block: all the code lines after the try statement are checked. If there is any error (overflow, format exception…), log function writes a simple message to know where the problem is. The rest of the program must continue without showing any problem (obviously, output data won't be correct). Log data can be shown by the command windows instead of creating another file.

.OUT file shows calculation data in a human-friendly way, which allows the operator to know more about the process. It contains information about the power of the signal, the number of point calculated, etc.

3.4 FAR-FIELD TRANSFORMATION

If the input field is a NF signal, the calculation will give the FF transformation. The program will take both .DAT matrix as input, like all the .PINP parameters. These values include the frequency, input domain (XY) and output domain ($\phi\theta$).

The mathematical process is just the same as the MATLAB program. First of all, some vectors with input domain information are calculated. MATLAB has a perfect method implemented to work with vector and matrix, but C# don't, so the solution is using a loop (two loops in case of a matrix) to complete all positions. The calculated vectors have the information about the number of points in each direction (XY). The total number of points can be different from the number of acquisitions if an interpolation value is selected: the Fourier transform can be programmed with a zero padding, to increase signal resolution, so more grid points are needed.

Those input points are used to get spectral frequency (k_x, k_y and k_z) vectors, because the amplitude of the wave depends on the spectral frequency values. Setting kx, ky and kz grid, we can calculate the amplitude of the wave for all the spatial points we have. To ease the matrix calculation, a small meshgrid function is implemented in a "like-MATLAB" way.

After the input data calculation, let's set the output domain in a matrix view. The same method is used: we put output domain values in two vectors ($\phi\theta$), and we use those vectors as input for the meshgrid function.

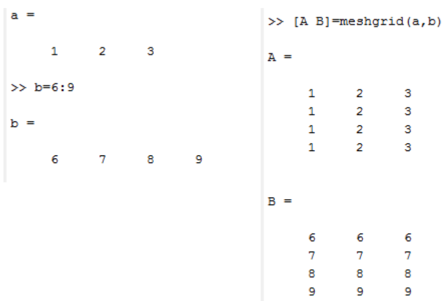

Figure 3.3: Meshgrid function in MATLAB. Input vector are on the left, and output matrices are on the right.

The next step is to apply the Fourier inverse-transform to the near field data. As discussed before, the radiated wave of a rectangular aperture is the Fourier inverse-transform between the aperture limits. We have used Math.NET library to obtain fx and fy applying Fourier, but not directly. Math.NET give us the option to set a vector Fourier transform in a MATLAB way, but we need a 2D Fourier transformation. In other words, our calculation is applied to a matrix, not a single vector.

To avoid this problem, we have programmed a 2D Fourier function using the single one: 2D Fourier transform can be decomposed as two Fourier modules. First, we apply Fourier transformation to each row, and then, we apply the same operation to each column. Complex data management is implemented, so there is no any other issue. After the transformation, we apply data shifting, to order the field, like MATLAB ifftshift function. Amplitude of the wave along Z axis (fz) is directly calculated from fx, fy and kz grid.

After that, filtering of the evanescent modes is needed: an evanescent wave is an oscillating field that does not propagate as an electromagnetic wave, but whose energy is spatially concentrated in the vicinity of the source, so it won't be radiated. To filter the signal, we will check the kz grid, because if the vector has imaginary components, it may have a magnitude that is less than its real components. Maxwell's equations in a dielectric medium impose a boundary condition of continuity for the components of the fields, and it is only confirmed with evanescent modes. If there is a complex value, this position will be an evanescent mode, so we will turn fx, fy and fz into a zero.

Now, we need to change the signal into the output domain. We have the $\phi\theta$ grid, but we need to project it into the XY domain.

$$x = rsin\theta cos\phi \quad y = rsin\theta sin\phi \quad z = rcos\theta$$

With the grids in the same domain, we can interpolate the signal from the XY to the $\phi\theta$ projected one. But C# has no a MATLAB interpolation function, and Math.NET doesn't provide a good 2D interpolation into different distributions. The reason is that XY domain has a rectangular distribution, but the $\phi\theta$ projected grid follows a concentric circles way (Fig. 3.4).

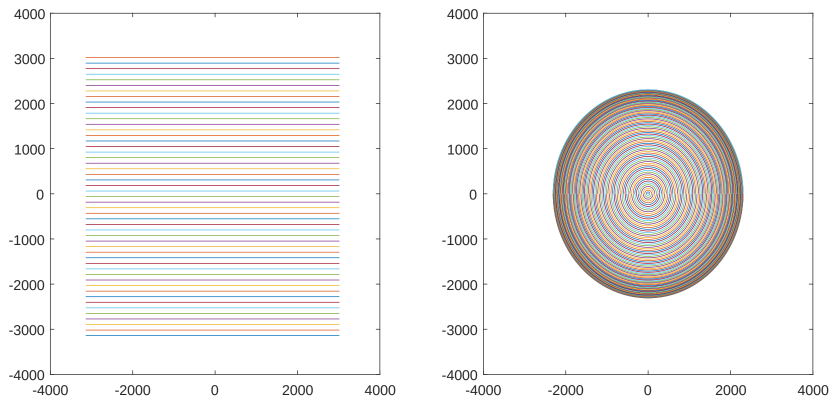

Figure 3.4: Different grid distribution: rectangular grid (left) and spherical projected grid (right).

To solve the interpolation, we have programmed an index search method: XY grid follows just two vectors (one for X, and one for Y), so we pair each projected matrix with one direction, looking for the closest value between the vector and matrix rows (or columns). Once we find the four point that enclose the projected point, we define two planes. We solve the plane equation for the projected point to approximates a value of the function (Fig. 3.5). If we sweep all projected positions, we can build an interpolated field with the closest field data for each output point. To increase accuracy, the interpolation module can be used to get more data. That avoid errors searching the optimal value: if two different point in $\phi\theta$ projected grid are too close, they may has associated the same field point, and that's not correct. The error values of the calculations are discussed below (Chapter 4). If a projected point is out of XY limit, the value of the function is set as 0.

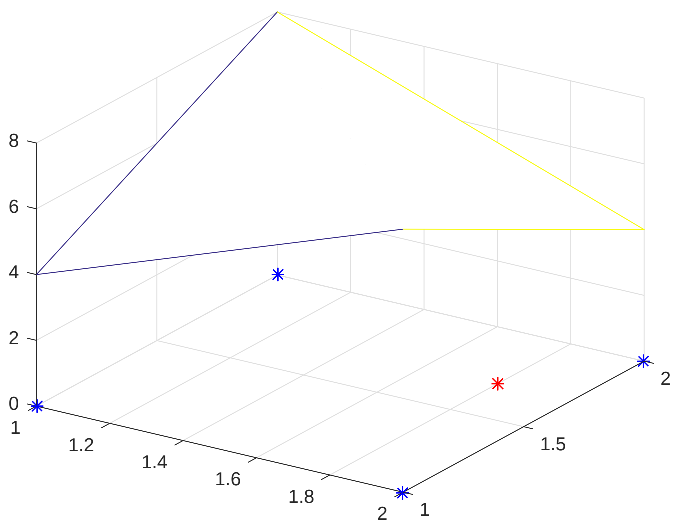

Figure 3.5: Interpolation by plane adjustment scheme. The Z value for XY coords at the red point is obtained solving the plane equation (in this case, the yellow plane equation).

Finally, we can obtain FF signal applying an exponential coefficient to "move" the field into a FF distance.

3.5 FIELD ROTATION

Due to pointing problems between the AUT and the probe, or the need to change a determined position of a point, field rotation can be really useful to correct an acquisition.

The rotation has two domains: XYZ, and phi-theta domain. Both domains use Euler matrix, R, to correct the position of the field, but phi-theta domain needs an extra processing. For direct rotation, R is multiplied by a coordinates vector, with XYZ data of a point, if both fields (with and without rotation) are represented, we will find a space rotation according to Euler angles. These angles will rotate the field by Z1 (Z axis before any rotation), X2 (X axis of the rotated field after first rotation) and Z3 (Z axis of the resultant field after two rotations).

To achieve the right rotated field, we just have to set two matrices: R and field XYZ position (each line will be related with a specific point). A loop will take each XYZ vector, will rotate the position, and will store the new coordinates. The rotation part is programmed in a friendly way thanks to Math.NET, that includes all matrix operation we need.

Phi-theta-domain rotation imply more calculation. As we commented at MATLAB implementation, it starts with a XYZ rotation as reference (v=[1 2 3]). This operation will set the roll (phi) and azimuth (theta) rotation. The index search method programmed for the NF to FF transformation is used here, too. It will give us the field position in roll-azimuth domain (that works as an extended domain from phi-theta). Program will search 0º position for both angles, and those indices will be taken as reference to shifting the field. Four submatrices (depends on the position, it could be two or none) are built to load in there the field data.

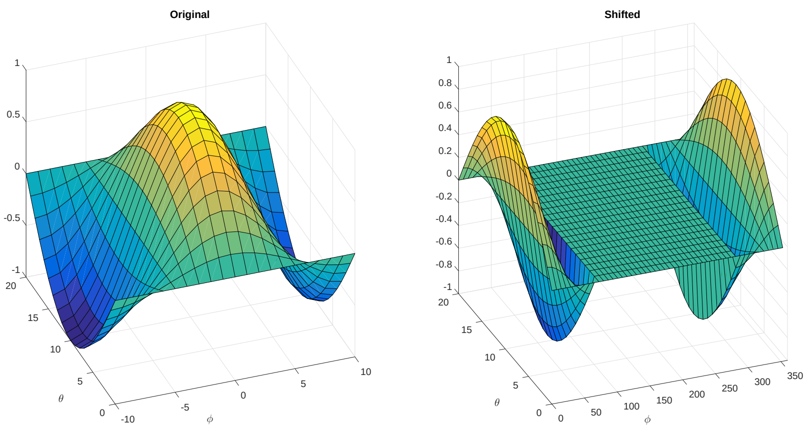

Figure 3.6: Example of shifting process. The signal is periodic (360º for ϕ and 180º for θ), so the negative values (left) appear before the 360º (right).

Once the field is extended into roll-azimuth domain, we set a margin. The domain is periodic, so the wave can be easily duplicated. This is really helpful to interpolate the signal into a real roll-azimuth domain: as we commented above, our shifting doesn't start at 0º, it starts at first positive angle value that index search function could find. Extending the signal assure a correct interpolation method.

The interpolation method has been programmed from a simple one. We manage complex data, so the first step is put them into two variables: one with real ones, and another with imaginary data. A linear interpolation with complex data can be set as a real data interpolation with an imaginary data interpolation. When both interpolation are done, results put together like $R_{data} + jI_{data}$.

Now, with the field perfectly set into the regular roll-azimuth domain ([0,360º-step] and [0,180º]), we can apply the Fourier transform, which allows us to easily move the signal across the domain.

$$f[m+M, n+N] = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F[k,l] e^{j2\pi(\frac{k}{M}(m+M)+\frac{l}{N}(n+N))}$$

The exponential factor that set the angle shifting is normalized with the number of points: due to the Fourier properties, the movement can't be expressed as an angle gap, it has to be set as a cell change (each point is considered in one cell). Fourier functions are implemented in the same way that NF to FF operations. After the exponential variable, inverse Fourier operation is applied.

The last step before finish Euler rotation is the re-shifting. To set the field into the input domain, we need to undo the field reorder. Four submatrix (or 2) are loaded at the correct order into an output matrix. This matrix has the same size that input field (input and output domain is the same).

3.6 INTERPOLATION

Interpolation module allows to increase data resolution. Euler rotation's interpolation is used as reference to program the module: complex data must be decomposed into real data and imaginary data to interpolate it separately.

As input we have the filed, the old steps and the new ones (for each dimension). The step are normalized by the old step to easily the calculation. The new size of the field will be

$$1 + \frac{length_{old} - 1}{step_{new}} = length_{new}$$

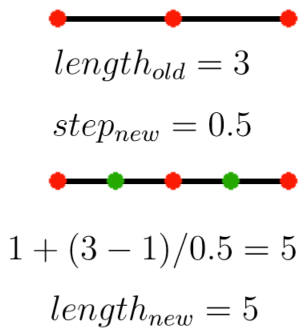

Figure 3.7: Example of new axis. Original points (red), with a $step_{old} = 1$, are splitted by the $step_{new}$ (green).

With the new size, we can fill the new vectors in each direction. The old ones are from 1 to the length of the field, one by one (as we have said above, its normalized). The new axis limits are the same, but they increase the value using the new step (that will be smaller than the old step).

After that, the interpolation will be set in two dimensions: a loop to interpolate the rows, and other loop to interpolate the columns. The interpolation method is the same that the function in Euler rotation.

3.7 CORRECTIONS

The measurements can have unexpected results if we don't take into account some issues that could happen: interference, calibration error… We have considered two errors to correct: noise error, and normalization.

3.7.1 NOISE CORRECTION

White noise is a random signal, characterized because its values at different time points have no relation to each other, that is, there is no statistical correlation between their values. It has uniform power across the frequency band. We will consider that it has a distribution, with an average value, so its Gaussian centered on a specific mean. This polarization noise always affects our measurements, as WGN (White Gaussian Noise) does, and it will be different at each point.

The power of the noise could be considered irrelevant if our power is high enough, and it happens at the center of our measurement: the main lobe of the AUT will point the center of the area. But at the edge of the measurement area, the power of the field is lower than the power of the noise.

The signal follows S=N+E, and if we consider the field close to zero at the edge of the measurement area, we will be able to cancel the noise. The mean of the field is zero, so the mean of the signal is the mean value of the noise. With this parameter, we can correct each point subtracting the signal mean.

The program load the noise correction with the BIAS parameters of the .PINP file (Fig. 3.2): activate or avoid correction (BIAS_CORR), dimension of the area to correct (N_corr, M_corr), and field of application (F_corr, wich can be X or Y). The measurement is done with two polarizations, and the most affected will be the crosspolar polarization. Crosspolar polarization is the one that its radiation pattern includes more radiated signal.

Once the four correction areas are delimited, each field component is accumulated to calculate the mean value. Both fields are loaded again to subtract the noise effect. The output of Bias correction is a same-size-input field.

3.7.2 NORMALIZATION

Data aren't always for seeing the AUT's performance. Sometimes, we are interested in comparing an antenna with another, and normalization helps to do that. This process change the module of all points, dividing its value by maximum. The phase of all points is preserved.

Applying this function, the maximum value just can be 1 (0 if we are working in logarithm). This is useful if we want to compare two fields by their behavior, not by their power: the main lobe width, the radiation form, etc.

Chapter 4

MEASUREMENTS AND RESULTS

In this part, a comparison between error parameters and performance of both programs is done. Each programing language has its own issues, so the results have to be taken considering each situation.

As we commented above, MATLAB is used as mathematical reference: the calculation can be implemented without error for NF to FF transformation, Euler rotation and interpolation. Furthermore, the extensive amount of functions that are implemented make really easy using MATLAB for this work. The graphical options are good, too: data can be plotted in many ways, so the results are easy to check.

C# options are focused to offer an easy execution and a good performance: once is compiled, the executable doesn't need another program to be used, and the lack of graphical options increase the performance of the program. To check the results, we should consider:

- Each module should be tested separately. Error can accumulate, and it's better to set the quantity of each module. Furthermore, the NF to FF calculation is not the same as the interpolation module, so the blocks that has a more complex calculation must be analyzed apart.
- The properties of the input data influence the results. All data that affect the quantity of points are very important for the calculation: the accuracy and the time of execution are directly affected by the number of measures, the factor of fundamental frequencies... So each measurement will have a critical calculation.
- Both programs start with the same measured data, so any acquisition error can be ignored. We can try to reduce its influence by software, but isn't necessary do it to compare two data processing.

4.1 MEASUREMENT ERRORS

Measurement process implies the management of many tools and devices (probes, computer systems, VNA's, cables, etc). The behavior and response of all these components are not perfect, and it doesn't follow the theoretical pattern at 100%. Fig. 4.1 shows a summary of the main parameters and errors.

Source of error	Method of evaluating: Computer Simulation	Method of evaluating: Test on measure	Method of evaluating: Error equation
Probe relative pattern			X
Probe polarization radio			X
Probe gain measurement			X
Probe alignment error			X
Normalization constant			X
Impedance mismatch factor			X
AUT alignment error		X	X
Data point spacing		X	X
Measurement data truncation	X		X
Probe x,y-position errors	X		X
Probe z-position errors	X		X
Multiple reflections		X	
Receiver amplitude nonlinearity	X	X	X
System phase error		X	X
Receiver dynamic range		X	
Room scattering		X	
Leakage and crosstalk			X
Random errors in amplitude/phase	X		X

Figure 4.1: Error sources in planar NF measurements [11].

4.1.1 ANTENNAS ERRORS

The probe is a critical element in the measuring process: The AUT and the probe are the two antennas of the system, and any property of one of them may affect the other. The main probe effects are:

- Relative pattern: as an antenna, probe has its own radiation pattern, and its peaks and lobules have to be determined to measure the effect.
- Polarization radio: a perfect probe should isolate different polarizations (a 0º linear polarization have to take no signal from a 90º polarization, for example). Because the probe is not perfect, the quantity of different-polarization signal that see the probe has to be set.
- Gain measurement: all systems will introduce gain or loss to our measurement, so the probe must be characterized.

- Alignment error: the mechanical parts precision affects the field (more relevant for high frequencies). If the step is not exactly the correct value ($\lambda/2$), or the AUT and the probe are not faced to each other, the acquisition can have errors. Laser systems help to reduce this problems.

- Leakage and crosstalk: some errors are defined by the geometry of the antenna. Power can be radiated where is not expected, creating a leakage error. A crosstalk effect can be produce by a pair of coupled cables, that induce a bad transmitted signal.

4.1.2 MECHANICAL ERRORS

Planar measurements have two mechanical axes (X-Y) that move the AUT or probe along all measurement area. Those axis, along with a third axis (Z) that can move the antennas closer or farther, can introduce some position errors. The shift should be as precise as possible, to take the measurement at the correct point. For high frequencies, the electrical distance is just a few of mm (for example: 12 GHz, $\lambda/2$=12.5 mm), so any position error can produce a bad acquisition.

A typical effect of a bad position error is the aliasing: if distance between points is bigger than $\lambda/2$, signal data will interference with next to points, inducing acquisition errors.

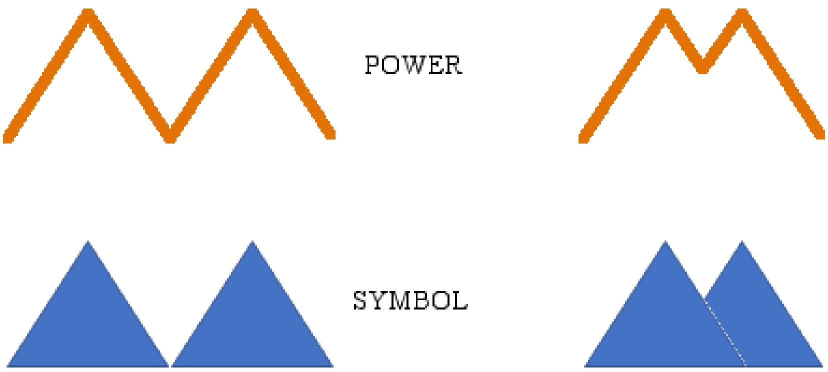

Figure 4.2: Example of aliasing error. Left symbols are not overlapped, so the receiver can identify them correctly. Right symbols are taken too close, so the power of the second symbol interfere with the first one, creating an error.

4.1.3 SOFTWARE ERRORS

Some errors are produce for the software calculation. Data adjustment, like normalization constant, can spoil the acquired data, especially secondary lobes. Both patterns (AUT and probe) must be normalized by the same factor. Other cases the signal could be distorted (Fig. 4.3). Secondary lobes are affected by the pattern of the probe, that decrease while the measure moves from the center of the probe. This effect minimize the effect of rebounds and secondary signals. That is the reason that main lobe is the most affected by a bad normalization process.

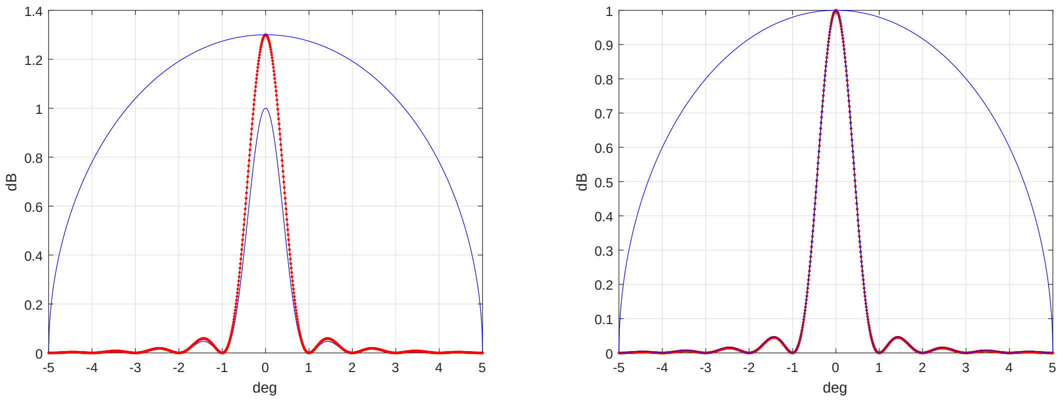

Figure 4.3: Example of normalization: if the AUT and the probe are not normalized at the same factor, the signal shows a false level (left). Blue lines are the antenna's radiation patterns, and red line is the field that the system obtains.

Another error of calculation is the measurement area truncation. When we set that the field just cover a sector of the total space that the AUT covers, we assume that all the power is in this area, and that's not true. In planar measurements, the maximum area calculated is a semi-sphere, so an intrinsic error affects our data. Spherical systems minimize this error because they cover a complete sphere around the AUT.

4.1.4 RECEIVER ERRORS

A good measurement needs a complex and calibrated receiver system, that includes the VNA, cables, amplifiers and many other components. The most common error are:

- Impedance mismatch error: like any other circuit, power can be reflected or transferred at connected ports. A huge reflection could damage our system, so we need to assure a good power transfer.
- Reflections: the chamber is part of the system, and its geometry can cancel many reflections, and attenuate the low-grade signal (with just a few of rebounds). The quiet zone may be compromised if the scattering creates many distortion signals.
- Amplitude nonlinearity: the response between input data and output data must cover the linear range of the system. A nonlinear response makes the acquisition invalid, because there is no sense between the input and output field.
- Dynamic range: dynamic range its defined as the range between the minimum signal that we can take, and the maximum level of field we can measure. These

levels have to include active elements, like amplifiers, to avoid any saturation point. They must keep the linearity criteria.

- Phase error: signals have and amplitude and a phase associated to each point. A bad phase induces error data, because the receiver can't read correctly the signal. The main phase error sources are rotary and cables effects, and temperature problems.

4.1.5 OTHER ERRORS

Random error points can appear while the signal is evaluated. Electrical system, like a bad earth wire, can produce signal derivations and affect to the measured signal. Active components, like mixers, could create fake signals (like image signals) if the work frequency is not the expected. Another source of error could be the isolation of all components. These complex systems need a huge accuracy level from their components.

4.2 NF TO FF TRANSFORMATION

4.2.1 TEST MEASUREMENT

To train the C# program, we need to manage a very known data. ASYSOL has many resources from proves, previous installations, so a complete planar NF measurement will be the test field. Those data include two polarization measurement, for many frequencies. We are going to use a 51x51 measurement for 110 GHz. The output will be calculated for all possible angles (ϕ from 0º to 180º, θ from -90º to 90º).

First of all, a correct MATLAB program is needed. A theoretical script has been developed by ASYSOL: The program takes all the calculations from Balanis [6] [7] and apply them to a input field (as we commented at the NF to FF transformation section in Chapter 2, MATLAB implementation). All parameters needed (steps, angles. . .) are defined in a .MAT file with the field data. Fig. 4.4 represents the output FF fields.

NF to FF transformation function has the more complex calculation: FFT, interpolation and domain change. That means that most of the error will come from this module. The reason is that MATLAB and Math.NET has no difference using simple operations included in the other modules. Furthermore, more complicated calculations, like Fourier transforms, have similar results. Until the projection from spherical coordinates to rectangular points, MATLAB and C# results are exactly the same. So, the domain change between the $\theta\phi$ projected area and the kx and ky grid becomes the main function to introduce error, because is programmed using an alternative method (2D interpolation is not implemented in Math.NET, so as is explained at Chapter 3, a interpolation adjustment by planes is used). The error is reduced increasing the number of points we have: a pre-interpolation is recommended before change the domain.

To compare both programs, C# results have been loaded into MATLAB. Fig. 4.5 shows the theta-Z axes. The left column corresponds to the MATLAB results. Right

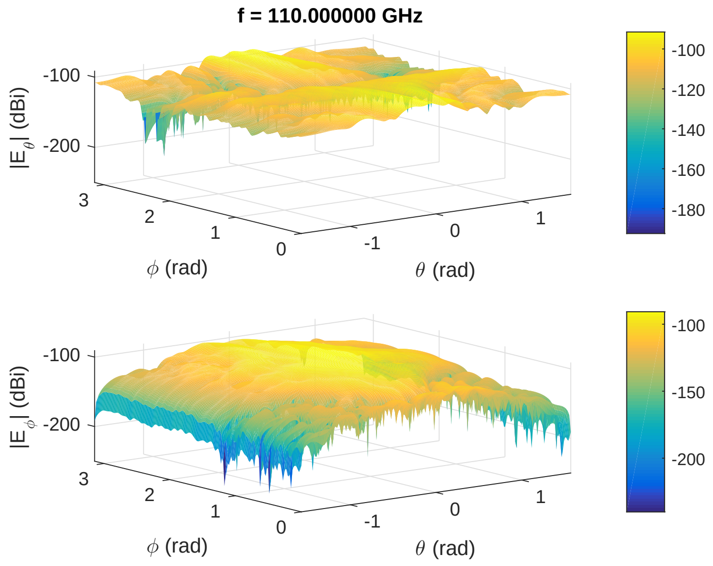

Figure 4.4: Output test FF calculated using MATLAB.

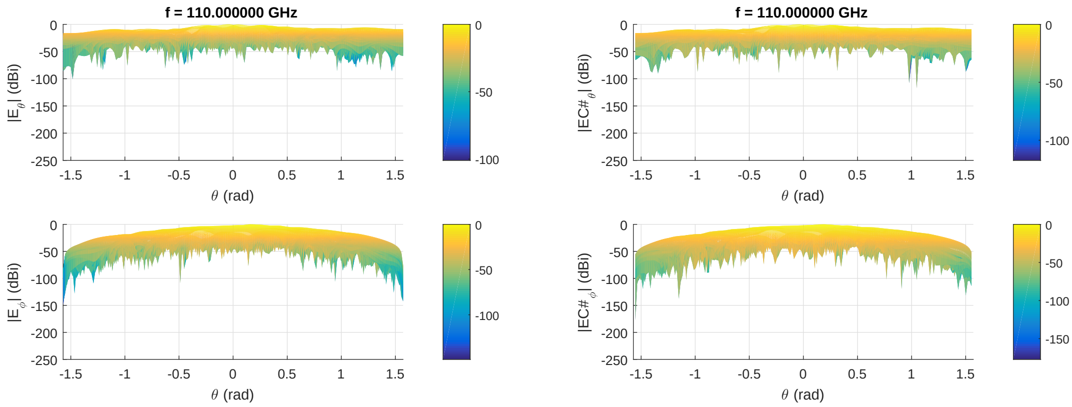

Figure 4.5: Output field calculated using MATLAB (left) and C# (right). All fields are normalized.

column shows the C# output. The figures are similar, but we need to process all data to evaluate the performance of both programs. We have calculated the error function for each point by

$$20log10(||Ematlab| - |EC\#||)$$

Both fields *Ematlab* and *EC#* are normalized before obtain the error values.

The error function is always under the -50 dB point, so we can assume that the calculation is correct. The distribution increases its value at the center of the measurement, which is normal, because the main beam matches this area. There are more power, and any peak can produce a bigger error due to the interpolation method. At the edge of the surface, the variation of magnitude decrease (there are less power of

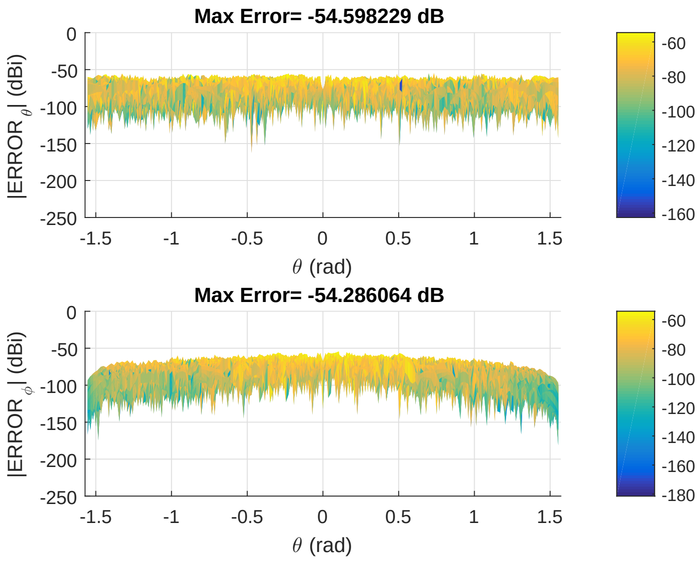

Figure 4.6: Error function between MATLAB and C# calculation.

field), so the error as well.

This error is adjusted with a pre-interpolation calculation: we reduce the step of the function to increase the number of points. This reduce the error until -100dB, but it can be a problem if the field has a big measurement surface, because its computational cost. The execution time and the memory behavior can be compromised. So the best solution is define an adaptative step interpolation: with good number of original points, we are going to calculate less pre-interpolated points.

4.3 ALGORITHM STABILITY

To see how acts the algorithm under different conditions, the test field is going to be checked with different MODEXP (no expansion, 4, 8, 16 and 32) variable and different BIASCORR (no noise correction, 1, 2 and 4) parameters. All no-corrected fields are taken as reference. MODEXP factor increase the number of points as result of Fourier transform, so the signal resolution increase, too. BIASCORR factor specify the size of the area of the noise measurement.

This new test field corresponds to planar antenna (Fig. 4.7) measured in a planar grid of 67x67 points, and the frequency of the measurement is 12 GHz. Its output has been calculated for all possible angles (ϕ from 0° to 180°, θ from -90° to 90°) with a step of 1°.

The first thing we can see after draw the fields with different MODEXP is the similar form, but the different level between them (Fig. 4.8): when the MODEXP increase its value, the number of points is bigger, so the mean power decrease. Same situation happens for both polarizations.

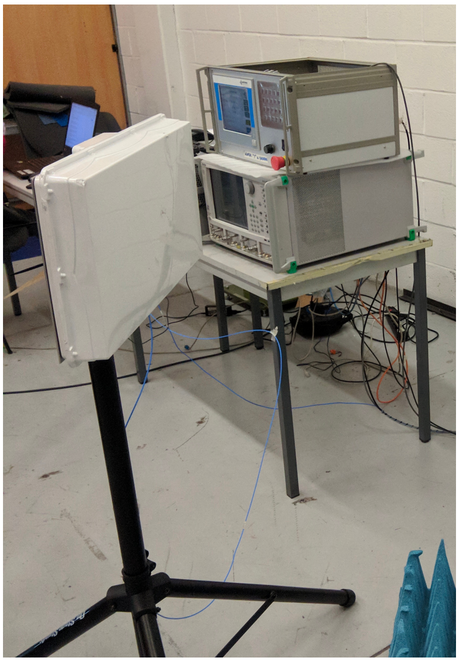

Figure 4.7: Planar antenna measured to obtain a real test field.

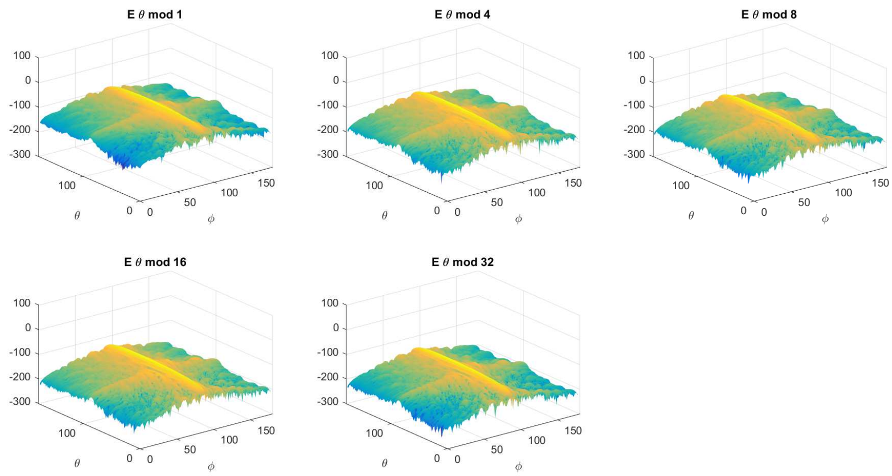

Figure 4.8: Reference crosspolar field for different number of modes calculation. All data are in dB.

Evaluating the error between the reference field (no noise correction, no expansion) and the different BIASCORR data (with 1 point, 2 points and 4 points of noise correction area) and MODEXP fields (1, 4, 8, 16 and 32 as expansion factor), we can see that all error is under 75 dB (some examples in Fig. 4.9). This indicates that the algorithm has a stable behavior: it calculates the same, no matter the input conditions for the same input .DAT. In this case, a small BIAS grid is selected, because the input has 51 points in both axes, so the BIAS grid has to be consequent with field size. Before the error calculation, all fields have been normalized.

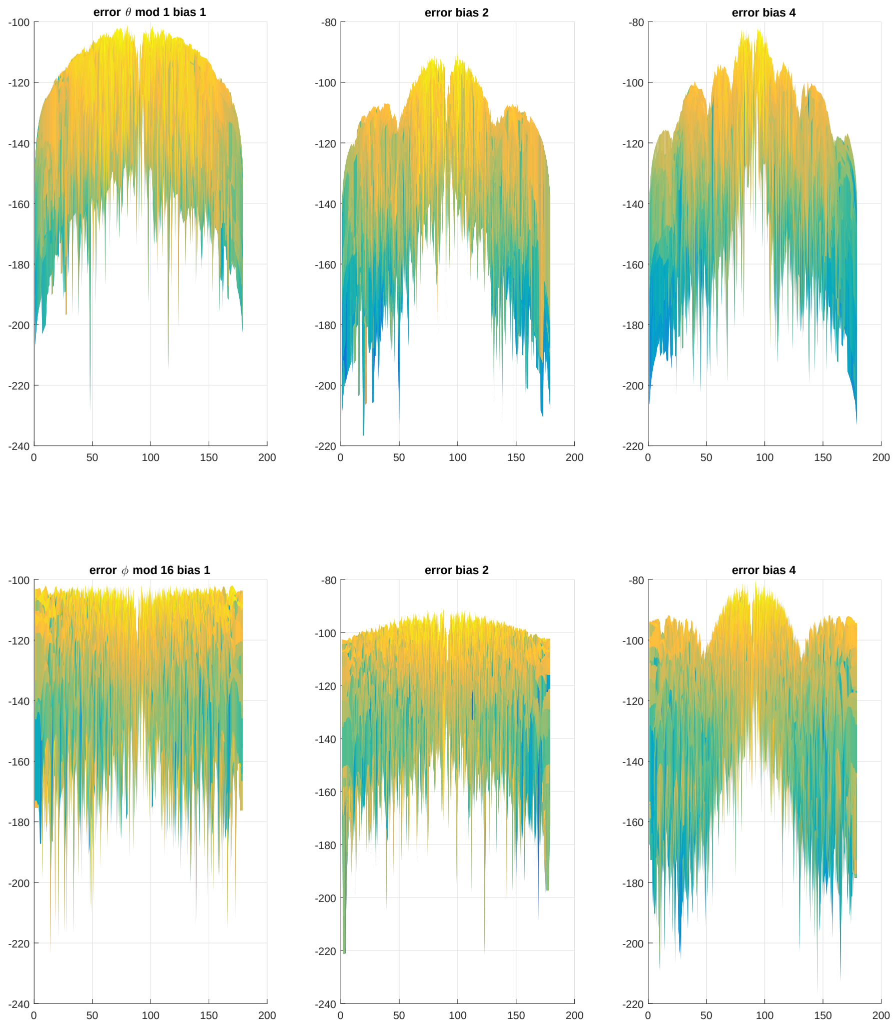

Figure 4.9: Different error for some MODEXP-BIAS combination. First row shows error between reference field without MODEXP and BIASCORR of (from left to right): 1, 2 and 4 points. Second row shows error between reference field with MODEXP = 16 and BIASCORR of (from left to right): 1, 2 and 4 points. All data are in dB.

4.4 TIME PERFORMANCE

Results are important, and the main target of any calculation. But there are more important parameters, like time of execution and memory performance. Comparing memory management between MATLAB and C# seems useless: algorithms are so different, and to compare it we need to define all the small functions under the main calculation, with all its variables.

But time performance is easy to extract and compare. Both programs have defined functions to measure the time between two points. Is important to consider some points to assure a fair comparation:

- MATLAB programs include some functions to draw the solutions. Those parts have to be excluded from the time measured. C# just calculates a text file as output, there is no any representation.
- The input data must be the same. The size of input field is a critical parameter, but all variables that affect the interpolation process, or the complex calculation (like FFT/IFFT) are really important to the execution time.
- The execution must be done under normal computer conditions: we are going to try to stop all secondary programs. If MATLAB or C# are running with other heavy programs, the performance won't be realistic.

To test the software, we have changed two parameters: pre-interpolation factor (the value that indicates the increase of points before the change between the rectangular and the spherical domain) and the size of the input field. With more point (for the size or the interpolation factor), more time of execution is expected.

C#		Field size			
		51x51	200x200	1000x1000	3000x3000
Interp. Factor	1	910ms	2s 370ms	19s 600ms	2min 32s 200ms
	4	1s 540ms	4s 910ms	39s 50ms	4min 3s 960ms
	10	2s 870ms	11s 500ms	1min 45s 90ms	Out of memory
	20	5s 260ms	24s 810ms	Out of memory	Out of memory

Matlab		Field size			
		51x51	200x200	1000x1000	3000x3000
Interp. Factor	1	1s 38ms	1s 971ms	2m 15s 642ms	Out of memory
	4	1s 300ms	2s 653ms	2min 18s 226ms	Out of memory
	10	1s 844ms	4s 853ms	2min 29s 49ms	Out of memory
	20	2s 733ms	8s 854ms	4min 40s 471ms	Out of memory

Figure 4.10: C# and MATLAB execution times. Biggest field combinations (grey background) are out of the computer capacity.

This test also can indicate the limit of point for the software: a large quantity of points can produce an out of memory exception. The characteristics of the computer affect directly to the amount of data that the computer can handle. All the times are obtained in a i7-5500U of 2.40 GHz CPU, with 8 Gb of RAM memory.

Each value in Fig. 4.10 is the mean time of 5 executions. After a brief assessment, C# is the best option while the number of data is supported. For small input files, the

C# speed is lower than MATLAB's, but it doesn't take too long. For big data files, MATLAB has more problems loading the points. The main problem for both programs is the memory management, which avoid a correct calculation for extreme (large field) conditions. To lead with this issue, the interpolation-factor must be adaptive: it takes a large value for small fields, and it is reduced with the field size increase.

4.5 FIELD ROTATION

After the NF to FF transformation, field rotation is the second complex system to check. To compare the calculation between MATLAB and C#, we are going to test only the roll and azimuth, because the program calculates XYZ and roll and azimuth, but the output is just in the roll and azimuth domain.

The input 51x51 field used for the test measurements is rotated with Euler angles: ϕ=15º, θ=-40º and ψ=10º.

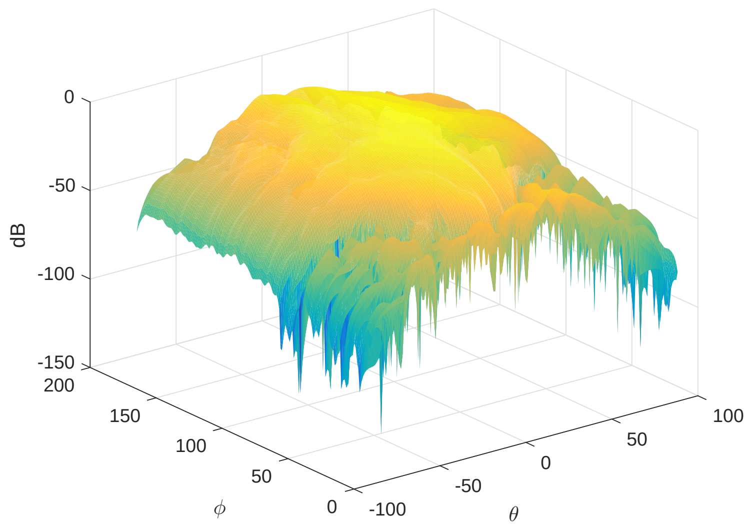

Figure 4.11: Original far-field without rotation.

Visually, the fields (Fig. 4.12) are quite similar, but there is the correct way to evaluate it. The same error function that in the NF to FF section is calculated.

This process shows a really good performance, with a max error of -264 dB, that only includes the rotation error (the input for MATLAB and C# is the same). So we can consider that this calculations doesn't add any error to our data.

Both process (FF transformation and rotation) use the interpolation block, so no interpolation error is assumed.

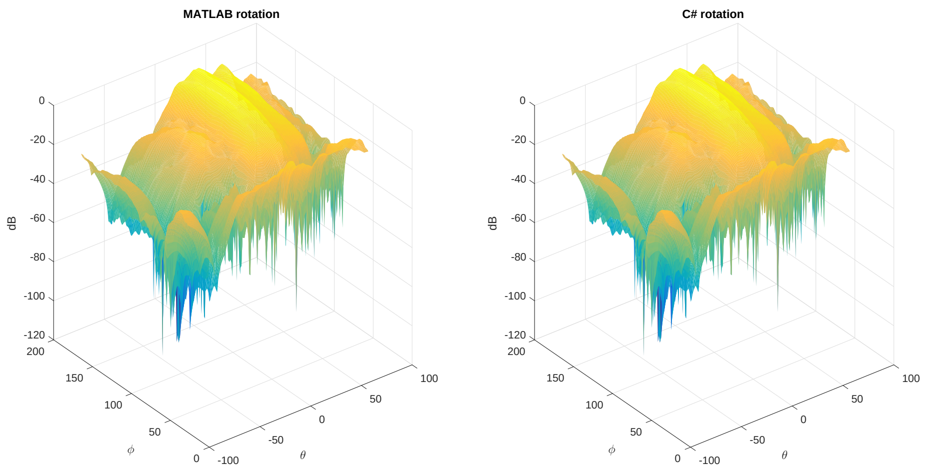

Figure 4.12: Euler rotation by MATLAB (left) and C# (right).

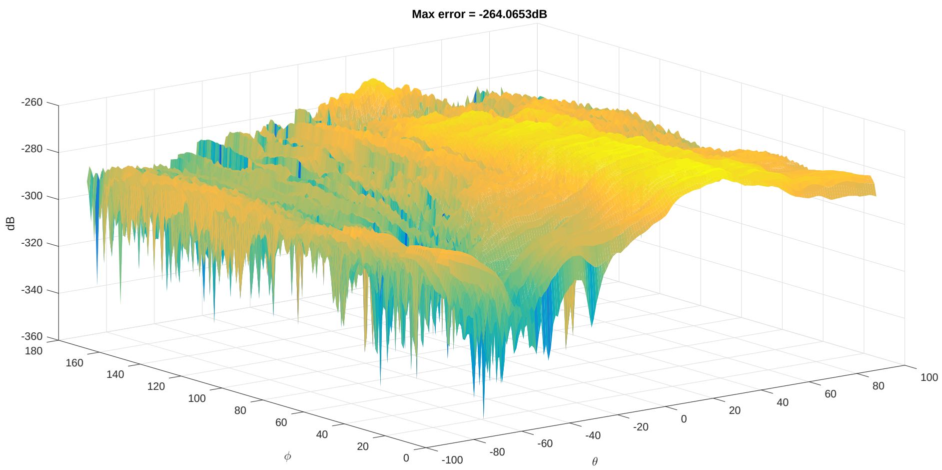

Figure 4.13: Error between both rotations of Fig. 4.12.

Chapter 5

CONCLUSIONS AND FUTURE WORK

Planar NF measurements are near field antenna measurement systems with the easiest mechanical implementation but this does not mean that they are trivial. Form the transformation from NF to FF point of view, is the kind of NF measurement system that involves the lowest computational cost. However, a complex mathematical process has to be considered that involves more correction factors than other measurement methods. That's why a correct approach is important before achieving a correct transformation.

The main objective of this TFM, has been reached: a fast and simple software tool to carry out the NF to FF transformation including some corrections and postprocessing capibilities has been developed. The software is a software development environment programing environment independent tool implemented in a .exe app. The tool uses as input files with with a known predetermined format and provides as output a file with the same input-format allowing its recurrent use in such a way that each call carries out just one calculation.

As demonstrated in the results Chapter, the program works properly for the NF to FF transformation with assumable error levels, and post-processing and implemented corrections works properly.

5.1 FUTURE WORK

Along this project it has been shown how the tool developed is working properly. However, that does not mean it is finished. The software can be improved, and in this section, a comment about different actions to carry out some improvements is done: algorithm optimization and new options addition.

Concerning algorithm optimization, the logical option is to analyze the developed code performance to improve speed and memory management. This implies checking all small functions defined using C# properties to obtain an optimal software. C# is an object-oriented programming-language and, taking advantage of those functionalities, the software will reach a great performance. Furthermore, the libraries (like Math.NET) and drivers should be updated: new corrections and new functions are released with each version, so revisions are needed from time to time. The aspect that requires a significant further improvement is the interpolation between the rectangular grid and the spherical projected domain, that is the is the biggest error source in the code.

Main calculation (NF to FF transformation) is a mathematical process that includes too many variables like the mechanical precision of the positioning system or the effect of the probe over all the measurement. The main effects that can be measured, can be corrected too and programmed to improve the transformation. However, the main source of errors form the transformation point of view is the probe correction and needs to be treated on the first place.

The probe, as an antenna, has its own radiation pattern and affects the measurements with its lobes, gain... Probe effects is one of the most representative parameters from the corrections point of view. The probe affects mainly to the secondary lobes due to its aperture. The perfect data acquisition would be just from a point, associated with the main beam, but its impossible to achieve because there are another signals. The secondary lobes are more affected than the main lobe because the power of the main beam makes interference from the probe almost irrelevant at this point. But secondary lobes, with less power, need to be corrected.

The idea is calculated the NF data for the probe, and subtract its effect to the AUT measured data. The simplest way to do it is to measure in the same planar conditions the probe, but is not the easiest method: all acquisitions implies two realizations (one for the AUT, and on for the probe). The optimal way is use the spherical measurement for the probe: because its geometry, just 4 cuts are needed for the majority of the probes, so the measured time is minimum. This method implies a complex mathematical calculation, including spherical to cartesian change domain and scattered interpolation.

The starting point will be the FF probe field. The FF data of the probe will be move to the aperture plane (applying an exponential factor). The next step will be the domain change: data are now in a ϕ-θ grid, and we have to interpolate the signal into a kx and ky grid. Once the interpolation is done, the probe field is ready to be applied to the measured data:

$$NF_{measured} = NF(AUT) + NF(Probe) \quad NF(AUT) = NF_{measured} - NF(Probe)$$

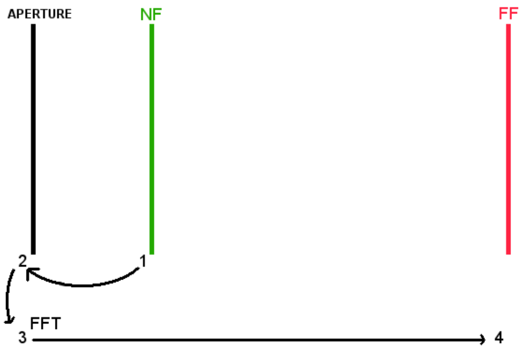

Figure 5.1: Workflow of a NF to FF transformaion. 1: NF measurement. 2: field movement to the aperture. 3: FFT transformation. 4: FF distance movement.

With the data corrected, the NF to FF transformation continuous as is described in Chapter 3.

BIBLIOGRAPHY

[1] C. Balanis, “Antennas,” in Antenna theory: Analysis And Design, 3rd ed. New Jersey:Wiley, 2005.

[2] C. Balanis, “Fundamental Parameters of Antenna,” in Antenna theory: Analysis And Design, 3rd ed. New Jersey:Wiley, 2005.

[3] A. Cardama Aznar, L. Jofre Roca, S. Blanch, “ Consideraciones generales sobre antenas,” in Antenas, 2nd ed. Barcelona, Spain:UPC, 2002.

[4] A. Cardama Aznar, L. Jofre Roca, S. Blanch, “Medida de antenas,” in Antenas, 2nd ed. Barcelona, Spain:UPC, 2002.

[5] S. Pivnenko, B. Bedia Exposito, A. R. Ruiz and E. Venero, ”Wideband low weight probe for near-field antenna measurements in UHF band,” 2017 11th European Conference on Antennas and Propagation (EUCAP), Paris, 2017, pp. 3783-3787.

[6] C. Balanis, “Aperture Antenna,” in Antenna theory: Analysis And Design, 3rd ed. New Jersey:Wiley, 2005.

[7] C. Balanis, “Antenna Measurement,” in Antenna theory: Analysis And Design, 3rd ed. New Jersey:Wiley, 2005.

[8] A. Yaghjian, ”An overview of near-field antenna measurements”, in IEEE Transactions on Antennas and Propagation, vol. 34, no. 1, pp. 30-45, Jan 1986.

[9] L. H. Hemming, “Chapter,” in Electromagnetic Anechoic Chambers: A Fundamental Design and Specification Guide, 1st ed. New York: Wiley, 2002.

[10] W. H. Kummer and E. S. Gillespie, ”Antenna measurements—1978,” in Proceedings of the IEEE, vol. 66, no. 4, pp. 483-507, April 1978

[11] A.C. Newell, ”Error analysis techniques for planar near-field measurements”, in IEEE Transactions on Antennas and Propagation, Vol. 36, No. 6, pp. 754-768, June 1988.