Alma Mater Studiorum · Bologna University

Cesena Campus
School of Engineering and Architecture
Department of Electrical Energy and Information Engineering
- Guglielmo Marconi -

Master's Degree Course in
Electronic Engineering and Telecommunications for Energy

# NON-LINE-OF-SIGHT LOCALIZATION WITH FREQUENCY-SELECTIVE METASURFACES

Master's degree thesis in
Telecommunications systems LM

Relator:
**PROF.**
**DAVIDE DARDARI**

Edited by:
**GIACOMO CALESINI**

Correlator:
**MARINA LOTTI**

Academic Year 2021-2022

# Contents

# Introduction

Wireless connectivity has now become an indispensable resource for our society. However, the significant increase in traffic demand, year after year, forces wireless networks to satisfy increasingly stringent requirements, which are emphasizing intrinsic limits, difficult to overcome by acting only on the network's end-points. Recently, this has given rise to the definition of the new paradigm of Smart Radio Environments (SREs), i.e. wireless networks in which the radio environment can be reconfigured (hence the term smart) in order to overcome the negative effects of natural electromagnetic (EM) propagation such as: multipath fading, shadowing and path-loss and thus optimize communication efficiency. Indeed, often exploit these effects in order to improve communication. An emerging technology, that SREs rely on to improve wireless link quality, are Reconfigurable Intelligent Surfaces (RISs).

A RIS, in general, can be understood as a thin layer of EM composite material, typically mounted on the walls or ceilings of buildings, which can be reconfigured even after its deployment in the network. RISs made by composing artificial materials in an engineered way, in order to obtain unconventional characteristics, are called metasurfaces. Through the programming of the RIS, it is possible to control and/or modify the radio waves that affect it, thus shaping the radio environment. In addition to this, RISs, being nearly-passive devices, have a reduced energy consumption and low cost, are easy to implement and install on site and allow to reduce the number of base stations because they recycle the wireless signals already "on the air" by introducing reflections, refractions which allow techniques such as focusing or beamsteering. They have many fields of application, in fact they can be used: in cellular networks, for example to reduce dead coverage areas or to circumvent obstacles between the user and the base station; in indoor communications affected by harsh propagation, to ensure wireless connectivity in all corners of the structure; in autonomous networks, for example to boost signals from base stations to autonomous vehicles; in IoT networks, for example to create smart sensor networks, smart agriculture and smart industries. RISs also have disadvantages, including that they are powered to ensure reconfigurability and that estimating Channel State Information (CSI) for configuration optimization can improve the system complexity, bring additional cost, and lead to a long training time that can be intolerable in dynamic scenarios. In addition, a dedicated communication link between the base

station and the RIS must be ensured in order to configure the RIS. These disadvantages could penalize RISs in certain applications. As an alternative, the metaprism has been recently proposed by the research group of the University of Bologna led by prof. Dardari, i.e. a low-cost, passive and non-reconfigurable frequency-selective metasurface that acts as a metamirror to improve the efficiency of the wireless link. In particular, using an OFDM (Orthogonal Frequency-Division Multiplexing) signaling it is possible to control the reflection of the signal, suitably selecting the sub-carrier assigned to each user, without having to interact with the metaprism or having to estimate the CSI.

This thesis investigates how OFDM signaling and metaprism can be used for localization purposes, especially to extend the coverage area at low cost, in a scenario where the user is in NLoS (Non-line-of-sight) conditions with respect to the base station, both single antenna. In particular, the paper concerns the design of the analytical model and the corresponding Matlab implementation of a Maximum Likelihood (ML) estimator able to estimate the unknown position, behind an obstacle, from which a generic user transmits to a base station, exploiting the metaprism. The results of the various simulations carried out are reported, for different scenarios, in order to understand how to act on the main design parameters that influence the performance of the estimator. Our work is structured in the following manner.

Chapter 1 introduces the Smart Radio Environments, the difference with respect to a conventional radio environment is clarified, the types of SREs are briefly indicated and the main advantages brought about by the introduction of RISs are listed. Then follows a description of RISs in terms of structure, implementation differences, operating and reconfigurability principles, functioning methods, application fields and main limits.

In chapter 2 the metaprism is presented as a possible alternative to the limits of the RISs. In the first part a 3D model of frequency selective metasurfaces is proposed, with particular interest on the phase profile. A hint follows regarding a possible physical implementation of the metaprism, under study at the University of Siena. A typical scenario of use of the metaprism is then illustrated, in which aspects is made use of OFDM signaling and the models adopted for the channel gains and of the received signal are derived. Two different design criteria of the metaprism phase profile are proposed: one that performing sub-carrier-dependent beamsteering therefore more suitable for the use with far-field users (for which the plane wave propagation assumption holds) and the other based on random values more suitable for use with near-field users. The scenario considered for the study of the localization application is then illustrated, described by the models presented previously but appropriately declined. In the final part, the derivation of the model of the ML estimator used for the estimation of the user's position is shown.

In chapter 3 the Matlab implementation of the derived ML estimator is explained. The first part introduces the generated scenario and the operating conditions on which the program is based, while the second part shows the Matlab code that creates the estimator and describes all the parameters used in the simulations.

Chapter 4 reports the results of the simulations, carried out to evaluate the performance of the ML estimator. The first part of the chapter is introductory to the simulations, while the following parts refer to the performance of the estimator as the value of the main parameters varies, such as the metaprism size or the number of subcarriers, in the two cases of metaprism phase profile design and for the two scenarios of far-field and near-field user's positions. Comparisons between the performance of the estimator obtained in the various simulations are reported, drawing conclusions regarding the best parameters configurations in terms of estimation accuracy.

# Chapter 1

# RIS-empowered Smart Radio Environment

## 1.1 Smart Radio Environments

The radio environment, in past generations of wireless networks, was thought as something uncontrollable, since the radio waves, once emitted by the transmitter and before being received, propagated through the wireless channel in an uncontrollable way. In fact, the design of wireless networks consisted of signals pre-processing at the transmitters and/or signals post-processing at the receivers, in order to compensate for the effect of the wireless channel. The desire to be able to control the propagation environment is driving research to study how to transform the wireless environment into a variable that can be programmed and controlled, thus making it intelligent. Hence the concept of Smart Radio Environments.

SREs are wireless networks, in which the environment is a space that can be exploited in order to optimize communication efficiency and signal quality, depending on the scenario considered. Typically, in fact, different replicas of the transmitted signal are generated due to the reflection, diffraction or scattering introduced by the objects along the channel. These, called multi-path components, arrive at the receiver with different phases, intensities and delays and their constructive or destructive sum causes a distortion of the received signal.

An emerging technology used by SREs is based on Reconfigurable Intelligent Surfaces, which combined with the use of specific software, allows to redesign the propagation of radio waves. In particular this technology tries to coherently combine the signals diffused by the surfaces and realize a deterministic and controlled environment capable of overcoming the negative effects of natural EM propagation, such as the aforementioned multi-path fading or the attenuation of signal power due to path-loss or shadowing. In Fig. 1.1 a conceptual scheme of SRE is represented, which unlike a non-Smart Radio

Environment, allows to optimize not only the transmitting and receiving nodes (Tx and Rx), but also the propagation environment (H).

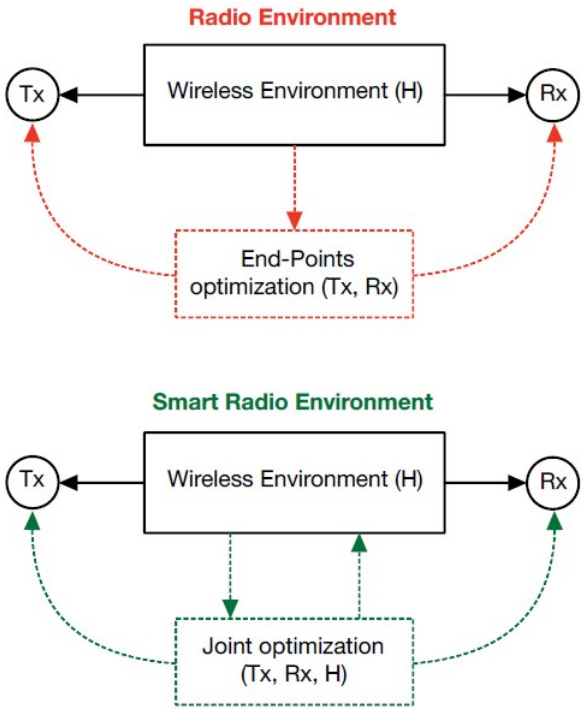


Figure 1.1: Radio Environment vs Smart Radio Environment conceptual models [1]

Depending on how RISs shape the incident radio waves, more states of the wireless environment can be obtained, which can then be programmed according to the desired one. This possibility, together with the operation of the transmitter and receiver, makes it possible to improve the performance of the communication system. In fact, while in a conventional radio environment it is only possible to control the receiver thanks to its feedback, in an SRE, in addition to this, it is possible to count on the control of the RIS, also thanks to its feedback, when present.

The implementation of an SRE can be of three types [1]: centralized, distributed or hybrid. In the centralized implementation, there must be a central controller which is responsible solely for the task of obtaining information on the state of the channel, through ad hoc protocols. Therefore RISs, in this case, are very basic, i.e. they do not require on-board signal detection or processing capabilities but must only be able to be reconfigured according to needs. In a distributed implementation, instead, a central controller is not required but the task of detecting the wireless channel, extracting information on the channel status and transmitting it, is entrusted to the RISs, which therefore must use sensors and nanocommunication protocols. The hybrid implementation instead involves both the central controller and the RISs, and is used to mitigate

the complexities of both previous implementations, i.e. signaling overhead for channel estimation in the centralized implementation and higher power consumption and more complex hardware in the distributed one.

Being able to create wireless channels generated by nature but controlled by communication designers, allows to overcome the limitations of wireless network design, including [1]:

- the introduction of RISs allows to obtain additional information on the state of the channel, therefore by jointly optimizing transmitter, receiver and environment it is possible to obtain a greater capacity of the wireless channel;
- multiple antenna structures can often involve very complex hardware and are not easy to implement, even in base stations. The introduction of RISs makes possible to move the operations typically performed by multi-antenna transmitters (digital) to the smart radio environment (analog), allowing optimization of the communication network;
- RISs allow for more efficient use of the emitted power, guiding radio waves towards desired directions and therefore reducing the energy scattered towards undesired directions;
- in scenarios where a sufficiently rich multipath scattering is not present, RISs, replacing conventional multiple antenna systems, can create orthogonal links to serve multiple devices simultaneously.

SREs are therefore enhanced by the introduction of RISs; the latter will be described in terms of structure, operation and applications in the next section.

## 1.2 RIS (Reconfigurable Intelligent Surface)

The building block of a typical Smart Radio Environment is the RIS. This, in general can be understood as a thin layer of EM composite material, typically mounted on the walls or ceilings of buildings, which can be reconfigured even after its deployment in the network. Through the programming of the RIS, it is possible to control and/or modify the radio waves that affect it, thus modeling the radio environment.

Some characteristics that make RISs excellent within a communication system are [2]:

- as they are nearly-passive devices, i.e. they require a minimum power supply only for the configuration phase, they have low energy consumption;
- they have a low cost, since they do not require any analog-to-digital converter, power amplifier or high hardware complexity, typical instead of the RF equipment

present in existing transceivers. This greatly simplifies the complexity of the network;

- the transversal dimension of a RIS is much greater than its thickness, for this often they are described as very thin sheets. This makes them much simpler to make and also to install on site, compared to three-dimensional surfaces of non-negligible thickness;
- they can work at any frequency;
- they improve the quality of the wireless link;
- they support full-duplex (FD) communication and are compatible with existing wireless standards;
- they reduce EM pollution, reducing the number of base stations introduced to oppose multipath. In fact, these emit additional signals that propagate in the environment, increasing EM radiation. This type of solution is better in economic but above all environmental terms and allows the reduction of EM waves in scenarios where they are important (e.g. hospitals).

### 1.2.1 RIS generic structure

A RIS consists of layers of composite material formed of metal or dielectric patches made on a grounded dielectric substrate (Fig. 1.2). Each patch, which acts as a passive scattering element, can have different forms and represents the basic unit of the RIS. Several patches, suitably organized on the surface, form the so-called cells.

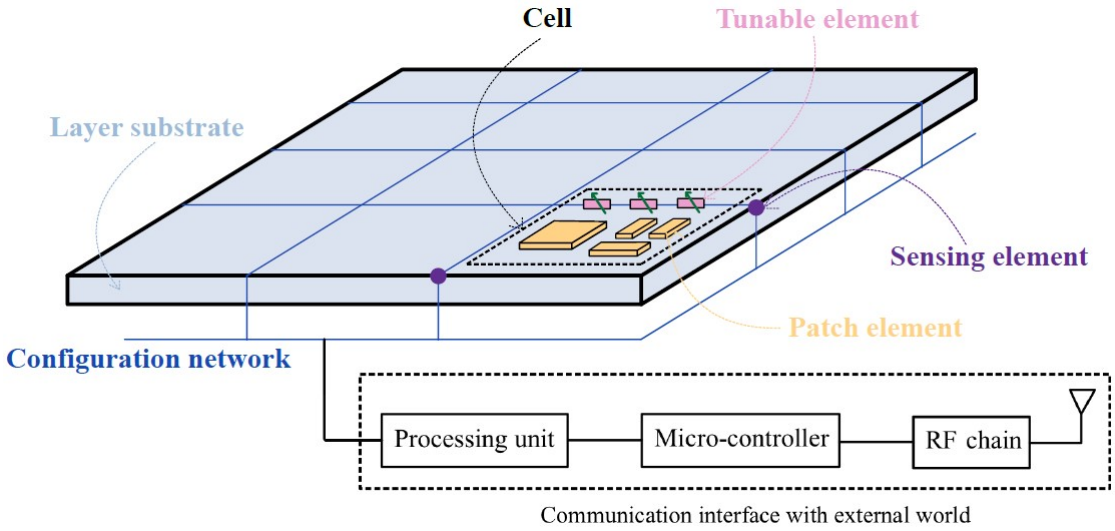


Figure 1.2: Conceptual structure of a generic RIS; for simplicity, only one cell is shown

The response of the RIS to incident waves depends on how each cell is designed. In order to interface with the external world and to be able to be programmed, a RIS must have at least one gateway that acts as both a transmitter and a receiver. Furthermore, it requires a microcontroller and a communication network between the cells to ensure that information is exchanged from the gateway to the entire smart surface. There is also a power requirement for the electronic circuits operations that enable RIS reconfigurability. It is possible that a RIS also has low-power sensors (optical, acoustic, temperature) for detecting information regarding the channel status (CSI), useful for optimizing its operation. However, these sensing devices increase the power consumption and cost of the RIS, so they are often omitted. In these cases, the CSI is estimated through ad hoc protocols, however made up of very often complex algorithms.

### 1.2.2 Antenna arrays vs Metasurfaces

The practical implementations of a RIS can be divided into two categories: smart surfaces formed by antenna arrays and smart metasurfaces.

Antenna arrays

In this type of smart surface, each cell is composed of an antenna element with dimensions and distance from neighboring ones in the order of half the incident wavelength. For this reason the reciprocal couplings between antenna elements can be neglected and each one is designed independently from the others. Therefore the RIS realized through antenna arrays (typically patches), can be thought of as a periodic distribution of single

radiating elements on a substrate. Each patch antenna is made from a thin layer of copper, typically rectangular or circular in shape and covered with a corrosion-resistant metal such as gold, tin, or nickel. The underlying substrate, made of dielectric and therefore non-conductive material, serves as a mechanical support to the patch array, and to separate it from any ground plane, when present. The latter is a metal plane connected to the electrical earth which acts as a reflecting surface for radio waves.

The Fig. 1.3 shows an example RIS implementation conceived by researchers at the Massachusetts Institute of Technology (MIT).

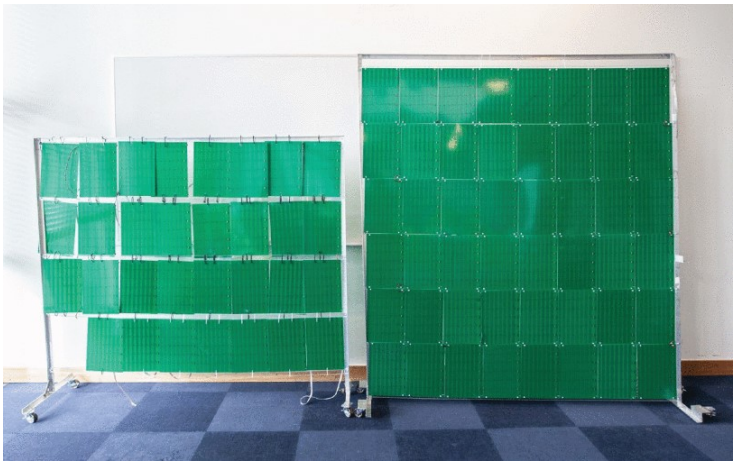

Figure 1.3: The MIT RFocus prototype [1]

This prototype consists of more than 3700 antennas, each costing a few cents, aligned over an area of about 6 square meters. These antennas can be configured, through low power electronic circuits, in such a way that the RIS is able to reflect the incident waves towards desired directions or focus them towards a specific position.

Metasurfaces

A metasurface is defined as any man-made smart surface with unconventional characteristics. In a metasurface, each cell is usually made up of full or slotted patches, straight or curved strips etc, also of different size and shape, arranged in a configuration such as to allow a specific function. Photolithography, electron beam lithography and focused ion beam lithography are just some of the technologies for modeling the nanostructures of a metasurface [4]. The structure and geometry of a cell, as well as the inter-coupling of elements within it, are designed by applying the desired boundary conditions while solving Maxwell's equations using a full-wave simulator [1]. Typically the size of these elements and the mutual distance are lower than the wavelength and this, if on the one hand forces to keep the reciprocal couplings under control through

appropriate engineering, on the other hand allows to consider the metasurface locally homogeneous and therefore be able to describe it through analytical models. That is, the electric and magnetic currents induced by EM fields impinging on the metasurface can be described using effective (surface) macroscopic parameters as the electrical permittivity and magnetic permeability of volumetric metamaterials, and which depend on geometry and cell spacing, material properties and wavelength. Knowledge of these allows to formulate the EM response of the metasurface (in terms of wavefront, angle of incidence and polarization) to the incident EM wave. In particular, the EM polarization density describes how the material responds to an applied EM field and how the material modifies it. The macroscopic description of the metasurface can be obtained by expressing this polarization density in terms of susceptibility tensors. An equivalent method to macroscopically describe the metasurface, is to use the surface impedances. In fact, if we consider for example a metasurface that acts as a reflector, the reflection coefficient is based on the surface impedances [1]. The latter is an important parameter for the design of metasurfaces, given that it provides a local response of the metasurface to the incident wave.

Even the artificial materials constituting these surfaces are studied ad hoc. In fact, the term "meta" indicates the fact that such artificial surfaces have characteristics that materials in nature do not have. These properties are due not only to the composition of these materials, i.e. structures of common materials such as glass or metals, bit also to their geometric arrangement and to their periodicity or non-periodicity. Typically having a thickness and in general reduced dimensions, metasurfaces (2D) are more easily implementable than metamaterials (3D) which are instead more complex to manufacture. A metasurface can also be made using reconfigurable materials that allow reconfigurability, thus avoiding the need for tunable elements, but exclusively for the configuration network to guarantee control from the outside.

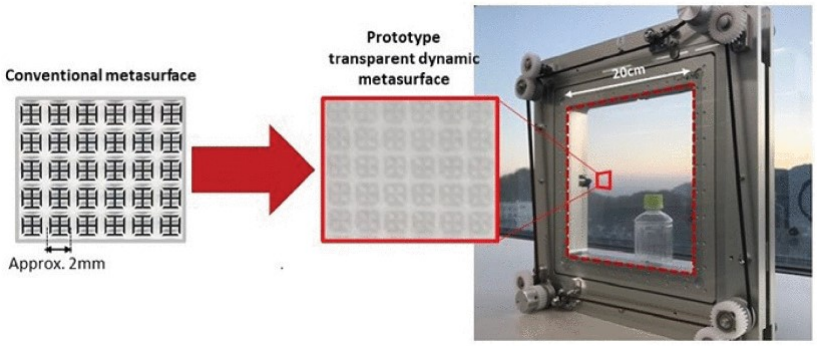


Figure 1.4: Smart glass prototype designed by researchers at NTT DOCOMO, Japan [1]

In Fig. 1.4 an example of metasurface is shown, in particular a prototype of smart glass, made in Japan [1] by arranging in a nearly-periodic way small elements (with

dimensions even lower than the wavelength) of engineered artificial material on a surface covered by a glass substrate. By moving the glass substrate it is possible to control the response of the metasurface to incident radio waves, allowing transmission, partial reflection or total reflection.

### 1.2.3 Principles of RIS operation

Typically, the operation of a RIS consists of two phases: the first is an initialization and control phase where the data collected from the environment with the intelligent surface, are used for its configuration. This is followed by the actual operational phase of the RIS, in which the transmission in the communication system in which it is located assists in order to increase the quality of the signal to the receiver.

The basic element of the study of the EM waves transformations is the Surface Equivalence Principle (SEP). According to this principle, the EM field generated on the RIS, is uniquely determined by the electric and magnetic currents on the surface, which are in turn induced by the incident EM wave. Through the Huygens-Fresnel principle it is possible to analytically quantify this EM field, to know the power of the scattered signal. Huygens' principle states that through a field discontinuity generated by electric and magnetic currents induced on the surface (according to Maxwell's equations), the incident EM field can be transformed into the EM field related to the desired reflected/refracted wavefront [1]. Namely, the interaction between the fields originating from the surface currents and the incident one generates the desired output EM field. Field discontinuity at the interface can be generated by two methods: one active and one passive. The active one foresees that the surface currents are physically generated through the use of external sources such as orthogonal electric and magnetic dipoles, the passive one instead is based on designing the structure, in particular the electric and magnetic properties (i.e. the surface impedance), in order to obtain, following the excitation of the incident wave, the boundary conditions of the desired surface currents. The latter method is much more beneficial in terms of power consumption.

So, an EM wave that affects a surface, induces surface currents, which are determined by the properties of the material constituting the surface, its thickness and the wavelength. These surface currents influence, in turn, the characteristics of the EM waves scattered from the surface. Therefore, if we consider a conventional surface (such as a generic wall), the surface currents and consequently the diffuse waves generated cannot be modified, because they depend on the fixed characteristics of the wall. In a smart surface instead, by adequately designing the boundary conditions at the surface interface, in particular the material of the cells, the space between one and the other, their size and the state of the electronic circuits of the configuration network, the way in which EM waves are scatterred can be controlled. That is, the surface currents that are generated are typically different because they depend on the properties of the surface itself, designed with ad hoc characteristics to suitably modify the reflected or diffracted

waves. The interaction between RIS and the incident wave can be described through beam optics, for which an EM wave can be represented as a beam of rays with variable phases depending on the length of the optical path. When the EM wave strikes the RIS, the phase of each ray that composes it varies according to the refractive index or the reflection coefficient of the surface material, which in turn depends on the surface impedance in that point. The desired wave is obtained if the phases of the rays are such that by combining with each other they generate a reflection/refraction along the desired direction.

In Fig. 1.5 a conceptual example of a metasurface with twenty cells is shown, each with an independently tunable phase. For simplicity, several incident rays are represented when in reality there is only one, while the reflected rays are all oriented towards the same direction.

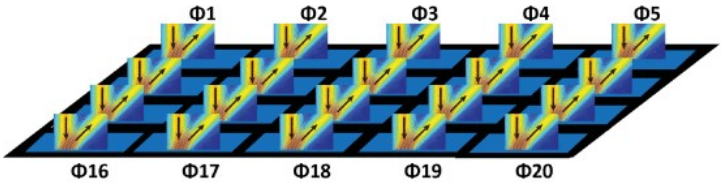


Figure 1.5: Metasurface with twenty cells, each can introduce a tunable phase shift [2]

Shalaev and Capasso were the main pioneers in the study of the amplitude and phase control of the incident wave via suitably engineered metasurface. Their study has given rise to continuous research, especially in the field of RF and microwave wireless communications. The limit to the obtainable specifications was set by the characteristics of the materials that made up the metasurfaces, but with the advent of nanometric manufacturing technology it was possible to start synthesizing metamaterials with the desired properties, in particular conductivity, permeability, permittivity are the most influential ones on the behavior of the waves in contact with the surface.

RIS reconfigurability

A RIS, depending on its characteristics, is classified as: static or reconfigurable. A reconfigurable or dynamic RIS is an intelligent surface that can be reconfigured to perform different EM wave processing. Typically a reconfigurable RIS has the advantage of adapting to changes in the environment but to do so it needs electronic circuits that must be powered. In any case, this power supply is only necessary during the configuration phase, which is why RISs are indicated as nearly-passive devices, in fact they do not radiate their own power (i.e. they are not active). Conversely, a static RIS, whose operation cannot be modified once installed, due to the lack of circuits suitable for this function, is typically passive. This allows it to have practically zero energy consumption,

at the expense, however, of being very limiting in highly dynamic scenarios.

Each cell of the RIS works by capturing the incoming wave and retransmitting it with a certain phase. The term intelligent refers to the fact that the RIS can be reconfigured in order to change its phase response. The reconfigurability of a RIS is ensured through thermal, mechanical or electrical mechanisms. The most widespread approach is the electric one, through the use of low-power electronic circuits that create two types of control: one analog and one digital. Analog control is typically accomplished through the use of varactor diodes or Barium-Strontium-Titanate (BST) capacitors or liquid crystal technology. The latter two, being more complex to implement, are more difficult to find, therefore the analog control is usually achieved through a technology based on varactor diodes. Each cell can be modeled as an equivalent circuit (Fig. 1.6 ) with some equivalent load impedance ($Z_l$ in this example) [3]. The latter can be controlled through the use of a varactor diode, a semiconductor diode with the characteristic of varying the capacitance as the bias voltage varies. Therefore by acting on the bias voltage of the varactor diode (through an external microcontroller for example), the impedance is modified, which in turn affects the reflection or refraction coefficient which models the reflection or refraction characteristics of the cell.

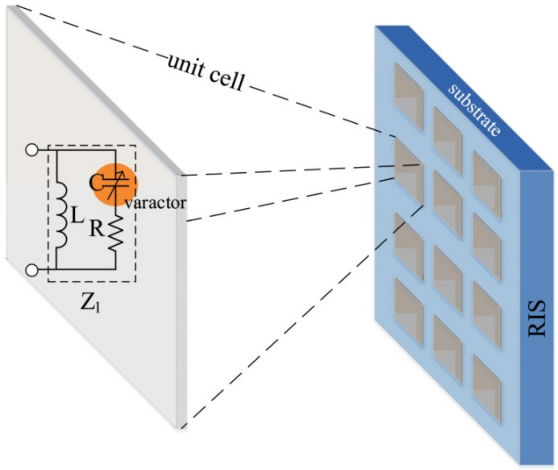


Figure 1.6: Schematic drawing of the cell of a varactor diode-based RIS [3]

Digital control, on the other hand, can be done via MEMs switches or relays, but most of the time PIN (Positive-Intrinsic-Negative) diodes are used due to the reduced RF insertion losses, in addition to the fact that they are small, light and low cost (Fig. 1.7).

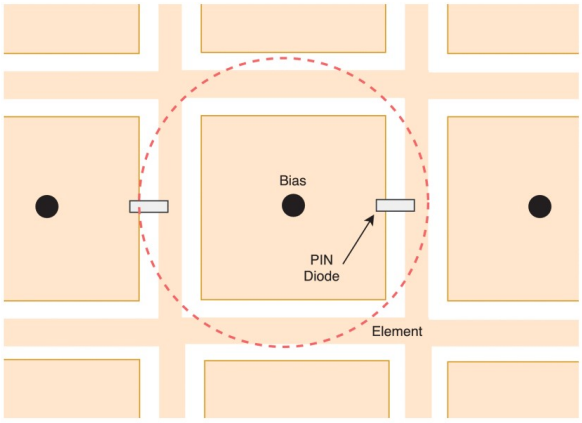


Figure 1.7: Example of RIS with PIN diodes embedded in the connecting metal parts of each surface element, which are switched by an external bias. In this specific case, when they are off, they let the incident wave pass, when they are on, they reflect it [2]

PIN diodes are managed by the control unit, such as a microcontroller or FPGA. In fact, the latter, by sending digital signals, are able to switch the PIN diodes (on/off state) and modify the phase properties of the surface. In digital control, the phase of the surface is quantized and a binary coding is associated with this, with as many bits as the obtainable phases. We assume, for example, that the cells of a RIS are designed to rotate the phase of the incident EM wave by 0, 90, 180 and 270 degrees. The phase can then be controlled by encoding it in the four possible states of two PIN diodes, configurable via just two bits. An n-bit coding of the RIS requires n input polarization lines to each cell, consequently the greater the bits required and the more complex the implementation of the PCB but also the higher the gain of the antenna and vice versa. The Tab. 1.1 shows an example of 1, 2 and 3 bit phase digital coding [5].

| **Bits phase** | **0°** | **45°** | **90°** | **135°** | **180°** | **225°** | **270°** | **315°** |
|---|---|---|---|---|---|---|---|---|
| 1 bit | 0 | | | | 1 | | | |
| 2 bit | 00 | | 01 | | 10 | | 11 | |
| 3 bit | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

Table 1.1: Digital phase mapping for a coded RIS [5]

Types of RIS operations

A RIS can be used, depending on the purposes, in several ways. One of these is focusing, in which each element of the surface introduces a controlled phase shift, in order to focus the signal towards a specific point. In this case, therefore, the RIS behaves like a lens, rather than a mirror. Collimation, on the other hand, is the complementary function of focusing. In Fig. 1.8 we notice the difference between the two and also other two functions: polarization and splitting. The first consists in modifying the polarization of the incident radio wave and generating a reflected wave with a different polarization; the second consist to create more reflected or refracted radio waves for a given incident radio wave [1].

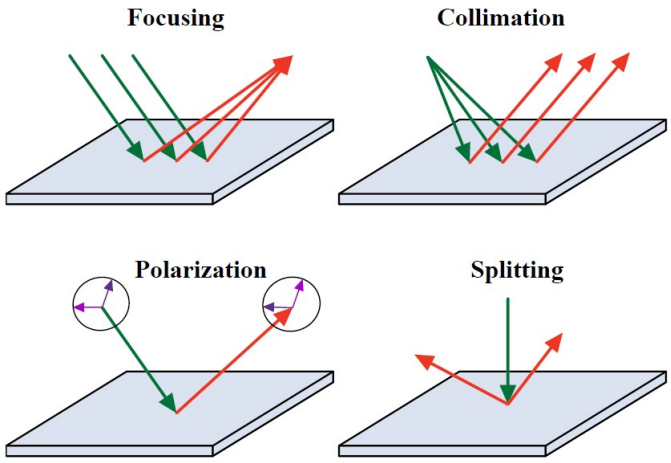


Figure 1.8: Four different functions of a RIS

Another mode of RIS operation, which is being studied, is the scattering for the reduction of the radar cross-section (RCS) [5]. That is, since the Second World War, a way has been sought to make RIS transparent to EM waves so as not to be detected by enemy radar. For this purpose, two hypotheses are being evaluated: to look for suitable materials capable of absorbing EM waves or to suitably size the surface to scattering the energy of the incident waves (a sort of generalized splitting).

Beamsteering is another way of using RIS, which allows to reflect waves incident on it towards specific directions (Fig. 1.9).

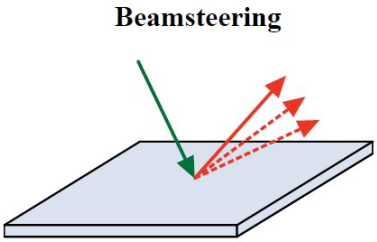


Figure 1.9: Beamsteering function of a RIS

Mostly, it is used in scenarios where there are multiple users, located in different positions, to try to serve each of them simultaneously and efficiently. Transmitters equipped with phased array antennas capable of generating multiple beams are used, which the RIS then reflects in the desired directions. Typically high gains are obtained, but the necessary hardware is very complex and expensive due to the presence of RF chains of attenuators, phase shifters and power amplifiers. However, these functions can be achieved only by controlling the reflection phase. For the control of the reflection amplitude, however, it is necessary to control the generation power of the multiple beams. So in this case, RISs operate as anomalous reflectors. In fact, unlike a uniform conductive surface with a constant surface impedance that specularly reflects the incident plane wave (i.e. with an angle of reflection, with respect to the surface normal, equal and opposite to that of incidence), a RIS, with suitably characterized surface impedance, can introduce a different reflection from the specular one, i.e. anomalous. To generate it, it makes use of a phase gradient that typically varies linearly along the surface. To reduce the complexity of the hardware, solutions are being studied which envisage the use of frequency selective metasurfaces, i.e. which, depending on the frequency of the signal affecting them, provide a different departure angle, so as to be able to serve users in different directions. Knowing this, an OFDM signal can then be used to serve multiple users simultaneously.

### 1.2.4 RIS applications

The main fields of applications based on RISs are [3]:

- RIS-enhanced cellular networks;

  In particular RISs are used to try to: bypass obstacles between users and the base station, improve the quality of the signal received from a user on the cell board, reduce interference between neighboring cells and also improve users safety, being able to be configured to worsen the signal detected by any interceptors. A promising use case is to introduce RISs, employed as reconfigurable reflectors, in Non-Line-of-Sight (NLoS) scenarios [7], to make the wireless link more robust and for example provide sufficient coverage to cell-edge users. It is mainly being

studied in high-frequency transmission systems, for example in the millimeter-wave spectrum. The introduction of RISs can reduce dead coverage zones, typical in urban scenarios of large cities such as alleys or underground metro stations. RISs can also support radio localization applications, exploiting their properties to estimate the positions of mobile terminals for example.

- RIS-assisted indoor communications;

  By covering the internal domestic walls with RISs, wireless connectivity can be guaranteed for the operation of all devices (tablets, mobile phones, etc.), even in the most blind spots of the house [1]. From this point of view, RISs can introduce important improvements in work or in school and university environments. RISs can also be introduced inside planes and trains to improve signal coverage and expose passengers to lower EM radiation. Especially inside hospitals, where the intensity level of EM waves must be kept under control, RISs can improve local coverage without requiring higher transmission power [1].

- RIS-enhanced unmanned systems;

  RISs can be used to improve wireless communications to autonomous vehicular networks, intelligent robotic networks, drones, etc. For example, RIS can be implemented in vehicle-to-infrastructure assisted autonomous driving (V2I) systems. In this type of network, roadside base stations (RBSs) send real-time traffic data to the BSs, which send them to autonomous vehicles (AV). The introduction of RISs in this scenario, for example on building facades, advertising boards, vehicle windows and even on pedestrians' clothing, allows to improve the performance of vehicular networks trying to improve the AV driving safety [3].

- RIS-enhanced Internet of Things (IoT) networks;

  RISs are introduced into smart wireless sensor networks, smart agriculture, or smart factories. For example, IoT devices sometimes need a lot of energy to transmit the data collected from the environment in which they are placed. Thanks to RISs, the detected data can be transported via reflected signals with virtually no energy cost [3]. As far as smart factories are concerned, however, RIS can exploit the multiple reflections caused by the various metal objects present inside the factories to improve coverage and transmission speed [1].

Fig. 1.10 shows examples of these applications based on the use of RISs.

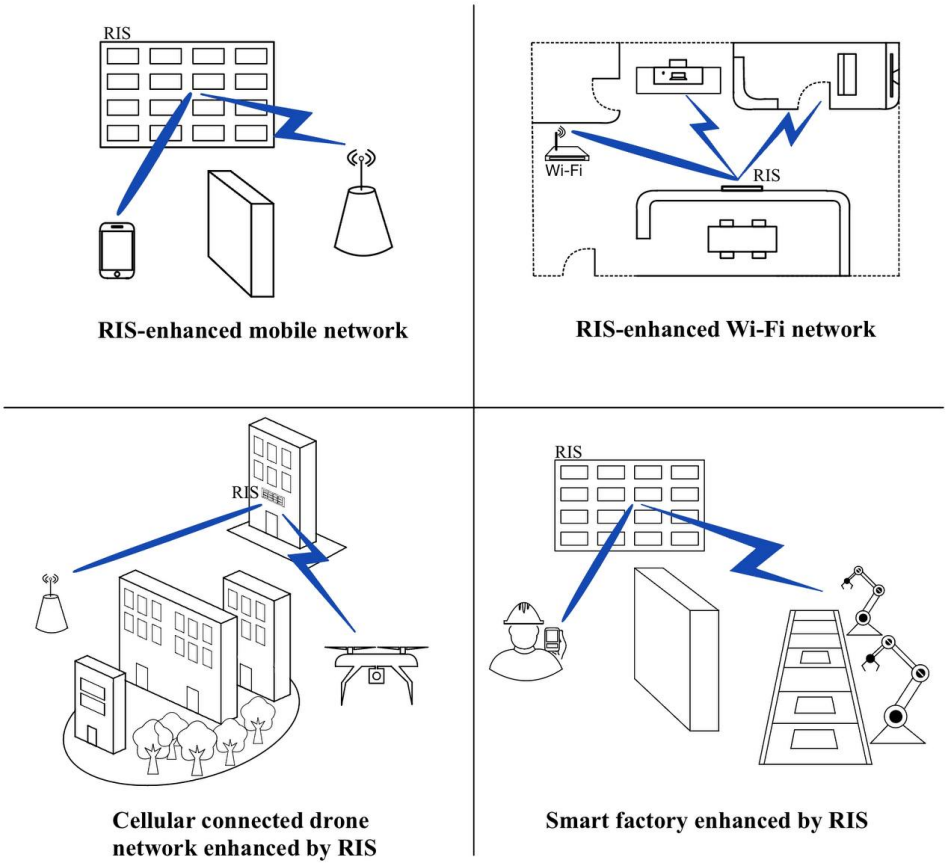


Figure 1.10: RIS applications examples

### 1.2.5 RIS limits and introduction of a potential solution

RISs represent an important technology for next-generation communication systems, but they also have some disadvantages, including that of necessarily having to be powered and hence not always being possible or convenient in certain scenarios. Furthermore, it is necessary to estimate the CSI (Channel State Information), in fact the optimal configuration of the reflection coefficients of a RIS (i.e. on how the RIS reflects the EM wave that hits), is a function of the channel state between the RIS and the base station and between the RIS and the mobile user. The CSI estimation may require complexity and additional costs due, for example, to any hardware to perform channel sensing. Furthermore, a RIS should be able to reconfigure itself in a very short time, following the dynamic changes of the propagation environment. Basically for each channel change, the CSI must be estimated and it is necessary to calculate, based on this, the reflection coefficients and send them to the RIS through a dedicated control channel. For high dimensional channels, it may be necessary to estimate thousands of channel coefficients and

this can involve a control channel with excessive signal overload, especially in dynamic networks. These steps can take a long time, particularly due to the long training time (especially in multi-user MIMO systems) which may be intolerable in dynamic scenarios [6]. The speed of RIS reconfiguration is still an open challenge. These disadvantages could penalize the use of RIS for certain applications, but also lay the foundations for introducing a possible alternative: the metaprism, i.e. a particular low-cost, passive and non-reconfigurable metasurface that can act as a metamirror to improve the quality of the wireless link, without power consumption [6].

# Chapter 2

# User's location estimation using metaprism

## 2.1 Introduction

In the first part of this chapter, the metaprism will be described, particularly a possible model and an idea of physical implementation. The second part of the chapter instead takes into consideration, first of all, a typical metaprism usage scenario, in which its function is explained in a communication system and that includes a transmitter, the metaprism as a meta-mirror and a receiver. From this typical scenario, by reversing the roles of transmitter and receiver, we obtain the scenario analyzed for the implementation of the localization application. In fact, assuming that the user transmits from an unknown position, a way to locate it will be described, starting from the reflection introduced by the metaprism, the signal received by the BS and using a maximum likelihood estimator obtained through the steps described in the last part of the chapter.

## 2.2 Metaprism

The metaprism is a frequency-selective metasurface, i.e. with reflective properties depending on the signal's frequency, completely passive and non re-configurable. Thanks to these two characteristics, which differentiate it from common RIS, it is possible to make the metaprism work without having to interact with it or having to estimate the CSI. The metaprism can be used as a reflector to extend and / or improve the coverage of short-range low-cost wireless networks. Particularly, through an appropriate design of the metaprism, it is possible to direct the reflected signal towards a specific position (for example in the direction of a receiving user) and significantly reduce the path-loss by playing with the frequency of the impinging signal [6]. Since the reflection that introduces the metaprism is dependent on the frequency of the signal that affects it, it is

advantageous to study its behavior in the presence of an OFDM signaling. In this case, in fact, each sub-carrier of the signal incident on the metaprism will be reflected towards a certain direction dependent on its central frequency, creating an overall effect similar to that of white light passing through a prism, from which the metaprism takes its name (Fig. 2.1).

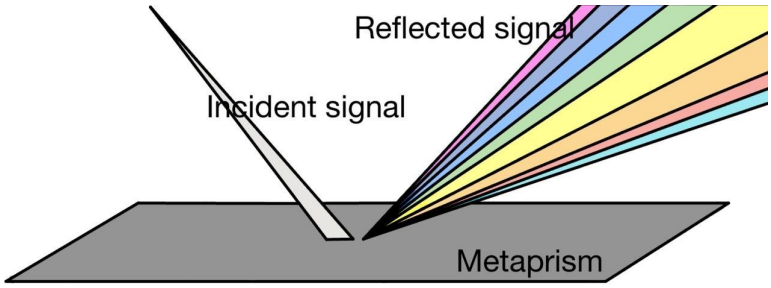


Figure 2.1: OFDM incident signal and reflection introduced by the metaprism. Each different color of the reflected signal represents a sub-carrier.

By properly designing the metaprism, particularly its phase profile, it is possible to guide the signal towards a user, according to the sub-carrier assigned to him, ensuring an improvement of the wireless link and this is especially useful in NLoS (Non-Line-of-Sight) situations [6]. Depending on whether users are in the near or far-field region, the metaprism design criteria changes. Obviously, the scattered signal from the surrounding walls may also be sufficient to cover a certain position in NLoS channel conditions, but often the received Signal-to-Noise-Ratio (SNR) is insufficient. In addition to increasing the SNR thanks to its reflective properties, the metaprism also has other advantages, such as: low cost, no energy consumption and compatibility with existing wireless standards, which make it a valid alternative to RIS.

In the next section, a possible model for frequency-selective metasurfaces, applicable also to the metaprism, is described.

## 2.3 Metaprism model

We can think of a metasurface as made of specific metamaterials and composed of several cells. In Fig. 2.2 it is possible to see the 3D model of the metaprism and the equivalent model of a single cell.

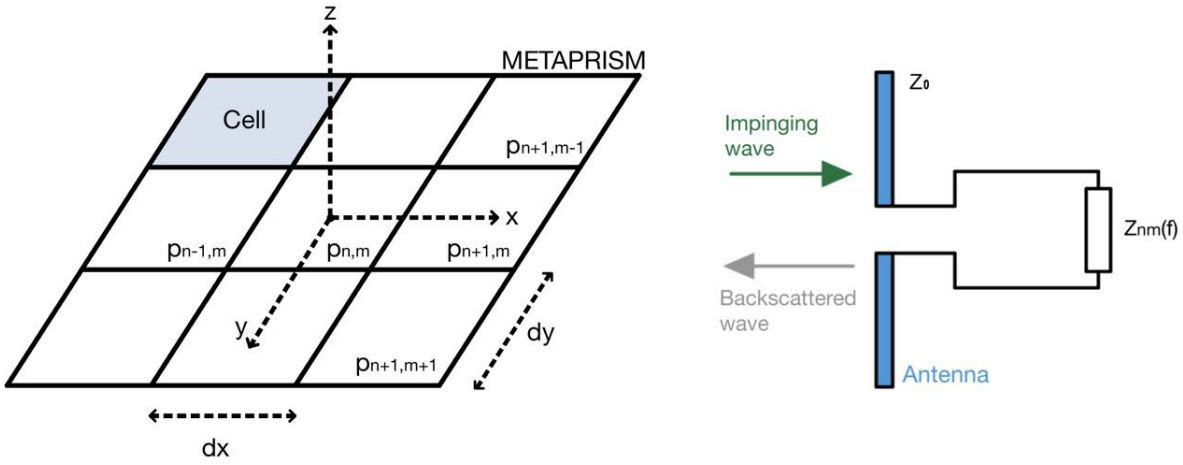


Figure 2.2: On the left the metaprism 3D model and on the right the cell equivalent model

The cells of a metasurface can be seen as small radiating elements with adjustable load impedance. With reference to Fig. 2.2 we consider a metasurface in the $xy$ plane with $N \times M$ cells, each of dimension $dx \times dy$ and with coordinate center $p_{nm} = (x_n, y_m, 0)$, where $x_n = n\,dx - N\,dx/2$ with $n = 0, 1, ..., N-1$ and $y_m = m\,dy - M\,dy/2$ with $m = 0, 1, ..., M-1$. The center of the metasurface is at the coordinates $p_0 = (0, 0, 0)$. The size of the total surface is $L_x \times L_y$, with $L_x = N\,dx$ and $L_y = M\,dy$.

A fairly general equivalent model of the $nm$th cell of the meta-surface in $p_{nm}$ position consists of a radiation element (antenna) above a ground screen with a load impedance $Z_{nm}(f)$ dependent on the cell and on the frequency $f$. The ground screen is used to reduce the effects of the earth whose conductivity is uneven and poor. This also increases the directivity of the antenna. The impedance is designed in such a way as not to be adapted to the impedance $Z_0$ of the antenna, thus causing a reflected wave (back-scattered wave) radiated back by the radiant element. This is how the reflection, introduced by the metasurface, is generated. The reflection characteristics are summarized in one parameter: the reflection coefficient, one per cell. The reflection coefficient (frequency dependent) of the $nm$th cell, in the presence of an incident plane wave with 3D angle $\varphi = (\varphi_{xy}, \varphi_{xz})$ and observed at $\vartheta = (\vartheta_{xy}, \vartheta_{xz})$ angle is:

$$r_{nm}(\varphi, \vartheta, f) = \sqrt{F(\varphi)\,F(\vartheta)}\,G_c\,\Gamma_{nm}(f) \tag{2.1}$$

with *F(ϑ)* the normalized power radiation diagram, expressed by the following equation:

$$F(\vartheta) = \begin{cases} cos^q(\vartheta_{xy}), & \vartheta_{xy} \in [0, \pi/2],\ \vartheta_{xz} \in [0, 2\pi] \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

where $q$ is a parameter that depends on the cell size and the technology adopted and can be related to the boresight gain $G_c$ of the antenna by the expression: $G_c = 2(q+1)$. If

we assume an ideal radiation efficiency, $G_c$ can be described by the following expression:

$$G_c = A_c \frac{4\,\pi}{\lambda^2} \tag{2.3}$$

where $A_c = dx \times dy$ that is the cell geometric area.
$\Gamma_{nm}(f)$ is the reflection coefficient of the load, given by the following equation:

$$\Gamma_{nm}(f) = \frac{Z_{nm}(f) - Z_0}{Z_{nm}(f) + Z_0} \tag{2.4}$$

Through an adequate design of the impedance $Z_{nm}(f)$ of each cell, a metasurface with different reflective characteristics can be realized. Equivalently, always for an incident plane wave, the $nm$th cell reflection coefficient can also be expressed by the following equation:

$$r_{nm}(\varphi, \vartheta, f) = \beta_{nm}(\varphi, \vartheta, f)\ \exp(j\Psi_{nm}(f)) \tag{2.5}$$

where: $\beta_{nm}$ is the amplitude of the reflection and $\Psi_{nm}(f)$ is the phase of the reflection introduced by the $nm$th cell and described by the following expression:

$$\Psi_{nm}(f) = \alpha_{nm}(f - f_r) + \gamma(f) \tag{2.6}$$

i.e. it is given by a coefficient $\alpha_{nm}$ and a factor $(f - f_r)$ linearly dependent on the frequency $f$ (a value within the signal band $W$) and where $f_r$ is a reference frequency which could be the center frequency $f_0$ of the signal or the signal's lower-band frequency $f_0 - W/2$. The third term $\gamma(f)$ adds a phase shift common to all the cells, but not depending on the position, it is irrelevant. Therefore, even if the incident wave arrives in phase to all the array elements (angle $\varphi$) for the plane wave hypothesis, each cell introduces a phase shift which depends on the frequency and on the coefficient $\alpha_{nm}$. The design criteria of this coefficient changes whether we consider users in the near-field or far-field region of the metaprism. Subsequently, a possible model of coefficients $\alpha_{nm}$, in both cases, will be presented.

## 2.4 Possible physical implementation and use of the metaprism

As regards the implementation of the metaprism, a possible technology under study by the University of Siena, is based on a metasurface divided into two parts. The first part is the one that receives the signal from the trasmitter (for example a base station BS) and transforms it into a surface wave. This wave is directed towards the other part of the metasurface, which, unlike the first, is designed to transform the surface wave into a transmitted wave. The idea is to make sure that this surface wave is frequency-dependent

so that, depending on the frequency and how the surface wave affects the metasurface, it is reflected in a different way. The Fig. 2.3 better represents what happens.

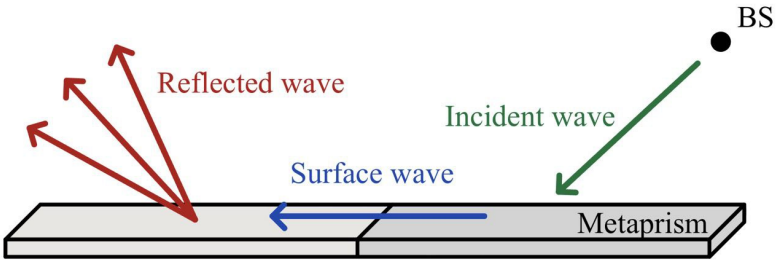


Figure 2.3: Possible metaprism implementation: behavior of the metasurface in terms of waves

The advantage of this implementation idea is that, compared to having one metasurface, here the receiving part is separated from the transmitting part. In fact, the disadvantage of typical metasurface implementations is that both the base station and the users have to see it to ensure the right operation; while, in this way, the base station can only see the first part of the metaprism and users only the other part, ensuring a certain degree of flexibility at deployment level.

On a physical level these metasurfaces, which are similar to panels positioned in the walls of buildings, even if they can reach dimensions close to the square meter, in reality in the context of localization (in particular in 2D, for example of objects that are on the floor), it is more convenient to give them a stripe shape, i.e. to exaggerate in one dimension rather than in two. This is because a strip form has less chance of being obscured by obstacles than a square form.

This type of technology can be very useful in an Industrial IoT scenario, for example. This is because, in addition to being a controlled scenario, from the point of view of propagation it is very challenging due to the possible high presence of metal. In fact, for communications that take place where there is a lot of metal, having good coverage in certain points or wanting to locate with high precision, is very difficult and often not feasible without increasing the number of base stations. An alternative is to use the metaprism, a sort of film wall mountable, passive and therefore low-cost.

However, this is a fairly complicated situation, so to explain how metaprism works within a communication system, let's consider a simpler scenario like the one described in the next section.

## 2.5 Typical metaprism based scenario

As a typical scenario, we firstly consider the one represented in Fig. 2.4, i.e. an example of a wireless down-link BS-users system, based on OFDM signaling and improved by the

introduction of the metaprism.

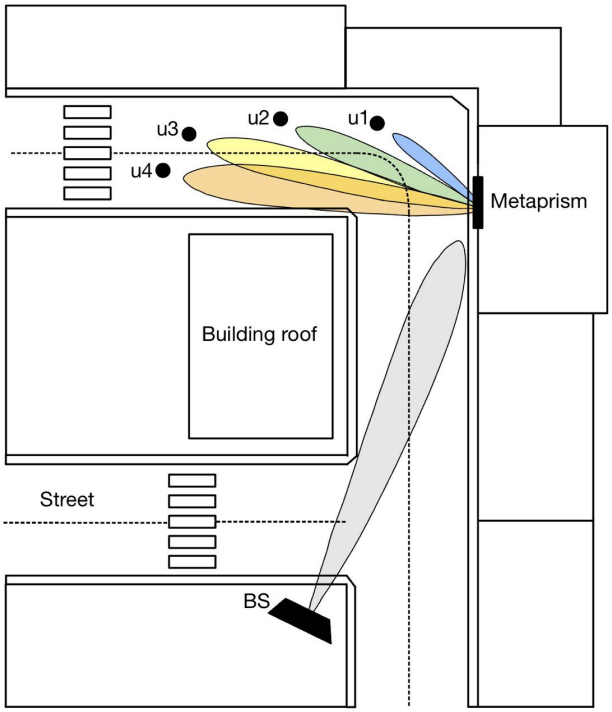


Figure 2.4: Typical scenario in which the signal transmitted by the base station is reflected by the metaprism towards four users, each with a single antenna.

In the scenario considered, the base station, positioned for example on the roof of a building, acts as a transmitter, while the metaprism, typically fixed on the buildings wall, acts as a metamirror reflecting the signal towards the users ($u_1$, $u_2$, $u_3$, $u_4$) that are in the street. As previously said, we consider an OFDM signal as transmitted signal.

### 2.5.1 OFDM signalling

OFDM (Orthogonal Frequency Division Multiplexing) is part of a family of techniques called multi-carrier techniques. An OFDM signal is typically obtained starting from a signal with broadband $W$ and using a serial-parallel converter (S/P converter) according to the scheme in Fig. 2.5.

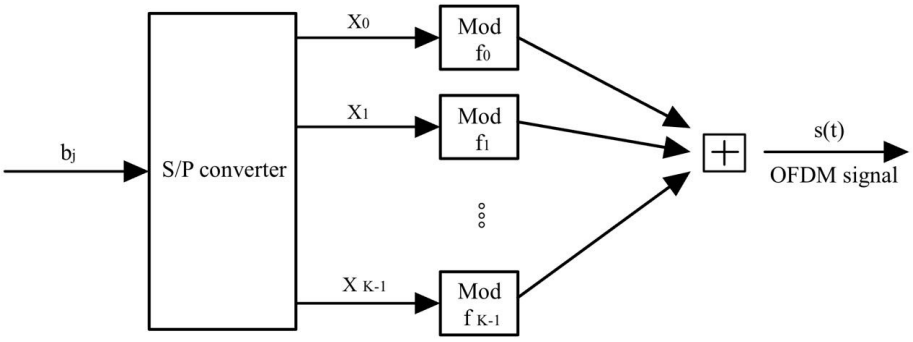


Figure 2.5: By means of an S/P converter and K modulators, the bit stream is converted into an OFDM signal

The S/P block converts the bit stream $b_j$ (with bit rate $1/W$) into $K$ bit streams in parallel. In general, inside the S/P converter there can also be a mapper (typically M-QAM), so $x_k$ can represent complex values taken from a constellation (for example M-QAM symbols). Each of these flows is followed by a modulator, designed so that the carrier frequency of the output signal to each modulator has band $\Delta f = W/K$. Then, we can say that $\Delta f$ represents the spacing between two adjacent sub-carriers. There are, therefore, many signals in parallel, each with a narrow band ($\Delta f$): the signal generated by the first modulator for example will have a carrier frequency $f_1$, the signal generated by the second modulator will have a carrier frequency $f_2$ etc. The OFDM signal is given by the sum of all the $K$ signals into which it can be decomposed.

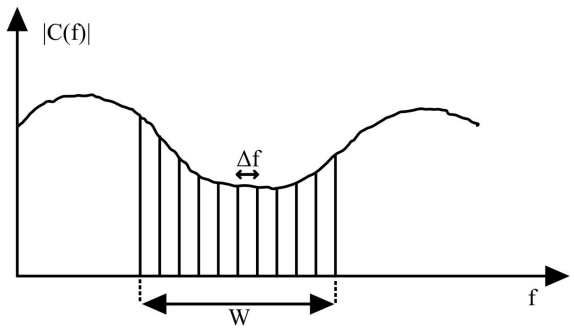


Figure 2.6: OFDM signal spectrum

From the spectral point of view, as can be seen from Fig. 2.6, the $K$ sub-carriers occupy about the same band that they would occupy using a conventional scheme but not with only one carrier.

### 2.5.2 Channel gains and received signal

Returning to the example of Fig. 2.4, we assume a number of sub-carriers $K \geq U$, with $U$ the number of users considered, and sub-carriers orthogonal to each other, so that the receiver is able to separate them without mutual interaction. Given the user $u = 1,2, ..., U$, we indicate with $x_k \in C$, its information symbol transmitted in the generic OFDM frame. Each user is uniquely associated with a sub-carrier $k$, with frequency described by:

$$f_k = f_0 - \frac{W}{2} + k\Delta f \tag{2.7}$$

where $k = 1,2, .., K$ and $f_0$ the central frequency of the signal.

If $W/f_0 \ll 1$, the channel gain between the transmitter BS and the $nm$th cell of the metaprism for the $k$th sub-carrier is given by the following equation:

$$h_{nm,k}(BS) = \frac{\sqrt{G_t}\,\lambda}{4\,\pi\,|BS - p_{nm}|} \exp\left(-j\frac{2\pi\,f_k}{c}\,|BS - p_{nm}|\right) \tag{2.8}$$

where $G_t$ is the transmitting antenna gain, $\lambda = \mathrm{c}/f_0$ is the signal wavelength, with $c$ the light speed. The norm of the difference (in absolute value) between the base station position ($BS$) and the $nm$th cell position ($p_{nm}$), indicates the distance between the two. The channel gain from the $nm$th cell of the metaprism to the receiver (user) in position $u$ is:

$$g_{nm,k}(u) = \frac{\sqrt{G_r}\,\lambda}{4\,\pi\,|u - p_{nm}|} \exp\left(-j\frac{2\pi\,f_k}{c}\,|u - p_{nm}|\right) \tag{2.9}$$

where $G_r$ is the receiving antenna gain.

The signal received by the user at the $k$th sub-carrier is given by the following summation equation:

$$y_k = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} h_{nm,k}(BS)\,r_{nm,k}(\varphi,\vartheta)\,g_{nm,k}(u)\sqrt{P_t}\,w_k\,x_k + n_k \tag{2.10}$$

where $w_k\,x_k$ is the product between the transmitted symbol $x_k$ and a weight $w_k$ and represents the fraction of the total power $P_t$ allocated to the $k$th sub-carrier. $n_k$ is a Gaussian random variable with zero mean and variance $\sigma^2$ that models the thermal noise introduced by the entire channel. $r_{nm,k}(\varphi,\vartheta)$ is the reflection coefficient of the metaprism at frequency $f_k$, $\vartheta$ is the angle relative to the user's position $u$, $\varphi$ is the angle of arrival (AOA) wherewith the transmitted signal hits the metaprism, both valued relative to normal to the metaprism.

For greater clarity, these two angles and the reflection angles of the signal components relative to sub-carriers 1 ($\vartheta_{0,1}$) and 2 ($\vartheta_{0,2}$), are shown in Fig. 2.7 where it was considered, for example, only the user $u_2$.

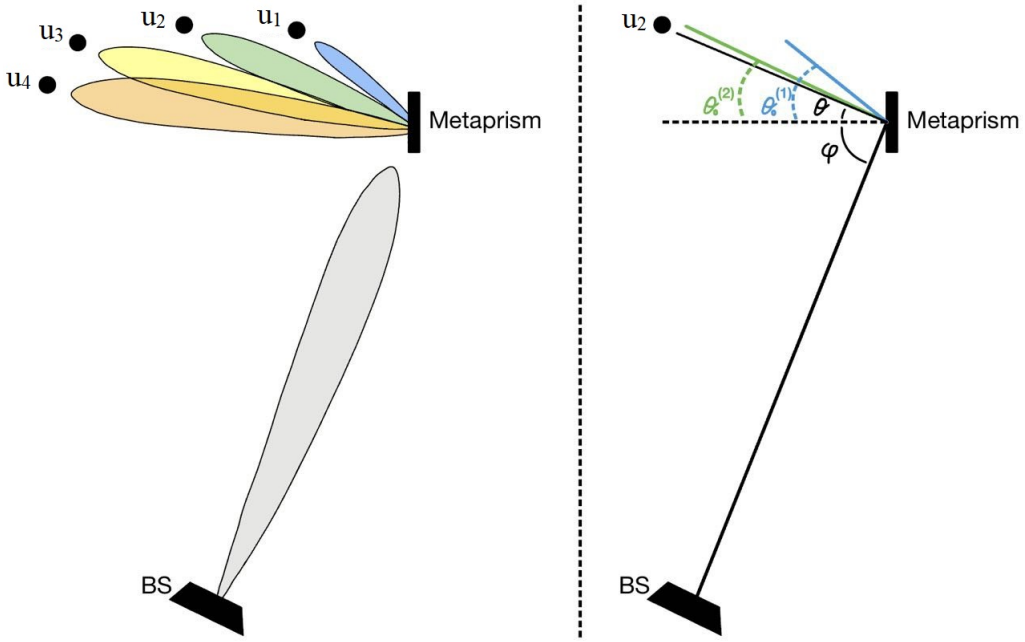


Figure 2.7: View from above of the scenario considered, in which the main 3D angles are highlighted on the right side figure: $\vartheta$ indicates the user's direction, $\vartheta_{0,k}$ indicates the reflection direction of the $k$th sub-carrier

### 2.5.3 Far-field hypotesis and beamsteering design

Let us consider the situation of user and base station, located somewhere in the territory, both in LoS conditions with respect to the metaprism and in particular in the far-field region, i.e. with a distance from the reflecting surface greater than the *Fraunhofer distance*, defined as:

$$d_{Fraunhofer} = 2D^2/\lambda \tag{2.11}$$

where $D = max(L_x, L_y)$ is the surface size. Under the far-field hypothesis, $h_{nm,k}(BS)$ and $g_{nm,k}(u)$, that are respectively the transmitter-$nm$th cell channel gain and the $nm$th cell-receiver channel gain, can be approximated as:

$$\begin{aligned} h_{nm,k}(BS) &= \frac{h_0}{|BS|} \exp\left(j\frac{2\pi}{\lambda}(n\,dx\,u_x(\varphi) + m\,dy\,u_y(\varphi))\right) \\ g_{nm,k}(u) &= \frac{g_0}{|u|} \exp\left(j\frac{2\pi}{\lambda}(n\,dx\,u_x(\vartheta) + m\,dy\,u_y(\vartheta))\right) \end{aligned} \tag{2.12}$$

valid if $\mathrm{W}/f_0 << 1$ and where:

$$\begin{aligned} h_0 &= \sqrt{G_t}\,\frac{\lambda}{4\,\pi}\,\exp\left(-j\frac{2\pi}{\lambda}\,|BS|\right) \\ g_0 &= \sqrt{G_r}\,\frac{\lambda}{4\,\pi}\,\exp\left(-j\frac{2\pi}{\lambda}\,|u|\right) \end{aligned} \tag{2.13}$$

in which the exponential describes the phase shift with respect to the center of the metaprism $p_0$. In this case, we can substitute the gain equations just found in (2.10) and we find the expression of the received signal in the far-field case:

$$\begin{aligned} y_k &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} h_{nm,k}(BS)\, r_{nm,k}(\varphi,\vartheta)\, g_{nm,k}(u)\sqrt{P_t}\, x_k + n_k \\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} \frac{h_0}{|BS|}\exp\left(j\frac{2\pi}{\lambda}(n\,dx\,u_x(\varphi) + m\,dy\,u_y(\varphi))\right)\beta_{nm,k}(\varphi,\vartheta)\exp\left(j\Psi_{nm,k}\right) \\ &\quad \frac{g_0}{|u|}\exp\left(j\frac{2\pi}{\lambda}(n\,dx\,u_x(\vartheta) + m\,dy\,u_y(\vartheta))\right)\sqrt{P_t}\, x_k + n_k \\ &= \frac{\beta_{nm,k}(\varphi,\vartheta)\,\sqrt{P_t}\,h_0\,g_0\,x_k}{|u|\,|BS|}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\exp\left(j\frac{2\pi}{\lambda}\,n\,dx\,(u_x(\varphi)+u_x(\vartheta)) + \right. \\ &\quad \left. + j\frac{2\pi}{\lambda}\,m\,dy\,(u_y(\varphi)+u_y(\vartheta)) + j\Psi_{nm,k}\right) + n_k \end{aligned} \tag{2.14}$$

From the far-field hypothesis, we can suppose the transmitted signal incident as a plane wave, so we can assume equation 2.5 as model of the reflection coefficient (frequency dependent), where the metaprism phase profile of the generic $nm$th cell can be described through the equation 2.6, recalled here: $\Psi_{nm}(f) = \alpha_{nm}(f - f_r)$.

When both the transmitter and receiver are in the LoS far-field region with respect to the metaprism, through a suitable design of the coefficients $\alpha_{nm}$ of the cells it is possible to perform sub-carrier-dependent *beamsteering*. The beamsteering is a technique used in far-field to improve communication efficiency in which tipically the multi-antenna transmitter sends signals to a specific direction through a linear phase shift along the array. In this case we consider a single antenna transmitter but the same thing can be re-created by the various cells of the metaprism, which act as multiple antennas, suitably designing the coefficients $\alpha_{nm}$. The model proposal to describe these coefficients is expressed by the following equation:

$$\alpha_{nm} = a_0\,x_n + b_0\,y_m \tag{2.15}$$

According to this model, which we can call beamsteering design, $\alpha_{nm}$ grows incrementally as the coordinates of the cell and $a_0$ and $b_0$ are two constants to be properly designed.

To find the values $a_0$ and $b_0$, it is enough to equate Eq. 2.6 where $\alpha_{nm}$ is replaced by its expression as in Eq. 2.15 and the following equation:

$$\Psi_{nm}(f_k) = -\frac{2\,\pi\,n\,dx}{\lambda}\left(u_x(\varphi) + u_x(\vartheta_{0,k})\right) - \frac{2\,\pi\,m\,dy}{\lambda}\left(u_y(\varphi) + u_y(\vartheta_{0,k})\right) \tag{2.16}$$

wich represents the phase profile of the metaprism $nm$th cell, at frequency $f_k$, in order, for the signal's component at the $k$th sub-carrier, to be reflected towards a target direction $\vartheta_{0,k} = (\vartheta_{0xy,k}, \vartheta_{0xz,k})$; so that all phasors add up coherently in the direction $\vartheta_{0,k}$. Doing this, we obtain:

$$\begin{aligned} a_0\,(f_k - f_0) &= -\frac{2\,\pi}{\lambda}\left(u_x(\varphi) + u_x(\vartheta_{0,k})\right) \\ b_0\,(f_k - f_0) &= -\frac{2\,\pi}{\lambda}\left(u_y(\varphi) + u_y(\vartheta_{0,k})\right) \end{aligned} \tag{2.17}$$

where:

$$\begin{aligned} u_x(\varphi) &= \sin(\varphi_{xz})\,\cos(\varphi_{xy}) \\ u_y(\varphi) &= \sin(\varphi_{xz})\,\sin(\varphi_{xy}) \end{aligned} \tag{2.18}$$

Furthermore, if we consider the transmitter - metaprism system as an array of equivalent planar antennas, we can represent its frequency-dependent array factor $AF$ (relating to the $k$th sub-carrier) by the following expression:

$$\begin{aligned} AF_k(\vartheta) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} \exp\Bigg(& j\frac{2\pi\,n\,dx}{\lambda}\left(u_x(\vartheta) - u_x(\vartheta_{0,k})\right) + \\ &+ j\frac{2\pi\,m\,dy}{\lambda}\left(u_y(\vartheta) - u_y(\vartheta_{0,k})\right)\Bigg) \end{aligned} \tag{2.19}$$

The array factor of a multiple antenna summarizes the radiative characteristics of the array. The value of this parameter for a certain direction ($\vartheta$ angle), indicates the intensity wherewith the $k$th sub-carrier of the signal transmitted by the base station and incident on the metaprism, is reflected in that direction. Consequently, the array factor allows us to understand the direction of maximum reflection of each sub-carrier and therefore deduce, on the basis of the point (and hence the direction) that we want to reach through the reflection, the sub-carrier to which the base station must transmit. For example, with reference to Fig. 2.7, we can see how, in order to reach user $u_2$ with the maximum intensity of the signal, the BS must transmit at the sub-carrier relative to the radiation pattern represented in green.

### 2.5.4 Near-field hypotesis and random design

Whereas in typical far-field free-space channel conditions the propagation distances from each antenna element to a target receiver are approximately identical (and thus also the

path gains) and the antenna array shares the same angle of arrival/departure, in near-field the above approximations do not apply and the antenna elements have different distances and angles of arrival/departure with the target receiver. Therefore, in the near-field channel condition, each antenna element has different path gain and phase shifts in the link with the target.

In the far-field region the EM wavefronts reaching the receiver are planar, so the transmitter can only control the angle at which to radiate the energy. On the contrary, in the near-field region, the non-negligible spherical wavefront is tipically exploited to focus the radiated energy in a specific spatial position, i.e. not only by angle, but also by a certain distance along the direction of propagation. In this case we no longer speak about beamsteering but of focusing. This one is a transmission technique based on trying to combine the output wave-fronts of each single antenna element in a constructive way in the desired focal point, producing a high power. Figure 2.8 shows the difference between beamsteering and focusing.

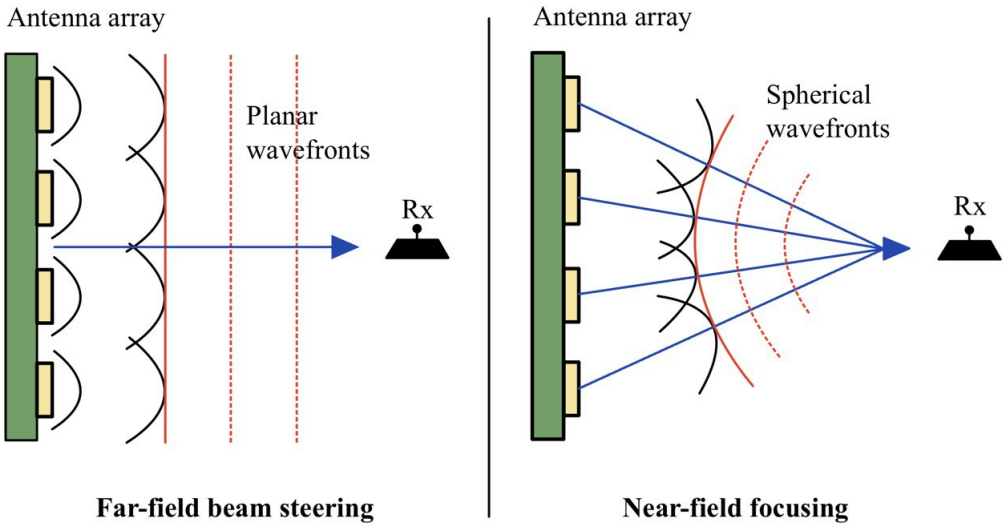


Figure 2.8: On the left the beamsteering tecnique used in far-field and on the right the focusing tecnique used in near-field. Metaprism cells can be seen as an antenna array

For instance, if we consider a metaprism size of 20-30 cm, then we can consider the users in the near-field of the metaprism. If the distance increases until it exceeds the Fraunhofer distance, the receiver enters the far-field region and it is no longer possible to discriminate distances and positions but only directions (angles). So, focusing can be useful when there is interest in evaluating the distances of receivers located in different positions but in the same direction, as often happens in one-dimensional scenarios such as along the streets.

In this case the approximate formulas usable in far-field, are no longer suitable, there-

fore, if we want to study the behavior of the metaprism under the near-field hypothesis, we have to consider the exact formulas of the gains and the received signal previously described in the section 2.5.2, i.e the equations 2.8, 2.9 and 2.10.

The phase profile of the metasurface can still be described by 2.6, in which, however, the design of the $\alpha_{nm}$ coefficients can no longer be the one described by Eq. 2.15 but must be re-adapted to the near-field case. Focusing would require a different project of the metaprism coefficients, certainly more complex and for this reason it will not be studied in this thesis. From this point of view, the proposal is to model the coefficient of each $nm$th cell as a random real number:

$$\alpha_{nm} = random\, real\, number \tag{2.20}$$

For this reason we adopt the nomenclature *random design*. In doing so, however, it is no longer correct to speak of focusing because random $\alpha_{nm}$ coefficients will generate different phase responses from cell to cell and consequently the output wavefronts of each single cell element will not be combined constructively in a specific focal point (Fig. 2.9).

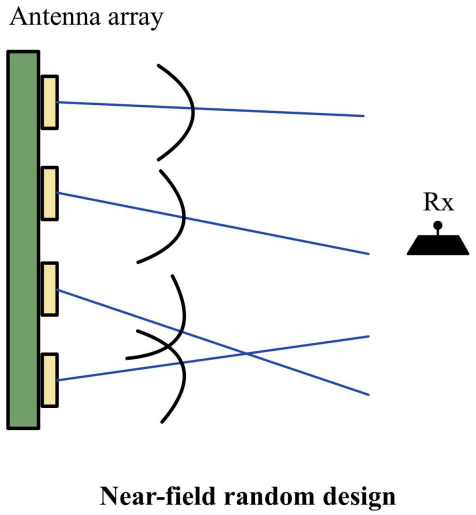


Figure 2.9: Output wavefronts of random design metaprism

So what is the advantage of random design? Setting these metaprism coefficients randomly, ensures that the frequency profile received by the receiver is unique (or nearly so) for each user's position evaluated. Indeed, since the $\alpha_{nm}$ are random, as soon as the user's moves, this received profile changes completely, thus becoming a sort of signature of each position. Obviously we expect very different profiles only under near-field conditions. In fact, in far-field conditions, transmitters who are located along the same direction but at different distances will most likely give rise to approximately the same signature, and when this happens it will no longer be possible to discriminate one position from the other. This is why we will use the random design of the $\alpha_{nm}$ coefficients

under the near-field assumption. However, if a large number of sub-carriers is considered, given the randomness, the probability that two different user's positions have the same frequency profile drops drastically. Furthermore, unlike the beamsteering approach, whereby all the metaprism gain was pointed towards a certain direction (based on the design method), in the random design we can no longer speak of gain of the entire array (being all random), then the gain and consequently the overall link budget will worsen. In this case the solution could be to use a closer receiver or a higher transmit power.

## 2.6 Scenario considered for localization

Through the use of the metaprism and considering a system based on a OFDM signaling, in which a base station BS serves fixed users, it is possible to control the reflection of the signal through an appropriate selection of the sub-carrier assigned to each user, without interact with the metasurface and without the need for CSI (channel state information) estimates. However, if, instead of considering the BS as transmitter, we consider a generic user in a unknown location on the territory as transmitter, we could study the reflection provided by the metaprism and the signal received by the BS, to deduce the angle of arrival and also to trace the user's position, conceiving a localization application. This section presents the scenario, with the metaprism as protagonist, on which the studied localization application is based. About that, we consider an OFDM-based wireless system in which a single antenna user (u) transmits a signal to a base station (BS). Let's assume a typical model scenario like the one shown in Fig. 2.10.

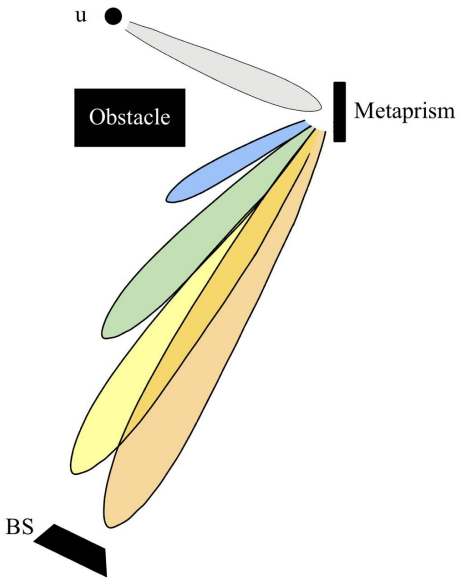


Figure 2.10: Scenario considered: user u, obstacle, base station BS and metaprism

The base station is in NLoS conditions with respect to the user, due to the presence of a generic obstacle for example, and the metaprism is introduced with the aim of extending the coverage area at a low cost. If we assume, for the metaprism, the model proposed in section 2.3 and we consider a reference system, in which the user and BS positions are expressed through Cartesian coordinates, the 3D model of the scenario is the one represented in Fig. 2.11.

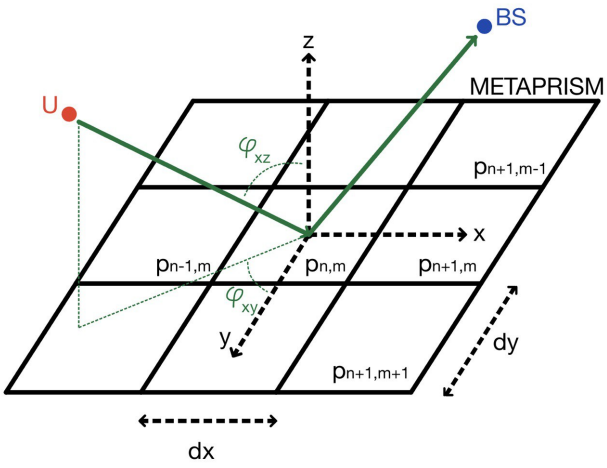


Figure 2.11: 3D model of scenario considered

$\varphi = (\varphi_{xy}, \varphi_{xz})$ is the angle of arrival (i.e. the angle that forms the transmitted signal by the user when affects the metaprism with respect to its normal), while $\vartheta = (\vartheta_{xy}, \vartheta_{xz})$ is the angle that indicates the position of the BS receiver (i.e. the angle from which the signal reflected by the metaprism is observed). Suppose that the position of the user is unknown and that, since it wants to be located, it transmits a pilot tone (i.e. known to the base station) on all the $K$ sub-carriers, because it is not known a priori the sub-carrier which, after the reflection of the metaprism, exactly affects the BS (and is not sure if it exists). This difference with respect to the case studied previously, leads to significant expedients. First of all, as far as the channel gains and the signal received are concerned, we consider the exact models, presented in section 2.5.2, which can be used both in the far-field and near-field cases, but where transmitter and receiver exchange their roles. Channel gains expressions become:

$$\begin{aligned} h_{nm,k}(u) &= \frac{\sqrt{G_t}\,\lambda}{4\,\pi\,|u - p_{nm}|} \exp\left(-j\frac{2\pi\,f_k}{c}\,|u - p_{nm}|\right) \\ g_{nm,k}(BS) &= \frac{\sqrt{G_r}\,\lambda}{4\,\pi\,|BS - p_{nm}|} \exp\left(-j\frac{2\pi\,f_k}{c}\,|BS - p_{nm}|\right) \end{aligned} \tag{2.21}$$

given in function of the transmitter position $(u)$ and the receiver position $(BS)$, while the expression 2.22 described the signal received by the BS.

We just need to figure out how to model the phase profile of the metaprism, of known position, in this case. If the user's position were known, for example in the far-field region of the metaprism, we could design its phase profile according to the equations 2.17, so that, considering that the user transmits an OFDM signal, each sub-carrier is reflected at a certain angle. In fact, by appropriately designing the reflection coefficient of each cell of the metaprism, it is possible to reflect the signal incident on it with a certain angle. In this case of localization, however, we cannot apply these design rules as the $\varphi$ angle, which indicates the user's position, is unknown.

Therefore, in the far-field case, a possible metaprism coefficients design applicable thanks to the reciprocity of the transmitter-metaprism-receiver system, is to use the same design parameters $a_0$ and $b_0$ calculated for the scenario described above, in which the transmitter was represented by the BS of known and fixed position. In fact, from the point of view of the project, the metaprism is reciprocal, i.e. its behavior must not change if it is used in one direction of transmission or the other. However, by designing the metaprism for an opposite transmitter-receiver scenario, the resulting radiation pattern will be exactly the same, but it must be contextualized. In fact, while in the previous scenario the AF represented in practice the radiation pattern of the transmitting antenna, i.e. it indicated the maximum directions towards which each sub-carrier was reflected, now in this scenario, it must be considered as the radiation pattern of the receiving antenna (meaning the metaprism as a sort of receiving antenna). This allows us to deduce that, if the user transmits from a position coinciding with a maximum of the radiation pattern relating to a certain sub-carrier, it means that, in that direction and for that sub-carrier, the metaprism has maximum receptivity, therefore the $k$th sub-carrier transmitted will be reflected at its maximum towards the BS. If the user transmits the $k$th sub-carrier from another position, the reflection introduced by the metaprism will not be maximum and therefore the BS will receive the signal with a lower intensity.

In the example of Fig. 2.12, user must transmit at the frequency related to the sub-carrier represented in green in the radiation pattern, to ensure that the base station is maximally receptive. The decision of the estimator that we will use to estimate the angle of arrival is based on this.

Instead, as regards the estimation of the user's position, which is possible if is in the near-field region of the metaprism, we can use the random design of the $\alpha_{nm}$ coefficients described by Eq. 2.20.

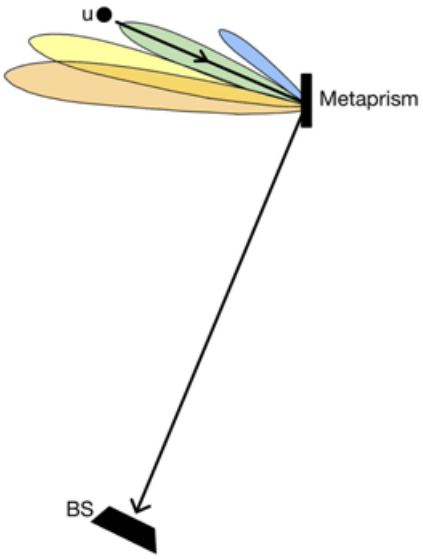


Figure 2.12: Scenario considered under far-field hypothesis: in black the signal transmitted by the user and received by the BS, in color the radiation pattern of the metaprism

In the next section it is shown how the model for the Maximum Likelihood (ML) estimator, which allows the estimation of the user's position, has been derived.

## 2.7 ML estimator design

The signal received by the base station at the $k$th sub-carrier $y_k$, is given by the summation of all the contributions of the different metaprism cells and is expressed by:

$$y_k = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h_{nm,k}(u)\, r_{nm,k}(\varphi, \vartheta)\, g_{nm,k}(BS) \sqrt{P_t}\, w_k\, x_k + n_k \tag{2.22}$$

where: $x_k$ is the transmitted information symbol (pilot tone), $\sqrt{P_t}\, w_k$ is the power allocated to the $k$th sub-carrier. In this regard, we assume that the power allocated to all the $K$ sub-carriers is the same, i.e. $w_k = 1$. $n_k$ is the thermal noise, which we assume Gaussian with zero mean and variance $\sigma^2$.

Developing eq. 2.22:

$$\begin{aligned} y_k &= \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h_{nm,k}(u)\, r_{nm,k}(\varphi, \vartheta)\, g_{nm,k}(BS) \sqrt{P_t}\, x_k + n_k \\ &= \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h_{nm,k}(u)\, \beta_{nm,k}(\varphi, \vartheta) \exp\left(j\Psi_{nm,k}\right)\, g_{nm,k}(BS) \sqrt{P_t}\, x_k + n_k \end{aligned} \tag{2.23}$$

Considering, for simplicity, the reflected signal amplitude of the $n$th cell $\beta_{nm,k}(\varphi, \vartheta) = \beta_0$ for all cells and equal for all sub-carriers, and taking all the independent terms out of the double summation, we get:

$$y_k = \beta_0 \sqrt{P_t}\, x_k \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h_{nm,k}(u)\, \exp\left(j\Psi_{nm,k}\right)\, g_{nm,k}(BS) + n_k \tag{2.24}$$

where we can indicate the term independent of noise, i.e. the useful signal at $k$th sub-carrier, as:

$$s_k(u) = \beta_0 \sqrt{P_t}\, x_k \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} h_{nm,k}(u)\, \exp\left(j\Psi_{nm,k}\right)\, g_{nm,k}(BS) \tag{2.25}$$

The expression 2.24 just obtained represents the signal received from the base station at the $k$th sub-carrier. The spectrum, that the base station receives, will be something like that reported in Fig. 2.13.

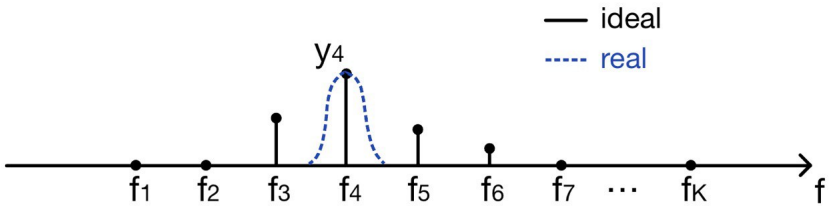


Figure 2.13: Frequency profile of the received signal by the base station

The base station measures the intensity of the $y$ signal received from each $k$th sub-carrier (i.e. at each frequency). For example, with reference to Fig. 2.13, at frequencies $f_1$ and $f_2$, it receives almost nothing (apart from noise), then at frequencies $f_3$, $f_4$, $f_5$, $f_6$ it receives a non-zero signal, which we ideally represent with Dirac deltas, even if in reality each will have a little wider spectrum, if we consider the real radiation pattern of the antennas. Hence, the signal received from the base station at the $k$th sub-carrier will be:

$$y_k = s_k(u) + n_k \tag{2.26}$$

i.e. the useful signal $s_k$ at the $k$th sub-carrier, to which the thermal noise sample $n_k$, modeled as a Gaussian random variable, is added. Consequently, considering all the $K$ sub-carriers, we obtain the vector of the received signals:

$$\underline{y} = \underline{s}(u) + \underline{n} \tag{2.27}$$

The $y_k$ profile is a function of the user's position $u$, so if the user moves, the profile changes. $u$ is therefore our unknown, in fact, finding $u$, means to locate the user. To be

able to estimate the user's position, one hypothesis is to use the Maximum Likelihood estimator, in which the observation is the $\underline{y}$ vector that contains all the $y_k$'s and so the likelihood function is $P(\underline{y}|u)$. Consequently, we have:

$$\widehat{u} = \arg\max_u \; P(\underline{y}|u) = \arg\max_u \; \ln P(\underline{y}|u) \tag{2.28}$$

Given $u$ and therefore fixed $s_k(u)$, the $y_k$ are indipendent Gaussian random variables with mean value $s_k(u)$ and variance $\sigma^2$, so we can write the pdf of the vector $\underline{y}$ which, in the case of complex signals, is:

$$P(\underline{y}|u) = \frac{1}{(2\pi\sigma^2)^K} \exp\left(\frac{-1}{2\sigma^2}\sum_{k=0}^{K-1}|y_k - s_k(u)|^2\right) \tag{2.29}$$

than in logarithmic form and setting the constant that multiplies the exponential as *costant*, we obtain:

$$\ln P(\underline{y}|u) = costant - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}|y_k - s_k(u)|^2 \tag{2.30}$$

The expression of the ML estimator is obtained by developing 2.30:

$$\begin{aligned}
\ln P(\underline{y}|u) &= costant - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}(y_k - s_k(u))\,(y_k^* - s_k^*(u)) \\
&= costant - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}|y_k|^2 + |s_k(u)|^2 - (y_k\, s_k^*(u) + y_k^*\, s_k(u)) \\
&= costant - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}|y_k|^2 + |s_k(u)|^2 - 2\,Re\,(y_k\, s_k^*(u)) \\
&= costant - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}|y_k|^2 - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}|s_k^2(u)| + \frac{1}{\sigma^2}\sum_{k=0}^{K-1}Re(y_k\, s_k^*(u))
\end{aligned} \tag{2.31}$$

where the symbols with $^*$ represent the conjugate complex of the respective signals. We have to find the maximum of the function with respect to $u$, so everything that does not depend on $u$ can be neglected. The first two terms do not depend on $u$, the third seems to depend on it, but represents the signal's energy, which therefore does not depend on the phase. In the end, it is enough to find the maximum of the fourth term, where the pre-multiplicative coefficient can also be neglected. Therefore we get the following expression describing the ML estimator:

$$\widehat{u} = \arg\max_u \; \sum_{k=0}^{K-1} Re(y_k\, s_k^*(u)) \tag{2.32}$$

which provides the $u$ that maximizes the real part of the correlation between $y_k$ e $s_k(u)$ conjugated.

It is important to point out that if we consider the far-field scenario with the beam-steering approach, the estimator will never be able (for the reasons explained above) to estimate the user's position, and therefore of the estimate of $u$, only its relative angle (to the normal to the metaprism) is significant. Conversely, in the case of random metaprism coefficient design, it is expected that, as long as the system works in the near-field, estimated $u$ (carriers both the distance and angle information i.e. the position) is significant. These observations will be validated by the simulations results which will be presented later.

In the following chapter, the code that implements the estimator will be described.

# Chapter 3

# Simulator implementation

This chapter describes the Matlab implementation of the ML estimator user's position previously obtained. The first part presents the scenario and the operating conditions on which it is based the program, while the second part describes the Matlab code that realizes the estimator.

## 3.1 Introduction to simulation

As a simplifying hypothesis, the Matlab program was conceived on the assumption that the angular components relating to the $xy$ plane, both based on the arrival angle $\varphi$, and the observation angle $\vartheta$ (associated to the position of the BS), are considered null, i.e. $\varphi = (0, \varphi_{xz})$ and $\vartheta = (0, \vartheta_{xz})$. In practice, the $y$ coordinate of the user and base station positions, has been chosen equal to zero. So only the $xz$ plane is relevant in describing the user and base station positions. Under this assumption, the scenario generated looks like the one in Fig. 3.1, where:

- the point $u$, which can be expressed through a triad of coordinates ($u_x$, $u_y$, $u_z$) in the negative half plane $xz$ (i.e. to the left of the reference system origin), describes the user's position;
- the point in the reference system origin of coordinates (0,0,0) describes the position of the metaprism;
- the point $BS$, which can be expressed through a triad of coordinates ($BS_x$, $BS_y$, $BS_z$) in the positive half plane $xz$ (i.e. to the right of the reference system origin), describes the position of the base station;
- the obstacle places the user in NLoS conditions with respect to the BS.

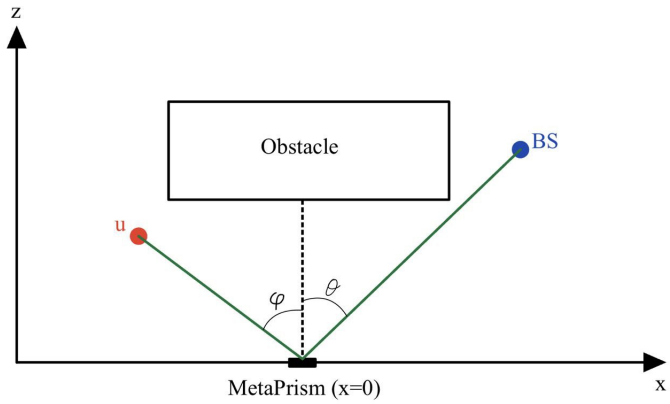


Figure 3.1: Matlab program scenario considered: user $u$ (in red) and base station $BS$ (in blu) are in NLoS conditions because of the obstacle. The introduction of metaprism (in black) allows to improve the wireless link

In order to conduct a more rigorous analysis, let us assume that, while the user's position may vary along the $xz$ half plane, the base station remains fixed in a certain position. Consistently with what previously described, $\varphi$ is the angle between the position of the user and the normal to the metaprism, while $\vartheta$ is the angle evaluated between the position of the base station and the normal to the metaprism. These angles are calculated as follows:

$$\begin{aligned} \varphi &= -\left(\frac{\pi}{2} - \sin^{-1}\left(\frac{u_z}{|p|}\right)\right) \\ \vartheta &= \frac{\pi}{2} - \sin^{-1}\left(\frac{BS_z}{|p_{BS}|}\right) \end{aligned} \tag{3.1}$$

where, in this case, $|p|$ is the user's Euclidean distance from the metaprism and $|p_{BS}|$ is the base station's Euclidean distance from the metaprism. By hypothesis $\varphi$ can only assume negative values, since it is calculated counterclockwise with respect to the normal to the metaprism, while we assume positive values for $\vartheta$, since it is calculated clockwise with respect to the normal to the metaprism. A scenario of this kind is then generated in Matlab as in Fig. 3.2, in which the BS remains fixed (therefore the $\vartheta$ angle will always be the same), and according to the user's position considered, the true angle $\varphi$ is calculated as in 3.1.

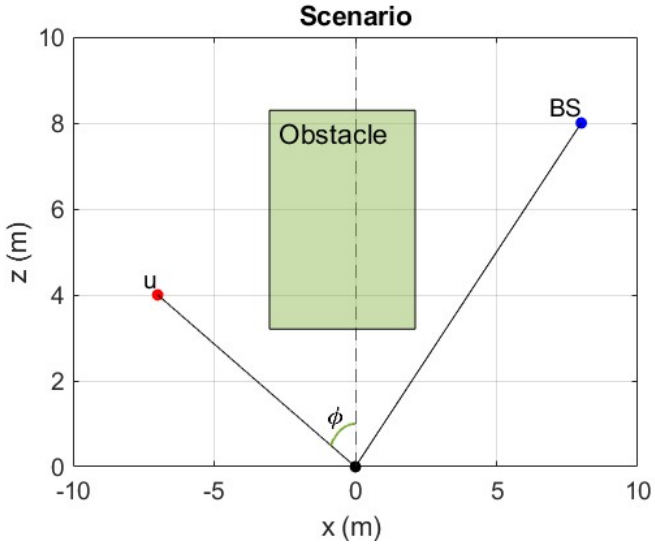


Figure 3.2: Matlab geometric configuration (scenario) generated for one user's position

The Matlab program, based on the estimator obtained, will output the estimate of the user's position, and the closer this estimate is to the true user's position, the better the estimator accuracy will be.

## 3.2 Matlab script description

The Matlab program implements a Monte Carlo simulator, capable of simulating the performance of the obtained ML estimator. This simulator, for each geometric configuration (i.e. for each position of the user) considered, calculates the vector $\underline{s}$ that is the useful signal which contains the $s_k$ (described by 2.25) of all sub-carriers and then performs a certain number of Monte Carlo iterations in which, at each cycle, generates a different realization of the noise vector $\underline{n}$ and therefore a different realization of the signal vector $\underline{y}$. Through that, an estimate of the user's position is obtained by means of the designed estimator. The same procedure will be applied for multiple user's positions, within a certain area of interest, so that in the end the simulator can provide a degree of accuracy in estimating the position for each one.

The Matlab program consists of a main script and a function. Both will be explained in detail through figures showing excerpts of Matlab code that compose the entire program, each commented for clarity. Parameter names are shown in italics. The parameter values used are those typical of wireless communications (reference is made to the values used in [6], typical of millimeter wave links). Some of these were modified during the analysis, depending on the objective of each simulation.

### 3.2.1 Main script

The main script consists of a first part in which the system parameters are initialized, a central part which represents the actual main program and a final part aimed at processing and printing the results on screen.
The script starts by initializing the useful parameters. Variables names are indicated in italics.

```
It = 10;            % number of Montecarlo iterations considered

Metp = [0 0 0];     % x y z coordinates of metaprism position
BS = [8 0 8];       % x y z coordinates of BS position
u_x = -11:-7;       % x coordinates of user position
u_y = 0;            % y coordinates of user position
u_z = 12:15;        % z coordinates of user position

K = 256;            % number of subcarriers considered
Pt_dBm = 10;        % transmission power [dBm]
Gt_dB = 6;          % transmission gain [dB]
Gr_dB = 6;          % receive gain [dB]
c = 299792458;      % light speed [m/s]
f0 = 28*10^9;       % signal central frequency [Hz]
lambda = c/f0;      % signal wavelength
W = 100*10^6;       % bandwidth [Hz]
de_f = W/K;         % subcarriers spacing
x = 1;              % transmitted pilot tone
F_dB = 3;           % receiver noise figure [dB]
```

Figure 3.3: Parameters initialization: coordinates and transmitter/receiver values

In Fig. 3.3 are shown the values of the useful parameters, concerning the number of Monte Carlo iterations considered (*It*), the 3D coordinates of the metaprism (*Metp*), base station (*BS*), user and the values of the parameters relating to the wireless link. The metaprism position coincides with the origin of the reference system, therefore the $x$ coordinates to the left of it will be all negative, while those to the right will be all positive. Variables *u_x, u_y, u_z* are vectors that express user's position ($u_x$, $u_y$, $u_z$), considering that multiple positions are evaluated within a certain area of interest. In this example the area of interest is the one that goes from the -11 to -7 coordinate of the $x$ axis and from the 12 to 15 coordinate of the $z$ axis (far-field case). Each point identified by a triad of integer coordinates represents a user position; therefore a total of 20 positions are evaluated. A center frequency ($f_0$) of 28 GHz and therefore a wavelength (*lambda*) equal to $c/f_0 = 0.0107$ m with $c$ the speed light, a bandwidth ($W$) of 100 MHz and a spacing between sub-carries ($de\_f = 390625$ Hz) given by $W/K$, were chosen for the OFDM signal. For the pilot tone ($x$), a unitary and equal value was assumed for each of the ($K = 256$) sub-carriers considered. The values of transmission power (*Pt_dBm*), antenna gains (*Gt_dB, Gr_dB*) and receiver noise figure (*F_dB*) were set at 10 dBm, 6 dB, 6 dB and 3 dB, respectively. Particularly, we consider the transmitting and receiving antenna gains equal to each other and equal to 6 dB. In this way the two antennas

are not too much directive, therefore we can consider the main lobe of their radiation patterns large enough to 'illuminate' the whole metaprism. Thanks to this expedient, the more useful the closer the antennas are to the metaprism, we can assume the gain of the antennas constant, thus avoiding to consider the relative radiation patterns. In Fig. 3.4 is shown the second part of parametric initialization.

```
B0 = 1;            % reflected signal amplitude
N = 100;           % metaprism cells number along x axis
M = 100;           % metaprism cells number along y axis
dx = lambda/2;     % single cell x dimension
dy = lambda/2;     % single cell y dimension
Lx = N*dx;         % metaprism x dimension
Ly = M*dy;         % metaprism y dimension

step = 0.1;                                    % grid step
u_x_t= -11:step:-7;                            % test grid x coordinates
u_z_t = 12:step:15;                            % test grid z coordinates
pertur_x = lambda*randn(1,length(u_x_t));      % random x test coord. perturbation
pertur_z = lambda*randn(1,length(u_z_t));      % random z test coord. perturbation
u_x_test = u_x_t + pertur_x;                   % test grid x coordinates (perturb.)
u_z_test = u_z_t + pertur_z;                   % test grid z coordinates (perturb.)

% Approch choice: beamsteering design o random design
approch = input(['To select the beamsteering design' ...
    ' press 1, otherwise press 2 to select the random design: ']);
```

Figure 3.4: Parameters initialization: metaprism and step grid values and approch choice

The first block code shown in Fig. 3.4 represents the metaprism related parameters. We assume the amplitude of the signal reflected by each cell $\beta_0$, here named *B0*, equal to 1 by hypothesis. $N$ and $M$ represent the number of metaprism cells along $x$ and along $y$ axis, respectively. We consider a square metaprism, that is $N = M$ and in this example a surface of $100 \times 100$ cells. The dimension of the single cell, respectively $dx$ and $dy$, are typically equal to half the signal wavelength. Multiplying the number of cells along the two dimensions by the dimensions of each one, gives the total dimension of the metaprism, indicated by $Lx \times Ly$.

The second code block is about the grid of ML test positions at which the ML function is evaluated to find its maximum when assessing the performance for a specific user's position. In the example shown, the test grid pitch (*step*) was set to 0.1 meters, equal along both the $x$ and $z$ axes. This corresponds to testing user positions, within the area of interest, every 10 cm. To avoid that the coordinates of the test grid match perfectly with those of the real positions, thus forcing the results to ideality, but to ensure that they are on average around them, these are slightly perturbed. In fact, the simulations are more truthful when the test grid is uncorrelated with respect to the real user's positions. To do this, a random gaussian variable, with standard deviation equal to $\lambda$, is added to each coordinate of the test grid. By generating a test grid with about 1 cm of noise, which is the $\lambda$ value in our case, all the estimates obtained will have a bound of 1 cm of

error. This is the resolution of the simulator that can be decreased by selecting a more dense grid at the expend of a longer simulation time. That is, if the simulator reports an estimate of the position with an error of 1 cm compared to the true one, we know that this error could potentially be even smaller, conversely an error of a few meters for example is certainly effective, being an order of magnitude greater than the resolution in use.

The value of the *approch* parameter, which is requested as input to the program, determines the functioning hypothesis of the transmitter-metaprism-receiver system. In fact, if we choose the value 1 for this parameter, then we are choosing to study the behavior of the system under the beamsteering design hypothesis, while if we choose the value 2 we select the random design hypothesis. Depending on the selected case, the $\alpha_{nm}$ coefficients of the metaprism will be calculated differently. How they are calculated will be shown later.

Starting from these values initialized, the other parameters useful for the estimator implementation are calculated, such as the distances of the system nodes, the angles involved, etc.

```
pBS = norm(BS - Metp);                    % euclidean distance metaprism-BS

theta_rad = pi/2 - asin(BS(3)/abs(pBS));  % observation angle in radiant
theta = rad2deg(theta_rad);               % observation angle in degrees

k_boltz = 1.38*10^-23;                    % Boltzmann constant [J/K]
T0 = 290;                                 % reference temperature [K]
Ta = T0;                                  % antenna noise temperature [K]
T_syst = Ta + T0*(10^(F_dB/10)-1);        % system noise temperature [K]
N0 = k_boltz*T_syst;                      % noise power [W/Hz]
sigma = de_f*N0;                          % noise variance
dev_std = sqrt(sigma);                    % noise standard deviation

Pt = 10^((Pt_dBm-30)/10);                 % transmission power [W]
Gt = 10^(Gt_dB/10);                       % transmission gain [ ]
Gr = 10^(Gr_dB/10);                       % receive gain [ ]

cost = B0*sqrt(Pt)*x;                     % constant of s(k) calculation

[A_test,B_test] = meshgrid(u_x_test,u_z_test);
cc_test = cat(2,A_test',B_test');
u_t = reshape(cc_test,[],2);
zero = zeros(1,length(u_t));
u_test = [u_t(:,1) zero' u_t(:,2)];       % matrix of test x y z coordinates
```

Figure 3.5: Parameters calculation: BS angle and distance from metaprism, noise, power and gain linear conversion, test coordinates matrix

The first line of code calculates the (Euclidean) distance between the base station and the metaprism and is indicated by the parameter *pBS*. *theta_rad* is the variable that indicates the observation angle, i.e. the one that identifies the direction in which the BS is located, and is calculated, in radians, according to Eq. 3.1. Based on the coordinates

that identify the position of the base station, i.e. [8 0 8] in this example, the $\vartheta$ angle is 45°. The third code block represents the calculation of the thermal noise introduced by the communication channel, in particular its standard deviation (indicated by the parameter *dev_std*) which will be used in the generation of the relative Guassian random variable that models it. The noise standard deviation is calculated as the root of its variance $\sigma^2$ (described by the parameter *sigma*), which is given by:

$$\sigma^2 = \Delta f \cdot N_0 \tag{3.2}$$

where $\Delta f$ is the spacing between sub-carriers and $N_0$ is the single-side noise power density, calculated through the following equation:

$$N_0 = k_{Boltzmann} \cdot T_{syst} \tag{3.3}$$

i.e. the product between the Boltzmann constant ($\approx 1.38 \cdot 10^{-23}$) and the system noise temperature (variable *T_syst*), that is given by:

$$T_{syst} = T_a + T_0 \cdot (10^{F_{dB}/10} - 1) \tag{3.4}$$

where: $T_a$ is the antenna noise temperature, that in terrestrial links is typically equal to the reference temperature $T_0 = 290K$ and $10^{F_{dB}/10}$ is the conversion of the receiver noise figure from dB to linear.

The parameters *Pt, Gt, Gr* respectively represent the transmission power in Watt and the channel gains converted from dB to linear. The variable *cost*, that is the result of the product of $\beta_0$, the root of the transmitted power in Watts and the pilot tone, describes the constant that multiplies the double summation in the calculation of $s_k$, given by Eq. 2.25. To simplify the analysis, we assume that $\beta_0 = 1$, i.e. that the amplitude of the signal reflected by each cell, is equal to 1 for all sub-carriers. The last code block, instead, is useful to generate the matrix which, in each row, contains the coordinates $(x\, y\, z)$ of the ML test positions.

After this part of code relating to the useful parameters, the script continues with the main program. Firstly, vectors and matrices are pre-allocated in memory. This increases the execution speed of the code.

```
s_test = zeros(length(u_test),K);   % s_k test matrix
f = zeros(1,K);                     % subcarriers frequency vector
Alpha = zeros(M,N);                 % alpha coefficients metaprism matrix
dist_BS = zeros(M,N);               % BS-nm_th cell distance matrix
```

Figure 3.6: Matrices and vectors pre-allocation in memory

*s_test* is the variable that represents the matrix, with as many columns as the sub-carriers and as many rows as the ML test positions evaluated, which contains in each row

the vector $\underline{s}$ relative to a certain ML test position. $f$ represents the vector of the sub-carriers frequency, while *Alpha* is the matrix that has as many elements as the cells of the metaprism ($M \times N$) and each element represents the $\alpha_{nm}$ coefficient of the relative $nm$th cell. Depending on the chosen approach (1 or 2), this matrix will be filled differently. *dist_BS* is a $M \times N$ matrix in which each element represents the Euclidean distance between the $nm$th cell and the BS position.

The code shown in the Fig. 3.7 makes it possible to choose the analysis approach of the scenario considered, i.e. beamsteering or random design, according to the method of calculating the $\alpha_{nm}$ coefficients of the metaprism.

```
if approch==1               % beamsteering design selection
    theta_m = 40;           % angle that defines the angular range of the reflection (in degrees)
    theta_m_rad = deg2rad(theta_m);                                          % theta_m in randians
    a0 = (-4*pi/(lambda*W))*(-sin(theta_rad+theta_m_rad)+sin(theta_rad));    % metaprism coeff.
    b0 = 0;                                                                  % metaprism coeff.
    for n_te = 1:N                                                  % scrolling of cells along x
        xn_te = n_te*dx - N*(dx/2);                                 % nm-th cell x coordinate
        for m_te = 1:M                                              % scrolling of cells along y
            ym_te = m_te*dy - M*(dy/2);                             % nm-th cell y coordinate
            Alpha(m_te,n_te) = a0*xn_te + b0*ym_te;         % nm-th cell alpha coeff. calculation
            p_nm_te = [xn_te ym_te 0];                      % nm-th cell coordinates
            dist_BS(m_te,n_te) = norm(BS-p_nm_te);     % nm-th cell-BS distance calculation
        end
    end
elseif approch==2           % random design selection
    for n_te = 1:N
        xn_te = n_te*dx - N*(dx/2);
        for m_te = 1:M
            ym_te = m_te*dy - M*(dy/2);
            Alpha(m_te,n_te) = rand()*10^-6;
            p_nm_te = [xn_te ym_te 0];
            dist_BS(m_te,n_te) = norm(BS-p_nm_te);
        end
    end
end
```

Figure 3.7: This code lines implements the selection between the $\alpha_{nm}$ beamsteering or random coefficient calculation approach

The calculation of the coefficients $\alpha_{nm}$ in the beamsteering design case is given by the Eq. 2.15, reported here for clarity: $\alpha_{nm} = a_0\, x_n + b_0\, y_m$, based on the two metaprism design coefficients: $a_0$ and $b_0$. Suppose we want that the signal component relative to the sub-carrier $k = K$ (256 in this example) is reflected by the metaprism with an angle $\vartheta_{0,K}$ = - $\vartheta$ - $\vartheta_m$ for a certain angle $\vartheta_m$ (indicated by the variable *theta_m*). If we set $\vartheta_m$ = 40° and with $\vartheta$ = 45° then $\vartheta_{0,K}$ = - 85°, so we are designing the metaprism so that the main lobe of the equivalent array factor moves from $\vartheta_{0,1} \cong$ - 25° to $\vartheta_{0,256}$ = - 85°, when the sub-carrier index ranges from 1 to $K$. Thanks to this assumption, from Eq. 2.17 by

setting k = $K$ we obtain the design coefficients values $a_0$ and $b_0$, which are provided by:

$$\begin{aligned} a_0 &= -\frac{4\,\pi}{\lambda\, W}\,(sin(-\vartheta - \vartheta_m) + sin(\vartheta)) \\ b_0 &= 0 \end{aligned} \tag{3.5}$$

Consequently, the expression that gives the $k$th sub-carrier angle of reflection is the following:

$$\vartheta_{0,k} = sin^{-1}\left(-sin(\vartheta) + \frac{2\,(f_k - f_0)}{W}\,(sin(-\vartheta - \vartheta_m) + sin(\vartheta))\right) \tag{3.6}$$

The calculation of the coefficients $\alpha_{nm}$ in the random design case, instead, is based on the idea discussed before in section 2.5.4, i.e. to model the coefficient of each $nm$th cell as a random real number. This, as we sad, ensures that the frequency profile received by the base station is unique for each user's position evaluated, becoming a sort of signature for each one. We choose random real numbers in the order of $10^{-6}$, i.e. similar values, in terms of magnitude, to those of the respective coefficients in the beamsteering design case and so that they are not excessively high, since they represent the exponential argument in the calculation of $s_k$.

The method used to fill matrix *Alpha* is the same under both coefficients design assumptions. That is, by means of a double inserted *for* loop, all the cells of the metaprism are scrolled, each identified by a triad of coordinates $p_{nm} = (x_n, y_m, 0)$, exclusively on the $xy$ plane, and depending on the chosen approach, the cell coefficient is calculated and stored in the respective matrix element. The nested $for$ loop is also useful for filling the $dist\_BS$ matrix with the Euclidean distances of each metaprism cell from the BS, necessary for the $s_k$ calculation.

The script continues with two $for$ loops, one after the other, as shown in Fig. 3.8.

```
for k = 1:K
    f(k) = f0 - W/2 + k*de_f;                          % k-th sub-carrier frequency
end

for t = 1:length(u_test)                               % ML test positions scrolling
    users = u_test(t,:);                               % ML test positions selection
    [ssignal] = transmission(users,K,N,M,dx,dy,Gr,Gt,lambda,f,f0,c,Alpha,cost,dist_BS);
    s_test(t,:) = ssignal;                             % s test vector
end
```

Figure 3.8: This code lines implements the sub-carriers frequency calculation and the vector $\underline{s}$ calculation for every ML test positions

The first $for$ cycle is useful for calculating the central frequency of each sub-carrier considered, described by Eq. 2.7. $f$ is then the vector of all sub-carriers frequency. The second $for$ loop, instead, it allows the calculation of the vector $\underline{s}$, containing the $s_k$

values for each $k$th sub-carrier, for all evaluated ML test positions. Scrolling the *u_test* matrix, the triad of coordinates of the ML test positions is selected and one at a time it is saved in a vector (*users*). Through the function *transmission*, is calculated the $\underline{s}$ vector of the ML test position in use. The rows of the *s_test* matrix correspond to the vector $\underline{s}$ calculated for each ML test position. The *transmission* function will be explained in detail in the section 3.2.2.

The main script continues with a part relating to the printing of the results, in particular this code fragment concerns the plot of the geometric configuration.

```
figure('units','normalized','outerposition',[0 0 1 1]); subplot(2,3,1);
annotation('textbox',[.18 .48 .5 .3]
,'String','Obstacle','Fontsize',12,'BackgroundColor',"#77AC30",'FaceAlpha',0.4);
plot(0,0,'ok','MarkerFaceColor','black'), grid on, title('Scenario','FontSize',12);
text(0,0,'Metp','VerticalAlignment','bottom','HorizontalAlignment','left','Fontsize',12);
hold on
mm = max(BS(3),max(u_z));
xlim([min(u_x)-1 BS(1)+1]); ylim([0 mm+1]);
xline(0,'LineStyle','--','LineWidth',1.5), xlabel('x (m)'), ylabel('z (m)');
plot(BS(1),BS(3),'ob','MarkerFaceColor','blue');
text(BS(1),BS(3),'pBS','VerticalAlignment','bottom','HorizontalAlignment','right', . . .
'Fontsize',12);
rx = 1; text(-rx/2,1.5*rx,'\phi','FontSize',12);
```

Figure 3.9: This code fragment implements geometric configuration plotting

The following lines of code, instead, create a matrix ($d$) with, in each line, the triad of coordinates of the user's position. This matrix, therefore, has three columns and as many rows as the possible combinations of user coordinates selected in the area of interest with a step of 1 m.

```
[A,B] = meshgrid(u_x,u_z);
cc = cat(2,A',B');
d = reshape(cc,[],2);
zeroo = zeros(1,length(d));
d = [d(:,1) zeroo' d(:,2)];     % user's position coordinates matrix
```

Figure 3.10: This code fragment implements the generation of all the user's positions coordinates matrix

In Fig. 3.11 is shown the matrix $d$ values for this example code, where each row contains the coordinates of the user's locations evaluated.

| x | y | z |
|---|---|---|
| -11 | 0 | 12 |
| -10 | 0 | 12 |
| -9 | 0 | 12 |
| -8 | 0 | 12 |
| -7 | 0 | 12 |
| -11 | 0 | 13 |
| -10 | 0 | 13 |
| -9 | 0 | 13 |
| -8 | 0 | 13 |
| -7 | 0 | 13 |
| -11 | 0 | 14 |
| -10 | 0 | 14 |
| -9 | 0 | 14 |
| -8 | 0 | 14 |
| -7 | 0 | 14 |
| -11 | 0 | 15 |
| -10 | 0 | 15 |
| -9 | 0 | 15 |
| -8 | 0 | 15 |
| -7 | 0 | 15 |

Figure 3.11: $d$ matrix values example

The following code block is useful to pre-allocate the used matrices and vectors in memory.

```
phi_rad = zeros(1,length(d));              % true phi vectors (radians)
phi = zeros(1,length(d));                  % true phi vectors (degrees)
s = zeros(length(d),K);                    % s matrix
p = zeros(1,length(d));                    % true user's position-metaprism distance vector
r = zeros(It,K);                           % noise matrix
y = zeros(It,K);                           % received signal matrix
argvalue = zeros(1,length(u_test));        % max correlation values vector
SNR_max = zeros(1,length(d));              % SNR max vector
error_p = zeros(length(d),It);             % distance error matrix
error_u = zeros(length(d),It);             % position error matrix
p_valued = zeros(1,It);                    % user's position estimated-metaprism distance vector
phi_valued_rad = zeros(1,It);              % estimated angles vector (radians)
phi_valued = zeros(1,It);                  % estimated angles vector (degrees)
error_phi = zeros(length(d),It);           % angle error matrix
variance = zeros(1,length(d));             % error variance vector
u_valued = zeros(It,3);                    % user's positions valued matrix
rmse_table_phi = zeros(length(d),4);       % angle RMSE and positions matrix
```

Figure 3.12: Matrices and vectors memory pre-allocation (part 1)

```
rmse_table_u = zeros(length(d),4);          % position RMSE and positions matrix
rmse_table_p = zeros(length(d),4);          % distance RMSE and positions matrix
error_phi_mean = zeros(1,length(d));        % mean angle errors vector
variance_phi = zeros(1,length(d));          % angle errors vector
RMSE_phi = zeros(1,length(d));              % angle RMSE vector
error_u_mean = zeros(1,length(d));          % mean position errors vector
variance_u = zeros(1,length(d));            % position error variance vector
RMSE_u = zeros(1,length(d));                % position RMSE vector
u = zeros(length(d),3);                     % true positions matrix
SNR = zeros(length(d),K);                   % SNR matrix
u_mean = zeros(length(d),3);                % mean user positions matrix
p_mean = zeros(1,length(d));                % mean distances vector
phi_mean = zeros(1,length(d));              % mean angles vector
error_p_mean = zeros(1,length(d));          % mean estimate distance vector
variance_p = zeros(1,length(d));            % estimate error variance vector
RMSE_p = zeros(1,length(d));                % distance RMSE vector
```

Figure 3.13: Matrices and vectors memory pre-allocation (part 2)

Subsequently we have the frame of the script, shown in Fig. 3.14, i.e. a *for* loop that allows us to evaluate all the user's positions in the area of interest.

```
for l = 1:length(d)                                 % user's true positions scrolling
    u(l,:) = d(l,:);                                % user position coordinates
    p(l) = norm(u(l,:) - Metp);                     % user-metapism euclidean distance
    phi_rad(l) = -(pi/2 - asin(u(l,3)/p(l)));       % true phi angle (radians)
    phi(l) = rad2deg(phi_rad(l));                   % true phi angle (degrees)

    % User's positions iterative plot
    plot(u(l,1),u(l,3),'or','MarkerFaceColor','red');
    x_coor = [u(l,1),Metp(1),BS(1)];                % x coordinates vector
    z_coor = [u(l,3),Metp(3),BS(3)];                % z coordinates vector
    plot(x_coor,z_coor,'k'); text(u(l,1),u(l,3),'p');
    ph = 0:1/360:-phi_rad(l);                       % phi angle plot
    xx = rx*sin(-ph); yy = rx*cos(-ph);
    plot(xx,yy,'LineWidth',1.5); pause(0.001);

    % s vector calculation
    User = u(l,:);
    [signals] = transmission(User,K,N,M,dx,dy,Gr,Gt,lambda,f,f0,c,Alpha,cost, dist_BS);
    s(l,:) = signals;

    % SNR calculation
    SNR(l,:) = (abs(s(l,:)).^2)/sigma;              % SNR vector
    [SNR_max(l), ~] = max(SNR(l,:));                % max SNR
```

Figure 3.14: These code fragments implement the calculation of $\underline{s}$ and SNR for every user's position evaluated

For each user's position generated ($u$), the distance from the origin ($p$) and the angle (in degrees) that this direction creates with the normal to the metaprism ($phi$) are calculated. A part of the code follows which is useful for iteratively plotting the user's position being evaluated. After that, the vector $\underline{s}$, relative to the position of the user

being considered, is calculated using the *transmission* function. This vector is stored in a row of the matrix $s$. The last two code lines represent the signal-to-noise ratio ($SNR$) calculation for the current user's position. Specifically, the maximum value ($SNR\_max$) is taken between the signal-to-noise ratios relating to each sub-carrier and the sub-carrier corresponding to this maximum value will be the one which should be transmitted by the user in order to have the best link quality. The Eq. 3.7 represents the calculation of the signal-to-noise ratio:

$$SNR = |s|^2/\sigma^2 \tag{3.7}$$

Subsequently, for each true position generated, through the lines of code shown in Fig. 3.15, a Monte Carlo cycle consisting of 10 iterations is implemented. At each iteration, the vector $\underline{y}$ is generated (indicated by the parameter $y$), which represents the signal received from the base station, by adding to each element of vector $\underline{s}$, a Gaussian noise complex sample contained in the noise vector (indicated by the variable $r$) with zero mean value and standard deviation equal to the root of the noise variance (previously shown in Eq. 3.2).

```
% Montecarlo cycle
for j = 1:It                                                    % Iterations number
    r(j,:) = normrnd(0,dev_std,1,K) + 1i*normrnd(0,dev_std,1,K);  % noise vector
    y(j,:) = s(l,:) + r(j,:);                                   % received signal vector
    for td = 1:length(u_test)                                   % ML test positions cycle
        argvalue(td) = sum(real(y(j,:).*conj(s_test(td,:))));   % estimator arg
    end
    [~, max_idx] = max(argvalue);                               % ML rule
    u_valued(j,:) = u_test(max_idx,:);                          % user's position evaluated
    p_valued(j) = norm(u_valued(j,:)-Metp);                     % evaluated user-met. dist.
    phi_valued_rad(j) = -(pi/2 - asin(u_valued(j,3)/p_valued(j))); % rel. angle (rad)
    phi_valued(j) = rad2deg(phi_valued_rad(j));                 % rel. angle (deg)
    error_phi(l,j)= abs(phi(l)-phi_valued(j));                  % angle errors matrix
    error_u(l,j) = norm(u(l,:)-u_valued(j,:));                  % position errors matrix
    error_p(l,j) = abs(p(l)-p_valued(j));                       % distance errors matrix
end
```

Figure 3.15: These code fragments implement the Monte Carlo cycle

Once the vector $\underline{y}$ has been obtained, the summation of Eq. 2.32 is calculated. This one, saved in the vector *argvalue*, implements the real part of the correlation between $y_k$ and $s_k^*(u)$, for each ML test position, through the relative test vector $s_k$ taken from the matrix $s$. According to the ML rule, the estimator provides as estimated position (indicated by the variable *u_valued*), the ML test position which corresponds to the element of greatest value of the *argvalue* vector (i.e. the maximum correlation). In practice, the estimator compares the profiles of the sub-carriers of the useful signal and of the test signals, then it chooses the test one that most closely resembles the useful one. The profile-based localization estimation approach falls into the category of fingerprinting techniques. Each row of the matrix contains 10 estimates of the user's position, one for each Monte Carlo iteration, relative to a true position. For each estimated position, the

following are also calculated: the Euclidean distance from the metaprism (indicated by the variable *p_valued*) and the angle in degrees that this position forms with respect to the metaprism normal (indicated by the variable *phi_valued*). *error_phi*, *error_u*, *error_p* are three estimation error matrices. In particular each element of the *error_phi* matrix represents the angle estimation error, given by the difference between the angle relative to the true user's position and that relative to the estimated one. *error_u* contains the position estimation error, i.e. the Euclidean distance between the true position and the estimated one, while, *error_p* contains the distance estimation error, i.e. the difference between the true position - metaprism distance and the estimated position - metaprism distance.

So, ultimately, the estimator provides, as an estimate, the average position (indicated by the variable *u_mean*) over the 10 Monte Carlo cycles.

```
x_mean = mean(u_valued(:,1));                % estimated positions mean x coordinate
z_mean = mean(u_valued(:,3));                % estimated positions mean z coordinate
u_mean(l,:) = [x_mean 0 z_mean];             % mean estimated position
p_mean(l) = norm(u_mean(l,:)-Metp);          % mean position – metaprism distance
phi_mean(l) = rad2deg(-(pi/2 - asin(u_mean(l,3)/p_mean(l)))); % mean position angle
```

Figure 3.16: Lines code about the calculation of the average position over the 10 Monte Carlo cycles

*p_mean* and *phi_mean* are respectively the Euclidean distance from the metaprism and the angle corresponding to this estimated mean position.

Subsequently, the root mean square errors relating to the angle, position and distance from the metaprism are calculated. That is, for each of the three error vectors, the mean value and the variance are calculated and these values are necessary for the calculation of the RMSE (Root Mean Square Error), defined as:

$$RMSE = \sqrt{(\bar{\varepsilon})^2 + Var(\varepsilon)} \tag{3.8}$$

where: $\bar{\varepsilon}$ is the mean value of the estimation error vector while Var($\varepsilon$) is its variance.

```
        % Angle RMSE
        error_phi_mean(l) = mean(error_phi(l,:)); % angle estim. error average value
        variance_phi(l) = var(error_phi(l,:));    % angle estim. error variance
        RMSE_phi(l) = sqrt((error_phi_mean(l))^2+variance_phi(l));   % Angle RMSE

        % Position RMSE
        error_u_mean(l) = mean(error_u(l,:)); % position estim. error average value
        variance_u(l) = var(error_u(l,:));    % position estim. error variance
        RMSE_u(l) = sqrt((error_u_mean(l))^2+variance_u(l)); % position RMSE

        % Distance RMSE
        error_p_mean(l) = mean(error_p(l,:)); % distance estim. error average value
        variance_p(l) = var(error_p(l,:));    % distance estim. error variance
        RMSE_p(l) = sqrt((error_p_mean(l))^2+variance_p(l)); % distance RMSE

        % User's true coordinates and relative RMSE matrices
        rmse_table_phi(l,:) = [u(l,:) round(RMSE_phi(l),2)];
        rmse_table_u(l,:) = [u(l,:) round(RMSE_u(l),2)];
        rmse_table_p(l,:) = [u(l,:) round(RMSE_p(l),2)];

    end
```

Figure 3.17: Lines code about the calculation of RMSE in terms of angle, position and distance from the metaprism with respect to their true relative values and generation of RMSE matrices

For each type of RMSE, a matrix (*rmse_table*) is created by concatenating the coordinates of user with the RMSE value for that position. The last three code lines in Fig. 3.17 implement these matrices.

With the RMSE values found, the heatmap is created, that is a sort of matrix that provides an idea of the degree of accuracy of the estimate as the true user's position changes. Starting from the top of Fig. 3.18, the three code portions generate the heatmap relating to the RMSE in terms of angle, position and distance respectively.

```
%% Printing of results
subplot(3,3,4);
x_p = rmse_table_phi(:,1);
z_p = rmse_table_phi(:,3);
rmse_phi = rmse_table_phi(:,4);
tbl = table(x_p,z_p,rmse_phi);
h = heatmap(tbl,'x_p','z_p','ColorVariable','rmse_phi','Colormap',flipud(autumn));
h.YDisplayData = flipud(h.YDisplayData); title('Heatmap of RMSE (°) angle-based');
xlabel('x (m)'); ylabel('z (m)');

subplot(3,3,5);
rmse = rmse_table_u(:,4);
tbl_u = table(x_p,z_p,rmse);
hu = heatmap(tbl_u,'x_p','z_p','ColorVariable','rmse','Colormap',flipud(autumn));
hu.YDisplayData = flipud(hu.YDisplayData); title('Heatmap of RMSE (m) position-based');
xlabel('x (m)'); ylabel('z (m)');

subplot(3,3,6);
rmse_p = rmse_table_p(:,4);
tbl_p = table(x_p,z_p,rmse_p);
hp = heatmap(tbl_p,'x_p','z_p','ColorVariable','rmse_p','Colormap',flipud(autumn));
hp.YDisplayData = flipud(hp.YDisplayData); title('Heatmap of RMSE (m) distance-based');
xlabel('x (m)'); ylabel('z (m)');
```

Figure 3.18: Code lines for the generation of RMSE heatmaps

```
% Empirical cdf
subplot(3,3,7)
cdf_vect_phi = reshape(error_phi.',1,[]); % Vector of all angle error vectors
[cdf_phi,x_phi] = ecdf(cdf_vect_phi);     % cdf, valued in x_phi, with data in cdf_vect_phi
ecdf(cdf_vect_phi); grid on; title('Angular Error Empirical CDF','Fontsize',10); % cdf plot
xlabel('error (°)'); ylabel('cdf(error)');

subplot(3,3,8)
cdf_vect_u = reshape(error_u.',1,[]);
[cdf_u,x_us] = ecdf(cdf_vect_u);
ecdf(cdf_vect_u); grid on; title('Position Error Empirical CDF','Fontsize',10);
xlabel('error (m)'); ylabel('cdf(error)');

subplot(3,3,9)
cdf_vect_p = reshape(error_p.',1,[]);
[cdf_p,x_po] = ecdf(cdf_vect_p);
ecdf(cdf_vect_p); grid on; title('Distance Error Empirical CDF','Fontsize',10);
xlabel('error (m)'); ylabel('cdf(error)');
```

Figure 3.19: Code lines for the generation of empirical cdf plot for each of the three error vectors (angle, position and distance)

The code lines in Fig. 3.19 allow to generate the empirical CDF (Cumulative Distribution Function). Given, for example, the error matrix in terms of angle *error_phi*, whose rows are the angle error vectors for every true user's position, is created a vector concatenating all the rows of that matrix one after the other. The "ecdf" command generates the CFD of this error vector. This operation is also performed for the other two

error matrices in order to obtain their relative CDFs. The CDF is useful, for example, because if a certain maximum acceptable error value is set for the application considered, the CDF tells us what percentage of positions is covered (i.e. with an estimate error lower than this maximum value). For example, if we consider 0.9 as a value, it means that 90 % of the points (area positions of interest) are covered with an error lower than the corresponding value in the abscissa.

The script ends as shown in Fig. 3.20, i.e. with a code block that creates the heatmap of the signal-to-noise ratio values in dB received from the base station and that creates a table of estimated values for each true user's position. The table in each row contains the coordinates of the user's true position, angle and distance to the metaprism and their respective estimated values.

```
% SNR heatmap
subplot(3,3,2);
SNR_dB = 10*log10(SNR_max);      % SNR_max in dB
SNR_dB = SNR_dB';
tbl_1 = table(x_p,z_p,SNR_dB);
h_snr = heatmap(tbl_1,'x_p','z_p','ColorVariable','SNR_dB');
h_snr.YDisplayData = flipud(h_snr.YDisplayData); title('Heatmap of SNR (dB)');
xlabel('x (m)'); ylabel('z (m)');

% Estimated values table
varNames ={'u_x'; 'u_z'; 'mean u_x valued'; 'mean u_z valued'; 'true distance from Metp (m)';
'mean valued distance from Metp (m)'; 'true phi(°)'; 'mean phi valued(°)'};
T = table(d(:,1), d(:,3), u_mean(:,1), u_mean(:,3), p', p_mean', phi',
phi_mean','VariableNames',varNames);
t = uitable('Data',T{:,:},'ColumnName',T.Properties.VariableNames,...
'RowName',T.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0, 1, 1]);
subplot(3,3,3); set(gca,'XColor', 'none','YColor','none'); set(gca, 'color', 'none');
title('Values tab','FontSize',10);
pos = get(subplot(3,3,3),'position'); set(t,'units','normalized'); set(t,'position',pos);
```

Figure 3.20: Code lines for the generation of the SNR heatmap and the estimated values table for each user's position

### 3.2.2 "Transmission" function

The *transmission* function is called within the main script and implements the calculation of the vector $\underline{s}$, i.e. the value of the signal transmitted for all the $K$ sub-carriers, for each user's position evaluated. Therefore it is used, in the main script, both for the calculation of the $\underline{s}$ vector for each ML test position, and for the calculation of the $\underline{s}$ vector for each position of the user. The Fig. 3.21 shows the code that implements this function.

```
function [signal] = transmission(user,K,N,M,dx,dy,Gr,Gt,lambda,f,f0,c,Alpha,cost,dist_BS)

    signal = zeros(1,K);                          % vector preallocation
    for k = 1:K                                   % sub-carriers scrolling
        ss = 0;                                   % ss vector re-initialization
        for n = 1:N                               % cells scrolling along x dimension
            xn = n*dx - N*(dx/2);                 % nm-the cell x coordinate
            for m = 1:M                           % cells scrolling along y dimension
                ym = m*dy - M*(dy/2);             % nm-th cell y coordinate
                p_nm = [xn ym 0];                 % nm-th cell coordinates
                dist_u = norm(user-p_nm);         % user-cell nm-th distance
                g_nm = ((sqrt(Gr)*lambda)/(4*pi*dist_BS(m,n)))*exp(-1i*(2*pi*f(k)/c)*dist_BS(m,n));% gain
                h_nm = ((sqrt(Gt)*lambda)/(4*pi*dist_u))*exp(-1i*(2*pi*f(k)/c)*dist_u);   % gain
                psi_nm = Alpha(m,n)*(f(k)-f0);    % nm-th cell phase profile
                arg = h_nm*g_nm*exp(1i*psi_nm);   % s(k) double summation argument
                ss = ss + arg;                    % double summation result
            end
        end
        signal(k) = cost*ss;                      % s(k) calculation
     end

end
```

Figure 3.21: Transmission function code lines

With an external $for$ loop all the sub-carriers are scrolled and for each one the value of the transmitted signal is calculated (indicated by the Matlab variable *signal*). This calculation is the one described by Eq. 2.25. To obtain the value of $s_k$, and therefore to implement the double summation along the two dimensions of the metaprism, it is necessary to scroll cell by cell, with a double $for$ loop, and to calculate the argument of this summation. The latter (indicated by $arg$ variable) is given by the product between: the transmission gain from the user to the $nm$th cell (indicated by $h\_nm$ variable), the receive gain from the $nm$th cell to the base station (indicated by $g\_nm$ variable) and the factor representing the phase profile of the metaprism (indicated by $psi\_nm$ variable). The calculation of the gains is the one described by Eq. 2.21, while the one concerning the reflection phase shift introduced by the metaprism is expressed by Eq. 2.6. To find the value of the channel gain between the user and the $nm$th cell it is necessary to obtain the distance between them ($dist\_u$ variable represents that value). In the same way to calculate $g\_nm$ it is necessary to recall for each cell, from the $dist\_BS$ matrix, the value of the distance between it and the BS. The elements of the $Alpha$ matrix must also be recalled, one by one, for the calculation of the metaprism phase profile. In the variable $ss$ the sum of the arguments of the current cell and of the previous one is saved each time, until all the cells are considered. At the end of the inner cycle, therefore, $ss$ contains the value of the double summation and by multiplying this value by the constant $cost$, we obtain the value of the signal relating to the $k$th sub-carrier in use. Once the outer $for$ loop has also completed, the vector $\underline{s}$ is obtained (saved in the *signal* variable), and this represents the output of the $transmission$ function.

# Chapter 4

# Numerical results

This chapter shows the main results obtained from the simulations of the previously described Matlab program. These simulations were carried out in order to evaluate the performance of the ML estimator for the localization of a user who transmits from an unknown position within a certain area of interest in a metaprism based communication scenario. The first part of the chapter is introductory to the simulations, while the following parts refer to the performance of the estimator, as the value of the main parameters varies, in the two cases of metaprism coefficients design: beamsteering and random. Through these tests, the aim is to understand the parameters on which to act and how, in order to be able to improve the estimation performance. Furthermore, comparisons between the performance of the estimator in the various simulated situations are reported.

## 4.1 Introduction to simulation

In each simulation, multiple user's positions, within a certain area of interest, are evaluated. These positions will also be called true user's positions, to differentiate them from the ML test positions, which are the ones tested by the estimator and which do not exactly coincide with the true ones (as described in the previous chapter). For each true position, the estimator provides an estimate of the user's position and reports the relative estimation errors for the angle of arrival with respect to the metaprism normal, distance from metaprism and position itself in terms of RMSE and empirical CDF. The SNR values and a table with the true and estimated values for each user's position, are also reported. RMSE and SNR, for every user's position, are given in the form of a heatmap, where darker colors correspond to higher values. In this way it is possible to see at a glance the most favorable or least favorable user's positions for localization estimation.

### 4.1.1 Simulations scenarios

The results of the simulations in the case of beamsteering design and random design of the metaprism will be reported. Furthermore, for each of the two cases, user's true positions were evaluated in two different areas of interest: one in the near-field region of the metaprism and one at the beginning of the far-field region. The Figs. 4.1 and 4.2 show the two NLoS scenarios considered. In both figures, the BS remains in a fixed position for a more rigorous analysis. For simplicity, we will call the scenario of Fig. 4.1 far-field scenario and that of Fig. 4.2 near-field scenario, even if, as we will describe below, the Fraunhofer distance varies with the variation of the metaprism dimensions considered.

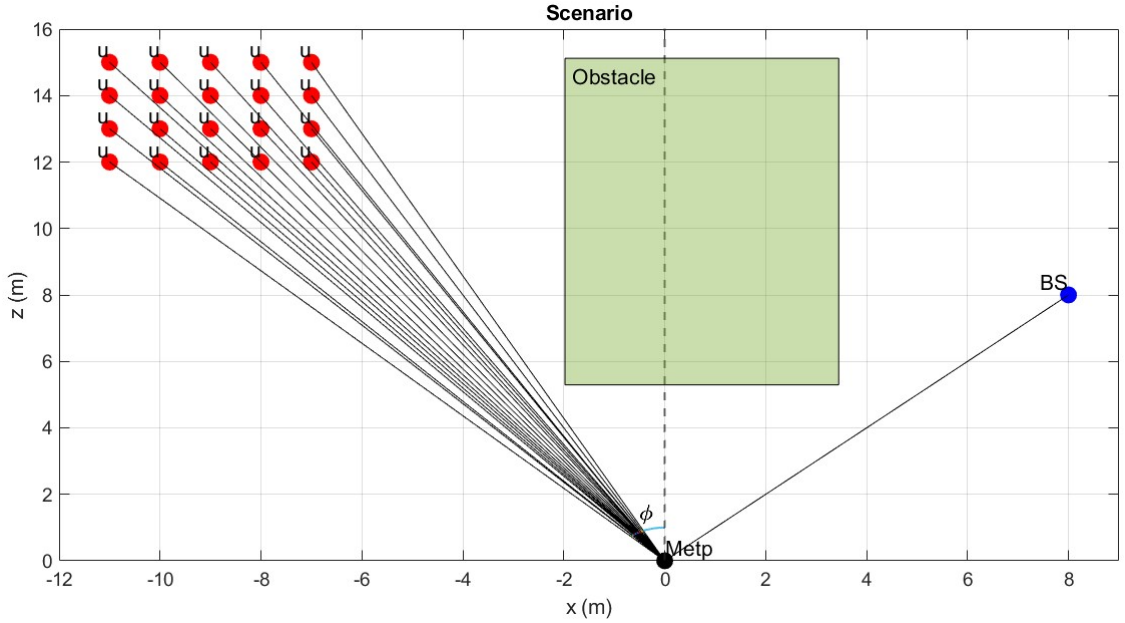


Figure 4.1: Far-field scenario. user x coordinates: -11,-10,-9,-8,-7 meters; user y coordinate: 0; user z coordinates: 12,13,14,15 meters

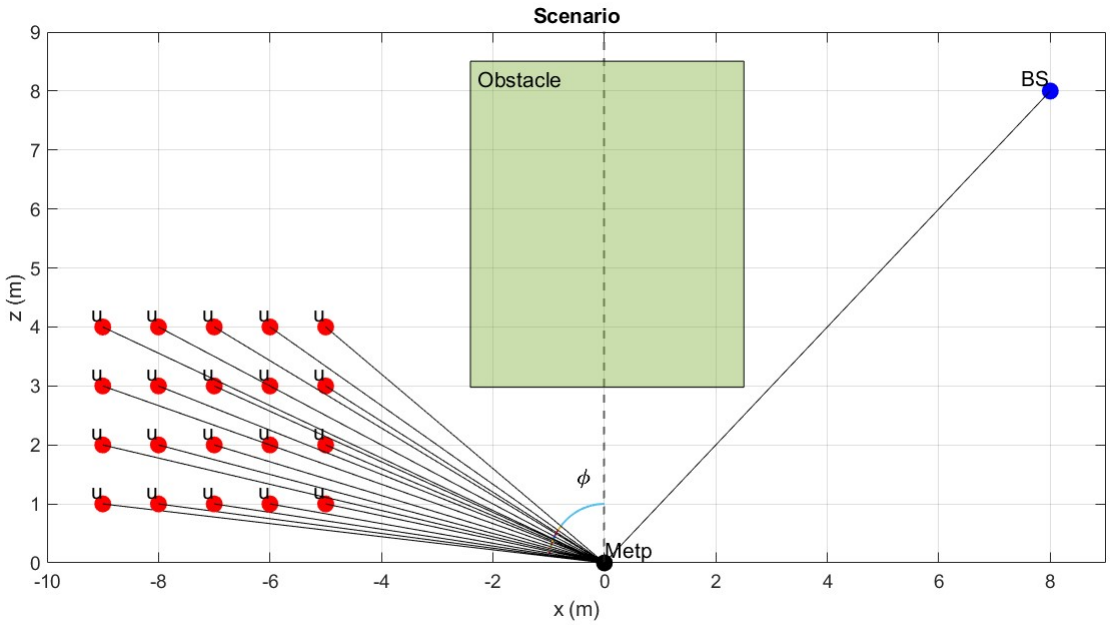


Figure 4.2: Near-field scenario. user x coordinates: -9,-8,-7,-6,-5 meters; user y coordinate: 0; user z coordinates: 1,2,3,4 meters

The near-field is that part of the radiated field that is below distances shorter than the Fraunhofer distance, which depends on the metaprism size and is given by Eq. 2.11. If we consider a metaprism size of $0.268m \times 0.268m$, i.e. with 50 cells per dimension, each equal to $\lambda/2$, equal to 0.0107 m for a operating frequency of 28 GHz, the Fraunhofer distance is equal to:

$$d_{Fraunhofer} = 2D^2/\lambda = 2 \times 0.268^2/0.0107 \cong 13.5\,m \tag{4.1}$$

where $D$ indicates the maximum size of the metaprism and $\lambda$ the signal wavelength. So with a metaprism of this size, user's positions as those in Fig. 4.1 will be in the far-field region, since the closest position is about 14 m from the metaprism. We will consider, instead, user's positions as those in Fig. 4.2 as if they were in the near-field, since the most distant position from the metaprism is about 9 m, although true near-field begins much earlier than Fraunhofer distance. Furthermore, since the performance of the estimator will be evaluated as the dimensions of the metaprism vary and since, the greater the size of the metaprism, the greater the Fraunhofer distance will be, therefore, for bigger metaprisms, user's positions considered in the far-field may no longer really be so. In any case, the dimensions of the metaprism and the distances involved are such as to consider the user's positions always more or less between the near-field and the beginning of the far-field. The performance of the estimator will be evaluated also in these intermediate areas.

### 4.1.2 Estimator choice example

We have already said that the rule used by the ML estimator to stimate the user's position is to compare the signal profile received by the base station with the signal profile relating to each ML test position and choose the most similar one. For example, Fig. 4.3 shows the useful received signal $\underline{s}$ in the absence of noise, relating for example to the user's true position (-10,0,12). The sub-carriers profiles of the signals relating to three ML test positions, are shown below the true one. From the figure, concerning a single Monte Carlo iteration in a far-field scenario and beamsteering design case, it is possible to notice how the blue profile, which is the one relating to the estimated position, is much more similar to the true red profile, compared to the two black test profiles.

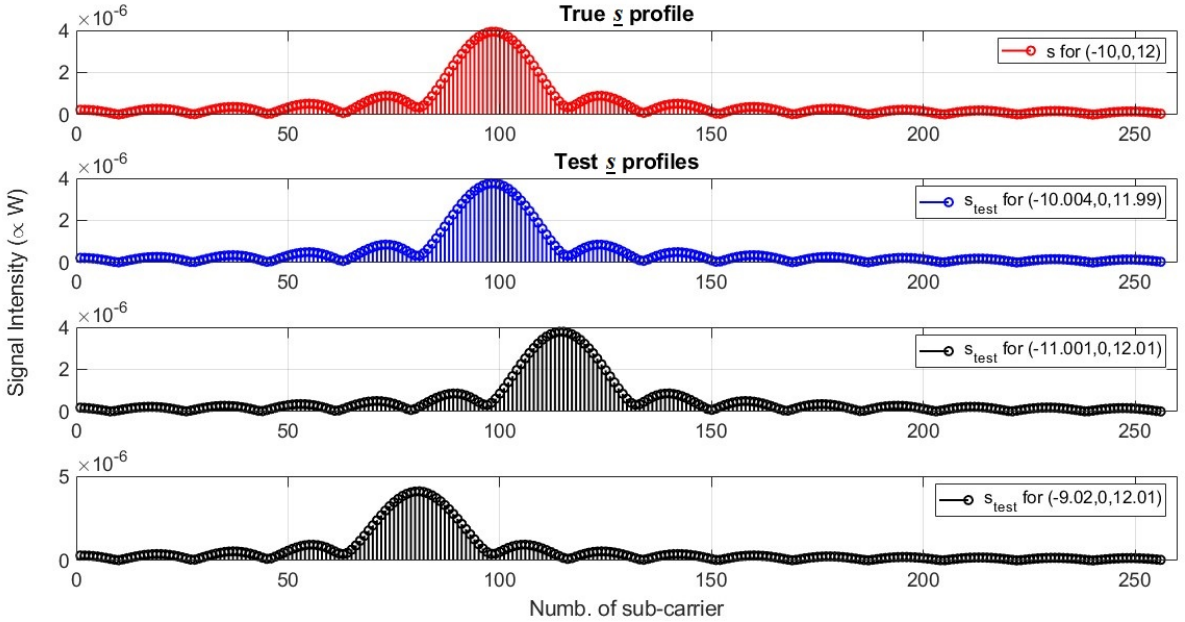


Figure 4.3: Between the three test profiles, the blue one is the closest to the real profile (red)

Based on this comparison, the estimator reports (10.004, 0, 11.99) as estimate of the user's position, i.e. the one relating to profile (in blue) most similar to the true position's profile. Such a profile is also informative of the sub-carrier to which the user should broadcast, based on his location. In fact, it has an intensity peak in correspondence with a certain sub-carrier (about the 100th in the example), and then gradually fades for the increasingly distant sub-carriers. The sub-carrier relative to the peak of this profile is the one on which the user, once localized, can transmit so that the received signal is maximum.

### 4.1.3 Simulation output example

Fig. 4.4 shows an example of output simulation using the beamsteering metaprism design criteria.

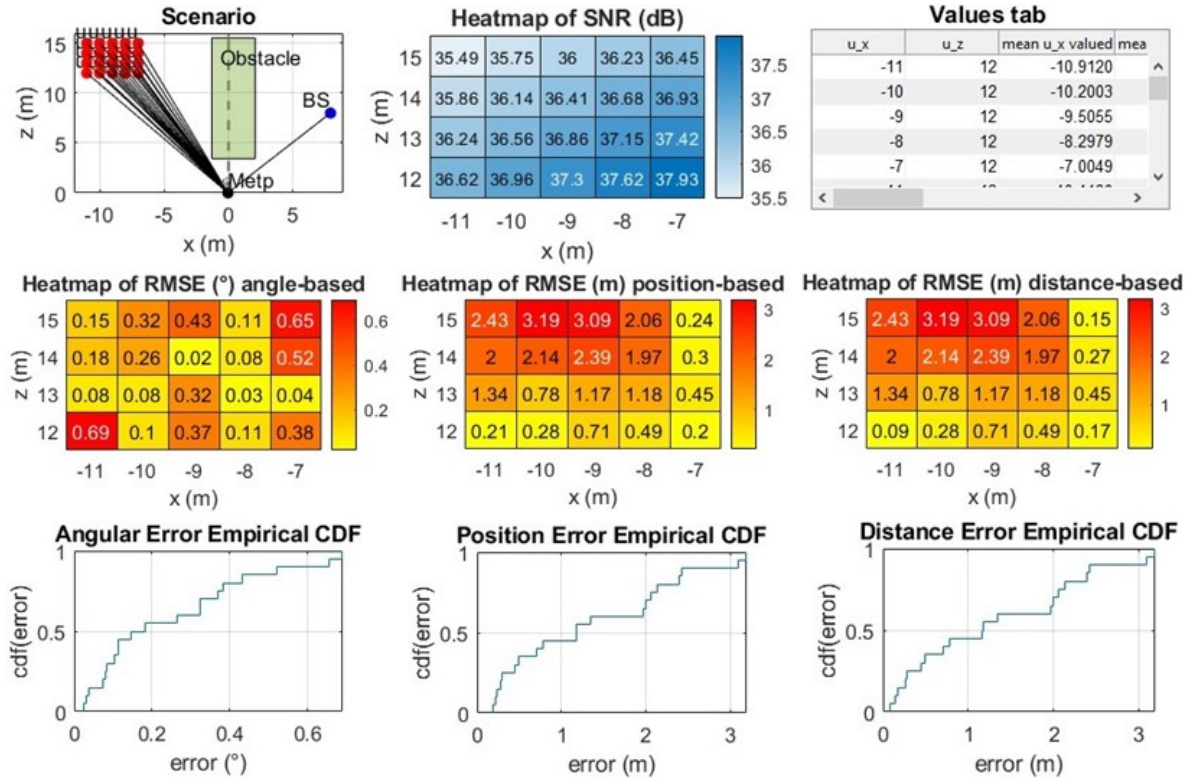


Figure 4.4: Example of output simulation in a far-field scenario and under beamsteering design hypothesis for a metaprism of $50 \times 50$ cells

At the top left of the figure we see the plot of the scenario, i.e. the geometric configuration analysed. The red dots represent the user's true positions, the metaprism is represented by the black dot and the base station by the blue dot, while the obstacle is in green color. Immediately to the right we find the heatmap relating to the SNR values received by the base station, depending on the user's coordinates from which the signal was transmitted. Refer to the values presented in Figs. 3.3 and 3.4. Each colored box of the heatmap represents a user's position, indicated by its coordinates (Fig. 4.5).

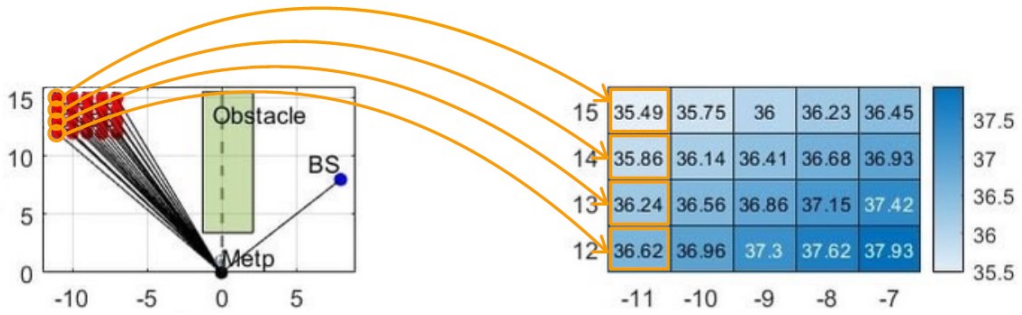


Figure 4.5: Each box of the heatmap corresponds to a user's position

The darker the blue color tends to be, the higher the SNR (given in dB). At the top right is the table of true and estimated values for each user's position. The complete table, relating to this example, is shown in Fig. 4.6. Instead, with reference to the central row of the figure, we find the three heatmaps relating to the RMSE, respectively based on the estimation error of the angle, the position and the distance from metaprism. Again, the user's positions that provide the highest estimation errors are those that tend to red color; on the contrary, positions characterized by a more yellowish color are those for which the estimator provides a smaller estimation error, i.e. that the estimator is better able to estimate. At the bottom of the figure we see the CDFs of the estimation errors, respectively on the angle of arrival, on the position and on the distance from metaprism.

| u_x | u_z | mean u_x valued | mean u_z valued | true distance from Metp (m) | mean valued distance from Metp (m) | true phi(°) | mean phi valued(°) |
|---|---|---|---|---|---|---|---|
| -11 | 12 | -10.6844 | 11.9959 | 16.2788 | 16.0642 | -42.5104 | -41.6904 |
| -10 | 12 | -10.2141 | 12.1007 | 15.6205 | 15.8352 | -39.8056 | -40.1674 |
| -9 | 12 | -9.0083 | 12.1007 | 15 | 15.0857 | -36.8699 | -36.6657 |
| -8 | 12 | -8.0970 | 12.1932 | 14.4222 | 14.6368 | -33.6901 | -33.5865 |
| -7 | 12 | -7.1254 | 12.1007 | 13.8924 | 14.0427 | -30.2564 | -30.4913 |
| -11 | 13 | -10.5846 | 12.5094 | 17.0294 | 16.3865 | -40.2364 | -40.2356 |
| -10 | 13 | -9.2364 | 12.0354 | 16.4012 | 15.1711 | -37.5686 | -37.5040 |
| -9 | 13 | -9.2787 | 13.4589 | 15.8114 | 16.3474 | -34.6952 | -34.5827 |
| -8 | 13 | -7.5117 | 12.1932 | 15.2643 | 14.3213 | -31.6075 | -31.6356 |
| -7 | 13 | -7.2020 | 13.1955 | 14.7648 | 15.0330 | -28.3008 | -28.6255 |
| -11 | 14 | -10.0075 | 12.7091 | 17.8045 | 16.1763 | -38.1572 | -38.2178 |
| -10 | 14 | -8.7124 | 11.9959 | 17.2047 | 14.8259 | -35.5377 | -35.9901 |
| -9 | 14 | -7.9932 | 12.5957 | 16.6433 | 14.9179 | -32.7352 | -32.3989 |
| -8 | 14 | -7.1254 | 12.7091 | 16.1245 | 14.5703 | -29.7449 | -29.2772 |
| -7 | 14 | -6.9954 | 14.0986 | 15.6525 | 15.7386 | -26.5651 | -26.3894 |
| -11 | 15 | -9.1961 | 12.5094 | 18.6011 | 15.5259 | -36.2538 | -36.3211 |
| -10 | 15 | -8.2047 | 12.1007 | 18.0278 | 14.6200 | -33.6901 | -34.1385 |
| -9 | 15 | -7.4078 | 12.2921 | 17.4929 | 14.3517 | -30.9638 | -31.0753 |
| -8 | 15 | -6.9954 | 12.8861 | 17 | 14.6624 | -28.0725 | -28.4958 |
| -7 | 15 | -7.1124 | 14.8835 | 16.5529 | 16.4956 | -25.0169 | -25.5417 |

Figure 4.6: Example of output table values in a far-field scenario and under beamsteering design hypothesis for a metaprism of $50 \times 50$ cells

In the table of Fig. 4.6, starting from the left, the first two columns show the values

of $x$ and $z$ coordinates, since the $y$ coordinate is always zero because we consider a 2D scenario, of the true user's positions evaluated, the two columns on the right instead are x and z coordinates of the estimated positions. Continuing on the right, the two other columns show the distance from the metaprism relative to the true positions and that relative to the estimated ones. The last two columns concern the angle of arrival with respect to the normal of the metaprism of the true and estimated positions.

## 4.2 Estimator performance in the metaprism beamsteering design case

The simulations results carried out in the case of metaprism beamsteering design are reported below, as the main parameters that influence the performance of the estimator vary. Once the noise variance is fixed ($\simeq 3.12 \times 10^{-15}$ Watt, linked to the 3 dB noise figure), the only parameters that can be modified to improve the performance of the estimator is the metaprism size, the transmitted power or the number of signal sub-carriers. In fact, in the case of beamsteering design, the performance is worse for a user who transmits from a position corresponding to a null of the metaprism equivalent array factor (AF). For the same position of the user, trying to increase the number of sub-carriers corresponds to making the AF peaks denser and therefore reducing the probability that the user falls into an AF null.

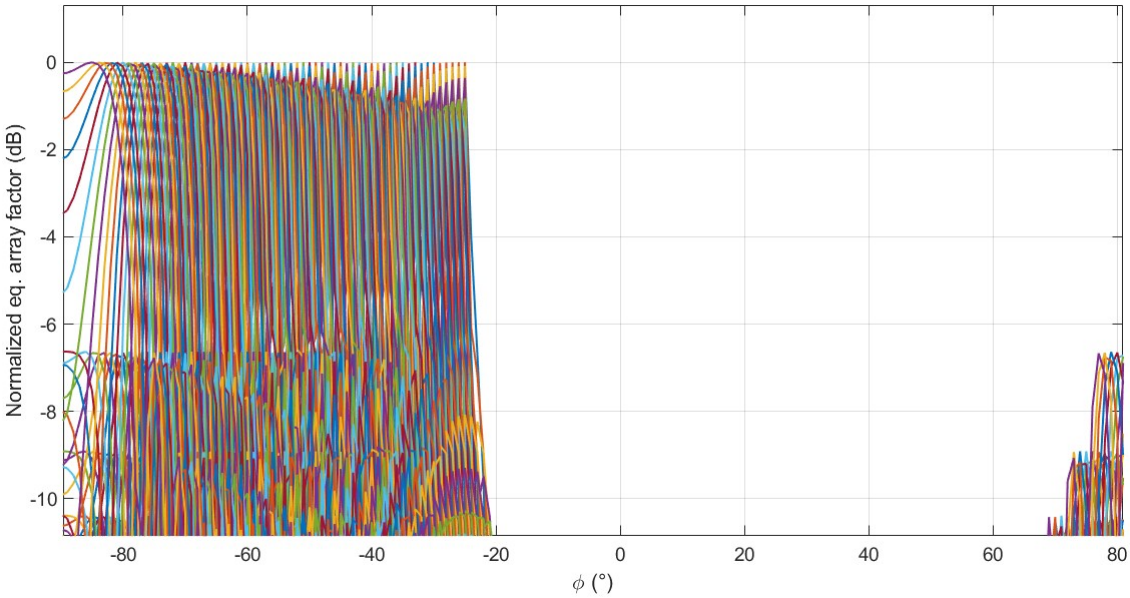


Figure 4.7: Normalized equivalent AF (dB) plot for all 256 sub-carriers

In this regard, the Fig. 4.7 shows the equivalent AF in dB, obtained from Eq. 2.19 and normalized to the product $M \times N$, in the case of base station located at an angle of $\vartheta = 45°$ with respect to the metaprism considered with $100 \times 100$ cells and with $\vartheta_m = 40°$. It is possible to notice that the AF peaks range is from -25° to -85°, as described in the previous chapter. For greater clarity, the equivalent AF for only some sub-carriers is reported in Fig. 4.8.

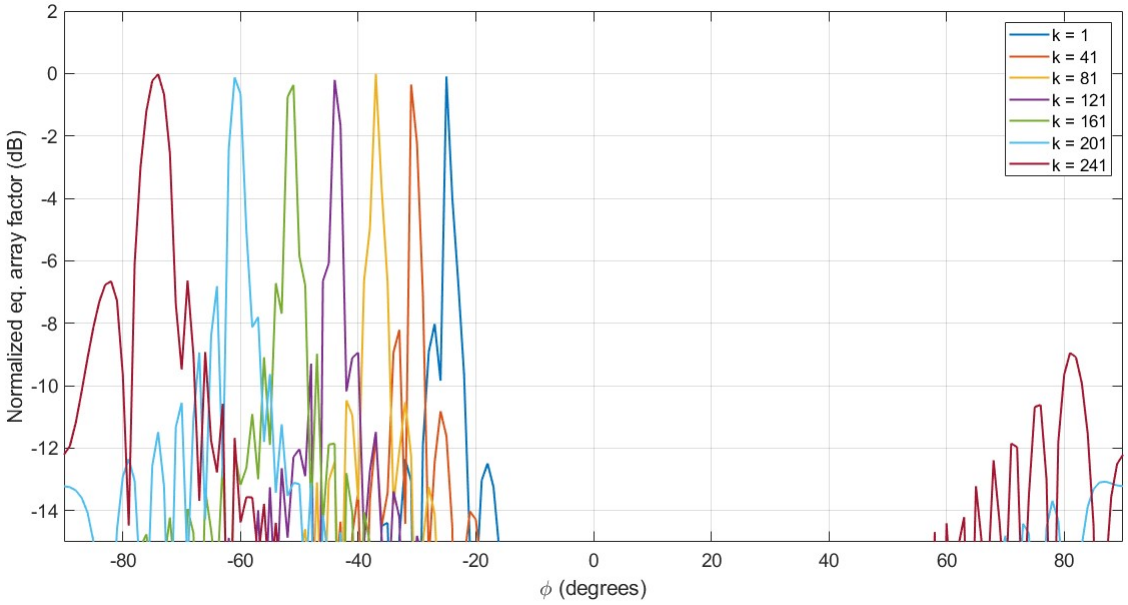


Figure 4.8: Normalized equivalent AF (dB) plot for only some sub-carriers: from 1 to 241 with step of 40

Furthermore, if the signal is transmitted at a higher power, the SNR received by the BS will be higher and consequently the estimation accuracy. On the other hand, however, the transmission power cannot be increased excessively: to be sure to comply with the regulations on radio emissions in the band used, we keep the transmission power below 15 dBm. If these power values still do not guarantee acceptable performance of the estimator, the only alternative would be to move the communicating nodes closer to the metaprism.

The basic values of the parameters [1] used in the simulations are those indicated by the following list, any changes from these are explicitly indicated.

- BS coordinates: [8 0 8]; Distance from the metaprism equal to $\cong$ 11.3 m and relative angle equal to 45°
- Transmission power: 10 dBm

[1] The values of the main parameters partially reflect those reported in [6]

- Transmitter gain: 6 dB
- Receiver gain: 6 dB
- Signal central frequency: $28 \cdot 10^9$ Hz;
- Signal wavelength: 0.0107 m
- Signal bandwidth: $100 \cdot 10^6$ Hz
- Sub-carriers number: 256
- Metaprism cell dimension: 0.0054 m
- Test grid step: 0.1 m

All the following simulations results of this setion refer to the far-field scenario represented in Fig. 4.1. As previously said, in the far-field case only the angle estimate is significant, consequently only results associated with the angle will be reported. In fact, from the simulation output example in Fig. 4.4 it is possible to note that the estimation errors relating to the distance from the metaprism and the user's position can even exceed 2 meters, much higher than the estimation error relating to the angle. This is to demonstrate that in the far-field region, all positions that are in the same direction give the same value of $s_k$ (the same sub-carriers profile), so the ML will take one at random. It happens that the estimations of the distance from metaprism and of the user's position are random while that of the angle of arrival is precise. So if we consider the far-field case, we can only evaluate the angle estimation accuracy. The final part of this section will be dedicated, instead, to the performance of the estimator evaluated for user's positions in the near-field region of the metaprism.

### 4.2.1 Simulations results as the metaprism size varies

The estimator performance are reported as the size of the metaprism varies. In particular, the size of each individual cell remains fixed but progressively increases the number of metaprism cells per dimension. For simplicity of analysis, we always assume a square-shaped metaprism, i.e. with the same number of cells on its two dimensions. For each metaprism size, the SNR and RMSE heatmaps and the empirical CDF, related to the angle of arrival estimation, are shown below, for an increasing number of metaprism cells of 10, 30, 50 and 100 per dimension.

10 cells per dimension (metaprism size: $0.054m \times 0.054m$)

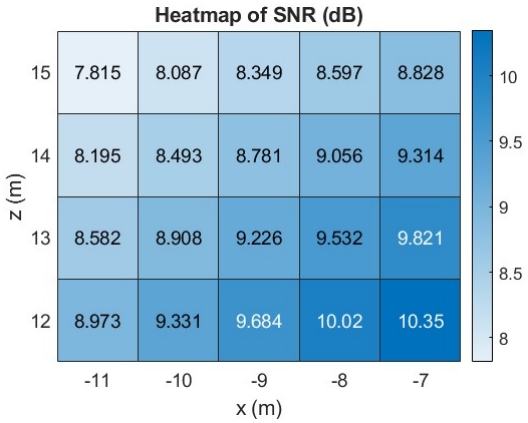


Figure 4.9: SNR Heatmap for 10 cells per dimension metaprism

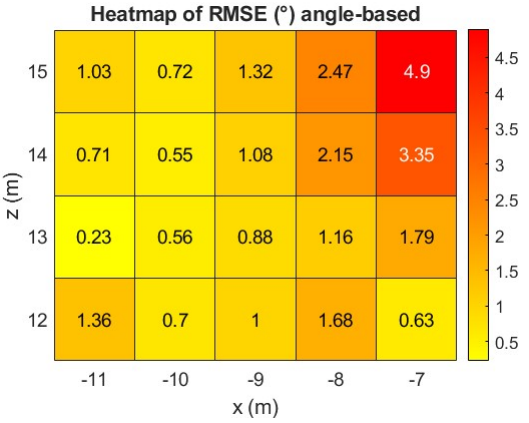


Figure 4.10: RMSE Heatmap for 10 cells per dimension metaprism

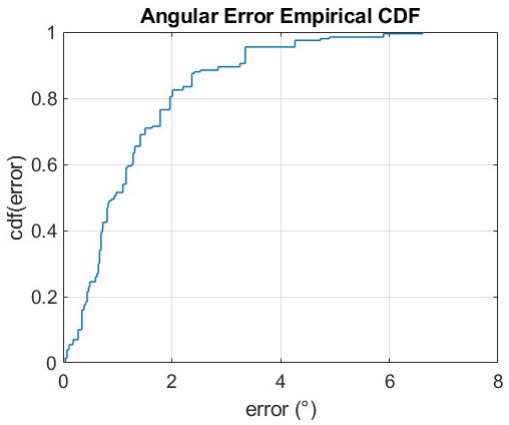


Figure 4.11: Empirical CDF for 10 cells per dimension metaprism

30 cells per dimension (metaprism size: $0.16m \times 0.16m$)

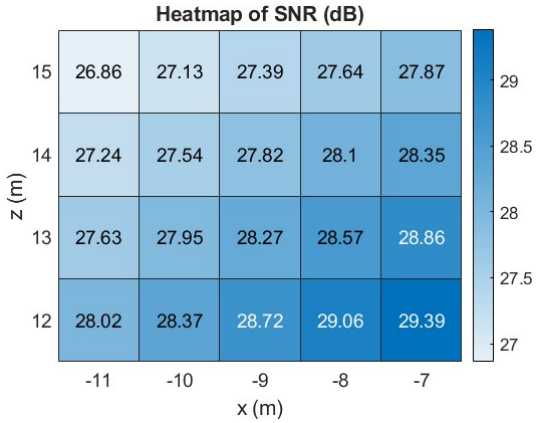


Figure 4.12: SNR Heatmap for 30 cells per dimension metaprism

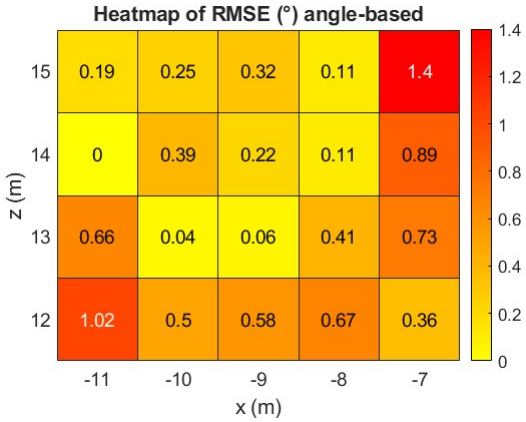


Figure 4.13: RMSE Heatmap for 30 cells per dimension metaprism

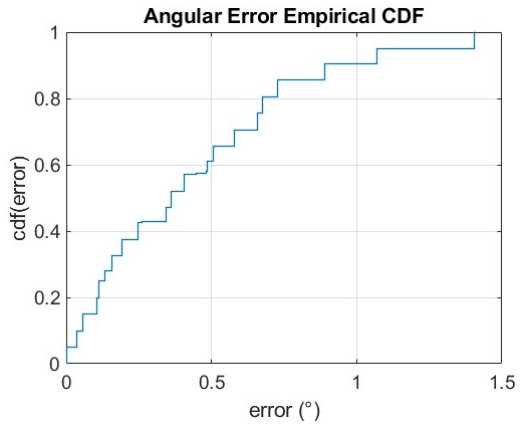


Figure 4.14: Empirical CDF for 30 cells per dimension metaprism

50 cells per dimension (metaprism size: $0.27m \times 0.27m$)

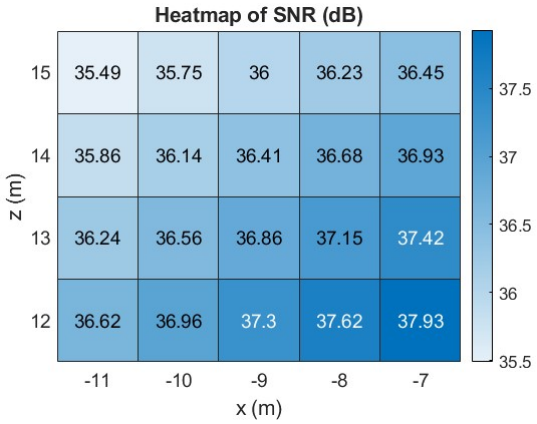


Figure 4.15: SNR Heatmap for 50 cells per dimension metaprism

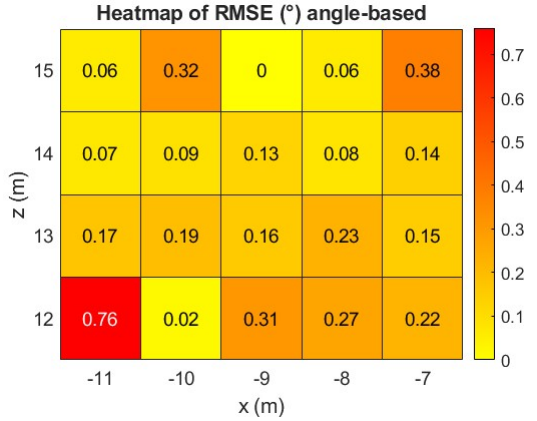


Figure 4.16: RMSE Heatmap for 50 cells per dimension metaprism

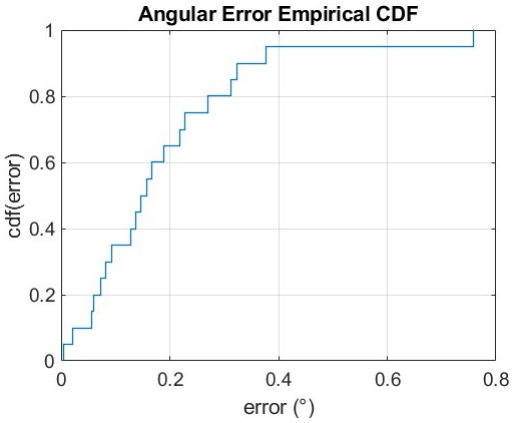


Figure 4.17: Empirical CDF for 50 cells per dimension metaprism

100 cells per dimension (metaprism size: $0.54m \times 0.54m$)

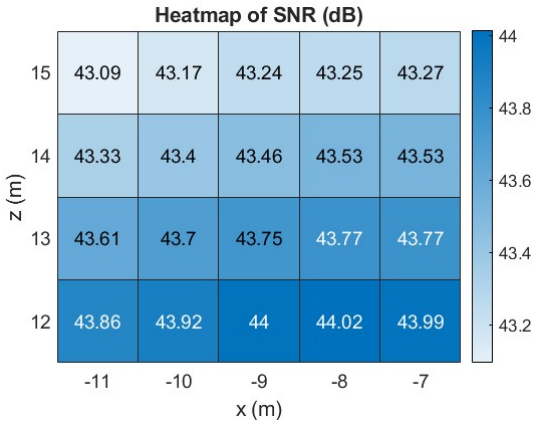


Figure 4.18: SNR Heatmap for 100 cells per dimension metaprism

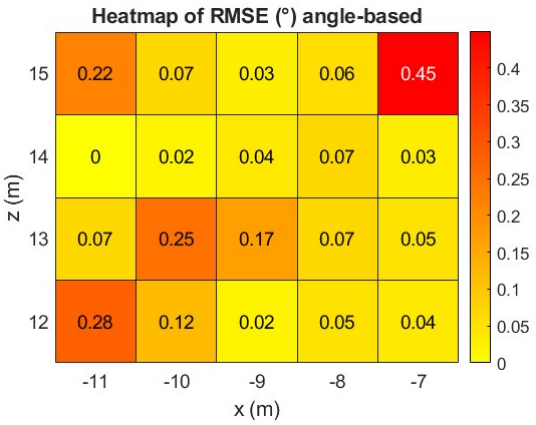


Figure 4.19: RMSE Heatmap for 100 cells per dimension metaprism

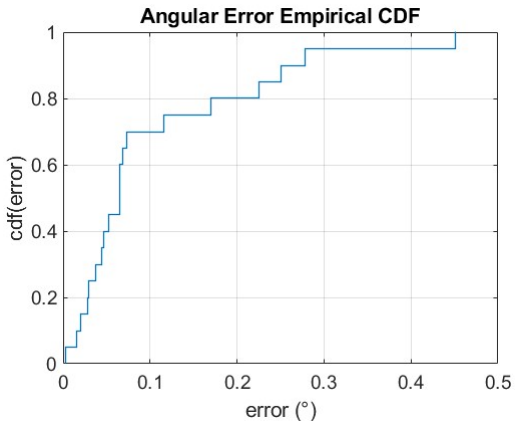


Figure 4.20: Empirical CDF for 100 cells per dimension metaprism

From the previous figures, which report the performance of the estimator, in terms of angle of arrival estimation, as the size of the metaprism vary, it can be seen that:

1. For a certain metaprism size, the SNR values decrease as the distance of the user's position from the metaprism increases. In fact, the signal power, due to the path-loss, tends to weaken as the distance increases. Furthermore, given a certain position of the user, the SNR values increase as the number of cells increases. In fact, a larger metaprism will be able to capture a greater portion of the transmitted signal power.

2. For a fixed metaprism cells number, user's positions which correspond to a higher error, in estimating the angle of arrival, appear to be those in the upper right and lower left of the area of interest. Let us take for example these two user's positions

and try to explain why they are critical in terms of localization. The position at the top right, of coordinates (-7,0,15), is relative to an angle of arrival of about -25°, i.e. at the lower extreme of the range of the main lobe of the metaprism equivalent array factor. For positions close to this or to the upper bound (-85°), the estimator struggles to estimate correctly. Instead, the user's position at the bottom left, of coordinates (-11,0,12), corresponds to an angle of arrival of approximately -42.5°. From Fig. 4.21 it can be seen how this angle corresponds to a null between the sub-carriers in the metaprism equivalent AF.

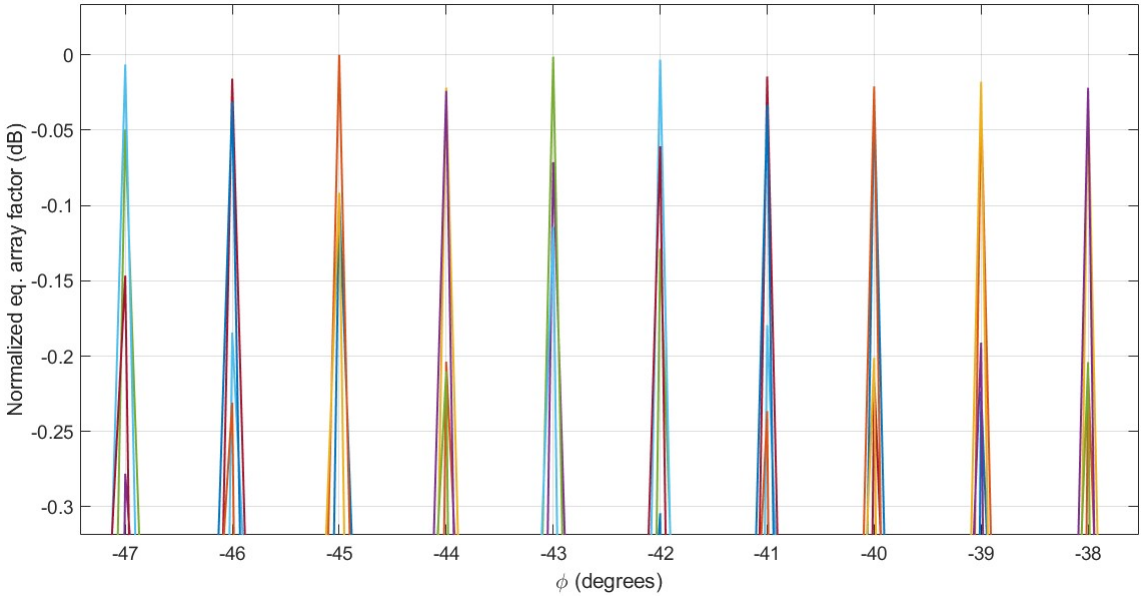


Figure 4.21: Normalized equivalent AF (dB) plot for 100 cells per dimension metaprism; the angle -42.5° is between two peaks of the equivalent AF

Moreover, the RMSE values, on average, tend to decrease as the metaprism size increases, because the SNR received by the base station increases and therefore the estimator is less wrong. For a metaprism of $100 \times 100$ cells, the highest RMSE is equal to 0.45° and this means that for all user's positions the estimate of the angle of arrival is not even half a degree wrong at most.

3. In fact, from the empirical CDF of Fig. 4.20 with 100 cells per dimension, it can be seen that more than 90% of the positions report an angle estimation error of less than 0.3°. In particular, Tab. 4.1 reports the angle estimation error relative to 90% of user's positions (CDF value = 0.9) and it is possible to note how this decreases as the number of cells (i.e. dimensions) of the metaprism increases.

| N. of cells per dimension | Angle estimation error |
|---|---|
| 10 | 3.2° |
| 30 | 0.85° |
| 50 | 0.33° |
| 100 | 0.25° |

Table 4.1: Angle estimation error relative to 90% of user's positions, as the number of metaprism cells varies

Going from 10 to 100 cells per dimension, the estimation error decreases by more than a 10 factor. Furthermore, as the metaprism size increases, the percentage of user's positions, among all those evaluated, relating to an angle estimation error of only 0.2° for example, increases. From Fig. 4.22, which shows these values in the case of metaprism with 10 cells and 100 cells, it is possible to note that the percentage of user's positions relative to an estimate of the angle with a maximum error of 0.2° increases from less than 10% to about 80%.

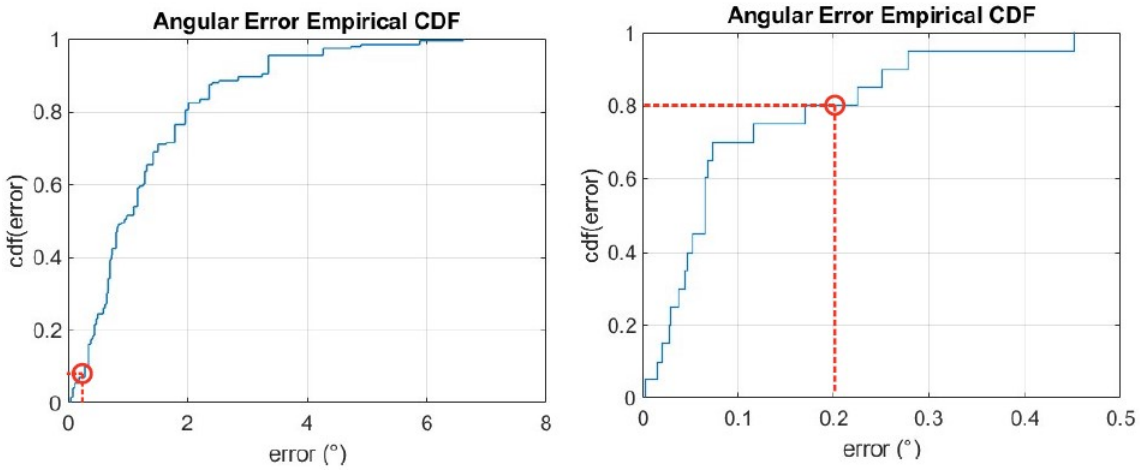


Figure 4.22: Empirical CDFs relating to angle of arrival estimation for 10 cells per dimension metaprism (left) and 100 cells per dimension metaprism (right)

More precisely, the Fig. 4.23 reports the percentage values of the empirical CDF relating to an estimation error on the angle of arrival of 0.2° for all the metaprism dimensions/number of cells evaluated. In any case the accuracy in angle estimation is very good.

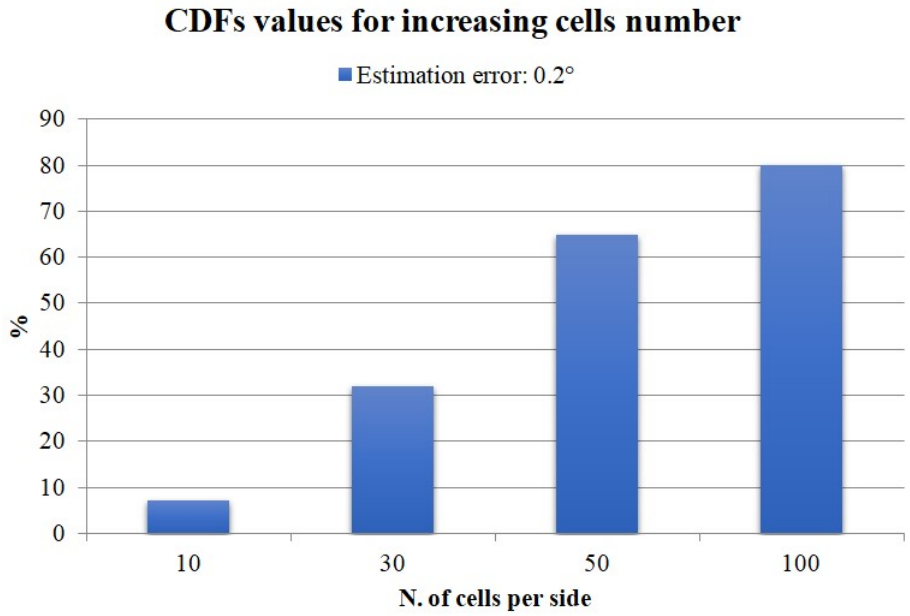


Figure 4.23: Empirical CDF percentage values, related to an angle estimation error of 0.2°, for an increasing number of metaprism cells per dimension

The results of these examples are not absolute, i.e, they are great for describing the overall performance of the estimator, but small variations are possible, due to the randomness of the noise added by the channel and of the noise introduced by the perturbation of the test grid.

### 4.2.2 Simulations results as the number of sub-carriers varies

The performance of the estimator is reported as the number of signal sub-carriers varies: 16, 64, 256, 1024. Let us consider a metaprism of $50 \times 50$ cells, while all other simulation parameters remain unchanged from those reported in (Sec. 4.2). For each simulation with a different number of sub-carriers, the corresponding equivalent metaprism AF is also reported where, in each plot, not all sub-carriers are shown but only one out of ten (and one out of fifty for the 1024 sub-carriers case). The plot of the equivalent AF is possible thanks to a specially created Matlab script, based on Eq. 2.19.

16 sub-carriers

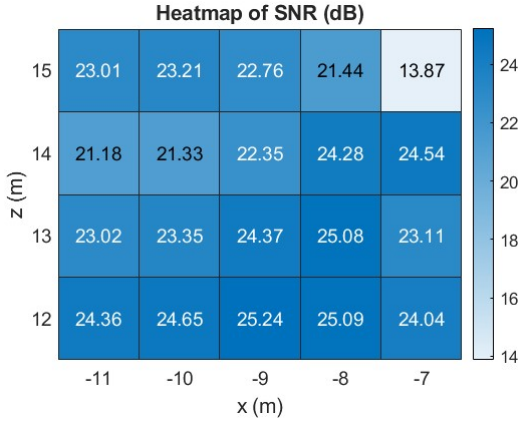


Figure 4.24: SNR Heatmap for 16 sub-carriers signal

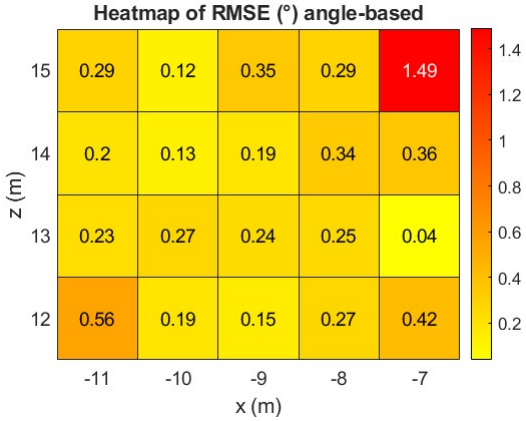


Figure 4.25: RMSE Heatmap for 16 sub-carriers signal

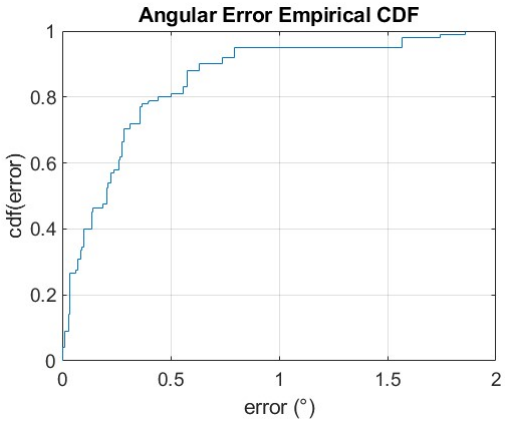


Figure 4.26: Empirical CDF for 16 sub-carriers signal

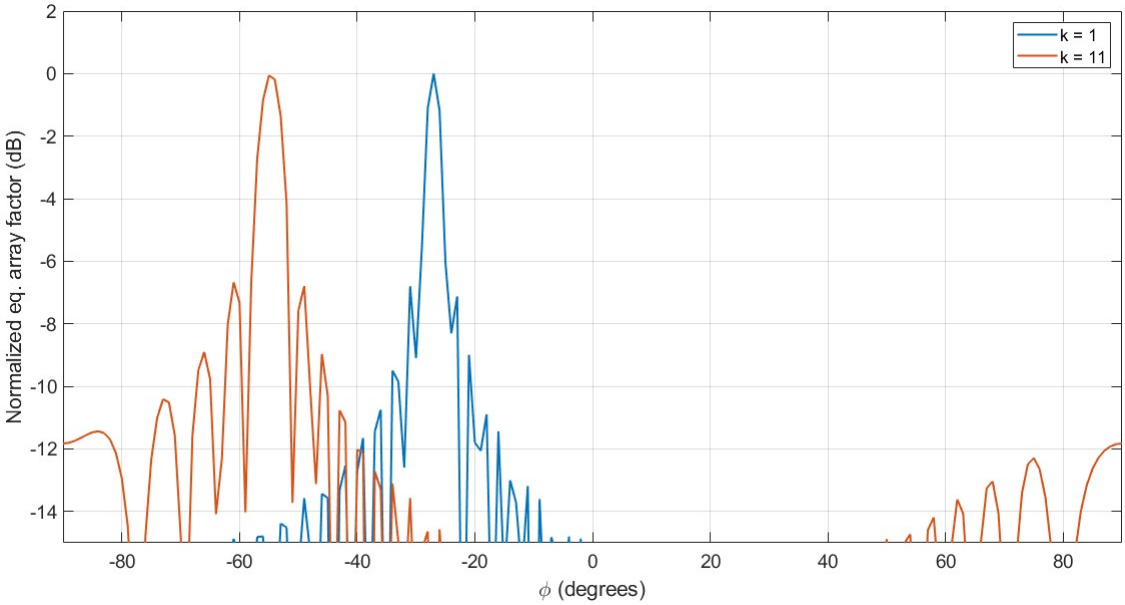


Figure 4.27: Equivalent AF for 16 sub-carriers signal

64 sub-carriers

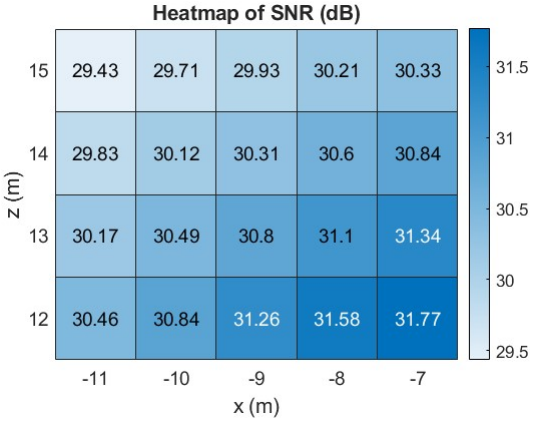


Figure 4.28: SNR Heatmap for 64 sub-carriers signal

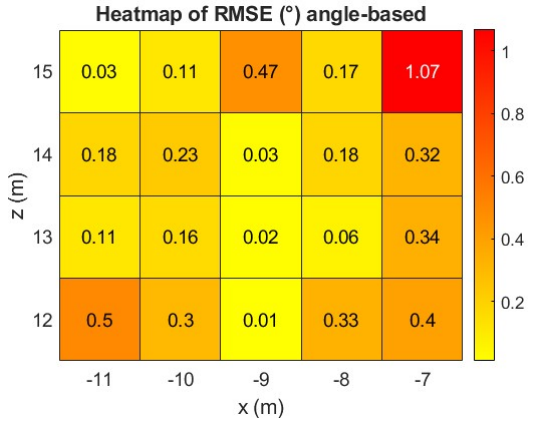


Figure 4.29: RMSE Heatmap for 64 sub-carriers signal

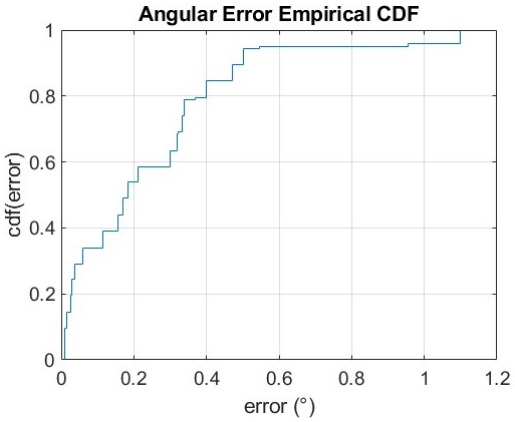


Figure 4.30: Empirical CDF for 64 sub-carriers signal

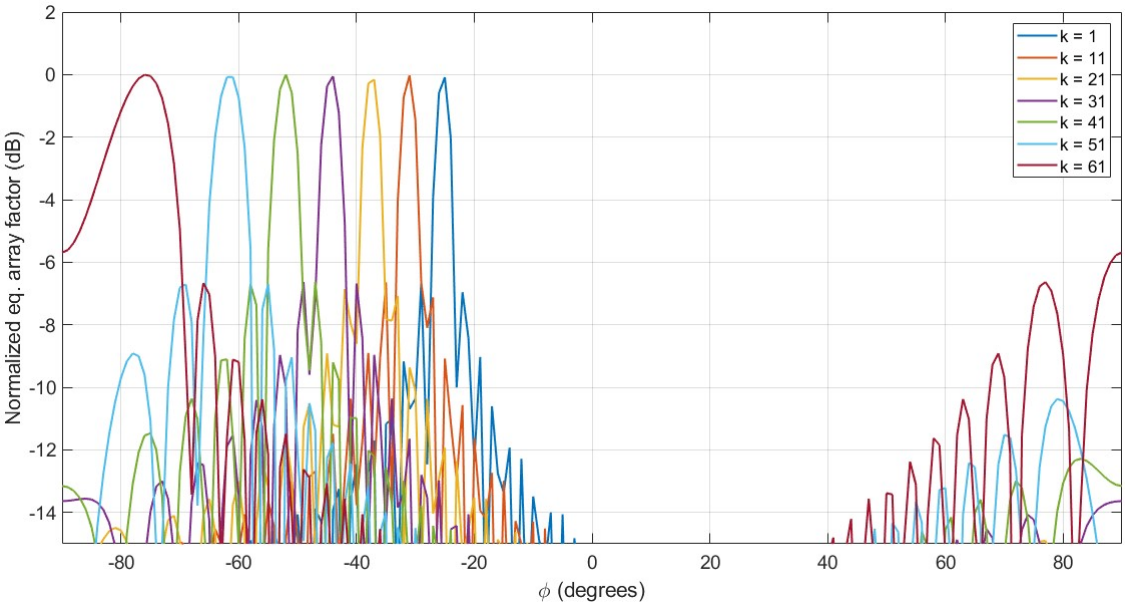


Figure 4.31: Equivalent AF for 64 sub-carriers signal

256 sub-carriers

For the case of 50 cells per dimension metaprism and 256 sub-carriers, only the relative equivalent AF is shown, given that the performance of the estimator is reported in the previous section (4.15, 4.16 and 4.17).

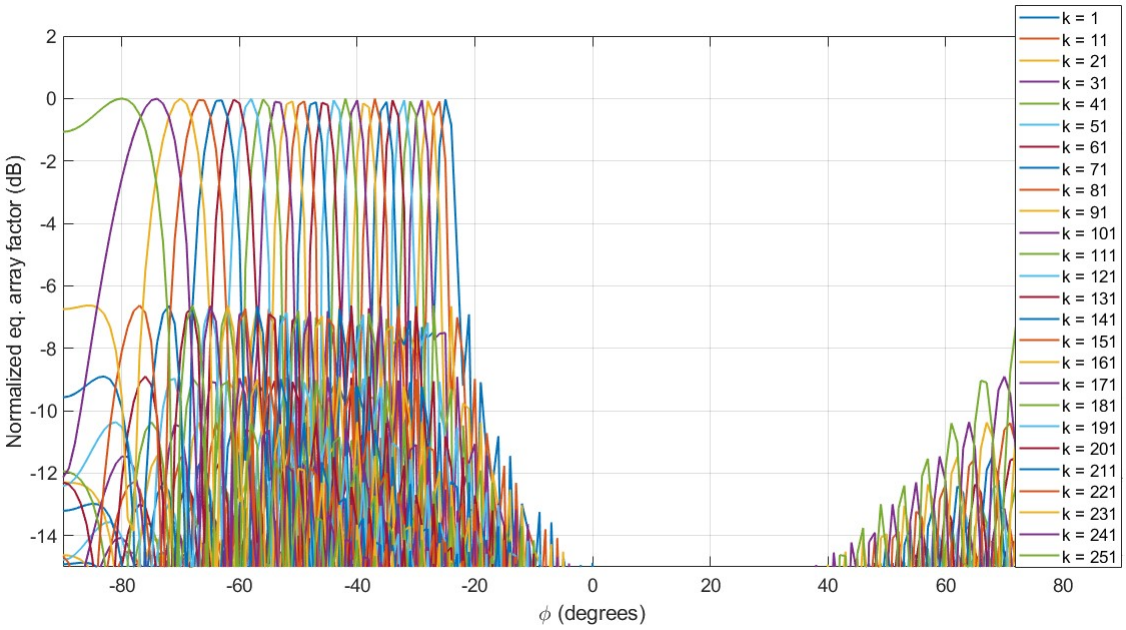


Figure 4.32: Equivalent AF for 256 sub-carriers signal

1024 sub-carriers

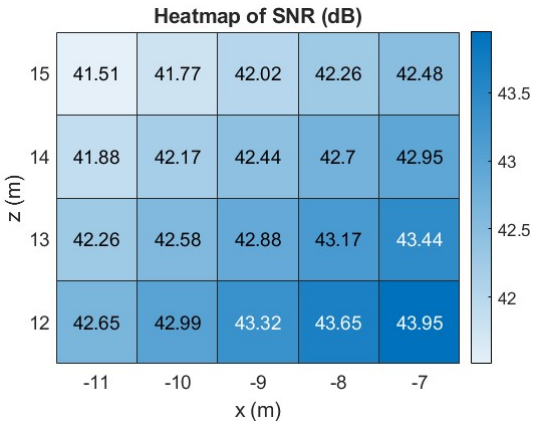


Figure 4.33: SNR Heatmap for 1024 sub-carriers signal

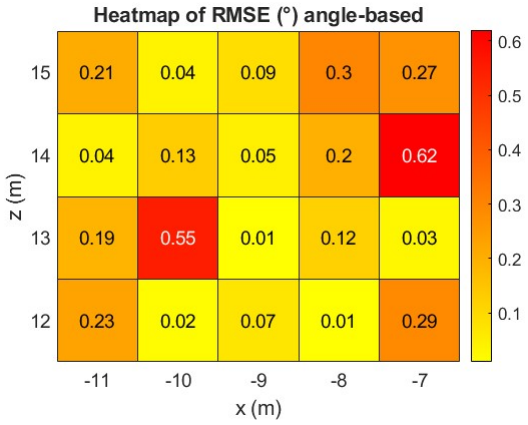


Figure 4.34: RMSE Heatmap for 1024 sub-carriers signal

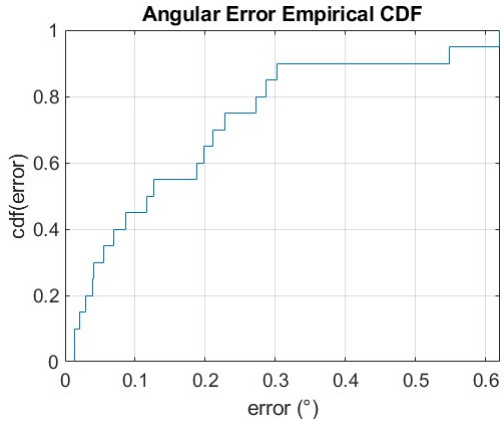


Figure 4.35: Empirical CDF for 1024 sub-carriers signal

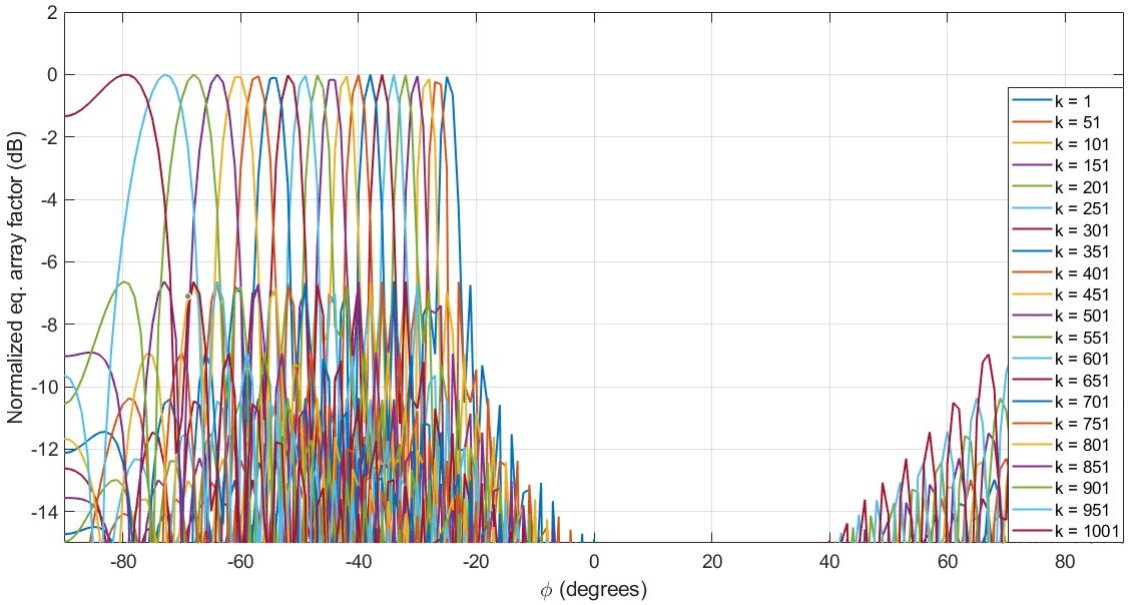


Figure 4.36: Equivalent AF for 1024 sub-carriers signal

From the previous results, it can be seen that as the number of sub-carriers increases, the performance tends to improve overall. This happens both because the SNR received by the BS increases, and because the peaks of the equivalent array factor of the metaprism tend to become increasingly dense and consequently the probability that a user transmits from a direction in correspondence of one peaks increases. For example if we consider the user's position (-9,0,15) we notice that the RMSE has a higher value in the case of 16 sub-carriers (0.35°) than in the case with 256 sub-carriers (0°). Indeed, the equivalent AF relating to 16 sub-carriers shows a null at the angle which identifies this position (about 31°), while the AF relating to 256 sub-carriers shows a peak for the same angle. The figures 4.37 and 4.38 respectively show the equivalent AF relative to 16 and 256 sub-carriers, highlighting the angle in question.

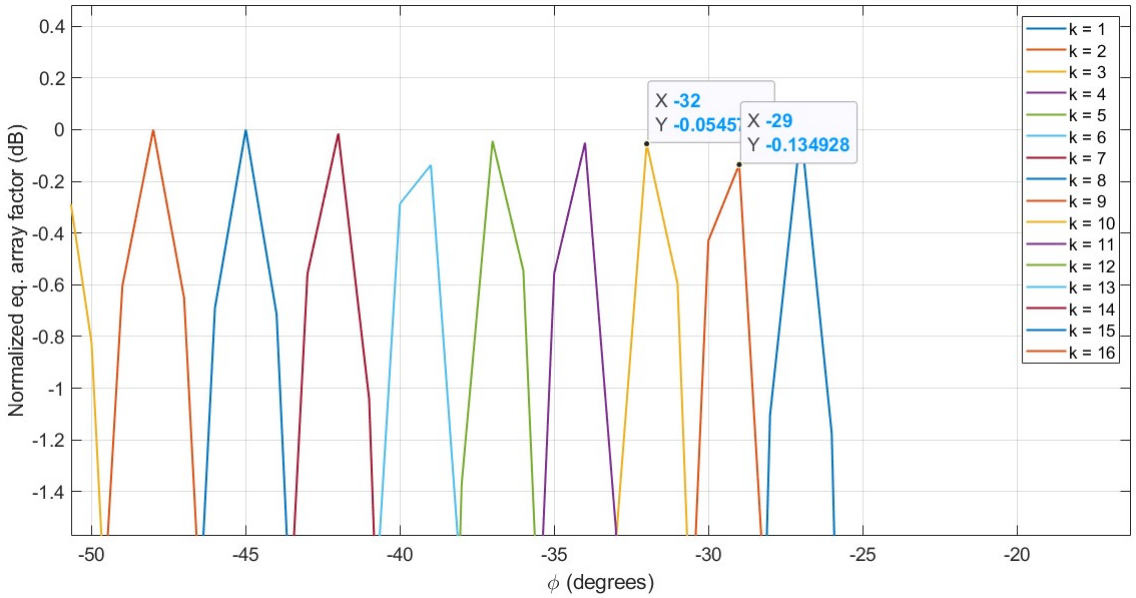


Figure 4.37: Equivalent AF for 16 sub-carriers (zoom in)

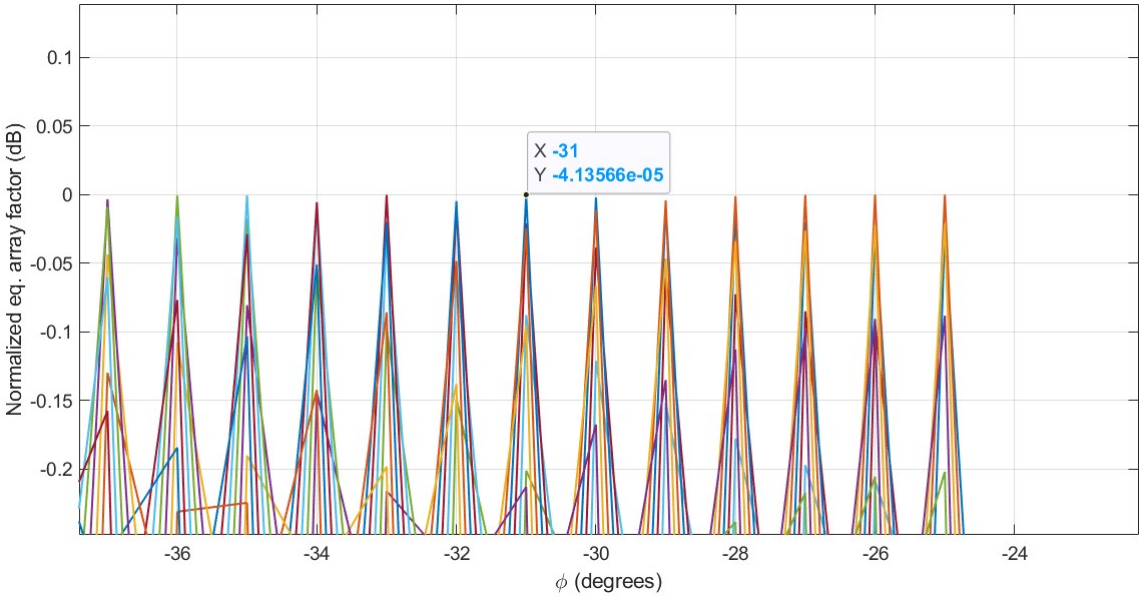


Figure 4.38: Equivalent AF for 256 sub-carriers (zoom in)

It is possible to notice the improvement of the performance also from the empirical CDFs. In fact, we see that as the number of sub-carriers increases, with the same probability (CDF value = 0.9), the estimation error on the angle of arrival decreases.

This trend is shown in Fig. 4.39 and in Tab. 4.2, in which the values of the angle estimation error it is reduced of about 0.33°, passing from considering 16 to 1024 sub-carriers of the signal.

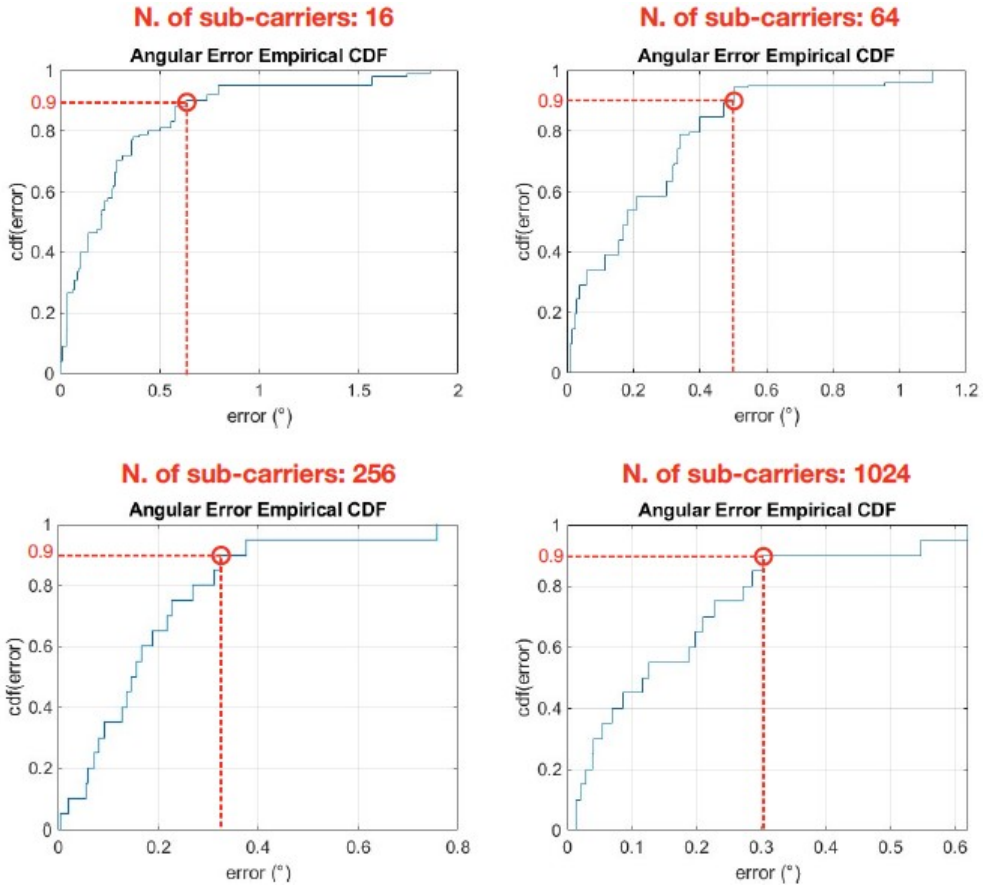


Figure 4.39: Empirical CDFs for increasing number of sub-carriers: 16, 64, 256, 1024; relative angle estimation errors, corresponding to 90% of user's positions, are highlighted

| N. of sub-carriers | Angle estimation error |
|---|---|
| 16 | 0.63° |
| 64 | 0.47° |
| 256 | 0.33° |
| 1024 | 0.3° |

Table 4.2: Angle of arrival estimation error relative to 90% of user's positions, as the number of sub-carriers varies

Equivalently, we can see improvements in the performance of the estimator as the number of sub-carriers increases, if we fix the value of the angle estimation error (0.2° in this example) and evaluate the percentage of these estimated with this maximum error. This analysis is shown by the graph in Fig. 4.40, from which it can be seen that by increasing the number of sub-carriers from 16 to 1024, the percentage of angles estimated with an error below or equal to 0.2° goes from 47.5% to 65.5%, corresponding to an increase of 18%.

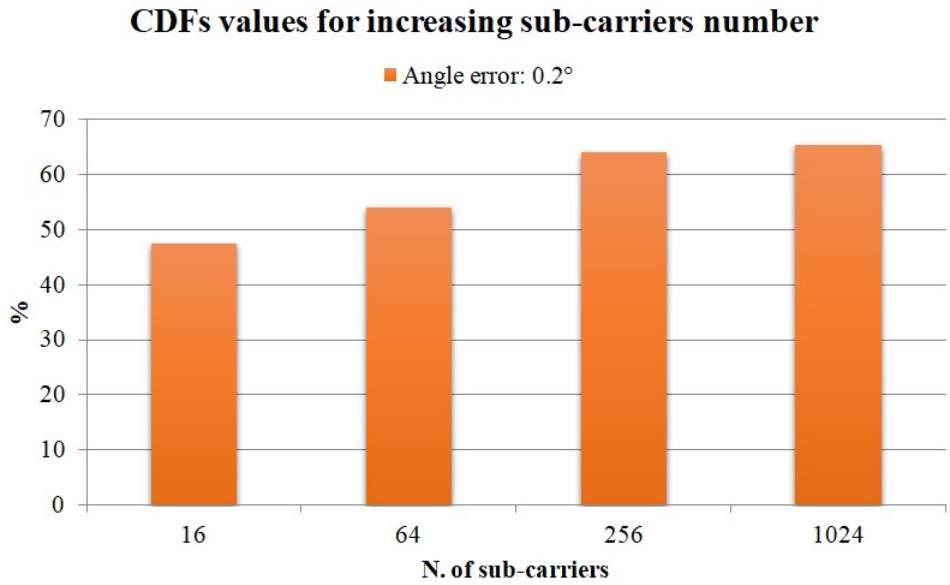


Figure 4.40: Empirical CDF percentage values, related to an angle of arrival estimation error of 0.2°, for an increasing number of signal sub-carriers

### 4.2.3 Simulation results as the resolution of the test grid varies

This subsection presents the results of simulations performed with a finer ML test grid. Since the time required to complete the simulation is proportional to the number of ML test positions evaluated, a grid with a step of 0.01 m was chosen as a minimum limit to have reasonable computational times. Below are shown the results in terms of RMSE and empirical CDF for a metaprism of 50 × 50 cells, 256 sub-carriers and with a test grid step of 0.5 m first and then reduced to 0.05 m and 0.01 m. The SNR heatmap is not reported because it is the same of the one previously reported in Fig. 4.15.

0.5 m test grid step

Below is shown the output of a simulation performed with a test grid step of 50 cm, increased by 5x from the one of 10 cm, used in all previous simulations.

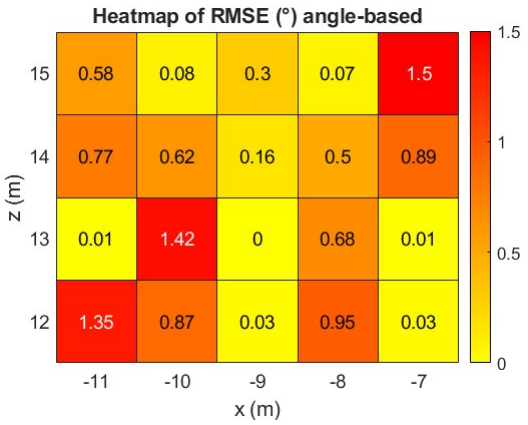


Figure 4.41: RMSE for test grid step of 0.5 m

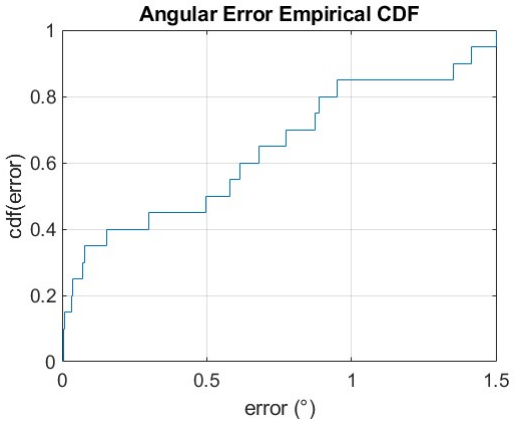


Figure 4.42: Empirical CDF for test grid step of 0.5 m

With a larger test grid step, the estimator has fewer user's positions available to test and consequently, where it struggles to locate, the error will be greater, due to poor grid resolution. This is confirmed by the Fig. 4.41 and 4.42, which report a higher estimation error compared to the one relative to 0.1 m test grid step.

0.05 m test grid step

The following results are related to a test grid with a step of 5 cm.

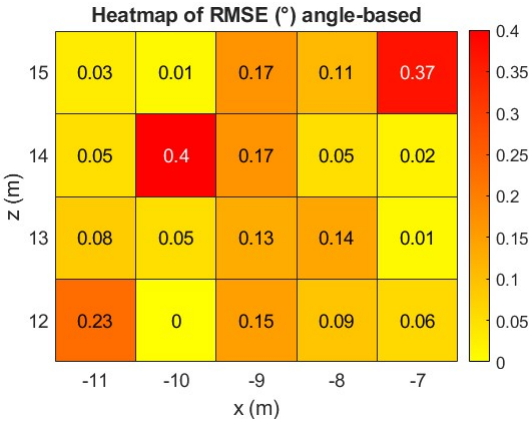


Figure 4.43: RMSE for test grid step of 0.05 m

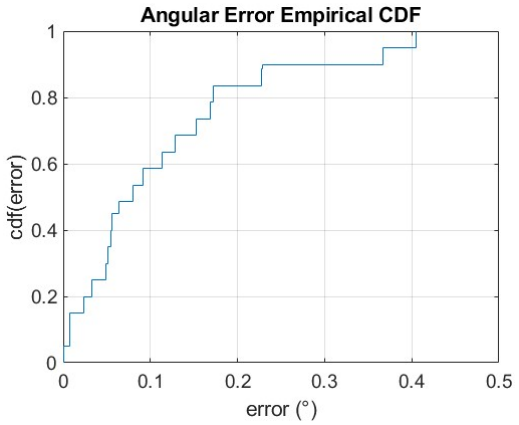


Figure 4.44: Empirical CDF for test grid step of 0.05 m

By halving the test grid step, compared to the one used up to now (10 cm), it is already possible to notice, in particular by the CDF, an improvement in the performance of the estimator (see Fig. 4.17).

0.01 m test grid step

The following results are related to a test grid with a step of 1 cm.

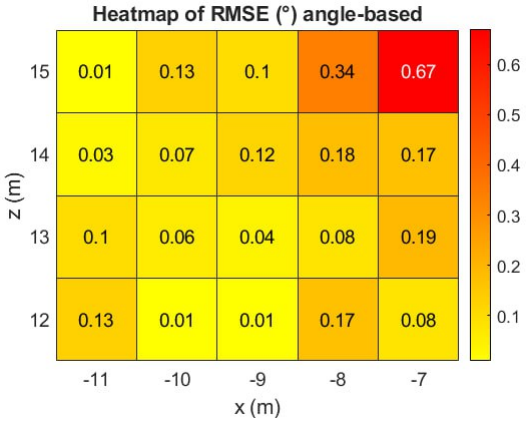


Figure 4.45: RMSE for test grid step of 0.01 m

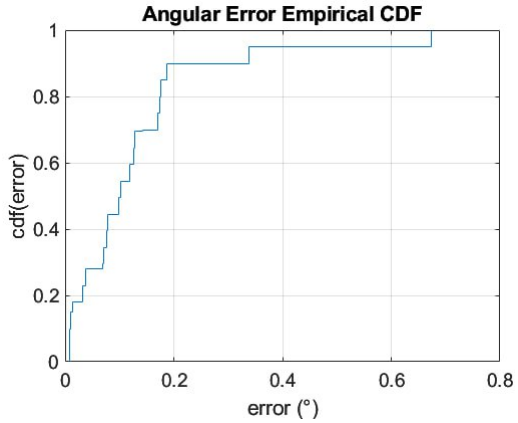


Figure 4.46: Empirical CDF for test grid step of 0.01 m

The angle of arrival estimation errors, in the case of test grid densification with a step equal to 1 cm, are considerably reduced. Also the angle relative to the position (-11,0,12), one of the two most critical, seems to be estimated with a lower error. From Fig. 4.46 it is possible to notice that the angles of arrival relative to more than 90% of user's positions are estimated with a maximum error of about 0.2°. Compared with the case of a ML test grid step of 0.5 m, the same angular error value is related to only 32% of the user's positions, to 64% for 0.1 m test grid step and to 83.5% for a step of 0.05 m test grid step. This comparison is depicted in Fig. 4.47 and also in Fig. 4.48 in a histogram form. From these figures it is possible to notice that the angles of arrival relative to a percentage of user's positions covered with a fixed angle estimation error, tends to increase as the test grid step decreases. This means that the performance of the estimator improves as the test grid resolution increases as expected.

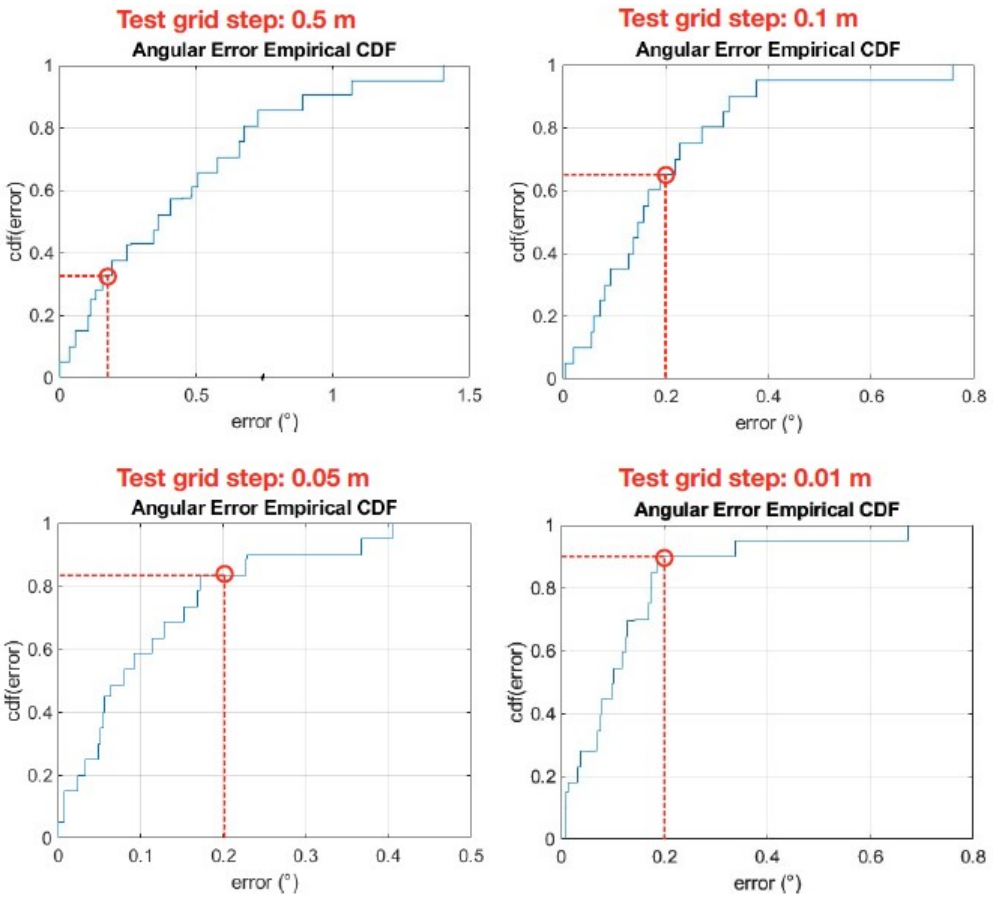


Figure 4.47: Empirical CDFs for decreasing test grid steps: 0.5 m, 0.1 m, 0.05 m, 0.01 m; relative values, corresponding to an angle estimation error of 0.2°, are highlighted

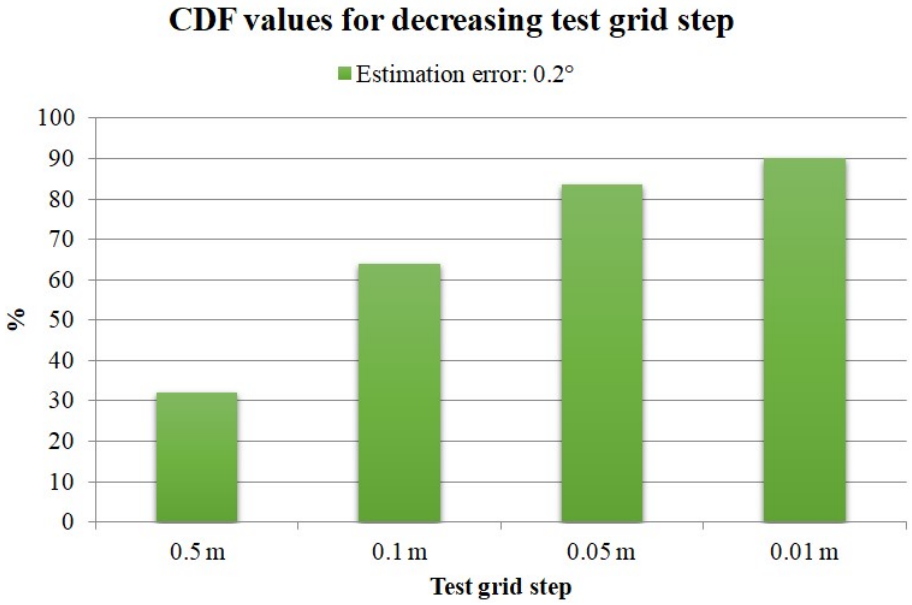


Figure 4.48: Empirical CDF percentage values, related to an angle estimation error of 0.2°, for a decreasing test grid step with a 50 cells per dimension metaprism

Also from the table 4.3 it is possible to note how the angle of arrival estimation error decreases as the step of the test grid decreases.

| **Test grid step** | **Angle estimation error** |
|---|---|
| 0.5 m | 0.9° |
| 0.1 m | 0.33° |
| 0.05 m | 0.23° |
| 0.01 m | 0.2° |

Table 4.3: Angle estimation error relative to 90% of user's positions, as the test grid step varies

From the obtained results it is possible to note that the finer the grid, i.e. the smaller the step between one ML test position and another, the better the performance of the estimator, given that the ultimate resolution of the localization depends on the discretization of the test grid. Obviously what is gained in estimation precision, is lost in terms of computational time. This improvement is more appreciable from the empirical CDF, given that, the average value represented by the RMSE could be altered by the outliers, and hence it might bring to erroneous conclusions.

### 4.2.4 Outliers

The presence of outliers, i.e. measurements far from what is expected, is one of the biggest problems in finger printing approaches.The general idea of finger printing is to measure electromagnetic quantities with the awareness that they are as different as possible from position to position, in order to be able to discriminate them. This process is based on the signature of the electromagnetic quantities, which could simply be the received power. In our case, the decision is based on the sub-carriers profile, rather than on the power, the idea is the same: to measure quantities in the hope that they are unique for each position. For certain geometric configurations, it can happen that positions that far away each other have a similar profile and in the presence of noise the simulator can get confused and instead of deciding for one point, he decides for the other. In fact, for a certain position, it can happen for example that for 9 out of 10 Monte Carlo iterations, the estimator shows good performance and then at the tenth it provides a higher estimation error. In the RMSE heatmap, since the average value of the error is displayed, it is probable that, due to these outliers, this average error increases, losing its reliability. On the contrary, the empirical CDF is indicative because even in the presence of outliers, it can be understood whether the estimator generally reports excellent estimates or not. The phenomenon of outliers should become less significant as the signal-to-noise ratio increases. In fact, the higher the SNR, the higher the probability that, even if two positions are similar, the estimator is able to discriminate them; conversely a low SNR could lead the estimator to confuse them. Furthermore, the incidence of outliers decreases as the test grid step decreases.

| **N. of iteration** | **Angle estimation** |
|---|---|
| 1 | -36.61° |
| 2 | -36.61° |
| 3 | -36.61° |
| 4 | -36.91° |
| 5 | -36.91° |
| 6 | -36.91° |
| 7 | -36.91° |
| 8 | -36.91° |
| 9 | -36.91° |
| 10 | -36.61° |

Table 4.4: AOA estimated values for user's position (-11,0,15), one for each Monte Carlo iteration for test grid step of 0.5 m

To verify this last statement we consider for example the angle of arrival relative to the position (-11,0,15) which reports an RMSE equal to 0.58° in the case of a test

grid step equal to 0.5 m. This average value is probably altered by the presence of at least one outlier among the 10 relating to each Monte Carlo iteration. These are presented in Tab. 4.4. From this one it is possible to see how the estimator oscillates in its decision between two possible angle of arrival estimation values. -36.61° is the angle relative to the estimated position (-11.51,0,15.49), while -36.91° is the angle relative to the outlier position (-9.03,0,11.98) that shifts the average with respect to the true value of the arrival angle (-36.25°), thus causing an increase in the RMSE value. In practice, if we consider the user's true positions, due to the noise, the estimator sometimes provides as an estimate the user's position (9,0,12) which is an outlier compared to the real true one (-11,0,15). This also confirms the fact that, in a far-field scenario, all the points corresponding approximately to the same angle, most likely provide a similar sub-carriers profile, which leads the ML to make the wrong choice. The Fig. 4.49 shows the user's true position and the outlier position.

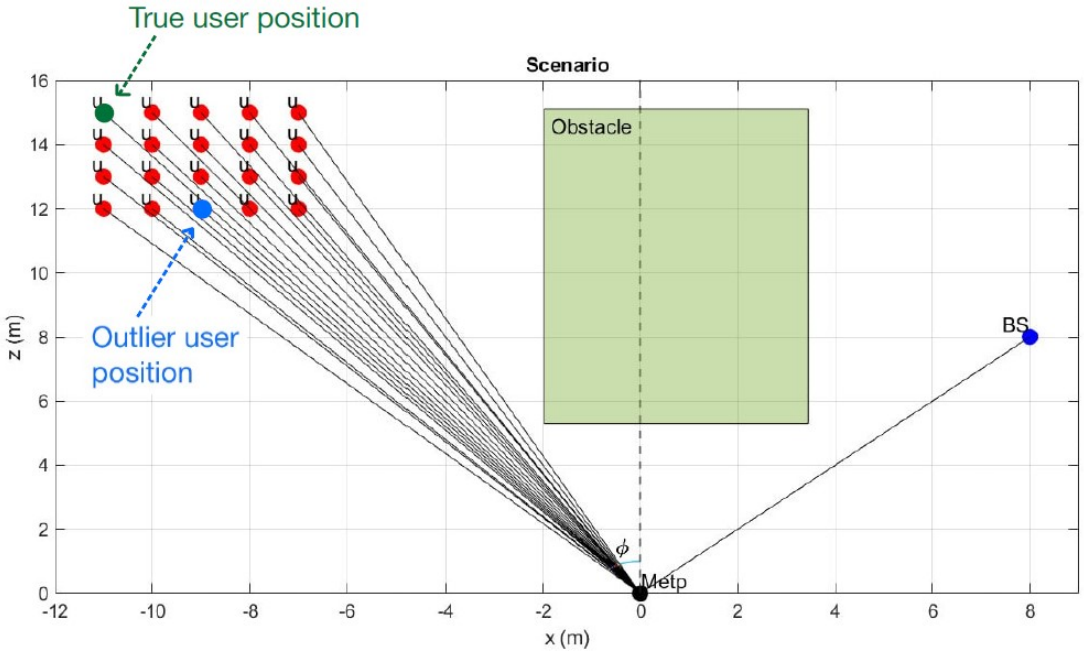


Figure 4.49: True user's position and outlier user's position estimated

On the contrary, from the Fig. 4.45, relating to the simulation with test grid step equal to 0.01 m, it is possible to notice how the estimation of the angle, corresponding to the same true position of the user, is clearly improved, reporting a minimum estimation error equal to the resolution of the grid related to the user's position (-11.03,0,-15.01). In Tab. 4.5 are shown the values provided by the estimator for each Monte Carlo iteration and the absence of outliers is evident.

| N. of iteration | Angle estimation |
|---|---|
| 1 | -36.26° |
| 2 | -36.26° |
| 3 | -36.26° |
| 4 | -36.26° |
| 5 | -36.26° |
| 6 | -36.26° |
| 7 | -36.26° |
| 8 | -36.26° |
| 9 | -36.26° |
| 10 | -36.26° |

Table 4.5: AOA estimated values for user's true position (-11,0,15), one for each Monte Carlo iteration for test grid step of 0.01 m

This confirms the fact that outliers decrease with decreasing test grid step.

### 4.2.5 Simulations results in the near-field scenario with the beamsteering design criteria for the metaprism

This subsection shows the main results obtained from a simulation carried out always considering the metaprism designed for beamsteering, but assuming a near-field scenario, i.e. with user's positions closer to the metaprism, such as those shown in Fig. 4.2. For this simulation, 256 sub-carriers and a metaprism with $50 \times 50$ cells were considered.

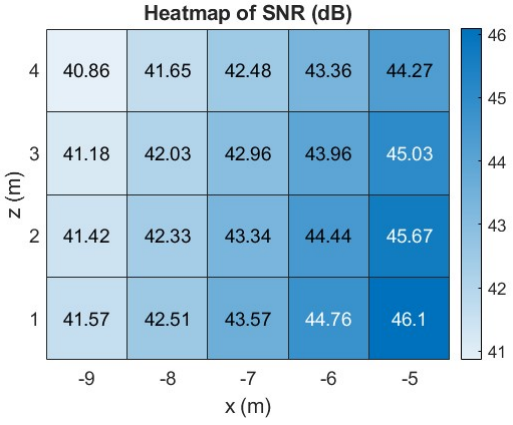


Figure 4.50: SNR Heatmap for near-field scenario

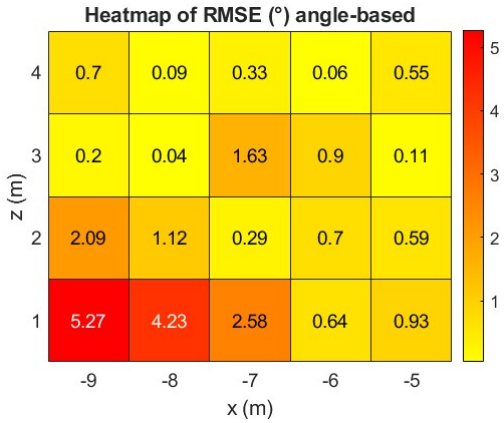


Figure 4.51: RMSE Heatmap for near-field scenario

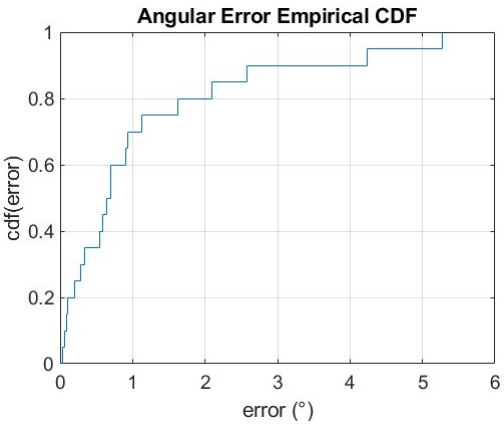


Figure 4.52: Empirical CDF for near-field scenario

Compared to the results related to the far-field scenario in the case of metaprism with $50 \times 50$ cells and 256 signal sub-carriers, the Fig. 4.51 and 4.52 show different critical user's position for localization and generally worse performance in terms of angle estimation. In particular, an RMSE in general higher for each user's position and an empirical CDF which reports about 30% of positions with an angle estimation error higher than 1°, while the maximum error was 0.8° in the far-field case, as shown in Fig. 4.17.

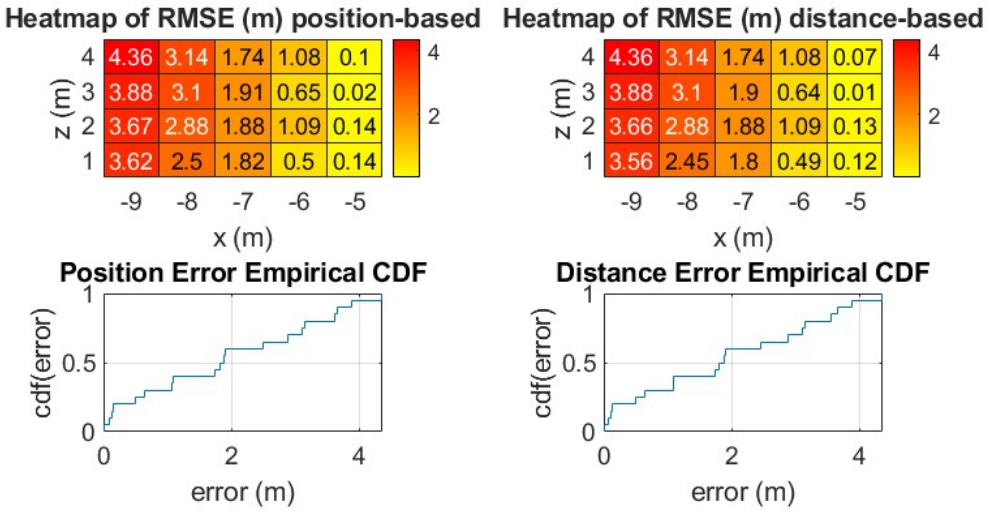


Figure 4.53: RMSE and empirical CDF relative to distance from metaprism and position estimation error for near-field scenario

In Fig. 4.54 are shown the estimation errors of the angle of arrival related to 90% of the user's positions, in the case of far-field and near-field scenarios. The increase in the angle of arrival estimation error is evident, going from 0.33° for the far-field scenario to more than 2.6° for the near-field one.

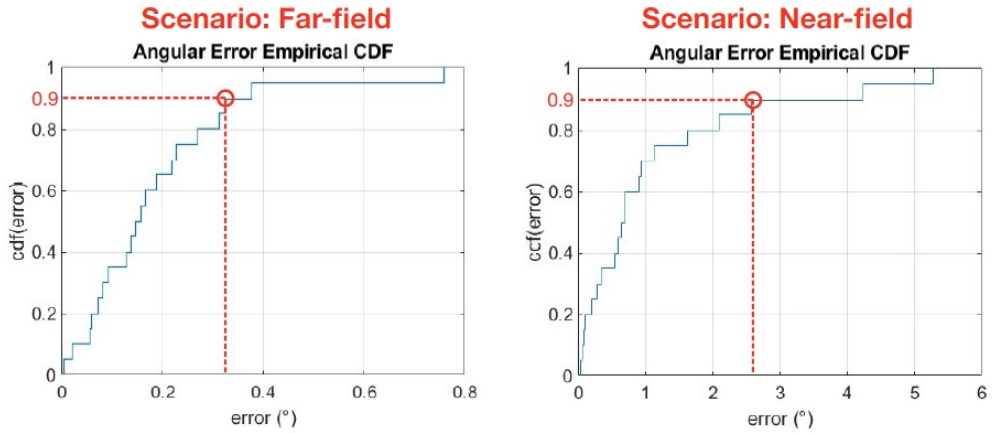


Figure 4.54: Empirical CDFs for 50 cells per dimension metaprism, test grid step 0.1 m, 256 sub-carriers in far-field scenario and near-field one; relative angle estimation errors, corresponding to 90% of user's positions, are highlighted

The Fig. 4.53 shows the RMSEs and empirical CDFs relating to distance from metaprism and position estimation. It is possible to note an error in estimating the position and distance even higher than 4 meters. This verifies that designing the metaprism

for beamsteering and using it in the near-field scenario is not optimal.

The next section shows the performance of the estimator evaluated in the metaprism random design case, which lead to an appreciable improvement for a near-field scenario.

## 4.3 Estimator performance in the metaprism random design case

The simulations results carried out in the case of metaprism random design are reported below, as the main parameters that influence the performance of the estimator vary, in particular the number of sub-carriers and the metaprism size. In the following, the performance of the estimator is presented firstly with user's positions in the near-field region, then considering user's positions further away from the metaprism, and in the end with BS in the near-field of the metaprism. These performances are reported in terms of RMSE heatmaps and empirical CDF, relating to errors in estimation of the angle of arrival, the distance from the metaprism and the user's position. In fact the random design of the metaprism coefficients for user's positions in the near-field, provides the possibility to estimate the position, which was not possible in the case of beamsteering design. Consequently, everything about the estimated position is significant, i.e. angle of arrival, distance from metaprism and position itself.

The Fig. 4.55 and 4.56 show the simulation output for a near-field scenario in the case of metaprism with $50 \times 50$ cells and random design coefficients. As regards the values of the main parameters, in this and in the following simulations, reference is made to those in (Sec.4.2), unless a modification is explicitly declared.

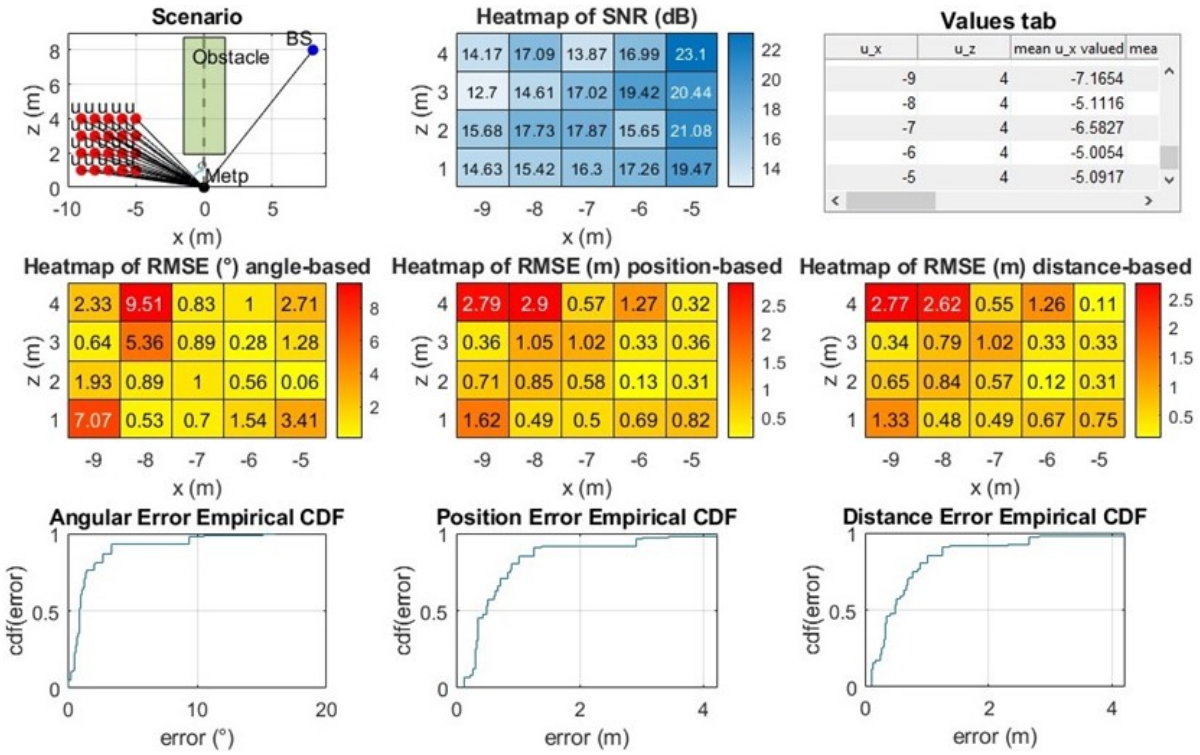


Figure 4.55: Estimator performance for near-field scenario, Pt=10 dBm, $50 \times 50$ cells and random design metaprism

| u_x | u_z | mean u_x valued | mean u_z valued | true distance from Metp (m) | mean valued distance from Metp (m) | true phi(°) | mean phi valued(°) |
|---|---|---|---|---|---|---|---|
| -9 | 1 | -8.1267 | 1.3588 | 9.0554 | 8.2395 | -83.6598 | -80.5078 |
| -8 | 1 | -7.5604 | 1.0103 | 8.0623 | 7.6276 | -82.8750 | -82.3886 |
| -7 | 1 | -6.5006 | 1.0103 | 7.0711 | 6.5786 | -81.8699 | -81.1659 |
| -6 | 1 | -5.3133 | 1.0297 | 6.0828 | 5.4121 | -80.5377 | -79.0319 |
| -5 | 1 | -5.6416 | 1.4810 | 5.0990 | 5.8327 | -78.6901 | -75.2908 |
| -9 | 2 | -8.5542 | 2.2052 | 9.2195 | 8.8339 | -77.4712 | -75.5442 |
| -8 | 2 | -7.1640 | 1.9049 | 8.2462 | 7.4130 | -75.9638 | -75.1098 |
| -7 | 2 | -6.5489 | 1.7553 | 7.2801 | 6.7801 | -74.0546 | -74.9960 |
| -6 | 2 | -5.9073 | 1.9049 | 6.3246 | 6.2069 | -71.5651 | -72.1275 |
| -5 | 2 | -5.2907 | 2.1103 | 5.3852 | 5.6961 | -68.1986 | -68.2545 |
| -9 | 3 | -8.7064 | 2.7943 | 9.4868 | 9.1438 | -71.5651 | -72.2059 |
| -8 | 3 | -7.7860 | 3.2278 | 8.5440 | 8.4285 | -69.4440 | -67.4827 |
| -7 | 3 | -6.1049 | 2.5045 | 7.6158 | 6.5987 | -66.8014 | -67.6948 |
| -6 | 3 | -5.6892 | 2.8789 | 6.7082 | 6.3762 | -63.4349 | -63.1594 |
| -5 | 3 | -5.2129 | 3.2883 | 5.8310 | 6.1633 | -59.0362 | -57.7563 |
| -9 | 4 | -7.1654 | 3.2139 | 9.8489 | 7.8532 | -66.0375 | -65.8423 |
| -8 | 4 | -5.1116 | 3.7232 | 8.9443 | 6.3238 | -63.4349 | -53.9307 |
| -7 | 4 | -6.5827 | 3.6367 | 8.0623 | 7.5205 | -60.2551 | -61.0806 |
| -6 | 4 | -5.0054 | 3.2125 | 7.2111 | 5.9477 | -56.3099 | -57.3073 |
| -5 | 4 | -5.0917 | 3.6928 | 6.4031 | 6.2898 | -51.3402 | -54.0484 |

Figure 4.56: True and estimated values table for near-field scenario, Pt=10 dBm, $50 \times 50$ cells and random design metaprism

It is possible to notice that the performance of the estimator is not very good. This strong penalty regarding the link budget, is due to the fact that, in the beamsteering design, the gain of the metaprism was all aimed at a certain angle while with the random design of the metaprism, due to inherent randomness, a much lower gain will be obtained as we can no longer speak of the gain of the entire array. This can also be seen from the SNR values which, for the same positions considered, with the same parameter values, in this case are much lower (see Fig. 4.15). To improve the performance of the estimator, in addition to considering a larger metaprism that can collect more energy, one possibility is to use a higher transmit power. For this reason, in analyzing the performance of the estimator in this case of random design, we consider a transmission power of 15 dBm.

The random design of the metaprism coefficients ensures that the frequency profile received by the BS is unique or almost unique for each user's position, becoming a sort of signature of the position itself which therefore allows its localization. For each true position of the user, the estimator compares the profile received with all the possible profiles relating to each ML test position and reports, as an estimate, the ML test position with the profile most similar to the one received. By way of example, the Fig. 4.57 shows, for each true position of the user considered, the sub-carriers profile of the useful received signal $\underline{s}$. It is possible to notice how these profiles are quite different from each other.

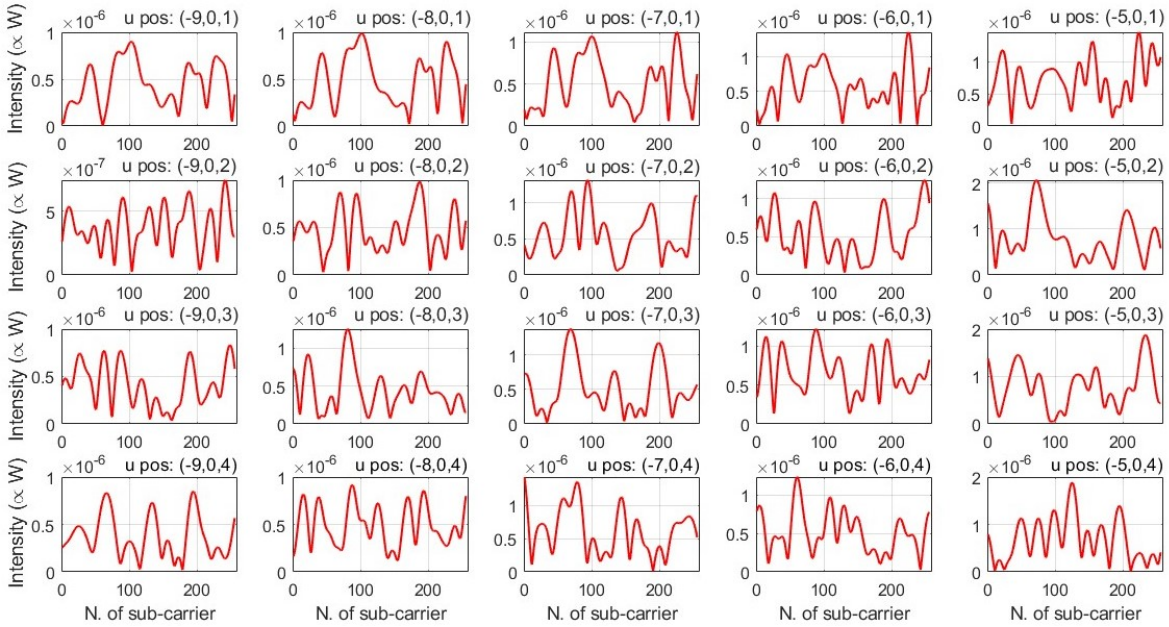


Figure 4.57: Signal profiles, as a function of the number of sub-carrier, for all user's true positions, obtained with a metaprism of 50 cells per dimension

The following subsections show the performance of the estimator for user's positions in the near-field scenario, while a specific subsection will be dedicated to the performance of the estimator for the far-field scenario.

### 4.3.1 Simulations results as the metaprism size varies

As for the beamsteering design case, the performance of the estimator is reported as the number of metaprism cells varies: 50, 100 and 200 per dimension. The values of all the other parameters are those of (Sec. 4.2).

50 cells per dimension (metaprism size: $0.27m \times 0.27m$)

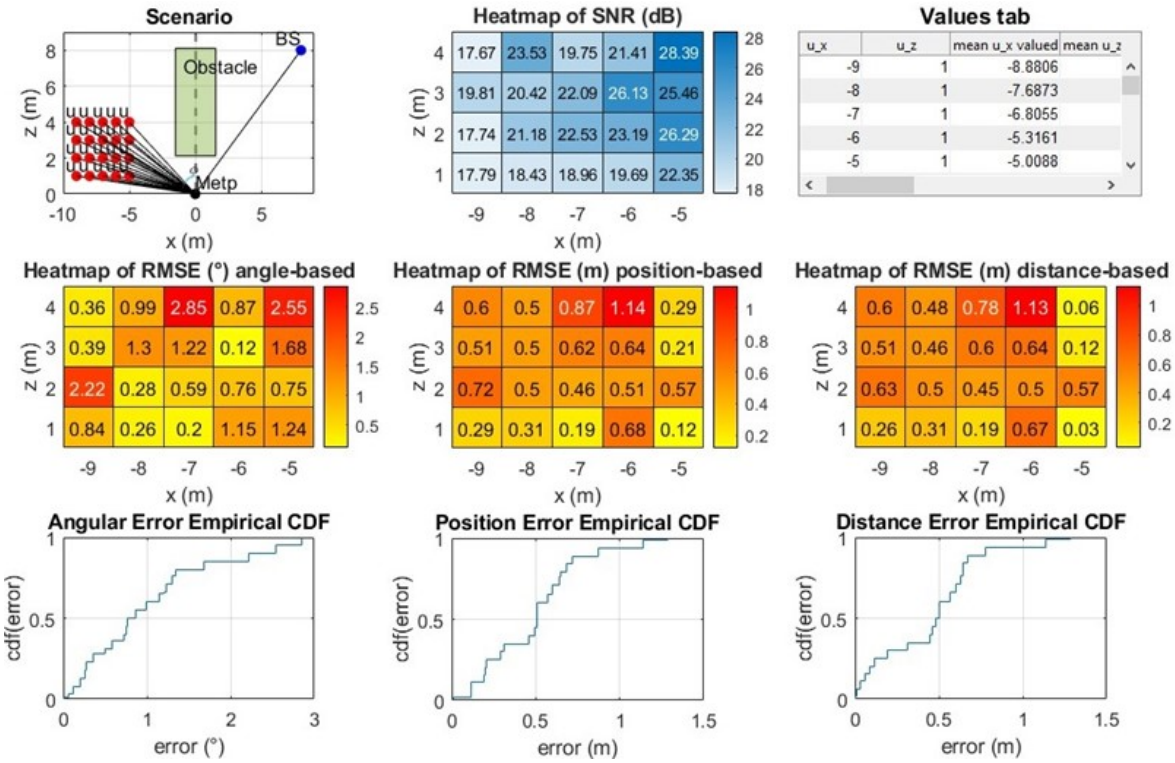


Figure 4.58: Estimator performance for $50 \times 50$ cells metaprism

100 cells per dimension (metaprism size: $0.54m \times 0.54m$)

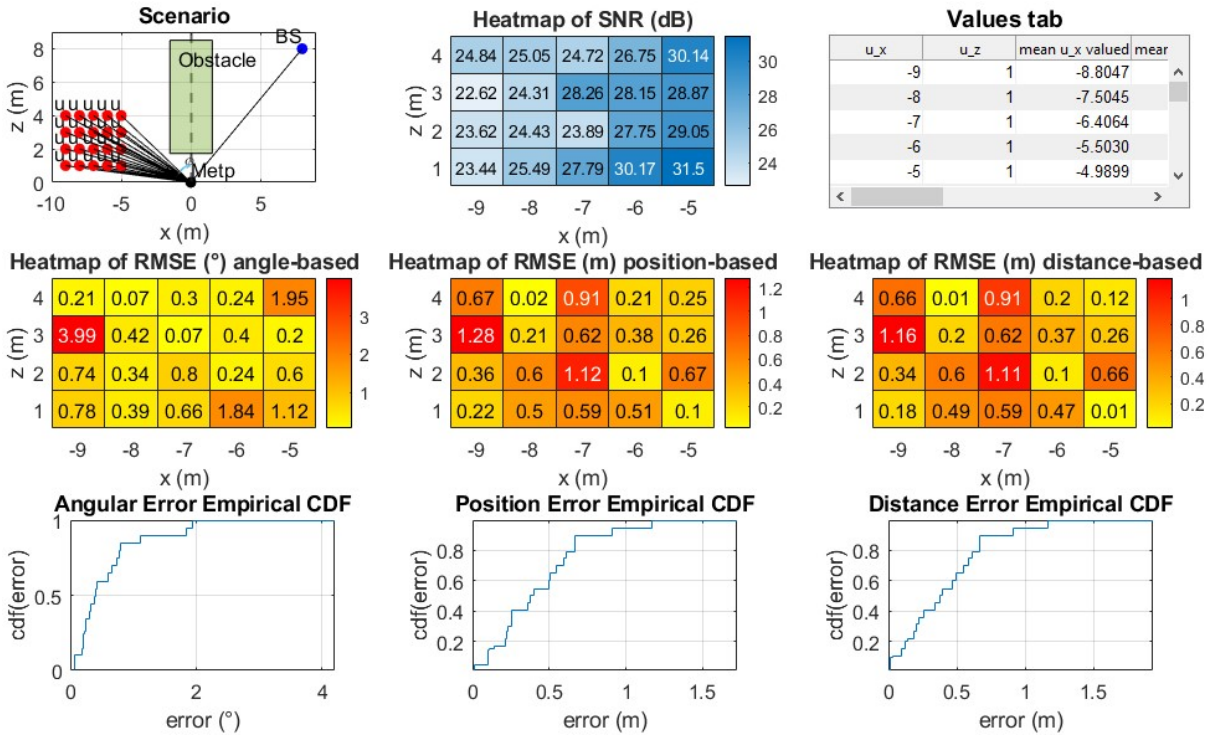


Figure 4.59: Estimator performance for $100 \times 100$ cells metaprism

200 cells per dimension (metaprism size: $1.08m \times 1.08m$)

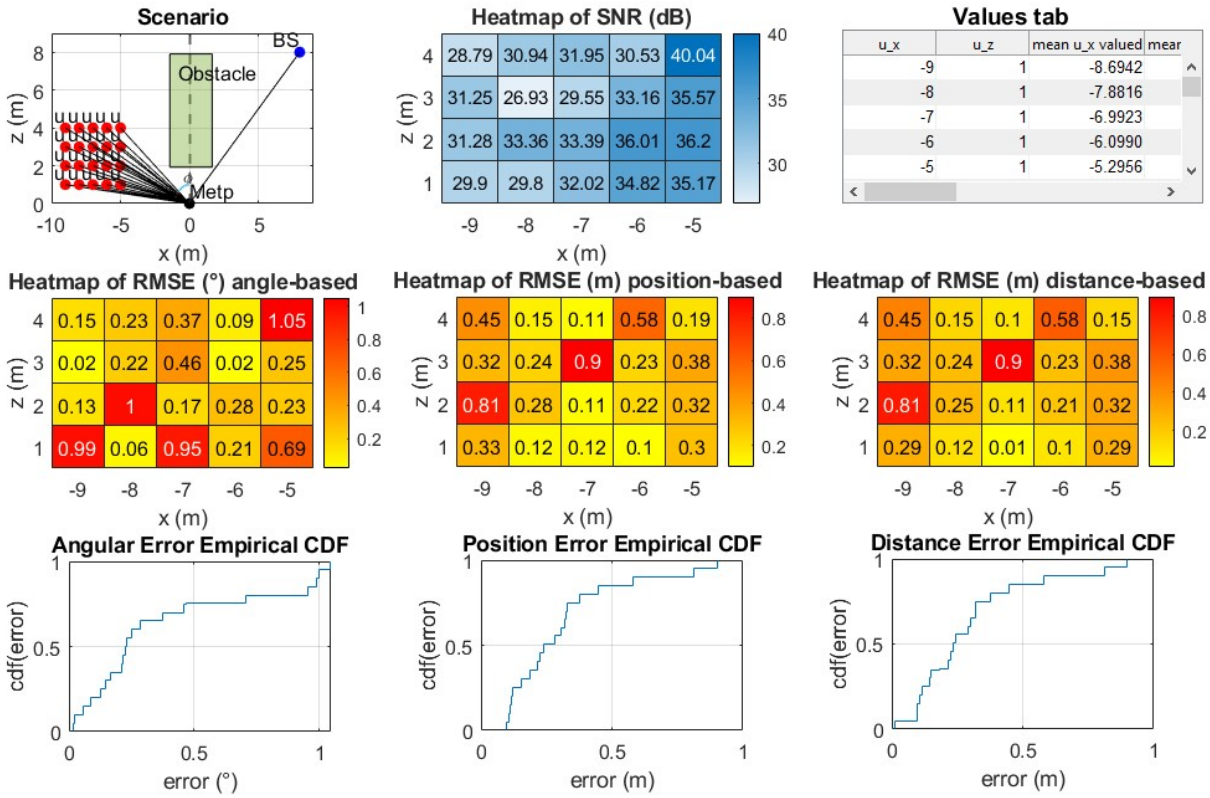


Figure 4.60: Estimator performance for $200 \times 200$ cells metaprism

From the previous figures it can be seen that the estimates of the distance from the metaprism and of the user's own position are are not high as in the beamsteering design case. Infact, for example, if we consider the case of $200 \times 200$ cells metaprism, the mean RMSE of all user's positions estimates, is only about 29 cm. This verifies the fact that the random design of the metaprism coefficients (in order to diversify the received profiles as much as possible for different positions of the transmitting user), allows to estimate, in addition to the angle of arrival, also the distance from the metaprism and consequently the user's location. Obviously the accuracy of the estimate depends on the user's position: the closest ones are better because are more inside the near-field where the information on the distance is stronger, while the farther the positions are, the more the error on the distance increases. The histogram shown in Fig. 4.61 represents the trend of the empirical CDFs values (reported as a percentage) relating to an error of 0.5° as regards the angle estimate and to an error of 0.5 m as regards the distance estimate from the metaprism and the position, for three different metaprism dimensions: $50 \times 50$, $100 \times 100$ and $200 \times 200$ cells. From the figure we deduce that, for all three estimates (angle, distance

and position), the percentage of user's positions covered with an estimation error equal to the fixed one, tends to increase as the number of metaprism cells increases. This means that the performance of the estimator improves as the metaprism size increases.

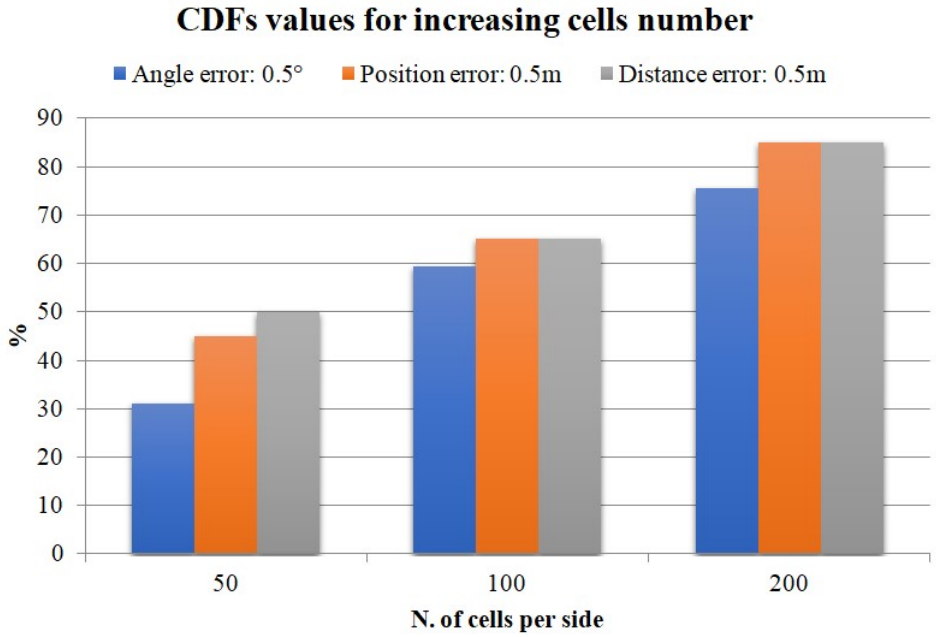


Figure 4.61: Empirical CDF percentage values related to an angle estimation error of 0.5° (blue) and to a position and distance estimation error of 0.5 m (orange and grey respectively), for an increasing number of metaprism cells

For example, if we consider the position estimation (orange bins), it is possible to notice that the number of user's positions estimated with a maximum error of 0.5 m goes from 45% for a metaprism with 50 cells per dimension, to 65% for a metaprism with 100 cells per dimension and 85% for a metaprism with 200 cells per dimension. So, with an estimation error of 0.5 m, by increasing the number of metaprism cells from 50 to 200 per dimension, it is possible to estimate 40% more user positions. Also regarding the estimation of the angle and the distance from the metaprism we have about the same percentage of improvement. Identical evaluations can be made if, instead of fixing the estimation error, we fix the user's position percentage (i.e. the value of the CDFs). In this case, the estimation error decreases as the size of the metaprism increases. The Tab. below (4.6) shows the estimation errors of angle, distance to metaprism and position relative to about 80% of user positions. In any case the position estimation error is dominated by the distance estimation error, while the angle estimation is in general very accurate.

| N. of cells per dimension | Angle estimation error | Position estimation error | Distance estimation error |
|---|---|---|---|
| 50 | 1.35° | 0.68 m | 0.64 m |
| 100 | 0.75° | 0.61 m | 0.61 m |
| 200 | 0.7° | 0.38 m | 0.38 m |

Table 4.6: Estimation errors on angle, position and distance from metaprism relative to 80% of user's positions (CDF value = 0.8), for an increasing number of metaprism cells

### 4.3.2 Simulations results as the sub-carriers number varies

The estimator performance are reported for the 64 and 1024 sub-carriers cases and then compared with the 256 sub-carriers case previously reported. Let us consider a metaprism of 100 × 100 cell, while all other simulation parameters remain unchanged from those reported in (4.2), with the transmission power kept fixed at 15 dBm as previously.

64 sub-carriers

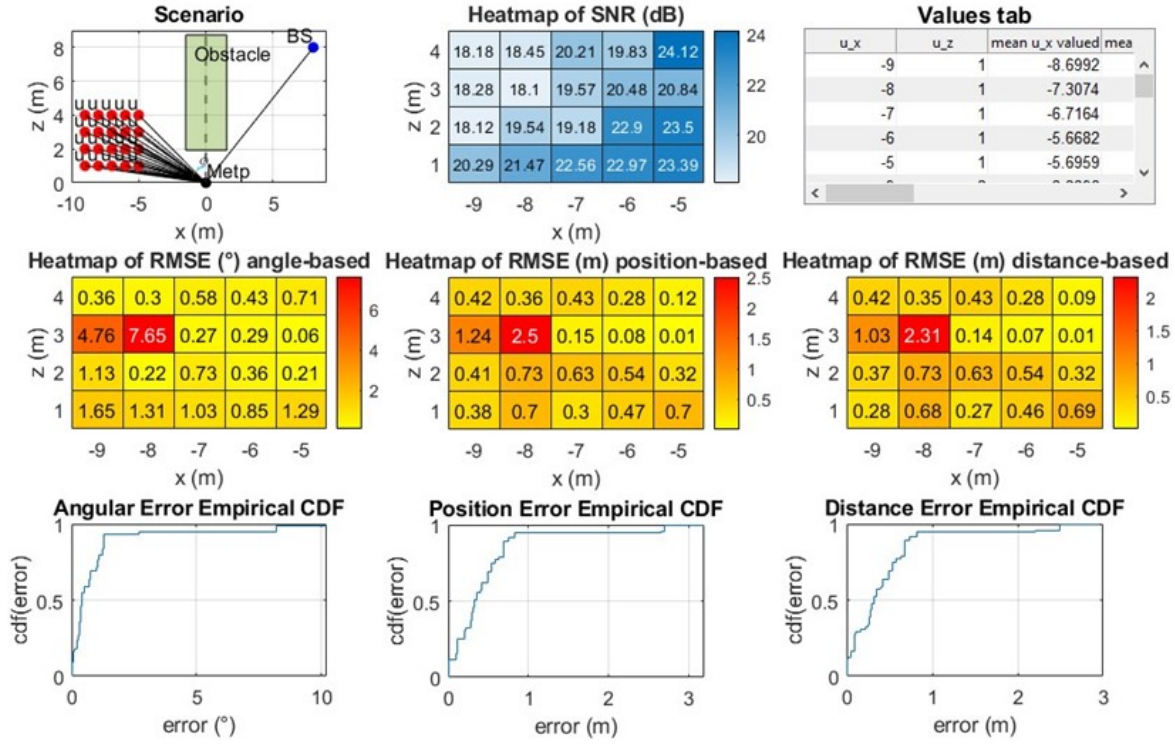


Figure 4.62: Estimator performance for 100 cells per dimension metaprism and 64 sub-carriers signal

1024 sub-carriers

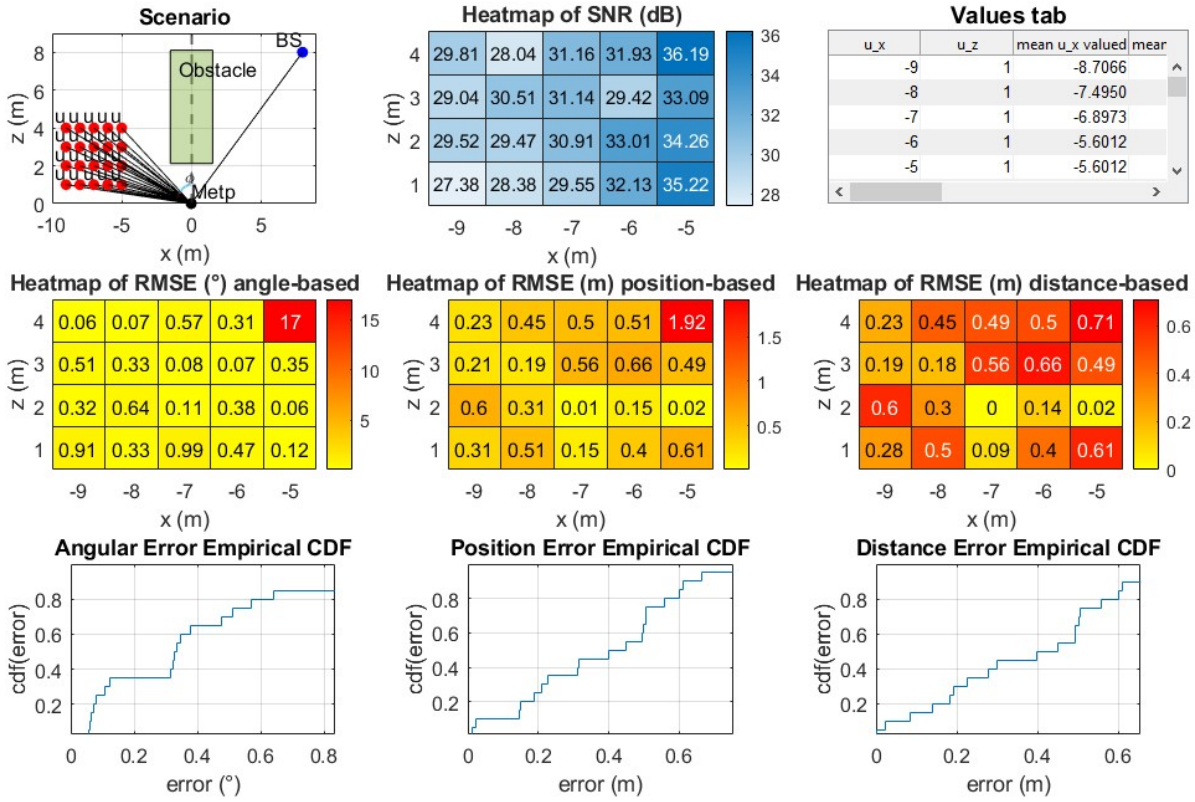


Figure 4.63: Estimator performance for for 100 cells per dimension metaprism and 1024 sub-carriers signal

As regards the performance of the estimator in the case of a signal with 256 sub-carriers, reference is made to Fig. 4.59. Again, given that the RMSE values can be perturbed by outliers, we focus on the values of the empirical CDF. The Fig. 4.63 reports the percentages of user's positions relative to a maximum angle estimation error of 0.5°, and a maximum estimation error of the position and distance from the metaprism of 0.5 m, for a signal with 64, 256 and 1024 sub-carriers. The performance of the estimator, reported in terms of values of empirical CDFs, improves as the number of signal sub-carriers considered increases. In fact, given that the calculation of the signal $\underline{s}$ depends on the number of sub-carriers, the higher this number is, the greater the signal-to-noise ratio received by the BS and therefore, in general, the better the quality of the estimate will be. Another way to verify the improvement of the estimation accuracy, as the number of sub-carriers increases, is by setting a CDF value (0.8 for example) and monitoring the value of the estimation errors as the number of signal sub-carriers varies. The Tab. 4.7 shows how, the estimation errors of angle, distance to metaprism and position relative to

about 80% of user's positions (CDF value = 0.8), decrease as the number of sub-carriers increases.

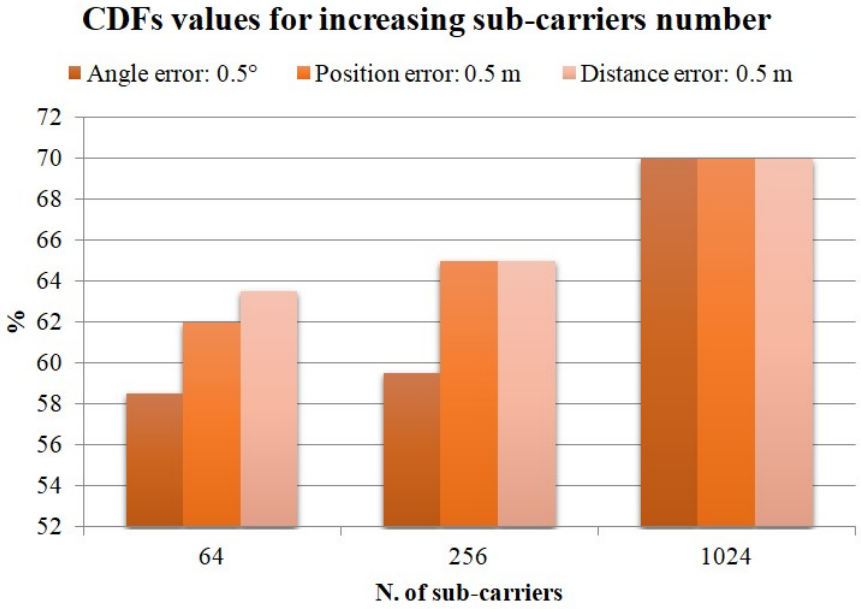


Figure 4.64: Empirical CDF percentage values related to an angle estimation error of 0.5° and to a position and distance estimation error of 0.5 m, for an increasing number of sub-carriers

| N. of sub-carriers | Angle estimation error | Position estimation error | Distance estimation error |
|---|---|---|---|
| 64 | 0.76° | 0.67 m | 0.66 m |
| 256 | 0.75° | 0.61 m | 0.61 m |
| 1024 | 0.57° | 0.56 m | 0.56 m |

Table 4.7: Estimation errors on angle, position and distance from metaprism relative to 80% of user's positions (CDF value = 0.8), for 64, 256 and 1024 sub-carriers

### 4.3.3 Simulations results as the test grid step varies

Below are shown the estimator performance in terms of RMSE and empirical CDF for a metaprism of 100 × 100 cells, 256 signal sub-carriers and with a test grid step of 0.5 m first and then 0.05 m and 0.01 m.

0.5 m test grid step

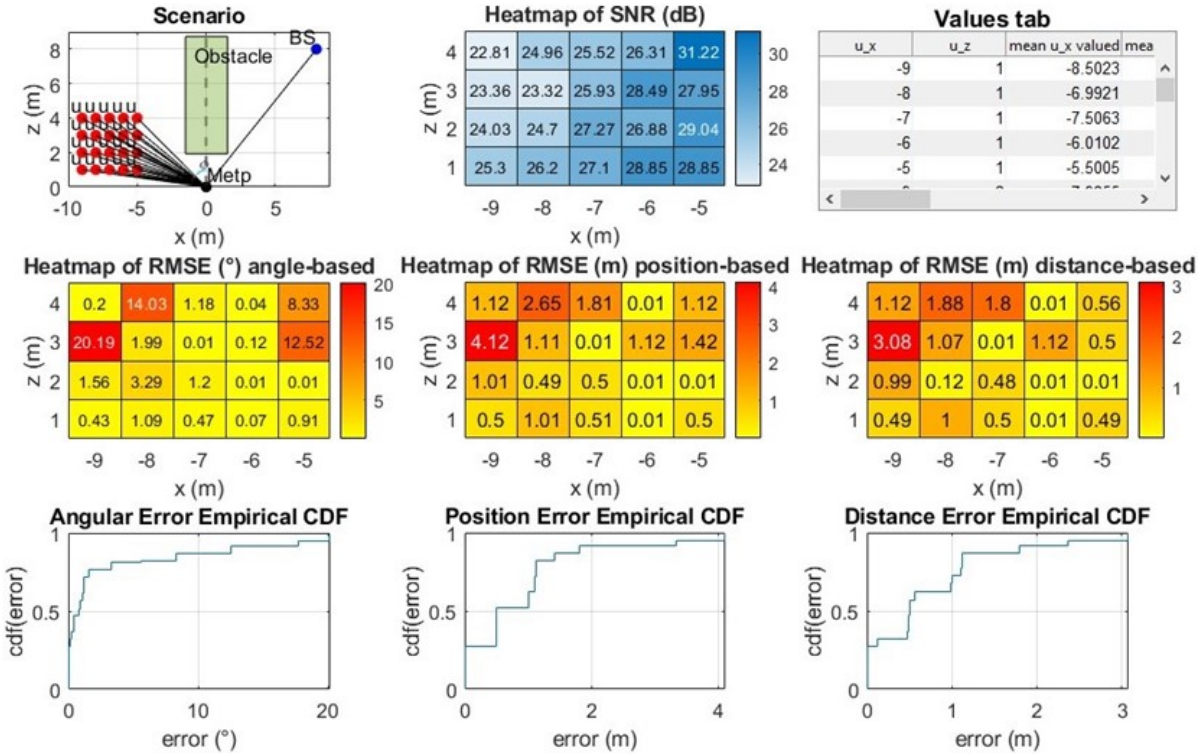


Figure 4.65: Estimator performance for a test grid step of 0.5 m

0.05 m test grid step

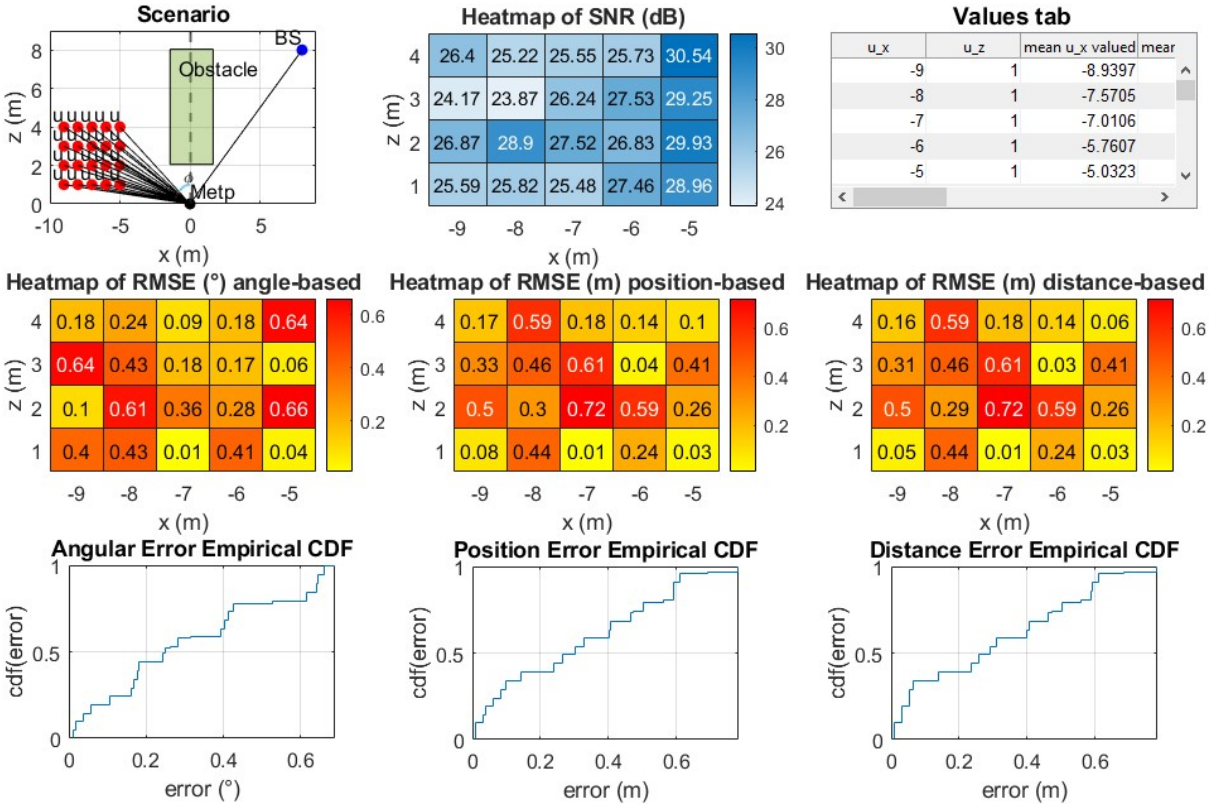


Figure 4.66: Estimator performance for a test grid step of 0.05 m

0.01 m test grid step

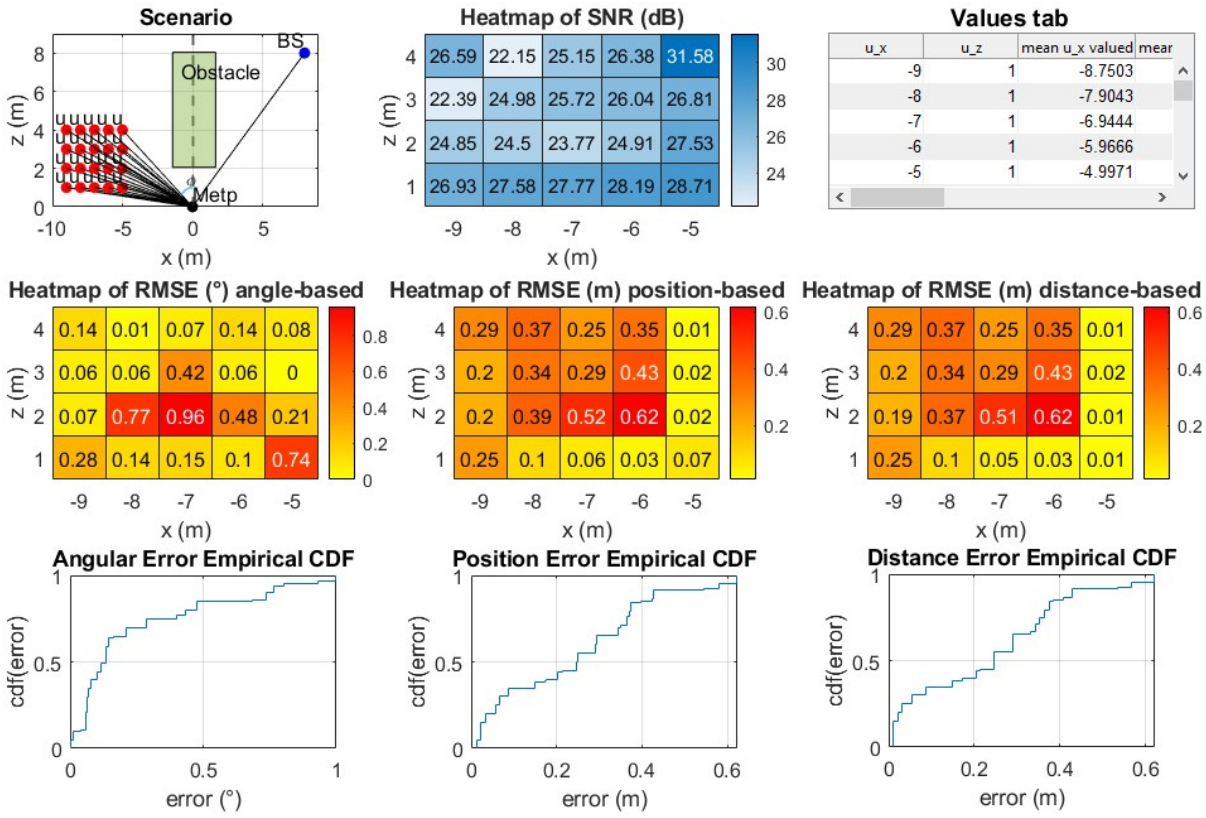


Figure 4.67: Estimator performance for a test grid step of 0.01 m

From these results it can be seen that, as happened in the case of beamsteering design, with a smaller step, i.e. with a higher resolution of the test grid, the performance of the estimator improves. In fact, by testing a greater number of user's positions, the estimator is more likely to find the real one or in any case positions closer to it.

The Tab. 4.8 shows how, the estimation errors of angle, distance to metaprism and position relative to about 80% of user's positions (CDF value = 0.8), decrease as the test grid step decreases. It is interesting to note that by increasing the resolution of the grid by 5 times (i.e. decreasing the step from 0.5 m to 0.1 m), the estimation errors are halved.

| Test grid step | Angle estimation error | Position estimation error | Distance estimation error |
|---|---|---|---|
| 0.5 m | 3.5° | 1.2 m | 1.2 m |
| 0.1 m | 0.75° | 0.61 m | 0.61 m |
| 0.05 m | 0.61° | 0.56 m | 0.56 m |
| 0.01 m | 0.43° | 0.37 m | 0.37 m |

Table 4.8: Estimation errors on angle, position and distance from metaprism relative to 80% of user's positions (CDF value = 0.8), for test grid step of 0.5 m, 0.1 m, 0.05 m, 0.01 m

The Fig. 4.68 reports the percentages of user's positions relative to a maximum angle estimation error of 0.5°, and a maximum estimation error of the position and distance from the metaprism of 0.5m, for a test grid step of 0.5 m, 0.1 m, 0.05 m and 0.01 m. Again, the percentage of user's positions estimated increases for a fixed estimation error, as the resolution of the test grid increases.

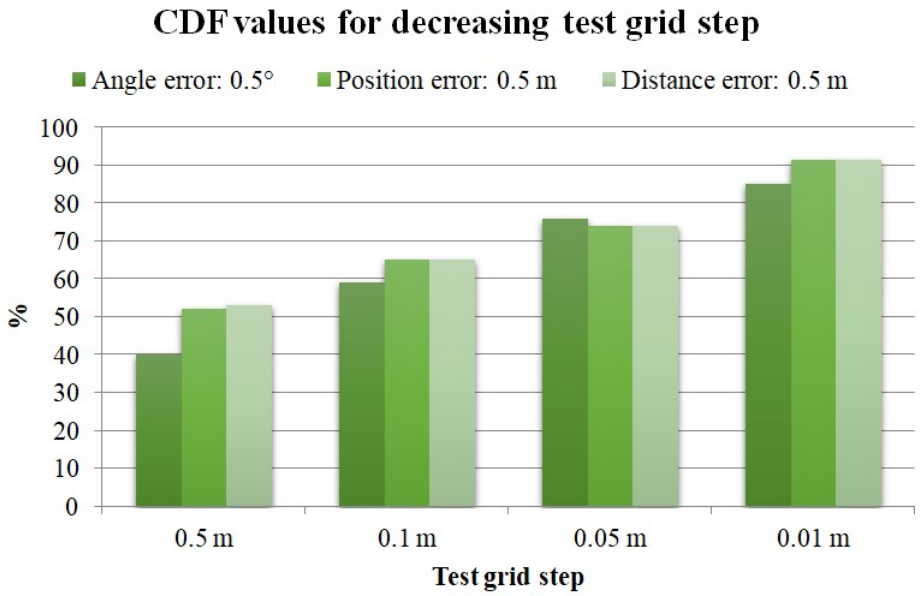


Figure 4.68: Empirical CDF percentage values related to an angle estimation error of 0.5° and to a position and distance estimation error of 0.5 m, for decreasing test grid step: 0.5 m, 0.1 m, 0.05 m, 0.01 m

### 4.3.4 Simulations results for the far-field scenario

The performance of the estimator is reported in the case of metaprism random design but considering a far-field scenario, obtained by moving the users away from the metaprism (as shown in Fig. 4.1).

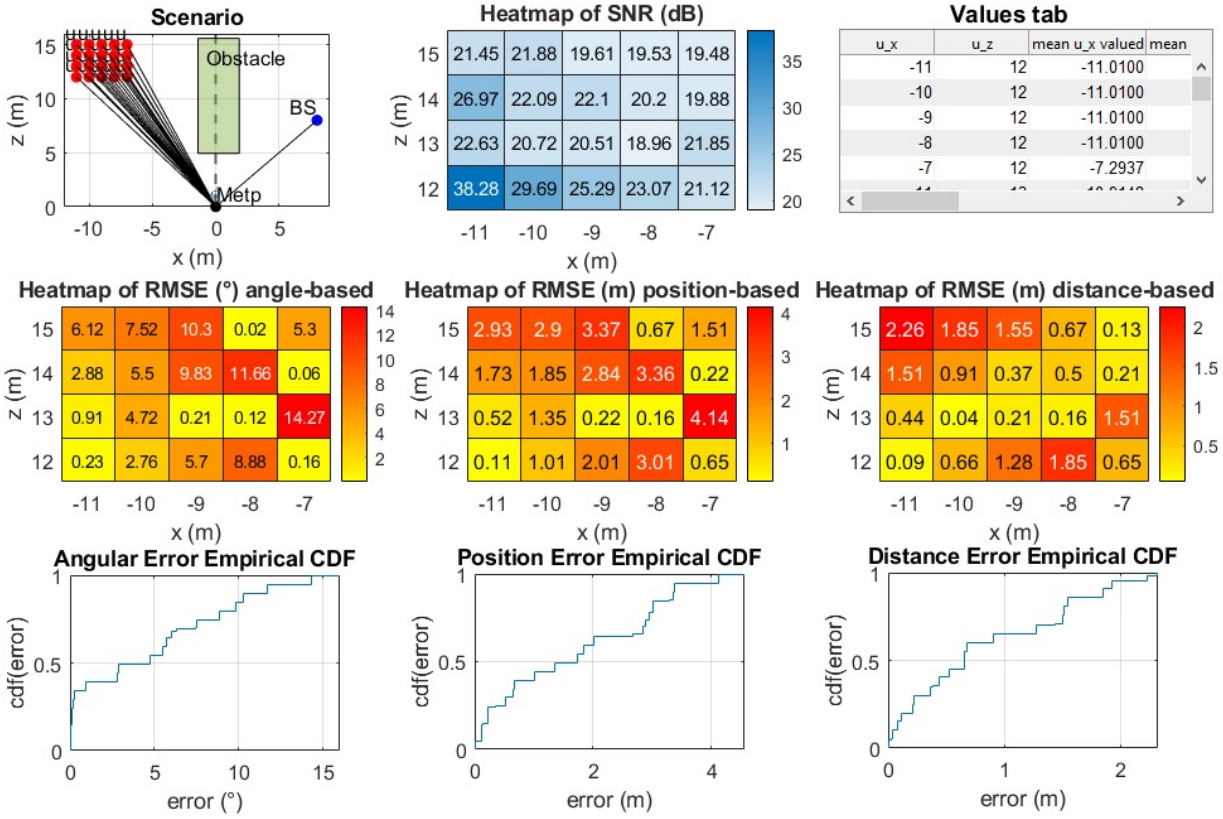


Figure 4.69: Estimator performance for metaprism with 100 cells per dimension and random design in the far-field scenario

From the Fig. 4.69, it is possible to notice how the resulting estimates are affected by even very large errors. For example, from the graph of the empirical CDF relating to the estimation of the user's position, it can be deduced that less than 20% of the positions are estimated with an error of less than 0.5 m, while the rest are estimated with errors of even a few meters. Again, if we compare this empirical CDF with the one obtained in the near-field scenario for the same parameter values, we note that 80% of user's positions (CDF value = 0.8) in this case is estimated with a maximum error of 3 m, against about 0.6 m for near-field scenario. The cause of this very high error is due to the fact that the further the user moves away from the near-field of the metaprism, the weaker the distance information is, and consequently the more difficult it is for the

estimator to correctly estimate the user's positions. The approach based on the random design in fact relies on the uniqueness or almost uniqueness of the frequency profiles for each different position of the user. The more the distance increases, the more users at different positions but at about the same angles with respect to the metaprism, will generate a very similar signature which will tend to confuse the estimator, who will struggle to discriminate them.

To confirm this, the profiles of the signal $\underline{s}$ are shown in Fig. 4.70 as a function of the number of sub-carriers, for each different position of the user. In particular, the profiles relating to positions, for which the estimator outputs the same estimated position, are highlighted with the same colour. For example, for the true positions relating to the five profiles in yellow, the estimator supplies as an estimate the position (-11.01, 0, 11.9867) for all of them (as shown in the table of Fig. 4.71). This happens because the profiles are similar enough to each other, such that, following the addition of the channel noise, the profiles received from these five positions can be traced back, for the estimator, to the test profile relating to the single position (-11.01, 0, 11.9867). In fact, these user's positions are relative to very similar angles, all close to about 30°. The same observations can be made with the profiles highlighted in green and blue.

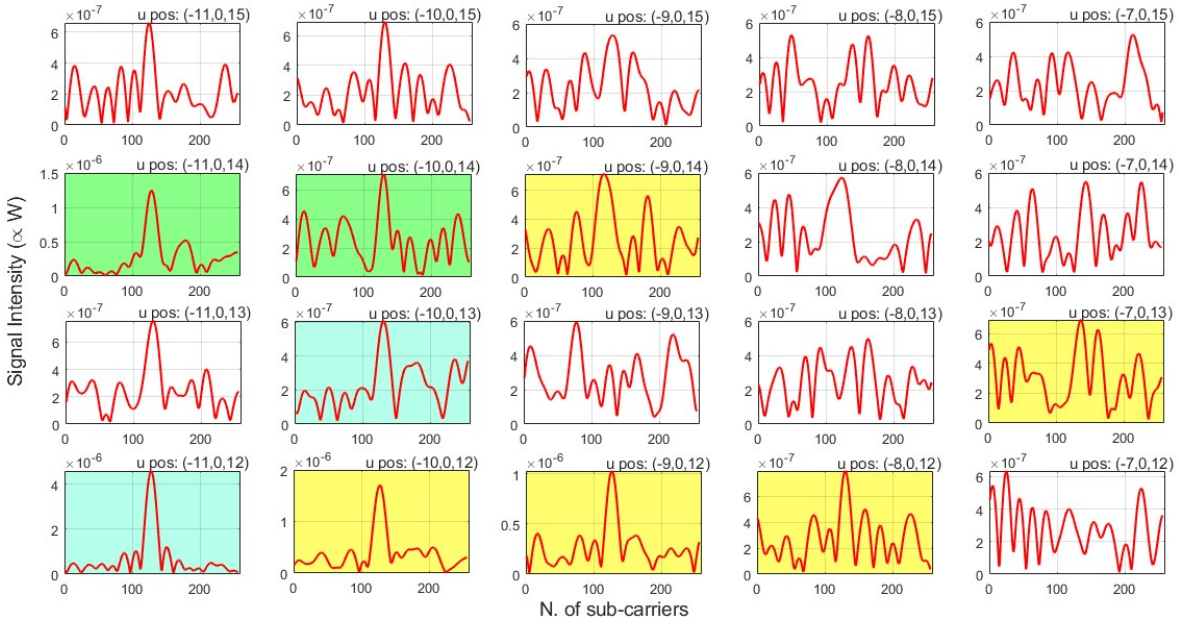


Figure 4.70: Signal $\underline{s}$ profiles, as a function of the number of sub-carrier, for all user's true positions, obtained with a metaprism of 100 cells per dimension; the profiles for which the estimator outputs the same estimated position are highlighted with the same color

| u_x | u_z | mean u_x valued | mean u_z valued | true distance from Metp (m) | mean valued distance from Metp (m) | true phi(°) | mean phi valued(°) |
|---|---|---|---|---|---|---|---|
| -11 | 12 | -11.0100 | 12.1066 | 16.2788 | 16.3643 | -42.5104 | -42.2842 |
| -10 | 12 | -11.0100 | 11.9867 | 15.6205 | 16.2758 | -39.8056 | -42.5682 |
| -9 | 12 | -11.0100 | 11.9867 | 15 | 16.2758 | -36.8699 | -42.5682 |
| -8 | 12 | -11.0100 | 11.9867 | 14.4222 | 16.2758 | -33.6901 | -42.5682 |
| -7 | 12 | -7.2937 | 12.5854 | 13.8924 | 14.5462 | -30.2564 | -30.0937 |
| -11 | 13 | -10.9148 | 12.4900 | 17.0294 | 16.5871 | -40.2364 | -41.1496 |
| -10 | 13 | -11.0100 | 12.1066 | 16.4012 | 16.3643 | -37.5686 | -42.2842 |
| -9 | 13 | -8.9250 | 12.8051 | 15.8114 | 15.6085 | -34.6952 | -34.8761 |
| -8 | 13 | -7.9013 | 12.8947 | 15.2643 | 15.1229 | -31.6075 | -31.4980 |
| -7 | 13 | -11.0100 | 11.9867 | 14.7648 | 16.2758 | -28.3008 | -42.5682 |
| -11 | 14 | -10.6973 | 12.2917 | 17.8045 | 16.2948 | -38.1572 | -41.0327 |
| -10 | 14 | -10.6973 | 12.2917 | 17.2047 | 16.2948 | -35.5377 | -41.0327 |
| -9 | 14 | -11.0100 | 11.9867 | 16.6433 | 16.2758 | -32.7352 | -42.5682 |
| -8 | 14 | -10.9788 | 12.4516 | 16.1245 | 16.6005 | -29.7449 | -41.4030 |
| -7 | 14 | -7.1110 | 14.1845 | 15.6525 | 15.8671 | -26.5651 | -26.6258 |
| -11 | 15 | -11.0100 | 12.0706 | 18.6011 | 16.3377 | -36.2538 | -42.3691 |
| -10 | 15 | -10.6679 | 12.1833 | 18.0278 | 16.1937 | -33.6901 | -41.2062 |
| -9 | 15 | -10.5156 | 11.9867 | 17.4929 | 15.9455 | -30.9638 | -41.2597 |
| -8 | 15 | -7.6874 | 14.4020 | 17 | 16.3253 | -28.0725 | -28.0922 |
| -7 | 15 | -7.3552 | 14.7242 | 16.5529 | 16.4590 | -25.0169 | -26.5435 |

Figure 4.71: Table of true and estimated values relative to Fig. 4.69; the user's positions highlighted, correspond to the profiles highlighted in Fig. 4.70

The weak uniqueness of the profiles is therefore the cause of the poor performance of the estimator in the far-field scenario.

### 4.3.5 Simulations results for BS in the metaprism near-field

Another way to try to improve the performance of the estimator is to move the receiver closer to the metaprism, in order to increase the quality of the wireless link through a higher SNR. The results of the simulations, considering the BS in the near-field of the metaprism, are presented below. The fact of having set the gain of the receiving antenna equal to 6 dB, allows us to neglect its radiation diagram, which instead for directional antennas must be considered, especially in the near-field region. This gain value ensures reduced directivity and ensures that the main lobe of the receiving antenna is able to 'illuminate' the entire metaprism, justifying the constant antenna gain approximation with respect to the metaprism observation angle.

When we bring the BS closer to the metaprism, if we maintain the same obstacle shown in the previous scenarios, it originates the Line-of-Sight (LoS) direct path between the base station and the user, more or less in every evaluated position. We could consider modifying the obstacle so that user and BS remain in an NLoS situation, but that would be a too remote scenario. Consequently we consider a LoS scenario, like the one shown in Fig. 4.72.

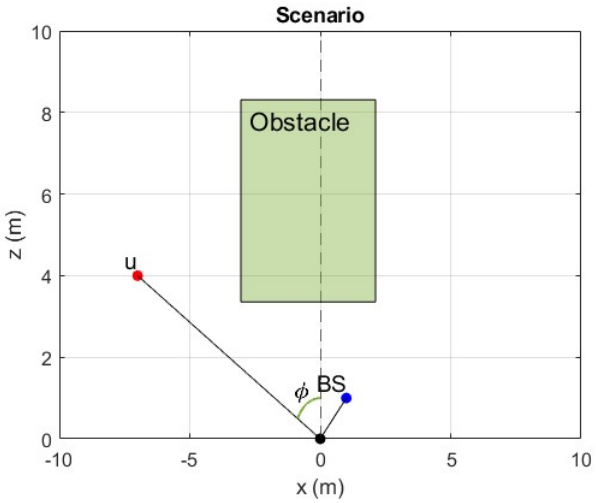


Figure 4.72: Example of scenario LoS with BS in the near-field of the metaprism

Now, in a LoS situation, the doubt about the usefulness of the metaprism could arise. In reality, it is not enough for the user and the BS to see each other directly, but for the base station to be able to discriminate the user's position, an ad hoc positioning system would be needed, such as an on-board antenna array that acts as a scanner for signal search. Since the metaprism has essentially the same function, it is convenient to introduce it into the communication system, despite the LoS scenario, and use a transmission linked to any standard rather than modifying the base station antenna or complicating it with expensive hardware.

The performance of the estimator in the case of BS gradually moved closer towards the metaprism is reported below, for a metaprism of 100 cells per dimension and a test grid step of 0.1 m.

BS coordinates: [3, 3] equivalent to a distance of 4.24 m from the metaprism

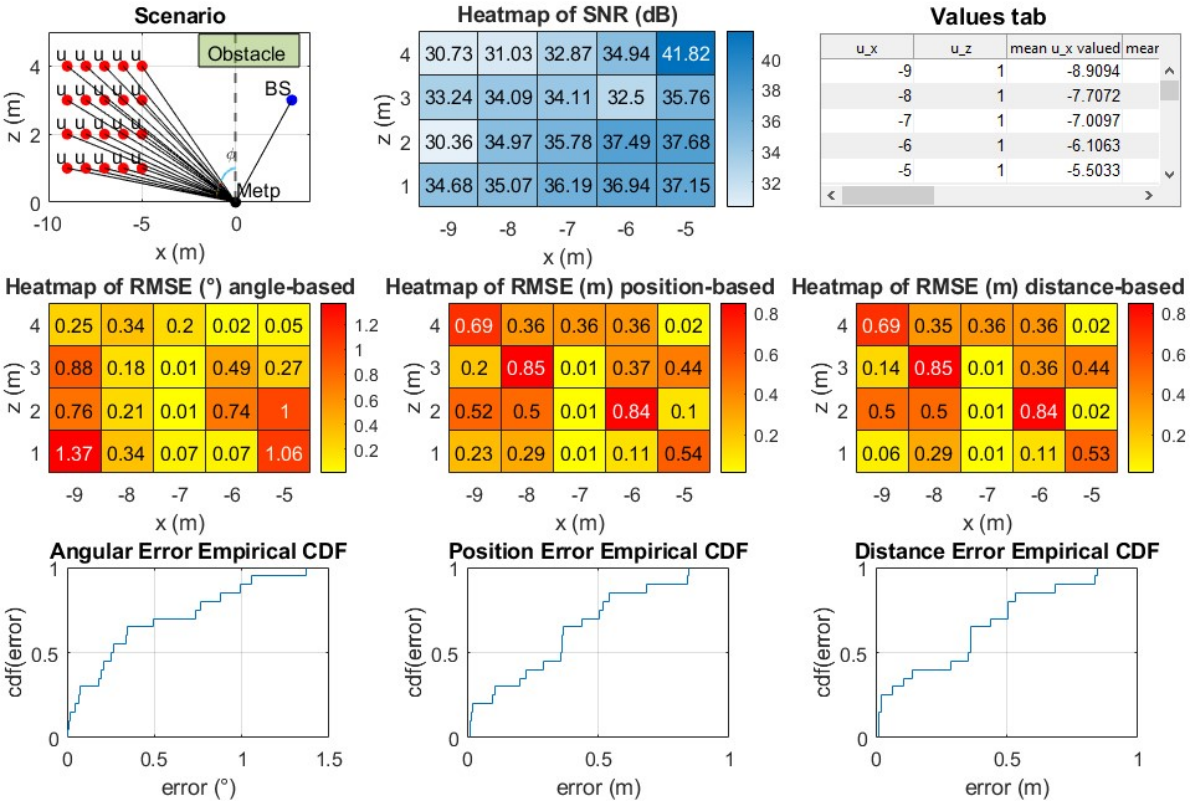


Figure 4.73: Estimator performance for 100 × 100 cells metaprism and BS coordinates: [3, 3]

BS coordinates: [1, 1] equivalent to a distance of 1.41 m from the metaprism

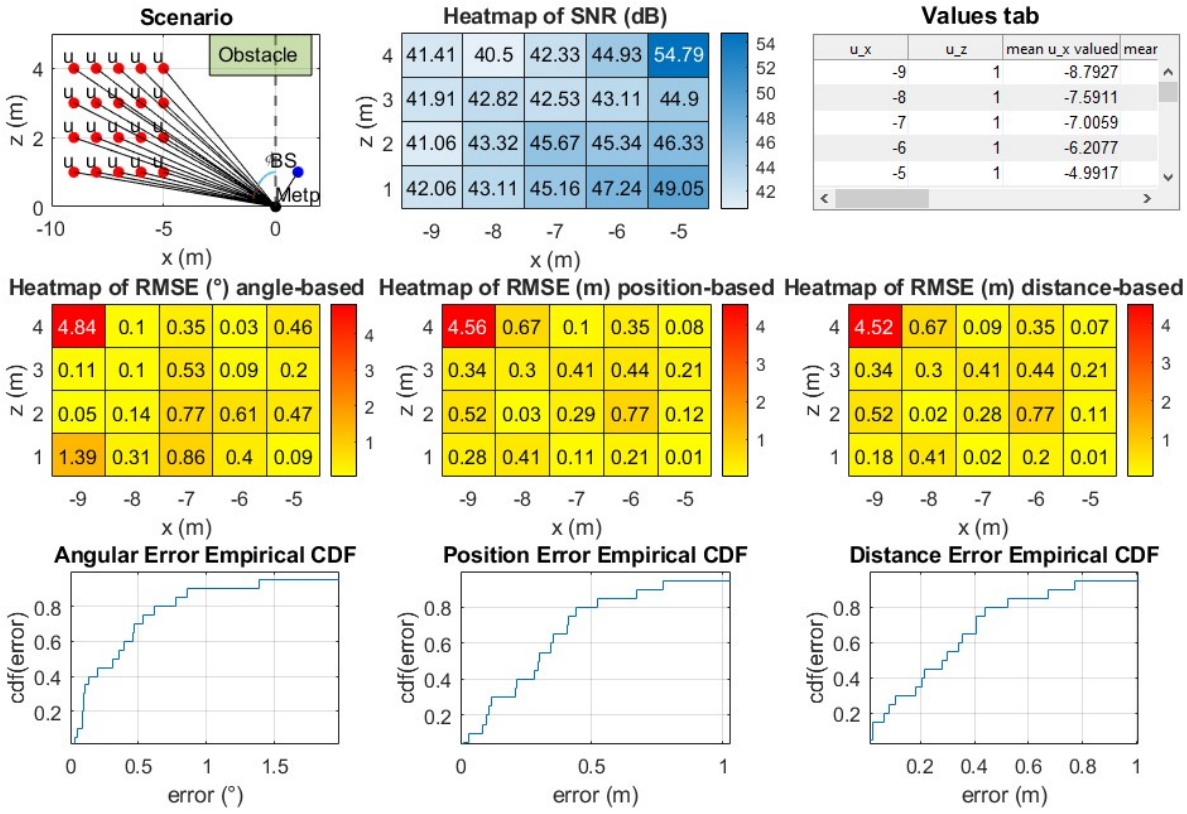


Figure 4.74: Estimator performance for 100 × 100 cells metaprism and BS coordinates: [1, 1]

From the figures presented it is possible to note that the performance of the estimator generally improves as the distance of the base station from the metaprism decreases; in fact in this way, under equal conditions, the SNR received by the BS increases. The Fig. 4.75 reports the percentages of user's positions relative to a maximum angle estimation error of 0.5°, and a maximum estimation error of the position and distance from the metaprism of 0.5 m, for a decreasing BS distance from metaprism.

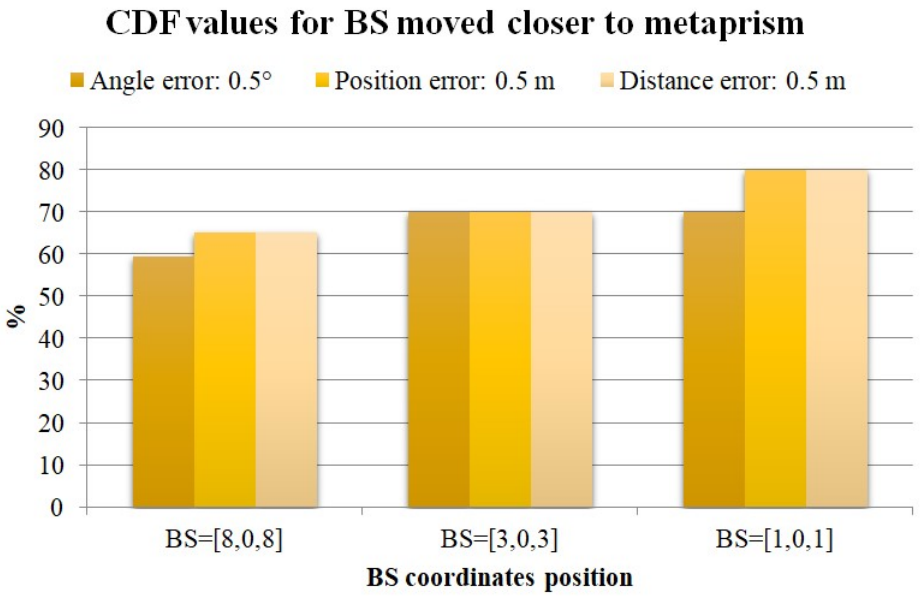


Figure 4.75: Empirical CDF percentage values related to an angle estimation error of 0.5° and to a position and distance estimation error of 0.5 m, for decreasing BS distance from metaprism: 11.31 m, 4.24 m, 1.41 m

The Tab. 4.9 shows how, the estimation errors of angle, distance to metaprism and position relative to about 80% of user's positions (CDF value = 0.8), decrease as the BS distance from metaprism decreases.

| **BS distance from metaprism** | **Angle estimation error** | **Position estimation error** | **Distance estimation error** |
|---|---|---|---|
| 11.31 m | 0.75° | 0.61 m | 0.61 m |
| 4.24 m | 0.76° | 0.52 m | 0.5 m |
| 1.41 m | 0.61° | 0.44 m | 0.44 m |

Table 4.9: Estimation errors on angle, position and distance from metaprism relative to 80% of user's positions (CDF value = 0.8), for decreasing BS distance from metaprism

It is interesting to note that moving the BS about 10 meters closer towards the metaprism, compared to the previously fixed distance (11.31 m), provides an improvement of about 20 cm in the estimate of the user's position.

BS coordinates: [1, 1] and far-field user's positions

We can also investigate how the performance of the estimator changes, by continuing to use a BS in the near-field of the metaprism but considering far-field positions of the user. Below are shown the results of a simulation under these conditions, for a metaprism of 100 cells per dimension and with reference to Sec. 4.2 for the values of the other parameters.

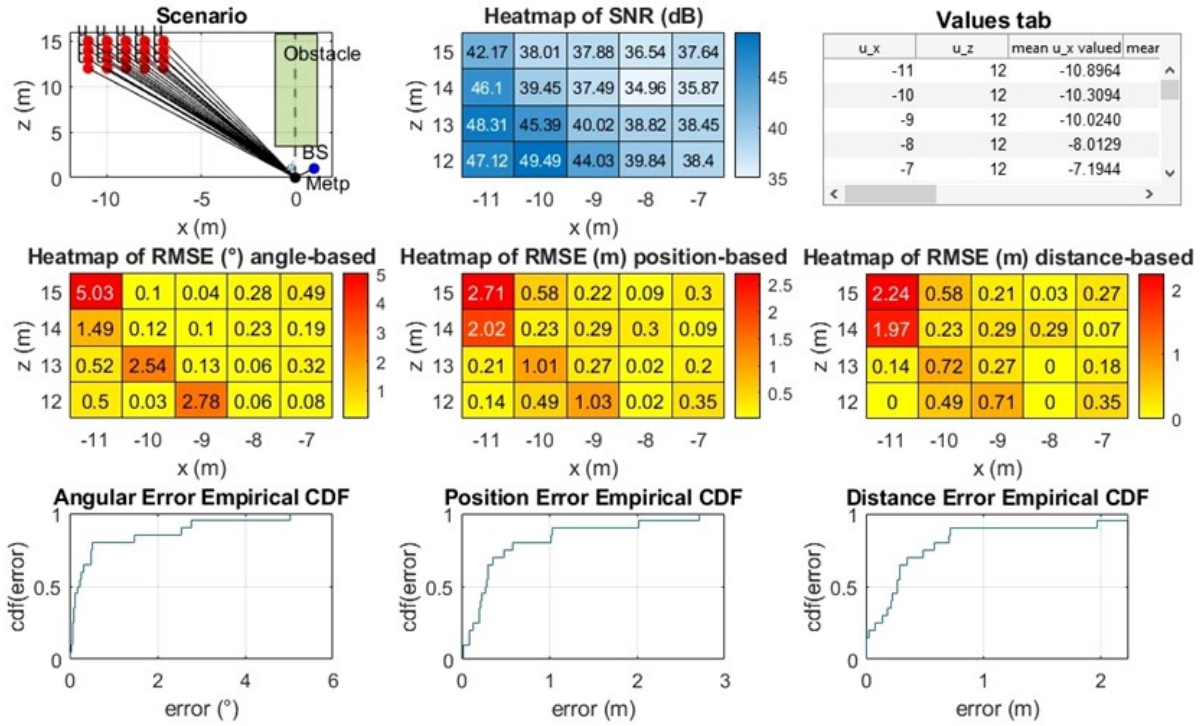


Figure 4.76: Estimator performance for 100× 100 cells metaprism, BS coordinates: [1,1] and far-field user's positions

The Fig. 4.77 reports the difference between the percentages of user's positions relative to a maximum angle estimation error of 0.5° and a maximum estimation error of the position and distance from the metaprism of 0.5 m, for user's positions in near-field and in the far-field one, for the BS in the near-field of the metaprism.

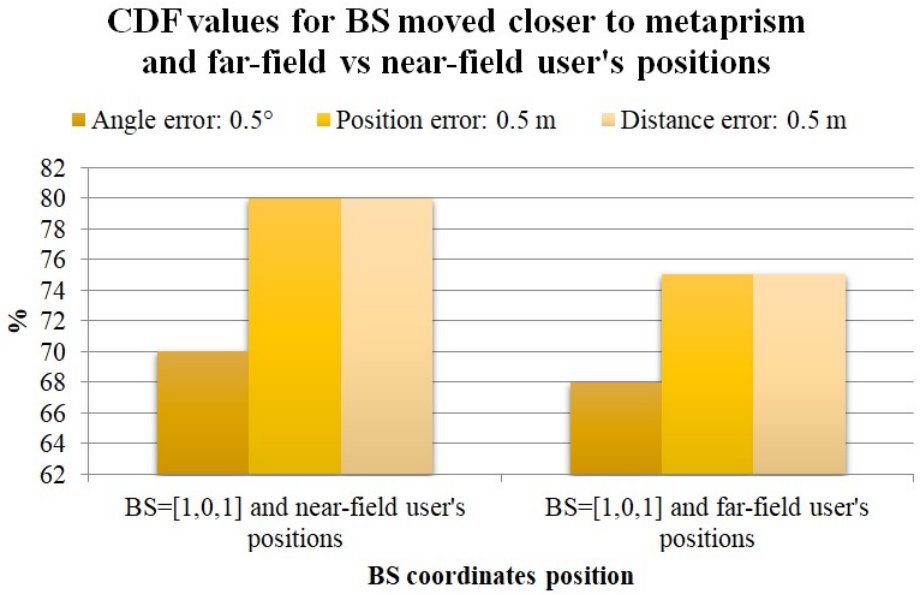


Figure 4.77: Empirical CDF percentage values related to an angle estimation error of 0.5° and to a position and distance estimation error of 0.5 m, for BS distance from metaprism: 1.41 m and user's positions in near-field and far-field

The Tab. 4.10 shows, the estimation errors of angle, distance to metaprism and position relative to about 80% of user's positions (CDF value = 0.8) in the near-field and in the far-field of the metaprism.

| **BS distance from metaprism: 1.41 m** | **Angle estimation error** | **Position estimation error** | **Distance estimation error** |
|---|---|---|---|
| Near-field users | 0.61° | 0.44 m | 0.44 m |
| Far-field users | 0.52° | 0.58 m | 0.58 m |

Table 4.10: Estimation errors on angle, position and distance from metaprism relative to 80% of user's positions (CDF value = 0.8), in the near-field and in the far-field of the metaprism

From the previous results it is possible to deduce that the performance of the estimator is good even if we consider BS in near-field and user's positions in far-field. Clearly the performance is not better than the case in which also the user's positions are in near-field, given that the information on the distance is stronger the closer the user is to the metaprism (Fig. 4.77 and Tab. 4.10). In fact, the most critical positions are those furthest from the metaprism, with an estimation error of more than one metre, due not

so much to noise, given that the SNR values are above 30 dB for all the positions evaluated, but to the fact that problably, with these parameter values, the estimator starts to struggle to estimate for such distances. However, the fact that the estimation error remains relatively low for most of the positions evaluated, allows us to state that if we manage to place the BS in a strong near field, i.e. very close to the metaprism, then it is not necessary to have users excessively close to the metaprism but also slightly further away because in any case the estimator manages to locate them quite well. This result may be relevant in situations where it is not possible to install the metaprism near the users, but exclusively near the BS.

# Conclusions

In new generation wireless networks, the desire to control the propagation environment has given rise to SREs based on RISs, in order to optimize the communication efficiency and performance. RISs have many advantages including low cost, low power consumption and easy implementation, but also some disadvantages, regarding CSI estimation and extra bandwidth and power required for reconfigurability. An alternative that allows to overcome these limits is represented by the metaprism, a particular frequency-selective metasurface.

The main purpose of this thesis was to investigate whether metaprism could be effective also to address localization problems, especially when operating in NLoS conditions. Under this perspective, a user-BS wireless link, based on OFDM signaling, under NLoS conditions due to an obstacle and improved by the introduction of the metaprism was studied. In particular, a generic user in an unknown position has been considered as a transmitter, and through the study of the reflection provided by the metaprism and the signal received by the BS, a localization application has been devised to deduce the angle of arrival and also track the position of the user. To do this, a ML estimator was designed and implemented in Matlab, based on a fingerprinting approach relative to the profiles of the signal components received across the sub-carriers.

In this work, different scenarios have been simulated for two different methods of metaprism design, in particular of the coefficients that identify its phase profile: beamsteering design and random design. The first method foresees that the value of the coefficients is linearly dependent on the position of the cells of the metaprism in order to perform the sub-carrier dependent beamsteering, whereas the second foresees that the values are set in a completely random way in order to better exploit the phase profile diversity offered by the channel when working in near-field conditions. Each of the two metaprism design methods was applied within two types of generated scenarios: one for users deployed within a certain area of interest in the far-field of the metaprism and one for deployed in the near-field, maintaining the BS in the far-field.

For each simulation, the performance of the ML estimator was reported and discussed, in relation to the estimation of the user's position, the estimation of the distance from the metaprism and of the AOA with respect to the normal of the metaprism, for different values of the size of the metaprism, number of sub-carriers and number of positions tested

by the estimator. This performance has been expressed in terms of RMSE in the form of a heatmap, which shows the entity of the estimation error for each user's position considered and in terms of empirical CDF, from which it is intuible the percentage of estimable positions with a certain maximum estimation error.

From the simulations carried out for the metaprism beamsteering design case, it can be observed that the estimate of the user's position and the relative distance from the metaprism are affected by considerable errors, even greater than 2 meters. This shows that in the far-field region, positions lying along the same direction give very similar sub-carrier profiles, which, due to noise, cause the ML estimator to get confused in its decision. It therefore happens that the estimates of the user's position and of the distance from the metaprism are random, while that of the AOA is precise. Consequently, for far-field user's positions, we can only obtain accurate angle estimations. In particular, considering a metaprism with 50 cells per dimension, i.e. $0.27m \times 0.27m$, it is possible to state that 90% of the arrival angles, relating to the user's positions evaluated, are estimated with a maximum error of 0.33°. The results of the simulations showed that this estimation error tends to further decrease as the size of the metaprism increases, i.e. as the number of cells increases for a fixed size. For example going from a metaprism with 10 cells to one with 100 cells per dimension (0.54m x 0.54m), the AOA estimation error decreases by more than a factor of 10 or equivalently, the percentage of relative user's positions at an angle estimate with a maximum error of 0.2°, it goes from less than 10% to about 80%. This is due to the fact that for a given position, the value of the SNR received by the base station increases as the number of cells increases. In fact, a larger metaprism will be able to capture a greater portion of the incident signal power. Between all the positions, the most critical ones in terms of AOA estimation are those at the extreme of the main lobe range of the metaprism equivalent array factor (-85° and -25° in our case) and those relating to angles corresponding to a null of the equivalent array factor (AF). Regarding this, it has been observed that as the number of signal sub-carriers increases, the performance of the estimator in terms of AOA improves in general. In fact, considering a greater number of sub-carriers, in addition to increasing the SNR received from the BS, the null points of the equivalent AF are reduced and consequently the probability that a user transmits at one of these is reduced. We can also state that by increasing the number of test positions screened by the estimator during the user localization operation, i.e. by increasing the resolution of the ML test grid used, the estimation performance improves. In fact, by testing a greater number of user's positions, it is more probable that the estimator will find the real one or in any case positions closer to it. For example, by decreasing the ML test grid step from 0.5 m to 0.01 m, the AOA estimation error decreases from 0.9° to 0.2°. All these results were obtained from the empirical CDF relating to the AOA estimate, that is more significant than the average value of the error represented by the RMSE, which could be altered by outliers, and therefore could lead to erroneous conclusions. As regards the performance of the estimator relating to user's positions in the near-field of the metaprism under

the hypothesis of beamsteering design, the increase in the estimation error of AOA is evident, which goes from 0.33° for the far-field at over 2.6° for the near-field one. This verifies that the use of metaprism, designed for beamsteering, in the near-field scenario is not optimal.

From the results obtained from a similar analysis conducted for the case of random design of the metaprism, it can be stated that this method, for user's positions in the near-field, offers the possibility of estimating the user's position, unlike the beamsteering design. This is due to the fact that the random setting of the metaprism coefficients ensures that the profile of the sub-carriers received is unique (or nearly so) for each user's position, thus becoming a sort of signature of each position. Under the random design hypothesis, however, the gain and consequently the overall link budget worsen, given that we can no longer speak of the gain of the entire array (being all random). For this reason, this analysis was conducted for slightly increased metaprism dimensions and transmission powers (from 10 to 15 dBm), in order to obtain higher SNR values. For the same reasons described in the case of beamsteering design, the performance of the estimator improves as the size of the metaprism increases. For example, for a $200 \times 200$ cell ($1.08m \times 1.08m$) metaprism, the average RMSE in localization across all user's positions is only 29 cm, which is a good result considering that the users are behind the obstacle, i.e., in NLOS with respect to the BS and only one BS is used. We can further state that the number of user's positions estimated with a maximum error of 0.5 m ranges from 45% for a metaprism with 50 cells per dimension, to 65% for a metaprism with 100 cells per dimension and to 85% for a metaprism with 200 cells per dimension. So, with an estimation error of 0.5 m, increasing the number of metaprism cells from 50 to 200 per dimension, 40% more user's positions can be estimated. Moreover, the performance of the estimator improves as the number of sub-carriers increases. In fact, if for example we consider the estimate of the user's position, going from 64 to 1024 sub-carriers we obtain an improvement in the estimate of about 10 cm, reducing the error from 0.67 m to 0.56 m. Furthermore, performance also improves for more test positions examined by the ML estimator. As regards the performance of the estimator relating to user's positions in the far-field of the metaprism under the hypothesis of random design, it can be deduced that less than 20% of the positions are estimated with an error of less than 0.5 m, while the remaining are estimated with much larger errors up to a few meters. The cause of these very large errors is due to the fact that the further the user moves away from the near-field of the metaprism, the weaker the distance information is embedded in the EM field wavefront (which tends to become plane and hence informative only for AOA estimation), and consequently the more difficult it is for the estimator to correctly estimate its positions, especially those along similar directions, due to the poor uniqueness of the received sub-carrier profiles.

Finally, the performance of the estimator was analysed, in the hypothesis of random design of the metaprism, for a base station close to the metaprism of about 1.5 m, i.e. in its near-field. After the simulations, for user's positions in the near-field, it can be stated

that the performance of the estimator generally improves as the distance of the base station from the metaprism decreases, thanks to the increase of the SNR received from the BS. Interestingly, moving the BS about 10 meters towards the metaprism results in an improvement of about 20 cm in estimating the user's position. From the same simulation, performed instead for user's positions in the far-field of the metaprism, we can state that the performance is good in this case too, although not better than for user's positions in the near-field since for greater distances the estimator struggles more to locate. This result may be relevant in situations where it is not possible to install the metaprism near the user, but only near the BS.

The results of the simulations conducted using the ML estimator obtained show very good performance in estimating the user's position in general. In particular for some positions, under certain conditions, it is possible to locate the user behind the obstacle, with less than 1 cm of error. This is a remarkable result, considering that it can be obtained with only one base station (typically, localization schemes need at least 3 base station to triangulate) without the need to use active devices but exclusively thanks to the presence of a passive metasurface, i.e. at practically zero cost.

In this work both the user and base station have been considered with a single antenna. A potential future challenge could be to analyze the necessary changes for multi-antenna users and BS, even though the optimization of the antennas configuration of both nodes could be very complex. Other open research problems, fundamental to reveal the benefits and limits of metasurfaces, are: the search for ever more efficient implementations, the quantification of the performance in large-scale distribution (in a city center for example), the adoption of more realistic models for the metaprism accouting for non-idealities, the study of machine learning algorithms for system optimization and the integration with the main emerging technologies for future 6G wireless networks.

# Bibliography


[1] Marco Di Renzo, Alessio Zappone, Merouane Debbah, Mohamed-Slim Alouini, Chau Yuen, Julien de Rosny and Sergei Tretyakov, "Smart Radio Environments Empowered by Reconfigurable Intelligent Surfaces: How It Works, State of Research, and The Road Ahead," *IEEE Journal on selected areas in communications*, vol. 38, no. 11, November 2020.

[2] Ertugrul Basar, Marco Di Renzo, Julien De Rosyn, Merouane Debbah, Mohamed-Slim Alouini, Rui Zhang, "Wireless Communications Through Reconfigurable Intelligent Surfaces," *Digital Object Identifier 10.1109/ACCESS.2019.2935192*, September 3, 2019.

[3] Yuanwei Liu, Xiao Liu, Xidong Mu, Tianwei Hou, Jiaqi Xu, Marco Di Renzo and Naofal Al-Dhahir, "Reconfigurable Intelligent Surfaces: Principles and Opportunities," *IEEE Communications surveys & tutorials*, vol. 23, no. 3, Third Quarter 2021.

[4] Hui-Hsin Hsiao, Cheng Hung Chu, Din Ping Tsai, "Fundamentals and Applications of Metasurfaces," *WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim*, 11 April 2017.

[5] Qammer H. Abbasi, Hasan T. Abbas, Akram Alomainy, Muhammad Ali Imran, "Backscattering and RF Sensing for Future Wireless Communication," *John Wiley & Sons Ltd*, 2021.

[6] Davide Dardari and Devis Massari, "Using MetaPrisms for Performance Improvement in Wireless Communications," *IEEE transactions on wireless communications*, vol. 20, no. 5, May 2021.

[7] D. Dardari, N. Decarli, A. Guerra and F. Guidi, "LOS/NLOS Near-Field Localization With a Large Reconfigurable Intelligent Surface," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4282-4294, June 2022.

# List of Figures

# List of Tables