POLITECNICO
MILANO 1863

**Wireless Communication Project**

# *Adaptive Beamforming*

**Giovanni Vacirca – 953763**
**Andrés Felipe Rodríguez Vanegas – 915761**
**Gianpaolo Alesiani – 915405**

## ▪ *INTRODUCTION*

The following report includes explanations, comments and theoretical approaches to the MATLAB codes of Level D, C and B elaborated according to the instructions given in the Wireless Communications course (A.Y. 2019/2020). All these levels have been deployed completely, starting from the easiest configuration (Level D) to the most elaborated one (Level B) in which we use OFDM signals propagated in a two-ray channel with an LMS beamforming algorithm.

Every level is divided in sections, following the structure of MATLAB script. Some sections are gathered together.

## ▪ *LEVEL D*

**PARAMETERS – GEOMETRY – CREATION OF GEOMETRY SCENARIO**

In the first part of the script we define parameters and geometry that will be useful along all the sections.

From the parameter section we can handle many important values, such as number of interferences (we have set 2 interferences), the frequency of the carrier (1 GHz in our case) and number of antennas in the array of the Base Station (BS).

About the geometry, the rectangular characteristic of the antenna is defined through *phased.URA* function. After that, we set up the positions and characteristics of all the elements we are interested to (vehicles and interferences) choosing manually their initial positions and the desired arrival position. Obviously, the description made for vehicle 1 can be easily replicated for vehicle 2, just changing name of the variables and position.

The only constraints in terms of distances and tracks regarding the vehicles are the following:

1. In order to see the effects of the overlapping signals of the two vehicles, they must be at the same position at one point of the track e.g. both vehicles at position (50,50,1.5)
2. Since the normal vector of the BS array is directed in the positive direction of the x-axes, none of the vehicles could overpass the BS in this axis; if BS is in position (0,0,25), the second and third quadrant of the XY plane are forbidden for both vehicles, since we are just considering the positive radiated pattern of the antennas in the BS. In future works we can overpass this issue by adjusting the phase shift of the signal when it comes from this position, for example with the help of a PLL. This constraint must be respected in the cases which the transmitted signal is complex (Level C and B). In case the signal is real (sinusoidal, as for this Level D) this phase shift does not affect the result.

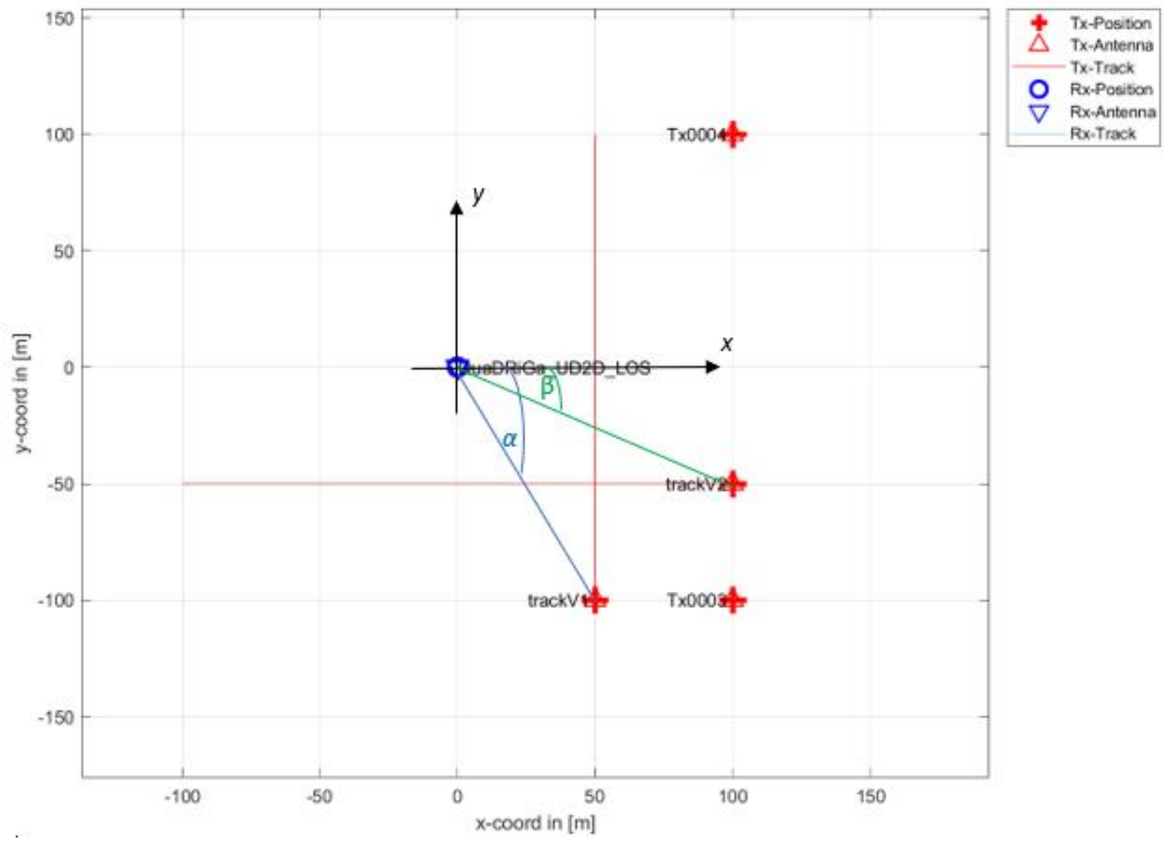


*Figure 1: 2D Geometry*

As also enlightened in *figure 1*, we describe an environment characterized by only two dimensions (2D, the third dimension will be studied starting from level C), and for this reason, all the elevation angles will be considered null (all elements find themselves at the same height). The two vehicles approach toward the direction of the antenna array and meet each other after 50 m. In the code we define the trajectory distance carried out by the vehicles and the progressive distance to the BS (in our case, a decreasing distance). In a second moment we find the Direction of Arrival (DoA), a vector composed by the Angle of Arrival (AoA, the azimuth angle, α for vehicle 1 and β for vehicle 2 in *figure 1*) and the elevation angle (ZoA), made by all null elements for this level D. The same work is then repeated for all the interferences (shown in the graph with labels *Tx0003* and *Tx0004*). Finally, the graph can be plotted with the section "*Creation of geometry scenario*".

## CREATION OF THE NARROWBAND SIGNAL – POWER OF THE TRANSMITTED AND RECEIVED SIGNAL

We want to set up two different sinusoids for the two vehicles and another one for interferences; for this reason, we assign different values for the frequency of the elements: 500 Hz for vehicle 1, 1000 Hz for vehicle 2 and 1500 Hz for the interferences. Exploiting these frequencies, we can now create the signals we need, giving the shape to the sinusoidal waveforms. We are dealing with a free space channel, so from the elements to the base station, the arrived signals will be attenuated according the following formula:

$$\mathrm{FSPL} = \frac{P_t}{P_r} = \left(\frac{4\pi d}{\lambda}\right)^2$$

*Figure 2: Equation of Free Space Path Loss*

At that point, it is interesting going to see how the signal transmitted by the elements arrive to the BS; after choosing a suitable sampling time, four graphs are plotted, showing the transmitted signals along the time of the two vehicles and the two signals received by the BS.

Then, a similar job is performed with the power: power transmitted and received by the elements are calculated and used to calculate the SINR for vehicle 1 and vehicle 2. For each point of the trajectory, we plot then the received power and the SINR.

**RECEIVED SIGNAL IN THE BASE STATION**

Now we need to collect all the arrived signals in the BS, in order to manage them. For this reason, the *collectPlaneWave* function is used, getting in input the columns of received signals by vehicles and interferences, the columns of the different AoA of every element for every position of the trajectory and the frequency of the carrier and light speed described in "*Parameters*" section. This gives us as result a signal described through a 3D matrix: one dimension is the signal, the second is in which antenna is approaching and the third is the position of trajectory in that moment. Afterwards, the function *awgn* is used to introduce the usual noise in our receiver.

The subsequent plot shows the resulting signals in the first 4 elements of the antenna array.

With this section we complete the description of the signals, from the creation to the receiver, passing through the channel.

**CALCULATED STEERED VECTOR – STEERING VECTOR BEAMFORMER**

At this point we can introduce the beamforming: we modify the transmission of the signals with the help of weights that can be calculated in different way. Introduction of the weights is done through the steering vector, a set of coefficients that are multiplying our signal.

In the conventional beamforming (the type of beamforming we are using) the phases are selected to steer the array in a particular direction (azimuth and elevation). In this section, the steered vector is calculated manually using the following formula:

$$S_n = e^{-j(kn \cdot d\ sin\phi)}$$

In fact, we can find on the MATLAB script the calculation of the wave vector (*k*), the distance between the antennas (*d*) and the AoA of the two vehicles ($sin\phi$).

With the parameter *n* we want to indicate the distance in modulus from the center of the array to the different antennas, going to separate the case in which number of antennas is even or odd.
Once we calculate the value for one row, we can just copy the values column wise, exploiting the fact we are working in 2D scenario. The same operations are then repeated for interferences.
In order to double check our results, we then calculate the steering vector using the MATLAB function *phased.SteeringVector*. Results are good and match with the steering vector we calculate without the function.

## BEAMFORMING NARROWBAND - PLOT ENVIRONMENT VEHICLES 1 – 2

In order the calculate the final signal, where the presence of beamforming is included, we just multiply the weights we found and the signal with noise included.
At this point we can finally plot in one polar graph the movement of the vehicles and appreciate how beamforming change the lobes of the signal, trying to follow the movement of the selected vehicle.

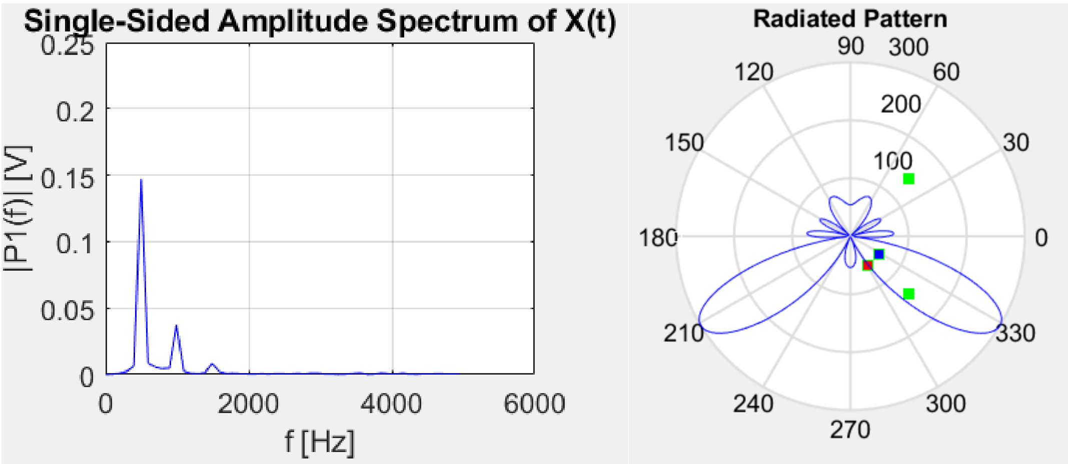


*Figure 3: FFT and radiation pattern of vehicle 1 for a generic point*

Another interesting contribute is given plotting the FFT; we can notice, in fact, a predominance of selection of the vehicle chosen, except for the moment when they cross each other: in *figure 4* we point out the behavior assumed by the single-sided amplitude spectrum of the signal in the above mentioned point.

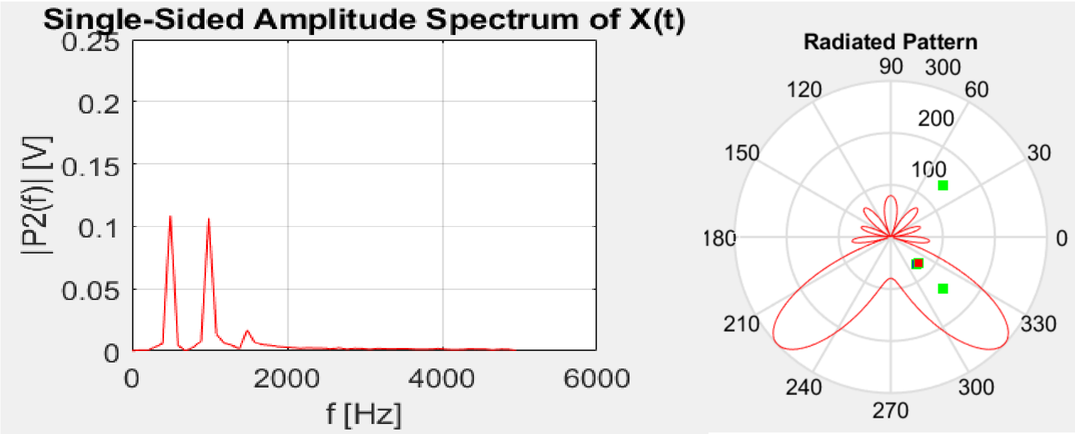


*Figure 4: FFT and radiation pattern of vehicle 2 - coincidence with vehicle 1*

## SINR – PHASE SHIFT BEAMFORMER

An important point of this level is calculating the resulting SINR for both the vehicles. This is done through a *plot* function, and SINR for every position of the trajectory is shown in a graph.

Finally, we use the *phased.PhaseShiftBeamformer* to compare the result we get in a graph, with the received waveform at the BS including noise and free space attenuation, and with the received signal with the steered vector beamforming.

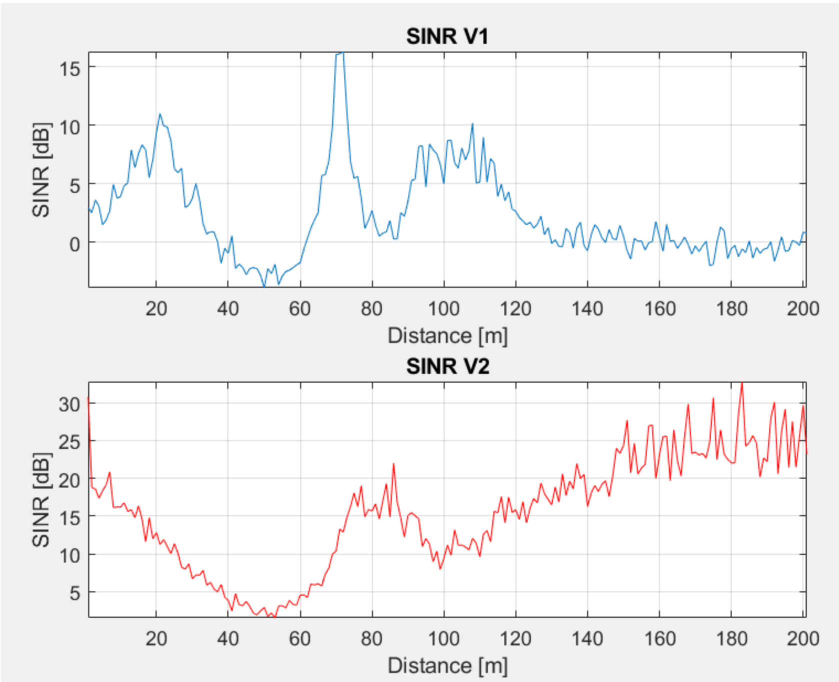


*Figure 5: SINR of both vehicles along the trajectory*

From *figure 5* we can appreciate the decrease of SINR in the exact point where the two vehicles cross each other, i.e. at distance 50 m.

- ## *LEVEL C*

### PARAMETERS – GEOMETRY – CREATION OF GEOMETRY SCENARIO

As in the previous level, here we start the script introducing parameters and geometry too; regarding the parameter section, everything remains the same of level D, while we introduce an upgrade in geometry section. In this case, we work in a 3D geometry, introducing the elevation (ZoA, 25 m for Base Station, 1.5 m for vehicles and interferences), element that was missing in the previous level. Subsequently, this will introduce an increasing complication in the calculation of the steering vector that will be treated in the "*Calculated Steered Vector*" section. Calculation of the ZoA is carried out through the arctangent, without exploiting the MUSIC algorithm, as requested. Angle of elevation can be appreciated in the *figure 6*, highlighted by α and β, respectively AoA of vehicle 1 and vehicle 2.

Also in this case, after the initialization of the measures and setting the values of the elements, we calculate trajectory of the two vehicles and direction of arrival of both vehicles and of interferences for all their positions.

The geometry scenario complies with vehicles movement and interferences positions.

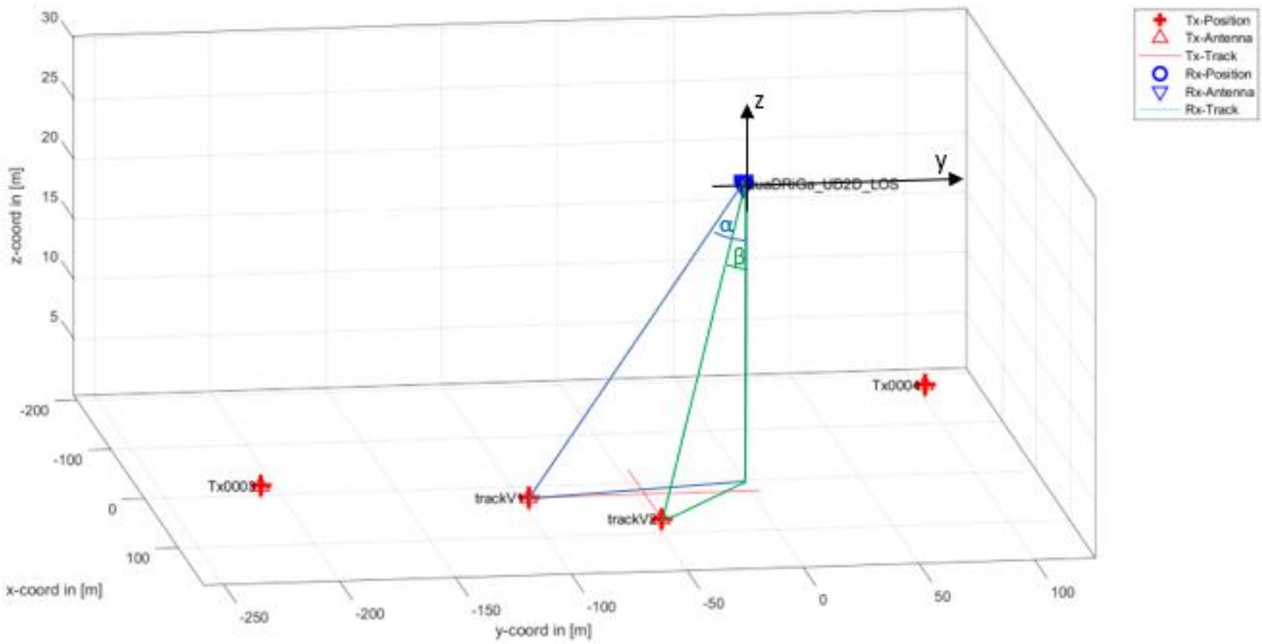


*Figure 6: 3D geometry*

### CREATION OF THE OFDM SIGNAL – POWER OF THE TRANSMITTED AND RECEIVED SIGNAL

In order to create the OFDM signal for vehicles and interferences the tool "*Wireless Waveform Generator*" is used; it also generates the MATLAB script. For these signals we specify the number of symbols, number of subcarriers, guard bands, position of the pilot and cyclic prefix.

The received signal and power for this level have been calculating considering only Free Space Attenuation given by the previous formula used for Level D; the MATLAB *phased.FreeSpace* function would have introduced a phase displacement that would have distorted the results.

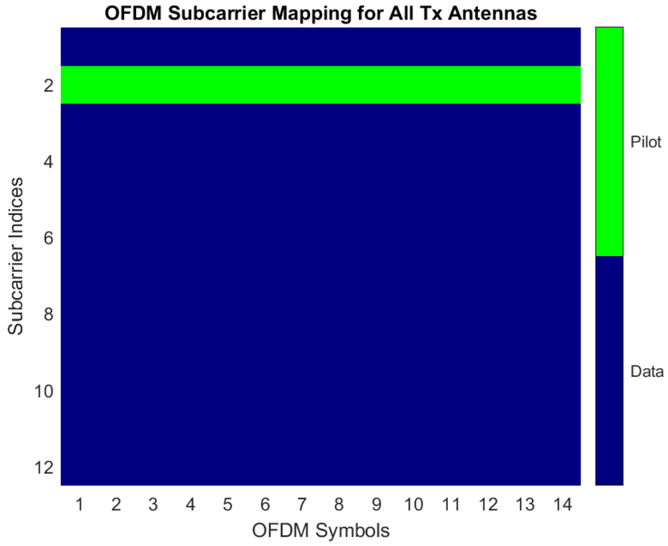


*Figure 7: 12 subcarriers, no guard bands, 2nd subcarrier for pilot.*

## CALCULATED STEERED VECTOR 3D – STEERING VECTOR

As said, we have to consider elevation angle for both vehicles. For an URA vector case literature [1] suggests a possible implementation taking advantage of the Kronecker product between two steered vectors, one considering the elevation case, the other considering azimuth case.

At the end we get a different value of the signal for each single antenna. Correct values can be found through the already existing function *phased.SteeringVector* provided by MATLAB, and it creates values of steering

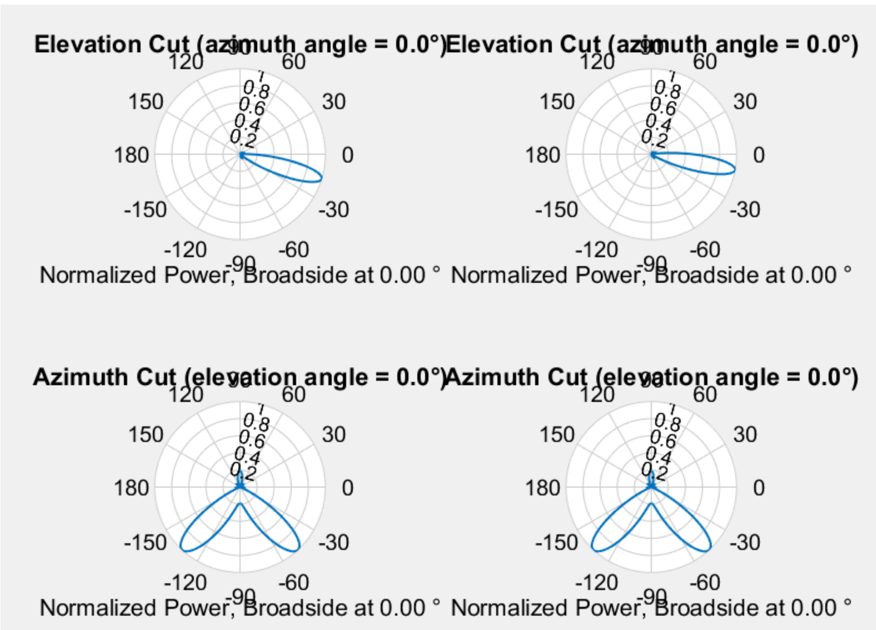


*Figure 8: Comparison between calculated weights by hand and calculated weights with MATLAB function for the first position*

vector that are comparable with the values calculated with the formulae. In *figure 8* we can appreciate in the first column the behavior of weights given by the function while in second column our weights.

### PHASED SHIFT BEAMFORMER - PLOT ENVIRONMENT & SCATTER PLOT FOR VEHICLES 1 – 2

For this level we consider not useful exhibit the results using FFT; we are, in fact, working with a multi-carrier signal, so we evaluate the correct operation of our beamforming with the scatter plot. We are able, indeed, to distinguish the 4-QAM modulation most of the times, with the exception when vehicles cross each other (in proximity of 50th point of the trajectory). Best results can be found when the vehicles have already crossed their paths and they are far away each other.

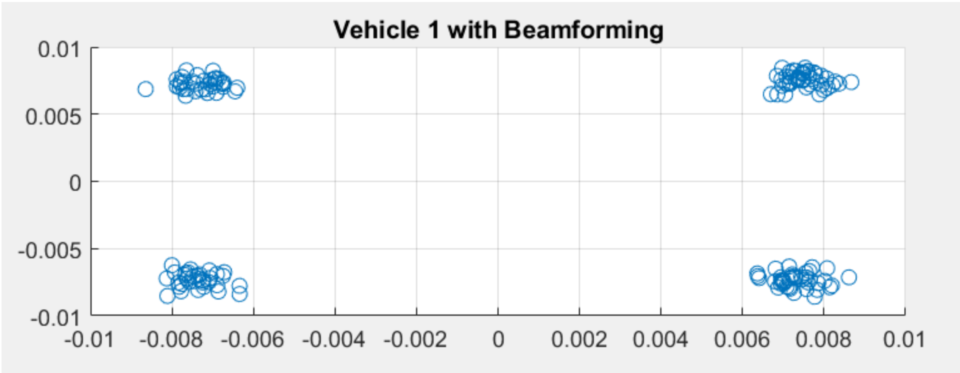


*Figure 9: Example of scatter plot with vehicles far apart each other*

### BER

We have evaluated the performances calculating Bit Error Rate and plot it in function of trajectory, so the ratio between the number of wrong bits received over the number of total bit transmitted. Around the cross position we can find a decay of the BER. At the end of the trajectory of vehicle 2 we find another decay of BER because the vehicle 2 pass over the normal axis of our base station that is not able to follow anymore the vehicle. In other case the BER is equal 0 and so the beamforming does its job.

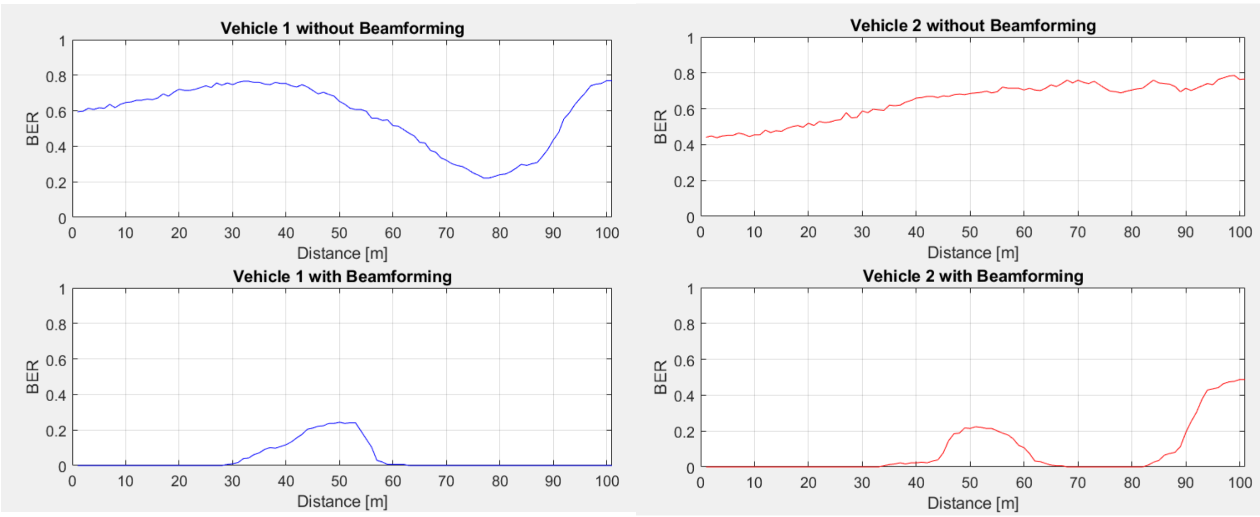


*Figure 10: BER for both the vehicles along all the positions*

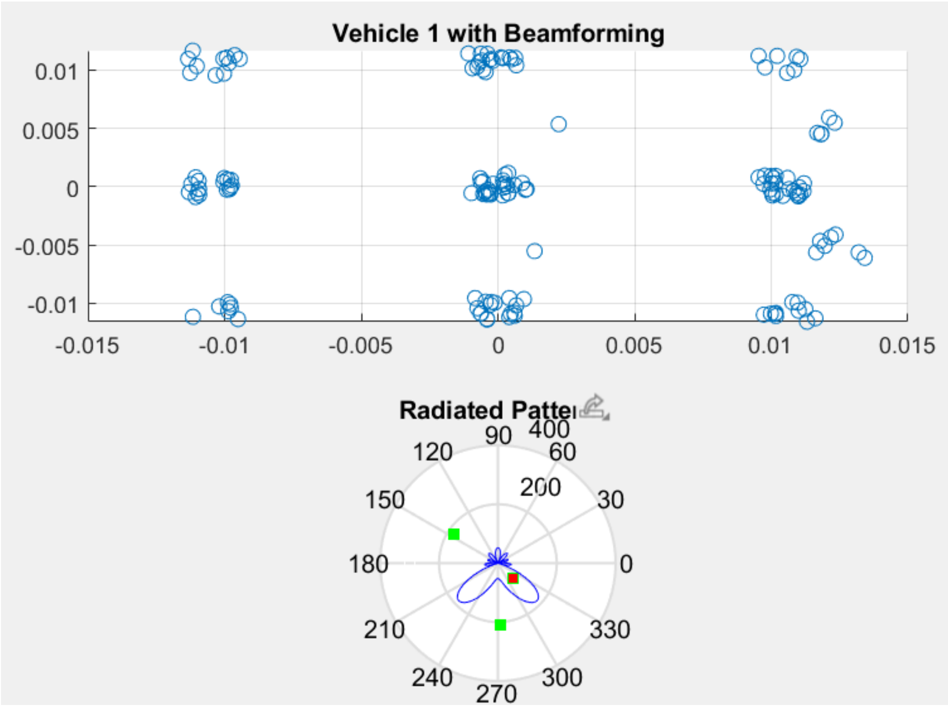

*Figure 11: Scatter and pattern of the particular case where vehicles are overlapped*

- ***LEVEL B***

### PARAMETERS – GEOMETRY – CREATE THE CONFIGURATION - CREATION OF THE OFDM SIGNAL

These sections remain unchanged from level C.

## *UPLINK CASE*

### POSITION ASSIGNMENT – CREATION OF TWO RAY MODEL

With this section we initialize some values and assign some variables that are used along the code, like positions and height of elements and Base Station.

About the channel, we introduce here the two-ray channel which replaces the free space propagation we used in the previous levels. Computation is carried out with the function *phased.WidebandTwoRayChannel*, which let us create the channel, imposing a ground reflection coefficient of -1.

At this point, after calculating the propagation signal and the received power at the BS, it is interesting showing received power at the antennas along all the positions taken by the vehicles.

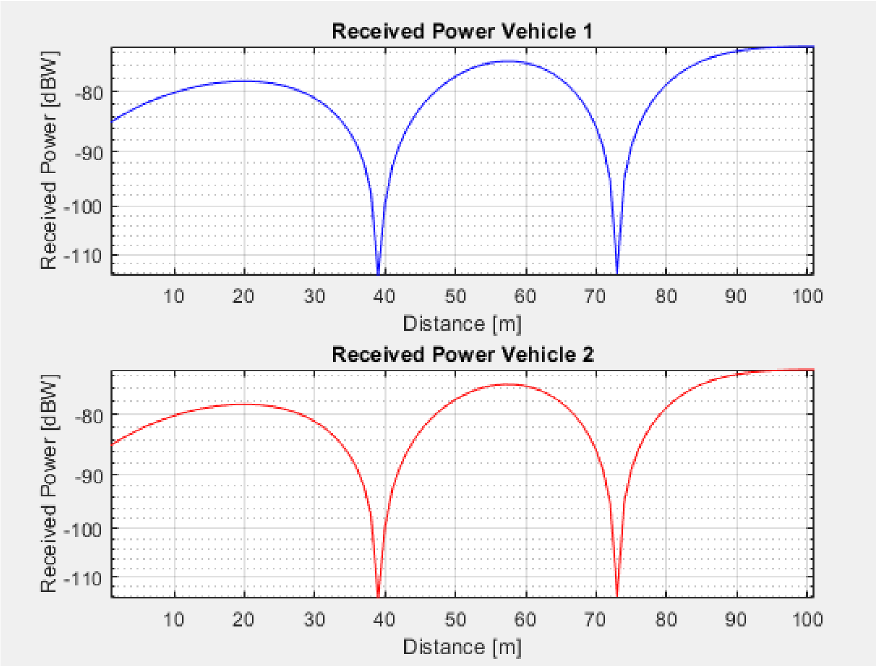


*Figure 12: Received power of the 2 vehicles for all the positions*

### RECEIVED SIGNAL IN THE BASE STATION

After working at the transmitter and at the channel side, we complete the description introducing the receiver side. With a *collectPlaneWave* function, signal is gathered at the base station and the noise is added. As for the previous levels, also in this level we plot the signal in the first 4 elements of the Base Station array (for the initial position of the vehicles).

## CALCULATED STEERED VECTOR – LAST MEAN SQUARE BEAMFORMING

The calculation of the steering vector is the same of Level C, but we can find differences in the adopted beamforming technique.

In this level we implemented LMS beamforming, which consists in a successive adjustment of the weights measured with the differences of the signal received and the transmitted signal. We implemented two different types of beamforming, the first one (LMS 1 in the code) follows the specification given by the theory in Godara paper [2] and based on the gradient of the minimum square error.

However, we did not find satisfactory results.

So, we proceed searching other algorithms and we decided to calculate the error from the subtraction between a reference signal (the original signal transmitted) and the expected signal after the propagation in the channel. Using this error, we can update the weights at each iteration.

This second algorithm has produced better results in terms of scatter plot and bit error rate.

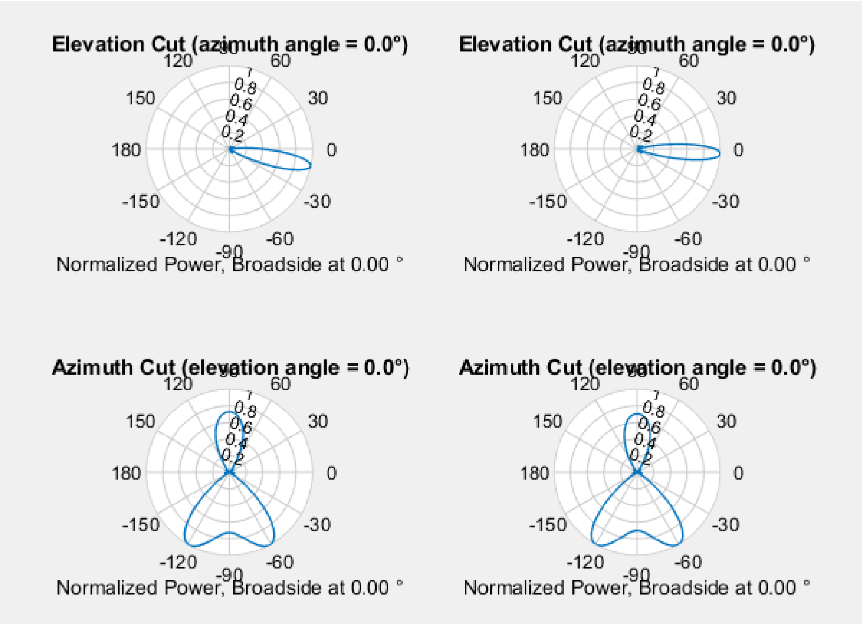


*Figure 13: Comparison between calculated weights by hand and calculated weights with MATLAB function for a fixed position*

## PHASED SHIFT BEAMFORMER - PLOT ENVIRONMENT & SCATTER PLOT FOR VEHICLES 1 – 2

The LMS signal is calculated multiplying the resulting signals at the receiver and the weights found in the previous section. At this point we have to demodulate the signal we got, and we can plot the result, as done in *figure 14* and *figure 15*.

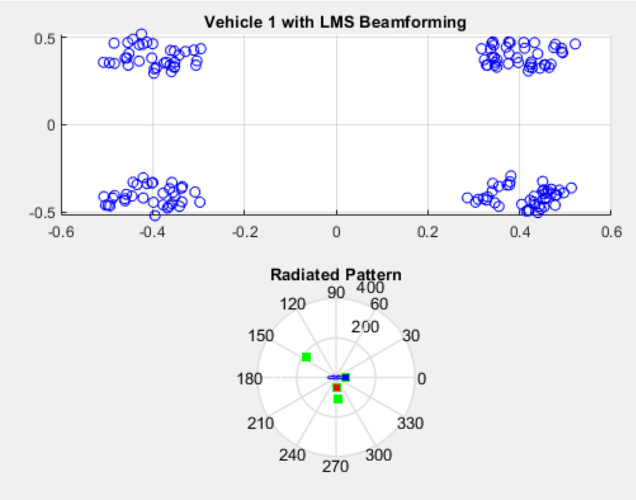


Figure 14: Example of scatter plot with vehicles not overlapped

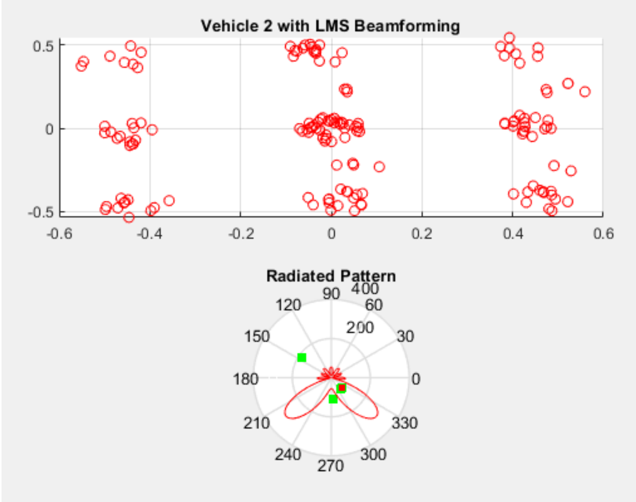


Figure 15: Scatter and pattern of the particular case in case vehicles are overlapped

**BER UPLINK**

It is easy with a graph to appreciate the great improvement generated by the introduction of the weights. In the following plots we can in fact see how the BER improve in this second case.

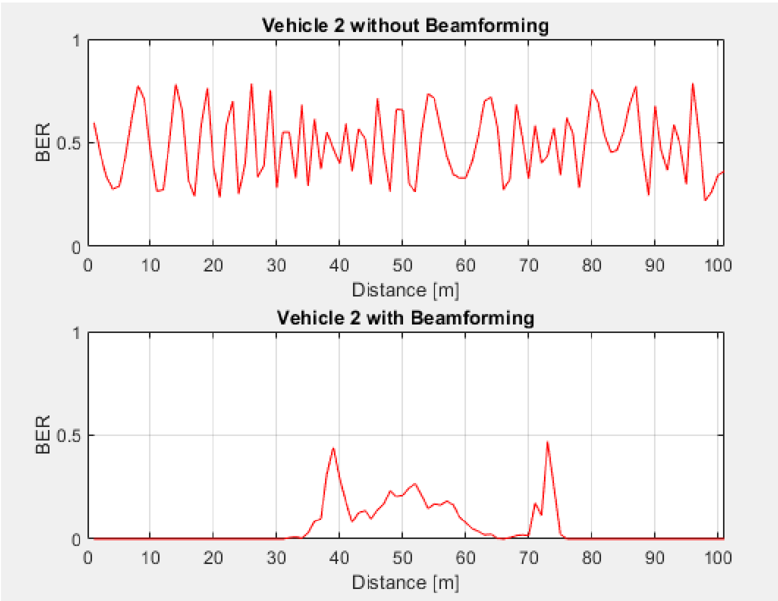


*Figure 16: BER without and with beamforming for vehicle 2*

## *DOWNLINK CASE*

### GEOMETRY

For the downlink scenario we have to take into account that now the signal emitted by interferences interfere with the reception at each vehicle, so in this section we have calculated, for each position of the vehicles, also the direction of arrival between interferences and each vehicle. Antenna is changed too: no more a rectangular array, but an isotropic antenna positioned in each vehicle.

### OFDM SIGNALS - POWER OF THE TRANSMITTED AND RECEIVED SIGNAL

The generation of signals for interferences has been done as in the previous case while no signals are created for vehicles. We consider, in fact, that when a vehicle is receiving the other is not transmitting. So, the main signal is transmitted by the base station. We consider the behavior of the BS as a unique equivalent antenna with the pattern due to the beamforming, so we create just one signal for base station instead of 25.

For the signal propagation we consider the same channel of the uplink (due to reciprocity) that is done with the *TwoRayChannel* function.

A big difference is found at the reception of the signal at two vehicles, that are two isotropic antennas. For this reason we have to change function since *collectPlaneWave* does not allow an isotropic element, so we have used the function *phased.Collector* that let us to receive all signals in an isotropic element.

In the figure below we can see the improvement due to beamforming in the received power at the vehicle 2.

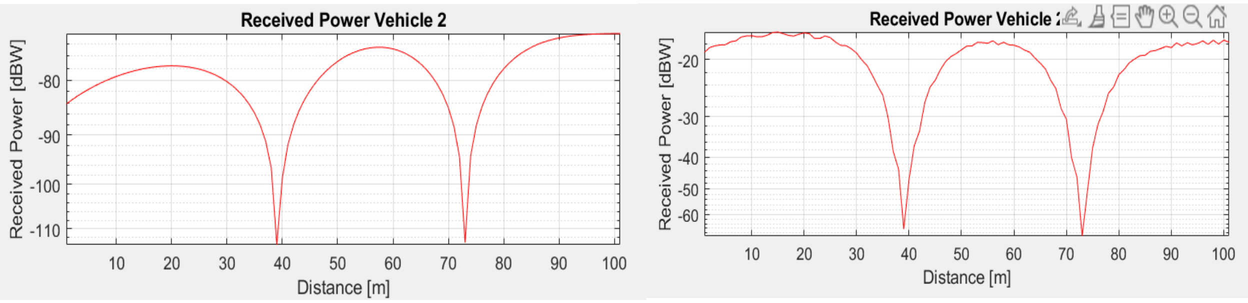


*Figure 17: Received power of all positions of vehicle 2 for all the positions without and with Beamforming*

### LAST MEAN SQUARE BEAMFORMING

In the downlink the beamforming has been done as in the uplink case, with the difference that we have changed the reference signal: now we have used as reference the transmitted signal from base station. Steps of algorithms are always the same.

### SCATTER PLOT FOR VEHICLES 1 – 2 & BIT ERROR RATE

To evaluate the correct working of downlink beamforming we plot the scatter figure for each vehicle at each position. We can distinguish the 4-QAM modulation clearer than in uplink case.

Around the position 40 we can see a decay in BER for both vehicles, caused by the cross position of vehicles. The other decay of BER around position 70-75 is due to the secondary lobe of radiation pattern.
Such as the clearer scatter plot in downlink case than in uplink case, also the BER is quite better than uplink. This improvement is probably due to the simpler scenario of the downlink where we do not consider anymore 25 five antennas for the base station.

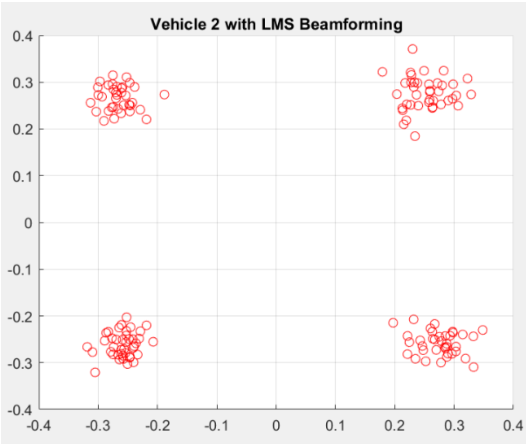


*Figure 18: Scatter plot Received Signal*


*Figure 19: Received BER with and without Beamforming*

## BIBLIOGRAPHY

[1] - Direction-of-Arrival Estimation for Uniform Rectangular Array: A Multilinear Projection Approach - Ming-Yang Cao, Xingpeng Mao, Xiaozhuan Long, and Lei Huang

[2] - Application of Antenna Arrays to Mobile Communications, Part II: Beam-Forming and Direction-of-Arrival Considerations