



UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Laurea Triennale in Ingegneria Informatica

**Enhancing PHY-Layer Security in Communication
Systems Using Directional Modulation**

Relatore:

Prof. Luca Sanguinetti
Prof. Giacomo Bacci

Candidato:

Rossi Paccani Francesco

ANNO ACCADEMICO 2024/25

Abstract

Directional modulation (DM) is an emerging technique for enhancing physical-layer (PHY) security in wireless communication systems. The key idea of DM is to intentionally manipulate the amplitude and phase of the transmitted signal in a direction-dependent manner, such that legitimate users located along the desired transmission beams receive distortion-free symbols, whereas receivers in other spatial directions observe scrambled or unintelligible signals. This effect is achieved through precise antenna-array processing, beamforming, and, in some implementations, dynamic waveform perturbation.

Unlike traditional cryptographic approaches that operate at higher network layers, DM provides security directly at the PHY layer and does not require key distribution or complex encryption-decryption procedures. This makes it particularly attractive for latency-sensitive, low-power, or resource-constrained wireless systems. Moreover, DM can naturally complement existing security mechanisms by adding an additional spatial dimension of protection.

As wireless networks continue to evolve toward dense deployments, massive connectivity, and open broadcasting environments, they become increasingly susceptible to passive eavesdropping and sophisticated active attacks. In such settings, DM offers a promising solution, improving confidentiality by exploiting the inherent spatial characteristics of the propagation medium and the directional capabilities of modern antenna arrays.

Contents

1	Introduction	3
1.1	Scope and Contribution of this Thesis	4
1.2	Beamforming Theory	4
1.2.1	Constellation Analysis	6
1.3	Directional Modulation	6
1.3.1	Mathematical Model of Directional Modulation	7
2	Proposed Code Analysis	8
2.1	System Model	8
2.2	Problem Setup	8
2.3	Proposed Algorithm	9
2.4	Real phase shifter implementation	10
2.5	BER performance evaluation	16
2.5.1	BER Analysis for QPSK modulation	17
2.5.2	BER Analysis for 16-QAM Modulation	23
2.5.3	16-QAM vs QPSK in Directional Modulation	26
2.5.4	Effect of Weight Vector Selection Margins on BER in Directional Modulation	28
3	Conclusions	30
3.1	Summary of Results	30
3.2	Final Considerations	31
3.3	Future Work	31

1 Introduction

In wireless communication systems, transmitters and receivers share a common channel for information exchange. Consequently, any transmitted message is inherently broadcast over the medium [1], and without appropriate protection mechanisms, any receiver within coverage can potentially intercept and decode the signal. Traditionally, this is addressed through cryptographic techniques implemented at higher layers of the protocol stack. While such methods ensure the confidentiality of the information even if an unauthorized receiver obtains the signal, they do not prevent interception itself. In other words, the eavesdropper still captures the waveform and may attempt brute-force decoding or other forms of cryptanalysis.

A fundamentally different approach to ensuring confidentiality relies on securing the communication directly at the physical layer, commonly referred to as PHY-layer security. The key principle is not to protect the information *after* reception, but to prevent unauthorized receivers from obtaining a meaningful version of the transmitted signal in the first place. In this framework, confidentiality arises from the inherent physical properties of the wireless propagation channel rather than from higher-layer encryption. Consequently, the signal is intelligible only within a designated spatial region, while outside this region it appears distorted, scrambled, or otherwise indecipherable.

Several techniques realize PHY-layer security by operating on different aspects of the transmitted waveform. Classical approaches include frequency-hopping spread spectrum (FHSS), time-hopping, and direct-sequence spread spectrum (DSSS), in which the temporal or spectral characteristics of the signal are deliberately randomized or spread to make interception more difficult for unintended receivers [2, 3]. While these methods improve robustness and provide an initial layer of PHY-layer security, they do not explicitly exploit the spatial properties of the propagation channel. For instance, in FHSS, if an eavesdropper successfully predicts the hopping sequence or temporarily tunes to the correct frequency, the transmitted signal becomes fully observable, as the information content is radiated identically in all directions.

More recently, spatial-domain security techniques have emerged, leveraging antenna arrays to shape the transmitted signal so that it becomes meaningful only along specific angular directions [4–6]. Among these methods, directional modulation (DM) embeds spatial dependence directly into the modulation process: constellation points are synthesized in a symbol-dependent manner such that only receivers located along the intended direction observe a valid constellation, while receivers outside this region receive scrambled, rotated, or otherwise distorted symbols that cannot be correctly demodulated [7, 8].

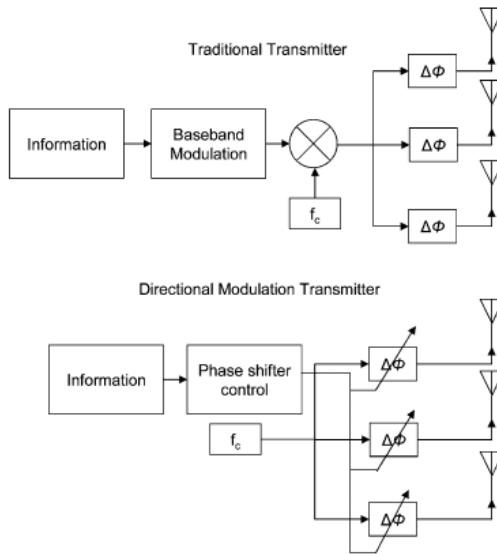


Figure 1: Traditional array transmitter (top) and one implementation of a DM transmitter using phase shifters [7].

The conceptual difference between a traditional phased-array transmitter and a directional modulation transmitter can be visualized in Fig. 1, obtained from [7]. In a traditional phased-array system, baseband modulation is performed prior to RF up-conversion, and the resulting signal is identically applied to all antenna elements, differing only by fixed phase shifts to steer the beam. In contrast, a DM transmitter embeds the modulation directly into the spatial radiation pattern: the phase (and, in some implementations, amplitude) of each antenna element is dynamically controlled on a per-symbol basis, ensuring that a valid constellation is formed only along the intended angular direction.

This capability distinguishes DM from conventional phased-array transmission. In a traditional phased-array system, the radiated energy can be focused toward the legitimate receiver; however, the same symbol content is also transmitted through sidelobes and can be intercepted by an eavesdropper equipped with a sufficiently sensitive receiver [5, 6]. In contrast, DM modifies the modulation on a symbol-by-symbol basis, ensuring that each symbol forms a valid constellation only along the intended direction, while an unauthorized receiver observes a distorted or scrambled signal [7, 8]. Early studies on time-modulated arrays demonstrated that switching schemes could reduce sidelobes [9–11], and subsequent work extended this concept to the direct synthesis of modulation via spatial processing [12–15].

DM is particularly relevant in scenarios requiring spatially constrained confidentiality, such as military communications, unmanned aerial vehicle (UAV) links, and satellite downlinks. In these applications, the signal directed toward the legitimate receiver must remain unintelligible to any other receiver, irrespective of its technological capabilities. While DM does not replace conventional encryption, it provides an additional layer of security that prevents meaningful interception. In highly sensitive contexts, this layered approach can substantially reduce the risk of passive eavesdropping.

1.1 Scope and Contribution of this Thesis.

The purpose of this thesis is to provide an in-depth study of directional modulation, with particular emphasis on its spatial selectivity and its potential as a physical-layer security technique. The study begins by introducing the mathematical framework describing array-based directional transmission and by outlining the conceptual differences between conventional phased-array beamforming and directional modulation.

Next, an algorithmic approach for synthesizing symbol-dependent weight vectors is presented and examined from both a structural and computational perspective. The influence of practical hardware constraints, such as discrete phase quantization, is also analyzed in order to assess the feasibility of implementing directional modulation in realistic transmitter architectures.

Finally, performance is evaluated through bit error rate (BER) analysis in the intended direction and for off-axis receivers, under different modulation schemes, array configurations, and spatial misalignments. These observations highlight the inherent spatial filtering capability of directional modulation and offer insight into the trade-offs associated with practical implementation, motivating further investigation toward efficient and secure physical-layer communication systems.

To support reproducibility, a GitHub repository has been made publicly available, containing all MATLAB source code used in the analyses, along with the datasets generated for the numerical evaluations and graphical results. The repository can be accessed at:

<https://github.com/FrancescoRossiPaccani/DirectionalModulation-Thesis>

It provides fully reproducible implementations of the experiments presented in this work, including symbol synthesis procedures, array weight computations, and BER performance evaluations. This ensures methodological transparency and facilitates future reference, extension, or independent verification of the results.

1.2 Beamforming Theory

Beamforming is a signal processing technique used in sensor arrays for directional signal transmission or reception [16]. This is achieved by combining elements in an antenna array in such a way that signals at particular angles experience constructive interference while others experience destructive interference [17]. Beamforming can be used at both the transmitting and receiving ends in order to achieve spatial selectivity [18]. Consider an array of M antenna elements. A signal arriving from direction θ can be modeled as:

$$\mathbf{x}(t) = s(t) \mathbf{a}(\theta) + \mathbf{n}(t) \quad (1)$$

where:

- $s(t)$: desired signal,
- $\mathbf{a}(\theta)$: *steering vector* of the array,
- $\mathbf{n}(t)$: noise or interference,
- $\mathbf{x}(t) \in \mathbb{C}^M$: received signal vector.

f

For a uniform linear array (ULA) with inter-element spacing d , the m -th component of the steering vector is:

$$a_m(\theta) = e^{-j \frac{2\pi}{\lambda} (m-1)d \sin(\theta)}, \quad (2)$$

so that:

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ e^{-j \frac{2\pi}{\lambda} d \sin(\theta)} \\ e^{-j \frac{2\pi}{\lambda} 2d \sin(\theta)} \\ \vdots \\ e^{-j \frac{2\pi}{\lambda} (M-1)d \sin(\theta)} \end{bmatrix}. \quad (3)$$

Beamformer Output

The beamformer combines the received signals linearly with a weight vector \mathbf{w} :

$$y(t) = \mathbf{w}^H \mathbf{x}(t), \quad (4)$$

where \mathbf{w}^H denotes the Hermitian transpose.

Beam Pattern

The directional response of the beamformer is:

$$B(\theta) = \mathbf{w}^H \mathbf{a}(\theta). \quad (5)$$

Choice of Weights

Different strategies exist for selecting the weight vector \mathbf{w} :

- **Conventional (delay-and-sum) beamforming:**

$$\mathbf{w} = \frac{1}{M} \mathbf{a}(\theta_0), \quad (6)$$

where θ_0 is the desired user direction.

- **MVDR (Minimum Variance Distortionless Response) beamforming:**

$$\mathbf{w}_{MVDR} = \frac{\mathbf{R}^{-1} \mathbf{a}(\theta_0)}{\mathbf{a}^H(\theta_0) \mathbf{R}^{-1} \mathbf{a}(\theta_0)}, \quad (7)$$

where $\mathbf{R} = \mathbb{E}[\mathbf{x}(t) \mathbf{x}^H(t)]$ is the covariance matrix.

1.2.1 Constellation Analysis

As an illustrative example, we consider the received QPSK constellation at both the intended user and a potential eavesdropper. The resulting constellations are shown in Fig. 2.

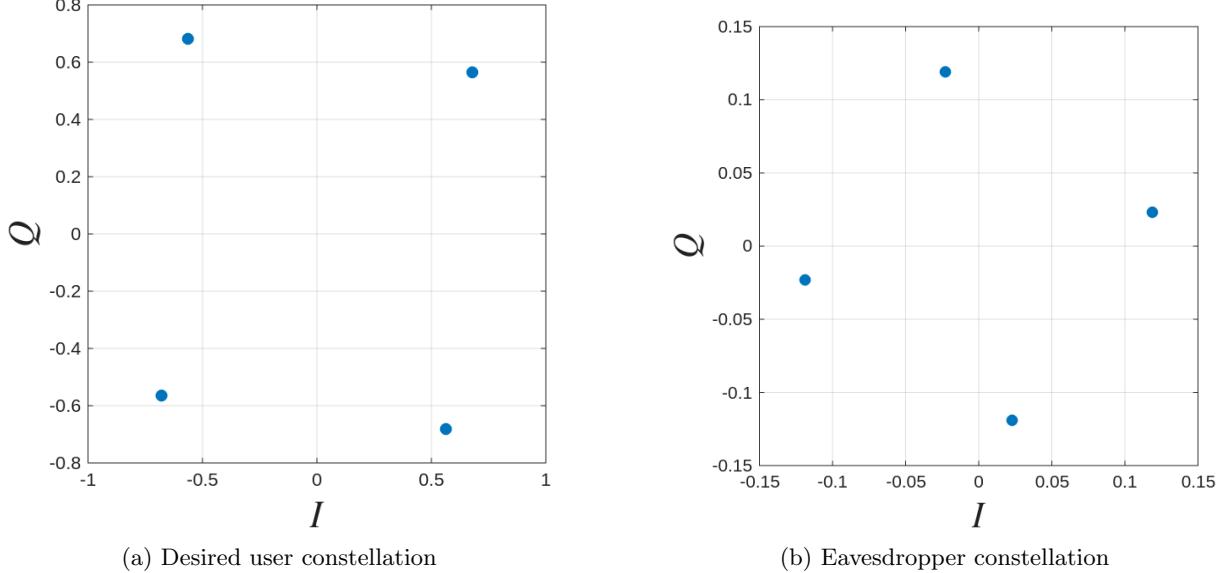


Figure 2: Comparison between the received constellations at the intended user and at a potential eavesdropper.

It can be observed that in Figure. 2:

- For the **desired user**, the received constellation maintains the correct QPSK structure.
- For the **eavesdropper**, the constellation has the same shape but appears rotated and scaled (either amplified or attenuated).

This implies that although the eavesdropper observes a QPSK-like constellation, the received symbols are distorted in both phase and amplitude. In the case of QPSK, where $M = 4$ symbols are transmitted, at most $M = 4$ decoding attempts are sufficient for the eavesdropper to correctly guess the transmitted symbol and all subsequent symbols. In the general case, when transmitting an M -ary constellation, the message can be decoded with at most M trials [19].

1.3 Directional Modulation

Building on the previous example, we now introduce the concept of *Directional Modulation* (DM). The key idea is that, instead of using a fixed steering vector for all transmitted symbols, a new steering vector is computed for each symbol based on beamforming theory.

As a result:

- For the **intended receiver**, the received symbols (apart from noise) accurately reconstruct the QPSK modulation, maintaining the correct constellation points.
- For the **eavesdropper**, the received symbols appear randomly scattered within the constellation diagram, with no meaningful QPSK structure. This makes symbol decoding particularly difficult or even infeasible.

Figure 3 illustrates this effect.

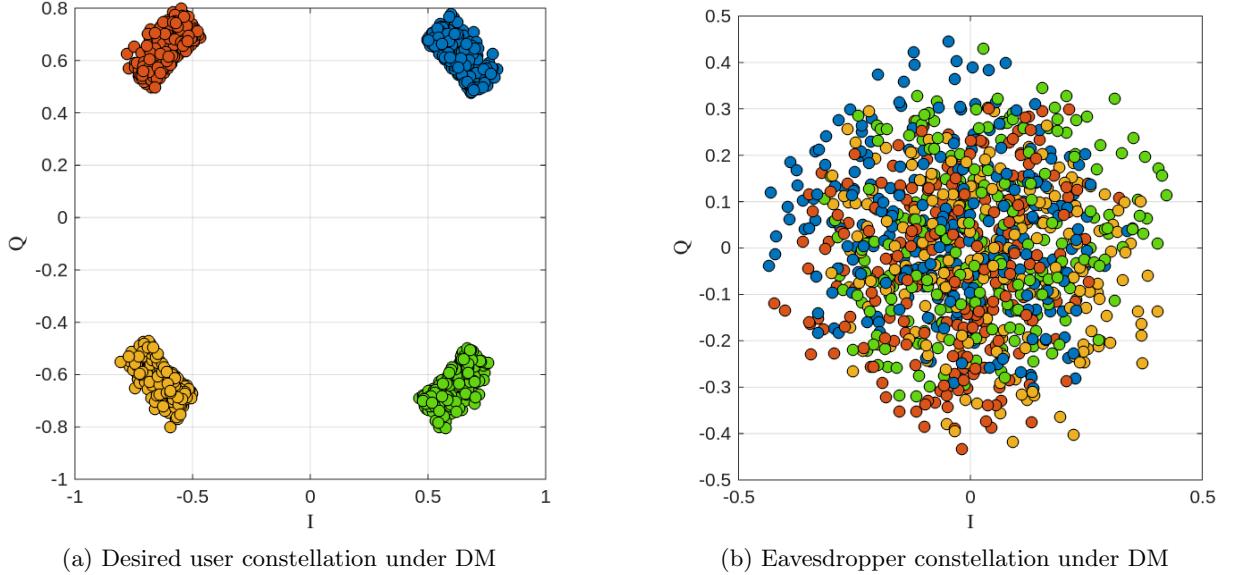


Figure 3: Constellations under directional modulation: the intended user recovers the QPSK structure, while the eavesdropper observes random symbol placement.

1.3.1 Mathematical Model of Directional Modulation

In conventional beamforming, a fixed weight vector \mathbf{w} is used for all transmitted symbols. [18, 20, 21] In contrast, in *Directional Modulation* the weight vector changes for each transmitted symbol s_k , i.e.:

$$\mathbf{w}_k = f(s_k, \theta_0), \quad (8)$$

where θ_0 denotes the direction of the intended receiver. The transmitted signal is then:

$$\mathbf{x}_k(t) = \mathbf{w}_k s_k, \quad (9)$$

and the received signal in direction θ is:

$$y_k(\theta) = \mathbf{a}^H(\theta) \mathbf{w}_k s_k + n(t). \quad (10)$$

By design:

- For $\theta = \theta_0$:

$$y_k(\theta_0) \approx s_k, \quad (11)$$

meaning that the desired receiver observes the correct constellation.

- For $\theta \neq \theta_0$:

$$y_k(\theta) \notin \text{constellation}, \quad (12)$$

and thus the eavesdropper observes distorted or random points, making symbol decoding particularly difficult or maybe impossible.

2 Proposed Code Analysis

This section provides a detailed examination of the MATLAB code developed to simulate and evaluate the performance of DM in wireless communication systems. The presentation is structured progressively, beginning with the theoretical implementation of DM assuming ideal continuous phase control, followed by the introduction of practical constraints due to phase quantization. Finally, the section concludes with a performance evaluation in terms of bit error rate (BER) under different modulation formats and varying antenna array sizes.

2.1 System Model

Consider a uniform linear array (ULA) situated on the yz -plane, consisting of N antennas separated uniformly by a distance d . The antenna positions are represented by Cartesian coordinates [17]:

$$\mathbf{u}_n = \begin{bmatrix} 0 \\ d \cdot n \\ 0 \end{bmatrix}, \quad n = 0, \dots, N - 1.$$

The steering vector based on phase shifts $\boldsymbol{\psi} = [\psi_0, \psi_1, \dots, \psi_{N-1}]^T$ applied at the antenna elements is expressed as

$$\mathbf{a}(\theta, \phi) = [\exp(j\psi_0), \exp(j\psi_1), \dots, \exp(j\psi_{N-1})]^T.$$

A wave impinging on the array with wavelength λ under azimuth ϕ and elevation θ is characterized by the channel vector

$$\mathbf{v}(\theta, \phi) = [\exp(j\mathbf{k}^T(\theta, \phi)\mathbf{u}_0), \dots, \exp(j\mathbf{k}^T(\theta, \phi)\mathbf{u}_{N-1})]^T,$$

where the wave vector is given as

$$\mathbf{k}(\theta, \phi) = \frac{2\pi}{\lambda} \begin{bmatrix} \cos \theta \cos \phi \\ \cos \theta \sin \phi \\ \sin \theta \end{bmatrix}.$$

The normalized array response is then calculated by

$$f(\theta, \phi) = \frac{1}{N} \mathbf{a}^H(\theta, \phi) \mathbf{v}(\theta, \phi) = \frac{1}{N} \sum_{n=0}^{N-1} \exp(j\mathbf{k}^T(\theta, \phi)\mathbf{u}_n - j\psi_n).$$

2.2 Problem Setup

The task is set as finding phase vectors $\boldsymbol{\psi}$ that satisfy magnitude and phase requirements within a target angular window

$$[\theta^* \pm \Delta\theta] \times [\phi^* \pm \Delta\phi],$$

for example $\Delta\theta = \Delta\phi = 1^\circ$, specifically:

- The amplitude constraint

$$\log_{10} \frac{|f(\theta, \phi)|}{|f(\theta^*, \phi^*)|} \geq \varepsilon_a,$$

where ε_a could be set to -3 dB.

- The phase constraint

$$|\angle f(\theta, \phi) - \angle f(\theta^*, \phi^*)| \leq \varepsilon_p,$$

for instance $\varepsilon_p = 5^\circ$.

Graphically, for $\theta = 0^\circ$, these constraints define rectangular areas around ϕ^* ; when considering $2D/3D$ angular spaces, these form polytopes.

2.3 Proposed Algorithm

The proposed algorithm is not the most computationally efficient; rather, it employs a brute-force strategy. Specifically, it generates a random steering vector and evaluates whether it meets the amplitude and phase constraints previously established. If the candidate vector is acceptable, it is returned as the solution; otherwise, the process continues until a valid vector is found or the maximum number of attempts is reached.

```

function [steeringVector, solutionIsFound] = FindSteeringVector(ulaSize, ...
lambda, antennaLocation, theta_ref, phi_ref, azimuthWidth_deg, ...
amplitudeTolerance, phaseTolerance_deg)

maxIter=10^(2*ulaSize);
phi=linspace(-pi/2,pi/2,1001);
theta=0;
solutionIsFound=false;

for n=1:maxIter
    phaseVector=2*pi*rand(ulaSize,1);
    steeringVector=exp(ji*phaseVector); % random steering vector

    arrayResponse=zeros(1,length(phi));
    for index=1:length(arrayResponse)
        arrayResponse(index)=(steeringVector'*exp(ji*myWaveVector(lambda,
theta, phi(index))'*antennaLocation).')/ulaSize;
    end
    %% check if the amplitude response fulfills the requirements

    selectedArrayResponse=arrayResponse(and(phi>=(phi_ref-deg2rad(azimuthWidth_de
g)/2), ...
phi<=(phi_ref+deg2rad(azimuthWidth_deg)/2)));

    %% amplitude
    maxAmplitude_dB=pow2db(abs(max(selectedArrayResponse)));
    minAmplitude_dB=pow2db(abs(min(selectedArrayResponse)));

    %% phase
    maxPhase_deg=rad2deg(max(angle(selectedArrayResponse)));
    minPhase_deg=rad2deg(min(angle(selectedArrayResponse)));
    if and(and(maxAmplitude_dB<=0,
minAmplitude_dB>=-amplitudeTolerance),...
        and(maxPhase_deg<=phaseTolerance_deg/2,
minPhase_deg>=-phaseTolerance_deg/2))
        %% set of phase shifts to be accepted
        solutionIsFound=true;
        break;
    end
end
end

```

Figure 4: Source code of the steering vector search algorithm implemented in MATLAB, illustrating the brute-force approach for finding a suitable steering vector subject to amplitude and phase constraints.

All presented code implementations will rely on this fundamental algorithm, albeit with some modifications to reflect ideal cases in later sections. Therefore, once the final version is introduced here, it will not be repeated in subsequent code presentations for clarity and brevity.

2.4 Real phase shifter implementation

The algorithm presented in the previous chapter solves the problem by generating a random phase within the continuous space $[0, 2\pi)$. However, digital communication systems operate with quantized phase values, and to make the simulations as realistic as possible, a B -bit phase quantization must be introduced. This implies that the number of possible phase combinations changes from a continuous infinite set to a discrete set of size $(2^B)^{\text{ulasize}}$, introducing additional complexity.

By fixing $\text{ulasize} = 5$, when the number of quantization bits B is low (e.g., 2 or 3) and the algorithm must satisfy a very strict amplitude and phase mask, it is not guaranteed that new valid weight vectors can always be found. On the contrary, it is likely that the algorithm will repeatedly use the same vectors. Conversely, increasing B to higher values (up to 16) reduces the quantization step and improves phase resolution, but in the worst case it may also increase the number of iterations required by the algorithm to find an acceptable weight vector, since a larger number of possible phase values are explored, not all of which satisfy the imposed constraints.

In this section, the scatter plots are examined as the quantization factor varies, while the number of transmitted symbols is fixed to 500, with the amplitude and phase constraints set to $\text{AmplitudeTol} = 0.8$ dB and $\text{PhaseTol} = 20^\circ$, respectively.

The intended user is placed at direction $(\theta_{\text{ref}}, \phi_{\text{ref}}) = (0^\circ, -20^\circ)$, while the eavesdropper is located at $(\theta_{\text{int}}, \phi_{\text{int}}) = (0^\circ, 25^\circ)$. These angular positions define the reference and interfering directions used throughout the simulations.

In each figure, four subplots are presented:

- the amplitude diagram (top-left),
- the phase diagram (bottom-left),
- the received constellation at the intended user (top-right),
- the constellation as observed by the eavesdropper (bottom-right).

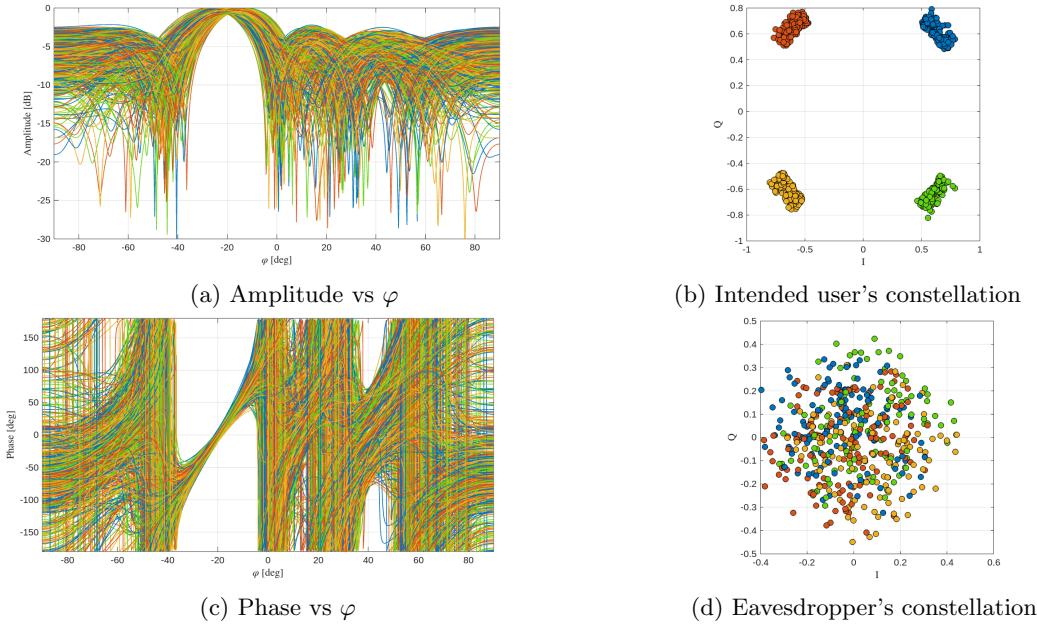


Figure 5: Simulation with $B = \infty$ bits of phase quantization.

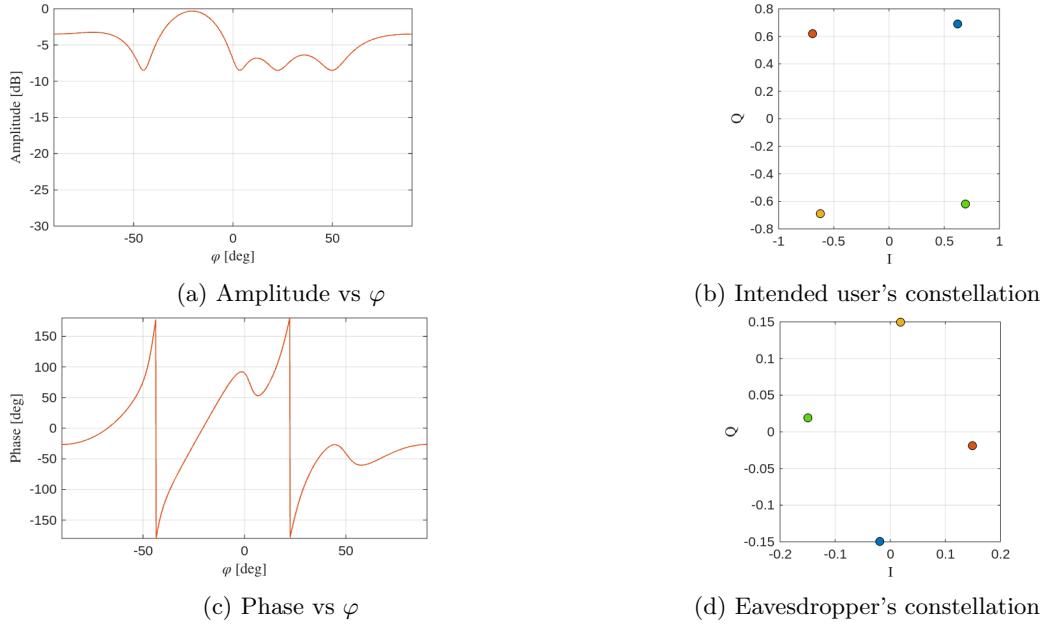


Figure 6: Simulation with $B = 2$ bits of phase quantization.

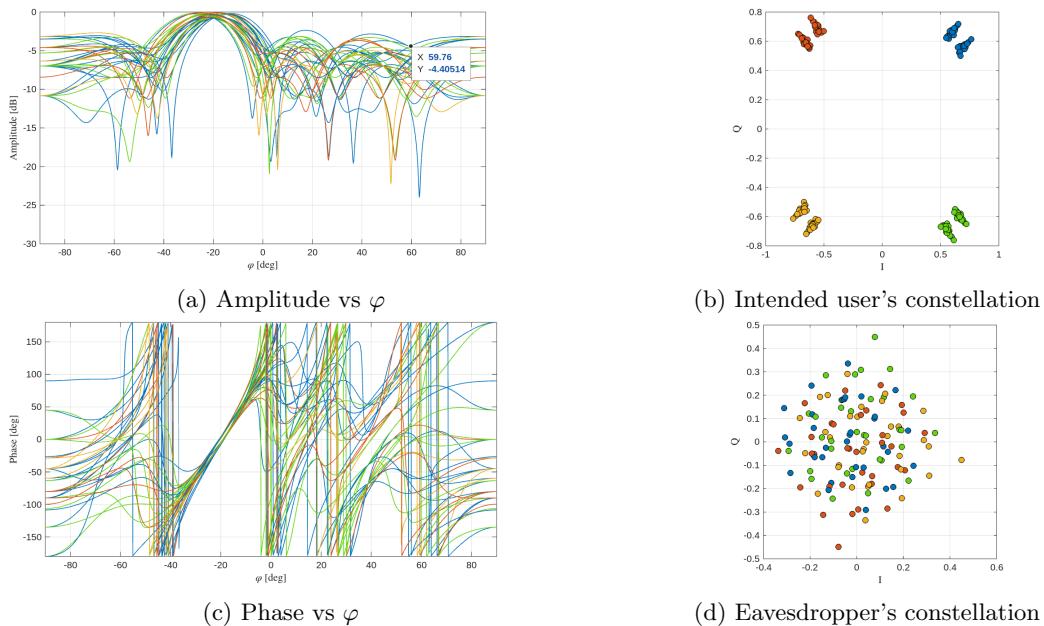


Figure 7: Simulation with $B = 3$ bits of phase quantization.

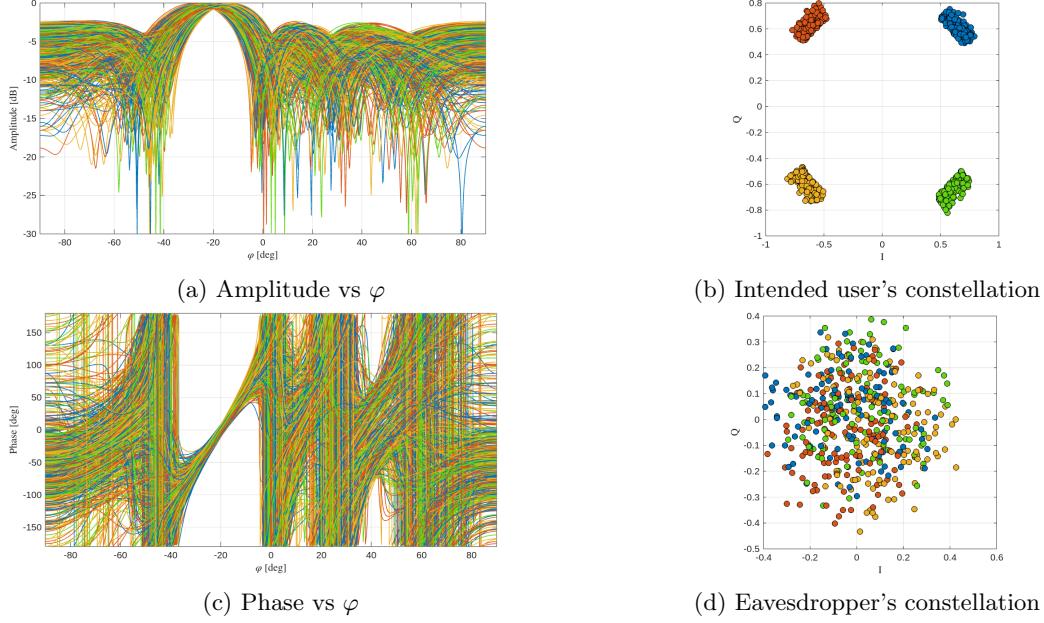


Figure 8: Simulation with $B = 5$ bits of phase quantization.

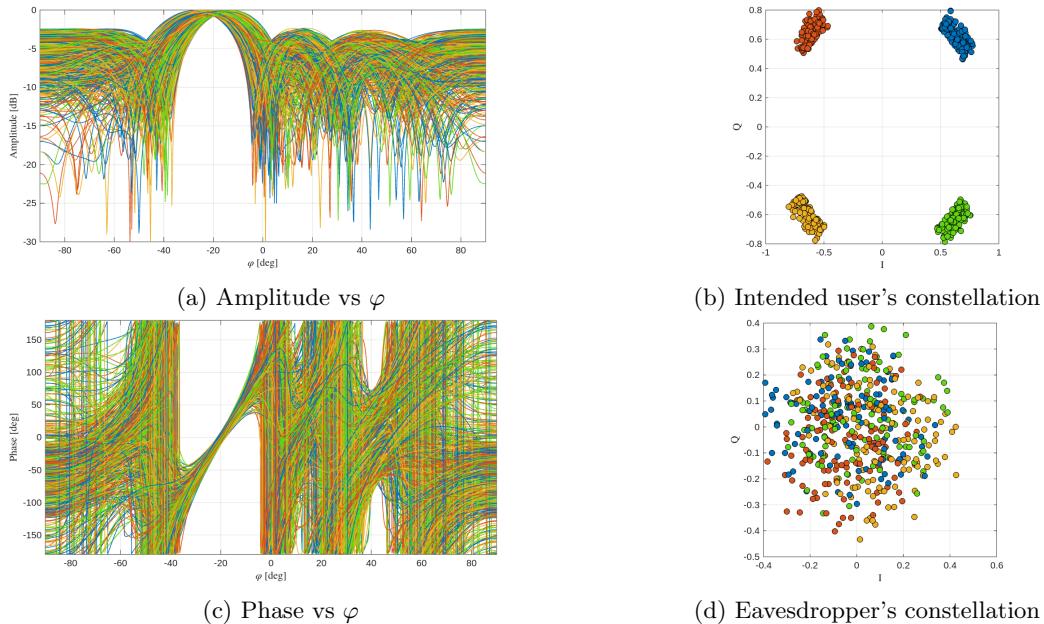


Figure 9: Simulation with $B = 6$ bits of phase quantization.

As shown in the presented figures, increasing the quantization factor B leads to an eavesdropper constellation that progressively approaches the ideal case, corresponding to $B = \infty$. In particular, when $B = 2$, the eavesdropper's received constellation remains essentially unchanged over all transmissions. This behavior is expected, since in this configuration the phases in the interval $[0, 2\pi[$ are quantized using only 2 bits, resulting in a very limited number of possible phase values. Consequently, across 500 transmitted symbols, the generated weight vector remains identical for all realizations. This phenomenon becomes clearer when analyzing the case $B = 3$. Here, the eavesdropper's constellation appears significantly more dispersed, and from the variations in both amplitude and phase margins, it is evident that multiple distinct signals are transmitted. This contrasts with the $B = 2$ scenario, where only a single signal configuration is repeatedly used, thus confirming the explanation provided above. As the quantization factor increases, the number of feasible vectors that satisfy the amplitude and phase constraints grows accordingly. Notably, with a quantization resolution of $B = 6$, the resulting performance observed over

500 transmitted elements already provides a close approximation to the ideal case.

B	Observed vectors	Unique vectors height2
100 000	1	
3	100 000	34
5	100 000	38 300
6	100 000	96 456

Table 1: Observed and unique vectors for different phase quantization levels.

Table 1 reports the number of unique vectors contained in several databases that I generated, each consisting of 100,000 weight vectors obtained using the algorithm described earlier with `ulaSize` = 5. As expected, for $B = 2$ only a single unique vector is produced across all samples. The number of unique vectors increases as the quantization level B grows.

In the following experiments, a quantization resolution of $B = 6$ is employed. Higher values are not considered, since the datasets used contain 100,000 elements, and as shown in Table 1 the case $B = 6$ already provides a close approximation to the ideal scenario. Although some vectors may be repeated, the number of unique vectors is nearly equal to the total number of generated elements.

B (bits)	Avg. Time [min]	Number of Explored Vectors
2	5.5	3.27×10^4
3	4.1	1.05×10^6
4	4.3	3.36×10^7
5	4.5	1.07×10^9
6	4.5	3.44×10^{10}
7	4.4	1.10×10^{12}
8	4.4	3.52×10^{13}
9	4.4	1.13×10^{15}
10	4.4	3.60×10^{16}
11	4.5	1.15×10^{18}
12	4.6	3.69×10^{19}
13	4.6	1.18×10^{21}
14	5.0	3.78×10^{22}
15	5.2	1.21×10^{24}
16	6.0	3.87×10^{25}
17	10.8	1.24×10^{27}
18	31.3	3.96×10^{28}
19	91.7	1.27×10^{30}

Table 2: Computational complexity analysis for `ulaSize` = 5

As shown in Table 2, which reports the average time required to find 5,000 feasible vectors for different values of B , together with the cardinality of the search space (given by $2^{B \cdot \text{ulaSize}}$), the average runtime remains nearly constant for quantization levels between $B = 4$ and $B = 16$. However, starting from $B = 17$, the computational time begins to increase significantly, indicating that the algorithm becomes progressively slower as the quantization resolution grows.

It is important to note that, for larger values of `ulaSize`, maintaining an acceptable runtime requires ensuring that B does not exceed a certain threshold, given fixed amplitude and phase tolerances. Otherwise, the algorithm would explore an excessively large search space, resulting in impractical execution times.

B (bits)	Avg.	Time [min]	Number of Explored Vectors
2	31.2		1.6348×10^4
3	30.5		2.09715×10^6
4	28.7		2.68435×10^8
5	31.5		3.43597×10^{10}
6	30.5		4.39805×10^{12}
7	29.5		5.62950×10^{14}
8	32.3		7.20576×10^{16}
9	28.5		9.22337×10^{18}
10	30.0		1.18059×10^{21}
11	28.7		1.51116×10^{23}
12	31.2		1.93428×10^{25}
13	31.7		2.47588×10^{27}
14	30.7		3.16913×10^{29}
15	32.3		4.05648×10^{31}
16	34.7		5.19230×10^{33}
17	39.7		6.64614×10^{35}
18	59.7		8.50706×10^{37}
19	227.3		1.08890×10^{40}

Table 3: Computational complexity analysis for `ulaSize` = 7.

Table 3 reports the computational time required to generate a feasible weight vector when `ulaSize` = 7. As expected, the time needed to obtain 5,000 acceptable vectors is significantly larger than in the case `ulaSize` = 5. This is because, instead of identifying 5 admissible amplitude–phase configurations, the algorithm must now find 7 simultaneous realizations that satisfy the amplitude and phase constraints. Since each antenna element is quantized independently, the probability of obtaining a feasible steering vector decreases exponentially with the array size. More precisely, if $p(B)$ denotes the probability that a single antenna element satisfies the quantization-dependent constraints, then the probability that an entire candidate steering vector is feasible is given by

$$P_{\text{feasible}}(B, \text{ulaSize}) = \mathbb{P}(\text{all } \text{ulaSize} \text{ elements satisfy constraints}) = [p(B)]^{\text{ulaSize}}. \quad (13)$$

which directly follows from the random and independent generation of the phases in the algorithm. Consequently, increasing the array size from 5 to 7 elements reduces P_{feasible} and increases the expected number of iterations required to find an acceptable vector, thus explaining the higher average runtime observed for `ulaSize` = 7.

Disregarding this constant shift in average runtime, it is evident that the qualitative behavior as a function of B remains essentially unchanged with respect to the `ulaSize` = 5 case. In particular, for `ulaSize` = 7 the generation time remains approximately constant for quantization levels below B = 16. Beyond this point, however, the runtime grows dramatically due to the exponential increase of the search space, which scales as $2^{B \cdot \text{ulaSize}}$. We may therefore conclude that increasing the number of antennas raises the average time required to generate feasible weight vectors, and that when the quantization order becomes too large specifically for $B \geq 16$ the average generation time grows uncontrollably, eventually becoming impractical for real-time or large-scale beamforming applications.

```

%% Vector generation
parfor n = 1:N_pattern
    [~,idx] = min(abs(tx_pattern(n) - qpskConstellation));
    colors(n,:) = myColor(idx,:);

    [w, ok] = FindSteeringVector(ulaSize, lambda, antennaLocation, ...
        theta_ref, phi_ref, azimuthWidth_deg, amplitudeTolerance, ...
        phaseTolerance_deg, B);

    fprintf("genero %d vettori \n", n);
    if ~ok
        warning('No solution found for the %d simbol \n', n);
        continue;
    end
    W_used(:,n) = w;
    a_ref = exp(ji*myWaveVector(lambda,theta_ref,phi_ref)'*antennaLocation)';
    a_int = exp(ji*myWaveVector(lambda,theta_int,phi_int)'*antennaLocation)';
    rx_ref_pattern(n) = (w'*a_ref)/ulaSize * tx_pattern(n);
    rx_int_pattern(n) = (w'*a_int)/ulaSize * tx_pattern(n);
end

```

Figure 10: Matlab code used for data generations

In Fig. 10, the MATLAB code used to generate the data for the previous scatter plots is shown. The script implements the synthesis of transmit weight vectors using a parallelized `parfor` loop to improve computational efficiency.

For each transmitted symbol s_n , the algorithm performs the following operations:

1. The symbol is associated with its corresponding QPSK constellation point from the set

$$\mathcal{S}_{\text{QPSK}} = \left\{ \frac{1+j}{\sqrt{2}}, \frac{-1+j}{\sqrt{2}}, \frac{-1-j}{\sqrt{2}}, \frac{1-j}{\sqrt{2}} \right\}.$$

2. A candidate weight vector $\mathbf{w}_n \in \mathbb{C}^{N \times 1}$ is synthesized through the function `FindSteeringVector`, ensuring compliance with the amplitude and phase masks defined around the desired direction $(\theta_{\text{ref}}, \phi_{\text{ref}})$:

$$\mathbf{w}_n = f(\theta_{\text{ref}}, \phi_{\text{ref}}),$$

If no valid \mathbf{w}_n satisfying these conditions is found, the iteration is skipped.

3. Once a valid weight vector is obtained, the array steering vectors toward the desired and interfering directions are computed as

$$\mathbf{a}_{\text{ref}} = \exp(j \mathbf{k}^T(\theta_{\text{ref}}, \phi_{\text{ref}}) \mathbf{U}), \quad \mathbf{a}_{\text{int}} = \exp(j \mathbf{k}^T(\theta_{\text{int}}, \phi_{\text{int}}) \mathbf{U}),$$

where $\mathbf{U} = [\mathbf{u}_0, \dots, \mathbf{u}_{N-1}]$ contains the antenna coordinates.

4. The received complex symbols at the legitimate receiver and at the eavesdropper are finally obtained as

$$r_{\text{ref}}(n) = \frac{1}{N} \mathbf{w}_n^H \mathbf{a}_{\text{ref}} s_n, \quad r_{\text{int}}(n) = \frac{1}{N} \mathbf{w}_n^H \mathbf{a}_{\text{int}} s_n.$$

In this simulations, the function `FindSteeringVector` uses the following parameters:

$$\text{azimuthWidth_deg} = 1^\circ, \quad \text{amplitudeTolerance} = 0.8 \text{ dB}, \quad \text{phaseTolerance_deg} = 20^\circ.$$

These values define the search range and the allowed deviations for amplitude and phase. They make it easier for the algorithm to find valid solutions under discrete phase quantization, although the results are not fully optimal. Because of these tolerances, the received symbols at the legitimate user appear slightly shifted from the ideal QPSK positions, showing the small imperfections introduced by the relaxed constraints.

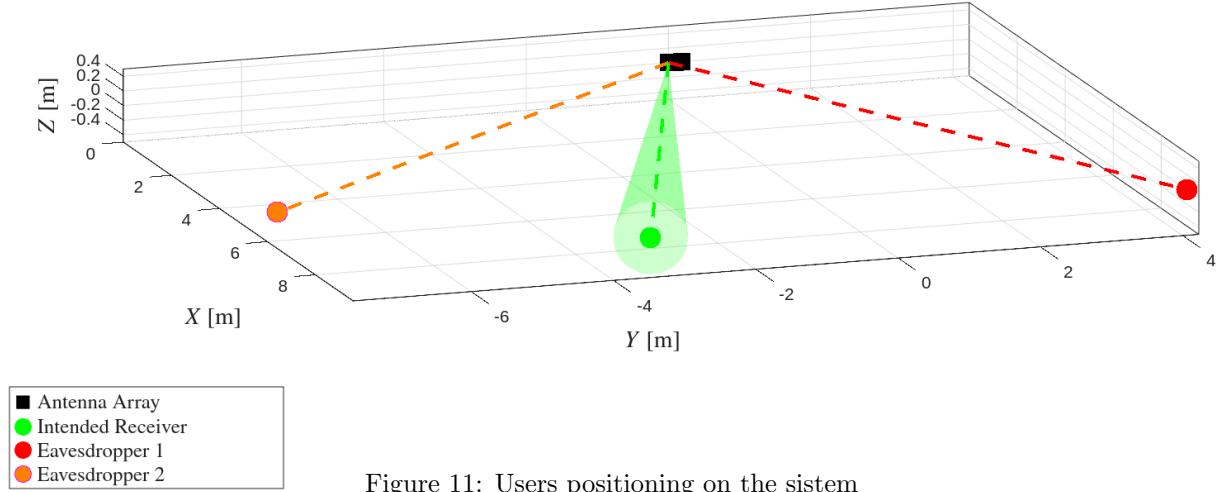
2.5 BER performance evaluation

In this section, we evaluate the Bit Error Rate (BER) performance of the proposed DM scheme as a function of the energy-per-bit to noise ratio, E_b/N_0 . The analysis considers the influence of several key system parameters, namely the modulation format, the number of antennas, and the amplitude and phase tolerances employed during the selection of feasible weight vectors.

In particular, we examine three array configurations with `ulaSize` = 3, 5, 7 antenna elements, and we assess BER performance for both normalized QPSK and 16-QAM modulation schemes.

The simulation setup adopted for BER evaluation is depicted in Figure 11. Since all users are located at distances greater than the Fraunhofer distance, the system operates in the far-field region. Under this assumption, the radiated wavefronts can be approximated as planar over the spatial extent of the receivers, and the directions of departure of the transmitted signal can be modeled solely by their azimuth angles. For simplicity, we further assume that all receivers lie at the same radial distance from the uniform linear array (ULA).

The ULA is placed along the y -axis and is symmetric with respect to the origin $(0, 0, 0)$. Three users are considered: the intended receiver (green marker), positioned at an azimuth angle of -25° , and two potential eavesdroppers located at 25° (red marker) and -50° (orange marker). This configuration enables the evaluation of the directional selectivity of the beamforming vectors as well as the system's capability to degrade the signal quality outside the desired transmission direction.



In the following plots, which compare BER versus E_b/N_0 , the curves are represented as follows:

- Blue: theoretical curve
- Green: curve of the desired receiver
- Orange: curve of an eavesdropper located at $\theta_{int1} = 0^\circ$, $\phi_{int1} = 25^\circ$
- Red: curve of an eavesdropper located at $\theta_{int2} = 0^\circ$, $\phi_{int2} = -50^\circ$

2.5.1 BER Analysis for QPSK modulation

In this section, we evaluate the bit error rate (BER) as a function of E_b/N_0 for QPSK modulation, considering different numbers of antennas.

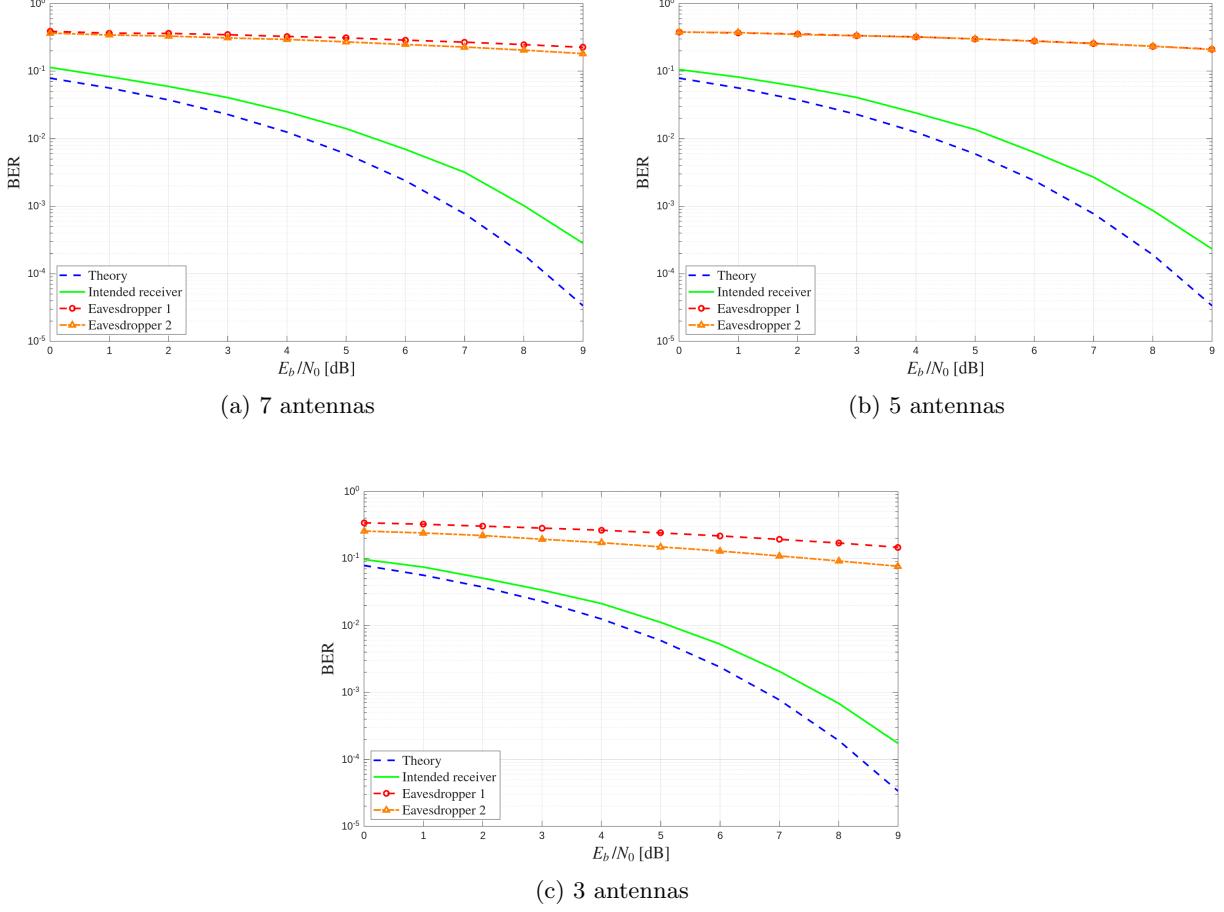


Figure 12: BER qpsk with different ula sizes.

As observed in Figure 12, the simulated BER does not perfectly match the theoretical curve, despite a large number of trials being used in its generation. This discrepancy arises because the proposed algorithm employs a suboptimal solution that does not coincide with the true optimal weight vector. Specifically, the gap between the theoretical and simulated curves is due to the acceptance of weight vectors that satisfy predefined amplitude and phase margins when generating the steering vectors.

From the figure, it is evident that the curve for the intended receiver that most closely follows the theoretical performance corresponds to `ulaSize` = 3. However, in the context of beamforming, choosing a small array size is not necessarily optimal. Although a lower `ulaSize` allows the desired user to achieve a lower BER, it negatively impacts spatial security: as shown in the figure, eavesdroppers experience relatively low BER when `ulaSize` = 3, meaning they can decode the confidential signal more reliably compared to scenarios with larger array sizes. Nevertheless, if turbo coding were applied, even in this case, since the BER remains below 10^{-1} , the transmitted signal would still be effectively undecodable for unauthorized users.

It is essential to emphasize that DM is designed not only to direct the signal toward the intended user but also to ensure that any receiver located in a different spatial position is unable to correctly decode the transmitted data. As observed, increasing the number of antennas slightly degrades the BER performance for the intended user within an acceptable range, while it significantly increases the BER experienced by the eavesdroppers. This trade-off improves the overall secrecy performance of the system.

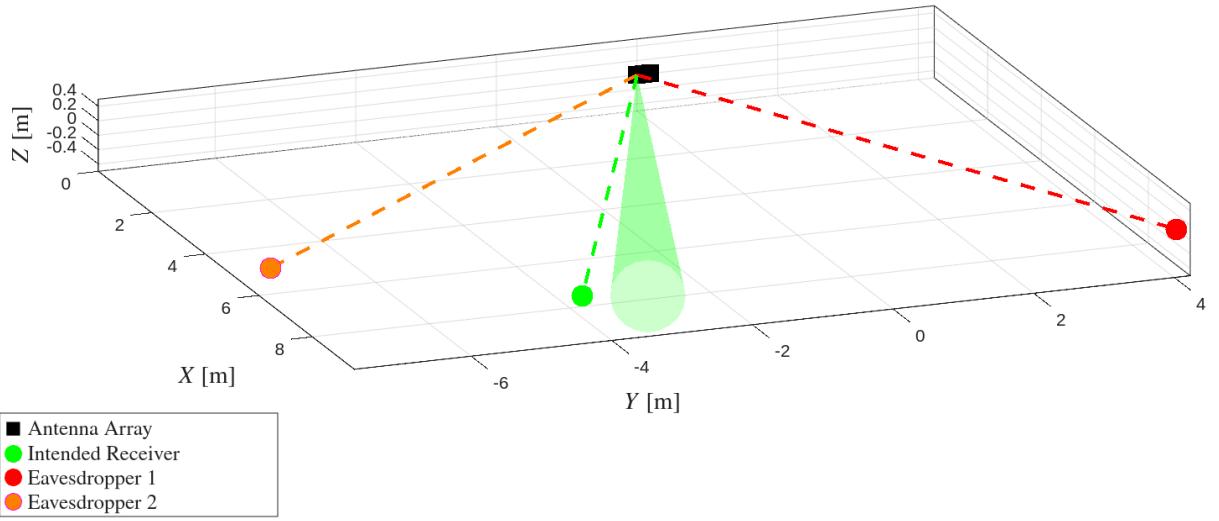


Figure 13: Users positioning on the sistem when the desired user is located 5° away from the assumed direction.

We now consider and analyze the scenario in which the transmitter assumes that the intended receiver is located at $\phi_{\text{ref}} = -20^\circ$, while in reality the user is positioned at $\phi_{\text{ref}} = -25^\circ$, as illustrated in Figure 11. This case represents a small but meaningful angular misalignment between the assumed and actual positions of the intended receiver, which can occur in practical wireless systems due to estimation errors, user mobility, or measurement inaccuracies.

In this scenario, the beamforming vector is designed based on the assumed user location, and therefore the main lobe of the transmitted signal is directed towards $\phi_{\text{ref}} = -20^\circ$. As the actual user is shifted by 5° , the received signal experiences a reduction in array gain, which may affect the bit error rate (BER).

By analyzing the BER performance under this condition, we can evaluate the robustness of the proposed scheme to small deviations in user positioning. The desired outcome is that the misaligned user still receives the signal with a low BER, ensuring reliable communication for legitimate users.

This is particularly important because, if an unintended or malicious receiver were able to decode the signal correctly despite being outside the intended spatial location, it could compromise the security of the system.

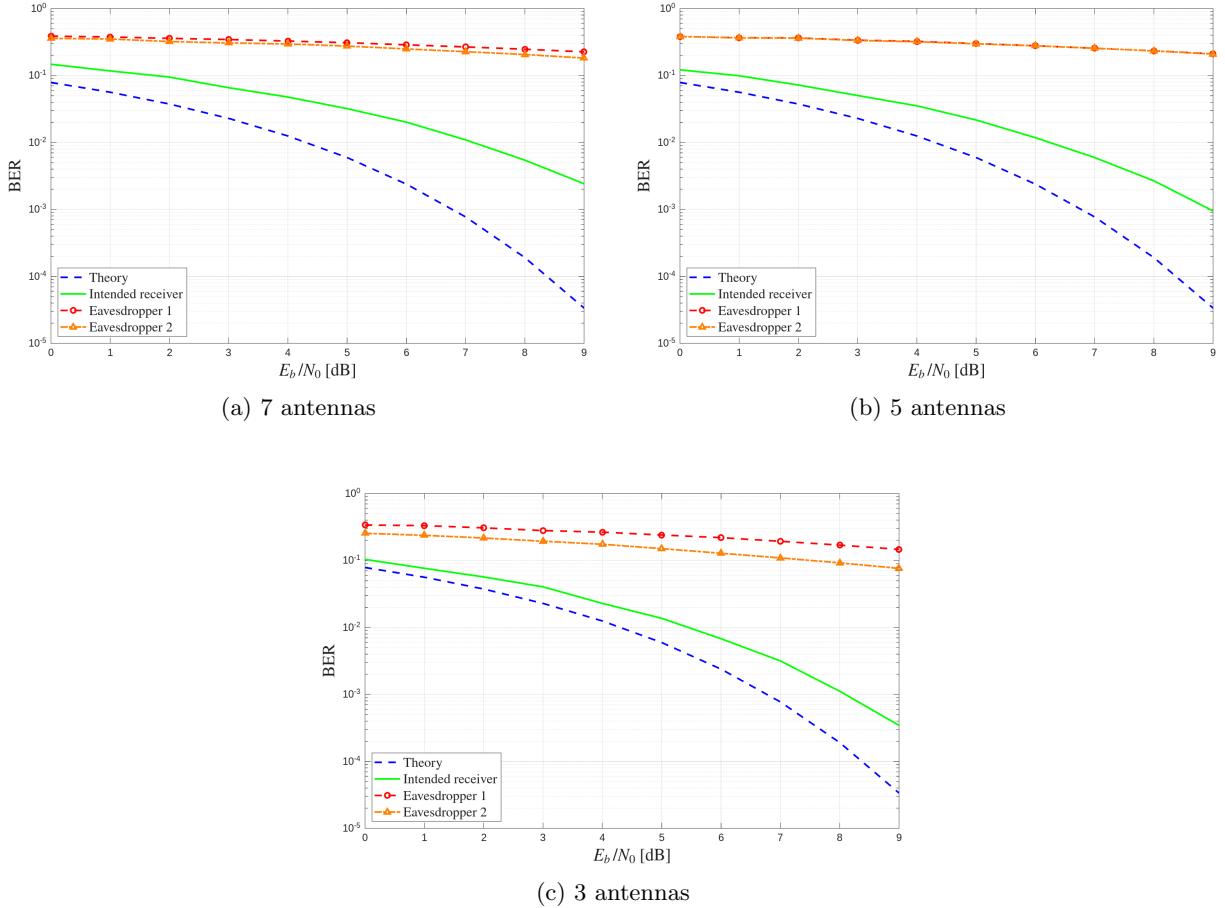


Figure 14: BER qpsk with different ula sizes when the desired user is located 5° away from the assumed direction.

As shown in Figure 14, a displacement of 5° between the expected and the actual direction of the intended user leads to a noticeable impact on the BER performance. As observed in the aligned scenario, the configuration with `ulaSize` = 3 still provides the lowest BER at the intended location, whereas `ulaSize` = 7 yields the highest BER degradation when the user is perfectly aligned with the assumed direction.

In this misalignment scenario, however, a low BER is no longer the desirable outcome. Since the objective of is to enhance spatial security, the system should ideally produce a *high* BER for any receiver that is not located exactly along the intended direction. In other words, a nearby user that is slightly displaced from the correct spatial position should experience a significantly higher error probability, preventing successful demodulation of the confidential signal.

This behavior is clearly reflected in the results: increasing the number of antennas improves spatial selectivity, thus enhancing the security level at the cost of higher computational effort. While generating weight vectors for `ulaSize` = 3 is considerably less demanding than for `ulaSize` = 7, the latter configuration exhibits a far stronger suppression of unintended receivers located close to the intended direction. Consequently, a larger array (`ulaSize` = 7) provides superior spatial security in misalignment conditions, despite the increased design complexity.

The observed behavior can be theoretically justified by considering the relationship between the beamwidth of a uniform linear array (ULA) and the number of antennas. For a ULA with N elements spaced by d , the approximate half-power beamwidth (HPBW) is given by:

$$\text{HPBW} \approx \frac{2\lambda}{Nd} \quad (14)$$

Where λ is the wavelength. This expression shows that as the number of antennas N increases, the beam becomes narrower. Consequently, even a small angular displacement of the intended user from the assumed direction results in a significant degradation of the received signal quality, which is consistent with the BER behavior observed in Figure 14.

This property illustrates a key advantage of directional modulation: increasing the number of antennas not only enhances signal directivity but also improves the physical-layer security by ensuring that unintended receivers or eavesdroppers experience a higher BER.

BER Simulation Algorithm

The following procedure is employed to compute the Bit Error Rate (BER) as a function of E_b/N_0 for the QPSK modulation scheme within the DM framework.

The simulation makes use of a precomputed database of valid weight vectors, stored in files whose names encode all relevant generation parameters according to the format:

`steeringVector%d.%d-%d.%d.%d.mat`

where the placeholders correspond to:

- `ulaSize`: number of antennas in the uniform linear array (ULA),
- `grad`: desired steering angle in degrees,
- `azimuthWidth_deg`: azimuthal beamwidth used during weight–vector validation,
- `amplitudeTolerance`: admissible amplitude deviation from the ideal response,
- `phaseTolerance_deg`: admissible phase deviation (in degrees),
- `B`: quantization resolution in bits.

For example, the file

`steeringVector3_-20-1_9.800000e-01_5_6.mat`

indicates that the stored weight vectors were generated for a 3-element ULA, with a target direction of -20° , an azimuthal width of 1° , an amplitude tolerance of 0.98, a phase tolerance of 5° , and a phase quantization of 6 bits.

Each file contains approximately 10^5 valid weight vectors that satisfy both the amplitude and phase margin constraints, ensuring a sufficiently large sample set to obtain statistically meaningful BER estimates.

Simulation process. For each value of E_b/N_0 , expressed in decibels, the algorithm performs a simulation where:

1. A random weight vector \mathbf{w} is selected from the database.
2. The corresponding *steering vectors* are computed for three spatial positions:

$$\mathbf{a}_{\text{ref}} = e^{j \mathbf{k}(\theta_{\text{ref}}, \phi_{\text{ref}})^T \mathbf{r}}, \quad \mathbf{a}_{\text{int1}} = e^{j \mathbf{k}(\theta_{\text{int1}}, \phi_{\text{int1}})^T \mathbf{r}}, \quad \mathbf{a}_{\text{int2}} = e^{j \mathbf{k}(\theta_{\text{int2}}, \phi_{\text{int2}})^T \mathbf{r}}$$

where $\mathbf{k}(\theta, \phi)$ is the wave vector and \mathbf{r} represents the antenna locations.

3. A random QPSK symbol $s \in \{e^{j\pi/4}, e^{3j\pi/4}, e^{5j\pi/4}, e^{7j\pi/4}\}$ is transmitted.

4. The received noiseless complex baseband signals for the three users are computed as:

$$r_{\text{ref}} = \frac{\mathbf{w}^H \mathbf{a}_{\text{ref}}}{N} s, \quad r_{\text{int1}} = \frac{\mathbf{w}^H \mathbf{a}_{\text{int1}}}{N} s, \quad r_{\text{int2}} = \frac{\mathbf{w}^H \mathbf{a}_{\text{int2}}}{N} s$$

where N is the number of antennas and $(\cdot)^H$ denotes the Hermitian transpose.

5. Complex additive white Gaussian noise (AWGN) is added to each signal:

$$n \sim \mathcal{N}(0, \sigma^2), \quad \text{with} \quad \sigma^2 = \frac{1}{E_b/N_0 \cdot k_{\text{QPSK}}}$$

where $k_{\text{QPSK}} = 2$ is the number of bits per symbol.

Thus, the received signals become:

$$\tilde{r}_i = r_i + n_i, \quad i \in \{\text{ref, int1, int2}\}$$

6. Each receiver performs maximum-likelihood (ML) detection by selecting the constellation point s' that minimizes the Euclidean distance:

$$s' = \arg \min_{s_k \in \mathcal{S}} |\tilde{r}_i - s_k|$$

where \mathcal{S} is the QPSK constellation set.

7. Bit errors are counted by comparing the transmitted bit sequence \mathbf{b}_{tx} with the detected one \mathbf{b}_{rx} . The loop continues until the intended receiver accumulates a predefined number of bit errors $N_{\text{max,err}} = 1000$.

BER computation. Once the error threshold is reached, the Bit Error Rate is computed as:

$$\text{BER}_i(E_b/N_0) = \frac{N_{\text{errors},i}}{k_{\text{QPSK}} \cdot N_{\text{symbols}}}$$

for each receiver $i \in \{\text{ref, int1, int2}\}$.

The theoretical BER for QPSK in an AWGN channel is given by:

$$\text{BER}_{\text{theory}}(E_b/N_0) = Q\left(\sqrt{2 \frac{E_b}{N_0}}\right)$$

where $Q(\cdot)$ denotes the Gaussian Q -function:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt.$$

```

%% Loop Eb/N0
for k = 1:length(EbN0_dB)
    EbN0_lin = 10^(EbN0_dB(k)/10)
    N_symbols = 0;
    N_errors_dest = 0;
    N_errors_int = 0;
    N_errors_int2 = 0;
    while N_errors_dest < nMax_error
        % Seleziona un vettore casuale dal file
        idxVec = randi(nVectors);
        W_test = allVectors{idxVec};
        col = randi(size(W_test,2));
        w = W_test(:,col);
        % steering vectors
        a_ref = exp(1j*myWaveVector(lambda,theta_ref,phi_ref)'*antennaLocation).';
        a_int = exp(1j*myWaveVector(lambda,theta_int,phi_int)'*antennaLocation).';
        a_int2 = exp(1j*myWaveVector(lambda,theta_int2,phi_int2)'*antennaLocation).';
        % random QPSK symbol
        symIdx = randi(4,1,1);
        txSym = qpskConstellation(symIdx);
        txBits = qpskBits(symIdx,:);
        gain_dest = abs((w'*a_ref)/ulaSize);
        gain_int = abs((w'*a_int)/ulaSize);
        gain_int2 = abs((w'*a_int2)/ulaSize);

        rx_dest = gain_dest * txSym;
        rx_int = gain_int * txSym;
        rx_int2 = gain_int2 * txSym;

        % noise generation
        sigma2 = 1 / (EaN0_lin * k_qpsk);

        noise_dest = sqrt(sigma2 / 2) * (randn + 1j*randn);
        noise_int = sqrt(sigma2 / 2) * (randn + 1j*randn);
        noise_int2 = sqrt(sigma2 / 2) * (randn + 1j*randn);

        rx_dest_noise = rx_dest + noise_dest;
        rx_int_noise = rx_int + noise_int;
        rx_int2_noise = rx_int2 + noise_int2;

        % ML
        [~, idx] = min(abs(rx_dest_noise - qpskConstellation));
        rxBits_dest = qpskBits(idx,:);
        [~, idx] = min(abs(rx_int_noise - qpskConstellation));
        rxBits_int = qpskBits(idx,:);
        [~, idx] = min(abs(rx_int2_noise - qpskConstellation));
        rxBits_int2 = qpskBits(idx,:);

        % update counters variables
        N_symbols = N_symbols + 1;
        N_errors_dest = N_errors_dest + sum(txBits ~= rxBits_dest);
        N_errors_int = N_errors_int + sum(txBits ~= rxBits_int);
        N_errors_int2 = N_errors_int2 + sum(txBits ~= rxBits_int2);
    end
    % BER generation
    BER_dest_sim(k) = N_errors_dest / (k_qpsk*N_symbols);
    BER_int_sim(k) = N_errors_int / (k_qpsk*N_symbols);
    BER_int2_sim(k) = N_errors_int2 / (k_qpsk*N_symbols);
    fprintf("Eb/N0 = %d dB \t| Symbols: %d \t| BER_dest = %.4e\n", EbN0_dB(k),
    N_symbols, BER_dest_sim(k));
end
%% BER theory QPSK
EbN0_lin = 10.^ (EbN0_dB/10);
BER_theory = qfunc(sqrt(2*EbN0_lin));

```

Figure 15: MATLAB simulation code used for BER estimation.

2.5.2 BER Analysis for 16-QAM Modulation

The same simulation framework used for QPSK is employed to evaluate the BER performance of the 16-QAM modulation scheme. For conciseness, we highlight only the aspects that differ from the QPSK case.

The 16-QAM constellation consists of

$$M_{\text{QAM}} = 16 \text{ symbols,}$$

each encoding

$$k_{\text{QAM}} = \log_2(M_{\text{QAM}}) = 4 \text{ bits.}$$

As a consequence, the constellation exhibits multiple amplitude levels, making it inherently more sensitive to noise, quantization effects, and distortions introduced by the process. In particular, for a fixed value of E_b/N_0 , the effective Euclidean distance between constellation points is smaller than in QPSK, leading to a higher theoretical BER. This behavior is illustrated in Figure 16, which compares the demodulated 16-QAM symbols for the intended user and an eavesdropper.

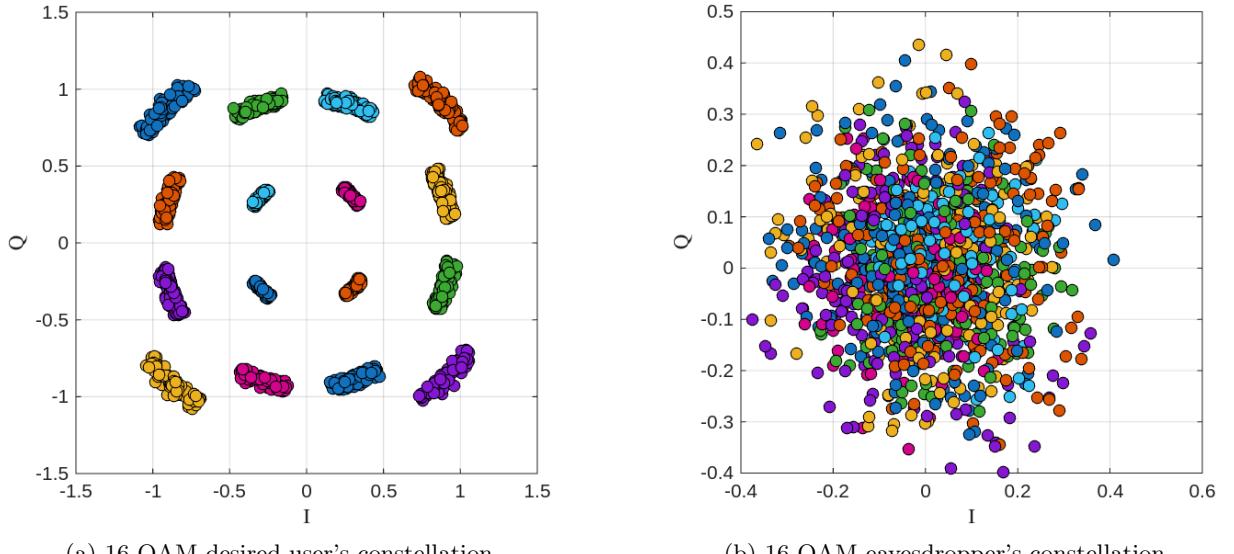


Figure 16: Comparison between the received 16-QAM constellations of the desired user and an eavesdropper.

The noise variance corresponding to each value of E_b/N_0 is computed as

$$\sigma^2 = \frac{1}{(E_b/N_0) k_{\text{QAM}}}.$$

The theoretical BER for 16-QAM modulation in AWGN is given by:

$$\text{BER}_{\text{theory}} = \frac{1}{k_{\text{QAM}}} \left[1 - \left(1 - 2 \left(1 - \frac{1}{\sqrt{M_{\text{QAM}}}} \right) Q \left(\sqrt{\frac{3k_{\text{QAM}} E_b/N_0}{M_{\text{QAM}} - 1}} \right) \right)^2 \right].$$

All remaining simulation steps—including random extraction of feasible weight vectors, steering vector computation, noisy transmission, maximum-likelihood symbol detection, and BER accumulation—are identical to the QPSK case. This ensures that the performance comparison between the two modulation schemes is carried out under perfectly consistent array, quantization, and conditions.

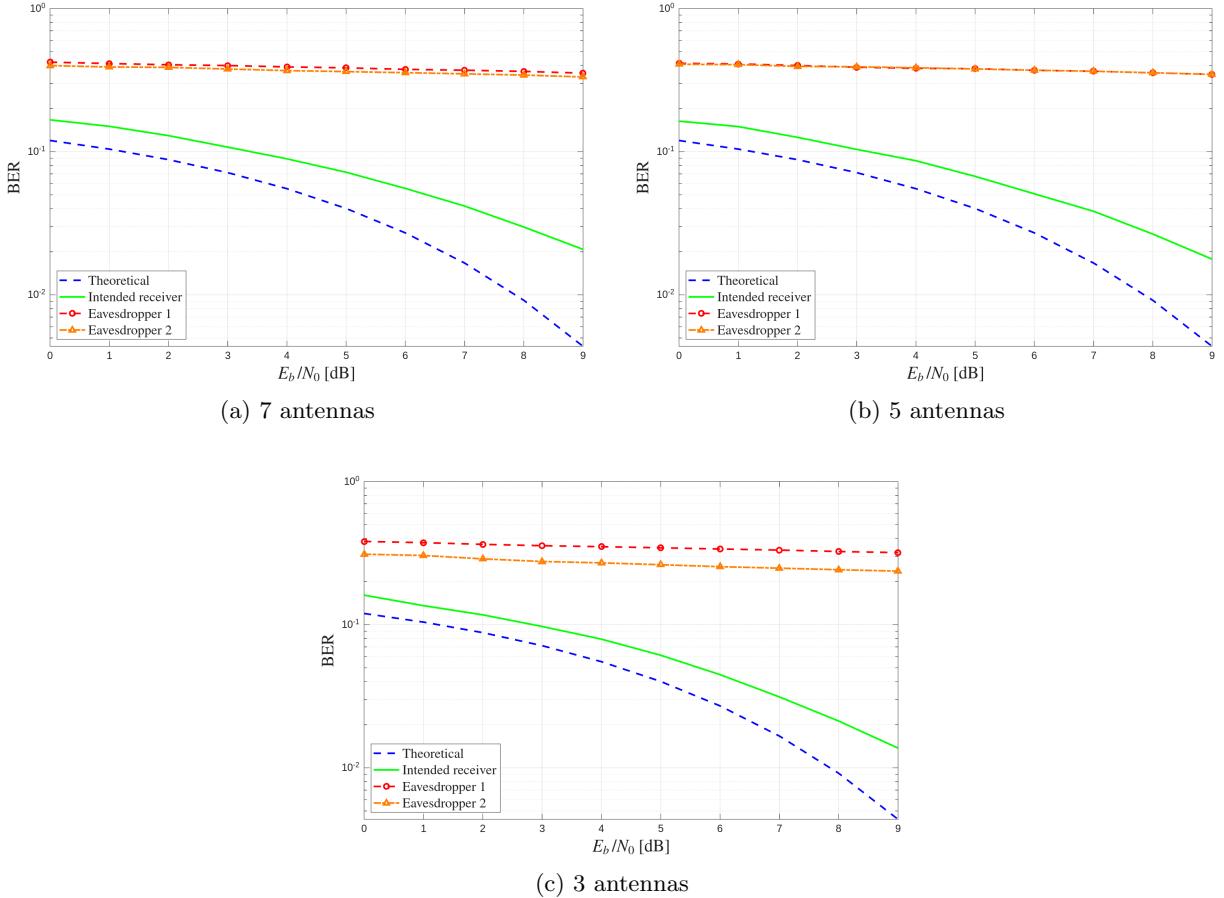


Figure 17: BER 16-QAM with different ula sizes.

As shown in Figure 17, the scheme employing 16-QAM exhibits a behavior consistent with the trends observed for the QPSK case. However, both the simulated and theoretical BER curves attain higher error levels compared to QPSK, reflecting the increased sensitivity of 16-QAM to noise and distortion for the same E_b/N_0 range.

This outcome is expected, as 16-QAM conveys a larger number of bits per symbol. Consequently, a higher symbol energy is required to maintain the same target BER. In practical terms, although 16-QAM yields a higher BER at equal E_b/N_0 , it compensates for this drawback by offering higher spectral efficiency, carrying twice as many bits per symbol as QPSK.

As observed in previous analyses, the configuration with `ulaSize = 3` provides the simulated BER curve that most closely resembles the theoretical performance. Nevertheless, this configuration also enables eavesdroppers to achieve a relatively low BER, thereby weakening the spatial security of the system. Increasing the number of antennas enhances spatial directivity and reduces the probability that unintended users can correctly decode the confidential signal, albeit at the cost of a slight degradation in the BER performance experienced by the intended receiver.

Following the same methodology as before, we also evaluate the impact of angular misalignment by considering the scenario in which the desired user is displaced from $\phi_{\text{ref}} = -20^\circ$ to $\phi_{\text{ref}} = -25^\circ$, while the transmitter continues to steer the beam towards the original direction. This analysis allows us to quantify the degradation in system performance under small deviations between the assumed and actual user positions and to assess the robustness of the strategy in practical, imperfectly aligned scenarios.

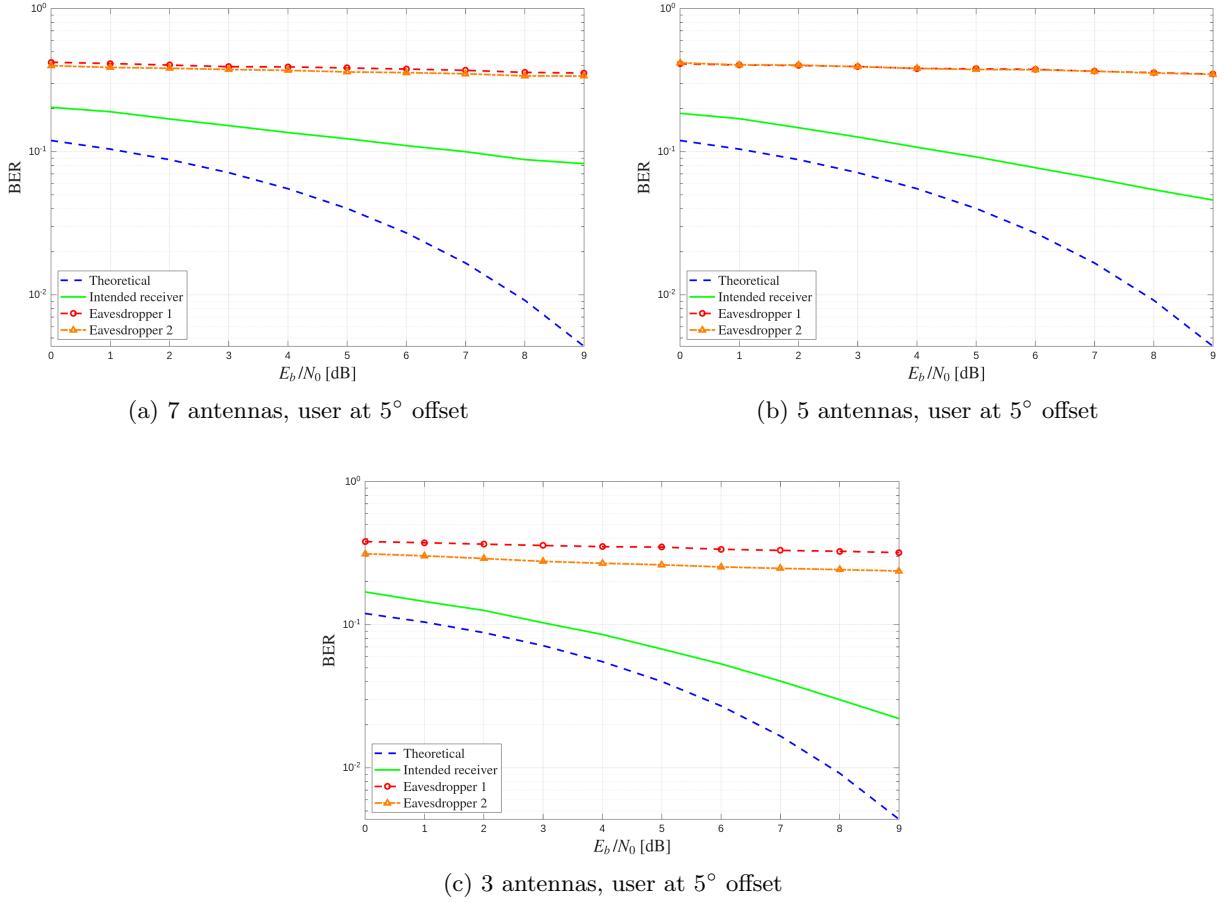


Figure 18: BER 16-QAM with different ula sizes when the desired user is located 5° away from the assumed direction.

As expected, and as shown in Figure 18, the same considerations discussed for the QPSK case also apply to the 16-QAM scenario. In particular, increasing the number of antennas in the uniform linear array improves the spatial security provided by the scheme.

By comparing the two experiments reported in Figures 17 and 18, we observe a clear trade-off between the BER experienced by the intended receiver when it is perfectly aligned with the assumed direction and the BER measured by a receiver located near the desired user. Specifically, the configuration with `ulaSize = 7` yields the worst BER performance for the intended user under perfect alignment, with a simulated BER higher than those obtained for `ulaSize = 3` and `ulaSize = 5`.

However, this configuration simultaneously provides the highest degree of spatial security, as unintended receivers located close to the target direction experience significantly deteriorated constellation quality. It is also worth noting that the 16-QAM case with `ulaSize = 7` exhibits even stronger security performance than the corresponding QPSK scenario, further highlighting the enhanced spatial selectivity achieved with a larger antenna array.

2.5.3 16-QAM vs QPSK in Directional Modulation

In the previous sections, we analyzed the behavior of the scheme under two different modulation formats, namely QPSK and 16-QAM.

The objective of the present section is to compare these two approaches and assess which one provides better performance when DM is employed. As demonstrated earlier, for both modulation schemes the configuration with `ulaSize` = 7 proved to be the most effective in terms of spatial security. Increasing the number of antennas enhances the directivity of the radiation pattern and strengthens the robustness against eavesdroppers, although this comes at the cost of higher computational complexity in the generation and selection of valid weight vectors.

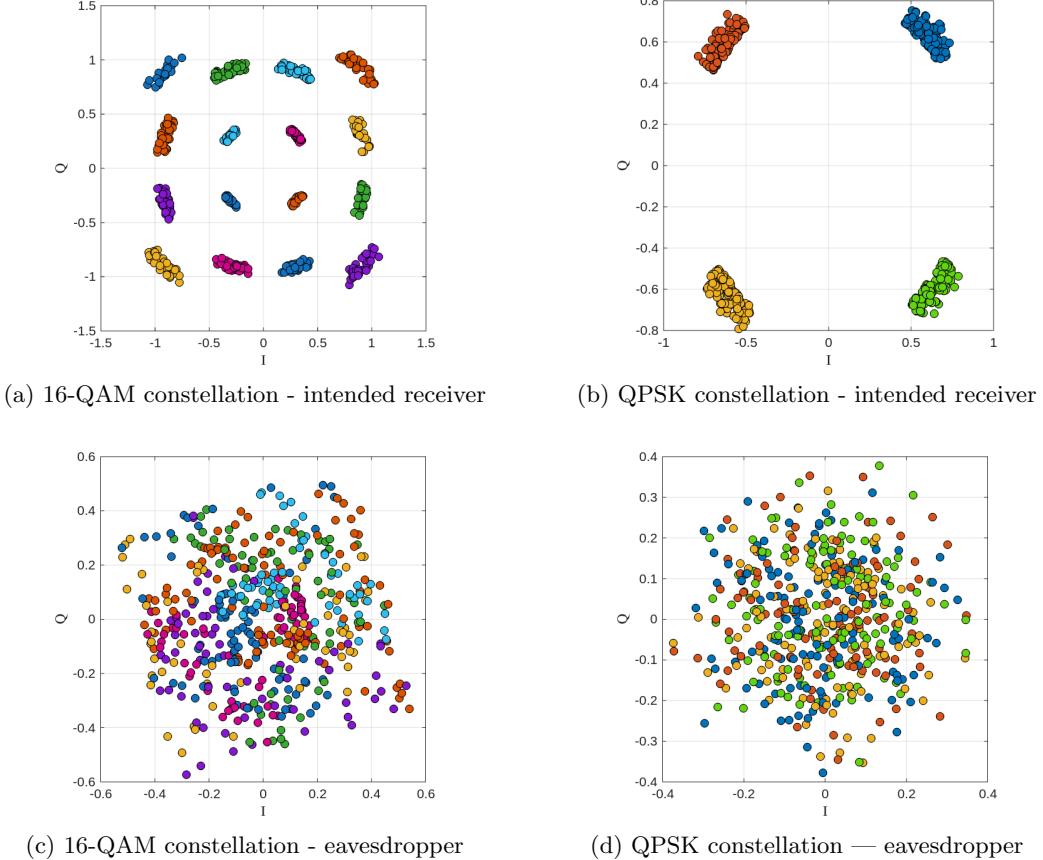


Figure 19: Comparison between QPSK and 16-QAM constellations for the desired user (top row) and the eavesdropper (bottom row)

As shown in Figure 20, the intended receiver consistently observes well-defined symbol mappings for both 16-QAM and QPSK constellations. The plots were generated with 500 transmitted symbols for each modulation scheme, and none of the received symbols appears to experience significant misclassification into neighboring decision regions. This indicates that, for the intended receiver, both modulations are reliably received under the considered configuration.

It is important to note that 16-QAM transmits 16 distinct symbols, whereas QPSK transmits only 4 symbols. When the constellations are normalized to the same average power, the distance between symbols is scaled accordingly, and the noise is considered relative to this power. Consequently, the relative susceptibility of 16-QAM versus QPSK to noise remains consistent under normalization, and it is not inherently "more secure" simply due to symbol packing.

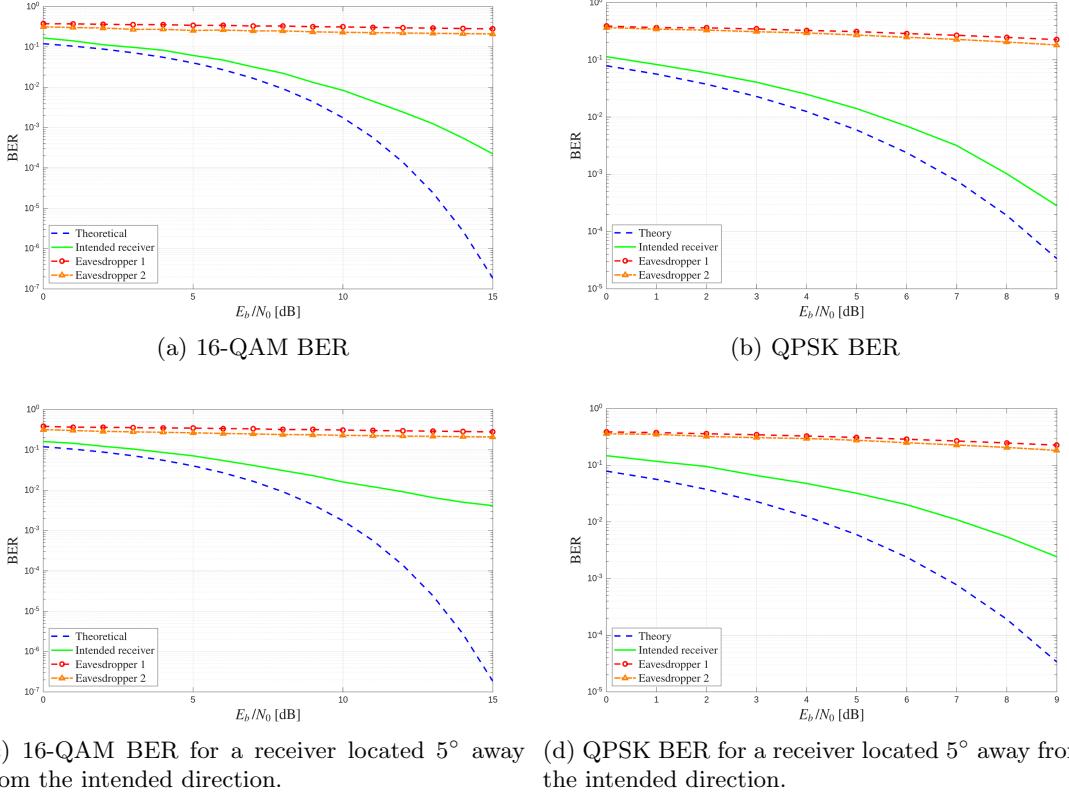


Figure 20: Comparison between QPSK and 16-QAM BER

As shown in Figure 20, the BER performance of QPSK and 16-QAM exhibits a clear trade-off between reliability for the intended receiver and spatial security against nearby eavesdroppers. When observing the upper panels, which represent the BER of the intended user, it is evident that QPSK achieves significantly better performance at the same E_b/N_0 level. For example, to reach a BER comparable to that of QPSK at $E_b/N_0 = 9$ dB, the 16-QAM scheme requires approximately $E_b/N_0 \approx 15$ dB, highlighting the intrinsically higher energy requirement of higher-order constellations.

Conversely, the lower panels show that 16-QAM provides a clear advantage in terms of spatial security. When the receiver is displaced by 5° from the assumed direction, the BER for 16-QAM increases significantly faster than that of QPSK. Even when the 16-QAM transmission power is adjusted so that the intended user achieves the same BER performance as QPSK at $E_b/N_0 = 9$ dB, the eavesdropper still experiences noticeably worse performance. Specifically, the eavesdropper BER for 16-QAM remains below 10^{-2} , whereas for QPSK at the same E_b/N_0 level the BER lies between 10^{-2} and 10^{-3} , approximately in the midrange between the two thresholds.

This behavior confirms that, although 16-QAM is less robust for the legitimate receiver at equal energy levels, its stronger sensitivity to angular misalignment significantly enhances spatial security. Therefore, if additional transmit power can be allocated to compensate for the reduced robustness, 16-QAM can simultaneously achieve the same BER performance as QPSK for the intended user while providing a considerably higher BER for any nearby eavesdropper. This makes higher-order constellations particularly appealing in when prioritizing security in angle-dependent communication scenarios.

2.5.4 Effect of Weight Vector Selection Margins on BER in Directional Modulation

As observed in the previously analyzed figures, a noticeable deviation exists between the theoretical (optimal) BER curve and the simulated one. This discrepancy arises from the fact that the MATLAB function `FindSteeringVector`, which is responsible for generating the weight vectors, does not necessarily compute the optimal solution. Instead, it selects a suboptimal steering vector, provided that it satisfies predefined amplitude and phase margin constraints.

In the previous simulations, these constraints were set to:

$$\text{azimuthWidth_deg} = 1^\circ, \quad \text{amplitudeTolerance} = 0.8 \text{ dB}, \quad \text{phaseTolerance_deg} = 20^\circ.$$

For the following analysis, we focus on a QPSK modulation scheme with a ULA of size `ulaSize` = 3. This choice is motivated by the significantly faster computation of the weight vectors for smaller arrays, as tightening the amplitude and phase margins becomes increasingly computationally demanding. While this comes at the cost of reduced spatial security performance for our use case, the observed effects are expected to become more pronounced as the number of antennas increases. Therefore, analyzing this smaller array provides a representative example of the trends and behaviors that would occur in systems with larger ULA sizes.

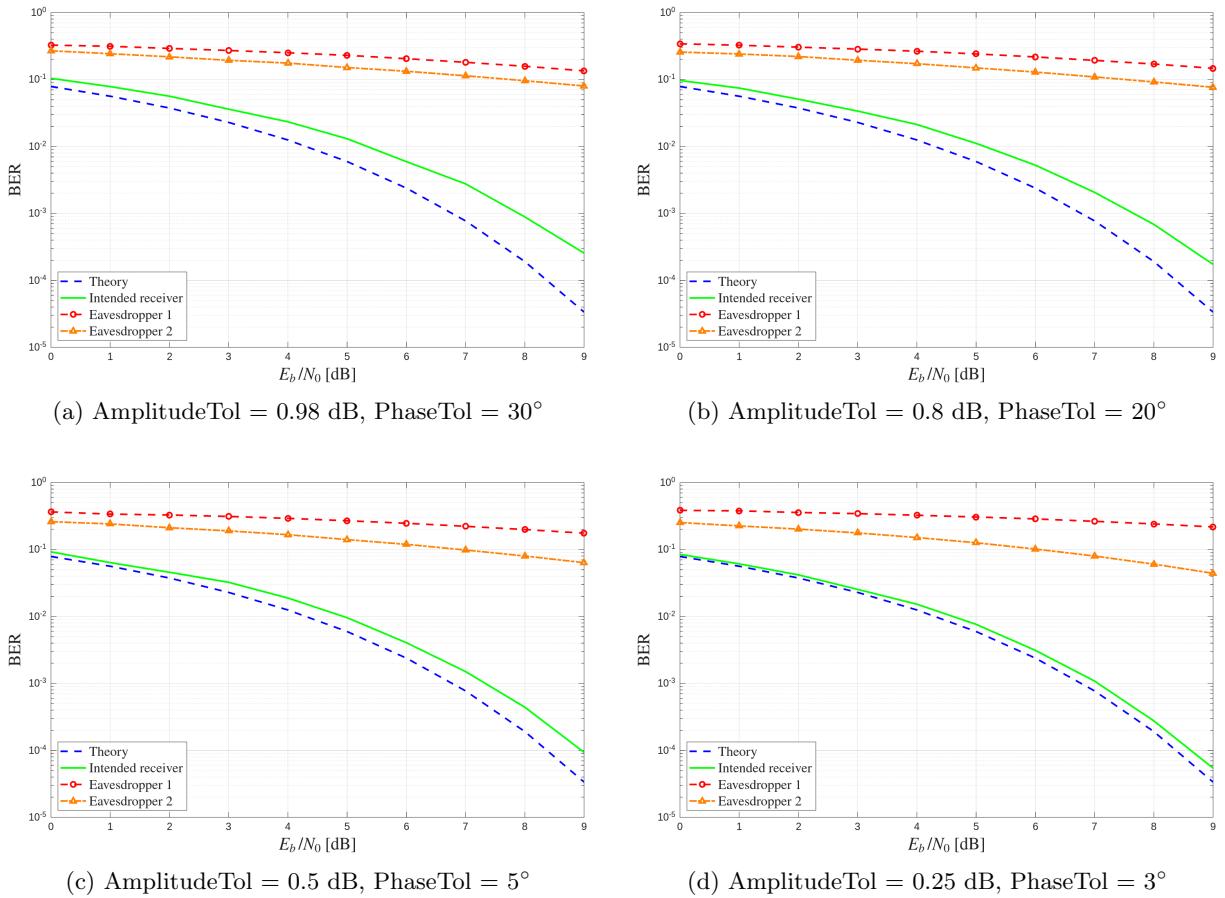


Figure 21: BER performance with 3 antennas under different amplitude and phase tolerances.

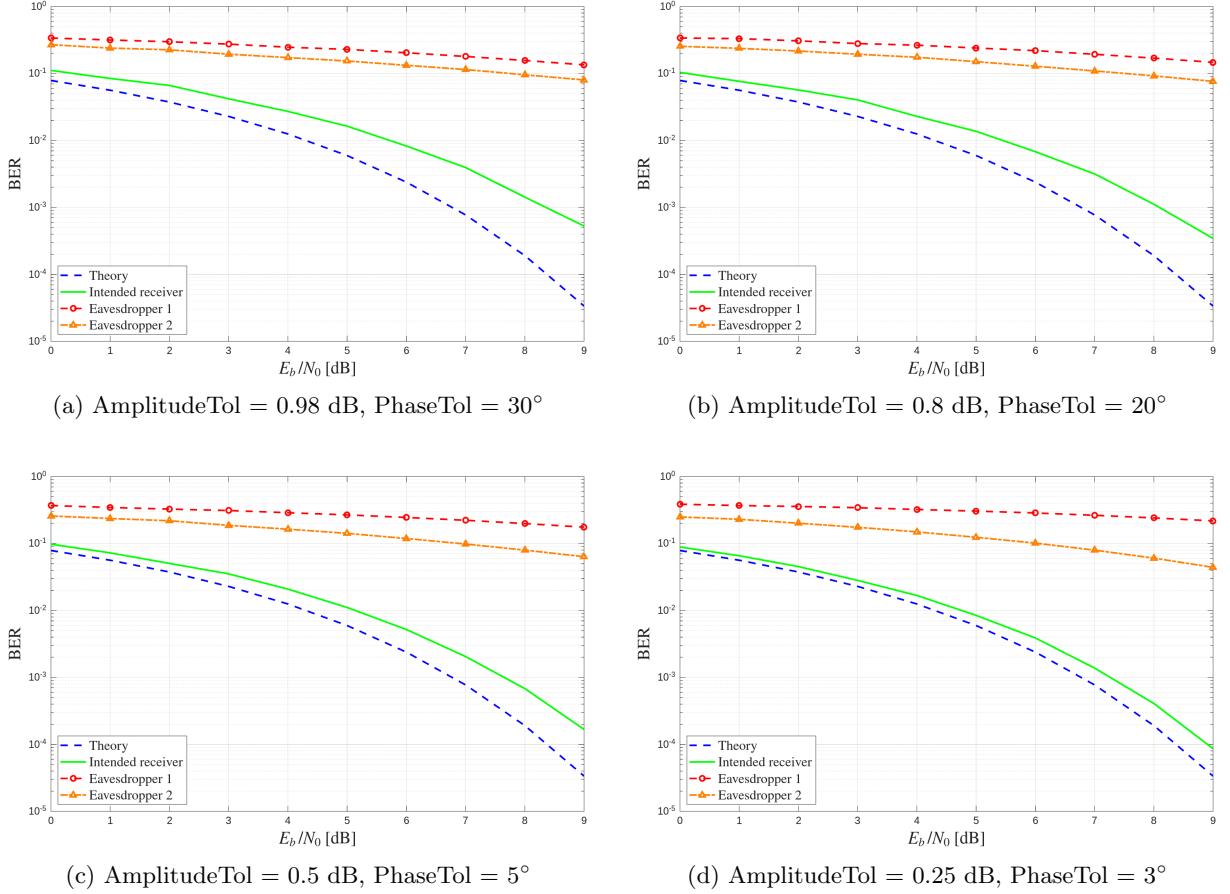


Figure 22: BER performance with 3 antennas under different amplitude and phase tolerances for a receiver located 5° away from the intended direction.

As expected, the figures confirm that as the amplitude and phase tolerances decrease, the discrepancy between the measured and theoretical BER curves becomes smaller. This is because fewer weight vectors satisfy the imposed constraints, which has several important implications.

Increasing the amplitude and phase margins improves the BER for both the intended receiver and for a user located near the intended direction. However, this comes at the cost of higher computational complexity, since generating valid weight vectors becomes more demanding. Additionally, a larger set of feasible vectors leads to a higher probability of vector duplication, which may reduce the system's variability.

Conversely, choosing suboptimal weight vectors represents a reasonable trade-off. While it slightly degrades the BER of the intended receiver, it also increases the BER for potential eavesdroppers, thereby enhancing spatial security. This effect is clearly illustrated in Figures 21 and 22, where more relaxed constraints result in higher BER for both the intended and unintended receivers, demonstrating that suboptimal solutions can be beneficial for overall system performance in terms of security.

3 Conclusions

3.1 Summary of Results

In this work, a qualitative analysis was conducted using a reduced number of antennas in the uniform linear array. This choice allowed us to clearly isolate and understand the fundamental behaviors of the technique without incurring the computational overhead associated with large-scale, practical antenna configurations. The main findings of this analysis can be summarized as follows.

Effect of the Quantization. When the quantization factor satisfies $B > 2$, the system already begins to exhibit the desired degradation in the signal received by an unintended user. Increasing B enlarges the set of unique weight vectors that the algorithm is able to generate. However, beyond a certain threshold (approximately $B = 18$ in our experiments), the time required to compute a single valid weight vector becomes excessively high. This effect becomes even more pronounced as the array size (`ulaSize`) increases, since the probability of finding a vector that satisfies the imposed amplitude and phase constraints decreases with the number of antennas.

Performance of QPSK and 16-QAM with Increasing ULA Size. As the number of antennas grows, the BER experienced by the intended receiver tends to diverge from the theoretical curve. This deterioration is mainly due to the fact that the algorithm must increasingly rely on suboptimal weight vectors when the constraints prevent an optimal solution from being found. At the same time, a larger number of antennas significantly improves spatial security by further degrading the signal observed by eavesdroppers positioned close to the intended direction, as well as those farther away.

Comparison Between QPSK and 16-QAM. The analysis reveals that 16-QAM generally produces a higher BER for the intended receiver compared to QPSK under the same E_b/N_0 . However, this behavior also extends to unintended receivers: for an eavesdropper located near the main beam, 16-QAM leads to a more distorted and dispersed constellation than QPSK. Consequently, the choice between these two modulation schemes depends strongly on the available E_b/N_0 . When sufficient energy per bit is available, 16-QAM may offer better spatial security, whereas QPSK remains more suitable under low-power operating conditions.

Influence of Amplitude and Phase Margins. Tightening the amplitude and phase tolerances used in the weight-vector selection algorithm results in improved BER for the intended receiver. However, this improvement also applies to an eavesdropper positioned near the intended direction, and it substantially increases the computational effort required to generate valid weight vectors. Importantly, using theoretically ideal margins (i.e., zero amplitude and phase tolerances) is not necessarily beneficial: such strict constraints drastically increase computational cost and may inadvertently enhance the signal quality observed by an eavesdropper. Therefore, accepting suboptimal solutions is not only practical but also advantageous, as it strikes a balance between computational complexity, communication performance, and spatial security.

Overall, the study demonstrates that inherently involves a trade-off among array size, quantization resolution, constraint strictness, and achievable spatial security. Despite using small arrays for clarity of analysis, the qualitative trends observed here are representative of larger systems and highlight the fundamental mechanisms that govern the effectiveness of directional modulation.

3.2 Final Considerations

The analysis presented in this document does not reflect the behavior of real-world antenna arrays, but rather provides a qualitative insight into the principles and potential of .

This technology could be applied, for instance, in satellite communications where maximum security of transmitted messages is required. In such scenarios, however, the antenna configuration is typically no longer a linear array but rather a planar or matrix array, and the number of antennas (`ulaSize`) is considerably larger than the small values considered here (e.g., 3, 5, or 7), potentially reaching 100–150 elements.

This implies an extremely high computational cost for generating secure weight vectors with DM. As a result, it is currently more practical to rely on conventional cryptographic techniques for ensuring message security.

If, in the future, an algorithm were discovered capable of generating the required weight vectors in an acceptable time, could become an excellent solution for secure communications. At present, however, the computational cost is prohibitive: even transmitting a single symbol securely requires an enormous amount of computation, making real-time implementation infeasible.

3.3 Future Work

As anticipated, the development of a more efficient algorithm for generating weight vectors is necessary. Another potential direction for future research is to make secure not only as a function of azimuth and elevation angles, as analyzed in this work, but also dependent on the distance between the transmitter and receiver.

Moreover, the performance of could be evaluated for planar arrays or matrix-type antenna configurations, bringing the analysis closer to realistic scenarios.

Finally, the introduction of coding techniques, such as turbo codes, could be investigated to further enhance security, ensuring that the signals received by an eavesdropper are even more distorted and difficult to decode.

References

- [1] V. Pellegrini, F. Principe, G. de Mauro, R. Guidi, V. Martorelli, and R. Cioni, “Cryptographically secure radios based on directional modulation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014.
- [2] A. Goldsmith, *Wireless Communications*, 1st ed. New York, NY: Cambridge University Press, 2005.
- [3] J. G. Proakis, *Digital Communications*, 4th ed. Boston, MA: McGraw-Hill, 2001.
- [4] R. L. Haupt, “Phase-only adaptive nulling with a genetic algorithm,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 6, pp. 1009–1015, Jun. 1997.
- [5] R. L. Haupt and R. A. Shore, “Experimental partially adaptive nulling in a low sidelobe phased array,” in *Proc. IEEE Antennas Propagat. Society Int. Symp.*, 1984, pp. 823–826.
- [6] R. J. Mailloux, *Electronically Scanned Arrays*, 1st ed. San Rafael, CA: Morgan & Claypool, 2007.
- [7] M. P. Daly and J. T. Bernhard, “Directional modulation technique for phased arrays,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 9, pp. 2633–2640, 2009.
- [8] E. Baghdady, “Directional signal modulation by means of switched spaced antennas,” *IEEE Transactions on Communications*, vol. 38, pp. 399–403, Apr 1990.
- [9] T. F. A. Villenueve and F. Terrio, “Ultra-low sidelobes from time-modulated arrays,” *IEEE Transactions on Antennas and Propagation*, vol. 11, pp. 633–639, 1963.
- [10] B. L. Lewis and J. B. Evans, “Technique for reducing radar response to signals entering antenna sidelobes,” *IEEE Transactions on Antennas and Propagation*, vol. 31, pp. 993–996, 1983.
- [11] S. Yang, Y. B. Gan, and P. K. Tan, “Linear antenna arrays with bidirectional phase center motion,” *IEEE Transactions on Antennas and Propagation*, vol. 53, pp. 1829–1835, May 2005.
- [12] S. D. Keller, W. D. Palmer, and W. T. Joines, “Electromagnetic modeling and simulation of a directly-modulated l-band microstrip patch antenna,” in *Proc. Int. Symp. on Antennas and Propagation*, Jun 2007, pp. 4489–4492.
- [13] C. M. Elam and D. A. Leavy, “Secure communication using array transmitter,” Patent US 6,275,679, Aug 14, 2001.
- [14] A. Babakhani, D. B. Rutledge, and A. Hajimiri, “A near-field modulation technique using antenna reflector switching,” in *Proc. IEEE Int. Solid State Circuits Conf.*, Feb 2008, pp. 188–189.
- [15] ——, “Transmitter architectures based on near-field direct antenna modulation,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2674–2692, Dec 2008.
- [16] F. Shu, W. Zhu, X. Zhou, J. Li, and J. Lu, “Robust secure transmission using main-lobe-integration based leakage beamforming in directional modulation mu-mimo systems,” *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–1, Nov 2017.
- [17] M. P. Daly and J. T. Bernhard, “Directional modulation technique for phased arrays,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 9, pp. 2633–2640, 2009.
- [18] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive mimo for next generation wireless systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb 2014.
- [19] V. Pellegrini, F. Principe, R. Guidi, G. Scozza, and G. de Mauro, “Adding phy-layer crypto to cofdm radios through a large array with directional modulation,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, 2018.
- [20] E. Björnson, J. Hoydis, and L. Sanguinetti, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*, ser. Foundations and Trends in Signal Processing. Now Publishers, 2017, vol. 11, no. 3–4.
- [21] F. Shu, X. Wu, J. Li, R. Chen, and B. Vucetic, “Robust synthesis scheme for secure multi-beam directional modulation in broadcasting systems,” *IEEE Access*, vol. 4, pp. 6614–6623, Oct 2016.